

Fault Diagnosis and Fault-Tolerant Control of Wind Turbines



by
Thomas Esbensen and
Christoffer Sloth

Title:

**Fault Diagnosis and
Fault-Tolerant Control of
Wind Turbines**

Theme:

Intelligent Autonomous Systems

Project period:

September 5th 2008 - June 3rd 2009

Project group:

09gr1030

Group members:

Thomas Esbensen
Christoffer Sloth

Supervisor:

Professor Jakob Stoustrup

Number of pages:

Main report: 129
Total: 167

Abstract:

Since many wind turbines are installed at remote locations, the introduction of fault diagnosis and fault-tolerant control is considered a suitable way of improving reliability of wind turbines and lowering costs of repairs. In this project, a 4.8 MW fictitious but realistic wind turbine is considered, for which a dynamical model is derived.

A fault analysis is conducted to identify the frequency of occurrence and the severity of the end-effects of possible component faults. Methods for diagnosis and accommodation of the most significant faults are then applied.

The diagnosis algorithms are based on a common structure relying on a reconfigurable extended Kalman filter, which allows diagnosis of multiple simultaneous faults. Generally, the abrupt faults are diagnosed using hypothesis testing based methods, while the incipient faults are diagnosed using parameter estimation based methods. The fault diagnoses algorithms can be used for both conditioning monitoring and active fault-tolerant control purposes.

Fault-tolerant capabilities are obtained by correcting the faulty signals or by incorporating fault-tolerance in the control system. Both active and passive fault-tolerant control systems are designed based on LPV methods, due to the parameter-varying nature of the wind turbine, and are compared in terms of design complexity and performance.

Verification of the control systems confirm that they are capable of controlling the wind turbine exposed to multiple simultaneous faults; consequently, the reliability of wind turbines can be improved.

Preface

This Master's thesis is written by two students, specializing in Intelligent Autonomous Systems, at the section for Automation and Control at Aalborg University. The thesis is the documentation of the work conducted in the period from September 5th 2008 to June 3rd 2009 and is focused on fault diagnosis and fault-tolerant control of wind turbines. The project group will like to thank Peter Fogh Odgaard from kk-electronic a/s for his help throughout the project, where he has supervised the group and provided model parameters.

References in the Thesis

- References to sources are referred to like [Esbensen *et al.*, 2008, p. 42], where the name of the author, year of publication, and page number are in square brackets.
- Figures and tables are referred to by the number of the object. In a reference like Figure 2.1 the first number refers to the chapter and the second number refers to the consecutive figure number of the chapter.
- References to other parts of the thesis are done by referring to the number of the chapter or section where the content is located, as in Section 1.4.
- Equations are referred to much like figures and tables, however, with the number enclosed in brackets; i.e. Eq. (3.6).

The nomenclature found on Page ii describes the notation and symbols specific to the wind turbine model.

A DVD is attached on Page 167 containing MATLAB scripts and Simulink models developed for the wind turbine system. Furthermore, model parameters and tuning parameters are also located here.

Thomas Esbensen

Christoffer Sloth

Nomenclature

The model parameters and variables are mentioned in the following list:

Symbol	Description	Unit
a	Radius of the tower	[m]
$a_t(t)$	Tower acceleration	[m/s ²]
A	Rotor swept area	[m ²]
B_{dt}	Torsion damping coefficient of the drive train	[Nm/(rad/s)]
B_g	Viscous friction of the high-speed shaft	[Nm/(rad/s)]
B_r	Viscous friction of the low-speed shaft	[Nm/(rad/s)]
B_t	Tower damping coefficient	[N/(m/s)]
$C_p(\lambda(t), \beta(t))$	Power coefficient	[·]
$C_t(\lambda(t), \beta(t))$	Thrust coefficient	[·]
$F_t(t)$	Thrust exerted by the wind on the rotor	[N]
$F_{t,i}(t)$	Thrust exerted by the wind on Blade i	[N]
$F_{th}(t)$	Force acting on the tower at hub height	[N]
$F_{th,i}(t)$	Force transferred to the tower from Blade i at hub height	[N]
h	Height of the tower	[m]
I	Turbulence intensity	[%]
J_g	Moment of inertia of the high-speed shaft	[kgm ²]
J_r	Moment of inertia of the low-speed shaft	[kgm ²]
k	Distance from the tower midline to the blade	[m]
K_{dt}	Torsion stiffness of the drive train	[Nm/rad]
K_t	Tower torsion coefficient	[N/m]
M_t	Top mass of the tower	[kg]
N_g	Drive train gear ratio	[·]
$P_a(t)$	Power captured by the rotor	[W]
$P_g(t)$	Power produced by the generator	[W]
$P_w(t)$	Power available from the wind	[W]
R	Radius of the rotor	[m]
r_0	Radius at which the blade profile begins	[m]
r_t	Distance from the hub to where the force acts on the blade	[m]
$T_{a,i}(t)$	Aerodynamic torque applied to the rotor by Blade i	[Nm]
t_d	Communication delay to the pitch actuator	[s]
$T_g(t)$	Generator torque	[Nm]
$t_{g,d}$	Communication delay to the converter	[s]
$T_{g,ref}(t)$	Reference for the generator torque	[Nm]
$T_h(t)$	Torque acting on the high-speed shaft	[Nm]
$T_l(t)$	Torque acting on the low-speed shaft	[Nm]
$v_{hub}(t)$	Wind speed at hub height	[m/s]
$v_r(t)$	Rotor effective wind speed	[m/s]
$v_{ts}(t)$	Wind speed tower shadow component	[m/s]
$v_{tu}(t)$	Wind speed turbulence component	[m/s]
$v_w(t)$	Wind speed including tower shadow, turbulence, and wind shear	[m/s]
$\bar{v}_w(t)$	Mean wind speed	[m/s]
$v_{ws}(t)$	Wind shear component	[m/s]
$x_t(t)$	Displacement of the nacelle from its equilibrium position	[m]
α	Empirical wind shear exponent	[·]
$\beta(t)$	Pitch angle	[°]
$\beta_{ref}(t)$	Reference to the pitch angle	[°]
ζ	Damping ratio of the pitch actuator model	[·]
η_g	Efficiency of the generator	[·]
$\theta_g(t)$	Angle of the high-speed shaft	[rad]
$\theta_r(t)$	Angle of the low-speed shaft	[rad]
$\theta_\Delta(t)$	Torsion angle of the drive train	[rad]
$\lambda(t)$	Tip-speed ratio	[·]

ρ	Air density	[kg/m ³]
σ_w	Standard deviation of the wind speed	[m/s]
τ_g	Time constant of the first order system	[s]
$\psi_i(t)$	Azimuth angle of Blade i	[rad]
$\omega_r(t)$	Rotor speed	[rad/s]
$\omega_g(t)$	Generator speed	[rad/s]
ω_n	Natural frequency of the pitch actuator model	[rad/s]

Throughout the thesis the following shorthand notation is utilized:

- For denoting the set-point to a controller, this reference value is written using 'ref' in the index. For instance, $\omega_{g,\text{ref}}(t)$ denotes the reference to the generator speed, $\omega_g(t)$.
- For denoting the innovation of a certain variable, the letter 'e' is used as index. For instance, $\omega_{g,e}(t)$ denotes the error defined as $\omega_{g,\text{ref}}(t) - \omega_g(t)$.
- For denoting the measurement of a given variable, the index is extended by 'mes'. For instance, $\omega_{g,\text{mes}}(t)$ denotes the measurement of the generator speed.
- Estimates of variables emerge by writing a hat above the symbol. For instance, $\hat{\omega}_{g,\text{mes}}(t)$ denotes the estimated generator speed.
- The symbol $v(t)$ is used to denote measurement noise. For instance, $v_{\omega_g}(t)$ is a zero-mean Gaussian distributed noise sequence with variance $\sigma_{\omega_g}^2$.
- The notation does not explicitly indicate whether a variable is a small-signal value or large-signal value. This is stated only in the linearization appendix, but is not used elsewhere in order to simplify the notation.
- To shorten the notation in symmetric matrices the following notation borrowed from [Bianchi *et al.*, 2007, p. 161] is utilized, where M and P are symmetric matrices:

$$\begin{bmatrix} M + N + (*) & * \\ Q & P \end{bmatrix}$$

will be read

$$\begin{bmatrix} M + N + N^T & Q^T \\ Q & P \end{bmatrix}$$

Contents

Preface	i
Nomenclature	ii
Contents	iv
1 Introduction	1
1.1 Background	1
1.2 Introduction to Fault Diagnosis and Fault-Tolerant Control	1
1.3 Project Scope	3
1.4 Thesis Outline	4
2 System Description	5
2.1 Wind Turbine Components	5
2.2 General Control Strategy	6
3 Wind Turbine Modeling	9
3.1 Model Structure	9
3.2 Wind Model	10
3.3 Aerodynamic Model	13
3.4 Drive Train Model	16
3.5 Tower Model	17
3.6 Power System Model	19
3.7 Pitch System Model	20
3.8 Neglected Dynamics of the Model	21
3.9 Assembled Model	22
3.10 Sensor Parameters	23
4 Fault Analysis	25
4.1 Model Partitioning	26
4.2 Fault Propagation Analysis	27
4.3 Fault Assessment	31
4.4 Structural Analysis	33
4.5 Fault Specification	39
4.6 Remedial Action Selection	42
5 Fault Diagnosis	45
5.1 Reconfigurable Wind Speed Estimator	45
5.2 Reconfigurable Extended Kalman Filter	50
5.3 Diagnosis of Single Pitch Sensor Bias	52
5.4 Diagnosis of Multiple Pitch Sensor Biases	61
5.5 Estimation of Pitch System Dynamics	64
5.6 Detection of Hydraulic Leakage	70
5.7 Detection of Stuck Pitch Actuator	71
5.8 Diagnosis of Proportional Speed Error	81
5.9 Detection of Fixed Output from Generator Speed Sensor	87

6 Fault-Tolerant Control by Signal Correction	89
6.1 Accommodation of Single Pitch Sensor Bias	89
6.2 Accommodation of Multiple Pitch Sensor Biases	91
6.3 Accommodation of Proportional Speed Error	93
6.4 Accommodation of Fixed Output from Generator Speed Sensor	94
7 Fault-Tolerant Control by LPV Methods	97
7.1 Introduction to Active and Passive Fault-Tolerant LPV Control Methods	97
7.2 Active Fault-Tolerant LPV Controller Design Procedure	100
7.3 Passive Fault-Tolerant LPV Controller Design Procedure	104
7.4 Common LPV System Description	109
7.5 Active Fault-Tolerant Controller Design	112
7.6 Passive Fault-Tolerant Controller Design	114
7.7 Comparison of AFTCS and PFTCS	116
8 Integration Test	121
9 Conclusion	123
Bibliography	127
A Severity and Occurrence Analysis	131
A.1 Severity and Occurrence Analysis of Rotor Sub-Model	131
A.2 Severity and Occurrence Analysis of Drive Train Sub-Model	137
A.3 Severity and Occurrence Analysis of Power System Sub-Model	140
B Linearization	145
B.1 Aerodynamic Torque Applied to the Rotor	145
B.2 Thrust Acting on the Rotor	146
B.3 Output Power	148
C Reference Controller	149
C.1 Controller Operating in Partial Load Operation	149
C.2 Controller Operating in Full Load Operation	151
C.3 Drive Train Stress Damper	157
C.4 Bumpless Transfer	158
D Derivation of Reliable LPV Controller Design Procedure	161
DVD	167

1 Introduction

In this chapter the background and motivation for the need of fault diagnosis and fault-tolerant control of wind turbines are described. A brief overview of fault diagnosis and fault-tolerant control and the application to wind turbines is then given. This is followed by presenting the scope of the project and outlining the content of the thesis.

1.1 Background

Evolution of technology has increased power demands to operate the modern electrical equipment. This has increased the demand for fossil fuels and has made electrical energy more expensive. Because of such high demands for electric power, it is necessary to focus on renewable energy sources, as fossil fuel resources are limited. Furthermore, to protect the environment the emissions of greenhouse gases and undesired particles into the atmosphere have to be reduced.

Among the renewable energy sources available today, wind power is the world's fastest growing [Wind Energy News, 2007]. With an annual growth rate in installed wind energy capacity of 30% on average throughout the past 10 years, wind turbines are definitely up and coming [GWEC, 2009, p. 15]. For several reasons wind energy is growing fast: it is cheap, inexhaustible, widely distributed, clean, and climate friendly [Wind Energy News, 2007].

As many wind turbines are installed offshore, a non-planned service can be highly costly, so it would be beneficial if fault-tolerant control schemes could help the turbines produce some energy from the time a fault is detected to the next planned service. Furthermore, the implementation of fault diagnosis schemes entails operational benefits due to its feature of early detection of faults, which can make the wind turbine operate safer and reduce costs as a result of possible improved maintenance procedures [Hameed *et al.*, 2009, p. 3]. Therefore, fault diagnosis and fault-tolerant control of wind turbines may offer several benefits:

- Prevent catastrophic failures and faults deteriorating other parts of the wind turbine by early fault detection and accommodation.
- Reduce maintenance costs by avoiding replacement of functional parts, by applying condition-based maintenance instead of time-based maintenance.
- Provide diagnostic details to the maintenance staff by remote diagnosis.
- Increase energy production when a fault has occurred by means of fault-tolerant control.

This section has addressed benefits of exploiting wind power and improving the reliability of wind turbines using fault-tolerant control. The next section gives a brief introduction to fault diagnosis and fault-tolerant control and their current application to wind turbines.

1.2 Introduction to Fault Diagnosis and Fault-Tolerant Control

The purpose of this section is to give an introduction to fault diagnosis and fault-tolerant control, since these topics are addressed in this thesis. This is accomplished by providing a brief overview of the terminology and available methods in these fields. Finally, the available fault diagnosis and fault-tolerant control algorithms for wind turbines are discussed.

Terminology

In this subsection the terms used in relation with fault diagnosis and fault-tolerant control are explained, to avoid any confusion about the terminology used throughout this thesis. The terms which are explained are highlighted using bold and italic fonts.

A ***fault-tolerant control system*** is a system, which prevents component failures from becoming failures on the system level. The control system is though allowed to have degraded performance in some cases when exposed to a fault. A ***fault*** is a change in the characteristics of a component, while a ***failure*** makes a component completely dysfunctional. Overall, there are two different types of fault-tolerant control systems; these are called passive fault-tolerant control systems (PFTCS) and active fault-tolerant control systems (AFTCS) [Zhang and Jiang, 2003].

Passive fault-tolerant control systems are designed to be resilient to a specified set of faults. This implies that the same controller is utilized both for the fault-free as well as the faulty system. In the design of passive fault-tolerant control systems, different performance requirements are set up for the normal system and for the faulty system [Niemann and Stoustrup, 2005b]. Therefore, these systems are not referred to as robust systems, but as ***reliable systems***.

Active fault-tolerant control systems have, in contrast to passive fault-tolerant control systems, different controllers for the normal system and for the faulty system. This implies that the state of the system has to be determined by fault diagnosis algorithms. The information from the fault diagnosis algorithms is utilized in a ***supervisor***, to reconfigure the control system for accommodating faults.

Fault diagnosis used in active fault-tolerant control systems consists of multiple parts, since faults both have to be detected, isolated, and in some cases estimated. ***Fault detection*** should detect that a fault has occurred and can rely on either an active or a passive approach. ***Passive fault detection*** should detect faults by comparing the expected system behavior with the observed system behavior; hence, it does not affect the system. In contrast to this, ***active fault detection*** uses injection of auxiliary signals into a system to improve the fault detection capabilities or in some cases make fault detection possible. ***Fault isolation*** should point out faulty components in the system. This is important information when faults should be accommodated, since the control system cannot rely on a faulty component. Some faults do not turn a component on or off, but have an intermediate state. This implies that ***fault estimation*** has to determine the fault sizes in order to accommodate these.

There are generally two types of faults: ***abrupt faults*** and ***incipient faults***. An abrupt fault is generally easier to detect than an incipient fault, but it might have severe consequences for the system, since it happens instantaneously.

Existing Fault Diagnosis and Fault-Tolerant Control Methods

There exists several methods for designing fault diagnosis algorithms and fault-tolerant controllers, and the basics of these are outlined in this subsection.

In the design of a passive fault-tolerant control system a good performance for the nominal control system has to be achieved while a graceful degradation is allowed in the case of a fault. In [Niemann and Stoustrup, 2005b] this is achieved by creating a controller structure relying on two separate controllers. One controller outputs nothing when the control system possesses nominal behavior, while the second controller equals the nominal controller. In the case of a fault, the first controller outputs a non-zero value; hence, changing the behavior of the control system. Other methods, as e.g. [Liao *et al.*, 2003], rely on a multi-objective control system, which has a set of minimum requirements to the faulty system and are optimized to improve the performance of the normal system.

In the design of an active fault-tolerant control system the first step is to design a fault diagnosis system. This essentially consists of designing a residual generator which is sensitive towards faults and insensitive towards other exogenous inputs to the system. Methods for this include parity space approaches where, if possible, a perfect decoupling between disturbances and residual is designed. Another approach is to design a change detection algorithm, e.g. based on a CUSUM test, which is able to detect a change in the mean value of a signal. Finally, Kalman filter approaches can be utilized by making a description of the fault become part of the system model, allowing the fault to be estimated. These approaches are suitable for diagnosing incipient faults.

When the fault has been diagnosed the active fault-tolerant control system must be reconfigured. This could for example be to reconfigure the controller to rely on estimates instead of measurements. The active fault-tolerant control system is reconfigured by use of a supervisor, which chooses an appropriate controller from a family of possible controllers, designed for each fault state.

Fault Diagnosis and Fault-Tolerant Control of Wind Turbines

In this subsection the current state of fault diagnosis and fault-tolerant control of wind turbines is outlined, by examining the available literature.

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

Most condition monitoring systems and fault detection systems in wind turbines are signal-based and utilize e.g. vibration analysis to detect and isolate faults. This has enabled successful condition monitoring of bearings in the gearbox and the generator among others. Numerous other signal-based approaches utilized in wind turbines can be found in [Hameed *et al.*, 2009].

Only a few model-based fault diagnosis approaches exist for wind turbines; among these are fault diagnosis systems for pitch sensors and pitch actuators [Wei and Verhaegen, 2008] and [Donders, 2002]. These diagnosis systems estimate some parameters in the pitch system, and determine if a fault has occurred based on these estimates.

It has not been possible to find any fault-tolerant control systems for wind turbines in the literature review. The common approach is to deploy condition monitoring systems and shut down the wind turbine in case of a fault. However, in a few cases thoughts about fault accommodation have been presented, but have not been tested or simulated.

In this section the terminology and available methods used in the fields of fault diagnosis and fault-tolerant control have been outlined. Additionally, fault diagnosis and fault-tolerant control applied to wind turbines have been investigated. The investigation has revealed that fault diagnosis algorithms exist for wind turbines, but mostly using signal-based methods. Additionally, only a few fault-tolerant control systems for wind turbines have been found. In the next section the scope of the project is described.

1.3 Project Scope

The overall scope of this project is to apply methods which can improve the reliability of wind turbines. This is accomplished by considering the model of a fictitious but realistic variable-speed, variable-pitch 4.8 MW wind turbine, provided by the collaborator kk-electronic a/s.

The project aims at improving the reliability of wind turbines by applying application-specific methods for model-based fault diagnosis and fault-tolerant control. This focus is chosen, since the literature survey in the previous section shows that research in these two areas are lacking.

Fault Diagnosis

As explained in the previous section some fault diagnosis algorithms have already been developed for wind turbines, where the main effort has been on signal-based fault diagnosis. This inspires for an investigation of the benefits of applying model-based fault diagnosis, which is performed in this project. It is expected that the introduction of model-based fault diagnosis will improve performance of the diagnosis system, since information about the control signals and the model is added to the information utilized in signal-based diagnosis methods.

The fault diagnosis approach should only use the main sensors associated with the control of the wind turbine excluding vibration and temperature sensors etc. Furthermore, the application-specific characteristics should be utilized in the fault diagnosis algorithms.

Fault-Tolerant Control

The project should also cover the design of fault-tolerant control systems, which should be resilient to a subset of the possible faults on the wind turbine. Additionally, a comparison between an active and a passive fault-tolerant control system should be accomplished for the relevant faults.

The faults that should be handled must be chosen based on how frequently they appear and how severe their effects are. The extent of this project prevents the designed control system to be tested

on a physical wind turbine; hence, simulations using a non-linear model are used instead. This limitation also has an effect on the faults which are able to be simulated on the considered model. Therefore, primarily sensor and actuator faults are considered.

The above presentation summarizes the scope of this project. In the subsequent section, the contents of the thesis are outlined.

1.4 Thesis Outline

The content of this thesis is divided into a number of chapters which are outlined in this section to provide an overview of the thesis.

Chapter 2: System Description

The purpose of this chapter is to introduce the wind turbine that is considered in the project, and to explain how a wind turbine is typically controlled. This forms the basis of the design of a reference controller described in Appendix C, which controls the considered wind turbine model unless otherwise stated.

Chapter 3: Wind Turbine Modeling

The purpose of this chapter is to derive a model of the wind turbine by modeling the most significant dynamics of the system. The model is linearized in Appendix B and a state space model is set up in the chapter to be used in the linear design methods presented throughout the thesis.

Chapter 4: Fault Analysis

The purpose of this chapter is to identify possible faults that can happen to the wind turbine and determine their impact on the system behavior based on simulation results available in Appendix A. A number of these faults are then selected to be diagnosed and accommodated in the project. Finally, the remedial actions that must be conducted to diagnose and accommodate the faults are outlined.

Chapter 5: Fault Diagnosis

The purpose of this chapter is to provide the diagnosis algorithm design and verification, which provide the sufficient information to decide an appropriate accommodation of the diagnosed faults. The fault diagnosis algorithms are only used by the active fault-tolerant controller, since the passive approach does not rely on a diagnosis scheme.

Chapter 6: Fault-Tolerant Control by Signal Correction

The purpose of this chapter is to provide the fault accommodation design and verification of the algorithms, which utilizes signal correction to obtain fault-tolerance capabilities. All the developed algorithms depend on estimates provided by the fault diagnosis system.

Chapter 7: Fault-Tolerant Control by LPV Methods

The purpose of this chapter is to design controllers which are able to accommodate faults which either change the dynamics of the system or reduce the available measurement information. Both active and passive fault-tolerant controllers are designed, to compare different approaches to identical faults.

Chapter 9: Conclusion

The main conclusions of this thesis as well as the recommendations for further research are given in this chapter.

This chapter has provided a motivation for considering fault diagnosis and fault-tolerant control of wind turbines. Additionally, the scope of the project has been set up, and has been put into the context of the current available methods and their application to wind turbines. Finally, an overview of the contents of the thesis has been provided. In the next chapter the components of a wind turbine and the typical control strategy are introduced.

System Description

The purpose of this chapter is to describe the components of the wind turbine that is considered in the project, and to explain how the wind turbine is typically controlled to maximize the performance dependent on the wind speed. Based on this description a reference controller is designed according to the basic principles of wind turbine control. The actual controller design is described in Appendix C, since the main focus of this thesis is fault-tolerant control, not nominal control of the wind turbine.

2.1 Wind Turbine Components

This section provides some basic knowledge about a wind turbine and its components. Notice that the section is inspired by [Esbensen *et al.*, 2008, pp. 12-13] and is only slightly modified.

The wind turbine considered in this project is selected by kk-electronic a/s and is a Danish concept wind turbine, which tends to be the standard design of modern wind turbines. A Danish concept turbine is a horizontal-axis wind turbine using a three-bladed rotor design with an active yaw system keeping the rotor oriented upwind [Krohn, 2002, p. 5].

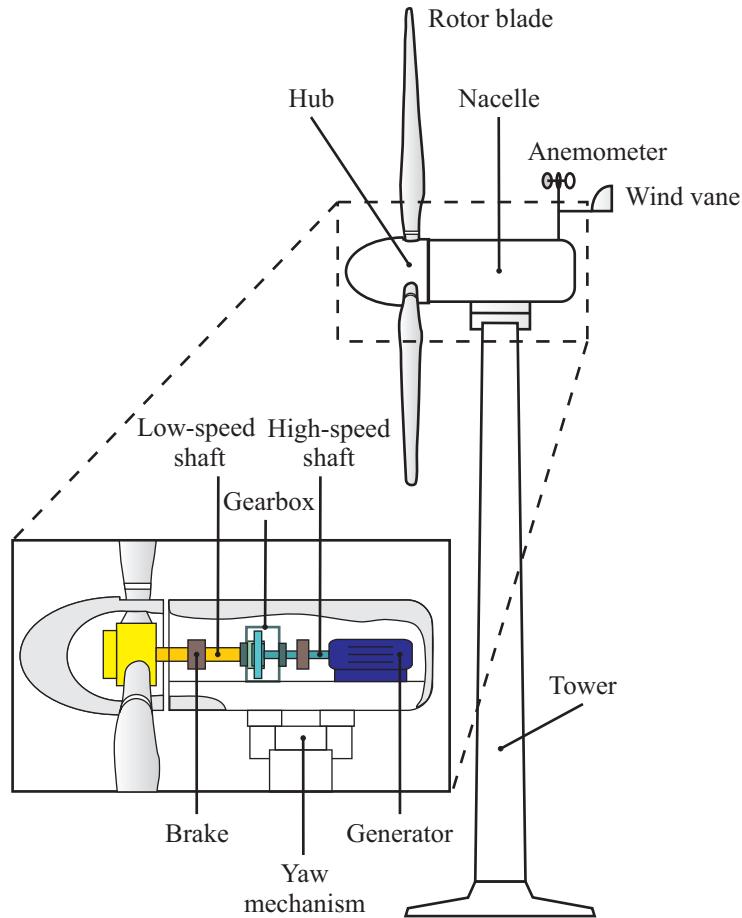


Figure 2.1: Main components of a horizontal-axis wind turbine.

Figure 2.1 illustrates the main components of a wind turbine and their interconnections. The figure is strongly inspired by [How Stuff Works, 2006]. The components and their purposes are described below in alphabetic order [The Encyclopedia of Alternative Energy and Sustainable Living, 2005].

- **Anemometer** is used to measure the wind speed. The wind turbine is started when the wind speed reaches a lower limit, while operation is cut-out when wind speeds become too high.
- **Brakes** can be applied mechanically, electrically, or hydraulically and function as parking brakes.
- **Gearbox** connects the low-speed shaft to the high-speed shaft, thus increasing the rotational speed to a level required by the generator to produce electric energy.
- **Generator** converts rotational energy into electric energy. On a modern wind turbine the maximum power output is typically from a few and up to five megawatts.
- **High-speed shaft** drives the generator.
- **Hub and rotor blades** together make up the rotor of the wind turbine. The hub connects the rotor blades to the low-speed shaft. Pitching the blades is used to maximize the efficiency in low winds and reduce efficiency in high winds to protect the wind turbine from structural damage.
- **Low-speed shaft** connects the rotor to the gearbox.
- **Nacelle** is located atop the tower and contains the gearbox, low- and high-speed shafts, generator, and brakes.
- **Tower** carries the nacelle and the rotor. Since the wind speed increases with the height, a taller tower generally enables a wind turbine to generate more electric energy.
- **Wind vane** is used to measure the direction of the wind. The wind direction is used by the yaw mechanism to orient the wind turbine perpendicular to the wind.
- **Yaw mechanism** uses electrical motors to orient the wind turbine rotor perpendicular to the direction of the wind.

Having introduced the main components of a wind turbine, the next section describes how a variable-pitch, variable-speed wind turbine is typically controlled.

2.2 General Control Strategy

The purpose of this section is to explain how a variable-speed, variable-pitch wind turbine is controlled along a typical operating trajectory, and also to explain the variables that are involved in the control. This is relevant since the fault diagnosis and fault-tolerant control systems are subject to the conditions in the closed-loop system.

When controlling a wind turbine, the overall target is to minimize the operational cost while maximizing the generated power. This cost depends on the conditions under which the wind turbine produces the power, and implies that wind turbines operate along a certain trajectory, as the one shown in Figure 2.2. This trajectory is obtained using a strategy of control shown in Figure 2.3, which illustrates the control signals and rotor speed for obtaining the desired output power.

From examining the power curve it is obvious that the wind turbine only produces power in a limited range of wind speeds, which is further divided into two distinct regions, I and II; denoted the partial load region and the full load region. At wind speeds below the cut-in speed, $V_{w,cut-in}$, the wind turbine does not produce any energy since the operational cost exceeds the value of the produced power. Similarly, no energy is produced when wind speeds exceed the cut-out wind speed, $V_{w,cut-out}$, where the wind turbine is shut down to protect it from structural overloads. To explain what happens in the two regions of power production, these are explained separately in the following numbered list:

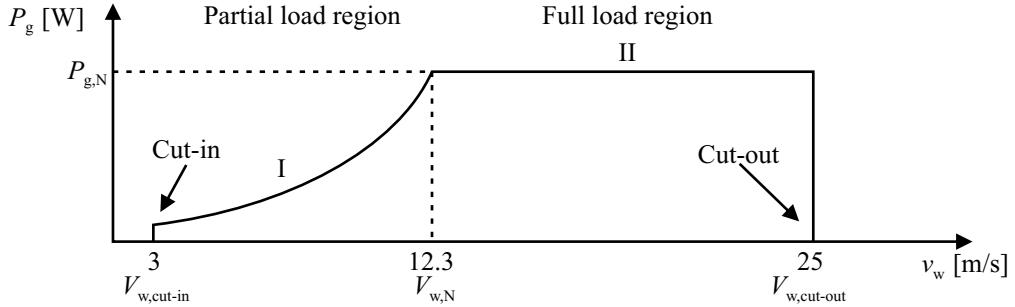


Figure 2.2: Ideal power curve for the wind turbine limiting the output power at high wind speeds to the rated value of 4.8 MW. The interval of the wind speed is divided into two regions, I and II, in which different control objectives exist.

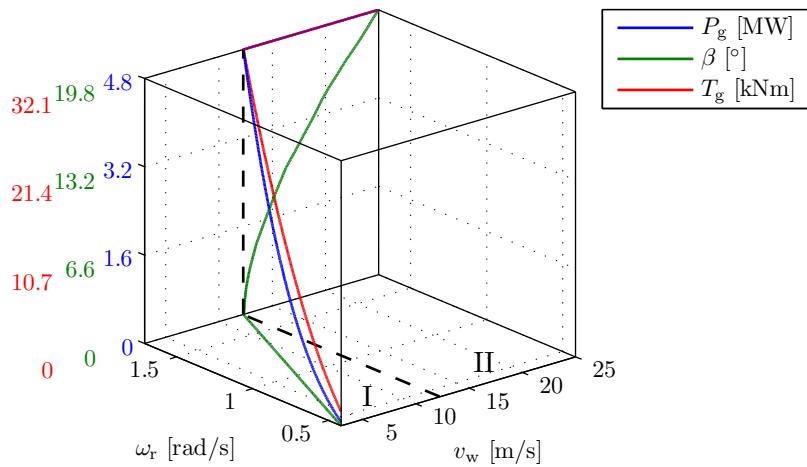


Figure 2.3: Typical control strategy for following the ideal power curve. The plots of the pitch angle (green) and the generator torque (red) show how the control signals are manipulated to obtain the desired output power (blue) as function of the wind speed v_w .

- I. The partial load region is located between the cut-in wind speed, $V_{w,cut-in}$, and the rated wind speed, $V_{w,N}$. In this region the wind turbine is controlled to generate as much power as possible. This is achieved by adjusting the generator torque to obtain an optimum ratio between the tip speed of the blades and the wind speed. Hereby, the efficiency of the aerodynamics is maximized.
- II. The full load region is located between the rated wind speed, $V_{w,N}$, and the cut-out wind speed, $V_{w,cut-out}$. Operation in the full load region involves that the produced power is kept at a rated value to minimize structural loads and thereby reduce fatigue damages. Additionally, the rotor speed is fixed to reduce acoustic noise emission from the wind turbine, which depends on the speed of the rotor.

Due to the existence of two distinct regions of control, the usual approach is to apply two different sets of controllers and interconnect these using a bumpless transfer mechanism. This is illustrated in Figure 2.4, which shows a block diagram of a reference controller that is based on classical methods and is introduced to establish a frame of reference. Notice that the structure of the reference controller is set up in compliance with kk-electronic a/s.

The strategy of the reference controller is to use two different controllers for the partial load region and the full load region. When the wind speed is below the rated value, the control system should maintain the pitch angle at its optimal value and control the generator torque in order to achieve the optimal tip-speed ratio. This behavior is achieved by setting the two switches in Figure 2.4 to be in Position I.

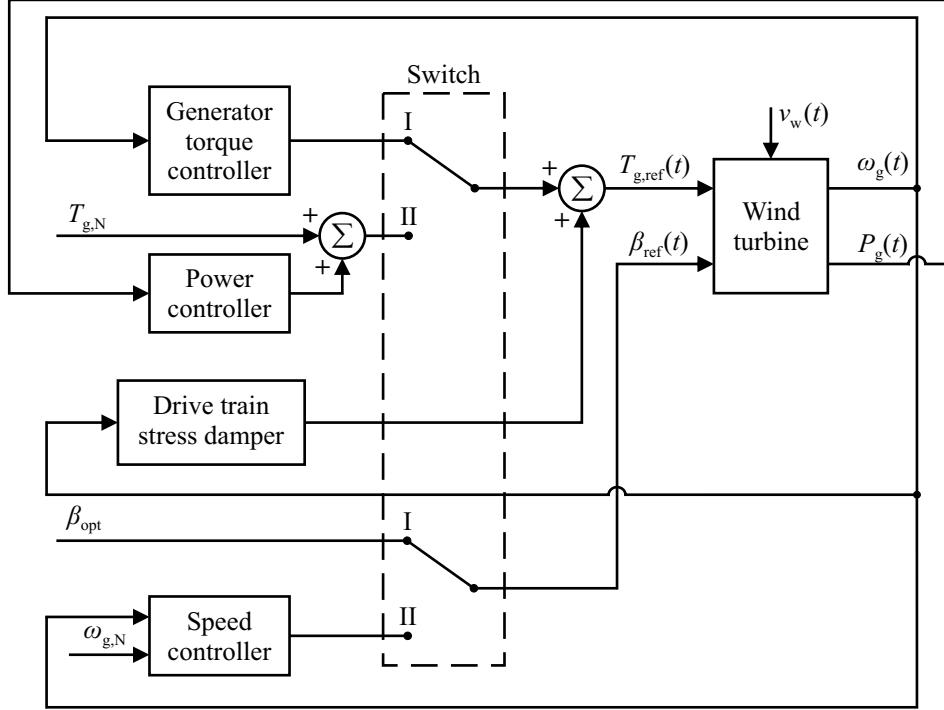


Figure 2.4: Structure of the reference controller. A switch reconfigures the control system to the current operating objectives.

Above the rated wind speed the output power is kept constant by pitching the rotor blades, while using a power controller that manipulates the generator torque around a constant value to remove steady-state errors on the output power. This behavior is obtained by setting the two switches in Figure 2.4 to be in Position II.

In both regions a drive train stress damper is utilized to dampen drive train oscillations actively. Together, the two sets of controllers are able to solve the control task of tracking the ideal power curve in Figure 2.2, by applying the control signals shown in Figure 2.3. In order to switch smoothly between the two sets of controllers a bumpless transfer mechanism is implemented.

The design of the reference controller is presented in Appendix C. The reference controller is considered as being the nominal control system throughout the thesis, and is applied to the wind turbine model throughout the analysis.

This chapter has provided the basic knowledge about wind turbines, including a description of the components. Furthermore, the ideal control of a wind turbine has been introduced. In the next chapter a wind model and a model of the considered wind turbine are set up.

Wind Turbine Modeling

The purpose of this chapter is to set up a mathematical model of the considered wind turbine. The model should be detailed enough to be used as simulation model. From the non-linear model a linear model is derived to be used in the linear design methods presented throughout the thesis. Notice that the chapter is inspired by [Esbensen *et al.*, 2008, pp. 23-38], and that the sections: Model Structure, Aerodynamic Model, and Drive Train Model are only slightly modified.

In the following section the structure of the wind turbine model is presented in a block diagram. Afterwards, each sub-model of the wind turbine is presented and combined to obtain a complete model of the wind turbine. To enable the further design, the model is linearized in Appendix B and rewritten on state space form.

3.1 Model Structure

The first stage of the modeling procedure is to divide the overall wind turbine system model into appropriate sub-models suitable of being modeled separately. These sub-models are shown in Figure 3.1.

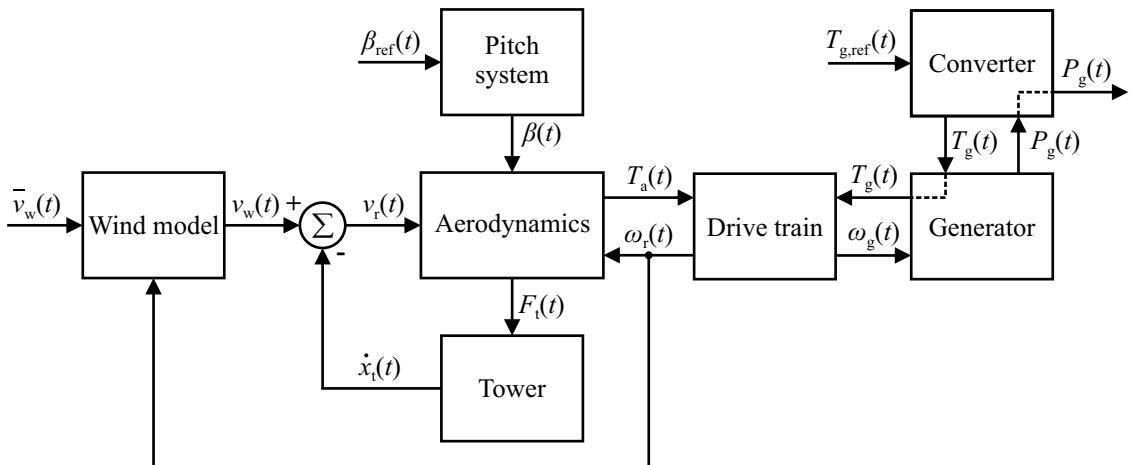


Figure 3.1: Relationship between sub-models of the wind turbine system model. The wind speed, $v_w(t)$, is the exogenous input and the generator power, $P_g(t)$, is the system output. The controllable inputs are the pitch angle reference, $\beta_{\text{ref}}(t)$, and the generator torque reference, $T_{g,\text{ref}}(t)$. The rotor speed is fed to the wind model to create wind shear and tower shadow effects.

The wind speed is the driving force of the system. Due to the swaying of the tower, the wind speed seen at the rotor is obtained by subtracting the speed of the nacelle from the speed of the wind.

The aerodynamic properties of the wind turbine are affected by the pitch angles of the blades, the speed of the rotor, and the wind speed. On this basis, an aerodynamic torque is transferred from the rotor to the generator through the drive train, and an aerodynamic thrust affects the rotor and thereby the tower.

The output of the wind turbine is electric power which comes from the converter. To operate the wind turbine according to the set of operating requirements, the pitch angles of the blades and the generator torque are adjusted. A pitch system controls the pitch angles of the blades, while a converter controls the generator torque.

In this section the wind turbine model has been divided into seven sub-models, in order to be individually modeled and combined afterwards. The purpose of the next section is to model the wind speed, being the exogenous input to the wind turbine.

3.2 Wind Model

The purpose of this section is to design a wind model, by determining and modeling the components that influence the wind speed applied to each blade. The model must generate the input to the wind turbine model; hence, an effective wind speed on each blade must be calculated. Additionally, the wind model should provide an output resembling the input to the anemometer, which is a point wind speed, contrary to the remaining considered wind speeds.

Generally, the wind speed is influenced by several components, which depend on the environment where the wind turbine is located. An equation giving an overview of the components is shown below.

$$v_w(t) = \bar{v}_w(t) + v_{ws}(t) + v_{ts}(t) + v_{tu}(t) \quad [\text{m/s}] \quad (3.1)$$

where:

- $v_{ts}(t)$ is the wind speed tower shadow component [m/s]
- $v_{tu}(t)$ is the wind speed turbulence component [m/s]
- $v_w(t)$ is the wind speed including tower shadow, turbulence, and wind shear [m/s]
- $\bar{v}_w(t)$ is the mean wind speed [m/s]
- $v_{ws}(t)$ is the wind shear component [m/s]

This general model is utilized in the designed wind model. The output from the wind model is wind speeds averaged over an area, instead of point wind speeds. Therefore, these are called effective wind speeds.

The next subsections describe the components, which form the basis for modeling the wind speed.

Wind Shear

The ground and other obstacles in the path of the wind cause frictional forces to act on the wind. The frictional forces imply that the mean wind speed becomes dependent on the height above ground level. This effect is called wind shear and can be calculated from the equation shown below [Dolan and Lehn, 2006]. The equation integrates the wind speed components from the beginning to the end of the blade profile and sums the components for the three blades obtaining the wind shear effect for the rotor plane. Figure 3.2 illustrates some of the parameters used in the equation.

$$\begin{aligned} v_{ws}(t) &= \frac{2v_{hub}(t)}{3qR^2} \sum_{i=1}^3 \int_{r_0}^R \left(\frac{r^2\alpha}{h} \cos(\psi_i(t)) + \frac{r^3\alpha(\alpha-1)}{2h^2} \cos^2(\psi_i(t)) \right. \\ &\quad \left. + \frac{r^4\alpha(\alpha-1)(\alpha-2)}{6h^3} \cos^3(\psi_i(t)) \right) dr \\ v_{ws}(t) &= \frac{2v_{hub}(t)}{3qR^2} \sum_{i=1}^3 \left(\frac{R^3\alpha}{3h} \cos(\psi_i(t)) + \frac{R^4\alpha(\alpha-1)}{8h^2} \cos^2(\psi_i(t)) \right. \\ &\quad \left. + \frac{R^5\alpha(\alpha-1)(\alpha-2)}{30h^3} \cos^3(\psi_i(t)) \right) \\ &\quad - \left(\frac{r_0^3\alpha}{3h} \cos(\psi_i(t)) + \frac{r_0^4\alpha(\alpha-1)}{8h^2} \cos^2(\psi_i(t)) + \frac{r_0^5\alpha(\alpha-1)(\alpha-2)}{30h^3} \cos^3(\psi_i(t)) \right) \quad [\text{m/s}] \end{aligned} \quad (3.2)$$

$$q \triangleq 1 - \left(\frac{r_0}{R} \right)^2$$

where:

- h is the height of the tower [m]
- R is the radius of the rotor [m]
- r_0 is the radius at which the blade profile begins [m]
- $v_{hub}(t)$ is the wind speed at hub height [m/s]
- α is the empirical wind shear exponent [·]
- $\psi_i(t)$ is the azimuth angle of Blade i [rad]

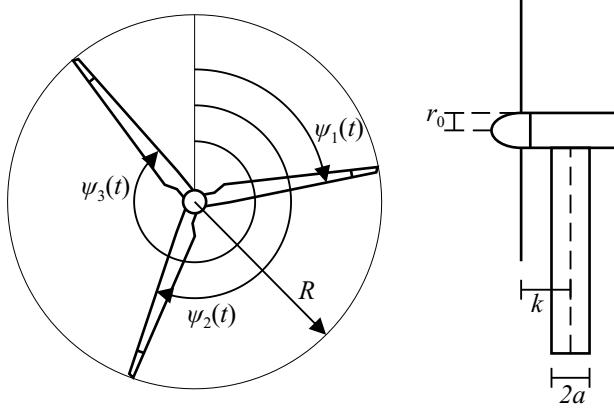


Figure 3.2: Sketch of a rotor and a wind turbine, showing parameters utilized in the wind model.

The wind shear component for one blade is illustrated in Figure 3.3, where it is clear that the wind speed is smallest at $\psi_i = \pi$ rad, since at this azimuth angle Blade i is closest to the ground.

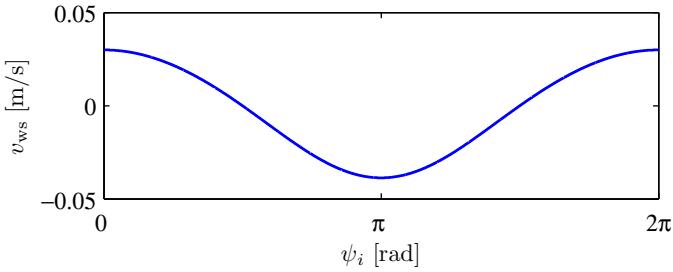


Figure 3.3: Wind shear component of Blade i as a function of the azimuth angle. The wind speed is smallest when the blade is closest to the ground.

Tower Shadow

When a blade is located in front of the tower, the lift on that blade decreases because the tower reduces the effective wind speed. This phenomenon is called tower shadow and implies that the force acting on each blade decreases every time a blade is located in front of the tower.

The magnitude of the tower shadow depends on the diameter of the tower and the distance between the blade and the tower. The tower shadow component can be calculated from the equation shown below [Dolan and Lehn, 2006]. As in the previous equation, the tower shadow component is calculated by integrating the wind speed components from the beginning to the end of the blade profile and summing the components for the three blades obtaining the tower shadow effect for the rotor plane. Note that the equation is only valid for $\psi_i \in [\frac{\pi}{2}, \frac{3\pi}{2}]$.

$$\begin{aligned}
 v_{ts}(t) &= \frac{2mv_{hub}(t)}{3qR^2} \sum_{i=1}^3 \int_{r_0}^R \left(\frac{ma^2 (r^3 \sin^2(\psi_i(t)) - rk^2)}{(r^2 \sin^2(\psi_i(t)) + k^2)^2} \right) dr \\
 v_{ts}(t) &= \frac{2mv_{hub}(t)}{3qR^2} \sum_{i=1}^3 \frac{a^2 \ln(R^2 \sin^2(\psi_i(t)) + k^2)}{2 \sin^2(\psi_i(t))} - \frac{a^2 \ln(r_0^2 \sin^2(\psi_i(t)) + k^2)}{2 \sin^2(\psi_i(t))} \\
 &\quad + \frac{a^2 k^2}{\sin^2(\psi_i(t)) (R^2 \sin^2(\psi_i(t)) + k^2)} - \frac{a^2 k^2}{\sin^2(\psi_i(t)) (r_0^2 \sin^2(\psi_i(t)) + k^2)} \quad [\text{m/s}] \quad (3.3)
 \end{aligned}$$

$$m \triangleq 1 + \frac{\alpha(\alpha - 1)R^2}{8h^2}$$

where:

a is the radius of the tower [m]

k is the distance from the tower midline to the blade [m]

To clarify the effect of the tower shadow component, it is plotted for one blade in Figure 3.4. Notice that the positive wind speeds are caused by the wind being forced around the tower.

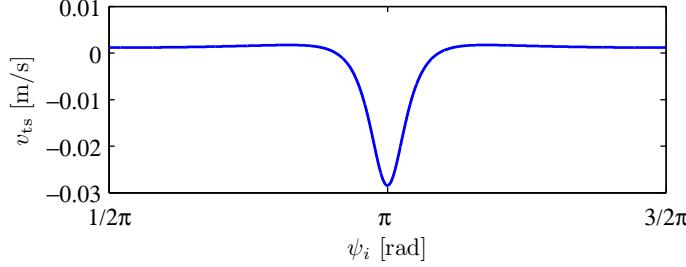


Figure 3.4: Tower shadow component of Blade i as a function of the azimuth angle.

Turbulence

The variations in the wind speed, which are not included in the mean wind speed, are called turbulence and are caused by multiple factors. The turbulence is often described by a turbulence intensity defined as:

$$I = \frac{\sigma_w}{\bar{v}_w} \cdot 100\% \quad [\%] \quad (3.4)$$

where:

I is the turbulence intensity [%]

σ_w is the standard deviation of the wind speed [m/s]

The expression reveals that the standard deviation of the wind speed is proportional to the mean wind speed.

The turbulence utilized in the implemented wind model originates from Wind Model SB-1 [Aalborg University and RISØ National Laboratory, 2005]. In this model the turbulence is based on the Kaimal spectrum, which describes the turbulence of a point wind. Since the wind model describes the wind speed averaged over the entire rotor plane, a low-pass filter is applied to smooth the wind speed signal.

Combined Wind Model

The structure of the wind model is sketched in Figure 3.5.

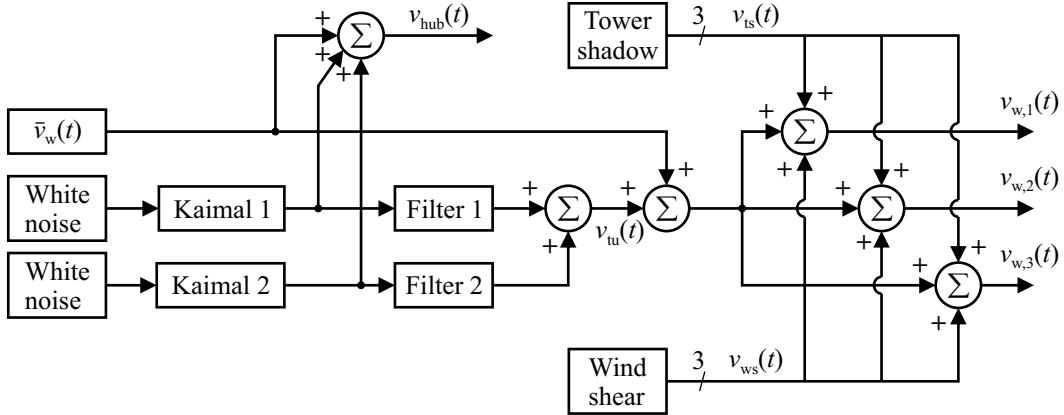


Figure 3.5: Structure of the wind model. The input to the model is a mean wind speed, from which the effective wind speeds on the three blades are calculated, by including tower shadow, wind shear, and turbulence.

The output of the model is three blade effective wind speeds, i.e. wind speeds averaged over the area of the blades. To model the wind speed at the anemometer, $v_{\text{hub}}(t)$, a point wind is fetched out of the wind model prior to the low-pass filtering. Notice that the anemometer is a low-pass filter itself.

To show how the wind model operates, a simulation at a constant rotor speed is performed and the result is shown in Figure 3.6.

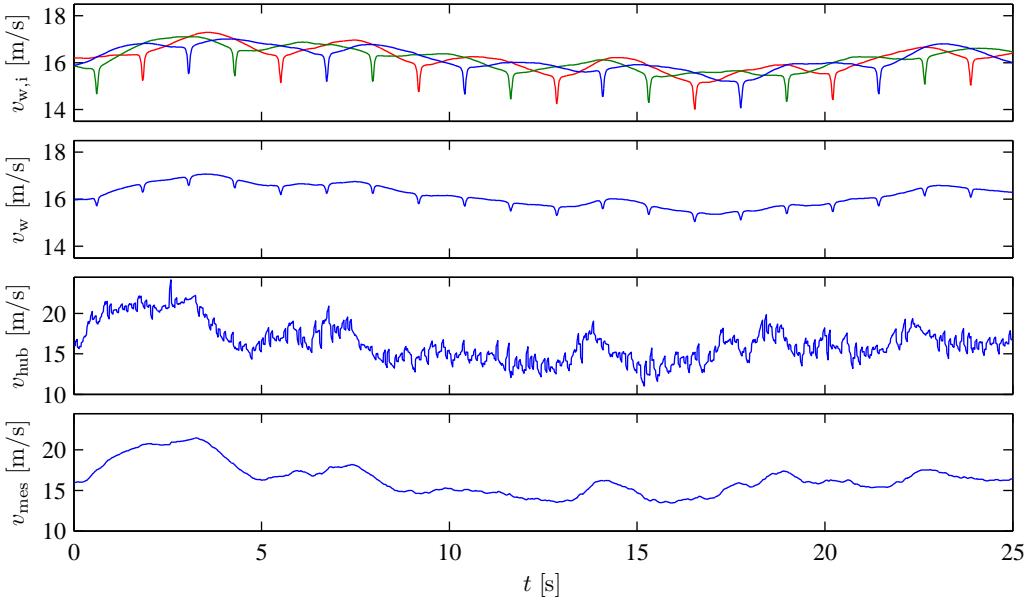


Figure 3.6: Output from the wind model for a mean wind speed of 16 m/s and at a constant rotor speed. The two lower subplots show a point wind speed at hub height, $v_{\text{hub}}(t)$, and the measured wind speed, $v_{\text{mes}}(t)$, which is a low-pass filtered version of $v_{\text{hub}}(t)$.

From the figure it is clear that even though both wind shear and tower shadow are significant on each blade, the wind shear is insignificant on the effective wind speed, $v_w(t)$. This is the effective wind speed experienced by the rotor when the tower does not move. Furthermore, there is an obvious difference between the effective wind speed, $v_w(t)$, and the wind speed measurement, $v_{\text{mes}}(t)$, since the latter is a point wind.

A wind model including wind shear, tower shadow, and turbulence has been set up in this section. The wind makes the blades rotate of reasons discussed in the next section.

3.3 Aerodynamic Model

In this section basic aerodynamic principles exploited by wind turbines are described, and a model describing the transfer from wind energy to rotational motion of the rotor is presented. The first part of the section assumes that the wind passing through the entire rotor plane has the same speed, but as explained in the previous section this is not the case. Hence, the aerodynamic model is modified at the end of this section to take into account the different blade effective wind speeds.

The three blades capturing the wind energy are illustrated in Figure 3.7(a), while the profile of the blades is shown in Figure 3.7(b). It is assumed that a yawing system exists, which always keeps the rotor plane perpendicular to the direction of the wind. When the blades rotate due to the rotor effective wind speed, $v_r(t)$, each blade experience a wind speed component, $v_b(t)$, opposite to the direction of rotation. Hence, the blade experiences the resulting wind speed, $v_{\text{res}}(t)$, as shown in Figure 3.7(b). The angle of the blade is called the pitch angle, $\beta(t)$, and is defined as the angle between the plane of rotation and the chord of the blade profile. Due to the profile of the blade, a pressure drop is generated on the upper surface of the blade, which results in a lift force, $F_l(t)$, perpendicular to the resulting wind speed, $v_{\text{res}}(t)$. In addition to the lift force, the resulting wind speed also generates a drag force, $F_d(t)$. The sum of $F_l(t)$ and $F_d(t)$ components in the direction

of the plane of rotation results in an aerodynamic torque acting on the rotor. In the same way, the components in the direction of the rotor axis results in an aerodynamic thrust, $F_t(t)$, acting on the rotor of the wind turbine.

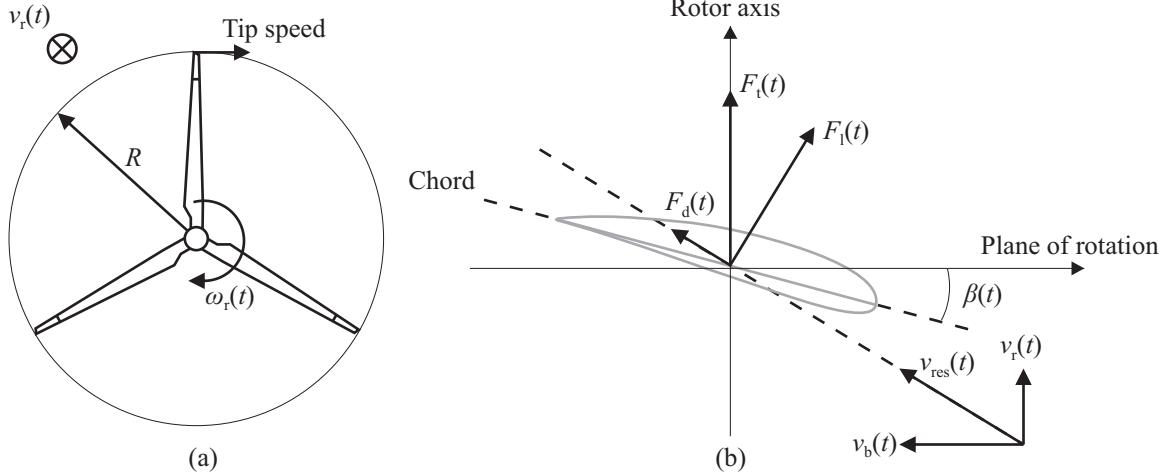


Figure 3.7: Rotor plane perpendicular to the wind field (a) and the aerodynamic forces acting on the blade profile (b).

The power available from the wind passing through the entire rotor swept area can be expressed as [Bianchi *et al.*, 2007, p. 19]:

$$P_w(t) = \frac{1}{2} \rho A v_r^3(t) \quad [\text{W}] \quad (3.5)$$

where:

A is the rotor swept area [m^2]

$P_w(t)$ is the power available from the wind [W]

$v_r(t)$ is the rotor effective wind speed [m/s]

ρ is the air density, which is assumed to be constant [kg/m^3]

From the available power in the wind, the power on the rotor is given based on the power coefficient, $C_p(\lambda(t), \beta(t))$, which depends on the tip-speed ratio and the pitch angle. The C_p -coefficient is not expressed as a mathematical function, but has to be looked up in a table. The C_p -surface is provided by kk-electronic a/s and is shown in the left subplot of Figure 3.8. Notice that the C_p -description implies that the aerodynamic model is static, which is a simplification. The power captured by the rotor is:

$$P_a(t) = P_w(t) C_p(\lambda(t), \beta(t)) \quad [\text{W}] \quad (3.6)$$

where:

$C_p(\lambda(t), \beta(t))$ is the power coefficient [·]

$P_a(t)$ is the power captured by the rotor [W]

$\beta(t)$ is the pitch angle [°]

$\lambda(t)$ is the tip-speed ratio [·]

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

$$\lambda(t) = \frac{\omega_r(t)R}{v_r(t)} \quad [\cdot] \quad (3.7)$$

where:

$\omega_r(t)$ is the rotor speed [rad/s]

The aerodynamic torque applied to the rotor is defined in Eq. (3.8) [Johnson *et al.*, 2006, p. 3]. By combining Eq. (3.5), Eq. (3.6), and Eq. (3.8) the torque applied on the rotor can be expressed as shown in Eq. (3.9).

$$T_a(t) = \frac{P_a(t)}{\omega_r(t)} \quad [\text{Nm}] \quad (3.8)$$

$$T_a(t) = \frac{1}{2\omega_r(t)} \rho A v_r^3(t) C_p(\lambda(t), \beta(t)) \quad [\text{Nm}] \quad (3.9)$$

where:

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

The wind acting on the rotor of the wind turbine also results in a thrust on the rotor. This thrust is calculated as shown below [Bianchi *et al.*, 2007, p. 19].

$$F_t(t) = \frac{1}{2} \rho A v_r^2(t) C_t(\lambda(t), \beta(t)) \quad [\text{N}] \quad (3.10)$$

where:

$C_t(\lambda(t), \beta(t))$ is the thrust coefficient [·]

$F_t(t)$ is the thrust exerted by the wind on the rotor [N]

The C_t -surface, based on a C_t -table provided by kk-electronic a/s, is shown in the right subplot of Figure 3.8.

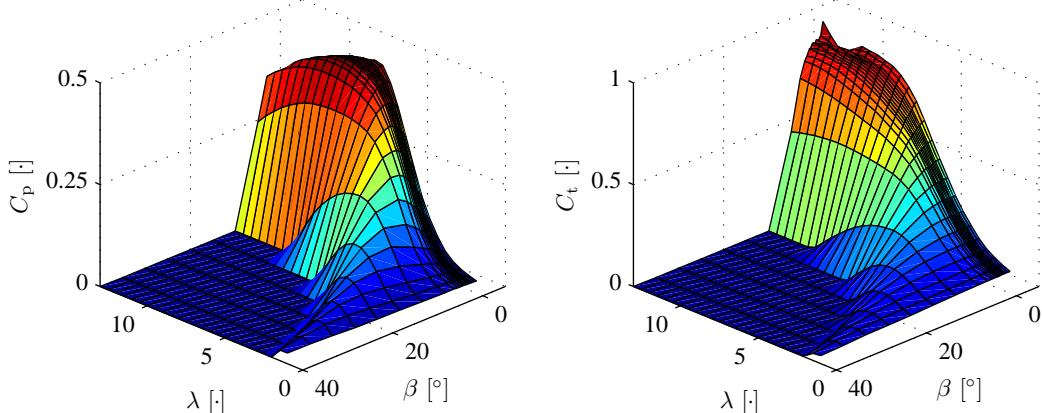


Figure 3.8: The C_p - and C_t -coefficients as function of the pitch angle and the tip-speed ratio. Notice that negative values have been set to zero.

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

$$T_a(t) = \frac{1}{3} \sum_{i=1}^3 T_{a,i}(t) \quad [\text{Nm}] \quad (3.11)$$

$$F_t(t) = \frac{1}{3} \sum_{i=1}^3 F_{t,i}(t) \quad [\text{N}] \quad (3.12)$$

where:

$F_{t,i}(t)$ is the thrust exerted by the wind on Blade i [N]

$T_{a,i}(t)$ is the aerodynamic torque applied to the rotor by Blade i [Nm]

The basic aerodynamic principles of a wind turbine have been described, and models for the aerodynamic torque and the aerodynamic thrust acting on the rotor have been set up. In the next section a model of the drive train is derived, which converts the aerodynamic torque into a torque applied to the generator.

3.4 Drive Train Model

In this section the drive train of the wind turbine is modeled. The aerodynamic torque is transferred to the generator through the drive train in order to upscale the rotational speed of the rotor, to a higher speed required by the generator. An equivalent diagram of the drive train is shown in Figure 3.9. The drive train model includes a low-speed shaft and a high-speed shaft, each composed of a moment of inertia and a frictional coefficient. The shafts are linked together by a gearbox modeled as a gear ratio without any loss. To describe the flexibility of the drive train, a torsion spring is included in the model.

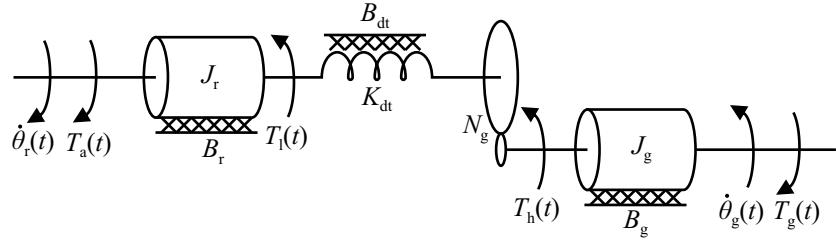


Figure 3.9: Drive train model divided into four components.

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

$$J_r \ddot{\theta}_r(t) = T_a(t) - T_l(t) - B_r \dot{\theta}_r(t) \quad [\text{Nm}] \quad (3.13)$$

where:

- B_r is the viscous friction of the low-speed shaft $[\text{Nm}/(\text{rad/s})]$
- J_r is the moment of inertia of the low-speed shaft $[\text{kgm}^2]$
- $T_l(t)$ is the torque acting on the low-speed shaft $[\text{Nm}]$
- $\dot{\theta}_r(t)$ is the angle of the low-speed shaft $[\text{rad}]$

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

$$J_g \ddot{\theta}_g(t) = T_h(t) - T_g(t) - B_g \dot{\theta}_g(t) \quad [\text{Nm}] \quad (3.14)$$

where:

- B_g is the viscous friction of the high-speed shaft $[\text{Nm}/(\text{rad/s})]$
- J_g is the moment of inertia of the high-speed shaft $[\text{kgm}^2]$
- $T_g(t)$ is the generator torque $[\text{Nm}]$
- $T_h(t)$ is the torque acting on the high-speed shaft $[\text{Nm}]$
- $\dot{\theta}_g(t)$ is the angle of the high-speed shaft $[\text{rad}]$

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

$$T_h(t) = \frac{T_l(t)}{N_g} \quad [\text{Nm}] \quad (3.15)$$

where:

- N_g is the drive train gear ratio $[.]$

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

$$T_l(t) = K_{dt} \theta_\Delta(t) + B_{dt} \dot{\theta}_\Delta(t) \quad [\text{Nm}] \quad (3.16)$$

$$\theta_\Delta(t) = \theta_r(t) - \frac{\theta_g(t)}{N_g} \quad [\text{rad}] \quad (3.17)$$

where:

B_{dt} is the torsion damping coefficient of the drive train [Nm/(rad/s)]

K_{dt} is the torsion stiffness of the drive train [Nm/rad]

$\theta_\Delta(t)$ is the torsion angle of the drive train [rad]

With the exception of the torsion angle, $\theta_\Delta(t)$, absolute angles of the shafts are not of interest for modeling the drive train dynamics. Therefore, the replacement $\omega(t) = \dot{\theta}(t)$ is utilized in the following rewriting, where a state space model of the drive train is pursued. The states of the model are $\omega_r(t)$, $\omega_g(t)$, and $\theta_\Delta(t)$. First, Eq. (3.17) is substituted into Eq. (3.16) to obtain:

$$T_l(t) = K_{dt}\theta_\Delta(t) + B_{dt} \left(\omega_r(t) - \frac{\omega_g(t)}{N_g} \right) \quad [\text{Nm}] \quad (3.18)$$

where:

$\omega_g(t)$ is the generator speed [rad/s]

Substituting Eq. (3.18) into Eq. (3.13) results in Eq. (3.19). A similar approach is used to derive Eq. (3.20); however, in this case Eq. (3.18) first has to be substituted into Eq. (3.15) before inserting it in Eq. (3.14). Lastly, Eq. (3.17) is differentiated to obtain Eq. (3.21).

$$J_r\dot{\omega}_r(t) = T_a(t) - K_{dt}\theta_\Delta(t) - (B_{dt} + B_r)\omega_r(t) + \frac{B_{dt}}{N_g}\omega_g(t) \quad [\text{Nm}] \quad (3.19)$$

$$J_g\dot{\omega}_g(t) = \frac{K_{dt}}{N_g}\theta_\Delta(t) + \frac{B_{dt}}{N_g}\omega_r(t) - \left(\frac{B_{dt}}{N_g^2} + B_g \right) \omega_g(t) - T_g(t) \quad [\text{Nm}] \quad (3.20)$$

$$\dot{\theta}_\Delta(t) = \omega_r(t) - \frac{1}{N_g}\omega_g(t) \quad [\text{rad/s}] \quad (3.21)$$

Three first order differential equations have been derived in this section in order to describe the behavior of the drive train. In the next section, the effect on the tower from the aerodynamic thrust is considered.

3.5 Tower Model

The rotor effective wind speed causes a thrust to act on the rotor, which makes the tower sway back and forth. In this project it is assumed that the thrust on the rotor acts in the direction of the wind speed. This is not completely the situation due to forces from the blades and a counter torque from the drive train and generator, which make the tower sway sideways as well.

In the modeling of the tower it is assumed that the blades and the tower are stiff, and that the tower rotates around a point in the bottom, when affected by any forces, as shown in Figure 3.10. This assumption is introduced to be able to transfer the three tower forces into a resulting force, which creates an acceleration of the tower. Furthermore, the effect of the blades being located at different heights can be taken into account; thus, introducing a force which is dependent on the azimuth angle of the rotor defined in Figure 3.2 on Page 11.

The movement of the tower is modeled using a torsion spring and a friction component connected to an inertia. This is illustrated on the righthand side of Figure 3.10 and in Eq. (3.22).

$$J_t\ddot{\theta}_t(t) = T_t(t) - B_t\dot{\theta}_t(t) - K_t\theta_t(t) \quad [\text{Nm}] \quad (3.22)$$

where:

B_t is the friction of the rotating tower system [Nm/(rad/s)]

J_t is the inertia of the tower [kgm²]

K_t is the stiffness of the tower torsion spring [Nm/rad]

$T_t(t)$ is the torque affecting the tower [Nm]

$\theta_t(t)$ is the rotation angle of the tower [rad]

To calculate the torque, $T_t(t)$, from the three tower forces it is necessary to calculate their height above ground and the points on the blades where the forces act.

It is assumed that the thrust $F_{t,i}(t)$ exerted on Blade i attacks where the thrust components towards the hub and towards the blade tip are equal. Since the thrust is dependent on R^2 the

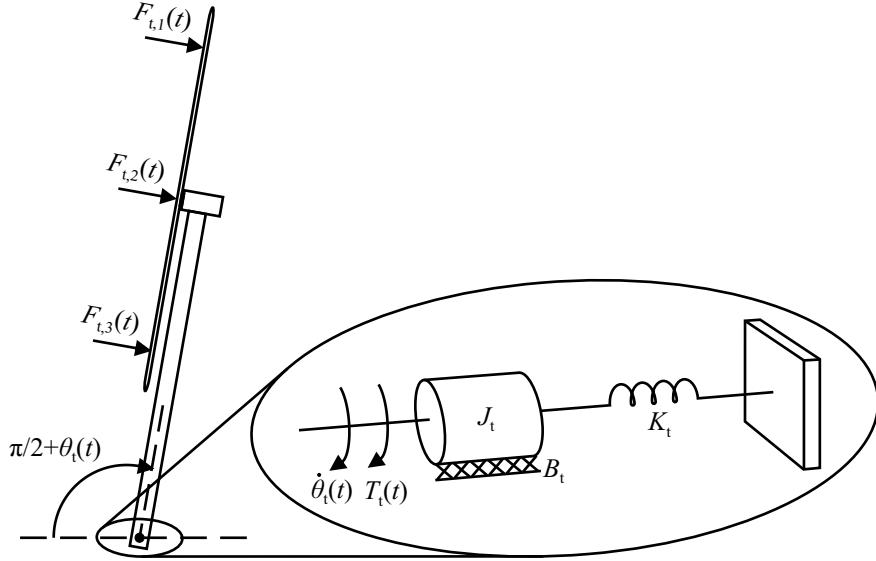


Figure 3.10: Shows the movement of the flexible tower modeled using a rotary system.

following equation can be set up to calculate the point where the thrust attacks:

$$\int_0^{r_t} r^2 dr = \int_{r_t}^R r^2 dr$$

$$r_t = 2^{-1/3} R \quad [\text{m}] \quad (3.23)$$

where:

r_t is the distance from the hub to where the thrust acts on the blade [m]

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

$$F_{th}(t) = \underbrace{F_{t,1}(t) \left(1 + \frac{r_t}{h} \cos(\psi_1(t))\right)}_{F_{th,1}(t)} + \underbrace{F_{t,2}(t) \left(1 + \frac{r_t}{h} \cos(\psi_2(t))\right)}_{F_{th,2}(t)} + \underbrace{F_{t,3}(t) \left(1 + \frac{r_t}{h} \cos(\psi_3(t))\right)}_{F_{th,3}(t)} \quad [\text{N}] \quad (3.24)$$

where:

$F_{t,i}(t)$ is the thrust acting on Blade i [N]

$F_{th}(t)$ is the force acting on the tower at hub height [N]

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

As explained in Section 3.10 the acceleration of the nacelle is the only measured variable in the tower model. Therefore, the movement of the tower is now described by a linear displacement of the nacelle. An illustration of this model is given in Figure 3.11.

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

$$M_t \ddot{x}_t(t) = F_{th}(t) - B_t \dot{x}_t(t) - K_t x_t(t) \quad [\text{N}] \quad (3.25)$$

where:

B_t is the tower damping coefficient [$\text{N}/(\text{m}/\text{s})$]

K_t is the tower torsion coefficient [N/m]

M_t is the top mass of the tower [kg]

$x_t(t)$ is the displacement of the nacelle from its equilibrium position [m]

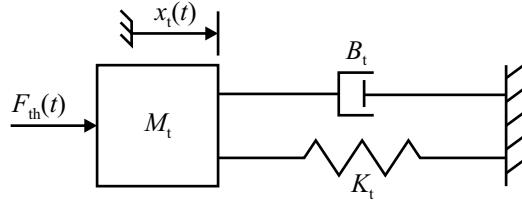


Figure 3.11: The movement of the flexible tower is modeled using a spring-damper system.

The swaying of the tower affects the effective wind speed seen on the rotor, i.e. when the tower moves towards the wind the effective wind speed increases. This implies that the rotor effective wind speed can be described as:

$$v_r(t) = v_w(t) - \dot{x}_t(t) \quad [\text{m/s}] \quad (3.26)$$

From this section it has been established how the rotor effective wind speed applies an aerodynamic thrust on the rotor, which make the tower sway. The next section describes the generator and converter, which enable generation of electric power and make it possible to adjust the load applied to the drive train by the generator.

3.6 Power System Model

In this section the power system is modeled, which includes presenting models of the generator and converter. Electric power is generated by the generator, and to enable variable-speed operation, currents in the generator are controlled using power electronics. Therefore, power electronic converters interface the wind turbine generator output with the utility grid.

It is assumed that the converter consists of four similar units sketched in Figure 3.12, each having an internal controller. These units together load the generator with a certain torque, which depends on the currents drawn from the generator. Since torque and electric power are the only variables of interest in the simplified model of the energy conversion system, currents and voltages are not considered at all.

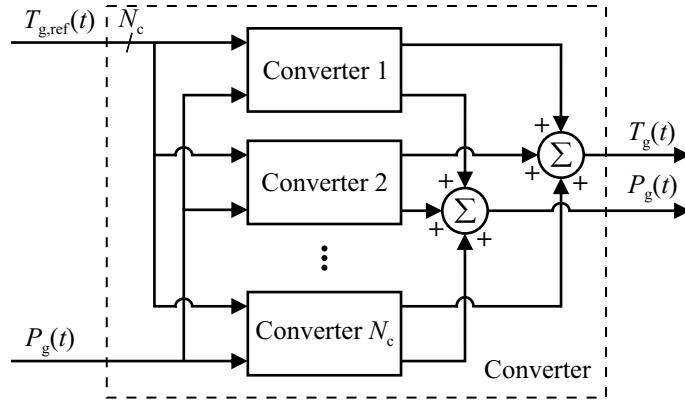


Figure 3.12: The converter consists of N_c units capable of loading the generator by a certain torque, specified by a torque reference.

According to kk-electronic a/s the dynamics of each converter can be approximated by a first order system with a time delay. Since the converter consists of several converters having equal characteristics, this section describes only one of these. Hereby, a single converter is modeled as:

$$\frac{T_g(s)}{T_{g,\text{ref}}(s)} = \frac{\exp(-t_{g,d}s)}{\tau_g s + 1}$$

$$\dot{T}_g(t) = -\frac{1}{\tau_g} T_g(t) + \frac{1}{\tau_g} T_{g,\text{ref}}(t - t_{g,d}) \quad [\text{Nm/s}] \quad (3.27)$$

where:

- $T_{g,\text{ref}}(t)$ is the reference for the generator torque [Nm]
- $t_{g,d}$ is the communication delay to the converter [s]
- τ_g is the time constant of the first order system [s]

In addition to the delayed first order system, the output from the converter is saturated and has a slew rate, $\max|\dot{T}_g(t)|$. A block diagram of a single converter model is shown in Figure 3.13.

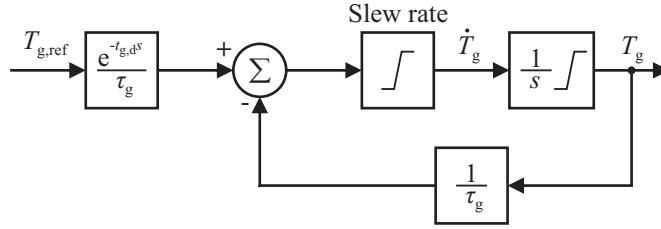


Figure 3.13: Block diagram of a converter including slew rate, delay, and limited operational range.

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

$$P_g(t) = \eta_g \omega_g(t) T_g(t) \quad [\text{W}] \quad (3.28)$$

where:

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

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

3.7 Pitch System Model

The purpose of this section is to explain how the pitch system is modeled. The pitch system consists of three identical pitch actuators, each having an internal controller, which is considered to be a part of each pitch actuator. Hence, this section describes only one pitch actuator.

The pitch actuator adjusts the pitch of a blade by rotating it. The pitch system is a hydraulic system as roughly sketched in Figure 3.14.

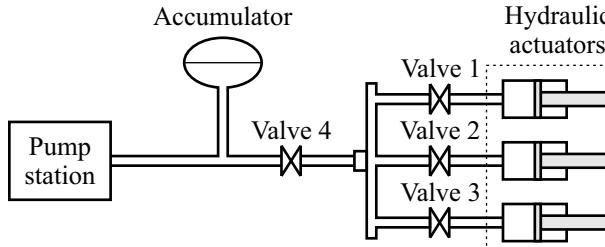


Figure 3.14: Hydraulic pitch system with three pitch actuators, which can be controlled individually.

The pitch actuator is modeled as a second order system with a time delay, described as:

$$\frac{\beta(s)}{\beta_{\text{ref}}(s)} = \frac{\exp(-t_d s) \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\ddot{\beta}(t) = -2\zeta\omega_n \dot{\beta}(t) - \omega_n^2 \beta(t) + \omega_n^2 \beta_{\text{ref}}(t - t_d) \quad [\text{°}/\text{s}^2] \quad (3.29)$$

where:

- t_d is the communication delay to the pitch actuator [s]
- $\beta(t)$ is the pitch angle [°]
- $\beta_{\text{ref}}(t)$ is the reference to the pitch angle [°]
- ω_n is the natural frequency of the pitch actuator model [rad/s]
- ζ is the damping ratio of the pitch actuator model [·]

The equation explains the operation of the pitch actuator when it operates within its limitations. However, the pitch actuator has both a limited slew rate and limited operational range, as illustrated in Figure 3.15.

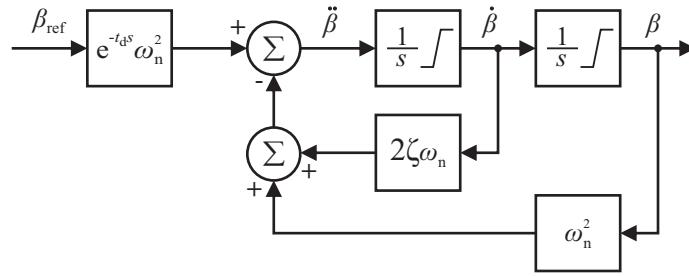


Figure 3.15: Block diagram of a pitch actuator including slew rate, delay, and limited operational range.

In this section the actuator for pitching each blade has been modeled as a delayed second order system. This finalizes the modeling of the wind turbine. In the next section, simplifications made in the modeling are discussed.

3.8 Neglected Dynamics of the Model

The purpose of this section is to outline simplifications, which are introduced in the modeling of the wind turbine. The simplifications are made since only a reduced set of model parameters are available, and to limit the extent of the modeling effort.

Generally, two simplifications are introduced: model only the dominant modes of the system while neglecting all others and assume that the parameters of the wind turbine are constant, although they depend on the operating conditions.

In addition to the general assumptions the following significant simplifications are introduced:

- **Perfect yaw alignment:** The wind is assumed to be perpendicular to the rotor plane at all times. This simplification eliminates some periodic fluctuations caused by yaw misalignment of the wind turbine.
- **Static aerodynamic model:** The aerodynamics is assumed to possess static properties; thus, neglecting the dynamical properties caused by changes in the wind, rotor speed, and pitch angles. This assumption is made since only a C_p -table is available for the considered wind turbine.
- **Stiff blades:** The blades are assumed to be stiff, but are in fact flexible, especially on large wind turbines similar to the considered one. This simplification eliminates all bending modes of the blades and transfers all forces acting on the blades directly to the tower.

The presented simplifications should be taken into account when assessing the results of this thesis, since they may affect the performance of the designed algorithms when applied to a real wind turbine.

To obtain the non-linear simulation model, all sub-models derived in the preceding sections are combined directly. However, in some parts of the design procedure it is favorable to utilize a linearized model. Therefore, a combined model is arranged on state space form in the next section, implying linearization of the non-linear parts of the model.

3.9 Assembled Model

The non-linear simulation model of the wind turbine is assembled directly from the sub-models derived in the previous sections. The purpose of this section is to present the corresponding linear model on state space form, which is desired for the utilized design methods. Since not all parts of the sub-models are linear, the non-linear functions are linearized in Appendix B.

The state space model of the wind turbine is shown in Eq. (3.30). It should be noticed that the included variables are small signal values, as this is not indicated explicitly in the notation.

The aerodynamic model is affected by the rotor effective wind speed, $v_r(t)$, which originates from swaying of the tower, $\dot{x}_t(t)$, affecting the rotor effective wind speed without tower movement, $v_w(t)$. The system has the following inputs: the generator torque references and the pitch angle references.

Since there are four generator torque references and three pitch angle references, the elements in the corresponding matrices should be considered as being matrices with dimensions indicated in the indices.

$$\begin{aligned}
 \dot{x}(t) = & Ax(t) + B_1 w(t) + B_2 u(t) \\
 \begin{bmatrix} \dot{T}_g(t) \\ \dot{x}_t(t) \\ \ddot{x}_t(t) \\ \dot{\beta}(t) \\ \ddot{\beta}(t) \\ \dot{\theta}_\Delta(t) \\ \dot{\omega}_g(t) \\ \dot{\omega}_r(t) \end{bmatrix} = & \begin{bmatrix} A_{11} & 0_{4 \times 1} & 0_{4 \times 1} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 1} & 0_{4 \times 1} & 0_{4 \times 1} \\ 0_{1 \times 4} & 0 & 1 & 0_{1 \times 3} & 0_{1 \times 3} & 0 & 0 & 0 \\ 0_{1 \times 4} & -\frac{K_t}{M_t} & a_{33} & a_{34} & 0_{1 \times 3} & 0 & 0 & a_{38} \\ 0_{3 \times 4} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0_{3 \times 4} & 0_{3 \times 1} & 0_{3 \times 1} & A_{54} & A_{55} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0_{1 \times 4} & 0 & 0 & 0_{1 \times 3} & 0_{1 \times 3} & 0 & -\frac{1}{N_g} & 1 \\ a_{71} & 0 & 0 & 0_{1 \times 3} & 0_{1 \times 3} & \frac{K_{dt}}{J_g N_g} & a_{77} & \frac{B_{dt}}{N_g J_g} \\ 0_{1 \times 4} & 0 & a_{83} & a_{84} & 0_{1 \times 3} & -\frac{K_{dt}}{J_r} & \frac{B_{dt}}{N_g J_r} & a_{88} \end{bmatrix} \begin{bmatrix} T_g(t) \\ x_t(t) \\ \dot{x}_t(t) \\ \beta(t) \\ \dot{\beta}(t) \\ \theta_\Delta(t) \\ \omega_g(t) \\ \omega_r(t) \end{bmatrix} \\
 + & \begin{bmatrix} 0_{4 \times 3} \\ 0_{1 \times 3} \\ e_{31} \\ 0_{3 \times 3} \\ 0_{3 \times 3} \\ 0_{1 \times 3} \\ 0_{1 \times 3} \\ e_{81} \end{bmatrix} v_w(t) + \begin{bmatrix} B_{11} & 0_{4 \times 3} \\ 0_{1 \times 4} & 0_{1 \times 3} \\ 0_{1 \times 4} & 0_{1 \times 3} \\ 0_{3 \times 4} & 0_{3 \times 3} \\ 0_{3 \times 4} & B_{42} \\ 0_{1 \times 4} & 0_{1 \times 3} \\ 0_{1 \times 4} & 0_{1 \times 3} \\ 0_{1 \times 4} & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} T_{g,\text{ref}}(t) \\ \beta_{\text{ref}}(t) \end{bmatrix} \quad (3.30)
 \end{aligned}$$

$$\begin{aligned}
 A_{11} = & -\frac{1}{\tau_g} I_{4 \times 4} & A_{54} = & -\omega_n^2 I_{3 \times 3} \\
 A_{55} = & -2\zeta\omega_n I_{3 \times 3} & & \\
 a_{33} = & -\frac{B_t}{M_t} - \frac{1}{M_t} \frac{\partial F_t(t)}{\partial v_r} \Bigg| \begin{array}{l} \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} & a_{34} = & \frac{1}{3M_t} \frac{\partial F_t(t)}{\partial \beta} \Bigg| \begin{array}{l} \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} & 1_{1 \times 3} \\
 a_{38} = & \frac{1}{M_t} \frac{\partial F_t(t)}{\partial \omega_r} \Bigg| \begin{array}{l} \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} & a_{71} = & -\frac{1}{J_g} 1_{1 \times 4} \\
 a_{77} = & -\left(\frac{\eta_{dt} B_{dt}}{J_g N_g^2} + \frac{B_g}{J_g}\right) & a_{83} = & -\frac{1}{J_r} \frac{\partial T_a(t)}{\partial v_r} \Bigg| \begin{array}{l} \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} \\
 a_{84} = & \frac{1}{3J_r} \frac{\partial T_a(t)}{\partial \beta} \Bigg| \begin{array}{l} \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} & 1_{1 \times 3} & a_{88} = & -\frac{B_{dt} + B_r}{J_r} + \frac{1}{J_r} \frac{\partial T_a(t)}{\partial \omega_r} \Bigg| \begin{array}{l} \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array}
 \end{aligned}$$

$$\left| \begin{array}{l} B_{11} = \frac{1}{\tau_g} I_{4 \times 4} \\ e_{31} = \frac{1}{3M_t} \frac{\partial F_t(t)}{\partial v_r} \\ \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} \right|^{1_{1 \times 3}} \quad \left| \begin{array}{l} B_{42} = \omega_n^2 I_{3 \times 3} \\ e_{81} = \frac{1}{3J_r} \frac{\partial T_a(t)}{\partial v_r} \\ \beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r \end{array} \right|^{1_{1 \times 3}}$$

where:

- A is the system matrix
- B_1 is the disturbance matrix
- B_2 is the input matrix
- $u(t)$ is the input vector
- $w(t)$ is the disturbance vector
- $x(t)$ is the state vector

In this section a linearized model of the combined wind turbine system has been presented, which is based on the linearization made in Appendix B. In order to end the modeling chapter, the next section describes the available measurements on the considered wind turbine and specifies the sampling rate and the parameters of the assumed zero-mean Gaussian distributed measurement noise.

3.10 Sensor Parameters

The purpose of this section is to list the measured variables of the system model and characterize the measurements by the amount of noise on each of them.

Generally, it is determined not to model the dynamics of the sensors, since they are significantly faster than the dynamics of the wind turbine. The only exception is the anemometer measuring the wind speed, which is modeled as a first-order low-pass filter with a time constant of half a second according to [Hristov *et al.*, 2000].

All measurements are sampled at a rate of 100 Hz. The measurements are emulated by adding zero-mean Gaussian distributed noise to the deterministic values according to the standard deviation shown in Table 3.1. Notice that although both continuous and discrete design methods are considered in this thesis, all simulations are performed using discretized or discrete implementations of the control system including measurement noise for a realistic behavior.

Measured Variable	Sensor Type	Symbol	Standard Deviation
Electric power	Soft sensor (estimate)	$P_{g,j}$	-
Generator speed	Speed encoder	ω_g	0.0158 rad/s
Generator torque	Soft sensor (estimate)	$T_{g,j}$	45 Nm
Pitch angle	Encoder	β_i	0.2°
Rotor speed	Speed encoder	ω_r	0.025 rad/s
Tower acceleration	Accelerometer	a_t	0.01 m/s ²
Wind speed (point wind)	Anemometer	v_{hub}	0.5 m/s

Table 3.1: *Sensor types and standard deviations of the measured variables of the wind turbine. Notice that $i = 1, 2, 3$ and $j = 1, 2, 3, 4$ correspond to the number of blades and converters.*

The different measurements have different characteristics, since some variables are easier to measure than others. The rotor speed, for instance, is hard to measure, because large forces act on the low-speed shaft causing it to make transversal movements inside the nacelle.

Regarding the wind speed, it is very favorable to have a good measurement of the wind speed affecting the rotor, since it is the driving force of the system. However, the measurement of the wind speed is performed by an anemometer located on the nacelle. This provides only a measurement of the wind speed in a single point of the rotor plane and is disturbed by the turbulence from the rotor. Due to the characteristics of the wind speed measurement, it is only used for detecting cut-in and cut-out incidents. The measured wind speed at hub height appears in Figure 3.6 on Page 13 where it can be compared to the effective wind speed.

The pitch angles of the three blades are measured on the cylinders of the pitch actuators. The components of the generator torque are provided by independent estimates, based on a soft sensor

in each converter. The torque is estimated based on current and flux, whereas the electric power output is determined from current and voltage. Therefore, it is assumed that the measurements of torque and electric power are correlated, and implies that a bias on either the measured torque or electric power induces a bias on the other measurement too; this property is used when injecting faults into the system model.

No standard deviation is stated for the measured electric power in Table 3.1. The reason for this is that the value is not fixed, since the power is supposed to be evaluated as the product of current and voltage, where the voltage is assumed constant at 33 kV. Both the voltage and current are assumed to have an additive measurement noise with amplitude corresponding to one percent of the signal amplitude at half the nominal values.

In this chapter a mathematical model of the wind turbine has been set up and a linearized model has been derived to be used in the linear design methods presented throughout the thesis. In the next chapter a fault analysis is presented to select the faults to be handled in this thesis.

Fault Analysis

The purpose of this chapter is to identify possible faults on the wind turbine and determine their effect on the system behavior. A number of the analyzed faults are then selected for further work in this project, and redundancies in the system are identified to determine the detectable faults. Finally, the remedial actions that must be conducted to stop the propagation of the fault are established.

The method utilized to structure the fault analysis is inspired by [Izadi-Zamanabadi, 1999, p. 36], and the steps in the analysis and design procedures are outlined in Figure 4.1.

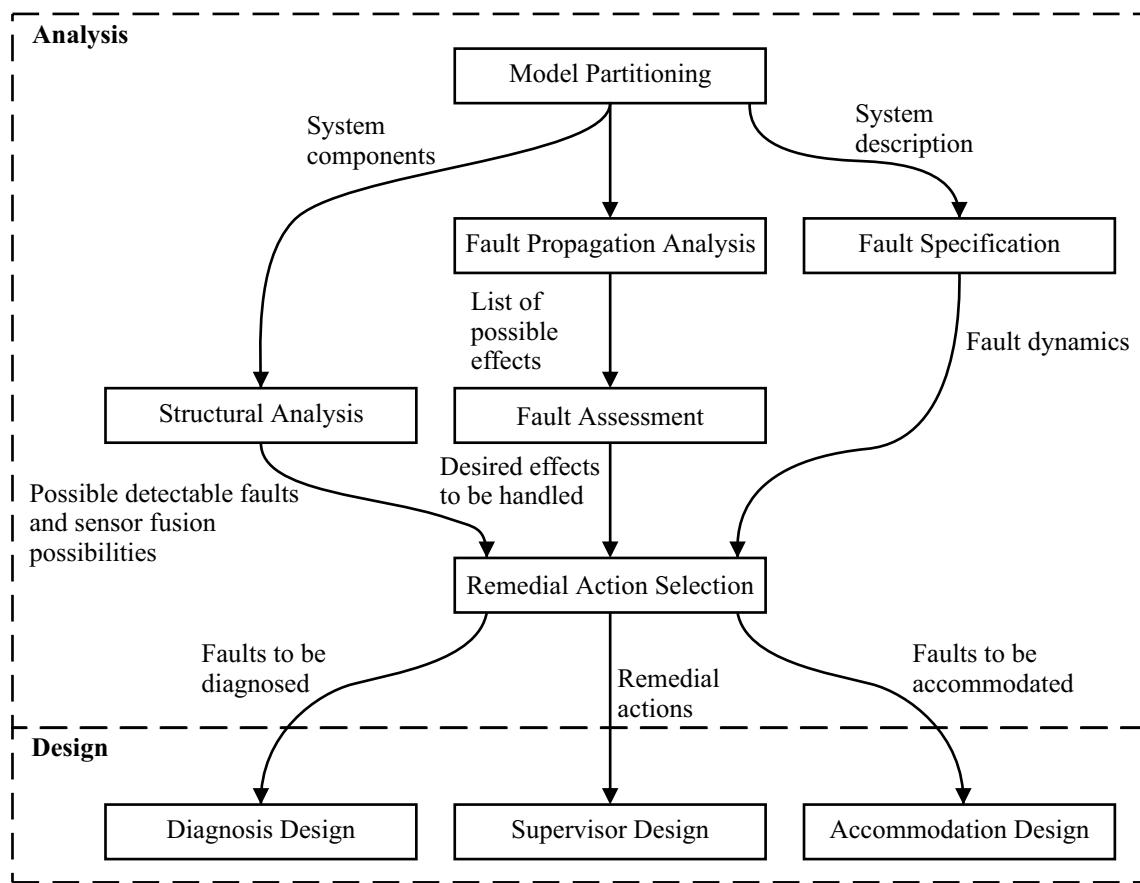


Figure 4.1: Overview of the method utilized in the fault analysis. Notice that Diagnosis Design and Supervisor Design only exists when designing an active fault-tolerant control system.

The figure shows the steps in the procedure and their interconnections, but to give an insight into the purpose of every step in the analysis, they are explained in the following bullet list:

- **Model Partitioning:** The wind turbine model is divided into sub-models suitable for analysis and identification of the possible component faults in each subsystem.
 - **Fault Propagation Analysis:** The fault propagation analysis propagates the component faults through the system and determines their end-effects at system level.
 - **Fault Assessment:** The fault assessment assesses the faults identified in the fault propagation analysis, by determining their occurrence and their impact on the performance of the wind turbine control system. The fault assessment furthermore determines the end-effects

which must be handled in this project. Finally, the component faults are traced back from the end-effects to the associated component faults.

- **Structural Analysis:** The structural analysis determines analytical redundancy relations in the system and determines the detectable faults.
- **Fault Specification:** The fault specification specifies the dynamics of the faults identified in the fault assessment.
- **Remedial Action Selection:** The remedial action selection determines the action that must be taken to stop the propagation of the faults. This could be switching between redundant sensors or reconfiguring controllers. Additionally, the requirements to the fault diagnosis and fault-tolerant control system are determined.

An outline of the method utilized in this chapter has been provided, and the next section handles the first step in the fault analysis.

4.1 Model Partitioning

The purpose of this section is to divide the system model described in Chapter 3 into appropriate sub-models, which can be analyzed separately. This division makes it possible to identify all possible component faults in each sub-model.

The wind turbine model can be divided into sub-models based on separate functionalities: rotor, drive train, and power system, which are described in Chapter 3. Figure 4.2 shows the components included in each of the sub-models and their interconnections. Notice that internal controllers are located in each pitch actuator and converter unit, which has a significant impact on the effects of the component faults. One direct consequence is that an offset in the output value of an actuator is compensated for by the internal controller, so that no fault is seen at the output.

The analysis is performed in closed loop, where the wind turbine is controlled according to two different strategies as explained in Section 2.2. This is elaborated in the reference controller design described in Appendix C. To distinguish between the different control strategies, the configurations of the controller in the two regions are marked with different colors.

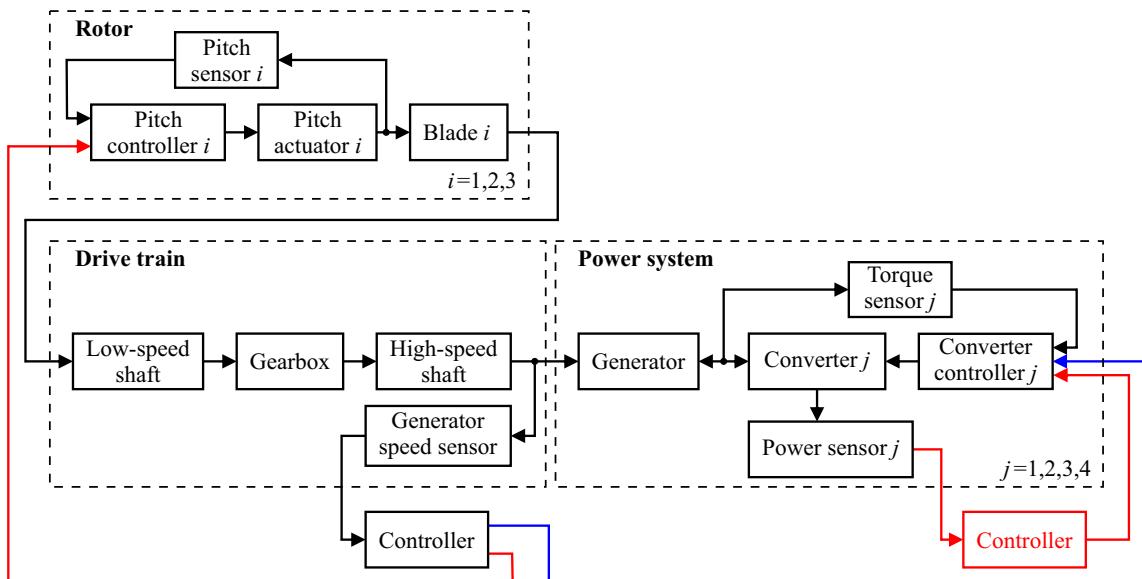


Figure 4.2: Overview of the three sub-models of the wind turbine. To establish an overview of the closed-loop system, the configurations of the controller in partial load operation (blue) and full load operation (red) are included in the figure. The symbols i and j indicate that there are multiple identical blocks.

In this project only a subset of the possible faults in the wind turbine is considered, excluding mechanical faults that cause destructible damage to the wind turbine. This is chosen, since these faults cannot be accommodated and is therefore not relevant in terms of fault-tolerant control. The considered component faults are shown in Figure 4.3-4.5 in the next section.

The wind turbine has been divided into three sub-models, suitable for being analyzed separately. In the next section the possible component faults are identified and propagated through the system, to determine their end-effects on system level.

4.2 Fault Propagation Analysis

The purpose of this section is to describe how component faults propagate through the system, by describing the effects on surrounding components of the system. This is accomplished by making a Failure Mode and Effect Analysis (FMEA) for the three main components; rotor, drive train, and power system, identified in the previous section. A FMEA is a commonly accepted technique for making a fault propagation analysis [Blanke *et al.*, 2006, p. 85].

In the FMEA it is assumed that no fault handling exists in the wind turbine control system even though this is not true. However, since each wind turbine manufacture has its own fault handling system it is difficult to assume something general. Furthermore, it is assumed that the nominal control system is fault-free.

Rotor

The purpose of this subsection is to set up a FMEA for the rotor sub-model and to describe the faults and their propagation to the effect on the rotor.

The FMEA scheme for the rotor is shown in Figure 4.3. Even though there exist three pitch actuators, pitch sensors, and blades, only one box is drawn for each of these, since they are identical and cause the same effects.

According to Figure 4.2 a pitch actuator is internally controlled by a pitch controller connected to a pitch sensor; hence, forming a closed-loop system. This has some consequences for the effects of the pitch sensor faults, since a fixed output from this sensor turns the blade to one of its extreme positions depending on the sensor output compared to the pitch angle reference signal. The detection of the internal fault resulting in no output from a pitch sensor is assumed to already exist, since no measurement is sent to the controller in this situation. The control system is utilizing the last valid measurement from the sensor when this fault occurs.

Drive Train

The purpose of this subsection is to set up a FMEA for the drive train sub-model and to describe the faults and their propagation to the effects on the drive train.

The drive train consists of a low-speed shaft, a high-speed shaft, and a gearbox. A speed controller controls the speed of the high-speed shaft in the full load operation of the wind turbine. The FMEA scheme for the drive train is shown in Figure 4.4.

The high- and low-speed shafts can both have bearing damages. A damaged bearing implies uneven rotation of the drive train, since it is assumed that it can be modeled as a friction, which is dependent on the angle of the shaft. According to Section 3.10 the speed sensor is an encoder and can therefore possess a proportional error, which is denoted 'prop. error'.

Power System

The purpose of this subsection is to set up a FMEA for the power system sub-model and to describe the faults and their propagation to the effect on the power system.

The power system consists of two types of sensors; converter torque sensors and power sensors. Besides the sensors, a converter and generator are also part of the power system. Notice however that no faults in the generator are considered. The FMEA scheme for the power system is shown in Figure 4.5 on Page 30.

Besides the power controller, which ensures that the wind turbine produces the desired power in the full load region, the converter also has an internal controller. Even though multiple converters

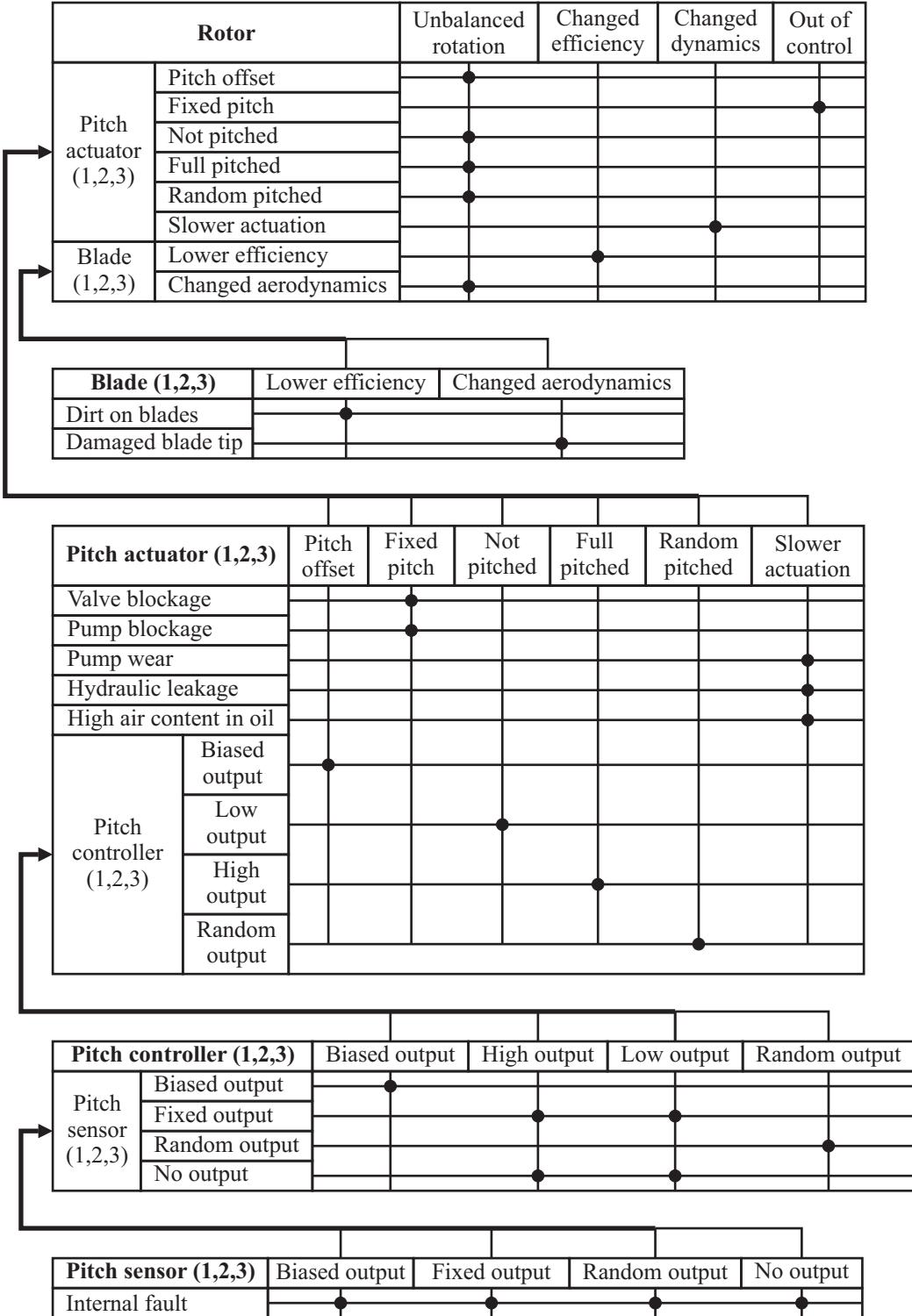


Figure 4.3: FMEA scheme illustrating the propagation of fault effects in the rotor sub-model.

and converter torque sensors exist, only one box is drawn for each of these, since they are identical and cause the same effects.

In this section it has been described how component faults propagate to end-effects in the sub-models of the wind turbine operating in closed loop. In the next section the faults are assessed by determining the severity of the end-effects and occurrences of the component faults.

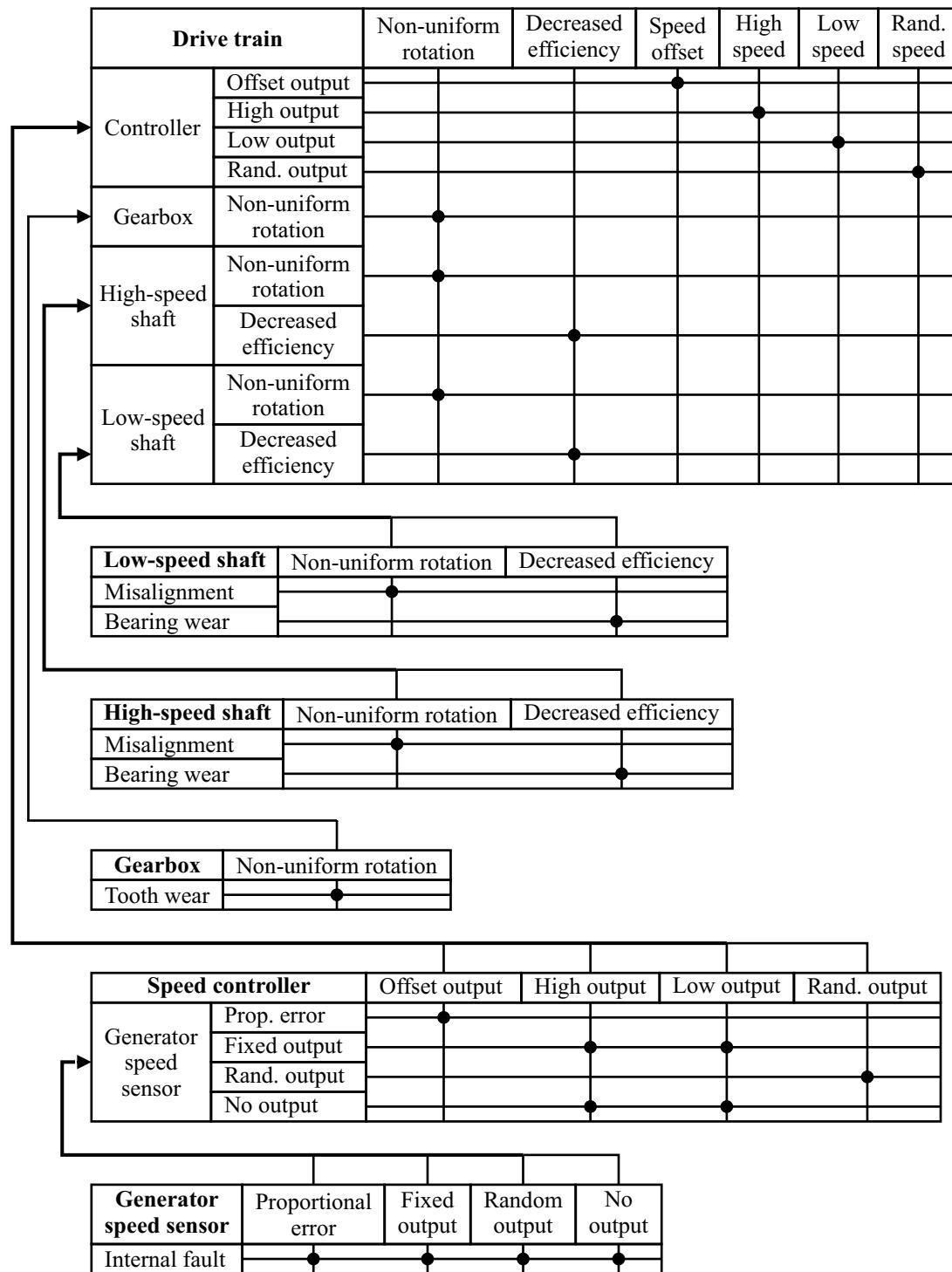


Figure 4.4: FMEA scheme illustrating the propagation of fault effects in the drive train sub-model.

Fault Analysis

Power system		Power offset	High power	Low power	Random power	Changed dynamics
Converter (1,2,3,4)	Biased output	•				
	High output		•			
	Low output			•		
	Random output				•	
	Slower actuation					•

Converter (1,2,3,4)		Biased output	High output	Low output	Random output	Slower actuation
Converter failure				•		•
Converter ctrl. (1,2,3,4)	Biased output	•				
	High output		•			
	Low output			•		
	Random output				•	

Converter ctrl. (1,2,3,4)		Biased output	High output	Low output	Random output
Controller	Biased output	•			
	High output			•	
	Low output				•
	Random output				•
Converter torque sensor (1,2,3,4)	Biased output	•			
	Fixed output		•		
	Random output			•	
	No output				•

Converter torque sensor (1,2,3,4)		Biased output	Fixed output	Random output	No output
Internal fault		•	•	•	•

Power controller		Biased output	High output	Low output	Random output
Pitch sensor (1,2,3)	Biased output	•			
	Fixed output		•		
	Random output			•	
	No output				•

Power sensor (1,2,3,4)		Biased output	Fixed output	Random output	No output
Internal fault		•	•	•	•

Figure 4.5: FMEA scheme illustrating the propagation of fault effects in the power system sub-model.

4.3 Fault Assessment

The purpose of this section is to evaluate the severity of the end-effects and the occurrence of the faults identified in Section 4.2, and to determine which of these should be treated in this project. Furthermore, causal relations are established to trace back the faulty components from the end-effects.

In this section, definitions of occurrence and severity are provided first. The occurrence rates are found from statistics about the distribution of failures described in the literature. To find the severities of the end-effects, simulations with fault injections are conducted to determine the severity indices. The simulations are carried out on the wind turbine model controlled by the reference controller described in Appendix C. The simulations and the descriptions of these appear in Appendix A to avoid disruption of the flow of the thesis. The section is finalized by a conclusion of the severity and occurrence analysis and a selection of the fault to be handled.

Faults are classified based on their likelihood of occurrence and the severity of their end-effect on scales from 1 to 10. These can be combined in the Severity Occurrence Index, which is obtained through multiplication of the severity and occurrence values [Izadi-Zamanabadi, 2004, p. 9]. The idea is that faults with the highest Severity Occurrence Index should have the highest concern. However, to draw higher attention to the severe failures of a system, a classification of faults based on their placement in Figure 4.6 is often used instead. This figure is based on [Quality Associates International, 2008].

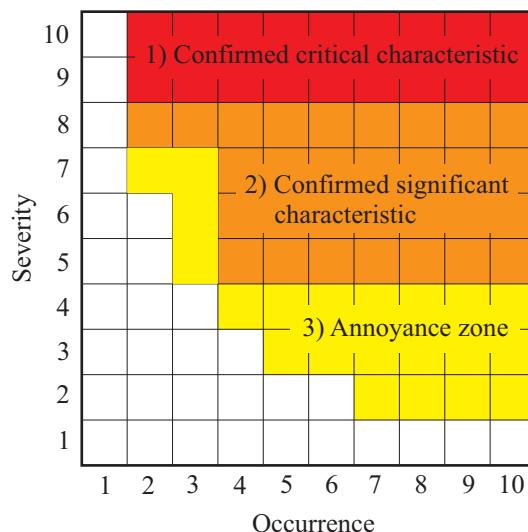


Figure 4.6: Graph to identify which faults should be accommodated. Faults with priority 1 (red) and 2 (orange) require actions, but for faults with priority 3 (yellow) actions are not required but only preferred.

Occurrence

Occurrence is the frequency of the fault and is quantified on a scale from 1 (unlikely) to 10 (inevitable). The occurrence scale is defined in Table 4.1 based on [Ireson *et al.*, 1996, p. 6.18]. Notice that the occurrence scale has a relative meaning.

Statistics about the distribution of failures experienced by wind turbines in Denmark, Germany, and Sweden can be found in [Ribrant, 2006] and [DOWEC team, 2002]. However, the failure rates in the literature relate to subsystems rather than individual components, which have to be estimated.

Severity

Severity is the potential harm which effects of faults inflict on the system. The severity scale is provided in Table 4.2 and originates from [Ireson *et al.*, 1996, p. 6.17]. Notice that severity scales found in most literature are directed towards the automotive industry, because the employment of

Fault Analysis

Probability of failure	Likely failure rate	Ranking
Very high: Almost inevitable failure.	$\geq 20\%$ of failures	10
	15-20% of failures	9
High: Repeated failure.	10-15% of failures	8
	5-10% of failures	7
Moderate: Occasional failure.	4-5% of failures	6
	3-4% of failures	5
	2-3% of failures	4
Low: Rare failure.	1-2% of failures	3
	0.1-1% of failures	2
Remote: Unlikely failure.	<0.1% of failures	1

Table 4.1: *Occurrence evaluation criteria and interpretation.*

FMEA originates from the early 1980s where American automotive companies began to incorporate FMEA into their product development process [Vijayaraghavan, 2003, p. 13]. However, engineers in a variety of industries have adopted and adapted the tool over the years. Therefore, the descriptions written in Table 4.2 should be considered with respect to wind turbines.

Effect	Severity of effect	Ranking
Hazardous without warning	Very high severity ranking when a potential failure mode affects safe turbine operation and/or involves noncompliance with government regulation without warning.	10
Hazardous with warning	Very high severity ranking when a potential failure mode affects safe turbine operation and/or involves noncompliance with government regulation with warning.	9
Very high	Turbine/item inoperable with loss of primary function.	8
High	Turbine/item operable but at a reduced level of performance. Customer dissatisfied.	7
Moderate	Turbine/item operable but comfort/convenience item(s) inoperable. Customer experiences discomfort.	6
Low	Turbine/item operable but comfort/convenience item(s) operable at a reduced level of performance. Customer experiences some dissatisfaction.	5
Very low	Fit and finish/squeak and rattle item does not conform. Defect noticed by most customers	4
Minor	Fit and finish/squeak and rattle item does not conform. Defect noticed by average customer.	3
Very minor	Fit and finish/squeak and rattle item does not conform. Defect noticed by discriminating customers.	2
None	No discernable effect.	1

Table 4.2: *Severity evaluation criteria.*

The simulations results obtained to determine the severity indices of the end-effects appear in Appendix A, as mentioned in the introduction. The next subsection summarizes the results and a selection of faults to be handled is provided.

Conclusion of Severity and Occurrence Analysis

To finalize the severity and occurrence analysis, all the end-effects are compared to identify the faults, which should be considered further throughout this thesis. The comparison is based on the information provided in Appendix A. This is accomplished by tracing back the faults, which cause the most severe end-effects. Among these faults the most frequent faults should be handled.

It is seen from the severity indices stated in Appendix A that several faults have severe effects on the wind turbine, and should therefore be handled. However, in order to narrow down the set of possible faults to be handled during this project, it is decided to focus on the faults related to the pitch sensors, pitch actuators, and generator speed sensor. The motivation behind this choice is that faults happening to the pitch system, which e.g. affect the rotor balance, are generally difficult to detect based on the typical outputs such as generator speed and output power. However, these faults result in increased fatigue loads on the wind turbine structure. It has further been seen that changed dynamics of the pitch system, caused by low pressure or high air content, may result in an unstable closed-loop system, as described in Table A.3 on Page 135. Concerning the measurement of the generator speed, wind turbine control systems rely on this in both the control regions as shown in Figure 2.4 on Page 8. Therefore, it is essential to diagnose and accommodate any troubles regarding this particular measurement.

The faults to be diagnosed and accommodated in this project are stated in Table 4.3. Since a random output from a sensor is unlikely to occur, this fault is disregarded for the considered sensors. Regarding fixed or no outputs from a pitch sensor, these particular faults are disregarded since it is assumed that the wind turbine cannot be controlled satisfactorily if not having this measurement.

Effect	Fault	O	S
Rotor			
Unbalanced rotation	Pitch sensor (1,2,3) - Biased output	3	6
Changed dynamics	Pitch actuator (1,2,3) - Pump wear	4	5
	Pitch actuator (1,2,3) - Hydraulic leakage	3	8
	Pitch actuator (1,2,3) - High air content in oil	5	5
Out of control	Pitch actuator (1,2,3) - Valve blockage	3	8
	Pitch actuator (1,2,3) - Pump blockage	2	9
Drive train			
Speed offset	Generator speed sensor - Proportional error	4	4
High speed	Generator speed sensor - Fixed output	2	9
	Generator speed sensor - No output	3	9
Low speed	Generator speed sensor - Fixed output	2	8
	Generator speed sensor - No output	3	8

Table 4.3: *Faults which should be diagnosed and accommodated and their associated severity and occurrence indices.*

In this section the occurrence of the faults and the severity of their end-effects have been determined, in order to perform a qualified selection of the faults to be diagnosed and accommodated in this project. It is decided to focus on the faults related to the pitch sensors, pitch actuators, and generator speed sensor. In the next section a structural analysis of the system is presented to determine where redundancy relations can be exploited in the fault diagnosis.

4.4 Structural Analysis

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

The structural analysis is set up according to the procedure shown in the bullet list below [Izadi-Zamanabadi, 1999, p. 17].

- **Structural Relations:** Determine the structural relations of the wind turbine model and organize the structural relations in an incidence matrix. An incidence matrix shows structural relations in the model and separates known from unknown variables.
- **Matching:** Perform a matching of the unknown variables. A matching shows how the unknown variables can be estimated from the known variables using the constraints in the model.
- **Analytical Redundancy Relations:** Determine the analytical redundancy relations and sensor fusion possibilities.

Structural Relations

In this section the mutual dependence of the variables must be described; hence, the basis for the structural relations is the functional equations set up in Chapter 3. The difference between functional equations and structural relations can be demonstrated by rewriting Eq. (3.13) on Page 16 as shown below.

$$T_a(t) - T_l(t) - B_r \dot{\theta}_r(t) - J_r \ddot{\theta}_r(t) = 0 \text{ Nm} \quad (\text{functional equation}) \quad (4.1)$$

$$f(T_a(t), T_l(t), \dot{\theta}_r(t), \ddot{\theta}_r(t)) = 0 \quad (\text{structural relation}) \quad (4.2)$$

Hence, the difference between functional equations and structural relations is that functional equations explains how variables are linked together, while structural relations only show the variables which are linked together.

To form the structural relations it is necessary to use the system equations set up in Chapter 3. These are all present in the state space representation of the system shown in Eq. (3.30) on Page 22. In addition to the system equations the measurement equations, which can be deducted from Section 3.10, also form some structural relations. Notice that the measurement of the wind speed is not utilized, of reasons explained in Section 3.10 on Page 23.

Besides the system and measurement equations there are some differential relations between the variables and their derivatives. These relations, called differential constraints, are special, since the following holds: a signal $x(t)$ cannot be computed from its derivative $\dot{x}(t)$, but $\dot{x}(t)$ can be computed from $x(t)$. This is called integral causality, and is due to the lack of knowledge about the initial condition, $x(0)$, as indicated below [Blanke *et al.*, 2006, p. 129].

$$x(t) = \int_0^t \dot{x}(\tau) d\tau + x(0) \quad (4.3)$$

An incidence matrix is a way to represent the structural relations by separating the known variables \mathcal{K} from the unknown variables \mathcal{X} . When using the structural relations in an incidence matrix it is necessary to distinguish between known variables and unknown variables, since the incidence matrix is used in the matching. The known variables are control signals, measured variables, and known parameters, while structural relations show the variables which are linked together. The unknown variables are unmeasured variables and unknown parameters of the system. Combined the known and unknown variables form the entire set of variables which are denoted $\mathcal{Z} = \mathcal{K} \cup \mathcal{X}$.

The structural relations of the wind turbine model all appear in the incidence matrix shown in Table 4.4, and are divided into three types of constraints, combined denoted \mathcal{C} :

- d denotes a differential constraint.
- c denotes a constraint imposed by the system equations.
- m denotes a constraint from the measurement equations.

Furthermore, an 'x' means that a variable cannot be calculated from the structural relation due to integral causality. Some of the '1's are also underlined in the incidence matrix to show the matching of the unknown variables, which is explained in the next subsection. The ranking in the rightmost column relates to the matching, and indicates how many unknown variables there are used in the matching of every unknown variable.

	Unknown Variables												Known Variables						Ranking			
	\dot{T}_{g}	T_{g}	\dot{x}_{t}	x_{t}	\ddot{x}_{t}	$\ddot{\beta}$	$\dot{\beta}$	θ_{Δ}	$\dot{\theta}_{\Delta}$	ω_{g}	$\dot{\omega}_{\text{g}}$	ω_{r}	$\dot{\omega}_{\text{r}}$	v_{w}	$T_{\text{g,ref}}$	β_{ref}	$P_{\text{g,mes}}$	$\omega_{\text{g,mes}}$	$T_{\text{g,mes}}$	β_{mes}	$\omega_{\text{r,mes}}$	$\dot{x}_{\text{t,mes}}$
d_{1-4}	$\underline{1}$	x																			1	
d_5		1	x																			
d_6		x		1																		
d_{7-9}					$\underline{1}$	x															2	
d_{10-12}						$\underline{1}$	x														1	
d_{13}							1	x														
d_{14}									$\underline{1}$	x											1	
d_{15}											$\underline{1}$	x									1	
c_{1-4}	1	1																			2	
c_5		1	1	1	1			1					1	1								
c_6-8						1	1	1													3	
c_9									$\underline{1}$		1	1									1	
c_{10}		1								$\underline{1}$	1	1									2	
c_{11}		1					1			1	1	1	1									
m_1		1									1											
m_2												$\underline{1}$									0	
m_{3-6}		1																			0	
m_{7-9}									$\underline{1}$												0	
m_{10}																					0	
m_{11}										$\underline{1}$											0	

Table 4.4: Incidence matrix of the wind turbine model showing the structural relations between the variables and a possible matching. The underlined '1's show the matching of the unknown variables.

Matching

A matching of the unknown variables is required to identify the monitorable part of the system, i.e. the subsystem in which faults can be detected and isolated. This part of the system must be observable to be in the over-constraint part of the system (\mathcal{C}, \mathcal{Z}). This means that there should be more constraints describing this part of the system than unknown variables [Blanke *et al.*, 2006, p. 110].

The matching of the unknown variables is performed as shown in Figure 4.7, where the unknown variables are matched from the known variables using the structural relations. The unknown variables are matched using as few structural relations as possible to get the matching as close as possible to the known variables. This involves minimizing the so-called ranking, which indicates the number of unknown variables utilized in each matching. The structural relations which are not used to estimate the unknown variables are shown in Figure 4.7 using the red lines. These relations are ARR's and can be used for fault detection and fault identification purposes.

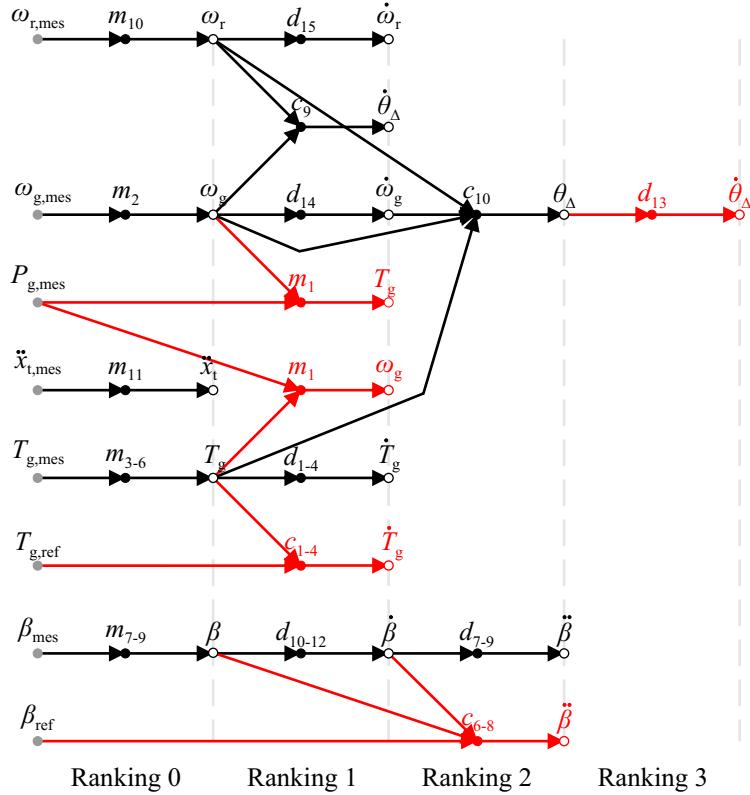


Figure 4.7: Matching of unknown variables through known variables. The black lines show the matching from Table 4.4 and the red lines are the redundant relations. The gray dots indicate known variables, the black dots indicate structural relations, while the circles indicate unknown variables.

The matching of the unknown variables is also shown in Table 4.4, where an underlined variable indicates that it is calculated from the particular constraint. This indicates that the unknown variable in the column can be determined from the known variables and the constraint written in the associated row. The matching starts at the measurement equations and propagates to the system constraints and differential constraints.

Analytical Redundancy Relations

The matching shows that the position of the tower, the velocity of the tower, and the wind speed are not observable. However, note that the rotor effective wind speed is observable. Therefore, no faults in these variables can be detected directly from the following residuals. This is clear, since these three variables cannot be matched. The remaining system is observable, and has some

redundancy relations, since some constraints are not utilized in the matching. The constraints that are not matched can be utilized to detect and isolate faults. The unmatched constraints are c_{1-4} , c_{6-8} , d_{13} , and m_1 , which are summarized in Table 4.5. Notice, the variables which are not part of any of the ARRs are deleted from this table.

	Unknown Variables								Known Variables		
	\dot{T}_g	T_g	$\dot{\theta}_\Delta$	θ_Δ	ω_g	$\ddot{\beta}$	$\dot{\beta}$	β	$T_{g,\text{ref}}$	β_{ref}	$P_{g,\text{mes}}$
c_{1-4}	1	1							1		
c_{6-8}						1	1	1		1	
d_{13}			1	x							
m_1		1			1						1

Table 4.5: Incidence matrix of the wind turbine model showing the analytical redundancy relations.

From the analytical redundancy relations it is possible to create residuals for detecting faults. Figure 4.8 to Figure 4.12 show the structural relations which can be traced back from the residuals. The gray dot at the left side of each figure indicates the available information.

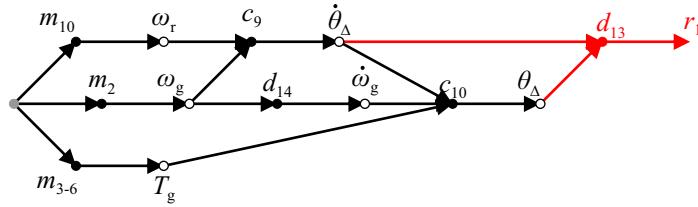


Figure 4.8: Illustration of the connection between the constraints used in the matching and the residual r_1 , which is generated from constraint d_{13} . The red lines between variables indicate the relations which generate the residual.

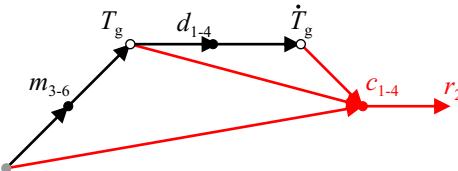


Figure 4.9: Illustration of the connection between the constraints used in the matching and the residual r_2 , which is generated from constraint c_{1-4} . The red lines between variables indicate the relations which generate the residual.

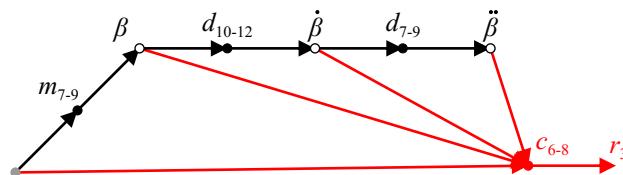


Figure 4.10: Illustration of the connection between the constraints used in the matching and the residual r_3 , which is generated from constraint c_{6-8} . The red lines between variables indicate the relations which generate the residual.

From Figure 4.8 to Figure 4.12 it is possible to determine the residuals which become non-zero when one of the structural relations used to generate the residuals is violated. A relation between structural relations and residuals are shown in Table 4.6.

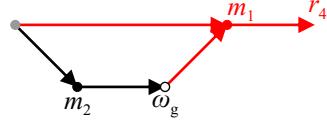


Figure 4.11: Illustration of the connection between the constraints used in the matching and the residual r_4 , which is generated from constraint m_1 . The red lines between variables indicate the relations which generate the residual.

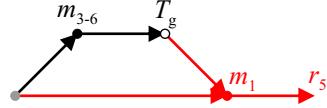


Figure 4.12: Illustration of the connection between the constraints used in the matching and the residual r_5 , which is generated from constraint m_1 . The red lines between variables indicate the relations which generate the residual.

Structural Relation	r_1	r_2	r_3	r_4	r_5
d_{1-4}		x			
d_5					
d_6					
d_{7-9}			x		
d_{10-12}			x		
d_{13}	x				
d_{14}	x				
d_{15}					
c_{1-4}		x			
c_5					
c_{6-8}			x		
c_9	x				
c_{10}	x				
c_{11}					
m_1				x	x
m_2	x			x	
m_{3-6}	x	x			x
m_{7-9}			x		
m_{10}	x				
m_{11}					

Table 4.6: List of non-zero residuals (x), when a fault is introduced in one of the structural relations utilized in the matching.

From the table it can be seen that only faults in m_1 , m_2 , and m_{3-6} can be detected and isolated from the residuals, since these constraints have unique combinations of residuals. This means that other approaches for fault diagnosis, e.g. based on functional equations, have to be developed in order to detect the remaining faults in the wind turbine.

Notice that this structural analysis does not reveal all redundancies in the system, as only the structural relations of the system are utilized. An example of additional redundancy is the relation between the generator and rotor speed measurements. When averaging these two values they should be proportional to each other. This cannot be seen from the structural relations, since the dynamics of the system is not exploited.

In this section the analytical redundancy relations have been determined from the structural relations of the wind turbine model. These redundancy relations make it possible to detect a number of faults in the system. However, other approaches are needed to diagnose all faults shown in Table 4.3

on Page 33. In the next section the faults are categorized into abrupt and incipient faults, and the dynamics of the incipient faults are specified to enable the diagnosis algorithms to be designed according to the dynamics of the faults.

4.5 Fault Specification

The purpose of this section is to model the selected faults, which are listed in Table 4.3 on Page 33. The incipient faults are modeled by their magnitude and the rate at which they can be introduced. The abrupt faults are modeled by rewriting the system equations and measurement equations, corresponding to the changes enforced by the faults. The section is divided into three subsections describing: pitch sensor faults, pitch actuator faults, and generator speed sensor faults. In these sections parameters specifying the dynamics of the incipient faults are needed. These parameters are therefore listed in Table 4.7.

Fault	Specification
Pitch Sensor	
Bias	$\dot{\beta}_{\text{bias}}(t) \in [-1^\circ/\text{month}, 1^\circ/\text{month}]$ $\beta_{\text{bias}}(t) \in [-7^\circ, 7^\circ]$
Pitch Actuator	
High Air Content	$\dot{\alpha}_{\text{ha}}(t) \in [-1/\text{month}, 1/\text{month}]$ $\alpha_{\text{ha}}(t) \in [0, 1]$
Pump Wear	$\dot{\alpha}_{\text{pw}}(t) \in [0, 1/(20 \text{ years})]$ $\alpha_{\text{pw}}(t) \in [0, 1]$
Hydraulic Leakage	$\dot{\alpha}_{\text{hl}}(t) \in [0, 1/(100 \text{ s})]$ $\alpha_{\text{hl}}(t) \in [0, 1]$
Generator Speed Sensor	
Proportional Error	$\dot{\alpha}_{\text{pe}}(t) \in [-1/\text{month}, 1/\text{month}]$ $\alpha_{\text{pe}}(t) \in [-0.1, 0.1]$

Table 4.7: Specification of the ranges and the rate limits of the incipient faults.

Pitch Sensor Faults

An internal fault in a pitch sensor may cause biased output. This fault is modeled in the following subsection.

Biased Output from Pitch Sensor

A biased pitch sensor measurement affects both the closed-loop pitch system and the pitch angle measurement. When the bias is introduced, the pitch actuator model shown in Eq. (3.29) on Page 21 is modified as shown in Eq. (4.4a), and likewise the modified measurement equation is shown in Eq. (4.4b).

$$\ddot{\beta}(t) = -2\zeta\omega_n\dot{\beta}(t) - \omega_n^2(\beta(t) + \beta_{\text{bias}}(t)) + \omega_n^2\beta_{\text{ref}}(t - t_d) \quad [\text{ }^\circ/\text{s}^2] \quad (4.4a)$$

$$\beta_{\text{mes}}(k) = \beta(k) + \beta_{\text{bias}}(k) + v_\beta(k) \quad [^\circ] \quad (4.4b)$$

A bias can either be a result of inaccurate calibration of the pitch system or be an incipient fault. To characterize the incipient fault, the rate of $\beta_{\text{bias}}(t)$ together with its range is specified in Table 4.7.

Pitch Actuator Faults

The faults considered for the pitch actuators are: pump wear, hydraulic leakage, high air content in the hydraulic oil, valve blockage, and pump blockage. These faults are modeled separately in the next subsections. Notice that the parameters for the faulty pitch actuator are shown in Table 4.8.

Fault	Parameters
No fault	$\omega_n = 11.11 \text{ rad/s}$, $\zeta = 0.6$
High Air Content in the Oil	$\omega_{n,ha} = 5.73 \text{ rad/s}$, $\zeta_{ha} = 0.45$
Pump Wear	$\omega_{n,pw} = 7.27 \text{ rad/s}$, $\zeta_{pw} = 0.75$
Hydraulic Leakage	$\omega_{n,hl} = 3.42 \text{ rad/s}$, $\zeta_{hl} = 0.9$

Table 4.8: Parameters for the pitch system under different conditions. The normal air content in the hydraulic oil is 7%, whereas high air content in the oil corresponds to 15%. Pump wear represents the situation of 75% pressure in the pitch system while the parameters stated for hydraulic leakage corresponds to a pressure of only 50%.

Pump Wear

Pump wear is an incipient fault which is introduced very slowly and results in low pump pressure. This fault changes the parameters of the closed-loop pitch system described in Eq. (3.29) on Page 21 as shown below.

$$\ddot{\beta}(t) = -2\tilde{\zeta}(t)\tilde{\omega}_n(t)\dot{\beta}(t) - \tilde{\omega}_n^2(t)\beta(t) + \tilde{\omega}_n^2(t)\beta_{ref}(t - t_d) \quad [\text{°}/\text{s}^2] \quad (4.5)$$

with

$$\begin{aligned} \tilde{\zeta}(t) &= (1 - \alpha_{pw}(t))\zeta + \alpha_{pw}(t)\zeta_{pw} \quad [.] \\ \tilde{\omega}_n(t) &= (1 - \alpha_{pw}(t))\omega_n + \alpha_{pw}(t)\omega_{n,pw} \quad [\text{rad/s}] \end{aligned}$$

where:

$\alpha_{pw}(t)$ is an indicator for the wear on the pump [.]

When $\alpha_{pw}(t) = 0$ the pump delivers the nominal pressure, but as $\alpha_{pw}(t)$ goes to one the pressure drops. The dynamics of $\alpha_{pw}(t)$ is specified in Table 4.7. Notice that $\dot{\alpha}_{pw}(t) \geq 0$ for all t , since the wear is irreversible without replacing the pump. It is estimated by kk-electronic a/s that the fault described by $\alpha_{pw} = 1$, corresponding to a pressure level of 75%, can emerge after approximately 20 years of operation.

Hydraulic Leakage

Hydraulic leakage is an incipient fault, which is introduced considerably faster than e.g. pump wear. Hydraulic leakage results in changed parameters for the closed-loop pitch system as shown below.

$$\tilde{\zeta}(t) = (1 - \alpha_{hl}(t))\zeta + \alpha_{hl}(t)\zeta_{hl} \quad [.] \quad (4.7a)$$

$$\tilde{\omega}_n(t) = (1 - \alpha_{hl}(t))\omega_n + \alpha_{hl}(t)\omega_{n,hl} \quad [\text{rad/s}] \quad (4.7b)$$

The constraints on the introduction of the fault are shown in Table 4.7. Again $\dot{\alpha}_{hl}(t) \geq 0$ for all t , since a leakage cannot be reversed without repair of the system. Notice that the pressure for $\alpha_{hl} = 1$ corresponds to 50% of the nominal pressure.

High Air Content in Oil

High air content in the oil is an incipient fault, which changes the parameters for the closed-loop pitch system as shown below.

$$\tilde{\zeta}(t) = (1 - \alpha_{ha}(t))\zeta + \alpha_{ha}(t)\zeta_{ha} \quad [.] \quad (4.8a)$$

$$\tilde{\omega}_n(t) = (1 - \alpha_{ha}(t))\omega_n + \alpha_{ha}(t)\omega_{n,ha} \quad [\text{rad/s}] \quad (4.8b)$$

The constraints on the introduction of the fault are shown in Table 4.7. In contrast to pump wear and hydraulic leakage the fault can disappear; hence, $\dot{\alpha}_{\text{ha}}(t)$ can be both positive and negative. The extreme values caused by $\alpha_{\text{ha}} = 0$ and $\alpha_{\text{ha}} = 1$ correspond to air contents of 7% and 15% in the hydraulic oil.

Valve Blockage

A valve blockage is assumed to happen abruptly and might only affect one blade, as it is equivalent to a situation where either Valve 1, Valve 2, or Valve 3 in Figure 3.14 on Page 20 is blocked. The fault modifies the equation describing the pitch system as shown below.

$$\beta_i(t) = \beta_i(t_f) \quad \forall t > t_f \quad (4.9)$$

where:

t_f is the time where the fault occurs [s]

Pump Blockage

A pump blockage is assumed to happen abruptly and affects all three blades. It is assumed equivalent to closing Valve 4 in Figure 3.14 on Page 20 and modifies the equation describing the pitch system as shown in Eq. (4.9) for $i = 1, 2, 3$.

Generator Speed Sensor Faults

The faults that are considered for the generator speed sensor are: proportional error on the output, fixed output, and no output. These faults are modeled separately in the next subsections.

Proportional Error on Generator Speed Sensor

A proportional error on the generator speed sensor changes the sensor gain as shown below.

$$\omega_{g,\text{mes}}(k) = \underbrace{(1 + \alpha_{\text{pe}}(k))}_{\text{sensor gain}} \omega_g(k) + v_{\omega_g}(k) \quad [\text{rad/s}] \quad (4.10)$$

The fault can either be a result of a faulty configuration of a sensor or be an incipient fault. To characterize the incipient fault, its rate of change together with its value space is specified in Table 4.7.

Fixed Output from Generator Speed Sensor

A fixed output from the generator speed sensor is an abrupt fault, which can happen at any time resulting in the following measurement equation:

$$\omega_{g,\text{mes}}(k) = \omega_{g,\text{mes}}(k_f) \quad \forall k > k_f \quad (4.11)$$

No Output from Generator Speed Sensor

No output from a generator speed sensor results in the same changes in the measurement equation as a fixed output does. In contrast to a fixed output the control system is notified when no output is received from the sensor.

In this section the considered faults have been specified. The next section describes the remedial actions that must be taken to diagnose and accommodate the considered faults.

4.6 Remedial Action Selection

The purpose of this section is to determine the actions that must be taken to stop propagation of component faults to become failures on the system level. Furthermore, the requirements to the fault diagnosis and accommodation must be determined. A table showing the methods used in the design of the diagnosis and accommodation algorithms are shown in Table 4.9 on Page 44. Only the faults which are selected in Table 4.3 on Page 33 are considered in this section.

Pitch Sensor Faults

The pitch sensors can be affected by a biased output. Requirements to the fault diagnosis and accommodation of this fault are explained in the following subsection.

Biased Output from Pitch Sensor

As shown in Table 4.3 on Page 33 a biased output from a pitch sensor results in unbalanced rotation of the rotor. This effect, which should be noticeable on the measurement of the tower acceleration, is utilized to detect that one blade has an offset and to isolate the faulty pitch sensor. The detection should be able to reveal a pitch offset having a magnitude greater than 0.25° to avoid long term damage on the wind turbine.

To accommodate a biased pitch angle measurement, a bias-corrected measurement has to be provided to the controller. This correction should enable the controller to keep all blades equally pitched, despite the biased pitch measurement. The accommodation of the biased pitch measurement is required to keep the difference between the individual pitch angles less than 5° at all times. This requirement is introduced, since kk-electronic a/s has stated that differences above this bound may cause severe structural damage to the wind turbine.

Pitch Actuator Faults

Five different faults can happen to the pitch actuators. These are: pump wear, hydraulic leakage, high air content in the hydraulic oil, valve blockage, and pump blockage. Notice that pump wear, hydraulic leakage, and high air content in the oil are incipient faults, which change the dynamics of the pitch system and are therefore treated together. Additionally, valve blockage and pump blockage are treated as one fault, since they introduce similar effects on a pitch actuator.

Changed Dynamics of the Pitch Actuator System

Hydraulic leakage eventually makes the pitch actuators uncontrollable. Therefore, it should be detected before the pressure has drop to half the nominal pressure. This implies that a hydraulic leakage should be detected within 100 s to satisfy this requirement for the assumed fastest evolving hydraulic leakage.

The accommodation of pump wear and high air content in the oil is similar. Both active and passive fault-tolerant control methods are considered to be relevant for the accommodation of these faults. To delimit the extend of this project, fault-tolerant controllers are only designed for accommodation of high air content in the oil. Notice that the fault diagnosis is only utilized by the active fault-tolerant control approach.

To accommodate hydraulic leakage in the oil of the pitch system, the wind turbine must be shut down prior to a complete loss of controllability of the pitch system. Though, this accommodation is not going to be designed to limit the extend of this project, as the accommodation requires further considerations to shut down the wind turbine as fast as possible while minimizing the mechanical loads.

Valve Blockage and Pump Blockage

To detect a valve blockage or pump blockage a hypothesis test should be designed, since these are abrupt faults, in contrast to the other faults in the pitch system. The detection of the faults should happen as fast as possible to enable a fast accommodation, to avoid damaging the wind turbine.

Notice that the pitch system is not actuated in partial load operation, which makes it necessary to inject an auxiliary signal into the system to detect any fault in the pitch system.

The fault should only be detected not accommodated, since kk-electronic a/s has an accommodation system for this fault. Their accommodation system consists of a series of valves which are capable of by-passing hydraulic oil around the stuck valve with a hydraulic line from an accumulator. This makes it possible to pitch the blades out of the wind; hence, shutting down the wind turbine.

Generator Speed Sensor Faults

The three faults considered for the generator speed sensor are: proportional error, fixed output, and no output. The only difference between the faults 'fixed output' and 'no output' from the generator speed sensor is that fixed output has to be detected. Therefore, these two faults are treated together.

Proportional Error on Generator Speed Sensor

To detect a proportional error on the generator speed sensor the redundant sensor information from the rotor speed sensor can be utilized as explained in Section 4.4. To be certain that the wind turbine is operated below an acceptable maximum speed, a safety margin of 10% of the rated generator speed exists. The fault detection is required to detect the fault before the proportional error equals 50% of the safety margin. This means that a proportional error greater than ± 0.05 should be detected.

To accommodate for the proportional error, the size of this has to be estimated, and a corrected generator speed measurement must be fed to the controller. The proportional error cannot be estimated by use of the sensor redundancy, since the faulty sensor cannot be isolated by only using information from the two speed sensors. Note that in theory a relation between generator torque and output power can also be used to estimate the generator speed, but due to e.g. the time varying efficiency of the power system, this relation is discarded. Therefore, the proportional error must be estimated from the 3P frequency, which is infallible and is present on e.g. the generator speed.

Fixed Output and No Output from Generator Speed Sensor

From Table 4.3 on Page 33 it is clear that fixed or no output from the generator speed sensor has severe consequences for the wind turbine; hence, the detection of fixed output of the generator speed sensor has to be detected quickly. The detection of the fault must exploit that no noise is present on the measurement after the fault has occurred.

The accommodation of the faulty generator speed sensor must be to reconfigure the controller to utilize the redundant sensor information, which is available due to the measurement of the rotor speed. However, the variance of the measurement noise on the rotor speed sensor is larger than on the generator speed sensor, i.e. the performance of the controller degrades accordingly, making it necessary to reconfigure the controller in full load operation.

Both faults described in this subsection will be referred to as fixed output from the generator speed sensor in the rest of the thesis, since the handling of a fixed output can be used for no output as well, by excluding the fault detection.

In this section all remedial actions involved with the considered faults have been selected. In the remaining part of the thesis, all incipient faults are introduced in maximum 30 min., to limit the simulation time involved in the design and test procedures, which already exceeds multiple weeks.

In this chapter a fault analysis of the wind turbine control system has been accomplished, by identifying the faults that can happen to the system and determining their propagations through the system. To select the faults to be treated in this thesis, their frequency of occurrence and the severity of their end-effects have been determined. The selected faults and the methods, which should be applied for diagnosing and accommodating the faults are summarized in Table 4.9. In the next chapter the fault diagnosis algorithms used in the active fault-tolerant controller are designed.

Component	Fault	Fault Diagnosis Method	Fault Accommodation Method
Pitch sensor	Biased output	Diagnosis of unbalance in rotor plane	Signal correction of measurement and reference signals
	High air content in oil	Multiple-model parameter estimation	Active and passive fault-tolerant LPV control <i>(delimited from this project)</i>
	Pump wear		
Pitch actuator	Hydraulic leakage	CUSUM test	Shutdown of the wind turbine <i>(delimited from this project)</i>
	Valve blockage	Active fault diagnosis hypothesis testing (<i>PL</i>)	
	Pump blockage	Passive fault diagnosis hypothesis testing (<i>FL</i>)	
Generator speed sensor	Proportional error	Peak detection and estimation of 3P frequency	Signal correction of measurement signal
	Fixed output	Detection of zero measurement noise	Signal correction of measurement signal (<i>PL</i>)
	No output	None (<i>the control system is notified</i>)	Active and passive fault-tolerant LPV control (<i>FL</i>)

Table 4.9: Overview of the selected faults and the applied fault diagnosis and fault accommodation methods. Notice that for some faults different methods are utilized in relation to handling faults in the partial load region (*PL*) and in the full load region (*FL*).

Fault Diagnosis

The purpose of this chapter is to provide an explanation of the designed fault diagnosis algorithms. These algorithms can be used both for condition monitoring purposes and in an active fault-tolerant control system to provide the supervisor with sufficient information to reconfigure the controller.

It is decided to create a model-based fault diagnosis system, which relies on measured as well as estimated variables. Therefore, a reconfigurable estimator must be designed to make it possible to design the entire fault diagnosis system according to the structure shown in Figure 5.1.

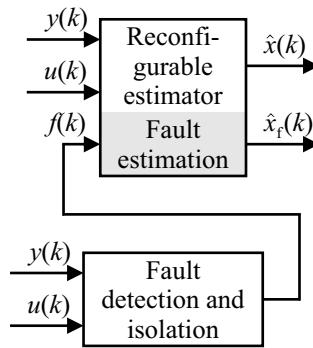


Figure 5.1: *Fault diagnosis algorithms reconfiguring the estimator in case of faults. The reconfigurable estimator provides estimates of the system states and the exogenous input.*

The figure shows an estimator being reconfigured by the fault vector, $f(k)$, provided by the fault detection and isolation block. The reconfiguration makes it possible for the estimator to provide a fault-corrected state estimate, $\hat{x}(k)$, and to estimate additional states, $x_f(k)$, introduced by the faults. The fault-corrected state estimates enable the fault diagnosis algorithms in the common structure to maintain operation even when other faults exist. This makes the fault diagnosis system able to diagnose multiple simultaneous faults.

The reconfigurable estimator consists of both a reconfigurable wind speed estimator and a reconfigurable extended Kalman filter. Hence, this chapter includes the design of these estimators and the fault diagnosis algorithms for the faults shown in Table 4.9. The reconfigurable wind speed estimator provides an input to the reconfigurable extended Kalman filter; hence, it is explained first.

5.1 Reconfigurable Wind Speed Estimator

It is desirable to have an estimate of the effective wind speed available as input to the reconfigurable extended Kalman filter, since the wind speed acts as the driving force of the wind turbine.

An anemometer is located on top of the nacelle, but its measurement cannot be used as input to the state estimator of reasons described in Section 3.10. Therefore, the purpose of this section is to design a wind speed estimator capable of estimating the effective wind speed from the system model and the available measurements. Additionally, the wind speed estimator should be able to reconfigure, if the measurement of the generator speed becomes unavailable or if the outputs of the pitch sensors are biased. This requirement exists since the estimator should be tolerant towards the same faults as the control system can experience. These faults are outlined in Table 4.9.

There exist several methods for designing an effective wind speed estimator. In [Thiringer and Petersson, 2005, pp. 7-8] the wind speed is calculated from an approximation of the relation between the effective wind speed and the output power in the partial load operation. Similarly, a relation between the pitch angle and the wind speed is used in the full load operation. A disadvantage of this method is that it only relies on the pitch angle in full load operation even though the rotor speed

also affects the relation; consequently, the approximation gets inaccurate. Furthermore, scheduling between the partial load and the full load regions is necessary. In [van der Hooft and van Engelen, 2004] a similar approach is utilized, but instead of an approximation of the relation between the wind speed and either the output power or the pitch angle, the C_p -table is utilized. This makes the approximation more accurate, but the method is not suited for the considered wind turbine, since the measurement of the output power is contaminated by a lot of noise.

The method used to design the wind speed estimator in this project originates from [Østergaard et al., 2007], and is based on the block diagram shown in Figure 5.2. The principle is to estimate the aerodynamic torque using an input estimator and then use this estimated torque in a lookup table, based on the C_p -table, to find the effective wind speed. The method was originally designed for collective pitching, but due to the possibility of faults in the pitch system, it cannot be assumed that all pitch angles are identical at all times. Therefore, the method is extended to be capable of handling three different pitch angles.

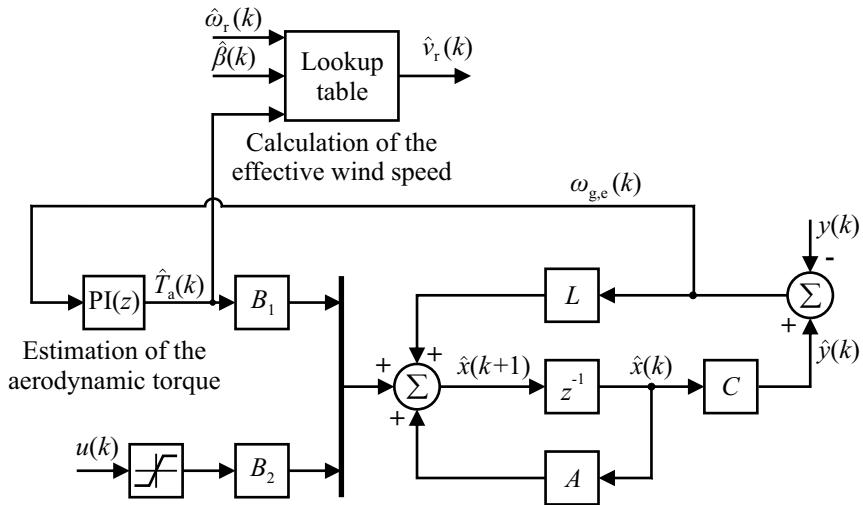


Figure 5.2: Block diagram of the wind speed estimator consisting of a state estimator with an input estimator providing the aerodynamic torque estimate. Notice that $\text{PI}(z)$ denotes a PI controller. The wind speed estimator utilizes the estimated aerodynamic torque in a lookup table to find the effective wind speed.

The wind speed is estimated following this procedure:

1. A Kalman estimator estimates the generator speed based on the drive train model, control signals, and the estimated aerodynamic torque.
2. The aerodynamic torque is unknown and is therefore estimated by a PI controller, having the difference between the measurement and the estimated generator speed as input.
3. The estimate of the aerodynamic torque makes it possible to estimate the effective wind speed from the table describing the aerodynamic properties of the rotor, when the pitch angles are known.

The division between a state estimator and an input estimator makes it possible to design the wind speed estimator in two steps, making the design procedure easier.

The section is divided into a number of parts each having their own subsection explained below.

- **State Estimator:** Design the state estimator based on the drive train model.
- **Input Estimator:** Design the input estimator, which estimates the aerodynamic torque.
- **Calculation of Effective Wind Speed:** Set up equations for calculating the wind speed based on the aerodynamic model.
- **Verification of Reconfigurable Wind Speed Estimator:** Verify the reconfigurable wind speed estimator in its two configurations and with different pitch angles.

Notice that using the aerodynamic torque as input to the model eliminates the partial derivatives of the aerodynamic torque, which are part of the model shown in Eq. (3.30) on Page 22. Additionally, the model is reduced to only including the drive train and pitch system models, since estimation of all states is not necessary to estimate the effective wind speed. Finally, only a reduced measurement vector is utilized, to feed the entire estimation error to the input estimator. This implies that only $T_g(t)$, $\beta(t)$, and $\omega_g(t)$ are utilized in the normal scenario, while $T_g(t)$, $\beta(t)$, and $\omega_r(t)$ are used when the generator speed measurement is unavailable.

In the next subsection the design of the state estimator is explained.

State Estimator

The state estimator utilized in the effective wind speed estimator must provide estimates of the rotor speed and pitch angles to the lookup table, from which the effective wind speed is calculated as shown in Figure 5.2. The state estimator is a full-order Kalman estimator and is explained in this subsection.

To design the full-order Kalman estimator, the MATLAB function `dlqe` is utilized. This function calculates a constant Kalman gain based on the discrete system matrices, additive measurement noise variances, and additive state noise variances:

$$x(k+1) = \Phi x(k) + \Gamma_1 w(k) + \Gamma_2 u(k) \quad (5.1a)$$

$$y(k) = Hx(k) + D_1 v(k) + D_2 u(k) \quad (5.1b)$$

$$Q = E(ww^T)$$

$$R = E(vv^T)$$

where:

Q is the variance matrix of the state noise

R is the variance matrix of the measurement noise

The estimator only utilizes measurements of the generator speed, generator torque, and pitch angles as input. The variances of the additive measurement noise are known from Table 3.1 on Page 23; these values determine the elements in the measurement noise matrix, R .

During the tuning of the state noise matrix, Q , the correct aerodynamic torque is utilized as input, because at this point the input estimator is not designed. This is one of the nice features of the utilized design procedure. When the measurement of $\omega_g(t)$ is unavailable the state noise variances are changed, to improve the performance of the state estimator after reconfiguration.

A full-state Kalman estimator has now been designed. The second step is to design an input estimator capable of estimating the aerodynamic torque from the estimation error of the generator speed.

Input Estimator

A PI controller is utilized for estimating the aerodynamic torque, since it has good tracking properties. The input to the PI controller is the difference between the measured and the estimated generator speed, as shown in Figure 5.2. The transfer function of the PI controller, $\text{PI}(z) = T_a(z)/\omega_{g,e}(z)$, is shown below.

$$\text{PI}(z) = K_{pw} \left(1 + \frac{T_s}{T_{iw} \cdot (z - 1)} \right) \quad \begin{bmatrix} \text{Nm} \\ \text{rad/s} \end{bmatrix} \quad (5.2)$$

where:

K_{pw} is the proportional gain [Nm/(rad/s)]

T_{iw} is the reset rate [s]

T_s is the sampling time [s]

The estimate of the aerodynamic torque is used as input to the Kalman estimator and to calculate the wind speed. The parameters of the PI controller are chosen based on the prior knowledge about the estimate. This implies that the parameters in this controller also have to be modified when the measurement of $\omega_g(k)$ is unavailable, since this increases the noise on the input to the PI controller.

Calculation of Effective Wind Speed

The final step in the wind speed estimator design procedure is to derive a conversion between the aerodynamic torque and the effective wind speed. The conversion is designed by rewriting the equation describing the aerodynamic torque shown below, based on Eq. (3.9) on Page 15. The conversion shown in this subsection is an extension of that explained in [Østergaard *et al.*, 2007], as the three pitch angles are allowed to be different.

$$T_a(t) = \frac{1}{2\omega_r(t)} \rho A \frac{1}{3} \sum_{i=1}^3 v_{r,i}^3(t) C_p(\lambda_i(t), \beta_i(t)) \quad [\text{Nm}] \quad (5.3)$$

The above equation enables the three pitch angles and the three blade effective wind speeds to be different. It is however not directly possible to calculate separate wind speeds for each blade from one aerodynamic torque, and therefore it is assumed that the effective wind speed, $v_r(t)$, is identical on each blade. Notice that this assumption implies that the tip-speed ratio is the same for all blades. The simplified equation is shown below.

$$T_a(t) = \frac{1}{2\omega_r(t)} \rho A \frac{1}{3} v_r^3(t) \sum_{i=1}^3 C_p(\lambda(t), \beta_i(t)) \quad [\text{Nm}] \quad (5.4)$$

Since $T_a(t)$ is estimated, the above equation has five unknown variables, which are $C_p(\lambda(t), \beta_1(t))$, $C_p(\lambda(t), \beta_2(t))$, $C_p(\lambda(t), \beta_3(t))$, $v_r(t)$, and $\lambda(t)$. However, the wind speed can be eliminated as shown below by exploiting the expression of the tip-speed ratio in Eq. (3.7) on Page 14.

$$\begin{aligned} T_a(t) &= \frac{\omega_r^2(t)}{6} \rho \pi R^5 \frac{C_p(\lambda(t), \beta_1(t)) + C_p(\lambda(t), \beta_2(t)) + C_p(\lambda(t), \beta_3(t))}{\lambda^3(t)} \\ \frac{6T_a(t)}{\rho \pi R^5 \omega_r^2(t)} &= \frac{C_p(\lambda(t), \beta_1(t)) + C_p(\lambda(t), \beta_2(t)) + C_p(\lambda(t), \beta_3(t))}{\lambda^3(t)} \quad [:] \end{aligned} \quad (5.5)$$

The left side of the equality sign is known when the aerodynamic torque is estimated; hence, if each value of the right side gives a unique tip-speed ratio, the expression can be reformulated as shown in Eq. (5.6), and the problem can be solved by designing a lookup table. To satisfy this requirement the function should be either monotonously decreasing or monotonously increasing. To be able to evaluate the expression and lookup only one value, the pitch angles have to be inserted and a new table must be generated. For this function to be monotonously decreasing or monotonously increasing it is required that $C_p(\lambda(t), \beta_i(t)) / \lambda^3(t)$ possesses this property, since the sum of monotonously decreasing or increasing functions is also a monotonously decreasing or increasing function. The function f shown below is determined using the C_p -table provided by kk-electronic a/s.

$$\lambda(t) = f \left(\frac{C_p(\lambda(t), \beta(t))}{\lambda^3(t)}, \beta(t) \right) \quad [:] \quad (5.6)$$

To verify if the function f possesses the desired property, it is plotted in Figure 5.3 at different pitch angles.

From the figure it is clear that the function is monotonously decreasing and hence a lookup table can be utilized to find the tip-speed ratio. When the tip-speed ratio is known, the following equation is used to establish an estimate of the wind speed:

$$v_r(t) = \frac{\omega_r(t)R}{\lambda(t)} \quad [\text{m/s}] \quad (5.7)$$

The wind speed estimator has been designed to operate in the entire operating range of the wind turbine. In the next subsection the performance of the wind speed estimator is verified.

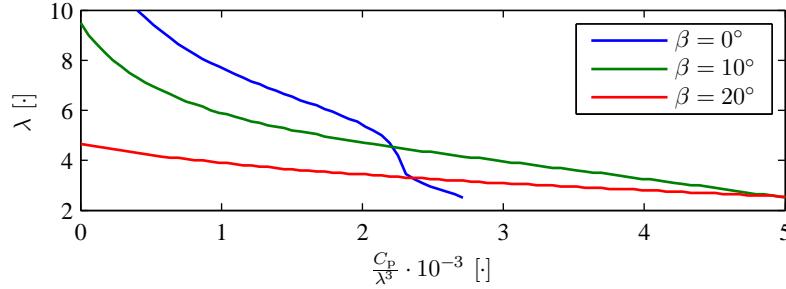


Figure 5.3: Mapping used to calculate the tip-speed ratio plotted at various pitch angles.

Verification of Reconfigurable Wind Speed Estimator

The verification of the reconfigurable wind speed estimator should examine the performance of the estimator in the entire operating range of the wind turbine, in three situations: with the measurement of $\omega_g(t)$, with the measurement of $\omega_g(t)$ using individual pitch angles for the blades, and without the measurement of $\omega_g(t)$. During the verification the already designed reference controller is applied to the wind turbine model, because the wind turbine model must be in the proximity of the expected operating trajectory.

To verify the performance of the wind speed estimator in the entire operating range of the wind turbine, a test is performed where the wind speed ranges from 5.7 m/s to 24.6 m/s; the turbulence intensity is 12%, which is considered to be sufficient. In the test using different pitch angles the reference for Blade 3 is 1° greater than the other references.

The estimated wind speed is compared to the correct wind speed, as shown in Table 5.1. Additionally, Figure 5.4 shows the wind speed sequences and the estimation errors.

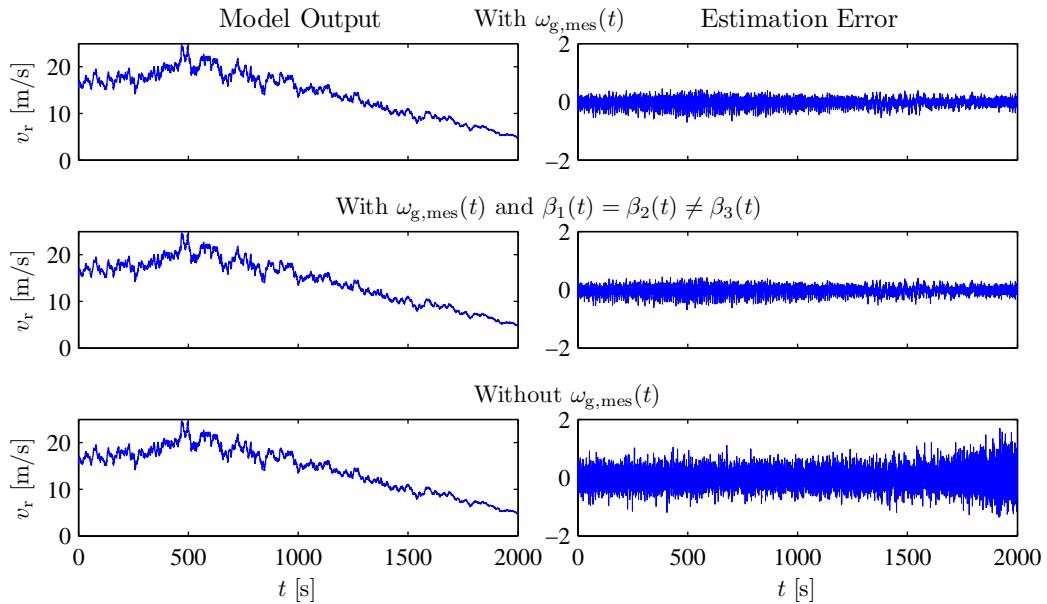


Figure 5.4: Simulation results showing the model output and the estimation error of the wind speed estimator in its three situations.

From the figure the estimated wind speed appears to follow the correct effective wind speed and it is seen that the performance of the effective wind speed estimator degrades when the generator speed measurement is not available, as expected. This is also apparent from Table 5.1, which shows that the standard deviation of the estimated wind speed increases about 2.5 times when the generator speed measurement is unavailable. The wind speed estimator is considered to perform sufficiently well in the entire region in all three situations.

Situation	Standard Deviation
With $\omega_{g,mes}(t)$	0.131 m/s
With $\omega_{g,mes}(t)$ and $\beta_1(t) = \beta_2(t) \neq \beta_3(t)$	0.131 m/s
Without $\omega_{g,mes}(t)$	0.329 m/s

Table 5.1: Standard deviations of the estimation error of the rotor effective wind speed, $v_r(t)$.

A wind speed estimator has been designed, providing an estimate of the effective wind speed suitable for being fed to the reconfigurable extended Kalman filter. The next section presents the design of the reconfigurable extended Kalman filter.

5.2 Reconfigurable Extended Kalman Filter

The purpose of this section is to explain the design of the reconfigurable extended Kalman filter and to verify its performance. Similar to the reconfigurable wind speed estimator described in the previous section, the extended Kalman filter should be able to reconfigure if the measurement of the generator speed is lost.

It is chosen to implement the state estimator as an extended Kalman filter instead of a linear filter, since the wind turbine model has varying parameters in the aerodynamic model, due to the partial derivatives of $T_a(t)$ and $F_t(t)$, and due to the faults in the system. An extended Kalman filter makes it possible to approximate the model better, since it is linearized in the current estimated state at every sample time. This property though comes at a cost, since the estimated states may diverge from the correct states due to the linearization. Furthermore, the computational time of an extended Kalman filter is increased compared to a linear Kalman filter. Though, it is assumed that the advantages of using an extended Kalman filter exceed the disadvantages.

Extended Kalman Filter Algorithm

The extended Kalman filter consists of two steps: a prediction step and an update step. In the prediction step the a priori state, a priori covariance matrix, and a priori system output are calculated from the state estimate from the previous sample and the input to the previous sample using the following equations [Grewal and Andrews, 2001, p. 180]:

$$\hat{x}(k|k-1) = f(\hat{x}(k-1|k-1), u(k-1)) \quad (5.8a)$$

$$P(k|k-1) = F(k-1)P(k-1|k-1)F^T(k-1) + Q(k-1) \quad (5.8b)$$

$$\hat{y}(k|k-1) = h(\hat{x}(k|k-1)) \quad (5.8c)$$

where:

- $f(\hat{x}(k-1|k-1), u(k-1))$ is the state transition model
- $F(k-1)$ is the Jacobian of $f(\hat{x}(k-1|k-1), u(k-1))$
- $h(\hat{x}(k|k-1))$ is the observation model
- $P(k|k-1)$ is the a priori covariance matrix
- $Q(k-1)$ is the covariance of the process noise
- $\hat{x}(k|k-1)$ is the estimated a priori state vector
- $\hat{y}(k|k-1)$ is the estimated a priori system output

When the measurements to the current time step have been fetched from the sensors, the a posteriori estimates can be calculated in the update step using the following equations:

$$\tilde{e}(k) = y(k) - \hat{y}(k|k-1) \quad (5.9a)$$

$$S(k) = H(k)P(k|k-1)H^T(k) + R(k) \quad (5.9b)$$

$$K(k) = P(k|k-1)H^T(k)S^{-1}(k) \quad (5.9c)$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\tilde{e}(k) \quad (5.9d)$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \quad (5.9e)$$

where:

- $\tilde{e}(k)$ is the innovation
- $H(k)$ is the Jacobian of $h(\hat{x}(k|k-1))$
- $K(k)$ is the Kalman gain
- $P(k|k)$ is the a posteriori covariance matrix
- $R(k)$ is the covariance of the measurement noise
- $S(k)$ is the covariance of the innovation
- $\hat{x}(k|k)$ is the estimated a posteriori state vector

Notice that the extended Kalman filter exploits a possibly non-linear model to estimate the states and system output, while the Jacobians are utilized to propagate the covariances.

Deployment of Reconfigurable Extended Kalman Filter

The linear model utilized to propagate the covariances is shown in Eq. (3.30) on Page 22, while the non-linear part of the reconfigurable extended Kalman filter directly utilizes the equations shown in Chapter 3. In addition to the system model, the reconfigurable extended Kalman filter must be provided with a measurement noise covariance matrix, an initial a posteriori covariance matrix, and a covariance matrix for the process noise. The measurement noise covariance matrix is set up from Table 3.1 on Page 23, while the initial a posteriori covariance matrix and the covariance matrix for the process noise are used to tune the reconfigurable extended Kalman filter.

In order to utilize the effective wind speed in the propagation of the covariances in the reconfigurable extended Kalman filter, it is included in the model as a random walk. This is useful in the aerodynamic model, since this includes parameters which are dependent on the wind speed.

If the generator speed sensor outputs a fixed value, the generator speed measurement is useless and the estimator should be reconfigured to handle the fault. The loss of the generator speed measurement does not affect the observability of the wind turbine model, since the rotor speed sensor is available, but this measurement is much noisier than the generator speed measurement. Furthermore, the generator speed measurement forms the basis for the wind speed estimator; hence, the estimate of the wind speed also degrades as explained in the previous section. When reconfiguring the estimator, the measurement noise associated with the generator speed measurement is increased in the measurement noise covariance matrix to a large value resulting in zero gain on the generator speed measurement. This makes it possible to maintain the same structure of the estimator in both the fault-free and faulty cases.

In the next subsection the performance of the reconfigurable extended Kalman filter is evaluated.

Verification of Reconfigurable Extended Kalman Filter

In this subsection the reconfigurable extended Kalman filter is verified both with and without using the generator speed measurement. This is accomplished by conducting long lasting simulations in the two configurations and by switching between these to verify that the transition is possible.

Verification of the Two Configurations

To verify the performance of the reconfigurable extended Kalman filter, simulations having duration of 5,000 s are conducted both with and without the measurement of $\omega_g(t)$ being available, in order to evaluate the quality of the estimates. The standard deviations of the state estimation errors are displayed in Table 5.2.

From the table it is noticeable that the estimates degrade when the generator speed measurement is unavailable. This is both a consequence of losing the measurement in the reconfigurable extended Kalman filter and having degraded performance of the wind speed estimator as shown in Table 5.1.

Verification of Reconfiguration of the Extended Kalman Filter

To complete the verification of the reconfigurable extended Kalman filter, a switch between its two configurations is simulated. The result of this simulation is shown in Figure 5.5.

From the figure it is obvious that the estimates get worse when the generator speed measurement becomes unavailable at $t = 50$ s; this complies with Table 5.2. Furthermore, the estimates do not

Variable	Standard Deviation	
	With $\omega_{g,mes}(t)$	Without $\omega_{g,mes}(t)$
T_g	11.781 Nm	11.781 Nm
x_t	0.002 m	0.005 m
v_t	2.041 mm/s	5.577 mm/s
$\beta_{1,2,3}$	(0.054°, 0.054°, 0.054°)	(0.131°, 0.130°, 0.130°)
$\dot{\beta}_{1,2,3}$	(0.404°/s, 0.401°/s, 0.404°/s)	(0.984°/s, 0.980°/s, 0.977°/s)
θ_Δ	0.014 mrad	0.056 mrad
ω_g	0.013 rad/s	0.191 rad/s
ω_r	0.598 mrad/s	1.614 mrad/s
v_r	0.108 m/s	0.217 m/s

Table 5.2: Standard deviations of the estimated states evaluated from simulations lasting 5,000 s for a mean wind speed of 16 m/s. The table both displays the case where the generator speed measurement is available and the reconfigured case where it is not.

jump when the fault is introduced; hence, the reconfigurable extended Kalman filter is considered to perform satisfactory.

In this section the design of the reconfigurable extended Kalman filter has been outlined and it has been verified by simulation. In the next section a diagnosis algorithm used to diagnose a single pitch sensor bias is explained.

5.3 Diagnosis of Single Pitch Sensor Bias

The purpose of this section is to design a diagnosis algorithm, which should be capable of diagnosing a pitch offset of a single blade. This diagnosis algorithm is used to estimate the magnitude of a biased output from a pitch sensor, which causes the actual pitch angle to be offset compared to the reference value applied to the pitch system. Throughout this section it is assumed that only one pitch sensor fails at a time, but the results are extended in the next section to allow all blades to have a pitch offset.

The section is divided into three subsections having the purposes described below.

- **Preliminary Study of Single Pitch Sensor Bias:** Provide the background necessary to understand the phenomenon which is exploited in the fault diagnosis.
- **Fault Diagnosis of Single Pitch Sensor Bias:** Describe the diagnosis algorithm used to detect, isolate, and estimate the bias of the pitch sensor.
- **Verification of the Diagnosis Algorithm for a Single Pitch Sensor Bias:** Verify the performance of the diagnosis algorithm in terms of minimum detectable bias and estimation accuracy.

Preliminary Study of Single Pitch Sensor Bias

In this subsection a series of simulation results are presented where one blade has a pitch offset. The simulation results show the changes on the tower force, tower acceleration, and aerodynamic torque. These are illustrated to show that it is possible to detect a pitch offset on a single blade based on some distinctive changes in these variables. In order to examine the influence of the different components of the wind model on these variables, a series of plots are provided. Each simulation has duration of 22 s, where the first 10 s show the operation of the wind turbine in the fault-free case. After 10 s the pitch references are changed to reflect an offset of 2° on Blade 3, corresponding to a sensor bias of -2° for Pitch Sensor 3. During the simulations fixed control signals

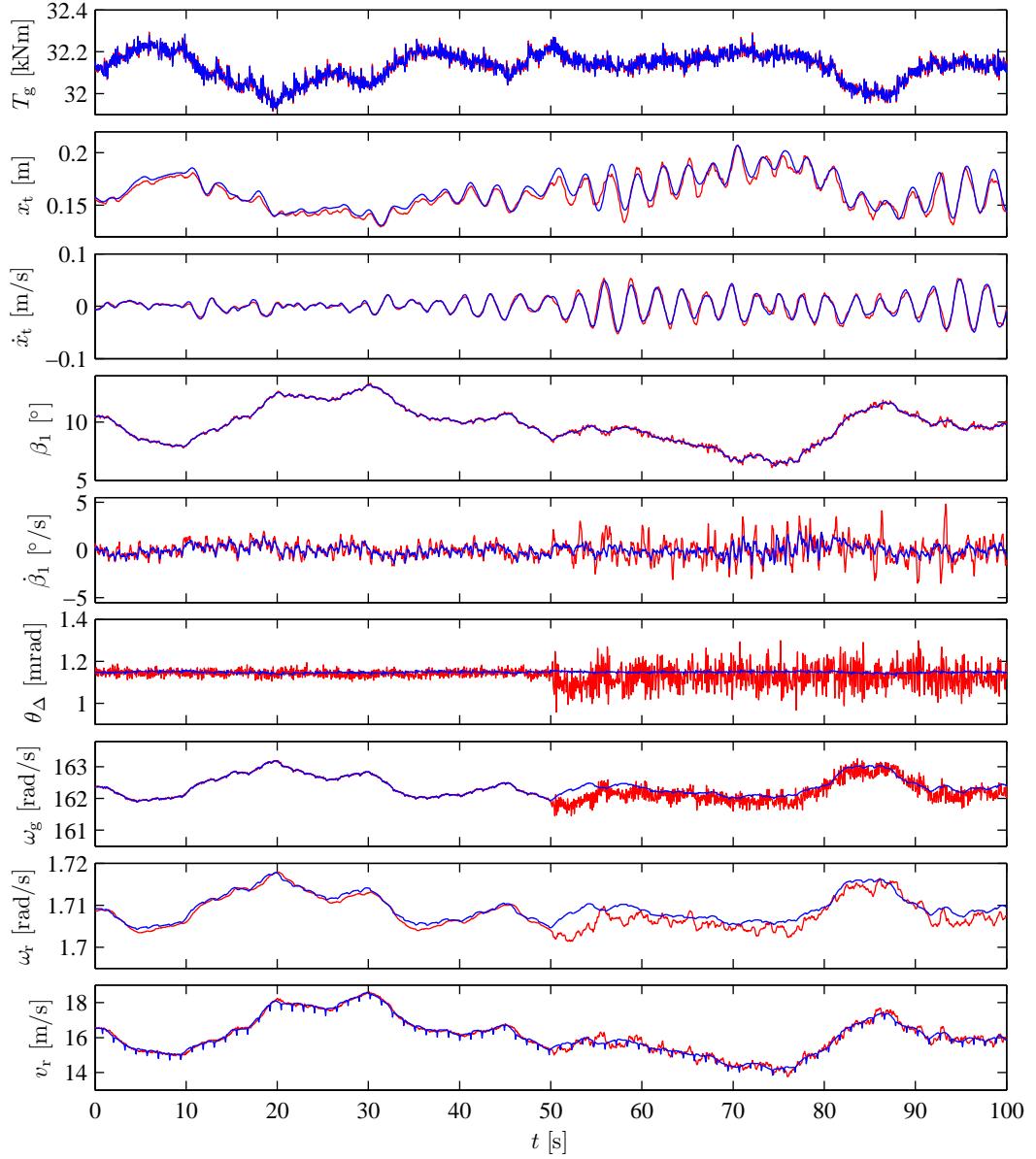


Figure 5.5: Simulation result showing the model output (blue) and the estimated states (red) using the measurement of $\omega_g(t)$ before 50 s and without the measurement of $\omega_g(t)$ after $t = 50$ s.

are utilized to clarify the distinctive changes on the tower force and aerodynamic torque, i.e. no controller is applied. Additionally, the turbulence intensity is set to 0% to eliminate the stochastic behavior of the wind signal.

Four different simulations are conducted for a mean wind speed of 16 m/s. The simulation results are shown in Figure 5.7 and Figure 5.8, where $F_{th}(t)$ and $T_a(t)$ are shown both in the three components resulting from each of the three blades and summed to the resulting force and aerodynamic torque.

Simulation 1: No wind shear or tower shadow

From the first simulation shown in the left subplots of Figure 5.7 it should be noticed that each component of the tower force originates from Eq. (3.24) on Page 18, which is repeated below for convenience. The relation between $\theta_r(t) + \varphi_i$ and the location of the blades are illustrated in Figure 5.6.

$$F_{th}(t) = \underbrace{F_{t,1}(t) \left(1 + \frac{r_t}{h} \cos(\psi_1(t)) \right)}_{F_{th,1}(t)} + \underbrace{F_{t,2}(t) \left(1 + \frac{r_t}{h} \cos(\psi_2(t)) \right)}_{F_{th,2}(t)} + \underbrace{F_{t,3}(t) \left(1 + \frac{r_t}{h} \cos(\psi_3(t)) \right)}_{F_{th,3}(t)} \quad [N] \quad (5.10)$$

with

$$\psi_1(t) = \theta_r(t) + \varphi_1, \psi_2(t) = \theta_r(t) + \varphi_2, \psi_3(t) = \theta_r(t) + \varphi_3$$

where:

- $F_{t,i}(t)$ is the thrust exerted by the wind on Blade i [N]
- $F_{th,i}(t)$ is the tower force from Blade i at hub height [N]

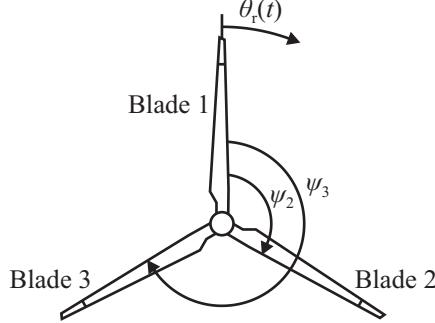


Figure 5.6: Explanation of $\theta_r(t) + \varphi_i$ in relation to the position of the blades. Notice that φ_1 is not marked, since it is 0 rad.

This equation shows that the resulting force on the tower at hub height, $F_{th}(t)$, depends on the azimuth angle of the rotor. Therefore, the location of the blades affects the tower force components. This is different from the aerodynamic torque, which is independent of the location of the blades. The sinusoids apparent on the tower force components $F_{th,1}(t)$, $F_{th,2}(t)$, and $F_{th,3}(t)$ are not visible on the resulting force acting on the tower. This is true, since the wind speed on each blade is identical; hence, introducing a signal that approximates the relation shown in Eq. (5.11).

$$0 = \cos(\theta_r(t) + \varphi_1) + \cos(\theta_r(t) + \varphi_2) + \cos(\theta_r(t) + \varphi_3) \quad (5.11)$$

given

$$\varphi_1 \triangleq 0 \text{ rad}, \varphi_2 \triangleq \frac{2\pi}{3} \text{ rad}, \varphi_3 \triangleq \frac{4\pi}{3} \text{ rad}$$

After 10 s, when one blade is pitched different from the two other blades, the amplitude of one of the cosines in Eq. (5.10) gets smaller than the others resulting in a tower force being a constant plus a cosine having a phase dependent on the blade that is pitched differently.

Simulation 2: Only wind shear

The second simulation in the right subplots of Figure 5.7 shows that wind shear has only an insignificantly small influence on the tower force, while the aerodynamic torque from each blade becomes dependent on the rotor azimuth angle. However, as the three torque components are added together the resulting influence gets small, since it approximately equals the cosine description shown in Eq. (5.11).

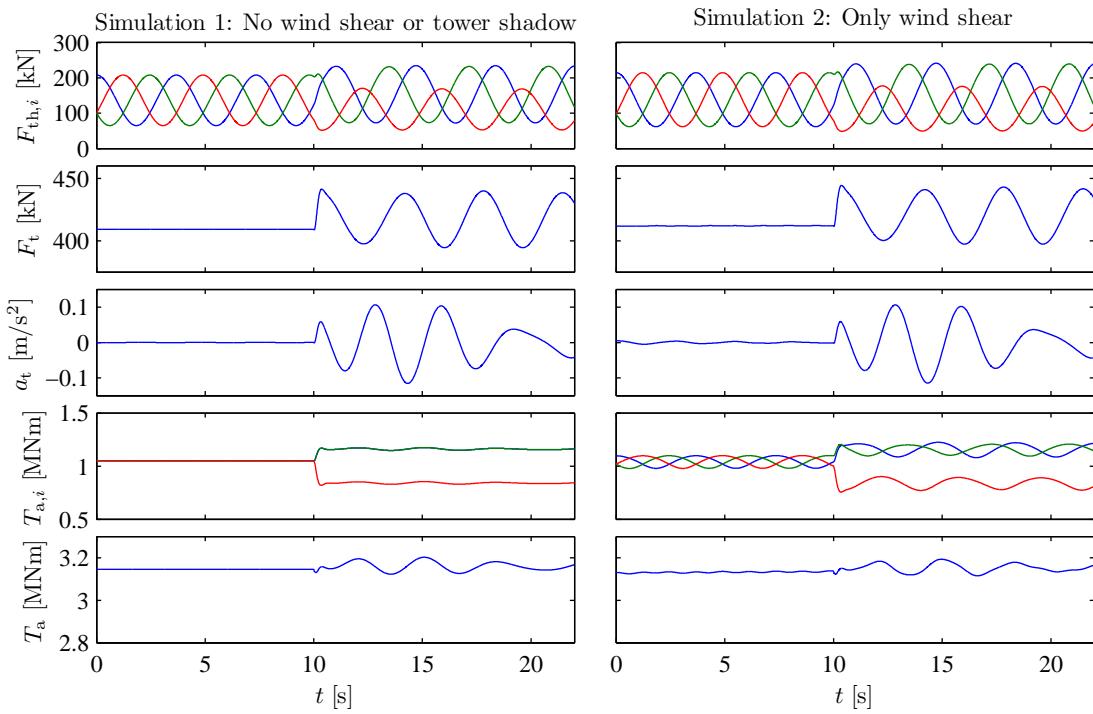


Figure 5.7: Simulation results showing the tower force, tower acceleration, and aerodynamic torque, where the turbulence intensity is set to 0% and fixed control signals are applied. At $t = 10$ s a pitch offset of 2° is introduced on Blade 3, corresponding to a pitch sensor bias of -2° .

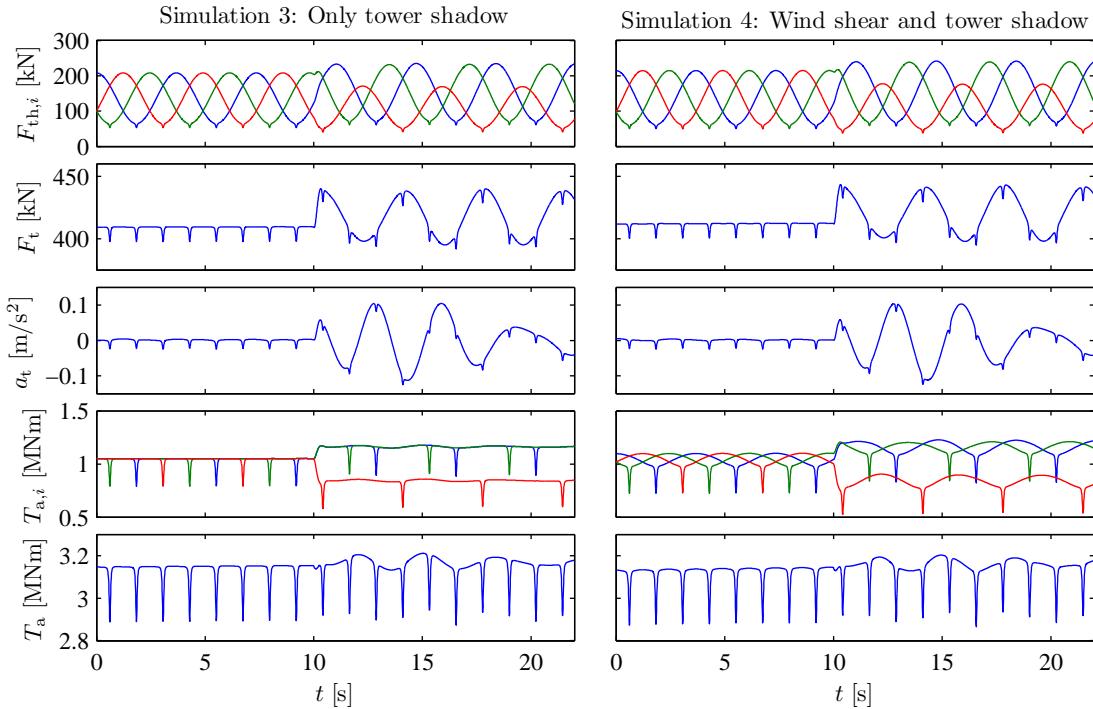


Figure 5.8: Simulation results showing the tower force, tower acceleration, and aerodynamic torque, where the turbulence intensity is set to 0% and fixed control signals are applied. At $t = 10$ s a pitch offset of 2° is introduced on Blade 3, corresponding to a pitch sensor bias of -2°

Simulation 3: Only tower shadow

From the third simulation shown in the left subplots of Figure 5.8, it is observed that tower shadow has a smaller impact on the tower force than on the aerodynamic torque, since the component $1 + r_t/h \cos(\theta_r(t) + \varphi)$ from Eq. (5.10) is multiplied on the force acting on the blade. Therefore, the tower force is at its minimum when the blade passes the tower. This implies that the cosine on the aerodynamic torque, introduced by the biased pitch angle, disappears in the tower shadow, while the cosine is still very clear on the tower force.

Simulation 4: Wind shear and tower shadow

In the fourth simulation shown in the right subplots of Figure 5.8, the wind turbine is simulated with both wind shear and tower shadow, and as concluded in Simulation 2, wind shear has almost no effect on the resulting force and torque. Therefore, the results obtained in Simulation 4 is similar to that in Simulation 3.

From the four simulations it can be concluded that, when a pitch angle has an offset, the resulting tower force can be approximated by a constant plus a cosine function with a mean value different from the nominal tower force. In contrast, almost no periodic signal is apparent on the aerodynamic torque, only a change in the mean value. From this conclusion a diagnosis algorithm is developed to estimate the bias on the pitch angle. Notice that it is necessary to use the tower model as the swaying of the tower affects the tower acceleration. This is apparent from the plots of $a_t(t)$ around $t = 20$ s, where $a_t(t)$ has another shape than $F_t(t)$.

The consequences of having different pitch angles of the blades have been determined by simulation. In the next subsections it is explained how a single pitch sensor bias is diagnosed.

Fault Diagnosis of Single Pitch Sensor Bias

The method utilized to estimate a pitch offset takes advantage of the sinusoidal signal, which appears on the tower force, when a single pitch offset exists, as shown in Eq. (5.10). A block diagram of the fault diagnosis algorithm is shown in Figure 5.9.

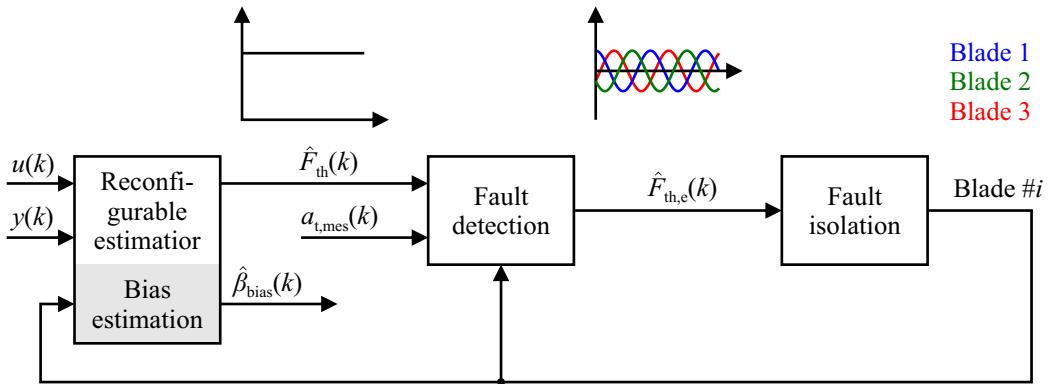


Figure 5.9: Block diagram of the algorithm utilized to determine the pitch offset of a blade. The graphs between the boxes depict the shape of the signals in these places.

In the following list the steps in the fault diagnosis algorithm are explained.

1. **Reconfigurable estimator:** The reconfigurable extended Kalman filter, described in Section 5.2, estimates the tower force from the pitch angles and the effective wind speed. The estimated tower force calculated without information about the fault does not contain the sinusoidal signal, which the true tower force possesses. Additionally, it has an offset compared to the true value.
2. **Fault detection:** The error on the estimated tower force, denoted $\hat{F}_{th,e}(k)$, is estimated based on the tower model, the estimated tower force, and the measured tower acceleration.

The estimation error is approximately zero in the fault-free case, while it contains a sinusoidal signal, originating from the pitch offset, in the fault case.

3. Fault isolation: The blade having an offset is determined from the phase of the sinusoidal signal on the estimation error, since this is related to the location of the blade.

4. Fault estimation: The reconfigurable extended Kalman filter is expanded with a bias estimator, which estimates the pitch angle of the isolated blade from the amplitude of the cosine on the estimation error of the tower force.

Detection of Single Pitch Sensor Bias

The purpose of this subsection is to describe how a pitch offset can be detected by exploiting the asymmetry in the rotor, which appears when this fault occurs. It is decided to detect the fault by using characteristics of the tower force, from which isolation is also possible.

When one blade has a pitch offset the tower force can be described as the addition of the nominal tower force, an offset, and a cosine function with a phase shift dependent on the blade having an offset; this is shown below.

$$F_{\text{th},\text{bias},i}(t) = F_{\text{th},\text{nom}}(t) + F_{\text{th},\text{offset}}(t) + F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi_i) \quad [\text{N}] \quad (5.12)$$

where:

$F_{\text{th},\text{bias},i}(t)$ is the force acting on the tower when Blade i is offset [N]

$F_{\text{th},\text{nom}}(t)$ is the force acting on the tower when the blades are equally pitched [N]

$F_{\text{th},\text{offset}}(t)$ is the offset on the force acting on the tower [N]

$F_{\text{th},\text{cos}}(t)$ is the amplitude of the cosine function [N]

φ_i is the phase shift of Blade i [rad]

The relation shown in Eq. (5.12) is exploited to isolate the blade which has a pitch offset. This is done by exploiting that φ will be different for each blade and it is known that only three possible phase shifts can occur, assuming only one fault is introduced at a time.

The tower force estimated by the extended Kalman filter explained in Section 5.2 does not contain the cosine function from Eq. (5.12) when a pitch sensor has a biased output, since the pitch angle is estimated erroneously by the extended Kalman filter. This is exploited in the detection of the fault explained in this subsection.

The general idea behind the fault detection is to feed the estimated tower force, $\hat{F}_{\text{th}}(t)$, into the tower model, to get an estimate of the tower acceleration, which can then be compared to a measurement of the tower acceleration. By assuming that the estimation error is caused by a term $F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi)$, its amplitude and phase can be estimated. The estimated amplitude should be approximately zero in the fault-free case and non-zero in case of a fault.

The fault detection is based on an extended Kalman filter due to the non-linear nature of the equations. By modifying the tower model to include $F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi)$ the following state space description can be set up:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_t}{M_t} & -\frac{B_t}{M_t} & \frac{1}{M_t} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_r^2 & 0 \end{bmatrix} \begin{bmatrix} x_t(t) \\ \dot{x}_t(t) \\ F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi) \\ -\omega_r(t) F_{\text{th},\text{cos}}(t) \sin(\theta_r(t) + \varphi) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M_t} \\ 0 \\ 0 \end{bmatrix} \hat{F}_{\text{th}}(t) \end{aligned} \quad (5.13)$$

To ease the isolation of the biased pitch measurement the term $F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi)$ is not introduced directly, but instead Eq. (5.14) is utilized to obtain Eq. (5.15).

$$\cos(\alpha \pm \beta) = \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta) \quad (5.14)$$

$$F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi) = F_{\text{th},\text{cos}}(t) \cos(\varphi) \cos(\theta_r(t)) - F_{\text{th},\text{cos}}(t) \sin(\varphi) \sin(\theta_r(t)) \quad [\text{N}] \quad (5.15)$$

From Eq. (5.15) it is clear that $F_{\text{th},\text{cos}}(t) \cos(\theta_r(t) + \varphi)$ can be rewritten as a function of $\cos(\theta_r(t))$ and $\sin(\theta_r(t))$, which are only dependent on the rotor azimuth angle, and the terms $F_{\text{th},\text{cos}}(t) \cos(\varphi)$

and $F_{\text{th},\cos}(t) \sin(\varphi)$, which are dependent on the magnitude of the bias and the location of the biased blade. This trick is solely done to make the isolation easier.

The rearrangement made in Eq. (5.15) replaces the third and fourth state in Eq. (5.13) with two states equal to $F_{\text{th},\cos}(t) \cos(\varphi)$ and $F_{\text{th},\cos}(t) \sin(\varphi)$ in the extended Kalman filter. These are utilized in the fault detection together with two states equal to $\cos(\theta_r(t))$ and $-\omega_r(t) \sin(\theta_r(t))$, which are known due to the knowledge of $\theta_r(t)$. This expands the state vector to contain six states:

$$x(t) = \begin{bmatrix} x_t(t) \\ \dot{x}_t(t) \\ F_{\text{th},\cos}(t) \cos(\varphi) \\ F_{\text{th},\cos}(t) \sin(\varphi) \\ \cos(\theta_r(t)) \\ -\omega_r(t) \sin(\theta_r(t)) \end{bmatrix} \quad (5.16)$$

The non-linear function $f(x, u)$ used in the extended Kalman filter is shown below.

$$\begin{aligned} \dot{x}(t) &= f(x, u) \\ \dot{x}(t) &= \begin{bmatrix} \dot{x}_t(t) \\ -\frac{K_t}{M_t}x_t(t) - \frac{B_t}{M_t}\dot{x}_t(t) + \frac{F_{\text{th},\cos}(t) \cos(\varphi) \cos(\theta_r(t))}{M_t} - \frac{F_{\text{th},\cos}(t) \sin(\varphi) \sin(\theta_r(t))}{M_t} \\ 0 \\ 0 \\ -\omega_r(t) \sin(\theta_r(t)) \\ -\omega_r^2(t) \cos(\theta_r(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M_t}\hat{F}_{\text{th}}(t) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (5.17)$$

From $f(x, u)$ the Jacobian, which is used to propagate the covariances in the filter, can be calculated as shown below. Notice the states are renamed to x_i to shorten the notation.

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K_t}{M_t} & -\frac{B_t}{M_t} & \frac{x_5(t)}{M_t} & \frac{x_6(t)}{\omega_r(t)M_t} & \frac{x_3(t)}{M_t} & \frac{x_4(t)}{M_t\omega_r(t)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\omega_r^2(t) & 0 \end{bmatrix} \quad (5.18)$$

The extended Kalman filter used for the fault detection has been set up, and a biased pitch measurement is assumed to happen when $F_{\text{th},\cos}(t) \cos(\varphi)$ and $F_{\text{th},\cos}(t) \sin(\varphi)$ get above a preset threshold. When the fault has been detected the fault isolation algorithm explained next isolates the biased pitch sensor.

Isolation of Single Pitch Sensor Bias

To isolate the blade which has a pitch offset it is necessary to utilize the knowledge of the three possible phase shifts deduced from Figure 5.6 on Page 54: $\varphi \in \{0, 2\pi/3, 4\pi/3\}$ rad. Table 5.3 is set up to show how the blade having the pitch offset is isolated based on the estimates of $F_{\text{th},\cos}(t) \cos(\varphi)$ and $F_{\text{th},\cos}(t) \sin(\varphi)$ as φ changes. The table only displays constraints for $F_{\text{th},\cos}(t) > 0$, but these are easily derived for $F_{\text{th},\cos}(t) < 0$ by replacing ' $<$ ' with ' $>$ ' and vice versa.

From Table 5.3, $F_{\text{th},\cos}(t) \cos(\varphi)$, and $F_{\text{th},\cos}(t) \sin(\varphi)$ it is possible to isolate the blade having a pitch offset. This information should be sent to the extended Kalman filter, which then should be reconfigured to estimate the magnitude of the bias on the identified blade. This procedure is explained the next subsection.

Estimation of Single Pitch Sensor Bias

To estimate the bias of the pitch angle some extra reconfiguration possibilities are added to the extended Kalman filter in Section 5.2 in order to fulfill this task.

Condition	$F_{\text{th},\cos}(t) \cos(\varphi)$	$F_{\text{th},\cos}(t) \sin(\varphi)$	Blade
$F_{\text{th},\cos}(t) > 0 \wedge \varphi = 0 \text{ rad}$	$= F_{\text{th},\cos}(t)$	$= 0$	1
$F_{\text{th},\cos}(t) > 0 \wedge \varphi \in]0, \pi/2[\text{ rad}$	> 0	> 0	-
$F_{\text{th},\cos}(t) > 0 \wedge \varphi = \pi/2 \text{ rad}$	$= 0$	$= F_{\text{th},\cos}(t)$	-
$F_{\text{th},\cos}(t) > 0 \wedge \varphi \in]\pi/2, \pi[\text{ rad}$	< 0	> 0	2
$F_{\text{th},\cos}(t) > 0 \wedge \varphi = \pi \text{ rad}$	$= -F_{\text{th},\cos}(t)$	$= 0$	-
$F_{\text{th},\cos}(t) > 0 \wedge \varphi \in]\pi, 3\pi/2[\text{ rad}$	< 0	< 0	3
$F_{\text{th},\cos}(t) > 0 \wedge \varphi = 3\pi/2 \text{ rad}$	$= 0$	$= -F_{\text{th},\cos}(t)$	-
$F_{\text{th},\cos}(t) > 0 \wedge \varphi \in]3\pi/2, 2\pi[\text{ rad}$	> 0	< 0	-

Table 5.3: Value space of $x_3(t) = F_{\text{th},\cos}(t) \cos(\varphi)$ and $x_4(t) = F_{\text{th},\cos}(t) \sin(\varphi)$ and the blade associated with each interval.

A biased pitch measurement affects both the measurement equation and the closed-loop pitch actuator model, since the internal controller in the pitch actuator utilizes the erroneous measurement too. Hence, the model of the pitch system shown in Eq. (3.29) on Page 21 should be modified by adding $\beta_{\text{bias}}(t)$ to its description as shown below.

$$\ddot{\beta}(t) = -2\zeta\omega_n\dot{\beta}(t) - \omega_n^2(\beta(t) + \beta_{\text{bias}}(t)) + \omega_n^2\beta_{\text{ref}}(t - t_d) \quad [\text{°}/\text{s}^2] \quad (5.19)$$

where:

$\beta_{\text{bias}}(t)$ is the bias on the pitch angle measurement [°]

Furthermore, the measurement equation for the pitch angle should be modified as shown below.

$$\beta_{\text{mes}}(k) = \beta(k) + \beta_{\text{bias}}(k) + v_\beta(k) \quad [\text{°}] \quad (5.20)$$

To implement the changes in the system description three states are added to the state vector of the reconfigurable extended Kalman filter, representing the bias of each pitch angle sensor. The changed system and output equations shown above should be permanently implemented in $f(\hat{x}(k-1|k-1), u(k-1))$, and $h(\hat{x}(k|k-1))$, but not in $F(k)$ and $H(k)$, which are used to propagate covariances as explained in Section 5.2. These matrices should only be modified when the detection and isolation algorithm have identified a fault, because this makes it possible to keep the bias estimates at constant values when either no fault is detected or when the detected fault has already been estimated.

Notice that a biased pitch angle measurement also affects the wind speed estimator, since the pitch angle is used in its lookup tables. This makes it necessary to feed the bias estimate back to the wind speed estimator. This of course requires that the bias estimate is much slower than the dynamics of wind speed estimator.

This finalizes the design of the diagnosis algorithm for a single pitch sensor bias. In the next subsection the algorithm is verified.

Verification of Single Pitch Sensor Bias Diagnosis Algorithm

The purpose of this subsection is to verify the performance of the single pitch angle bias detection and estimation. The verification of the detection is performed by determining the minimum detectable pitch sensor bias, while the estimation is verified by evaluating the accuracy of the estimated bias.

Determination of Minimum Detectable Single Pitch Bias

The verification of the fault detection algorithm is performed by conducting a series of simulations where different pitch sensor biases are applied on Blade 1. The size of the pitch sensor bias is increased in steps of 0.05° between 0° and 0.25° to determine the smallest detectable bias. To find the smallest detectable bias, the mean values and standard deviations of φ and $F_{\text{th},\cos}(t) \cos(\varphi)$ are determined, since these are used in the detection and isolation. This is done by conducting

100 simulations with duration 2,000 s for each pitch sensor bias. The results of the simulations are shown in Figure 5.10, where the outcomes are assumed to be Gaussian distributed.

To detect and isolate a biased pitch sensor output, the following two conditions must be satisfied:

1. Detection: The estimated value of $F_{\text{th},\cos} \cos(\varphi)$ must be greater than the value of $F_{\text{th},\cos} \cos(\varphi)$ when no bias exists.
2. Isolation: The estimate of φ must deviate less than $\frac{\pi}{6}$ rad from its real value.

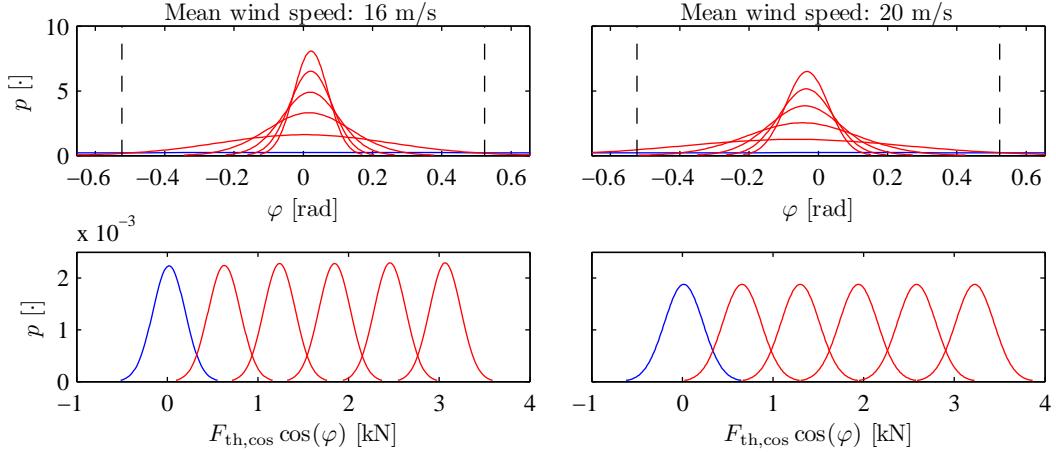


Figure 5.10: *Distributions of φ and $F_{\text{th},\cos}(t) \cos(\varphi)$, when applying biases on the pitch sensor associated with Blade 1 for mean wind speeds of 16 m/s and 20 m/s. The blue lines illustrate the fault-free cases, while the red lines illustrate the fault situations, where pitch sensor biases between 0° and 0.25° are applied in steps of 0.05° . Additionally, the dashed black lines indicate the interval which φ should be in to perform a correct identification.*

From the figure it is seen that it is possible to detect and isolate the pitch sensor biases greater than 0.1° . This bias can be detected, since the distribution of the phase of biases greater than 0.1° is inside the bounds marked with dashed black lines in the upper subplots. Furthermore, the distributions of $F_{\text{th},\cos}(t) \cos(\varphi)$ in the fault-free case and faulty case do not overlap for biases greater than 0.1° . This satisfies the requirement of a minimum detectable bias equal to 0.25° defined in Section 4.6. It is possible to improve the accuracy of the detection and isolation by averaging over a longer time period than the currently used 2,000 s. This would make the distributions of both φ and $F_{\text{th},\cos}(t) \cos(\varphi)$ narrower; hence, smaller biases could be detected.

Estimation of Single Pitch Sensor Bias

The verification of the estimation of a single pitch sensor bias is conducted by operating the wind turbine for mean wind speeds of 16 m/s and 20 m/s and introducing a pitch bias of 1° linearly between $t = 200$ s and $t = 2,000$ s on the pitch sensor associated with Blade 3. The simulation results of the Monte Carlo simulations are shown in Figure 5.11. The detection of the fault has been introduced manually to decrease the simulation time. This manually performed detection introduces the jump in the estimated pitch bias at $t = 250$ s.

The simulations show that the pitch angle estimate converges towards the correct bias in each case, as the mean values equal 1.04° and 1.02° for mean wind speeds of 16 m/s and 20 m/s, respectively. Furthermore, the standard deviations equal 0.03° for both wind speeds. Therefore, the bias estimation is assumed to perform satisfactory since the estimated bias is very close to the correct bias of 1° .

In this section a method for diagnosing a single pitch sensor bias has been presented. The method relies on the asymmetry in the rotor plane, which is introduced when a pitch offset is introduced. The next section extends this method to apply for multiple pitch sensor biases.

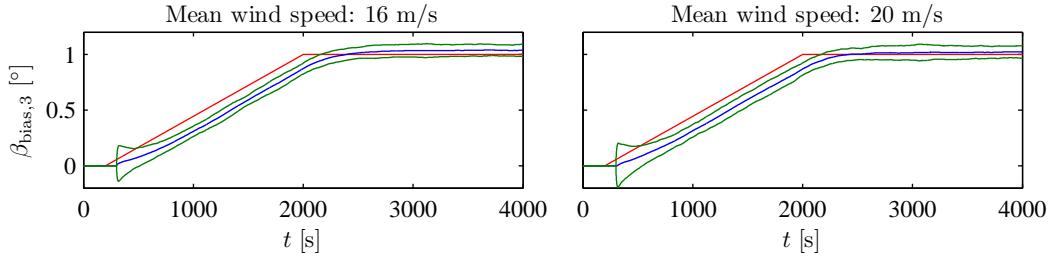


Figure 5.11: Simulation results showing the pitch sensor bias (red), the mean value of the estimated bias (blue), and the mean values plus and minus two times the standard deviations of the pitch bias estimates (green) for mean wind speeds of 16 m/s and 20 m/s.

5.4 Diagnosis of Multiple Pitch Sensor Biases

In the previous section a method is presented for estimating a single pitch sensor bias, when the two remaining pitch sensors are assumed to be unbiased. This method is based on determining the phase of the sinusoidal signal introduced on the tower force after injection of the bias fault. Since the method is based on imbalance in the rotor plane, it is possible to extend the method to a case of multiple simultaneous bias faults on the pitch sensors. This extension is described in this section.

From Eq. (5.10) on Page 54, which is repeated below for convenience, it is clear that only the differences between the individual forces acting on the blades affect the phase of the resulting cosine function on $F_{\text{th}}(t)$, not the absolute values $F_{\text{t},1}(t)$, $F_{\text{t},2}(t)$, and $F_{\text{t},3}(t)$.

$$\begin{aligned} F_{\text{th}}(t) &= F_{\text{t},1}(t) \left(1 + \frac{r_t}{h} \cos(\psi_1(t))\right) + F_{\text{t},2}(t) \left(1 + \frac{r_t}{h} \cos(\psi_2(t))\right) + F_{\text{t},3}(t) \left(1 + \frac{r_t}{h} \cos(\psi_3(t))\right) \\ &= \bar{F}_{\text{th}}(t) + F_{\text{th},\cos}(t) \cos(\theta_r(t) + \varphi(t)) \quad [\text{N}] \end{aligned} \quad (5.21)$$

where:

$\bar{F}_{\text{th}}(t)$ is the mean value of $F_{\text{th}}(t)$ [N]

$F_{\text{th},\cos}(t)$ is magnitude of the resulting cosine function on $F_{\text{th}}(t)$ [N]

$\varphi(t)$ is the phase shift of the resulting cosine function on $F_{\text{th}}(t)$ [rad]

From Eq. (5.21) it is clear that by evaluating $F_{\text{th},\cos}(t) \cos(\theta_r(t) + \varphi(t))$ it is only possible to diagnose the mutual offsets between the pitch angles of the blades, since a common bias on the sensors does not create imbalance in the rotor plane. Therefore, another method should be utilized to remove a common offset. A couple of ideas for removing this are shown below.

- In partial load operation the wind turbine is expected to operate at the maximum of the C_p -surface, where it is flat. Inserting an auxiliary signal on the pitch reference would make it possible to estimate whether or not the wind turbine is operated at this expected optimum point.
- In full load operation the slope of the C_p -surface could be estimated and compared with the expected slope.

The estimation of the common pitch offset is not implemented, since it is considered to be improper use of the simplified static aerodynamic description utilized in the model to estimate a slope of a surface, which only applies in steady-state, as explained in Section 3.8 on Page 21.

The existence of a common pitch offset does not introduce cyclical loads on the rotor, but the controller cannot be optimized to the correct operating conditions of the wind turbine.

Detection of Multiple Pitch Sensor Biases

To detect that multiple pitch faults exist, it should be detected that $F_{\text{th},\cos}(t)$ in Eq. (5.21) becomes non-zero. This is identical to the fault detection designed for a single biased pitch sensor in the previous section, and is therefore not described further.

Isolation of Multiple Pitch Sensor Biases

The purpose of the fault isolation is to categorize the blades according to their pitch biases, i.e. determine the blade having the largest, median, and smallest bias. By doing so, it is possible to determine a strategy for achieving a common offset of the blades.

To isolate the faults it is necessary to estimate the phase shift $\varphi(t)$ of the tower force shown in Eq. (5.21), since Table 5.3 on Page 59 is not sufficient. This can be obtained by rewriting $F_{\text{th},\cos}(t) \cos(\theta_r(t) + \varphi(t))$ into a cosine and a sine function, as formulated in Eq. (5.15) on Page 57 and shown in Eq. (5.22).

$$\begin{aligned} F_{\text{th}}(t) &= \bar{F}_{\text{th}}(t) + \underbrace{(F_{t,2}(t) + F_{t,3}(t) - 2F_{t,1}(t)) \frac{r_t}{h} \cos(\varphi_2) \cos(\theta_r(t))}_{F_{\text{th},\cos}(t) \cos(\varphi(t))} \\ &\quad - \underbrace{(F_{t,2}(t) - F_{t,3}(t)) \frac{r_t}{h} \sin(\varphi_2) \sin(\theta_r(t))}_{F_{\text{th},\cos}(t) \sin(\varphi(t))} \\ F_{\text{th}}(t) &= \bar{F}_{\text{th}}(t) + F_{\text{th},\cos}(t) \cos(\varphi(t)) \cos(\theta_r(t)) - F_{\text{th},\cos}(t) \sin(\varphi(t)) \sin(\theta_r(t)) \quad [N] \end{aligned} \quad (5.22)$$

From this equation it is possible to calculate the resulting phase shift, $\varphi(t)$, for arbitrary biases. To provide an easily comprehensible overview of the possible phase shifts, a unit circle is shown in Figure 5.12 with the possible combinations. The phase shift can be followed around the unit circle, and is associated with the mutual relationship between the forces at each blade to every phase shift. Notice that offsets can both be positive and negative resulting in a phase shift of π rad on φ .

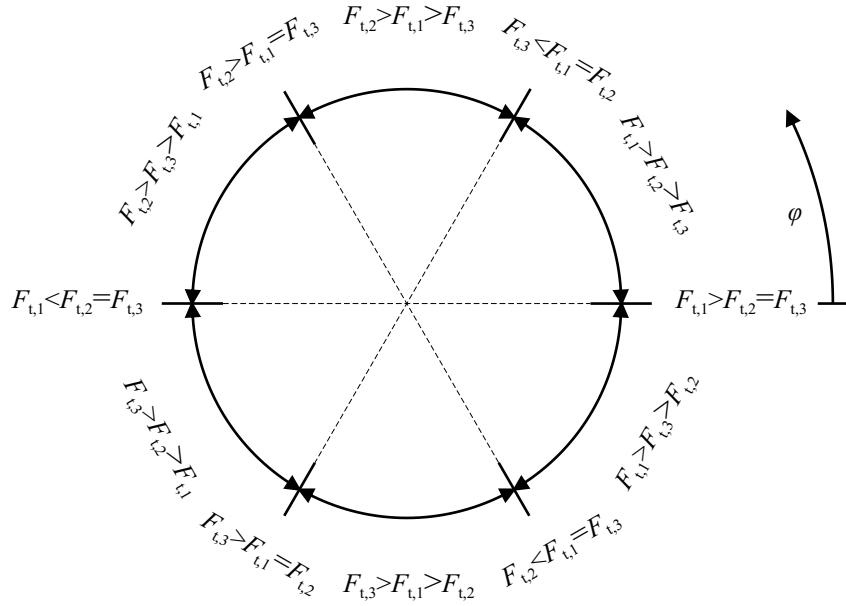


Figure 5.12: Illustration of a unit circle with the phase shift, φ , along its periphery. Outside the unit circle, the relation between the forces acting on the blades are associated to the phase shift experienced on the resulting tower force.

To isolate the fault, i.e. determine the current fault situation, the phase has to be estimated. This is possible using the estimator shown in Eq. (5.17) on Page 58, which is designed for isolation of one pitch sensor bias. The available estimates from this estimator are $F_{\text{th},\cos}(t) \cos(\varphi(t))$ and $F_{\text{th},\cos}(t) \sin(\varphi(t))$. From these two estimates it is possible to calculate $\varphi(t)$, which then can be utilized in combination with Figure 5.12 to determine the mutual relationship between the biases

on the pitch sensors. The equation for calculating $\varphi(t)$ from the two estimates is shown in Eq. (5.23).

$$\varphi(t) = \begin{cases} \tan^{-1} \left(\frac{F_{\text{th},\cos}(t) \sin(\varphi(t))}{F_{\text{th},\cos}(t) \cos(\varphi(t))} \right) & \text{for } F_{\text{th},\cos}(t) \cos(\varphi(t)) > 0 \\ \tan^{-1} \left(\frac{F_{\text{th},\cos}(t) \sin(\varphi(t))}{F_{\text{th},\cos}(t) \cos(\varphi(t))} \right) + \pi & \text{for } F_{\text{th},\cos}(t) \cos(\varphi(t)) < 0 \\ \text{sign}(F_{\text{th},\cos}(t) \sin(\varphi(t))) \frac{\pi}{2} & \text{for } F_{\text{th},\cos}(t) \cos(\varphi(t)) = 0 \end{cases} \quad (5.23)$$

Estimation of Multiple Pitch Sensor Biases

To estimate the pitch sensor biases, the fault isolation must provide the reconfigurable extended Kalman estimator with information about the two pitch sensor biases it has to estimate. When the biased sensors are isolated the estimation is identical to the procedure explained in Section 5.3. The difference is that there are two biases, but as $\varphi(t)$ can be estimated, these biases can be calculated separately. This is not further discussed in this section due to the similarities with the case of a single pitch sensor fault.

Verification of Multiple Pitch Sensor Biases Diagnosis Algorithm

In this section the fault estimation is verified by evaluating its ability to estimate mutual pitch sensor biases. This is accomplished by conducting Monte Carlo simulations with two pitch sensor biases.

To stress that the accommodation is not always able to remove the common pitch offset, since only the mutual biases can be estimated, the biases injected in the simulations are chosen to have a non-zero median. For this verification, 100 simulations are conducted with duration 4,000 s for mean wind speeds of both 16 m/s and 20 m/s. The biases are linearly introduced on two pitch sensors between $t = 200$ s and $t = 2,000$ s. The magnitudes of the introduced biases are 3° on Blade 2 and 2° on Blade 3.

To provide an overview of the expected outcome of the simulations, Table 5.4 provides the expected values of the pitch sensor bias estimates and the expected values of the resulting pitch angle offsets after accommodation, in the considered simulation example.

Blade	Pitch Sensor Bias	Pitch Sensor Bias Estimate	Pitch Angle Offset
1	0°	-2°	-2°
2	3°	1°	-2°
3	2°	0°	-2°

Table 5.4: *Expected outcome of the Monte Carlo simulations. In the last column, the expected pitch angle offsets after accommodation are shown. However, the accommodation is first presented in Section 6.2.*

It can be seen from Table 5.4 that the biases should be estimated on the pitch sensors associated with Blade 1 and Blade 2, since these sensors have the extreme values of the pitch sensor biases. Consequently, the blades are expected to obtain a common offset equal to minus the median of the pitch sensor biases, i.e. -2° when accommodated.

The results of the simulations are shown in Figure 5.13, where the estimated biases are evaluated by comparing them to the expected biases. From the figure it is clear that the pitch sensor biases are estimated as expected and that a collective bias exists, since the method is only able to estimate mutual biases. The mean values of the estimates equals -2.01° and 0.98° , respectively, while the standard deviations of the bias estimates equal 0.03° , which is considered sufficiently accurate.

In this section a method for diagnosing multiple pitch sensor biases has been presented. The method is an extension to the method for diagnosing a single pitch sensor bias, but is restricted to estimate only the mutual relationship between the biases on the blades. The accommodation of multiple bias faults is explained in Section 6.2. In the next section a method is presented for estimating the dynamics of the pitch system.

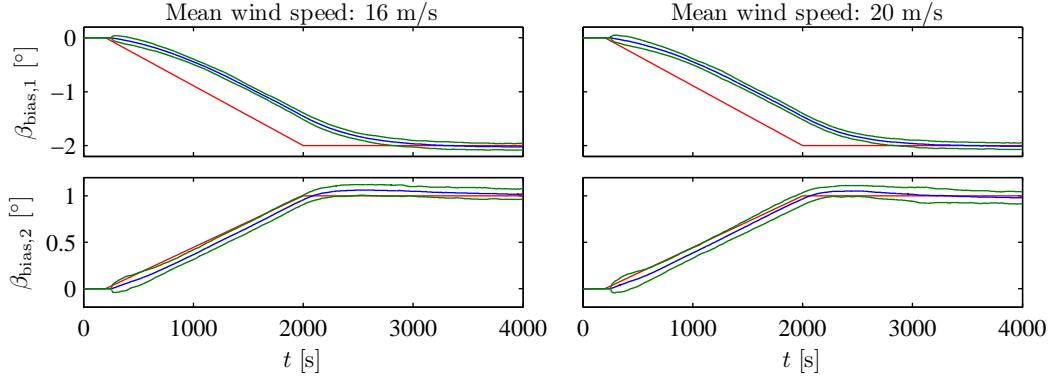


Figure 5.13: Results of Monte Carlo simulations, where the biases are introduced between $t = 200$ s and $t = 2,000$ s. This introduces a bias with respect to the median of the biases, illustrated with the red line. The blue lines show the mean values of the estimated biases, while the green lines show the mean values plus and minus two times the standard deviations of the estimated biases.

5.5 Estimation of Pitch System Dynamics

The purpose of this section is to present a method for determining the dynamics of the pitch system. This is done by estimating the parameters of the pitch system model.

Several parameter estimation methods exist, but it is decided to select a method that allows an extended Kalman filter to be used, since it enables the parameter estimation to be incorporated in the structure presented in Figure 5.1 on Page 45. One method for estimating the parameters is to augment the state vector of the model with the parameters to be estimated. A similar result can be obtained using a multiple-model framework, as the one described in [Hallouzi, 2008, pp. 23-49], which is selected due to its flexible structure and its intuitive modeling of faults.

The multiple-model framework requires that local models are generated at certain operating conditions. The correct model is then found as a convex combination of these models.

In the following bullet list the content of the section is summarized.

- **Limit Value Based Method for Multiple-Model Estimation:** Describe the parameter variations in a single pitch actuator model using the limit value based method.
- **Estimation Based on all Actuator Models:** Enlarge the obtained system to cover the models of all three pitch actuators.
- **Multiple-Model Estimation using an Extended Kalman Filter:** Present the algorithm for the multiple-model estimation using an extended Kalman filter.
- **Constraint Implementation:** Include constraints in the algorithm to restrict the estimates to remain within the generated model set.
- **Verification of Estimation of Pitch System Dynamics:** Verify the performance of the multiple-model estimation.

It should be noted that the parameter estimation exploits the pitch sensor measurements from all three pitch actuators to determine the dynamics of the pitch system. The pitch actuators are expected to have identical parameters, since they share the same oil supply. However, to simplify the notation in the first part of the explanation, only a single pitch actuator is considered. Later, the system is extended to cover all three actuators.

The estimator design is initiated by writing up the state space representation of a pitch actuator. The model is found in Eq. (3.29) on Page 21, and for convenience it is written below in a state

space form including its varying parameters.

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ \begin{bmatrix} \dot{\beta}(t) \\ \ddot{\beta}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2(t) & -2\zeta(t)\omega_n(t) \end{bmatrix} \begin{bmatrix} \beta(t) \\ \dot{\beta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2(t) \end{bmatrix} \beta_{\text{ref}}(t - t_d)\end{aligned}\quad (5.24a)$$

$$\begin{aligned}y(t) &= Cx(t) + v(t) \\ \beta_{\text{mes}}(t) &= [1 \ 0] \begin{bmatrix} \beta(t) \\ \dot{\beta}(t) \end{bmatrix} + v_\beta(t) \quad [\circ]\end{aligned}\quad (5.24b)$$

The dynamics of the pitch system is characterized by the parameters ω_n and ζ . The parameter variations are specified in Table 4.8 on Page 40. The changing dynamics results from pump wear, hydraulic leakage, or high air content in the hydraulic oil.

The next subsection describes how to generate a model set which captures the possible parameter variations in Eq. (5.24a). This involves choosing a model structure for the estimation problem.

Limit Value Based Method for Multiple-Model Estimation

In [Hallouzi, 2008, pp. 51-78] different approaches are described for generating a model set used in a multiple-model estimation. The model set should make the infinite set of models, originating from the continuous parameter space, be approximated by a convex combination of a few models.

A basic method is to divide the parameter space into a grid, and for each grid point assign a corresponding local model. The different local models can be automatically generated by a computer, but results in a computationally demanding algorithm. Instead, other approaches exist, which start by randomly generating models at different operating conditions. The idea of these methods is then to approximate the large set of sampled models by a few local models. However, these methods are better suited for systems which are much larger than the considered system. A third method presented in [Hallouzi, 2008, p. 63] is based directly on the limit values of the varying model parameters. In this particular case, these limits are available from the fault specification in Table 4.8 on Page 40. Furthermore, the model set can be intuitively designed and there exists a direct relation between the model weights and the parameters. Therefore, it is decided to select this method.

The limit value based method has its starting point in the limit values of the varying entries of the system matrices. The entry in the i -th row and j -th column of the system matrix can be expressed as a convex combination of the limit values of the parameter; a_{ij}^- and a_{ij}^+ :

$$\begin{aligned}a_{ij}(t) &= \mu_{ij}^a a_{ij}^- + (1 - \mu_{ij}^a(t)) a_{ij}^+ \\ &= a_{ij}^+ + \underbrace{\mu_{ij}^a(t)(a_{ij}^- - a_{ij}^+)}_{a_{ij}^\Delta} \quad \text{where} \quad \mu_{ij}^a \in [0, 1]\end{aligned}\quad (5.25)$$

where:

a_{ij}^- and a_{ij}^+ are the minimum and maximum values of the (i,j) -th entry of the system matrix
 $\mu_{ij}^a(t)$ is the weight on the (i,j) -th entry of the system matrix

Let A^+ be defined as the matrix in which all entries equal their maximal values and let A_{ij}^Δ be defined as the matrix in which all entries are 0, except the (i,j) -th element, which must equal a_{ij}^Δ :

$$A^+ = \{A : a_{ij} = a_{ij}^+ \ \forall i, j \in \{1, 2, \dots, n\}\} \quad (5.26)$$

$$A_{ij}^\Delta = \left\{ A : a_{ij} = a_{ij}^\Delta \wedge a_{i\tilde{j}} = 0 \ \forall (\tilde{i}, \tilde{j}) \neq (i, j) \right\} \quad (5.27)$$

By using a similar notation for the entries in the input matrix, output matrix, and direct transmis-

sion term, the following general system emerges:

$$\begin{aligned}\dot{x}(t) &= A_\mu(t)x(t) + B_\mu(t)u(t) \\ \dot{x}(t) &= \left(A^+ + \sum_{i,j} A_{ij}^\Delta \mu_{ij}^a(t) \right) x(t) + \left(B^+ + \sum_{i,j} B_{ij}^\Delta \mu_{ij}^b(t) \right) u(t)\end{aligned}\quad (5.28a)$$

$$\begin{aligned}y(t) &= C_\mu(t)x(t) + D_\mu(t)u(t) \\ y(t) &= \left(C^+ + \sum_{i,j} C_{ij}^\Delta \mu_{ij}^c(t) \right) x(t) + \left(D^+ + \sum_{i,j} D_{ij}^\Delta \mu_{ij}^d(t) \right) u(t)\end{aligned}\quad (5.28b)$$

where:

$A_\mu(t), B_\mu(t), C_\mu(t), D_\mu(t)$ are the matrices of an LPV system
 $\mu_{ij}^a(t), \mu_{ij}^b(t), \mu_{ij}^c(t), \mu_{ij}^d(t)$ are the weights on the entries of A, B, C, D , respectively

According to Eq. (5.24) the pitch actuator model has only parameter-varying entries in the system and input matrices, resulting in Eq. (5.29).

$$\begin{aligned}A^+ &= \begin{bmatrix} 0 & 1 \\ -(\omega_n^-)^2 & -2\zeta^-\omega_n^- \end{bmatrix} & A_{21}^\Delta &= \begin{bmatrix} 0 & 0 \\ -(\omega_n^+)^2 + (\omega_n^-)^2 & 0 \end{bmatrix} & A_{22}^\Delta &= \begin{bmatrix} 0 & 0 \\ 0 & -2\zeta^+\omega_n^+ + 2\zeta^-\omega_n^- \end{bmatrix} \\ B^+ &= \begin{bmatrix} 0 \\ (\omega_n^+)^2 \end{bmatrix} & B_{21}^\Delta &= \begin{bmatrix} 0 \\ (\omega_n^-)^2 - (\omega_n^+)^2 \end{bmatrix}\end{aligned}\quad (5.29)$$

From the above description three weights generally exist. However, the number of weights can be reduced to only two by exploiting that $a_{21} = -b_{21}$ in Eq. (5.24a). This reduces the LPV system in Eq. (5.28) into Eq. (5.30) with $\mu_j \in [0, 1]$ for $j = 1, 2$.

$$\dot{x}(t) = [A^+ + A_{21}^\Delta \mu_1(t) + A_{22}^\Delta \mu_2(t)] x(t) + [B^+ + B_{21}^\Delta(1 - \mu_1(t))] u(t) \quad (5.30a)$$

$$y(t) = Cx(t) \quad (5.30b)$$

In the next subsection the system description is extended to include all three pitch actuators, to exploit the information of each actuator when determining the dynamics of the pitch system.

Estimation Based on all Actuator Models

To ease the notation, the system matrices so far contain only a single actuator model. However, the parameter estimation can benefit from using measurements from the three identical actuators, as all pitch actuators are assumed to possess identical dynamic behavior. The matrices defined in the previous subsection are extended to cover all three actuators; by redefining:

$$\begin{aligned}x(t) &= \begin{bmatrix} \beta_1(t) \\ \dot{\beta}_1(t) \\ \beta_2(t) \\ \dot{\beta}_2(t) \\ \beta_3(t) \\ \dot{\beta}_3(t) \end{bmatrix} & u(t) &= \begin{bmatrix} \beta_{\text{ref},1}(t - t_d) \\ \beta_{\text{ref},2}(t - t_d) \\ \beta_{\text{ref},3}(t - t_d) \end{bmatrix} & y(t) &= \begin{bmatrix} \beta_{1,\text{mes}}(t) \\ \beta_{2,\text{mes}}(t) \\ \beta_{3,\text{mes}}(t) \end{bmatrix} \\ A^+ &= \begin{bmatrix} A^+ & 0 & 0 \\ 0 & A^+ & 0 \\ 0 & 0 & A^+ \end{bmatrix} & A_{21}^\Delta &= \begin{bmatrix} A_{21}^\Delta & 0 & 0 \\ 0 & A_{21}^\Delta & 0 \\ 0 & 0 & A_{21}^\Delta \end{bmatrix} & A_{22}^\Delta &= \begin{bmatrix} A_{22}^\Delta & 0 & 0 \\ 0 & A_{22}^\Delta & 0 \\ 0 & 0 & A_{22}^\Delta \end{bmatrix} \\ B^+ &= \begin{bmatrix} B^+ & 0 & 0 \\ 0 & B^+ & 0 \\ 0 & 0 & B^+ \end{bmatrix} & B_{21}^\Delta &= \begin{bmatrix} B_{21}^\Delta & 0 & 0 \\ 0 & B_{21}^\Delta & 0 \\ 0 & 0 & B_{21}^\Delta \end{bmatrix}\end{aligned}$$

Having established a model structure and specified the varying entries of the system matrices, it is described in the next subsection how the estimation is performed.

Multiple-Model Estimation using an Extended Kalman Filter

The estimation problem in Eq. (5.30) is non-linear due to products of the states and the model weights. Different methods exist for solving such a problem, as exemplified in [Hallouzi, 2008, pp. 23-49]. The method chosen in this project is to utilize an extended Kalman filter, which uses linearization in the immediate state as an approximation to the non-linear system. This choice allows the method to be integrated into the reconfigurable extended Kalman filter described in Section 5.2.

The weights are assumed to evolve as a random walk process depicted by:

$$\mu(k) = \mu(k-1) + w_\mu(k) \quad (5.31)$$

where:

$w_\mu(k)$ is a zero-mean Gaussian distributed noise sequence with covariance $Q_\mu(k)$

The weights are introduced in the EKF by augmenting the state vector, which results in the following augmented model:

$$\underbrace{\begin{bmatrix} x(k) \\ \mu(k) \end{bmatrix}}_{x_a(k)} = \begin{bmatrix} \Phi_\mu(k-1|k-1) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k-1) \\ \mu(k-1) \end{bmatrix} + \begin{bmatrix} \Gamma_\mu(k-1|k-1) \\ 0 \end{bmatrix} u(k-1) + \underbrace{\begin{bmatrix} Q(k-1)^{1/2} & 0 \\ 0 & Q_\mu(k-1)^{1/2} \end{bmatrix}}_{Q_a(k-1)^{1/2}} w_a(k-1) \quad (5.32a)$$

$$y(k) = [H_\mu(k) \quad 0] \begin{bmatrix} x(k) \\ \mu(k) \end{bmatrix} + R(k)^{1/2} v(k) \quad (5.32b)$$

where:

$(\Phi_\mu, \Gamma_\mu, H_\mu)$ is the discretization of the system (A_μ, B_μ, C_μ) in Eq. (5.28)

$x_a(k)$ and $Q_a(k-1)$ are the augmented state vector and the augmented covariance matrix

$w_a(k)$ and $v(k)$ are zero-mean white noise sequences of appropriate dimensions

To introduce the Jacobian matrices required for the EKF, the system in Eq. (5.32) can be written as shown below:

$$x_a(k) = f(x_a(k-1), u(k-1), w_a(k-1)) \quad (5.33a)$$

$$y(k) = h(x_a(k), u(k), v(k)) \quad (5.33b)$$

Linearization is then used to calculate $\bar{\Phi}$ and \bar{H} at the predicted state. These are used to propagate the covariances in the EKF.

$$\bar{\Phi}(k-1) = \left. \frac{\partial f}{\partial x_a} \right|_{x_a = \hat{x}_a(k-1|k-1)} \quad (5.34)$$

$$\bar{H}(k) = \left. \frac{\partial h}{\partial x_a} \right|_{x_a = \hat{x}_a(k|k-1)} \quad (5.35)$$

The complete EKF algorithm is described in the bullet point list on the next page. Since the system description is available in continuous time, the matrices for the bullet point 'Matrix formulation' represent the continuous system matrices sampled at the discrete time sample $k-1$. These matrices are later discretized; e.g. $\bar{A}(k-1)$ is a counterpart to $\bar{\Phi}(k-1)$.

The main tuning parameter is the covariance matrix, $Q_a(k)$, which is selected as a diagonal matrix. The entries of $Q_\mu(k)$ determine the compromise between convergence rate of the weights and fluctuations of the estimates. The next subsection describes how constraints can be implemented on the estimated weights.

- **Matrix formulation**

The following matrices are linearized at $t = (k-1)T_s$.

$$A_\mu(k-1|k-1) = A^+ + A_{21}^\Delta \hat{\mu}_1(k-1|k-1) + A_{22}^\Delta \hat{\mu}_2(k-1|k-1)$$

$$B_\mu(k-1|k-1) = B^+ + B_{21}^\Delta (1 - \hat{\mu}_1(k-1|k-1))$$

$$C_\mu(k|k-1) = C$$

$$\bar{A}(k-1) = \begin{bmatrix} A_\mu(k-1|k-1) & [A_{21}^\Delta \quad -B_{21}^\Delta] & \begin{bmatrix} \hat{x}(k-1|k-1) \\ u(k-1) \end{bmatrix} \\ 0 & 0 & \begin{bmatrix} A_{22}^\Delta & 0 \\ 0 & \hat{x}(k-1|k-1) \end{bmatrix} \end{bmatrix}$$

$$\bar{C}(k) = [C \quad 0 \quad 0]$$

- **Discretization**

The above matrices are discretized using the following general equations:

$$\Phi = I + A\Psi$$

$$\Gamma = \Psi B$$

$$H = C$$

where

$$\Psi = IT_s + \frac{AT_s^2}{2!} + \frac{A^2T_s^3}{3!} + \dots$$

- **Prediction**

$$\begin{bmatrix} \hat{x}(k|k-1) \\ \hat{\mu}(k|k-1) \end{bmatrix} = \begin{bmatrix} \Phi_\mu(k-1|k-1) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k-1) \\ \hat{\mu}(k-1) \end{bmatrix} + \begin{bmatrix} \Gamma_\mu(k-1|k-1) \\ 0 \end{bmatrix} u(k-1)$$

$$P(k|k-1) = \bar{\Phi}(k-1)P(k|k-1)\bar{\Phi}(k-1)^T + Q_a(k|k-1)$$

- **Measurement update**

$$K(k) = P(k|k-1)\bar{H}(k)^T (\bar{H}(k)P(k|k-1)\bar{H}(k)^T + R(k))^{-1}$$

$$\begin{bmatrix} \hat{x}(k|k) \\ \hat{\mu}(k|k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k|k-1) \\ \hat{\mu}(k|k-1) \end{bmatrix} + K(k) (y(k) - H_\mu(k|k-1)\hat{x}(k|k-1))$$

$$P(k|k) = (I - K(k)\bar{H}(k)) P(k|k-1)$$

Constraint Implementation

The presented algorithm does not take the constraints on the weights into account. This however will be necessary to restrict the estimates to remain within the generated model set specified by $\hat{\mu}_j \in [0, 1]$. If one of the model weights escapes its interval, an equality constraint is added to the update step of the algorithm; i.e. either $\hat{\mu}_j = 0$ or $\hat{\mu}_j = 1$.

The limit value based method permits the estimates of the parameters to be found directly from the estimated weights. Since only ω_n appears in a_{21} while a_{22} includes a product of both ζ and ω_n , then $\hat{\omega}_n$ is determined from $\hat{\mu}_1$, while $\hat{\zeta}$ is found using $\hat{\mu}_2$ and $\hat{\omega}_n$. Therefore, it is possible to narrow the range of the second weight dependent on the estimate of the first weight, as shown below.

$$-\hat{\omega}_n^2 \in \left[-(\hat{\omega}_n^+)^2, -(\hat{\omega}_n^-)^2 \right] \quad \mu_1 \in [0, 1]$$

$$-2\hat{\zeta}\hat{\omega}_n \in [-2\zeta^+\hat{\omega}_n, -2\zeta^-\hat{\omega}_n] \quad \mu_2 \in \left[\frac{-2\zeta^+\hat{\omega}_n - a_{22}^+}{a_{22}^\Delta}, \frac{-2\zeta^-\hat{\omega}_n - a_{22}^+}{a_{22}^\Delta} \right]$$

In the implementation the intervals are slightly increased to obtain the correct mean values of the

estimates when the weights are at their boundaries. In the next subsection the multiple-model parameter estimation algorithm is verified.

Verification of Estimation of Pitch System Dynamics

To verify the estimation of the parameters in the pitch system, simulations are performed to evaluate how the parameters are estimated in the case of high air content in the hydraulic oil, which is slowly introduced. To view the evolution of the estimated parameters in case of hydraulic leakage, refer to Figure 5.16 in the next section, which presents a detection test for hydraulic leakage, but builds on the multiple-model parameter estimation from this section.

The parameter estimation is verified using Monte Carlo simulations, where 100 simulation runs are conducted for mean wind speeds of 16 m/s and 20 m/s to show any possible influence on the result caused by the operating conditions.

In Figure 5.14, the two upper subplots in each column show the estimated weights, whereas the lower subplots show the estimated parameters calculated based on the estimated weights. The reason why the estimates of ζ are noisier than the remaining estimates, is that this variable is a function of the two estimated weights, of which the first weight appears in the denominator of the equation calculating ζ .

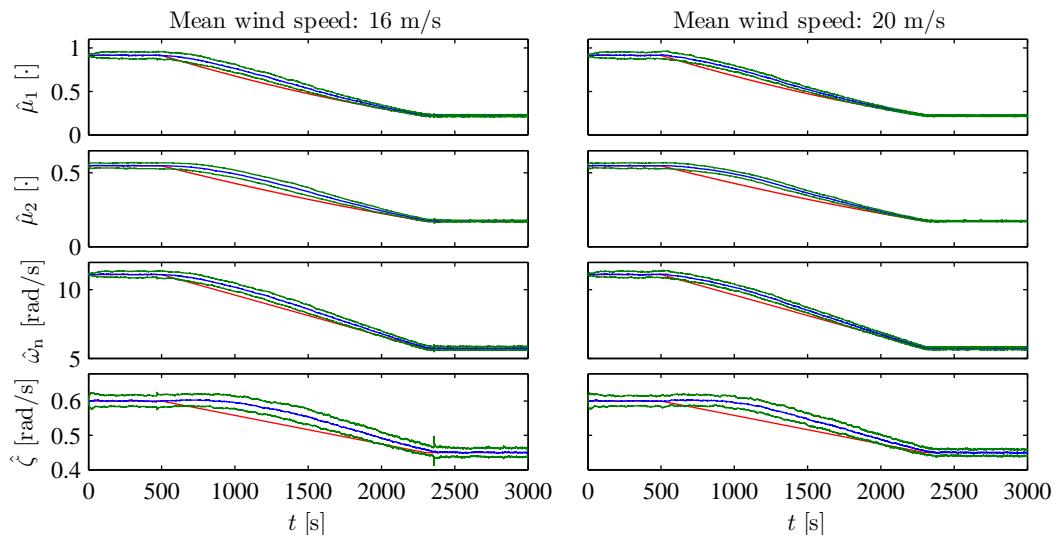


Figure 5.14: *Simulation results showing the estimation of the parameters in the pitch system. Between $t = 500$ s and $t = 2,200$ s the parameters in the pitch system are changed so that they eventually describe the condition of high air content in the oil. The mean values of the parameters (blue) are plotted with the mean plus and minus two times the standard deviations (green) and the correct values (red) to show how well the estimates follow the correct parameter values.*

From the figure it is concluded that the estimated parameters converge towards the correct parameters. The rate of convergence increases with increasing pitch activity. This implies that the estimate converges faster at high wind speeds.

In this section a method based on multiple-model estimation has been applied for estimating the parameters of the pitch system. The next section describes the design of a detection test taking advantage of the parameter estimation in this section to detect a hydraulic leakage in the pitch system.

5.6 Detection of Hydraulic Leakage

Hydraulic leakage differs from the other possible faults in the pitch system, since it eventually makes the pitch system uncontrollable, and since it is introduced much faster than the other faults. Therefore, it is not sufficient to estimate the parameters of the pitch system, since a hydraulic leakage should be detected such that the wind turbine can be shut down while the pitch system is still controllable. In this section a detection test is designed to detect a hydraulic leakage based on the parameter estimation described in the previous section.

A fast drop in the natural frequency of the pitch system model is utilized as an indicator of a hydraulic leakage. This can be detected by examining the slope of $\hat{\mu}_1(k)$ defined in Section 5.5. The upper left subplot in Figure 5.16 displays how a hydraulic leakage influences $\mu_1(t)$. Therefore, a method is now presented that detects a change in $\mu_1(t)$, which can be explained by a hydraulic leakage. It is decided to utilize a CUSUM test, since it can be used to detect a change in the mean of the derivative of $\hat{\mu}_1(k)$. The steps involved in the CUSUM test design are:

- Differentiate the estimate of the model weight $\hat{\mu}_1(k)$.
- Apply a CUSUM test to look for changes in the mean value of the derivative of $\hat{\mu}_1(k)$.

A CUSUM algorithm can be used for detecting a change in the mean of a Gaussian distributed sequence. In this case a one-sided test is applied since it is sufficient to detect a drop in the hydraulic pressure; i.e. detect a negative slope of $\hat{\mu}_1(k)$. The equation for calculating the decision variable, $g(k)$, is [Blanke *et al.*, 2006, p. 244]:

$$g(k) = \max \left(0, g(k-1) + \frac{\mu_b - \mu_a}{\sigma^2} \left(z(k) - \frac{\mu_a + \mu_b}{2} \right) \right) \quad (5.36)$$

The parameters: μ_a , μ_b , and σ are used to adjust the sensitivity of the decision function to changes in the input. The CUSUM algorithm then detects a change in the mean value from μ_a towards μ_b , where σ specifies the distribution of $\dot{\hat{\mu}}_1(k)$.

The example provided in Figure 5.15 shows how the decision function increases when a hydraulic leakage is emulated by decreasing the pressure in the oil to 50% between $t = 400$ s and $t = 500$ s. The specification of this simulation is identical to that in Figure 5.16. Therefore, the decision logic simply involves comparing $g(k)$ to a threshold and sending the detection signal to the supervisor, which much choose an appropriate control strategy.

The threshold is selected from a compromise between the mean detection time and the mean time between false alarms. In order to minimize the probability of false detections, and do not let high air content influence the detection of a hydraulic leakage, the threshold is selected such that no false alarms are made during the 100 simulation runs conducted to obtain Figure 5.14.

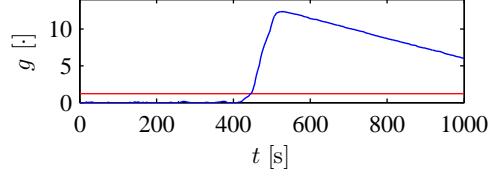


Figure 5.15: Decision variable (blue) and threshold (red) for the hydraulic leakage detection.

Verification of Hydraulic Leakage Detection

In this subsection simulations are performed to evaluate how hydraulic leakage in the system can be detected. Monte Carlo simulations of 100 runs are conducted to evaluate the results, and the test is conducted using mean wind speeds of 16 m/s and 20 m/s to show any possible influence on the result caused by a different wind speed.

In Figure 5.16 simulation results are shown for the estimation of parameters in the pitch system, when a hydraulic leakage is injected in the system at $t = 400$ s during 100 s. The distributions of the detection times are shown in Figure 5.17 using the CUSUM test.

The results confirm that the estimated parameters converge towards the correct parameters. The average detection time is about 35 s, which is a compromise that allows no false detections in the simulations performed in this verification. Note that the distribution is narrower at higher wind speeds, where the pitch activity is most extensive. From the resulting detection times it is

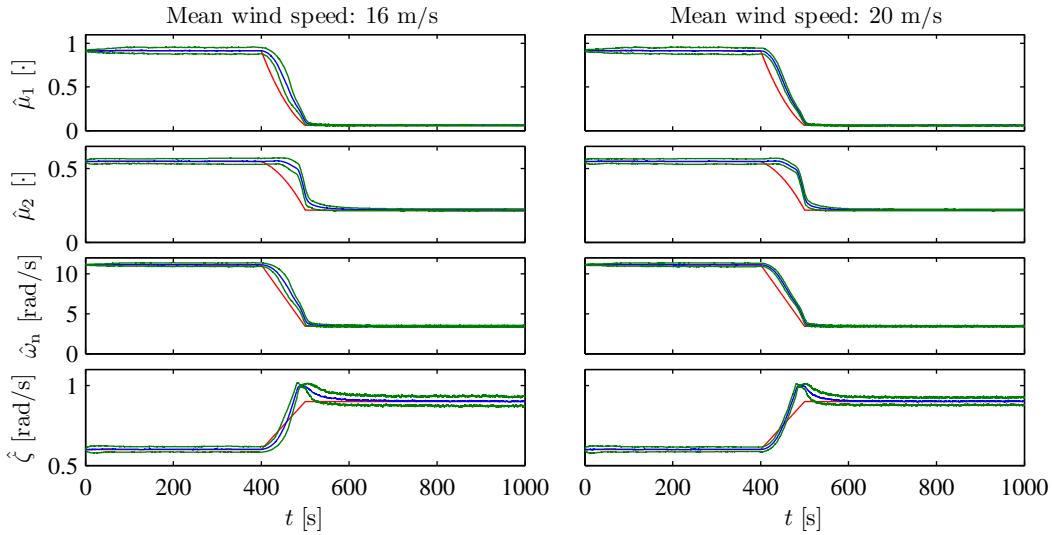


Figure 5.16: *Simulation results showing the estimation of the parameters in the pitch system. Between $t = 400$ s and $t = 500$ s the parameters in the pitch system are changed so that they eventually describe the condition of low pressure. The mean values of the parameters (blue) are plotted with the mean plus and minus two times the standard deviations (green) and the correct values (red) to show how well the estimates follow the correct parameter values.*

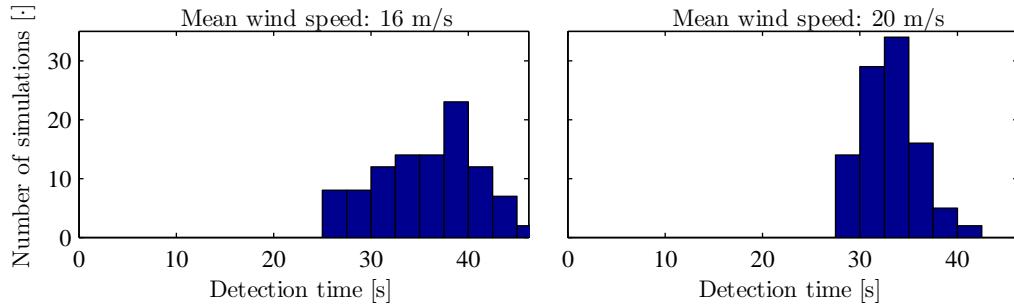


Figure 5.17: *Distributions of the detection times for the detection of a hydraulic leakage. The detection utilizes the estimate of the first weight in Figure 5.16 which is differentiated and used in a CUSUM test.*

concluded that the requirement in Section 4.6 is met.

In this section a method based on a one-sided CUSUM test has been described, which allows a hydraulic leakage to be detected based on the estimated dynamics of the pitch system explained in Section 5.5. As opposed to a hydraulic leakage, which develops during some time, a pitch actuator may also become stuck. The detection of this fault is described in the next section using two different methods for allowing fault detection in both partial load and full load operations.

5.7 Detection of Stuck Pitch Actuator

The purpose of this section is to describe how a stuck pitch actuator is detected. As mentioned in Section 4.6 a pitch actuator can either be stuck due to a valve blockage or pump blockage, of which the last affects all actuators. In contrast to the method described in Section 5.5, the methods considered in this section should, rather than estimating some parameters, be able to test the hypothesis: an actuator is stuck. The faults are going to be detected based on hypothesis testing using a multiple-model framework. The idea is to test if a fault-free or a faulty model description is the most likely description of the current system behavior.

A hypothesis test is directly performed in the full load region due to the mandatory pitch activity in this region. Contrarily, an auxiliary signal is applied to the system in the partial load region to actuate the pitch system, which is necessary to perform the hypothesis test. The main ideas of the two methods are summarized below; afterwards, the methods are presented and verified separately.

1. Multiple hypotheses testing

The first method uses the innovations and innovation covariances from two Kalman filters to find the most likely model.

2. Auxiliary signal design for failure detection

The second method uses a test signal of minimum energy combined with a hypothesis test to find the most likely model.

Detection of Stuck Pitch Actuator using Multiple Hypotheses Testing

To detect a stuck pitch actuator in the full load region a method based on multiple hypotheses testing is adopted, which is described in [Bak, 2000, pp. 16-18]. The method includes a number of models, each considered to be a candidate for the true model. A hypothesis is constructed for each model saying that the system is represented by that particular model. Additionally, the probability of each hypothesis being true is determined using the measurement data.

The multiple hypotheses testing set-up is shown in Figure 5.18. Notice that the filters are independent of each other, and that only their outputs are combined to calculate the model probabilities.

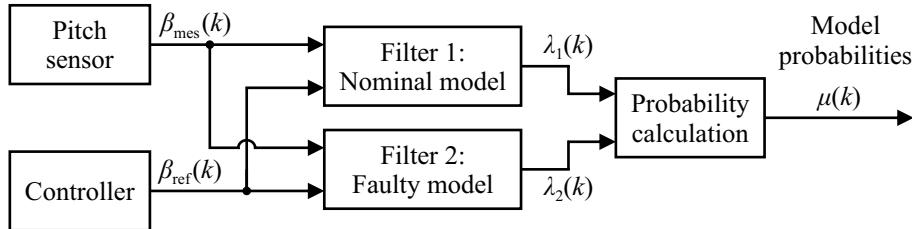


Figure 5.18: Structure of the multiple hypotheses testing method. The output $\mu(k)$ equals the probability of each model being correct, calculated based on their likelihoods $\lambda_1(k)$ and $\lambda_2(k)$.

The method has two assumptions listed below. The consequences of the first assumption are discussed in the subsection 'Robustness Associated with Detection of Stuck Pitch Actuator', while the consequences of the second assumption are discussed after having presented the algorithm.

1. The system is equivalent to one of the proposed models.
2. The same hypothesis has been true since $t = 0$ s.

The first assumption implies that there is no alternative hypothesis. To test if a pitch actuator is stuck, the model set is therefore made up of two models: one model describes the dynamics of a pitch actuator in the functional case, while the other model describes the scenario of a stuck actuator. The model of a functional pitch actuator appears in Eq. (5.24) on Page 65, while a stuck pitch actuator is modeled as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \begin{bmatrix} \dot{\beta}(t) \\ \ddot{\beta}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} \beta(t) \\ \dot{\beta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \beta_{\text{ref}}(t - t_d) \end{aligned} \quad (5.37a)$$

$$\begin{aligned} y(t) &= Cx(t) + v(t) \\ \beta_{\text{mes}}(t) &= [1 \quad 0] \begin{bmatrix} \beta(t) \\ \dot{\beta}(t) \end{bmatrix} + v_\beta(t) \quad [\circ] \end{aligned} \quad (5.37b)$$

It is decided to include a two-state model of the stuck pitch actuator to ease the tuning of the

algorithm, since this enables the use of similar covariances matrices in the two Kalman filters. The (2, 2)-th entry of the system matrix in Eq. (5.37a) ensures that any non-zero estimate of $\dot{\beta}(t)$ converges to zero.

At the end of this section a list is presented which shows the steps performed in each filter. This algorithm is based on the predictions and updates performed according to the Kalman filter equations, which can be found in [Grewal and Andrews, 2001, p. 121]. The idea is to write the probability of the i -th model being correct at sample k , conditioned on the measurements up to this time, so that it can be evaluated recursively. The recursive form can be obtained by using Bayes' rule, as shown in [Bak, 2000, p. 17]. This results in the following procedure, which is conducted for each filter, at each time step.

1. For filter i in Figure 5.18, perform the prediction and update step according to the Kalman filter equations.

- Find the innovation covariance maintained by the Kalman filter:

$$S_i(k|k-1) = H_i(k)P_i(k|k-1)H_i(k)^T + R_i(k)$$

- Compute the innovation:

$$r_i(k|k) = y(k) - \hat{y}_i(k|k-1)$$

2. Compute the likelihood of the observation $y(k)$ given that model M_i is valid (m denotes the number of measurements). The result is fed to the probability calculation box in Figure 5.18, to form the output of the hypothesis test described in item 3.

$$\lambda_i(k) = \frac{1}{(2\pi)^{m/2} \det(S_i(k|k-1))} \exp(-0.5 r_i(k|k-1)^T S_i^{-1}(k|k-1) r_i(k))$$

3. Find the probability that model M_i is correct at time step k , conditioned on the measurements up to that time, denoted Y_k . This is defined like $\mu_i(k) = p(M_i|Y_k)$ and is:

$$\mu_i(k) = \frac{\lambda_i(k)\mu_i(k-1)}{\sum_{j=1}^N \lambda_j(k)\mu_j(k-1)}$$

According to the second assumption, the method assumes that the same model has been in action since $t = 0$ s. This implies that the weight $\mu_i(k)$ converges to zero or one dependent on which model is most likely. In order to allow the weight to change during operation this assumption is relaxed by adding a small number to the weight $\mu_i(k)$, which enables it to move away from zero if the state of the system suddenly changes.

Robustness Associated with Detection of Stuck Pitch Actuator

If the nominal pitch actuator model and the model of a stuck pitch actuator are selected as the two candidate models, the algorithm will perform well in the scenarios of having either nominal performance or a stuck actuator. However, the method will not necessarily be robust towards deviations in the pitch dynamics. In that situation, e.g., high air content in the oil of the pitch system may result in false detections of a stuck pitch actuator.

The candidate models have to be selected such that it can be determined if an actuator is stuck, independent of the condition of the pitch system, i.e. for any parameter values in the range stated in Table 4.8 on Page 40. However, this compromises the assumption that the system is equivalent to one of the proposed models. Even though none of the hypotheses can formally be accepted, the test will still provide an answer to which model is most likely. Therefore, it is possible to let a model having degraded performance represents the functional case; this will still have a higher likelihood than that of a stuck actuator, even if the nominal model is the true model, although robustness cannot be guaranteed.

It has been found that the hypothesis test performs satisfactory if the functional case is modeled as the situation of low pressure and normal air content, with parameters found in Table 4.8 on Page 40. This however has to be verified, and is successfully done through simulations in the next subsection.

Verification of Stuck Pitch Actuator Detection in Full Load Operation

In order to evaluate the performance of the multiple-model hypothesis testing method, and to ensure that the test is functional even in cases where the system is not equivalent to one of the proposed models, Monte Carlo simulations are performed by considering 200 simulation runs. These runs are based on the same model, each with different seeds for the random sequences of the measurement noise and the wind model. The test is repeated for wind speeds of both 16 m/s and 20 m/s to reveal possible influences on the results caused by the operating point of the wind turbine. Since no a priori knowledge is assumed, the probabilities that the models are correct are initialized to one half for each model.

In Figure 5.19 the mean values of the model probabilities are shown separately for the tests performed at the two mean wind speeds. During the simulations the dynamics of the pitch system is altered, and it is verified that no false detections are made in none of the simulations.

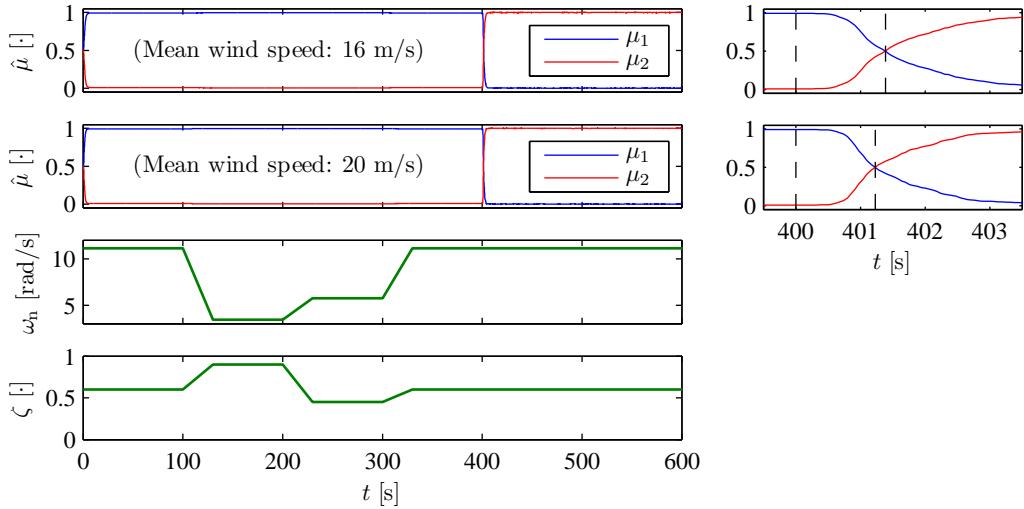


Figure 5.19: *Average results for the detection of a stuck pitch actuator for two different mean wind speeds. The dynamics of the pitch system is altered according to the two lower subplots to verify that no false detections are made in different scenarios. At $t = 400$ s the pitch actuator becomes stuck.*

The distributions of the detection times are shown in Figure 5.20, whereof it can be concluded that a detection time of 1 s is the most likely, but may be as long as 5.6 s in the considered simulations.

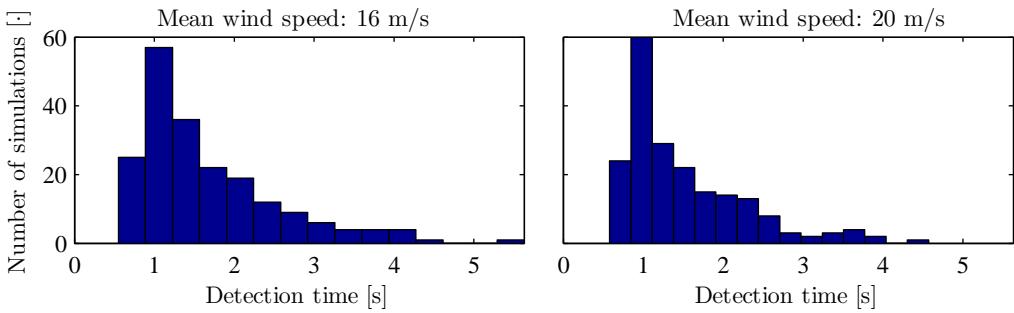


Figure 5.20: *Distributions of detection times. The mean detection time is $t = 1.66$ s and $t = 1.52$ s respectively for the simulations performed for the lower and higher mean wind speeds.*

Based on the results it is concluded that the detection test operates satisfactorily and that the detection time is only marginally affected by the average wind speed.

After having concluded that the detection of a stuck pitch actuator works satisfactorily in the full

load region, the next subsection will present a method for conducting a similar test in the partial load region.

Detection of Stuck Pitch Actuator using Auxiliary Signal Design

In this subsection a test is designed for detecting a stuck pitch actuator in the partial load region. In this region there is usually no pitch activity; hence, it is not directly possible to diagnose the pitch system since it requires excitation of the system. Therefore, an auxiliary input signal to the pitch system can be designed, which enables fault detection, while disturbing the system as little as possible.

The requirements to the detection algorithm are shown below:

1. Detect the fault even though the controller does not excite the faulty system.

The controller does not actuate the pitch system during the test; hence, a method based on active fault detection should be utilized for this fault detection.

2. Possess robustness towards changing dynamics of the pitch system.

The detection algorithm should separate a normal pitch system (having 80% to 100% hydraulic pressure) from a faulty pitch system (having 0% to 60% hydraulic pressure). Notice that it is not desired to estimate the parameters of the pitch system, but only to detect the fault.

A method that satisfies these two requirements is explained in [Campbell and Nikoukhah, 2004, pp. 76-90] and will be utilized for the detection.

Before designing the detection test, an overview of the detection procedure is provided in the block diagram shown in Figure 5.21.

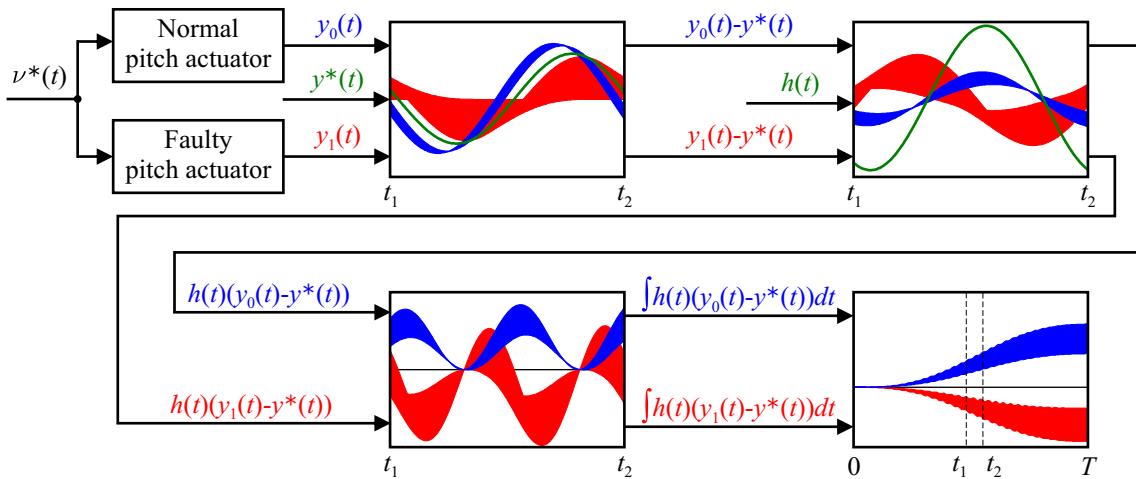


Figure 5.21: Block diagram of the active fault detection procedure for detecting a stuck pitch actuator. The signals shown between the blocks belong to both the normal system (blue) and faulty system (red). Notice that the three first plots only show a small part of the detection period.

The block diagram shown in Figure 5.21 visualizes the propagation of the signals through the detection test. Starting from the upper left, an optimal auxiliary signal, $\nu^*(t)$, is applied to the pitch system; this can either have normal or faulty behavior. All possible outputs of these models are contained in the two sets visualized by the blue and red regions. If the worst disturbance is applied to both systems, then they both equal $y^*(t)$. A signal $h(t)$ is designed such that $\int_0^T h(t)(y(t)-y^*(t))dt$ is greater than zero for all possible outputs from the normal system and less than zero for all outputs from the faulty system. This integral is illustrated in the last plot in Figure 5.21 and makes it possible to accept the correct hypothesis.

The section is divided into a number of subsections described below.

- **Problem Formulation for Auxiliary Signal Design:** Set up the optimization problem, from which the auxiliary signal should be found.
- **Calculation of Optimal Auxiliary Signal:** Solve the optimization problem and calculate the optimal auxiliary signal.
- **Design of Detection Test:** Design a hypothesis test for the fault detection.
- **Implementation Remarks:** Comment on the work that must be done to implement the detection test.
- **Verification of Stuck Pitch Actuator Detection in Partial Load Operation:** Verify that the detection test performs according to the specification.

Problem Formulation for Auxiliary Signal Design

In this subsection the state space formulations of the normal and faulty systems are set up, and the optimization problem involved with the design of the auxiliary signal is formulated.

Both the normal and faulty systems are uncertain. Therefore, the two parameter-varying models must be expressed as shown in Eq. (5.38).

$$\dot{x}_i(t) = A_i x_i(t) + B_i \nu(t) + M_i w_i(t) \quad (5.38a)$$

$$z_i(t) = G_i x_i(t) + H_i \nu(t) + K_i w_i(t) \quad (5.38b)$$

$$E_i y(t) = C_i x_i(t) + D_i \nu(t) + N_i w_i(t) \quad (5.38c)$$

where:

i indicates if it is the normal model ($i = 0$) or the faulty model ($i = 1$)

$w_i(t)$ is the additive uncertainty for Model i

$z_i(t)$ is the model uncertainty for Model i

$\nu(t)$ is the auxiliary signal

In this uncertain setting $w_i(t)$ is assumed to include the process noise, measurement noise, and uncertainty $z(t)$. It is required that N_i has full row rank, i.e. there should be noise on all measurements. In the considered case K_i is assumed to be zero. Additionally, it is assumed that the input is zero during the fault detection period, since the wind turbine is operated in the partial load region.

The idea behind the fault detection method is to design a signal which is only just large enough to let the normal and faulty models be completely separated within the detection time, given some bounds on the uncertainty and noise in the systems.

The auxiliary signal should be designed such that the constraint on $S_i(\nu(t), s)$ in Eq. (5.39) is only satisfied when Model i is the correct model of the system.

$$S_i(\nu(t), s) = x_i^T(0) P_{i0}^{-1} x_i(0) + \int_0^s \begin{bmatrix} w_i(t) \\ z_i(t) \end{bmatrix}^T \underbrace{\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}}_{J_i} \begin{bmatrix} w_i(t) \\ z_i(t) \end{bmatrix} dt$$

$$S_i(\nu(t), s) < 1, \forall s \in [0, T] \quad (5.39)$$

where:

J_i is a signature matrix for System i

Since N_i has full row rank, Eq. (5.38) can always be satisfied by selecting $w_i(t)$ appropriately. From Eq. (5.38) it is clear that if Model i is incorrect then $w_i(t)$ is larger than when Model i is correct. To form a single optimization problem, the normal and faulty models are combined into one model. This implies that the constraint in Eq. (5.39) is broken when the considered model is incorrect. Furthermore, Eq. (5.39) allows a greater $w_i(t)$ when the system is uncertain, to ensure that the system is outside its uncertainty space.

If an auxiliary signal is designed such that Eq. (5.38) and Eq. (5.39) cannot be satisfied for both $i = 0$ and $i = 1$, then the auxiliary signal is said to be proper. Notice that this can only happen by

violating Eq. (5.39), since N_i has full row rank. This implies that it is sufficient to check Eq. (5.39) subject to Eq. (5.38).

An expression including the maximum operator can be rewritten using the following general equation. This is utilized in the remaining equations:

$$\max(\eta_1, \eta_2) = \max_{0 \leq \epsilon \leq 1} (\epsilon\eta_1 + (1 - \epsilon)\eta_2) \quad (5.40)$$

This relation is used to write the following optimization problem, which can be utilized to find a proper auxiliary signal:

$$\sigma(\nu(t), s) = \max_{\epsilon \in [0, 1]} \phi_\epsilon(\nu(t), s) \quad (5.41)$$

where

$$\phi_\epsilon(\nu(t), s) = \inf_{\substack{w_0(t), w_1(t), y(t), \\ x_0(t), x_1(t)}} \epsilon S_0(\nu(t), s) + (1 - \epsilon) S_1(\nu(t), s)$$

A measure that indicates how hard it is to separate the two models is the separability index:

$$\gamma^* = \left(\inf_{\nu \in V} \int_0^T \|\nu(t)\|^2 dt \right)^{-\frac{1}{2}} \quad (5.42)$$

where:

- V is the set of proper auxiliary signals
- γ^* is the optimal separability index

From the equation it is seen that the separability index is large when a small proper auxiliary signal exists due to the models being easily separable, and vice versa for a small separability index.

To find the maximum separability index, i.e. the smallest auxiliary signal that separates the models, the constraint shown in Eq. (5.39) must be reformulated to an optimization problem. This implies combining the normal and faulty models into one model, to form a single optimization problem. The first step in this procedure is to eliminate $y(t)$ from the equations, by pre-multiplying F , calculated below, on the output equation.

$$F = [F_0 \quad F_1] = \begin{bmatrix} E_0 \\ E_1 \end{bmatrix}^\perp \quad (5.43)$$

where:

- $[]^\perp$ is a basis for the null space of $[]$

The second step is to combine the system matrices as shown below.

$$\begin{aligned} x(t) &= \begin{bmatrix} x_0(t) \\ x_1(t) \end{bmatrix}, w(t) = \begin{bmatrix} w_0(t) \\ z_0(t) \\ w_1(t) \\ z_1(t) \end{bmatrix}, A = \begin{bmatrix} A_0 & 0 \\ 0 & A_1 \end{bmatrix}, B = \begin{bmatrix} B_0 \\ B_1 \end{bmatrix}, M = \begin{bmatrix} M_0 & 0 & 0 & 0 \\ 0 & M_1 & 0 & 0 \end{bmatrix}, \\ C &= \begin{bmatrix} F_0 C_0 & F_1 C_1 \\ G_0 & 0 \\ 0 & G_1 \end{bmatrix}, D = \begin{bmatrix} F_0 D_0 + F_1 D_1 \\ H_0 \\ H_1 \end{bmatrix}, N = \begin{bmatrix} F_0 N_0 & F_1 N_1 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}, \\ P_\epsilon^{-1} &= \begin{bmatrix} \epsilon P_{0,0}^{-1} & 0 \\ 0 & (1 - \epsilon) P_{1,0}^{-1} \end{bmatrix}, J_\epsilon = \begin{bmatrix} \epsilon J_0 & 0 \\ 0 & (1 - \epsilon) J_1 \end{bmatrix} \end{aligned}$$

The combined system description reformulates Eq. (5.39) into the optimization problem:

$$\phi_\epsilon(\nu(t), s) = \inf_{w(t), x(t)} x^T(0) P_\epsilon^{-1} x(0) + \int_0^s w^T(t) J_\epsilon w(t) dt \quad (5.44)$$

subject to

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B\nu(t) + Mw(t) \\ 0 &= Cx(t) + D\nu(t) + Nw(t)\end{aligned}$$

The input signal is said to be optimal if its energy is minimized. As a consequence of this objective, the optimization problem becomes:

$$\min_{\nu(t)} \int_0^T \nu^T(t)\nu(t) dt \quad (5.45)$$

subject to

$$\phi_\epsilon(\nu(t), s) \geq 1$$

The final optimization problem can then be written as shown below.

$$J(s, \nu(t), w(t), x(t)) = \max_{\nu(t)} \inf_{w(t), x(t)} x^T(0)P_\epsilon^{-1}x(0) + \int_0^s (w^T(t)J_\epsilon w(t) - \nu^T(t)\lambda I\nu(t)) dt \quad (5.46)$$

The necessary matrices and an appropriate optimization problem have been set up. The following subsection is concerned with calculating the optimal auxiliary signal.

Calculation of Optimal Auxiliary Signal

To calculate the optimal auxiliary signal the following optimization problem must be solved:

$$J(s, \nu(t), w(t), x(t)) = \max_{\nu(t)} \inf_{w(t), x(t)} x^T(0)P_\epsilon^{-1}x(0) + \int_0^s (w^T(t)J_\epsilon w(t) - \nu^T(t)\lambda I\nu(t)) dt \quad (5.47)$$

subject to

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B\nu(t) + Mw(t) \\ 0 &= Cx(t) + D\nu(t) + Nw(t)\end{aligned}$$

According to [Campbell and Nikoukhah, 2004, pp. 46], the optimization problem above has a solution if the following Riccati equation has a solution on $[0, T]$.

$$\begin{aligned}\dot{P} &= (A - S_{\lambda,\epsilon}R_{\lambda,\epsilon}^{-1}C)P + P(A - S_{\lambda,\epsilon}R_{\lambda,\epsilon}^{-1}C)^T - PC^T R_{\lambda,\epsilon}^{-1}CP \\ &\quad + Q_{\lambda,\epsilon} - S_{\lambda,\epsilon}R_{\lambda,\epsilon}^{-1}S_{\lambda,\epsilon}^T, \quad P(0) = P_\epsilon\end{aligned} \quad (5.48)$$

where

$$\begin{bmatrix} Q_{\lambda,\epsilon} & S_{\lambda,\epsilon} \\ S_{\lambda,\epsilon}^T & R_{\lambda,\epsilon} \end{bmatrix} = \begin{bmatrix} M & B \\ N & D \end{bmatrix} \begin{bmatrix} J_\epsilon & 0 \\ 0 & -\lambda I \end{bmatrix}^{-1} \begin{bmatrix} M & B \\ N & D \end{bmatrix}^T$$

To find the maximum separability index a grid of values of ϵ is created ranging from 0 to 1. The index λ, ϵ indicates that the particular matrix is calculated at certain λ and ϵ values. At each of these points the largest value of λ , for which the Riccati equation has a solution on $[0, T]$, is found by iteration. The maximum separability index is called λ^* and is calculated at ϵ^* . At this point the auxiliary signal is expected to have minimum energy.

By using the solution of the Riccati equation at the optimum ϵ^* , the optimum input is calculated by determining the solution of the following two-point boundary value system for the state, $x(t)$, and Lagrange multiplier, $\zeta(t)$:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} \quad (5.49)$$

where

$$\begin{aligned}\Omega_{11} &= A - S_{\lambda^*, \epsilon^*} R_{\lambda^*, \epsilon^*}^{-1} C \\ \Omega_{12} &= Q_{\lambda^*, \epsilon^*} - S_{\lambda^*, \epsilon^*} R_{\lambda^*, \epsilon^*}^{-1} S_{\lambda^*, \epsilon^*}^T \\ \Omega_{21} &= C^T R_{\lambda^*, \epsilon^*}^{-1} C \\ \Omega_{22} &= -\Omega_{11}^T\end{aligned}$$

The boundary values of Eq. (5.49) must be:

$$x(0) = P(0)\zeta(0) \quad (5.50a)$$

$$\zeta(T) = 0 \quad (5.50b)$$

From the boundary values it is seen that $x(T)$ must be non-zero. Therefore, from Eq. (5.51) it is clear that $x(T)$ must be in the null space of $P^{-1}(T)$. This is possible, since $P(t)$ becomes singular at time T , when solving the Riccati equation.

$$x(t) = P(t)\zeta(t) \quad (5.51)$$

After solving the two-point boundary value system, the trajectories of $x(t)$ and $\zeta(t)$ are used to calculate the auxiliary signal, as shown in the following equation. To get the optimal proper auxiliary signal, α must be chosen such that $\|\nu^*(t)\| = 1/\gamma^*$.

$$\nu^*(t) = \alpha \left((S_{\lambda^*, \epsilon^*} R_{\lambda^*, \epsilon^*}^{-1} D - B)^T \zeta(t) + D^T R_{\lambda^*, \epsilon^*}^{-1} C x(t) \right) / \lambda^* \quad (5.52)$$

The optimal proper auxiliary signal has now been determined and the only remaining task is to design the detection test, which determines the accepted hypothesis.

Design of Detection Test

There are multiple ways for determining which of the two hypotheses that is the most probable. Since the test should run online it is chosen to design a detection test, which only relies on offline calculated signals in combination with the measured system output. Other simpler tests, relying on observers, could have been designed, but these would require a higher online computational effort.

The overall idea behind the test is to evaluate the integral shown in Eq. (5.53) over the test period, and if it is positive, the system is assumed to operate according to the normal model. If it is negative, the system is assumed to operate according to the faulty model. The resulting signal is shown in Figure 5.21 on Page 75 for both models.

$$l = \int_0^T h^T(t) (y(t) - y^*(t)) dt \quad (5.53)$$

where

$$\begin{aligned}h(t) &= \left(F \begin{bmatrix} E_0 \\ -E_1 \end{bmatrix} \right)^T R_{\lambda^*, \epsilon^*}^{-1} (Cx + S_{\lambda^*, \epsilon^*}^T \zeta) \\ y^*(t) &= \alpha \text{pinv} \left(\begin{bmatrix} E_0 \\ E_1 \end{bmatrix} \right) \left(\begin{bmatrix} C_0 & 0 \\ 0 & C_1 \end{bmatrix} - \Psi_{\lambda^*, \epsilon^*} \begin{bmatrix} -\zeta \\ F^T R_{\lambda^*, \epsilon^*}^{-1} (Cx + S_{\lambda^*, \epsilon^*}^T \zeta) \end{bmatrix} \right) \\ \Psi_{\lambda^*, \epsilon^*} &= \begin{bmatrix} J_{\epsilon^*} & 0 \\ 0 & -\lambda^* I \end{bmatrix}^{-1} \begin{bmatrix} M & B \\ \mathcal{N} & \mathcal{D} \end{bmatrix}^T, \mathcal{N} = \begin{bmatrix} N_0 & 0 \\ 0 & N_1 \end{bmatrix}, \mathcal{D} = \begin{bmatrix} D_0 & 0 \\ 0 & D_1 \end{bmatrix}\end{aligned}$$

where:

$\text{pinv}(\)$ is the pseudo inverse operator

If the worst noise is applied to each of the models, the signal $y^*(t)$ equals the output obtained by both models. This implies that $l = 0$ if the worst noise is applied. Generally, $y^*(t)$ has a value between the two sets of outputs from the systems, as shown in Figure 5.21 on Page 75. The signal $h(t)$ must ensure that Eq. (5.53) gets the correct sign. This implies that $h(t)$ must possess the following properties:

- Change sign when $y(t) - y^*(t)$ changes sign.
- Have a large magnitude when the outputs of the two models are far apart.
- Have a small magnitude when the outputs of the two models intersect.

The detection test has been described and only some implementation remarks remains.

Implementation Remarks

Since the method does not assume the noise to be stochastic, but defines an energy bound on the noise instead, the optimization problem solved to find the auxiliary signal does not fit exactly to the description of the system. Therefore, it is necessary to determine the probability of accepting the false hypothesis, calculated based on a stochastic noise sequence, in accordance with the assumptions about the measurement noise. This test must be evaluated when the auxiliary signal has been designed to ensure that it can be utilized for the stochastic model.

The overall requirement to the detection is: an erroneous detection can only happen in average once in the lifetime of the wind turbine. Assuming that the lifetime of the wind turbine is 20 years and that the test is executed once per hour, the probability of accepting an incorrect hypothesis can be calculated as shown below. This requirement is evaluated in the next subsection.

$$\begin{aligned} P_{\text{error}} &= \left(\frac{1}{24 \cdot 365.25 \cdot 20} \right) \cdot 100\% \\ &= 0.0006\% \end{aligned} \quad (5.54)$$

where:

P_{error} is the probability of accepting an incorrect hypothesis

The detection time, T , is chosen to 30 s, which is a compromise between obtaining a small amplitude of the auxiliary signal and getting a small detection time. The detection time is small compared to the time used to lubricate the pitch system in partial load operation, which has duration of several minutes.

To solve the optimization problem involved in designing the auxiliary signal, some numerical issues exist, since the matrix P in Eq. (5.48) ideally becomes singular at time T . Therefore, Eq. (5.48) is solved from time 0 until $T - \delta$, with δ being a small fraction of number T . Afterwards, the equivalent Riccati equation for P^{-1} is solved from $T - \delta$ until T . This implies that P^{-1} is obtained at time T , which should be utilized to calculate $x(T)$ in its null space, as previously explained. However, numerical problems exists in solving the optimization problem since the solver cannot provide a singular P^{-1} , and breaks down just before reaching this. To determine $x(t)$ and $\zeta(t)$ in the interval between 0 and T , two different methods are exploited to circumvent the numerical problems. In the interval between T and $T - \delta$ Eq. (5.49) is utilized directly, whereas in the remaining interval Eq. (5.51) is inserted into Eq. (5.49) and the resulting equation is solved.

From the design procedure the auxiliary signal shown in Figure 5.22 is obtained, which should be applied to the three pitch actuators.

The designed signal and matching detection test is evaluated in the next subsection.

Verification of Stuck Pitch Actuator Detection in Partial Load Operation

To verify that the probability of getting a faulty detection is less than the bound specified in Eq. (5.54), the probability is evaluated for pitch systems having pressures between 0% and 100% with a step size of 1%. From the expected value and the variance, the probability of l having the

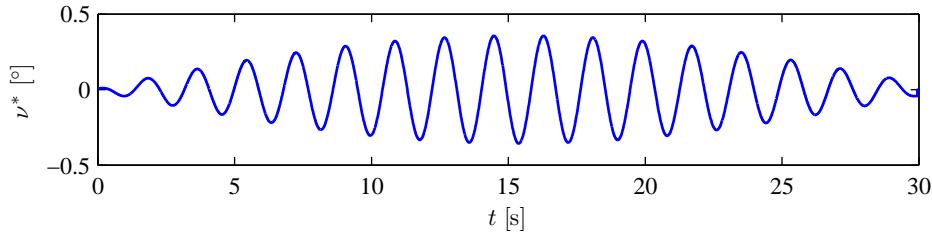


Figure 5.22: Optimal proper auxiliary signal for detection of a stuck pitch actuator. Notice that $\nu^*(t)$ is a representation of the optimal auxiliary signal applied to $\beta_{\text{ref}}(t)$.

incorrect sign can be calculated:

$$l = \sum_{k=1}^{T/T_s} h(k) (y(k) - y^*(k)) \quad (5.55)$$

$$\begin{aligned} E[l] &= \sum_{k=1}^{T/T_s} E[h(k) (y_{\text{det}}(k) + v_\beta(k) - y^*(k))] \\ &= \sum_{k=1}^{T/T_s} h(k) (y_{\text{det}}(k) - y^*(k)) \end{aligned} \quad (5.56)$$

$$\text{Var}[l] = \sum_{k=1}^{T/T_s} h^2(k) \text{Var}[v_\beta(k)] \quad (5.57)$$

where:

$y(k)$ is the measurement of the pitch angle with measurement noise $v_\beta(k)$ [°]

$y_{\text{det}}(k)$ is the de deterministic value of the pitch angle [°]

Figure 5.23 shows the probabilities of accepting the normal and faulty systems at different pressures of the hydraulic oil. From this figure it is apparent that the requirement is satisfied.

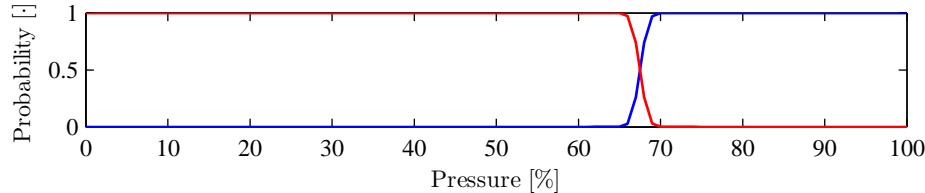


Figure 5.23: Probability of accepting either of the two hypotheses as a function of the pressure in the pitch system. At pressures below 60% the probability of accepting the for the faulty system (red) is high, while it is high for the normal system (blue) at pressures above 70%.

From the verification it is concluded that the auxiliary signal shown in Figure 5.22 combined with the designed detection test is capable of separating the two uncertain systems, with a probability of getting false detections that is smaller than one per 20 years.

5.8 Diagnosis of Proportional Speed Error

Proportional error of the generator speed sensor output arises due to improper configuration or impurity building up at the disc of the speed encoder, according to kk-electronic a/s. The purpose of this section is to present an algorithm for diagnosing this fault, for which a model is found in Eq. (4.10) on Page 41 and is repeated below for convenience.

$$\omega_{g,mes}(k) = \underbrace{\omega_g(k)(1 + \alpha_{pe}(k))}_{\text{sensor gain}} + v_{\omega_g}(k) \quad [\text{rad/s}] \quad (5.58)$$

where:

$v_{\omega_g}(k)$ is a zero-mean Gaussian distributed noise sequence with variance $\sigma_{\omega_g}^2$ [rad/s]

$\alpha_{pe}(k)$ determines the proportional error [·]

$\omega_{g,mes}(k)$ is the generator speed measurement [rad/s]

According to Section 4.4 the fault may be seen as inconsistency in the variables such as generator speed and rotor speed. However, this can only be used for fault detection, since inconsistency in two variables does not make it possible to isolate the fault. Instead, it can be exploited that a blade will pass in front of the tower three times per rotation, regardless of any other faults. Therefore, the fault is possible to diagnose by exploiting the reduced aerodynamic torque when a blade passes the tower. By comparing this information with the measured generator speed the proportional error can be estimated.

In this section fault detection based on a consistency check of the generator speed and rotor speed is presented, and two methods for estimating the proportional speed error are presented. The first method is based on feature extraction while the second method is model-based. Both methods are included in order to investigate the applicability of the different approaches. The main ideas of these methods are summarized below, afterwards the methods are presented in turns, and the section is finalized by verifying the two methods.

1. Peak detection using window function

The method uses signal analysis to detect peaks in the measured generator speed, which originates from a blade passing the tower.

2. Finding 3P frequency using oscillator model

The method uses a model-based approach to determine the difference between the presumed and actual 3P frequency.

Alternative methods such as Wavelets transform or Karhunen Loéve expansion, which exploit the shape of the signal in the time domain, could have been chosen. However, this has not been done due to their higher complexity.

Detection of Proportional Speed Error

To detect proportional error on the generator speed sensor, a consistency check of the generator speed and rotor speed measurements is performed. It is not possible directly to compare the output values of the sensors due to noise, but by calculating their mean values, it is possible to detect an inconsistency, such as a proportional error.

It is favorable to utilize a recursive algorithm to calculate the mean value, since this eliminates the necessity for saving old measurement data. Additionally, it is only desired to evaluate the mean value over the most recent samples, since a changing mean value should be detected. To obey these requirements a recursive algorithm with build in forgetting, i.e. new samples are weighted higher than old samples, is chosen for the detection test. The chosen equation used to calculate the mean value is shown in Eq. (5.59). This equation is similar to the recursive approach for calculating the true mean, except that N is a fixed number in the recursive algorithm instead of an ever increasing number.

$$\hat{\mu}(k) = \frac{1}{N}x(k) + \frac{N-1}{N}\hat{\mu}(k-1) \quad (5.59)$$

where:

N is a constant determining the forgetting factor of the algorithm [·]

The forgetting factor in the algorithm can be written in terms of N as shown below.

$$\hat{\mu}(k) = \sum_{i=0}^k \underbrace{\left(\frac{N-1}{N}\right)^{k-i}}_{\text{forgetting factor for sample } k} \frac{1}{N}x(i) \quad (5.60)$$

The value that is evaluated in the detection of a proportional error is the difference between the generator speed and rotor speed measurements as shown in Eq. (5.61). Notice that $\omega_g(k)$ is assumed to equal $N_g\omega_r(k)$ in the remaining part of the subsection. This is a reasonable assumption, since only the mean values of the two signals are evaluated.

$$\begin{aligned}\Delta_{pe}(k) &= \omega_{g,mes}(k) - N_g\omega_{r,mes}(k) \\ &= \omega_g(k)(1 + \alpha_{pe}(k)) + v_{\omega g}(k) - N_g(\omega_r(k) + v_{\omega r}(k)) \\ &= \omega_g(k)\alpha_{pe}(k) + v_{\omega g}(k) - N_gv_{\omega r}(k) \quad [\text{rad/s}]\end{aligned}\quad (5.61)$$

where:

$\Delta_{pe}(k)$ is the proportional error residual [rad/s]

To choose an appropriate value of N , some requirements are set up below using the following scenario: *a proportional error, corresponding to a fault equal to 1 rad/s on the generator speed measurement, is introduced over 30 minutes.*

1. A false positive or false negative detection must happen maximum once per 20 years.
2. The detection time must be less than 30 minutes.

Calculations based on the expectation and variance of $\Delta_{pe}(k)$ show that the value of N should be between 513 and 99,791 to satisfy the above requirements. To get the shortest detection time, N is chosen to 513. The successful detection of a proportional error resulting in a speed offset of 1 rad/s implies that a proportional error of 5% can be detected satisfying the requirement in Section 4.6.

The detection of a proportional error, by use of a recursive algorithm, has been described. The following two subsections describe how the proportional error can be estimated, using two different methods.

Estimation based on Peak Detection using Window Function

The method described in this subsection is inspired by a basic spike detection method presented in [Hassanpour *et al.*, 2004].

The starting point of the derived algorithm is the energy operator written in Eq. (5.62), which is used to amplify the signatures of peaks in the measured generator speed signal, originating from a blade passing the tower.

$$\kappa(k) = \sum_{i=k-\frac{N_w-1}{2}}^{k+\frac{N_w-1}{2}} \omega_{g,mes}^2(i) - \left(\sum_{j=k-\frac{3N_w-1}{2}}^{k-\frac{N_w+1}{2}} \omega_{g,mes}(j) \right) \left(\sum_{l=k+\frac{N_w+1}{2}}^{k+\frac{3N_w-1}{2}} \omega_{g,mes}(l) \right) \quad [(\text{rad/s})^2] \quad (5.62)$$

where:

N_w is an odd integer denoting the peak width in samples [.]

$\kappa(k)$ is the energy operator of the time series $\omega_{g,mes}(k)$ $[(\text{rad/s})^2]$

The energy operator amplifies peaks in the signal by calculating the difference between the energy in a given time interval and the energy in the surrounding time intervals. The local drops at the output of the energy operator that are lower than a predefined threshold are then considered as an indication of a peak at that location in the time series. Notice that the energy operator function is non-causal; hence, the detection of the blade pass is delayed in the implementation.

Excerpts of the generator speed, measured generator speed, and the energy operator applied at the measured generator speed are provided in Figure 5.24. The figure shows that the peak signatures in the measured generator speed are amplified although they are difficult to see on the measurement signal. The threshold is implemented such that it depends on a low-pass filtered version of the measured wind speed, since the clarity in the signatures of the peaks depend on the wind speed.

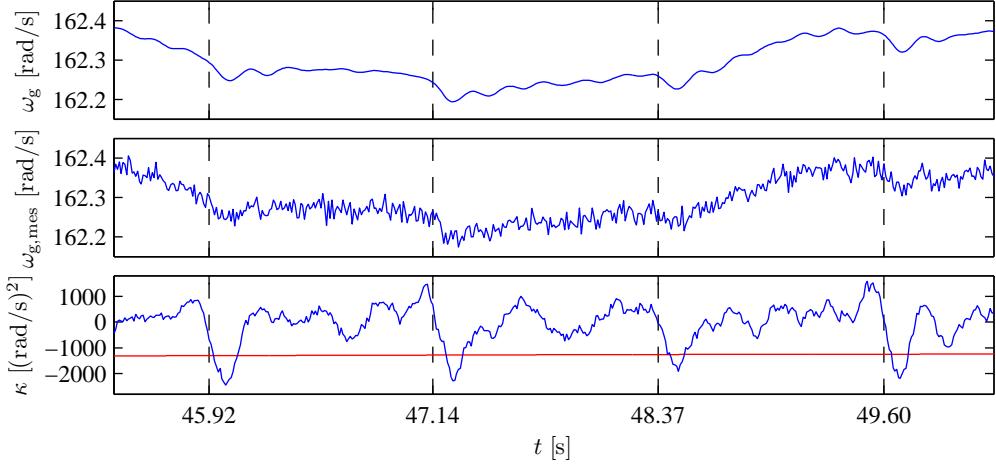


Figure 5.24: Peak detection on measured generator speed in full load operation. The lower subplot shows the energy operator (blue) and the threshold (red). The dashed vertical lines represent the instants when a blade passes the tower. Notice that propagation through the system makes the drop in the generator speed follow later than the drop in the resulting aerodynamic torque, which is the reason why the dashed lines and detected peaks do not coincide.

The detected blade passings can be used in the following way to estimate the proportional error: compare the angle obtained by integrating the measured generator speed to the angle obtained by exploiting that $\frac{2}{3}\pi$ rad separates the rotor angles at which two blades pass the tower. The following algorithm presents how to detect a peak and use it to achieve an estimate of the rotor angle:

0. Initialize $i = 1$, $j = 0$, $\hat{\theta}_r(j) = 0$ rad.
1. The $(j + 1)$ -th peak is expected to be located in the interval:
 $\tilde{\theta}_r(k_{j+1}) \in \hat{\theta}_r(k_j) + i [\frac{2}{3}\pi - \epsilon_l \quad \frac{2}{3}\pi + \epsilon_u]$, where k_j is the time at which the j -th peak is detected while ϵ_l and ϵ_u are small fractions of $\frac{2}{3}\pi$ rad.
2. If $\kappa(k)$ is below the threshold within the interval $\tilde{\theta}_r(k_{j+1})$ then:
 A blade has passed the tower. The angle at which the minimum value of $\kappa(k)$ occurs is denoted $\hat{\theta}_r(k_{j+1})$ and is set to $\hat{\theta}_r(k_{j+1}) = \hat{\theta}_r(k_j) + i \frac{2}{3}\pi$. Set $i = 0$.
3. Increment i and j . Go to 1.

The procedure described above provides an estimate of $\theta_r(k)$, which is updated each time a blade's passage of the tower is detected. In the samples where no peaks are detected, the estimate is calculated based on an integration of $\omega_{g,mes}(k)$:

$$\hat{\theta}_r(k+1) = \hat{\theta}_r(k) + \frac{\omega_{g,mes}(k)}{N_g(1 + \hat{\alpha}_{pe}(k))} T_s \quad [\text{rad}] \quad (5.63)$$

By utilizing the estimated rotor angle, $\hat{\theta}_r(k)$, and the measured generator speed, $\omega_{g,mes}(k)$, as input to an extended Kalman filter, the following system description enables the EKF to estimate the proportional error, $\alpha_{pe}(k)$. Note that the system equations are presented in continuous time for

convenience, but that they are discretized when implemented.

$$\begin{aligned}\dot{x}(t) &= Ax(t) \\ \begin{bmatrix} \dot{\theta}_r(t) \\ \dot{\omega}_g(t) \\ \dot{\alpha}_{pe}(t) \end{bmatrix} &= \begin{bmatrix} 0 & \frac{1}{N_g} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_r(t) \\ \omega_g(t) \\ \alpha_{pe}(t) \end{bmatrix}\end{aligned}\quad (5.64a)$$

$$\begin{aligned}y(t) &= h(x(t)) \\ \begin{bmatrix} \hat{\theta}_r(t) \\ \omega_{g,mes}(t) \end{bmatrix} &= \begin{bmatrix} \theta_r(t) \\ (1 + \alpha_{pe}(t))\omega_g(t) \end{bmatrix}\end{aligned}\quad (5.64b)$$

The method described in this subsection is based on feature extraction, whereas the method described in the next subsection is based on a model of an oscillator.

Estimation based on Finding 3P Frequency using Oscillator Model

The procedure of the method is illustrated in Figure 5.25, where a high-pass filter removes the slow-varying trends from the measurement of the generator speed allowing the 3P frequency to be extracted from this signal. The proportional error is then estimated by combining the measurement of the generator speed with the estimated 3P frequency.

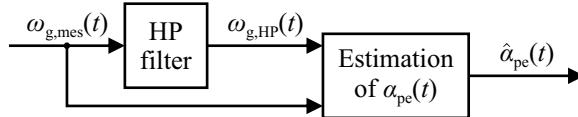


Figure 5.25: Block diagram of the estimation of the proportional error, $\alpha_{pe}(t)$, based on the measured generator speed, $\omega_{g,mes}(t)$.

The filter is a discretized version of the filter shown below, where $\omega_c = 1$ rad/s.

$$H_{HPF}(s) = \frac{s + \omega_c/1000}{s + \omega_c} \quad \left[\frac{\text{rad/s}}{\text{rad/s}} \right] \quad (5.65)$$

where:

$H_{HPF}(s)$ is the high-pass filter extracting the 3P frequency of the measured generator speed [·]

After $\omega_{g,mes}(t)$ has been high-pass filtered, the 3P frequency described by $\cos(3\theta_r(t))$ must be extracted from $\omega_{g,HP}(t)$. This is accomplished using an extended Kalman filter which is based on an oscillator model, having a frequency dependent of $\omega_{g,mes}(t)$ and $\alpha_{pe}(t)$, combined with a slow varying term, $c(t)$. The model of the high-pass filtered generator speed measurement is:

$$\omega_{g,HP}(t) = \cos \left(3 \frac{\theta_{g,mes}(t)}{N_g(1 + \alpha_{pe}(t))} \right) + c(t) \quad [\text{rad/s}] \quad (5.66)$$

where:

$\omega_{g,HP}(t)$ is the high-pass filtered generator speed containing the 3P frequency component [rad/s]

From this equation the state space representation shown in Eq. (5.67) can be set up. Here, the first two states are used to model the 3P frequency, the third state is the inverse of the sensor gain, and

the forth state captures the remaining slow variations in the filtered signal.

$$\dot{x}(t) = f(x(t))$$

$$\dot{x}(t) = \begin{bmatrix} x_2(t) \\ -\left(\frac{3\omega_{g,\text{mes}}(t)x_3(t)}{N_g}\right)^2 x_1(t) \\ 0 \\ 0 \end{bmatrix}, \quad x(t) = \begin{bmatrix} \cos(3\theta_r(t)) \\ -3\omega_r(t)\sin(3\theta_r(t)) \\ \frac{1}{1+\alpha_{pe}(t)} \\ c(t) \end{bmatrix} \quad (5.67a)$$

$$\omega_{g,\text{HP}}(t) = x_1(t) + x_4(t) \quad [\text{rad/s}] \quad (5.67b)$$

From $f(x(t))$ the Jacobian, which is used to propagate the covariances in the filter, can be approximated as shown below.

$$\frac{df}{dx} = \begin{bmatrix} 0 & 1 & \frac{x_2(t)}{x_3(t)} & 0 \\ -\left(\frac{3\omega_{g,\text{mes}}(t)x_3(t)}{N_g}\right)^2 & 0 & -2\left(\frac{3\omega_{g,\text{mes}}(t)}{N_g}\right)^2 x_3(t)x_1(t) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.68)$$

In order to tune the estimator, the term $c(t)$ should vary fast enough to remove all but the 3P frequency, and the cosine function must equal $\cos(3\theta_r(t))$ by changing $\alpha_{pe}(t)$ slowly towards the correct value.

Since two methods have now been described, the purpose of the next subsection is to verify and compare the two methods for estimating the proportional speed error.

Verification of Proportional Speed Error Estimation

To verify the estimation of the proportional speed error, both presented methods are evaluated using an identical test scenario, where a proportional error of -10% of the sensor gain is injected in a simulation during half an hour. In order to investigate the influence of the varying generator speed at partial load operation, a test is conducted for a mean wind speed of 9 m/s. In order to evaluate the performance in the full load region, another test is performed for a mean wind speed of 16 m/s.

Monte Carlo simulations of 100 runs are performed at each wind speed using both algorithms. In each situation the correct value of the proportional gain (red) is plotted in combination with the mean value of the estimate (blue) plus and minus two times the standard deviation of the estimate (green).

Results for Partial Load Operation

In Figure 5.26 simulation results are provided for the estimation of the proportional error in the partial load region. It is obvious from the figure that the two methods perform very different at this wind speed. The first method supplies an estimate which is a lot faster than that of the second method, since the update of the proportional error in the second method is affected by the lower rotational speed in the partial load region. Furthermore, the results of the simulation display the weakness of the first method, which is obvious from the noisy and biased estimate at the end of the simulation. This indicates that a combination of the two methods would be preferable for exploiting both their advantages.

The weakness of the first method emerges since the peaks become more difficult to detect when the wind speed decreases, due to the decreasing signature of the peaks on the generator speed. The problem exists since it is not certain that the peak detection algorithm will lock onto the blade passings after having missed some peaks. This makes the method inappropriate for low wind speeds. However, this is not a problem shared by the second method, where the estimate gradually converges towards the true value.

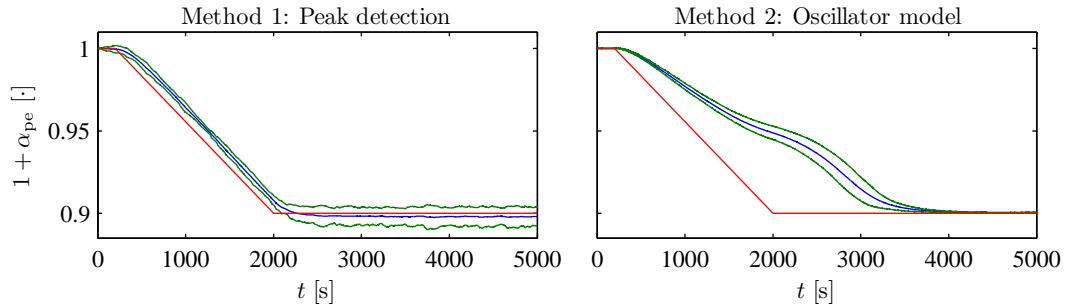


Figure 5.26: Simulation results showing the estimation of the speed error for a mean wind speed of 9 m/s. Between $t = 200$ s and $t = 2,000$ s a proportional error of -10% is introduced. The correct values of the proportional gain (red) are plotted in combination with the mean values of the estimates (blue) plus and minus two times the standard deviations of the estimates (green).

Results for Full Load Operation

In Figure 5.27 simulation results are provided for the estimation of the proportional error in the full load region. It can be concluded that both methods perform satisfactory, although the first method, which uses the peak detection approach, provides the least noisy estimate. This is caused by its superior performance when all blade passings of the tower are detected, since the estimated rotor angle is then almost exact at all times.

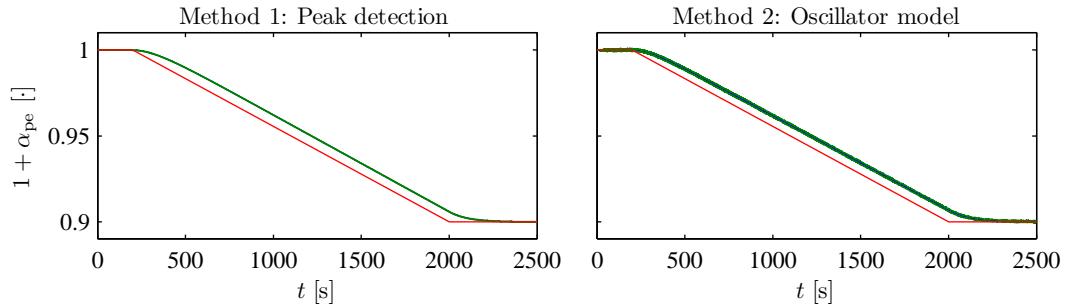


Figure 5.27: Simulation results showing the estimation of the speed error for a mean wind speed of 16 m/s. Between $t = 200$ s and $t = 2,000$ s a proportional error of -10% is introduced. The correct values of the proportional gain (red) are plotted in combination with the mean values of the estimates (blue) plus and minus two times the standard deviations of the estimates (green).

In this section redundancy has been exploited to detect a proportional speed error on the generator speed measurement, and two methods have been presented for estimating the proportional error. Both methods exploit detection of a blade passing the tower and are based on signal analysis and a model-based approach, respectively. Due to the higher reliability of the second method, it is recommended to select this method for implementation, or to exploit the advantages of both methods by combining these in a more advanced set-up. The accommodation of the proportional speed error based on the second method is described in Section 6.3.

In addition to the risk of getting a proportional speed error, this thesis also focus on the possibility of getting fixed or no measurements from the generator speed sensor. An algorithm for detecting fixed output from the generator speed sensor is described in the next section.

5.9 Detection of Fixed Output from Generator Speed Sensor

In this section it is described how a fixed output from a sensor is detected. Based on the selection of faults in Table 4.9 on Page 44 this specifically concerns the generator speed sensor, but the method applies to any sensor that may provide a fixed output.

To determine from a probabilistic point of view if the output from a sensor is fixed, the current output from the sensor should be compared to previous samples. The probability that a sensor outputs the same value in a number of consecutive samples depends on the quantization step size and the standard deviation of the assumed zero-mean Gaussian distributed measurement noise.

An equation is derived for calculating the mean time between getting a certain number of identical values in consecutive samples. This equation is used to determine the probability of false detection in different situations. First of all, a Gaussian probability density function is defined:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (5.69)$$

where:

$f(x)$ is the probability density function of a Gaussian distribution

The probability of receiving the same measurement from the sensor in k consecutive samples can be calculated from Eq. (5.70). It is assumed that the deterministic values of the measured variable are identical during the k samples, since this provides the worst-case probability.

$P\{\text{identical outputs in } k \text{ consecutive samples}\}$

$$= \sum_{j=-\infty}^{\infty} \left(F\left(\mu + \frac{x_{\text{quant}}}{2}(-1+2j) < x < \mu + \frac{x_{\text{quant}}}{2}(1+2j)\right) \right)^k \quad (5.70)$$

where:

$F(*)$ is the cumulative distribution function of a Gaussian distribution

x_{quant} is the quantization step size of the variable x quantized with N bits

The mean time between false alarms, stated in years per fault, appear in Table 5.5 as function of the quantization level and the number of samples used in the detection. The number of samples used in the detection should be selected such that this many identical, consecutive samples do seldom occur in the fault-free case, as this will provide a low rate of false alarms. As depicted by the table, the mean time between getting identical, consecutive samples depends heavily on the quantification.

Quantization	No. consecutive samples		2 samples	3 samples	4 samples	5 samples
	16 bit	24 bit	$6 \cdot 10^{-9}$	$93 \cdot 10^{-9}$	$1.4 \cdot 10^{-6}$	$20 \cdot 10^{-6}$
32 bit			$1.5 \cdot 10^{-6}$	$6 \cdot 10^{-3}$	$23 \cdot 10^0$	$87 \cdot 10^3$
			$381 \cdot 10^{-6}$	$398 \cdot 10^0$	$391 \cdot 10^6$	$372 \cdot 10^{12}$

Table 5.5: Mean time (in years) between getting a certain number of identical, consecutive outputs from the generator speed sensor when using different quantization. The following numbers are used in the calculations: a standard deviation of the zero-mean Gaussian distributed measurement noise $\sigma = 15.8 \cdot 10^{-3}$ rad/s and a domain [0 rad/s, 200 rad/s]. Notice that the sample rate is 100 Hz.

In this project quantization of the measurements have not received much attention, and have been quantized using 32 bit. Therefore, it can be seen from Table 5.5 that using three samples in the detection of a fixed output causes a low probability of false detections.

In this chapter the diagnosis algorithms for the considered faults have been designed, using the methods outlined in Table 4.9 on Page 44. The considered faults include both abrupt faults and incipient faults; hence, both hypothesis testing and parameter estimation methods have been exploited. All diagnosis algorithms are designed according to the overall structure shown in Figure 5.1 on Page 45, to be able to provide a fault-corrected state estimate at all times, and to ensure that multiple simultaneous faults can be diagnosed. Additionally, the diagnosis algorithms have been verified using Monte Carlo simulations to ensure that they are robust towards changing operating conditions.

In the next chapter the fault-tolerant control algorithms based on signal correction are presented.

Fault-Tolerant Control by Signal Correction

The purpose of this chapter is to present fault-tolerant control algorithms, which are based on signal correction. By signal correction it is meant that the control system is not modified; only the inputs and outputs of the controller are corrected according to the estimated faults. All fault-tolerant control algorithms explained in this chapter rely on the fault diagnosis design in Chapter 5. The faults accommodated using signal correction are listed in Table 4.9 on Page 44 and are: single pitch sensor bias, multiple pitch sensor biases, proportional error of generator speed sensor, and fixed output from generator speed sensor.

An important property of fault-tolerant control algorithms based on signal correction is that they can be implemented independent of the structure of the nominal controller, and without affecting the nominal performance of the controller. For the incipient faults treated in this chapter, it is assumed that the dynamics of the fault-tolerant control algorithms are much slower than the dynamics of the system. Therefore, they do not introduce any stability issues when applied.

The structure of the fault-tolerant control system is shown in Figure 6.1. As shown in the figure the nominal controller is not modified itself, only its inputs and outputs are modified. Notice that the supervisor is not described explicitly, as it is very simple in the considered case.

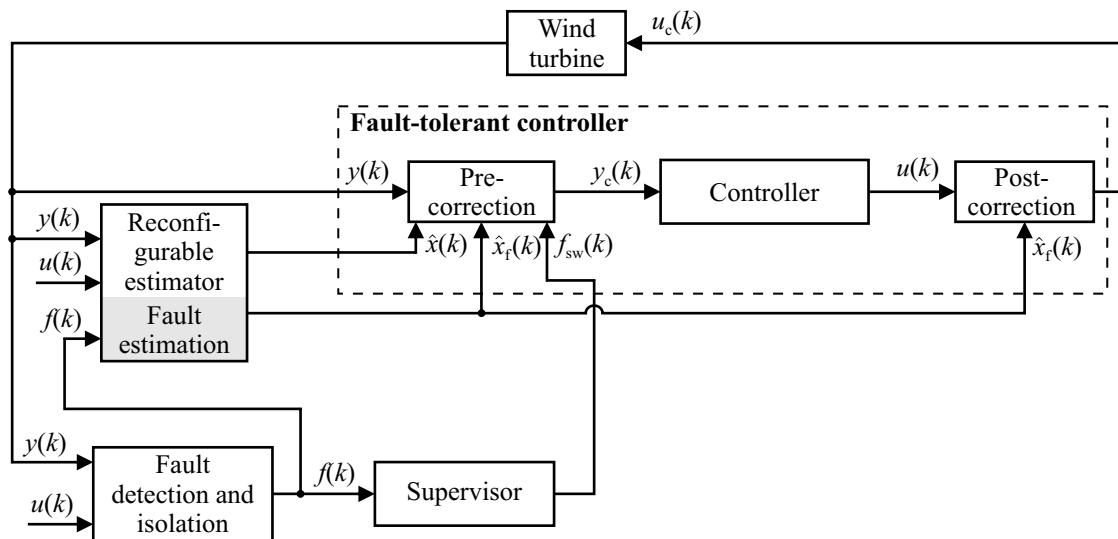


Figure 6.1: Block diagram of the fault-tolerant control system using signal correction based on the fault diagnosis algorithms. The supervisor decides if estimates or measurements are used as input to the controller.

In the next section the accommodation of a single biased pitch sensor is described.

6.1 Accommodation of Single Pitch Sensor Bias

The purpose of this section is to explain how a bias on a single pitch sensor is accommodated, by exploiting the estimated bias angle available from the fault diagnosis algorithm presented in Section 5.3. Finally, the accommodation is verified by use of Monte Carlo simulations for different wind speeds. Note that during all tests the fault detection signal to the controller is injected manually to reduce the simulation time.

Accommodation Design for Single Pitch Sensor Bias

The accommodation of a biased pitch sensor measurement consists of adding the estimated bias to the pitch reference signal. This signal is fed to the pitch actuator as shown in Figure 6.2. The accommodation is performed by manipulating the reference signal, since it is not possible to subtract the bias from the measured pitch angle used in the pitch controller, as the pitch system is a closed system. This system cannot be altered since it may be delivered by an external supplier. Notice that the corrected pitch measurement should be fed to all estimators in the control system, to make these tolerant to this fault.

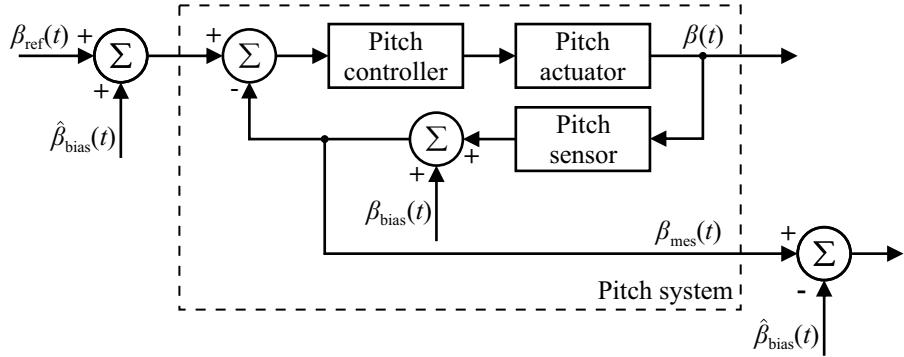


Figure 6.2: *Block diagram of the pitch system, which consists of the internally controlled pitch actuator. A bias on the pitch sensor measurement, $\beta_{bias}(t)$, can be compensated for by adding an estimate of the bias, $\hat{\beta}_{bias}(t)$, to the reference signal to the system and subtract it from the measurement.*

When the bias is estimated with adequate precision, the estimation of the bias should be disabled to enable estimation of another bias in the future. This is not implemented in this project, since the two parts are independent.

Verification for Single Pitch Sensor Bias

The accommodation of a single pitch sensor bias is verified by first conducting a single simulation run of a fault scenario both with and without applying the accommodation, to show the effect of the accommodation. Secondly, Monte Carlo simulations are conducted to show that the bias accommodation is robust towards changing operating conditions and that the accommodation enables the imbalance in the rotor plane to be minimized.

Single Run Simulation

To show the effect of accommodating the biased pitch sensor measurement, simulations are performed both with and without accommodating the fault. In the simulations a bias of 3° is introduced on the pitch sensor associated with Blade 3 during half an hour. By accommodating the fault, the difference between the pitch angles should be minimized, since a collective pitch strategy is utilized. The results of the simulations are shown in Figure 6.3.

From the figure it is obvious that the difference between the pitch angles is minimized by accommodating the biased pitch sensor measurement. This implies that the imbalance in the rotor plane is decreased. Thereby, the mechanical stress on the wind turbine is minimized.

Monte Carlo Simulations

The single run test presented above shows the functionality of the accommodation, but to verify the reliability and performance of the accommodation, 100 simulation runs are conducted at two different mean wind speeds, where a bias of 1° is introduced over half an hour on the pitch sensor associated with Blade 3. The results are presented in Figure 6.4.

The figure shows that the difference between the pitch angles of the blades is minimized in each test. The mean values of the differences end at 0.028° and -0.002° with standard deviations

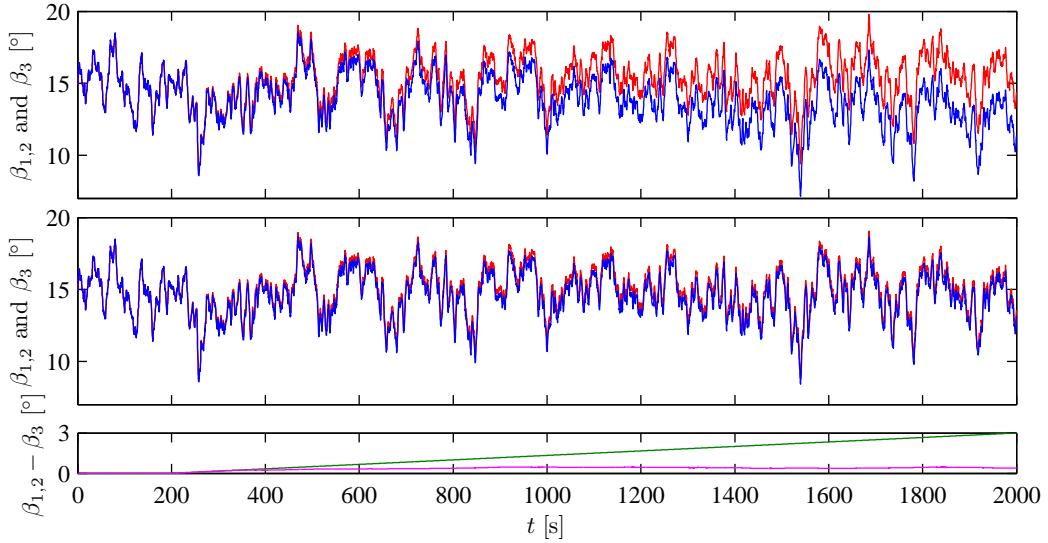


Figure 6.3: Simulation of the wind turbine for a mean wind speed of 20 m/s, where a 3° bias is introduced on the pitch sensor associated with Blade 3 between $t = 200$ s and $t = 2,000$ s. The first subplot shows the pitch angles when the fault is not accommodated. The second subplot shows the pitch angles when the fault is accommodated. The third subplot shows the difference between the pitch angles, when the fault is not accommodated (green) and when the fault is accommodated (magenta).

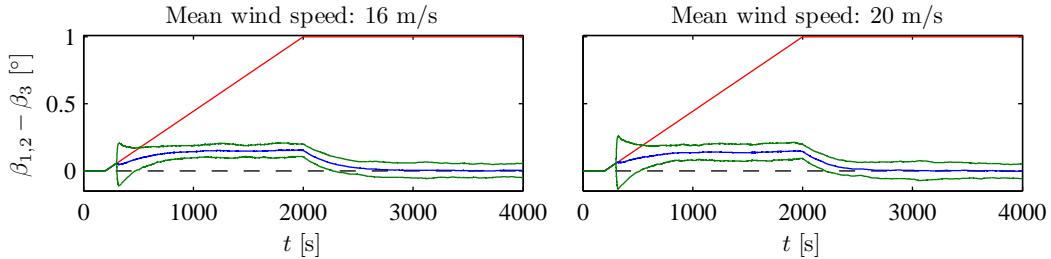


Figure 6.4: Results of Monte Carlo simulations, where a bias is introduced between $t = 200$ s and $t = 2,000$ s, introducing an expected pitch difference if accommodation was not applied (red). By applying the accommodation the mean values of the pitch differences (blue) lay near 0° throughout all the simulations. The green lines show the mean values plus and minus two times the standard deviations of the pitch differences.

of 0.027° and 0.028°, for the test at 16 m/s and 20 m/s. According to Section 4.6 the difference between the pitch angles of the blades cannot exceed 5°. This requirement is well satisfied even though the fault is introduced faster than specified. It is hereby concluded that the accommodation of a single pitch sensor bias performs satisfactory.

In this section a description of the accommodation of a single pitch sensor bias has been provided. Simulation results have shown that it is possible to accommodate the bias even though the imbalance in the rotor plane, from which the bias is estimated, is minimized by the accommodation itself. In the next section similar results are presented for multiple pitch sensor biases.

6.2 Accommodation of Multiple Pitch Sensor Biases

The purpose of this section is to describe how the accommodation of multiple simultaneous pitch sensor biases is designed and to verify its performance. Note that during all tests the fault detection signal to the controller is injected manually to minimize the simulation time.

Accommodation Design for Multiple Pitch Sensor Biases

The accommodation of multiple pitch sensor biases is identical to the accommodation of a single pitch sensor bias explained in Section 6.1, except that the accommodation is conducted on two sensors simultaneously. Additionally, the accommodation is not able to remove a common bias, but only to control the pitch angles of all three blades to a common value, possibly with a common offset as explained in Section 5.4.

Verification for Multiple Pitch Sensor Biases

In this subsection the accommodation design is verified by evaluating its ability to control all pitch angles to a common value, when multiple simultaneous pitch sensor biases exist. This is accomplished by conducting Monte Carlo simulations with two pitch sensor biases.

To get some statistical knowledge of the accommodation, 100 simulations are conducted with duration 4,000 s for this verification. The simulations are conducted for mean wind speeds of both 16 m/s and 20 m/s, where the biases on two pitch sensors are linearly introduced between $t = 200$ s and $t = 2,000$ s. The magnitudes of the introduced biases are 3° on the pitch sensor associated with Blade 2 and 2° on the sensor associated with Blade 3. Hereby, it is expected that the accommodation controls all blades towards having offsets of -2° in compliance with Table 5.4 on Page 63. The offset is adjusted towards minus the median of the three sensors biases to remove the rotor imbalance, where the negative sign originates from the fact that a positive sensor bias results in a negative pitch angle offset. To get an overview of the expected output of the simulations, consult Table 5.4 on Page 63.

The results of the simulations are shown in Figure 6.5, where the differences between the pitch angles are evaluated and compared to the differences, which would occur if no accommodation was applied.

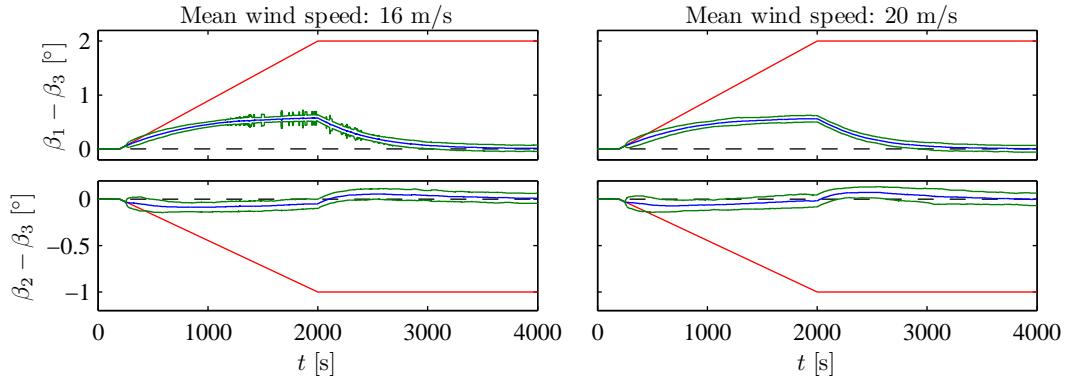


Figure 6.5: *Results of Monte Carlo simulations, where the biases are introduced between $t = 200$ s and $t = 2,000$ s, introducing expected pitch differences if accommodation was not applied (red). By applying the accommodation the mean values of the pitch differences (blue) lay near 0° throughout all the simulations. The green lines show the mean values plus and minus two times the standard deviations of the pitch differences.*

From Figure 6.5 it is obvious that the accommodation eliminates the difference between the blades, as the differences between the blades end by having mean values equal to 0.003° and 0.000° , with standard deviations of 0.031° and 0.034° . This implies that all blades have an offset of approximately -2° after the accommodation.

Generally, the fault accommodation eliminates unbalance in the rotor plane as requested, since all pitch angles become identical. However, some less serious effects still exist due to the common pitch offset.

The effect of having a common pitch angle offset in partial load operation is that the efficiency of the wind turbine is decreased; hence, the transition between partial load and full load operation happens at a higher wind speed. If the common pitch angle offset is negative, some additional issues are introduced in the transition, since the efficiency of the rotor is increased in a region where it is expected to decrease.

In this section a verification of the accommodation of multiple pitch sensor biases has been provided. Simulation results have shown that it is possible to accommodate the biases, but with a common pitch offset. In the next section the accommodation of a proportional error on the generator speed measurement is described.

6.3 Accommodation of Proportional Speed Error

The accommodation of a proportional speed error is based on the diagnosis algorithm described in Section 5.8, which allows the fault to be detected based on a consistency check and to be estimated from the 3P frequency afterwards. To estimate the size of the error, two methods have been presented in Section 5.8 that exploit the blade passing frequency. The accommodation described in this section utilizes the second of these methods, which is based on an oscillator model since it is the most reliable. This section provides a description and verification of the fault accommodation.

Accommodation Design for Proportional Speed Error

In order for the fault to be detected a test is performed which checks the consistency between the rotor speed measurement and the generator speed measurement, as described in Section 5.8. If these signals are not consistent, the accommodation is activated by a supervisor. The accommodation manipulates the generator speed measurement by dividing it with the estimated sensor gain. This implies that the resulting sensor gain ideally equals one.

Verification for Proportional Speed Error

In supplement to the estimation of the proportional speed error, being verified in Section 5.8, the accommodation of the fault introduces an additional feedback loop in the system, which is verified in this subsection.

The accommodation is verified by conducting Monte Carlo Simulations for mean wind speeds of 9 m/s and 16 m/s. In each situation 100 simulation runs are performed, where a proportional error of -10% of the sensor gain is injected during half an hour, starting at $t = 200$ s. This specification is identical to that used in the verification of the diagnosis algorithm.

Results for Partial Load Operation

The generator torque controller operating in the partial load region aims at keeping the tip-speed ratio at a certain value for maximizing the efficiency of the aerodynamics; hence, the tip-speed ratio is plotted in Figure 6.6. The average detection time is 68 s, so fault accommodation is initiated just after the fault is introduced. From the figure it is concluded that an unbiased tracking of the optimum tip-speed ratio is obtained when the fault size is eventually estimated. The situation when the fault is not accommodated corresponds to an average energy reduction of 0.7%, considering all 100 simulations.

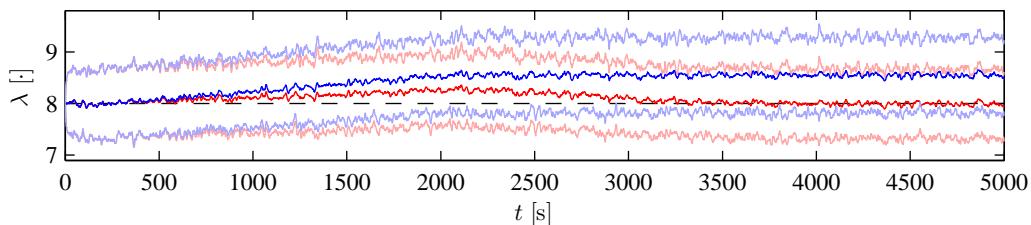


Figure 6.6: *Simulation results showing the accommodation of the proportional speed error for a mean wind speed of 9 m/s. The red lines represent the obtained tip-speed ratio when the fault is accommodated, while the blue lines represent the situation without accommodation of the fault. The optimum tip-speed ratio is marked by the dashed black line. Note that the mean values plus and minus two times the standard deviations of the tip-speed ratios are plotted using faded colors.*

Results for Full Load Operation

Since the objective of the speed controller operating in the full load region is to maintain a certain generator speed, this variable is plotted in Figure 6.7. It is obvious from the figure that the implemented fault-tolerant control system is able to diagnose and accommodate the fault, since the red line is almost kept at the set-point value marked by the dashed black line. Note that the energy production is not reduced in the full load operation, but overspeed may cause destructive damages to the wind turbine. From the figure it is clear that the generator speed does not exceed the safety margin, equal to 110% of the rated generator speed, when the accommodation is applied.

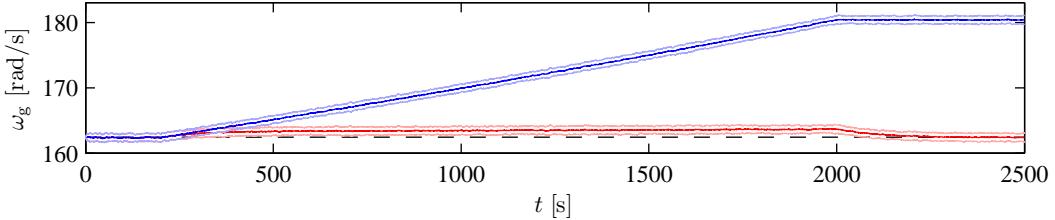


Figure 6.7: *Simulation results showing the accommodation of the proportional speed error for a mean wind speed of 16 m/s. The red lines represent the obtained generator speed when the fault is accommodated, while the blue lines represent the situation having no accommodation of the fault. The dashed black line represents the reference value to the controller. Note that the mean values plus and minus two times the standard deviations of the tip-speed ratios are plotted using faded colors.*

In this section accommodation of a proportional speed error has been discussed and verified. The fault is accommodated by correcting the measurement. The verification shows that the fault can be accommodated properly in both partial and full load operation. In the next section the accommodation of a fixed output from the generator speed sensor is addressed.

6.4 Accommodation of Fixed Output from Generator Speed Sensor

The accommodation of a fixed output from the generator speed sensor is triggered by the detection described in Section 5.9. The accommodation is implemented in a simple manner, because it takes advantage of the reconfigurable extended Kalman filter described in Section 5.2. This filter estimates the generator speed, which can be utilized as input to the controller replacing the fixed measurement. The method is explained and verified in this section.

It should be noticed that the signal correction approach is only applied to the controller in the partial load region, since the lost measurement has to be accounted for in the design of the controller operating in the full load region. Therefore, only a short discussion of the differences between the two regions is provided in this section, while the actual design of a fault-tolerant controller for the full load region is explained in Chapter 7.

Accommodation Design for Fixed Output from Generator Speed Sensor

The generator speed measurement is important to the control of the wind turbine, since it is used as input to the reference controller in both the partial and full load regions. The generator speed is observable, even when the generator speed measurement disappears as deduced from Section 4.4. This enables estimation of the generator speed at a degraded quality, due to the much noisier rotor speed measurement.

When the generator speed sensor outputs a fixed value, the estimated generator speed provided by the reconfigurable extended Kalman filter can be used as input to the controller instead of the measured generator speed. In the following bullet list the necessary modifications to the control system in the two control regions are described:

- **Partial Load Operation: Replace the measurement by an estimate**

In partial load operation the controller consists of a gain multiplied on the squared generator speed measurement; thus, it is sufficient to replace the measurement with the estimate of the generator speed. A timely estimate is not strictly necessary in the partial load region, since the dynamics of the control system is slow in this region.

- **Full Load Operation: Make a design accounting for the fault**

In full load operation it is not desired to operate the nominal speed controller using the estimated generator speed, since the degraded sensor information would seriously affect the performance of the controller. Therefore, another controller should be designed, which takes the absence of the measured generator speed into account in the design; this is discussed further in Chapter 7.

- **Overall**

- **Disable the drive train stress damper:** Common to the operation in the partial and full load regions is that the estimator is not able to provide an estimate including the drive train oscillations, since the rotor speed measurement does not possess these. Therefore, the drive train stress damper has to be deactivated when running the control system using the generator speed estimate, consequently increasing the drive train stress.
- **Introduce bumpless transfer:** Bumpless transfer must be introduced to avoid jumps in the control signals when reconfiguring the controller. This can be implemented similar to that described in Section C.4. Since the considered fault happens abruptly, it is not possible to make any smooth scheduling between the controllers.

The verification in the next subsection presents simulation results for the controller operating in the partial load region, while the results for the full load region is presented in Section 7.7.

Verification for Fixed Output from Generator Speed Sensor

In this subsection the performance of the fault-tolerant controller designed to handle a failed generator speed measurement is verified. This is done by evaluating the produced energy and the drive train stress for a single run of the simulation model in the partial load region, for a mean wind speed of 9 m/s. It is decided to show the simulation results for a single run to highlight the altered behavior of the controller in a desired manner.

In Figure 6.8 the blue lines illustrate the situation when using the measured generator speed, while the red lines represent the situation where the estimated generator speed is used. Note that the output of the generator speed sensor is fixed at $t = 100$ s, which allows the bumpless transfer mechanism to be evaluated in the same simulation, by studying the output of the controller in this transition point.

Since the generator torque controller aims at keeping the tip-speed ratio at a certain value, to maximize the efficiency of the aerodynamics, the tip-speed ratio is plotted in Figure 6.8 for both the normal and faulty cases.

The results in Figure 6.8 show that similar tracking of the optimal tip-speed ratio is achieved in the normal and faulty cases. The performance measures stated in Table 6.1 are based on simulations each having duration of 5,000 s, whereof Figure 6.8 shows only the first part. Note that the numbers in the table are normalized to the results obtained by the nominal controller.

Variable	With $\omega_{g,mes}(t)$	Without $\omega_{g,mes}(t)$
$\int \dot{\theta}_{\Delta}^2(t)dt$	1.00	4.80
$\int P_g(t)dt$	1.000	1.000

Table 6.1: Drive train stress and energy production obtained in the partial load region when using the generator speed measurement and when using an estimate replacing the unavailable measurement.

By comparing the second and third columns in the table it appears that the drive train stress is significantly increased when using the reconfigured controller. This is an expected consequence of

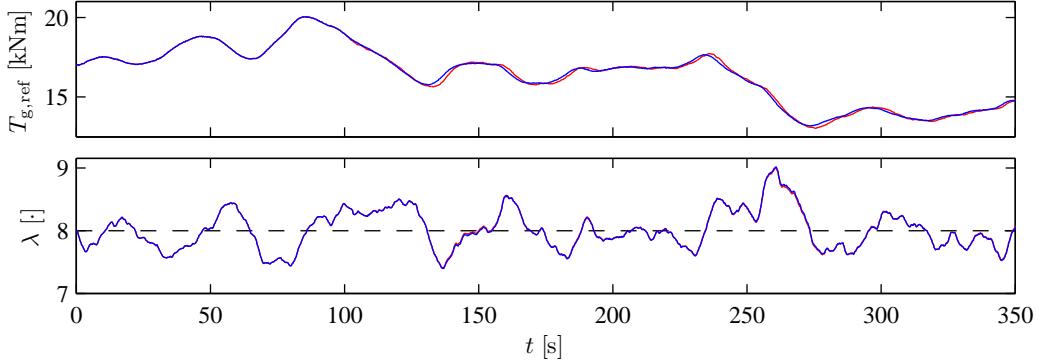


Figure 6.8: *Simulation results showing the accommodation of a fixed output of the generator speed sensor for a mean wind speed of 9 m/s. The blue lines illustrate the situation when using the measured generator speed, while the red lines correspond to the situation where the estimated generator speed is used, due to the absence of the measurement at $t = 100$ s. The optimum tip-speed ratio is marked by the dashed black line.*

losing the information about the drive train oscillations, as explained in the accommodation design section. This results in a shortened lifetime of the drive train and therefore it should be determined whether continued operation in the faulty case compensates for the higher operational costs.

In this section it has been described how to replace a fixed output of the generator speed sensor by an estimate provided by the reconfigured extended Kalman filter. The cost of losing the generator speed measurement results in increased drive train stress in both control regions, however the energy capture in the partial load region is not compromised. The effects seen when operated in the full load region is evaluated in Chapter 7, where a controller accounting for this fault is designed and verified.

This ends the design of the signal correction algorithms, which exploit the information provided by the fault diagnosis algorithms to obtain a fault-tolerant control system. Four different signal correction algorithms have been designed and successfully verified. All presented algorithms can be applied independent of the controller structure, since they only correct the measurements and the reference signals.

In the next chapter, active and passive fault-tolerant control systems are designed for accommodating the faults, which either change the dynamics of the system or reduce the available measurement information.

Fault-Tolerant Control by LPV Methods



The purpose of this chapter is to design and compare active and passive fault-tolerant control systems, capable of handling the faults which have to be taken into account in the controller design. These faults are: changed dynamics of the pitch system and loss of the generator speed measurement in full load operation. The comparison of the two approaches covers both the performance of the controllers and the complexity of their design procedures.

Changing dynamics of the pitch system cannot be accommodated by signal correction; hence, it should be considered in the controller design, to guarantee stability and a satisfactory performance. Among the possible causes for changed dynamics of the pitch system, it is chosen only to introduce fault-tolerance towards changing air content in the hydraulic oil. This is chosen since this fault is considered to be the most likely, as shown in Table 4.3 on Page 33, and since the reference controller becomes unstable when the hydraulic oil has a high air content, as shown in Table A.3 on Page 135. The fault-tolerant control systems are though easily extended to include the remaining faults causing changed dynamics of the pitch system.

When the generator speed measurement is unavailable, the controller should rely on the measurement of the rotor speed, which is contaminated with much more noise than the generator speed measurement. This makes it necessary to reconfigure the controller to obtain a reasonable performance of the control system.

To summarize, the purpose of this chapter is to design four fault-tolerant controllers according to the following list. It is sufficient to design either the active or passive fault-tolerant controllers for accommodating the faults, but they are both included in this chapter to enable a comparison of the active and passive approaches.

1. Active fault-tolerant controller relying on the generator speed measurement, which is tolerant towards changing air content in the oil of the pitch system.
2. Active fault-tolerant controller independent of the generator speed measurement, which is tolerant towards changing air content in the oil of the pitch system.
3. Passive fault-tolerant controller relying on the generator speed measurement, which is tolerant towards changing air content in the oil of the pitch system.
4. Passive fault-tolerant controller independent of the generator speed measurement, which is tolerant towards changing air content in the oil of the pitch system.

In the next section the structure of the two types of control systems are described and the methods used in the controller designs are selected.

7.1 Introduction to Active and Passive Fault-Tolerant LPV Control Methods

The purpose of this section is to discuss the differences between active and passive fault-tolerant control systems and to address how they fit to the considered application. Finally, the methods which are utilized in the controller designs are selected.

Structures of Active and Passive Fault-Tolerant Control Systems

To conduct a fair comparison of the controllers designed using active and passive fault-tolerant design approaches, these controllers must build upon identical controller descriptions in the fault-free case. Hence, any difference in controller performance or design complexity should be caused by the active or passive fault-tolerant nature, rather than the underlying controller descriptions.

Furthermore, the controllers must comply with the parameter-varying nature of the wind turbine along its nominal operating trajectory caused by non-linearities in the aerodynamics. To comply with these requirements it is chosen to design Linear Parameter-Varying (LPV) controllers for both the active and the passive fault-tolerant approaches. Notice that only operation in the full load region is considered, since the pitch system is only active in this region.

The two fault-tolerant control systems have different structures as shown in Figure 7.1, since only the active fault-tolerant controller relies on a fault diagnosis algorithm. This is the main difference between the two control systems.

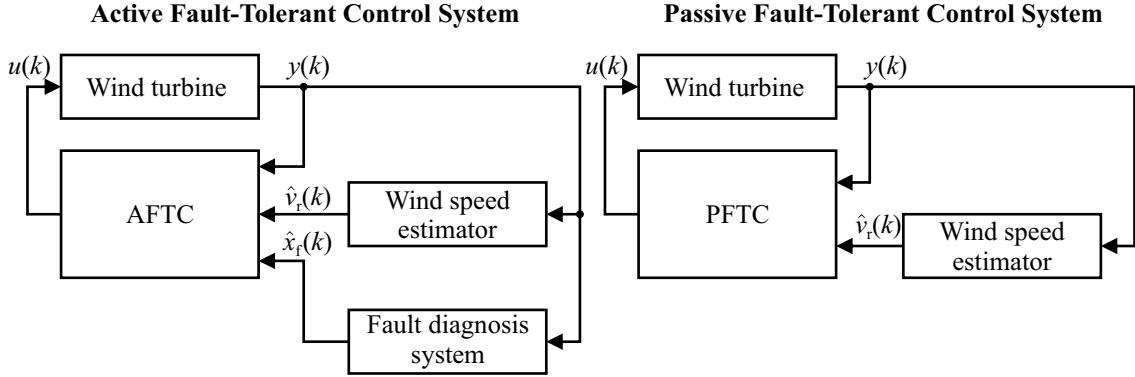


Figure 7.1: *Structures of the active and passive fault-tolerant control systems. The passive fault-tolerant controller (PFTC) only relies on the measured system variables and an estimated wind speed, while the active fault-tolerant controller (AFTC) also relies on information from the fault diagnosis system.*

The difference between an AFTC and a PFTC is that an active fault-tolerant controller relies on a fault diagnosis system, which should feed information about the faults to the controller. In the considered case the fault diagnosis system contains the estimation of the pitch system dynamics designed in Section 5.5. The knowledge of the fault makes it possible for the AFTC to reconfigure to the current state of the system, but it also introduces a risk of false positive and false negative diagnosis, due to e.g. model errors.

The PFTC does not rely on a fault diagnosis algorithm, but is instead designed to be robust towards the fault. This is accomplished by designing a controller which is optimized for the fault-free situation, while satisfying some degraded requirements in the fault scenario. The degradation of requirements is what differs robust controllers from reliable controllers, which has the same performance guarantee in the entire parameter space. This implies that the PFTC has no risk of making false detections or reconfigurations, and it has no detection time either. Conversely, it is not necessarily optimal at any given time, since some conservatism is introduced in its design.

Structures of Fault-Tolerant LPV Control Systems

The two faults which are considered in this chapter are very different, since changing air content is an incipient fault happening frequently, while the loss of the generator speed measurement is an abrupt fault, which happens very rarely. Therefore, it is chosen to design two independent fault-tolerant controllers capable of handling changing air content. The first of the controllers assumes that the generator speed measurement is available, while the second controller assumes that it is not. This choice is made to avoid degrading the performance of the nominal controller, by incorporating the loss of the generator speed measurement directly in the controller structure. The transition between the controllers is considered not to threaten the stability of the system, since it is assumed to happen very rarely. Both the active and passive approaches are adopted to this structure; resulting in a total of four controllers combined in two control systems.

Both the active and passive fault-tolerant controllers are able to utilize the signal correction algorithms explained in the previous section; hence, multiple simultaneous faults can be handled. The combined structure of the active and passive fault-tolerant controllers is shown in Figure 7.2. Notice that the passive fault-tolerant controllers do not use the fault estimate, $\hat{\theta}_f(k)$.

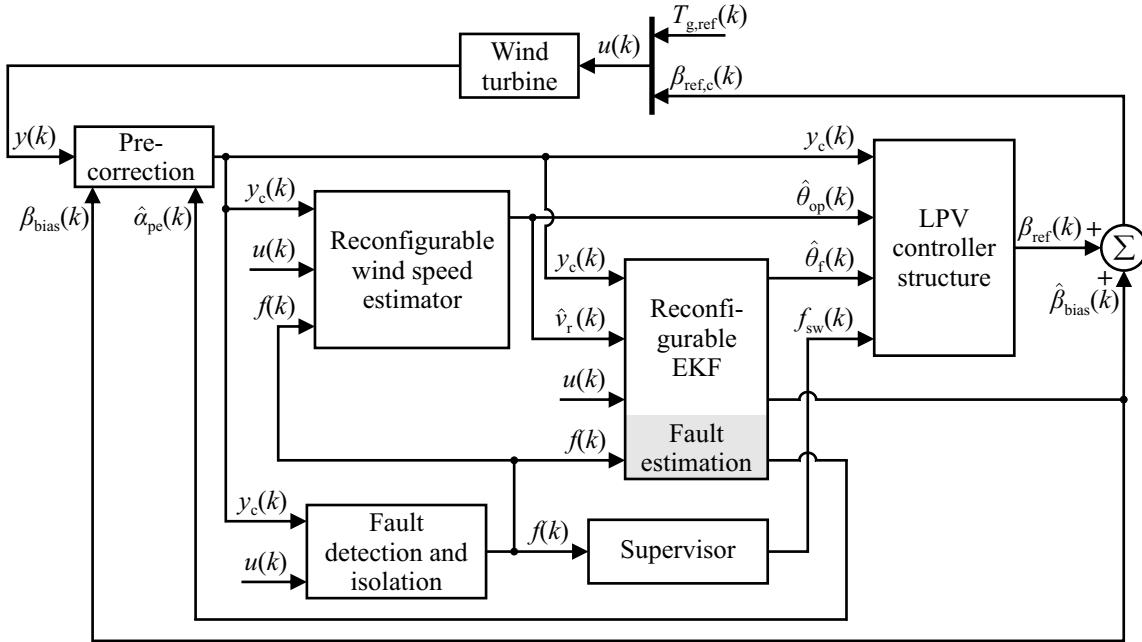


Figure 7.2: Block diagram of the LPV-based fault-tolerant control system combined with the signal correction algorithms. The LPV controller structure consists of two fault-tolerant controller; one with and one without the generator speed measurement. The vector $y_c(k)$ contains the measurements corrected by the signal correcting algorithms, while $\theta_{op}(k)$ and $\theta_f(k)$ are scheduling parameters. The supervisor reconfigures the controller using $f_{sw}(k)$.

Methods for Active and Passive Fault-Tolerant Controller Designs

The methods used in the fault-tolerant controller designs should rely on output feedback, since only part of the state vector is measured. Additionally, they should take measurement noise into account. Finally, the design methods should be suited for systems with varying parameters, since the wind turbine model is a parameter-varying system.

It is chosen to base both the active and the passive fault-tolerant controllers on a common LPV controller description, to which the fault-tolerance can be added, since LPV control methods are able to provide stability and performance guarantees despite the parameter variations. Additionally, LPV controller design methods are well proven in multiple applications including wind turbines [Bianchi *et al.*, 2007].

To add fault-tolerance to the common LPV controller formulation, two different approaches are utilized. For the AFTCs the LPV structure can be directly utilized by letting the parameters estimated by the fault diagnosis algorithm be used for scheduling the controllers. In contrast, a different structure should be used to obtain fault-tolerance in the PFTCs. For this purpose the group has extended the LPV design method described in [Bianchi *et al.*, 2007, pp. 159-179] to apply for systems with parametric uncertainties, which can be solved by applying an approach based on the projection lemma utilized in [Jabbari, 1997]. This is derived in Appendix D on Page 161. Alternatively, other methods could have been used such as [Köse and Jabbari, 1997], which is a similar LPV method, or [Niemann and Stoustrup, 2005a], which preserves the nominal performance. However, these methods are not used as their controller design procedures are very different from the procedure utilized for designing the active-fault tolerant controller, which would make the comparison unfair.

Generally, both design methods rely on solving some optimization problems where a controller is calculated subject to maximizing the disturbance attenuation. These problems are formulated as Linear Matrix Inequalities (LMIs) and are set up in YALMIP and solved using SeDuMi. For decreasing numerical problems the MATLAB function `ssbal` is used to reduce the condition numbers of the matrices, which are inserted into the LMIs.

Chapter Overview

This chapter covers a variety of topics. First, the two design procedures are described, then the system equations are adapted to fit the descriptions utilized in the optimization problems. Finally, the design methods are applied to the considered system to form active and passive fault-tolerant controllers, after which the control systems are compared.

- **Active Fault-Tolerant LPV Controller Design Procedure:** Describe the design procedure for obtaining an active fault-tolerant LPV controller.
- **Passive Fault-Tolerant LPV Controller Design Procedure:** Describe the design procedure for obtaining a passive fault-tolerant LPV controller.
- **Common LPV System Description:** Set up an affine LPV description of the system and include a performance specification in the optimization problem.
- **Active Fault-Tolerant Controller Design:** Design the active fault-tolerant controllers and provide the verification of the controllers.
- **Passive Fault-Tolerant Controller Design:** Design the passive fault-tolerant controllers and provide the verification of the controllers.
- **Comparison of the AFTCS and the PFTCS:** Conduct a comparison between the design methods and the performances of the designed control systems.

The structures of the controllers and the content of this chapter have been outlined. In the next section the active fault-tolerant LPV controller design procedure is explained.

7.2 Active Fault-Tolerant LPV Controller Design Procedure

The purpose of this section is to describe the LPV controller design procedure for the active fault-tolerant controller. This design is identical to an ordinary LPV controller design; it is due to the origin of the scheduling parameters that it is called an active fault-tolerant controller design. This section is inspired by [Bianchi *et al.*, 2007, pp. 159-179].

Active Fault-Tolerant LPV System Description

The purpose of this subsection is to describe the open-loop LPV system, which forms the background for the LPV controller design. This is accomplished by specifying a state space model of the system and characterizing the parameter variations of the system.

A general state space model of an open-loop LPV system is shown below, where all matrices in this system are parameter-varying.

$$\dot{x}(t) = A(\theta(t))x(t) + B_1(\theta(t))w(t) + B_2(\theta(t))u(t) \quad (7.1a)$$

$$z(t) = C_1(\theta(t))x(t) + D_{11}(\theta(t))w(t) + D_{12}(\theta(t))u(t) \quad (7.1b)$$

$$y(t) = C_2(\theta(t))x(t) + D_{21}(\theta(t))w(t) + D_{22}(\theta(t))u(t) \quad (7.1c)$$

where:

$z(t)$ is the performance output vector

$\theta(t)$ is the parameter vector

To ease the design of the LPV controller, some matrices in the general LPV system description are assumed to be constant. Additionally, the parameter variations in the considered system can be described in an affine manner according to Section 7.4. This further simplifies the design and synthesis of the LPV controller. These restrictions fit the controller design method explained in [Bianchi *et al.*, 2007, p. 174], and give a system description as shown below.

$$\begin{bmatrix} A(\theta) & B_1(\theta) & B_2(\theta) \\ C_1(\theta) & D_{11}(\theta) & D_{12}(\theta) \\ C_2(\theta) & D_{21}(\theta) & 0 \end{bmatrix} = \begin{bmatrix} A_0 & B_{1,0} & B_{2,0} \\ C_{1,0} & D_{11,0} & D_{12,0} \\ C_{2,0} & D_{21,0} & 0 \end{bmatrix} + \sum_{i=1}^{n_\theta} \theta_i \begin{bmatrix} A_i & B_{1,i} & 0 \\ C_{1,i} & D_{11,i} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.2)$$

where:

n_θ is the number of varying parameters

The restriction of parameter independent matrices B_2, C_2, D_{12} , and D_{21} can be applied without any loss of generality, as it is possible to pre-filter the control input $u(t)$ to convert B_2 and D_{12} into constant matrices, whereas post-filtering can be applied if C_2 and D_{21} are parameter-dependent [Apkarian *et al.*, 1995]. Furthermore, the assumption $D_{22} = 0$ can be relaxed by redefining the output.

Active Fault-Tolerant LPV Controller Optimization Problem

The purpose of this subsection is to set up the optimization problem from which an LPV controller can be designed. The optimization problem should fit an LPV system defined using the affine system description presented in Eq. (7.2).

The design procedure described in this section aims at designing a dynamic output feedback LPV controller on the following form:

$$\dot{x}_c(t) = A_c(\theta(t))x_c(t) + B_c(\theta(t))y(t) \quad (7.3a)$$

$$u(t) = C_c(\theta(t))x_c(t) + D_c(\theta(t))y(t) \quad (7.3b)$$

where:

$A_c(\theta(t)), B_c(\theta(t)), C_c(\theta(t)), D_c(\theta(t))$ are the LPV controller matrices

The obtained gain-scheduled output-feedback controller should enforce internal stability and a bound on the \mathcal{L}_2 -norm from the disturbance input, $w(t)$, to a chosen performance output, $z(t)$, i.e. $\|z(t)\|_2 < \gamma \|w(t)\|_2$ for all non-zero inputs with finite energy assuming zero initial conditions.

The described method requires that the parameter variations are affine. This requirement fits with the parameter variations which exist in the considered system as explained in Section 7.4. The affine parameter description simplifies the design procedure compared to the general case, by decreasing the number of LMIs involved in the calculation of the LPV controller. To give an understanding of the origin of the LMIs utilized in the controller design, the LMIs for the general case are set up first. Secondly, the LMIs for an affine parameter description are set up.

General Case

The LMIs that must be solved in the LPV controller design originates from the bounded real lemma, which can be expressed as an LMI having the closed-loop LPV system matrices as unknown parameters. This inequality is not linear in the unknown LPV controller matrices; hence, a congruence transformation is performed to obtain a description which is linear in the unknown variables. This LMI description does not include the controller matrices A_c, B_c, C_c , and D_c directly, but some other matrices shown in Eq. (7.4), which make the controller design a convex optimization problem. Notice that '($\theta(t)$)' is removed from the expressions to shorten the notation. Additionally, the matrices which should be found in the design problem are written using bold face to make it easier to distinguish between known and unknown variables.

$$\hat{\mathbf{A}} = NA_cM^T - \mathbf{X}\dot{\mathbf{Y}} - NM^T + \mathbf{X}(A + B_2D_cC_2)\mathbf{Y} + \mathbf{X}B_2C_cM^T + NB_cC_2\mathbf{Y} \quad (7.4a)$$

$$\hat{\mathbf{B}} = NB_c + \mathbf{X}B_2D_c \quad (7.4b)$$

$$\hat{\mathbf{C}} = C_cM^T + D_cC_2\mathbf{Y} \quad (7.4c)$$

$$\hat{\mathbf{D}} = D_c \quad (7.4d)$$

The congruence transformation and the substitution of the matrices above make it possible to write the following LMIs, which gives a stabilizing controller with performance level γ , where $\mathbf{X}, \mathbf{Y}, \hat{\mathbf{A}}$,

$\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{D}}$ are unknown and dependent on $\theta(t)$:

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}C_2 + (*) & * & * & * \\ \hat{\mathbf{A}}^T + A + B_2\hat{\mathbf{D}}C_2 & -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2\hat{\mathbf{C}} + (*) & * & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}D_{21})^T & (B_1 + B_2\hat{\mathbf{D}}D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12}\hat{\mathbf{D}}C_2 & C_1\mathbf{Y} + D_{12}\hat{\mathbf{C}} & D_{11} + D_{12}\hat{\mathbf{D}}D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (7.5)$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \quad (7.6)$$

for all $\theta \in \Theta$ and $\dot{\theta} \in \mathcal{V}$.

Notice that the rate of the parameter variation is part of Eq. (7.5) in the terms $\dot{\mathbf{X}}$ and $\dot{\mathbf{Y}}$, which are defined as:

$$\dot{\mathbf{X}} = \sum_{i=1}^{n_\theta} \dot{\theta}_i \frac{\partial \mathbf{X}(\theta)}{\partial \theta_i}, \quad \dot{\mathbf{Y}} = \sum_{i=1}^{n_\theta} \dot{\theta}_i \frac{\partial \mathbf{Y}(\theta)}{\partial \theta_i} \quad (7.7)$$

In the case where $\theta(t)$ is allowed to change arbitrary fast, the Lyapunov matrices \mathbf{X} and \mathbf{Y} must be constant; consequently, $\dot{\mathbf{X}}$ and $\dot{\mathbf{Y}}$ become zero. Additionally, if $\theta(t)$ is assumed to be constant, $\dot{\mathbf{X}}$ and $\dot{\mathbf{Y}}$ become zero; hence, the Lyapunov matrices at each value of $\theta(t)$ are independent.

The presented description involves solving an optimization problem with an infinite number of LMIs in the general case, since one LMI exists for each pair of θ and $\dot{\theta}$. In the general case the infinite number of LMIs are commonly circumvented by gridding the parameter space, $\Theta \times \mathcal{V}$, and then solving the LMIs in each of the grid points [Bianchi *et al.*, 2007, p. 174]. This method is though not straightforward, since the density of the grid must be chosen fine enough to ensure that the closed-loop system satisfies the LMIs in the entire parameter space. However, when the parameter variations are affine, the number of LMIs can be reduced, since it is only necessary to test the vertices of the parameter space, if an additional LMI is introduced.

Affine Case

In the case of affine parameter dependencies, Eq. (7.5) can be satisfied for all pairs of θ and $\dot{\theta}$ by testing the vertices of the parameter space, and introducing an additional LMI.

In the case of an affine LPV plant, the unknown matrices, which are found in the controller design, must be expressed as shown below.

$$\mathbf{X}(\theta) = \mathbf{X}_0 + \sum_{i=1}^{n_\theta} \theta_i \mathbf{X}_i \quad \mathbf{Y}(\theta) = \mathbf{Y}_0 + \sum_{i=1}^{n_\theta} \theta_i \mathbf{Y}_i \quad \hat{\mathbf{A}}(\theta) = \hat{\mathbf{A}}_0 + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{A}}_i \quad (7.8a)$$

$$\hat{\mathbf{B}}(\theta) = \hat{\mathbf{B}}_0 + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{B}}_i \quad \hat{\mathbf{C}}(\theta) = \hat{\mathbf{C}}_0 + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{C}}_i \quad \hat{\mathbf{D}}(\theta) = \hat{\mathbf{D}}_0 + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{D}}_i \quad (7.8b)$$

According to [Bianchi *et al.*, 2007, p. 175] an extra LMI must be satisfied to ensure that the system is stable in the entire parameter space. This though introduces some conservatism, but as the alternative is to introduce gridding, this conservatism is preferred. The LMIs used to design an LPV controller with performance level γ having an affine parameter description are written below, where the unknown matrices \mathbf{X} , \mathbf{Y} , $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{D}}$ are defined as shown in Eq. (7.8).

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}C_2 + (*) & * & * & * \\ \hat{\mathbf{A}}^T + A + B_2\hat{\mathbf{D}}C_2 & -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2\hat{\mathbf{C}} + (*) & * & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}D_{21})^T & (B_1 + B_2\hat{\mathbf{D}}D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12}\hat{\mathbf{D}}C_2 & C_1\mathbf{Y} + D_{12}\hat{\mathbf{C}} & D_{11} + D_{12}\hat{\mathbf{D}}D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (7.9)$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \quad (7.10)$$

for all $\theta \in \theta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$, and

$$\begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * & * \\ 0 & A_i \mathbf{Y}_i + (*) & * & * \\ B_{1,i}^T \mathbf{X}_i & 0 & 0 & * \\ 0 & C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \geq 0 \quad (7.11)$$

for $i = 1 \dots n_\theta$.

Notice that the first two LMIs are identical to the general case except that they should only be satisfied in the vertices of the parameter space and that the unknown matrices are affine in θ as specified in Eq. (7.8). Additionally, Eq. (7.11) is only dependent on the terms of the affine matrices on which two matrices that are dependent on θ are multiplied. This implies that Eq. (7.11) disappears if \mathbf{X} and \mathbf{Y} are constant.

It can be seen from the structure of Eq. (7.11) that \mathbf{X}_i should be in the null space of $B_{1,i}^T$, and \mathbf{Y}_i should be in the null space of $C_{1,i}$ to avoid getting an indefinite matrix. This requirement can be relaxed by rewriting the matrix inequality Eq. (7.9). If Eq. (7.9) is denoted $\Phi(\theta)$, it is possible to write the bounded real lemma for every point in the parameter space, denoted θ_* , using only a scaling of the matrices in the vertices and the extra LMI, as shown below.

$$\Phi(\theta_*) = \alpha\Phi(\theta_{v1}) + (1 - \alpha)\Phi(\theta_{v2}) + (\alpha^2 - \alpha)(\theta_{v1} - \theta_{v2})^2 \begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * & * \\ 0 & A_i \mathbf{Y}_i + (*) & * & * \\ B_{1,i}^T \mathbf{X}_i & 0 & 0 & * \\ 0 & C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \quad (7.12)$$

By introducing a variable $\epsilon \geq 0$, the optimization problem, which gives a stabilizing controller with performance level γ , can be reformulated for the considered scalar case as shown below. Here no restrictions exist to the structure of \mathbf{X}_i and \mathbf{Y}_i . A similar result is derived in Appendix D for the passive fault-tolerant controller design.

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}C_2 + (*) & * & * & * \\ \hat{\mathbf{A}}^T + A + B_2\hat{\mathbf{D}}C_2 & -\dot{\mathbf{Y}} + \mathbf{AY} + B_2\hat{\mathbf{C}} + (*) & * & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}D_{21})^T & (B_1 + B_2\hat{\mathbf{D}}D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12}\hat{\mathbf{D}}C_2 & C_1\mathbf{Y} + D_{12}\hat{\mathbf{C}} & D_{11} + D_{12}\hat{\mathbf{D}}D_{21} & -\gamma I_{n_z} \end{bmatrix} < -\epsilon \quad (7.13)$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \quad (7.14)$$

for all $\theta \in \theta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$ and

$$\begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * & * \\ 0 & A_i \mathbf{Y}_i + (*) & * & * \\ B_{1,i}^T \mathbf{X}_i & 0 & 0 & * \\ 0 & C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \geq \frac{-4\epsilon}{(\theta_{v1} - \theta_{v2})^2} \quad (7.15)$$

for $i = 1$ (scalar case).

The optimization problem for designing the active fault-tolerant controller has been set-up. In the next subsection the controller synthesis is explained.

Active Fault-Tolerant LPV Controller Synthesis

The purpose of this subsection is to explain how the controller is synthesized based on the matrices found by solving the optimization problems set up in the previous subsection.

To synthesize the controller, the controller matrices must be obtained from the affine matrices found using the LMIs in the previous subsection. Unfortunately, the LPV controller matrices are not affine themselves. Therefore, the controller matrices have to be calculated online to every given value of $\theta(t)$ and $\dot{\theta}(t)$. The procedure which is utilized to synthesize the LPV controller is shown in the following list:

1. Compute $\hat{\mathbf{A}}(\theta)$, $\hat{\mathbf{B}}(\theta)$, $\hat{\mathbf{C}}(\theta)$, $\hat{\mathbf{D}}(\theta)$, $\mathbf{X}(\theta)$, and $\mathbf{Y}(\theta)$ using the measured value of $\theta(t)$.
2. Find $M(\theta)$ and $N(\theta)$ by solving the factorization problem:

$$I - \mathbf{X}(\theta)\mathbf{Y}(\theta) = N(\theta)M^T(\theta) \quad (7.16)$$

3. Compute $A_c(\theta, \dot{\theta})$, $B_c(\theta)$, $C_c(\theta)$, and $D_c(\theta)$ from the equations:

$$\begin{aligned} A_c(\theta, \dot{\theta}) &= N^{-1}(\theta) \left(\mathbf{X}(\theta)\dot{\mathbf{Y}}(\theta) + N(\theta)\dot{M}^T(\theta) + \hat{\mathbf{A}}(\theta) - \hat{\mathbf{B}}(\theta)C_2\mathbf{Y}(\theta) \right. \\ &\quad \left. - \mathbf{X}(\theta) \left(A(\theta) - B_2\hat{\mathbf{D}}(\theta)C_2 \right) \mathbf{Y}(\theta) - \mathbf{X}(\theta)B_2\hat{\mathbf{C}}(\theta) \right) M^{-T}(\theta) \end{aligned} \quad (7.17a)$$

$$B_c(\theta) = N^{-1}(\theta) \left(\hat{\mathbf{B}}(\theta) - \mathbf{X}(\theta)B_2\hat{\mathbf{D}}(\theta) \right) \quad (7.17b)$$

$$C_c(\theta) = \left(\hat{\mathbf{C}}(\theta) - \hat{\mathbf{D}}(\theta)C_2\mathbf{Y}(\theta) \right) M^{-T}(\theta) \quad (7.17c)$$

$$D_c(\theta) = \hat{\mathbf{D}}(\theta) \quad (7.17d)$$

When $\dot{\theta}(t)$ is not measured, which is true in the considered case, the synthesis procedure cannot be utilized directly. To synthesize the controller in this situation some restrictions must be put on \mathbf{X} and \mathbf{Y} . According to [Apkarian and Adams, 1998] it is not possible to synthesize the LPV controller if both \mathbf{X} and \mathbf{Y} are parameter-dependent while the rate of $\theta(t)$ varies. If at least one of \mathbf{X} or \mathbf{Y} is constant, or if $\dot{\theta}(t) = 0$, then it is possible to synthesize the LPV controller by appropriately choosing M and N as shown in Table 7.1.

$\dot{\theta}$	\mathbf{X}	\mathbf{Y}	M and N	Implementable
$\dot{\theta} = 0$	$\mathbf{X} = \mathbf{X}(\theta)$	$\mathbf{Y} = \mathbf{Y}(\theta)$	$NM^T = I - \mathbf{X}(\theta)\mathbf{Y}(\theta)$	Yes
$\dot{\theta} \in \Theta$	$\mathbf{X} = \mathbf{X}(\theta)$	$\mathbf{Y} = \mathbf{Y}(\theta)$	$NM^T = I - \mathbf{X}(\theta)\mathbf{Y}(\theta)$	No
$\dot{\theta} \in \Theta$	$\mathbf{X} = \mathbf{X}(\theta)$	$\mathbf{Y} = \mathbf{Y}_0$	$N = I - \mathbf{X}(\theta)\mathbf{Y}_0$, $M = I$	Yes
$\dot{\theta} \in \Theta$	$\mathbf{X} = \mathbf{X}_0$	$\mathbf{Y} = \mathbf{Y}(\theta)$	$N = I$, $M = I - \mathbf{Y}(\theta)\mathbf{X}_0$	Yes
$\dot{\theta}$ is unbounded	$\mathbf{X} = \mathbf{X}_0$	$\mathbf{Y} = \mathbf{Y}_0$	$NM^T = I - \mathbf{X}_0\mathbf{Y}_0$	Yes

Table 7.1: Overview of possible selections of \mathbf{X} , \mathbf{Y} and an appropriate choice of M and N . Notice that the last column indicates when the LPV controller is implementable without measuring $\dot{\theta}(t)$.

If either \mathbf{X} or \mathbf{Y} is constant, or if $\dot{\theta}(t) = 0$, and the matrices M and N are chosen according to Table 7.1, then the term $\mathbf{X}(\theta)\dot{\mathbf{Y}}(\theta) + N(\theta)\dot{M}^T(\theta)$ vanishes from Eq. (7.17a), and the expression for $A_c(\theta)$ becomes independent of $\dot{\theta}(t)$:

$$\begin{aligned} A_c(\theta) &= N^{-1}(\theta) \left(\hat{\mathbf{A}}(\theta) - \hat{\mathbf{B}}(\theta)C_2\mathbf{Y}(\theta) - \mathbf{X}(\theta) \left(A(\theta) - B_2\hat{\mathbf{D}}(\theta)C_2 \right) \mathbf{Y}(\theta) \right. \\ &\quad \left. - \mathbf{X}(\theta)B_2\hat{\mathbf{C}}(\theta) \right) M^{-T}(\theta) \end{aligned} \quad (7.18)$$

Choosing either \mathbf{X} or \mathbf{Y} constant introduces some conservatism. Furthermore, the choice between selecting \mathbf{X} or \mathbf{Y} constant is not obvious and has to be decided by trial and error for getting the least conservative LPV controller for the particular application.

The controller synthesis procedure has been outlined in this subsection. This finalizes the description of the active fault-tolerant LPV controller design procedure. In the next section a method for designing a passive fault-tolerant LPV controller is derived in a similar fashion.

7.3 Passive Fault-Tolerant LPV Controller Design Procedure

The developed design procedure for establishing a passive fault-tolerant LPV controller is obtained by combining the LPV controller design method described in [Bianchi *et al.*, 2007, pp. 159-179] and the robust output feedback controller design method presented in [Jabbari, 1997], which is based on a structured uncertainty description. A derivation of the method is provided in Appendix D.

Passive Fault-Tolerant LPV System Description

To distinguish between the measured and unmeasured parameter variations, the system description presented in Eq. (7.1) on Page 100 is modified as shown below.

$$\dot{x}(t) = A(\theta(t), \Delta(t))x(t) + B_1(\theta(t), \Delta(t))w(t) + B_2(\theta(t), \Delta(t))u(t) \quad (7.19a)$$

$$z(t) = C_1(\theta(t), \Delta(t))x(t) + D_{11}(\theta(t), \Delta(t))w(t) + D_{12}(\theta(t), \Delta(t))u(t) \quad (7.19b)$$

$$y(t) = C_2(\theta(t), \Delta(t))x(t) + D_{21}(\theta(t), \Delta(t))w(t) + D_{22}(\theta(t), \Delta(t))u(t) \quad (7.19c)$$

where:

$\theta(t)$ is the measured parameter vector

$\Delta(t)$ is the unmeasured parameter vector

This system description is very general, but only a less general model is necessary for the considered design problem. Therefore, it is reduced by only allowing the system matrix to be dependent on Δ , due to the knowledge of the uncertainties of the considered system. This can be accomplished without loss of generality. Additionally, an affine description equivalent to Eq. (7.2) on Page 100 is adopted and shown below.

$$\begin{bmatrix} A(\theta, \Delta) & B_1(\theta, \Delta) & B_2(\theta, \Delta) \\ C_1(\theta, \Delta) & D_{11}(\theta, \Delta) & D_{12}(\theta, \Delta) \\ C_2(\theta, \Delta) & D_{21}(\theta, \Delta) & 0 \end{bmatrix} = \begin{bmatrix} A_0 & B_{1,0} & B_{2,0} \\ C_{1,0} & D_{11,0} & D_{12,0} \\ C_{2,0} & D_{21,0} & 0 \end{bmatrix} + \sum_{i=1}^{n_\theta} \theta_i \begin{bmatrix} A_i & B_{1,i} & 0 \\ C_{1,i} & D_{11,i} & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sum_{j=1}^{n_\Delta} \Delta_j \begin{bmatrix} A_{\Delta j} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.20)$$

where:

n_θ is the number of measured parameters

n_Δ is the number of uncertain parameters

The following matrices are defined for convenience, and are used in the remaining part of this chapter:

$$A^\theta(\theta) \triangleq \sum_{i=1}^{n_\theta} \theta_i A_i, \quad A^\Delta(\Delta) \triangleq \sum_{j=1}^{n_\Delta} \Delta_j A_{\Delta j} \quad (7.21)$$

From the presented reduced LPV model, the optimization problem involved with designing a passive fault-tolerant controller is outlined in the next subsection.

Passive Fault-Tolerant LPV Controller Optimization Problem

The active and passive fault-tolerant LPV controller optimization problems have very similar descriptions, but are very different to solve, since the passive fault-tolerant LPV controller optimization problem involves solving Bilinear Matrix Inequalities (BMIs) instead of LMIs. Therefore, the entire design procedure for the passive approach is outlined in this subsection.

The passive fault-tolerant LPV controller description is identical to Eq. (7.3) on Page 101 and is shown below. Notice that the controller matrices are independent of Δ , since this is uncertain and cannot be used as scheduling parameter.

$$\dot{x}_c(t) = A_c(\theta(t))x_c(t) + B_c(\theta(t))y(t) \quad (7.22a)$$

$$u(t) = C_c(\theta(t))x_c(t) + D_c(\theta(t))y(t) \quad (7.22b)$$

A controller design problem is set up using a similar approach as for the active fault-tolerant case. First, the optimization problem for the general case is set up then the optimization problem for the affine case is deduced from this.

General Case

The auxiliary matrix, $\hat{\mathbf{A}}$, is different for the passive than for the active fault-tolerant controller, since the dependency of $\Delta(t)$ should be removed. All auxiliary matrices are shown below.

$$\begin{aligned}\hat{\mathbf{A}}_{\Delta} &= NA_c M^T - \mathbf{X} \dot{\mathbf{Y}} - N \dot{M}^T + \mathbf{X} (A_0 + A^\theta + B_2 D_c C_2) \mathbf{Y} \\ &\quad + \mathbf{X} B_2 C_c M^T + N B_c C_2 \mathbf{Y}\end{aligned}\tag{7.23a}$$

$$\hat{\mathbf{B}}_{\Delta} = NB_c + \mathbf{X} B_2 D_c \tag{7.23b}$$

$$\hat{\mathbf{C}}_{\Delta} = C_c M^T + D_c C_2 \mathbf{Y} \tag{7.23c}$$

$$\hat{\mathbf{D}}_{\Delta} = D_c \tag{7.23d}$$

When the controller design problem is reformulated into matrix inequalities, the problem cannot be formulated using only LMIs, but becomes a BMI description, where \mathbf{X} , \mathbf{Y} , $\hat{\mathbf{A}}_{\Delta}$, $\hat{\mathbf{B}}_{\Delta}$, $\hat{\mathbf{C}}_{\Delta}$, and $\hat{\mathbf{D}}_{\Delta}$ are unknown and only dependent on $\theta(t)$. By solving the following BMIs, a stabilizing controller with performance level γ can be obtained:

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X} A + \hat{\mathbf{B}}_{\Delta} C_2 + (*) & * & * & * \\ \hat{\mathbf{A}}_{\Delta}^T + \mathbf{Y} A^{\Delta T} \mathbf{X} + A + B_2 \hat{\mathbf{D}}_{\Delta} C_2 & -\dot{\mathbf{Y}} + A \mathbf{Y} + B_2 \hat{\mathbf{C}}_{\Delta} + (*) & * & * \\ (\mathbf{X} B_1 + \hat{\mathbf{B}}_{\Delta} D_{21})^T & (B_1 + B_2 \hat{\mathbf{D}}_{\Delta} D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12} \hat{\mathbf{D}}_{\Delta} C_2 & C_1 \mathbf{Y} + D_{12} \hat{\mathbf{C}}_{\Delta} & D_{11} + D_{12} \hat{\mathbf{D}}_{\Delta} D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{7.24}$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \tag{7.25}$$

for all $\theta \in \Theta$, $\Delta \in \mathcal{D}$, and $\dot{\theta} \in \mathcal{V}$.

The optimization problem is now formulated as a BMI description, making it impossible to be solved directly, since the description is non-linear in the unknown variables. To circumvent this problem, the projection lemma is applied on Eq. (7.24) to form the necessary conditions for this inequality shown below. One of these necessary conditions must be solved before Eq. (7.24) to provide half the unknown variables; hence making Eq. (7.24) linear in the unknown variables.

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X} A + \hat{\mathbf{B}}_{\Delta} C_2 + (*) & * & * \\ (\mathbf{X} B_1 + \hat{\mathbf{B}}_{\Delta} D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12} \hat{\mathbf{D}}_{\Delta} C_2 & D_{11} + D_{12} \hat{\mathbf{D}}_{\Delta} D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{7.26}$$

for all $\theta \in \Theta$, $\Delta \in \mathcal{D}$, and $\dot{\theta} \in \mathcal{V}$.

or

$$\begin{bmatrix} -\dot{\mathbf{Y}} + A \mathbf{Y} + B_2 \hat{\mathbf{C}}_{\Delta} + (*) & * & * \\ (B_1 + B_2 \hat{\mathbf{D}}_{\Delta} D_{21})^T & -\gamma I_{n_w} & * \\ C_1 \mathbf{Y} + D_{12} \hat{\mathbf{C}}_{\Delta} & D_{11} + D_{12} \hat{\mathbf{D}}_{\Delta} D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{7.27}$$

for all $\theta \in \Theta$, $\Delta \in \mathcal{D}$, and $\dot{\theta} \in \mathcal{V}$.

By solving one of the necessary conditions shown above, either \mathbf{X} , $\hat{\mathbf{B}}_{\Delta}$, $\hat{\mathbf{D}}_{\Delta}$ or \mathbf{Y} , $\hat{\mathbf{C}}_{\Delta}$, $\hat{\mathbf{D}}_{\Delta}$ are found and can be inserted into Eq. (7.24) to search for the remaining variables, since this is now an LMI in the unknown variables.

To enable the controller to have a better performance in the normal case than in the faulty case, different values for γ should be chosen for each value of Δ . This implies that a performance degradation as shown in Figure 7.3 can be obtained.

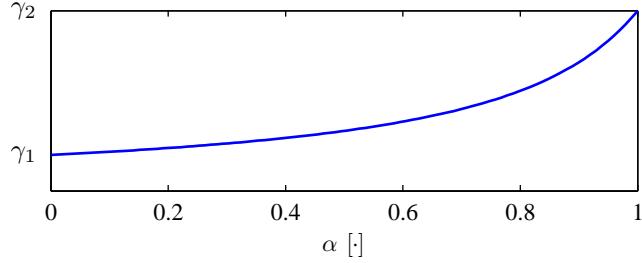


Figure 7.3: Evolution of γ between the vertices γ_1 and γ_2 , showing the degradation in the performance bound from the fault-free situation ($\alpha = 0$) to the faulty situation ($\alpha = 1$).

Affine Case

The parameter variations in the considered model can be described in an affine manner as explained in Section 7.4. Therefore, the controller design optimization problem is recast to fit this property.

Similar to the active fault-tolerant controller description, the Lyapunov matrices and auxiliary matrices should be described using an affine description. Notice that neither of the matrices is dependent on Δ , since the controller cannot be dependent on an uncertain parameter:

$$\mathbf{X}(\theta) = \mathbf{X} + \sum_{i=1}^{n_\theta} \theta_i \mathbf{X}_i \quad \mathbf{Y}(\theta) = \mathbf{Y}_0 + \sum_{i=1}^{n_\theta} \theta_i \mathbf{Y}_i \quad \hat{\mathbf{A}}_\Delta(\theta) = \hat{\mathbf{A}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{A}}_{\Delta i} \quad (7.28a)$$

$$\hat{\mathbf{B}}_\Delta(\theta) = \hat{\mathbf{B}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{B}}_{\Delta i} \quad \hat{\mathbf{C}}_\Delta(\theta) = \hat{\mathbf{C}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{C}}_{\Delta i} \quad \hat{\mathbf{D}}_\Delta(\theta) = \hat{\mathbf{D}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{D}}_{\Delta i} \quad (7.28b)$$

Even though the controller design problem involves solving BMIs, it is possible to add an extra LMI which makes it sufficient to solve the BMIs in the vertices of the uncertainty space, as shown below.

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}C_2 + (*) & * & * & * \\ \hat{\mathbf{A}}^T + \mathbf{Y}A^{\Delta T} \mathbf{X} + A + B_2 \hat{\mathbf{D}}C_2 & -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2 \hat{\mathbf{C}} + (*) & * & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}D_{21})^T & (B_1 + B_2 \hat{\mathbf{D}}D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12} \hat{\mathbf{D}}C_2 & C_1 \mathbf{Y} + D_{12} \hat{\mathbf{C}} & D_{11} + D_{12} \hat{\mathbf{D}}D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (7.29)$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \quad (7.30)$$

for all $\theta \in \theta_{\text{vex}}$, $\Delta \in \Delta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$, and

$$\begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * & * \\ \mathbf{Y}_i A^{\Delta T} \mathbf{X}_i & A_i \mathbf{Y}_i + (*) & * & * \\ B_{1,i}^T \mathbf{X}_i & 0 & 0 & * \\ 0 & C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \geq 0 \quad (7.31)$$

for $i = 1 \dots n_\theta$ and $\Delta \in \Delta_{\text{vex}}$.

To eliminate the restrictions on the structure of \mathbf{Y}_i and \mathbf{X}_i , the extra LMI can be reformulated as shown for the active fault-tolerant controller. Notice that the (2, 1) block of Eq. (7.31) becomes zero, when either \mathbf{X} or \mathbf{Y} is held constant. This implies that the constraint equals the resulting constraint for the active fault-tolerant controller design.

Due to the BMI description the optimization problem cannot be solved in one step. Therefore, one of the necessary conditions shown below must be satisfied first.

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}_\Delta C_2 + (*) & * & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}_\Delta D_{21})^T & -\gamma I_{n_w} & * \\ C_1 + D_{12}\hat{\mathbf{D}}_\Delta C_2 & D_{11} + D_{12}\hat{\mathbf{D}}_\Delta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (7.32)$$

for all $\theta \in \theta_{\text{vex}}$, $\Delta \in \Delta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$, and

$$\begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * \\ B_{1,i}^T \mathbf{X}_i & 0 & * \\ 0 & 0 & 0 \end{bmatrix} \geq 0 \quad (7.33)$$

for $i = 1 \dots n_\theta$ and $\Delta \in \Delta_{\text{vex}}$.

or

$$\begin{bmatrix} -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2\hat{\mathbf{C}}_\Delta + (*) & * & * \\ (B_1 + B_2\hat{\mathbf{D}}_\Delta D_{21})^T & -\gamma I_{n_w} & * \\ C_1\mathbf{Y} + D_{12}\hat{\mathbf{C}}_\Delta & D_{11} + D_{12}\hat{\mathbf{D}}_\Delta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (7.34)$$

for all $\theta \in \theta_{\text{vex}}$, $\Delta \in \Delta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$, and

$$\begin{bmatrix} A_i \mathbf{Y}_i + (*) & * & * \\ 0 & 0 & * \\ C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \geq 0 \quad (7.35)$$

for $i = 1 \dots n_\theta$ and $\Delta \in \Delta_{\text{vex}}$.

The design procedure for the passive fault-tolerant controller has been outlined. In the next subsection the controller synthesis is explained.

Passive Fault-Tolerant LPV Controller Synthesis

The controller synthesis procedures for the active and passive fault-tolerant controllers are almost identical. The only difference is the change in the equation describing $A_c(\theta, \dot{\theta})$. The synthesis procedure for the passive fault-tolerant controller is shown below.

1. Compute $\hat{\mathbf{A}}_\Delta(\theta)$, $\hat{\mathbf{B}}_\Delta(\theta)$, $\hat{\mathbf{C}}_\Delta(\theta)$, $\hat{\mathbf{D}}_\Delta(\theta)$, $\mathbf{X}(\theta)$, and $\mathbf{Y}(\theta)$ using the measured value of $\theta(t)$.
2. Find $M(\theta)$ and $N(\theta)$ by solving the factorization problem:

$$I - \mathbf{X}(\theta)\mathbf{Y}(\theta) = N(\theta)M^T(\theta) \quad (7.36)$$

3. Compute $A_c(\theta, \dot{\theta})$, $B_c(\theta)$, $C_c(\theta)$, and $D_c(\theta)$ from the equations:

$$\begin{aligned} A_c(\theta, \dot{\theta}) &= N^{-1}(\theta) \left(\mathbf{X}(\theta)\dot{\mathbf{Y}}(\theta) + N(\theta)\dot{M}_\Delta^T(\theta) + \hat{\mathbf{A}}_\Delta(\theta) - \hat{\mathbf{B}}_\Delta(\theta)C_2\mathbf{Y}(\theta) \right. \\ &\quad \left. - \mathbf{X}(\theta) \left(A_0 + A^\theta - B_2\hat{\mathbf{D}}_\Delta(\theta)C_2 \right) \mathbf{Y}(\theta) - \mathbf{X}(\theta)B_2\hat{\mathbf{C}}_\Delta(\theta) \right) M^{-T}(\theta) \end{aligned} \quad (7.37a)$$

$$B_c(\theta) = N^{-1}(\theta) \left(\hat{\mathbf{B}}_\Delta(\theta) - \mathbf{X}(\theta)B_2\hat{\mathbf{D}}_\Delta(\theta) \right) \quad (7.37b)$$

$$C_c(\theta) = \left(\hat{\mathbf{C}}_\Delta(\theta) - \hat{\mathbf{D}}_\Delta(\theta)C_2\mathbf{Y}(\theta) \right) M^{-T}(\theta) \quad (7.37c)$$

$$D_c(\theta) = \hat{\mathbf{D}}_\Delta(\theta) \quad (7.37d)$$

As shown in Table 7.1 on Page 104 it is not possible to realize the LPV controller where both \mathbf{X} and \mathbf{Y} are dependent on $\theta(t)$ in the situation where $\dot{\theta}(t)$ is not measured. If N and M are chosen according to Table 7.1 then $A_c(\theta)$ can be found as shown below.

$$\begin{aligned} A_c(\theta) &= N^{-1}(\theta) \left(\hat{\mathbf{A}}_\Delta(\theta) - \hat{\mathbf{B}}_\Delta(\theta)C_2\mathbf{Y}(\theta) - \mathbf{X}(\theta) \left(A_0 + A^\theta - B_2\hat{\mathbf{D}}_\Delta(\theta)C_2 \right) \mathbf{Y}(\theta) \right. \\ &\quad \left. - \mathbf{X}(\theta)B_2\hat{\mathbf{C}}_\Delta(\theta) \right) M^{-T}(\theta) \end{aligned} \quad (7.38)$$

The passive fault-tolerant controller design procedure has been outlined. In the next section the system equations are adapted to the affine LPV description required by the chosen controller design method.

7.4 Common LPV System Description

The purpose of this section is to set up an affine LPV description of the system including a performance specification. Initially, the parameter-dependent variables are identified and the affine LPV system model is derived. Secondly, the rate bounds of the scheduling parameters are determined. Finally, the open-loop system is augmented by input and sensitivity filters to include a performance specification.

The LPV controllers designed in this chapter replaces the classical speed controller, shown in Figure 2.4 on Page 8, and should therefore control the generator speed using the pitch system. To minimize the stress on the drive train, the drive train stress damper explained in Section C.3 is applied in parallel with the LPV controller. Notice, this is only possible when the generator speed is measured, as explained in Section 6.4. Additionally, the power controller for the classical design is applied to cancel steady-state errors on the power output.

Since the two controllers operating in parallel with the LPV controller keep the generator torque in the proximity of its rated value, the generator torque is assumed constant when designing the LPV controllers. Furthermore, tower dynamics is considered to be a disturbance included in the relative wind speed, since no effort is done to attenuate the tower movement. The final simplification relates to the collective pitching strategy, which permits a single pitch actuator model to replace the three identical models.

Affine LPV System Description

The two types of varying parameters of the system are explained below.

- **Parameter variations along the nominal operating trajectory:** The non-linear nature of the aerodynamics makes the partial derivatives of the aerodynamic torque shown in Eq. (3.9) on Page 15 be dependent on the operating point; i.e. $\partial T_a(t)/\partial \beta$, $\partial T_a(t)/\partial v_r$, and $\partial T_a(t)/\partial \omega_r$ change along the nominal operating trajectory.
- **Parameter variations caused by changing air content in the hydraulic oil of the pitch system:** Varying air content in the hydraulic oil of the pitch system changes the dynamics of the pitch system as specified in Section 4.5; i.e. ω_n and ζ in the pitch system equations depend on the air content.

According to Eq. (7.2) on Page 100, B_2 is assumed to be constant. However, the model of the system in Eq. (3.30) on Page 22 has a parameter-varying entry in this matrix, caused by changing dynamics of the pitch system. In this situation though, it is possible to perform a state transformation which confines the parameter dependencies to the system matrix. The system description shown in Eq. (7.39) originates from transforming Eq. (5.24) on Page 65 using: $\dot{\beta}'(t) = \frac{1}{\omega_n^2(t)} \dot{\beta}(t)$.

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ \begin{bmatrix} \dot{\beta}(t) \\ \ddot{\beta}'(t) \end{bmatrix} &= \begin{bmatrix} 0 & \omega_n^2(t) \\ -1 & -2\zeta(t)\omega_n(t) \end{bmatrix} \begin{bmatrix} \beta(t) \\ \dot{\beta}'(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \beta_{\text{ref}}(t - t_d) \end{aligned} \quad (7.39a)$$

$$\begin{aligned} y(t) &= Cx(t) + v(t) \\ \beta_{\text{mes}}(t) &= [1 \ 0] \begin{bmatrix} \beta(t) \\ \dot{\beta}'(t) \end{bmatrix} + v_\beta(t) \quad [\circ] \end{aligned} \quad (7.39b)$$

By inserting Eq. (7.39a) into the system model in Eq. (3.30) on Page 22, the following system

emerges, where parameter dependencies caused by the nominal operating condition and fault scenario are included:

$$\begin{aligned} \dot{x}(t) &= A(\theta(t))x(t) + B_1(\theta(t))w(t) + B_2u(t) \\ \begin{bmatrix} \dot{\beta}(t) \\ \ddot{\beta}'(t) \\ \dot{\theta}_\Delta(t) \\ \dot{\omega}_g(t) \\ \dot{\omega}_r(t) \end{bmatrix} &= \begin{bmatrix} 0 & a_{12}(\theta_f(t)) & 0 & 0 & 0 \\ -1 & a_{22}(\theta_f(t)) & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{N_g} & 1 \\ 0 & 0 & \frac{K_{dt}}{J_g N_g} & -\left(\frac{B_{dt}}{J_g N_g^2} + \frac{B_g}{J_g}\right) & \frac{B_{dt}}{N_g J_g} \\ \frac{1}{J_r} \frac{\partial T_a(\theta_{op}(t))}{\partial \beta} & 0 & -\frac{K_{dt}}{J_r} & \frac{B_{dt}}{N_g J_r} & -\frac{B_{dt} + B_r}{J_r} + \frac{1}{J_r} \frac{\partial T_a(\theta_{op}(t))}{\partial \omega_r} \end{bmatrix} \begin{bmatrix} \beta(t) \\ \dot{\beta}'(t) \\ \theta_\Delta(t) \\ \omega_g(t) \\ \omega_r(t) \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_r} \frac{\partial T_a(\theta_{op}(t))}{\partial v_r} \end{bmatrix} v_r(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \beta_{ref}(t) \end{aligned} \quad (7.40)$$

where:

$\theta_f(t)$ is the scheduling parameter specifying the air content in the hydraulic oil.

$\theta_{op}(t)$ is the scheduling parameter specifying the wind speed

The wind speed, $v_r(t)$, is the exogenous input to the system and introduces a parameter-dependent element in B_1 . To limit the bandwidth of the exogenous input in the design, a filter is applied to shape the frequency content of the wind speed. This moves the effective wind speed into the state vector in the affine LPV system and moves the parameter-dependent element from B_1 to A . By applying the described modifications to the system, an affine LPV system can be set up as shown below.

$$\begin{bmatrix} A(\theta) & B_1 & B_2 \\ C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} = \begin{bmatrix} A_0 & B_1 & B_2 \\ C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} + \underbrace{\theta_{op}}_{v_r} \begin{bmatrix} A_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \underbrace{\theta_f}_{\omega_n^2} \begin{bmatrix} A_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.41)$$

The system in Eq. (7.41) is a linear combination of constant matrices in the two scheduling parameters. This description is possible since the partial derivatives of the aerodynamic torque can be approximated by an affine function, having the wind speed as the independent variable, as illustrated in Figure 7.4. Similarly, the varying entries in the pitch actuator model fit well to an affine approximation in ω_n^2 , as shown in Figure 7.5.

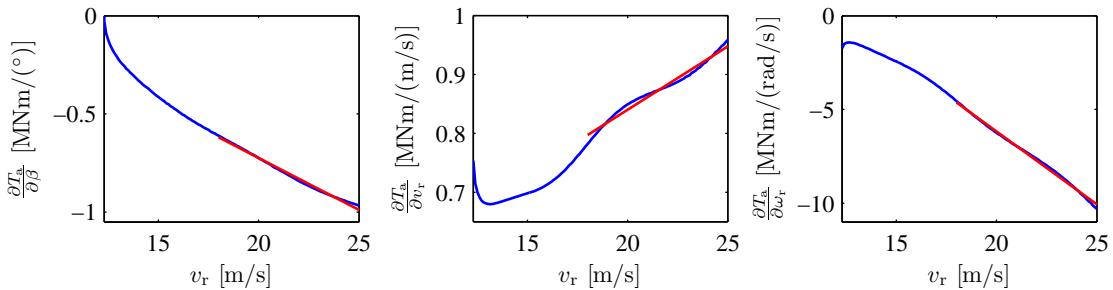


Figure 7.4: Parameter variations along the nominal operating trajectory in the full load region. The partial derivatives of the aerodynamic torque (blue) are approximated in the range from 18 m/s to 25 m/s by affine descriptions (red) using the wind speed as scheduling parameter.

Since the main focus of this project is to include fault-tolerance in the design, it is decided only to include the upper half part of the full load region, i.e. wind speeds ranging from 18 m/s to 25 m/s. The scheduling parameter, $\theta_{op}(t) = v_r(t)$, is provided by the wind speed estimator, as illustrated in Figure 7.2 on Page 99.

Figure 7.5 shows that the parameter-dependent entries in the pitch actuator model can be described using an affine approximation in $\theta_f(t) = \omega_n^2(t)$. An estimation of this value is provided by the fault diagnosis algorithm described in Section 5.5.

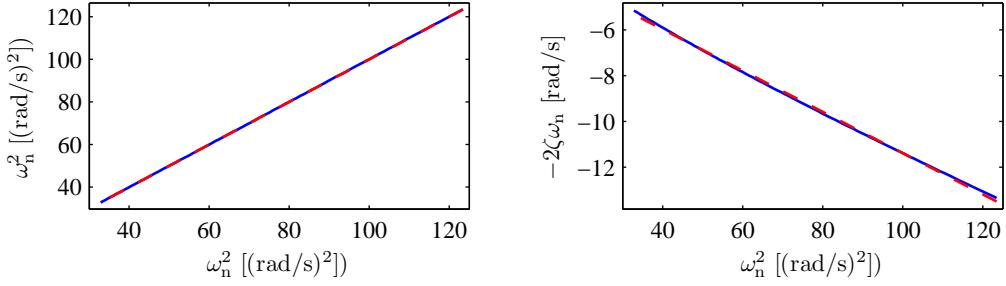


Figure 7.5: Parameter variations caused by changing air content in the hydraulic oil of the pitch system. The parameter-dependent entries of the pitch system model (blue) are approximated by affine descriptions (dashed red) using the squared natural frequency of the pitch system as scheduling parameter.

Rate Bounds

Besides determining the intervals of $\theta_{op}(t)$ and $\theta_f(t)$, it is necessary to specify the rate bounds of the scheduling parameters, since performance and stability of the LPV control system should be guaranteed for all $\dot{\theta} \in \mathcal{V}$.

By assessing the output of the wind model, it is considered that 2 m/s^2 is a good estimate of the upper bound on the rate of variation on $|\theta_{op}(t)|$. According to Section 4.5, high air content in the hydraulic oil is introduced over much longer time than the longest time constant of the system. In the design it is therefore assumed that $\dot{\theta}_f(t) = 0$.

System with Performance Specification

To provide the possibility of tuning the LPV controllers, a mixed sensitivity description is adopted. An augmented system including this description is shown in Figure 7.6, where $W_S(s)$ is the sensitivity filter and $W_M(s)$ is the control sensitivity filter. In addition to the sensitivity filters, the input disturbance filter, $W_D(s)$, band limits the exogenous input in the design and $W_N(s)$ adds measurement noise to the system outputs.

$W_S(s)$ stresses the importance of the low-frequency components of the generator speed error. Its pole in zero ensures integral action in the controllers which eliminates any steady-state error on the tracking of the generator speed reference. $W_M(s)$ weights the control effort with the aim of penalizing fast pitch angle variations. The performance measures are stated in Eq. (7.42) while the weighted performance measures appear from Eq. (7.43). The filters are specified in Eq. (7.44)-(7.46).

$$\tilde{z}(t) = \begin{bmatrix} \omega_{g,e}(t) \\ \beta_{ref}(t) \end{bmatrix} \quad (7.42)$$

$$z(t) = \begin{bmatrix} W_S(t) & 0 \\ 0 & W_M(t) \end{bmatrix} \tilde{z}(t) \quad (7.43)$$

$$W_S(s) = k_S \frac{1}{s} \quad (7.44)$$

$$W_M(s) = k_M \frac{s}{s/(10\omega_{3P}) + 1} \quad (7.45)$$

$$W_D(s) = \frac{1}{s/(1.5\omega_{3P}) + 1} \quad (7.46)$$

where:

k_S is used to determine the maximum error on $\omega_g(t)$

k_M is used to determine the maximum control signal applied by $\beta_{ref}(t)$

ω_{3P} is the 3P frequency [rad/s]

This finalizes the set-up of the necessary description to form the controllers. In the next section the design of the active fault-tolerant controller is explained.

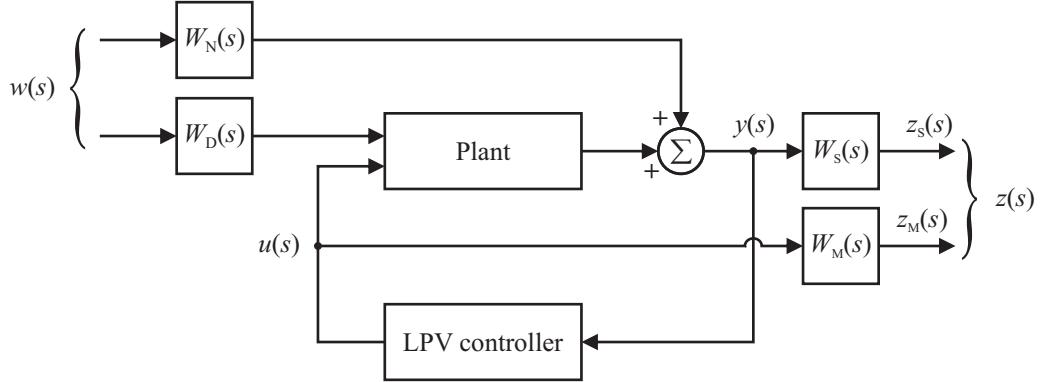


Figure 7.6: Block diagram of the augmented system for the LPV controller synthesis. The sensitivity filter, $W_S(s)$, and input disturbance filter, $W_D(s)$, have low-pass filter characteristics, whereas the control sensitivity filter, $W_M(s)$, is a high-pass filter. $W_N(s)$ adds noise directly to the system outputs.

7.5 Active Fault-Tolerant Controller Design

In this section the active fault-tolerant controllers are designed by determining the configuration of parameter-dependent Lyapunov matrices \mathbf{X} and \mathbf{Y} that results in the best performance. For both controllers, the best performance is determined as the configuration which allows minimizing γ the most, being the guaranteed maximum \mathcal{L}_2 -norm from disturbances to performance measures. Additionally, the controllers are verified to be able to operate at the vertices of the parameter space and at the rate bounds. Finally, the switch between the LPV controllers with and without the measurement of the generator speed is verified.

In Table 7.2 a comparison of different selections of Lyapunov variables is provided for the considered system. According to Table 7.1 on Page 104 the restriction is that \mathbf{X} and \mathbf{Y} cannot be dependent on θ_{op} simultaneously. However, dependencies of θ_f in both Lyapunov matrices are allowed, since $\dot{\theta}_f = 0$. Notice that the performance specifications for the two controllers are different, since the loss of the generator speed measurement reduces the available measurement information, as argued in Section 6.4. The performance specifications are made such that the two controllers have approximately the same pitch actuator usage.

The first column of γ values in Table 7.2 are obtained by solving the controller exploiting the measured generator speed, indicated by $f_{sw} = 0$. These values should not be compared to the γ values in the last column, since the controller designed for $f_{sw} = 1$ uses another performance specification. The variable $f_{sw} = 1$ indicates that the generator speed measurement is unavailable and therefore lower demands has to be placed on the tracking of the speed error. Therefore, the sensitivity filter gain, k_S , is reduced for the controller which does not rely on the generator speed measurement.

From the table it can be concluded that for the particular system it is advantageous to use $\mathbf{Y}(\theta)$ as the parameter-dependent Lyapunov matrix.

The γ values obtained in Table 7.2 are calculated based on continuous descriptions of the system and the controller. For implementing the controllers they have to be discretized. To obtain a similar behavior for a continuous controller and the discretized controller, γ is slightly increased to remove fast modes.

Verification of AFTCS

To verify that the controllers are able to operate in the entire parameter space and at the rate bounds, simulations are conducted where the wind turbine model is forced between the extremes of the parameter space. The first 50 s of the simulations are shown in Figure 7.7, for both normal air content (blue) and high air content (red) in the hydraulic oil, to enable a comparison of the behavior in the two situations. Likewise, the consequence of not using the generator speed measurement is evaluated by comparing the left and right subplots.

\mathbf{X}	\mathbf{Y}	γ (for $f_{sw} = 0$)	γ (for $f_{sw} = 1$)
$\mathbf{X} = \mathbf{X}_0$	$\mathbf{Y} = \mathbf{Y}_0$	0.660	0.627
$\mathbf{X} = \mathbf{X}_0$	$\mathbf{Y} = \mathbf{Y}(\theta_{op}, \theta_f)$	0.556	0.588
$\mathbf{X} = \mathbf{X}(\theta_f)$	$\mathbf{Y} = \mathbf{Y}(\theta_f)$	0.626	0.619
$\mathbf{X} = \mathbf{X}(\theta_f)$	$\mathbf{Y} = \mathbf{Y}(\theta_{op}, \theta_f)$	0.556	0.588
$\mathbf{X} = \mathbf{X}(\theta_{op}, \theta_f)$	$\mathbf{Y} = \mathbf{Y}_0$	0.657	0.627
$\mathbf{X} = \mathbf{X}(\theta_{op}, \theta_f)$	$\mathbf{Y} = \mathbf{Y}(\theta_f)$	0.624	0.618

Table 7.2: Values of γ for different selections of variables in the gain-scheduled control problem. The first columns of γ values are obtained solving the controller using the generator speed measurement ($f_{sw} = 0$), while the values of γ stated in the last column result from solving the controller which does not rely on the generator speed measurement ($f_{sw} = 1$). Notice that the γ values in the two columns should not be compared as the optimization problems are solved subject to different performance specifications.

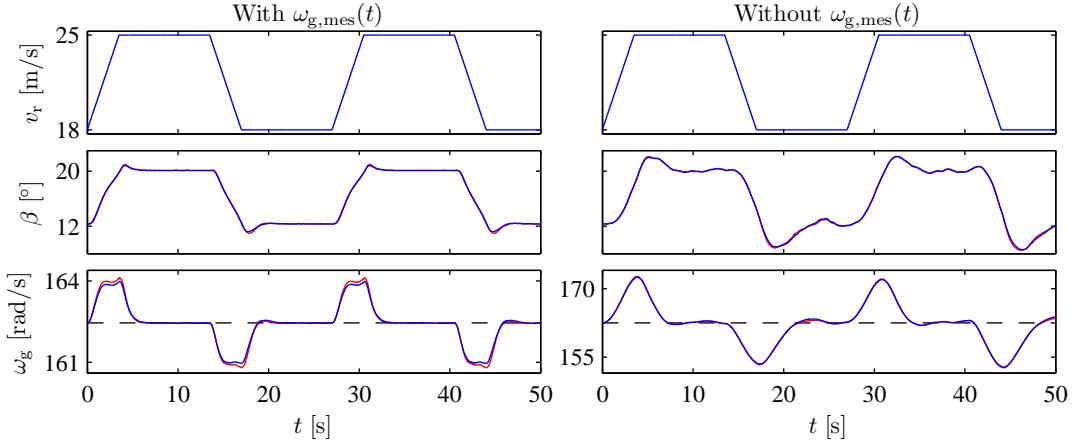


Figure 7.7: Simulation results where the AFTC is forced between the extremes of the operational range of the wind speed and at its rate limits. The simulations are conducted at both normal air content (blue) and high air content (red).

Based on the simulations it is concluded that the AFTCs are able to operate in the entire operating region specified in the design. Considering the changed dynamics of the pitch system, the differences on $\beta(t)$ are hardly noticed, although they cause the minor differences apparent on $\omega_g(t)$. The conclusion is that the performance on system level is not degraded much even considering the changed dynamics of the pitch system. When comparing the left and right subplots it is obvious that the absence of the generator speed measurement has a large impact on the performance.

Verification of Switch between the two Controllers

To verify that it is possible to switch between the two fault-tolerant LPV controllers when the measurement of the generator speed is lost, a simulation of the considered scenario is tested in this subsection.

The switch between the two fault-tolerant LPV controllers is implemented similar to the bumpless transfer used for the reference controller, as explained in Section C.4. This is chosen since no time is available for a smooth scheduling due to the abrupt fault. The switch, which should happen with no jumps in the control signal, is verified by conducting a simulation where the measurement of the generator speed measurement is lost at $t = 100$ s. The simulation result is shown in Figure 7.8. It is concluded that the switch works satisfactorily since no jumps are present on the control signal.

In this section the active fault-tolerant controllers have been designed and verified. In the next section a description of the passive fault-tolerant controllers is provided.

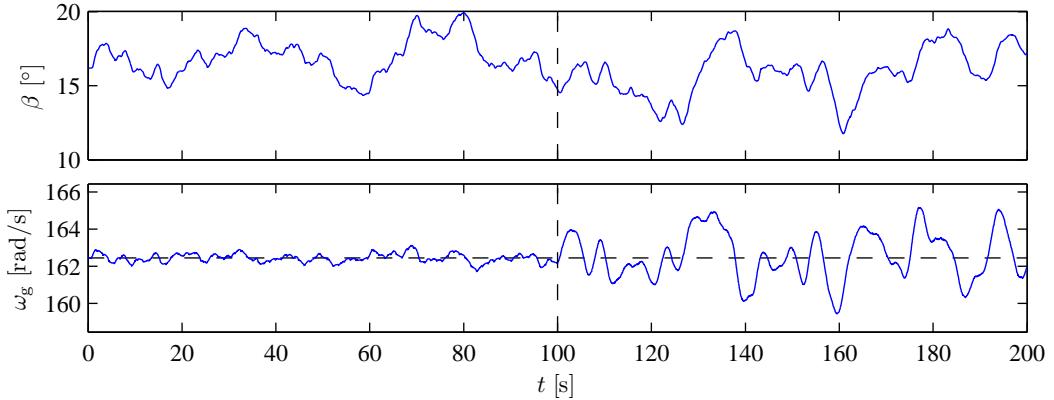


Figure 7.8: *Simulation result of a switch between the two active fault-tolerant LPV controllers, where the generator speed is lost at $t = 100$ s, which is indicated by the vertical dashed line. The horizontal dashed line represents the rated generator speed.*

7.6 Passive Fault-Tolerant Controller Design

In this section the passive fault-tolerant controllers are designed in a similar fashion as in the previous section, i.e. by minimizing γ . Additionally, the controllers are verified to be able to operate at the vertices of the parameter space and at the rate bounds. The transition between the controllers with and without the generator speed measurement is not tested, since it is identical to the switch verified in the previous section.

To design the passive fault-tolerant controllers the performance degradations in the faulty cases must be specified. The obtainable γ values for the normal and faulty cases are not known when initiating the design, and therefore some steps are performed to know the bounds on the γ values before designing the passive fault-tolerant controllers. The bounds on the γ values are determined based on Figure 7.9.

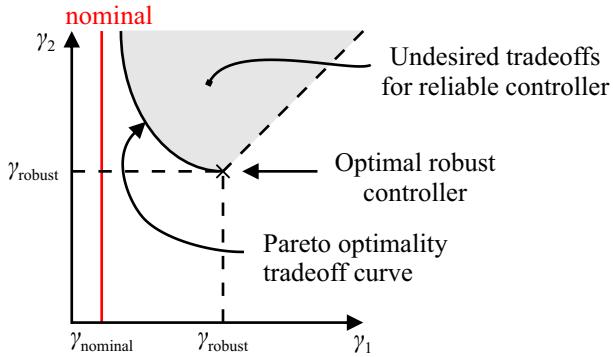


Figure 7.9: *Bounds on the γ values for the PFTC; γ_1 is associated with the fault-free case and γ_2 is associated with the faulty case. The gray area contains all feasible PFTCs, where a PFTC on the Pareto optimality tradeoff curve is desired. The Pareto optimality tradeoff curve illustrates the optimum tradeoff between conflicting requirements.*

The following requirements to γ_1 (normal system) and γ_2 (faulty system) are deduced from Figure 7.9:

- $\gamma_1 < \gamma_2$: The gray area in the figure is the region where $\gamma_1 < \gamma_2$, i.e. the region for reliable controllers defined to have better performance in the fault-free case than in the faulty case.
- $\gamma_1 < \gamma_{\text{robust}} < \gamma_2$: By designing an optimal robust controller γ_{robust} is found. Hence, the upper bound on γ_1 and lower bound on γ_2 can be found, since $\gamma_1 < \gamma_{\text{robust}} < \gamma_2$ must be satisfied to lay on the Pareto optimality tradeoff curve.

- $\gamma_1 > \gamma_{\text{nominal}}$: The lower bound on γ_1 can be found by designing an optimal nominal LPV controller, since the fault-tolerant controller cannot achieve better performance than this.

For the considered problem, the bounds stated below are calculated for the two design problems and specify $\gamma_{\text{nominal}} < \gamma_1 < \gamma_{\text{robust}}$ and $\gamma_{\text{robust}} < \gamma_2$. Notice that the values for the two controllers cannot be directly compared, since their performance specifications are different.

- With $\omega_{g,\text{mes}}(t)$: $0.496 < \gamma_1 < 0.652$ and $0.652 < \gamma_2$.
- Without $\omega_{g,\text{mes}}(t)$: $0.566 < \gamma_1 < 0.608$ and $0.608 < \gamma_2$.

Based on this information, both passive fault-tolerant controllers are designed for $\gamma_2 = 1$, while γ_1 is minimized.

To find the optimal passive fault-tolerant controllers, γ_1 is minimized for different selections of constant Lyapunov matrices and at different initializations of the algorithm. A number of different approaches for solving the optimization problem have been tried. These are mentioned in the following list. In the first two items in the list, three different choices are possible.

1. Select the parameter-dependent Lyapunov matrix.
 - (a) Lyapunov matrix \mathbf{X} is dependent on θ .
 - (b) Lyapunov matrix \mathbf{Y} is dependent on θ .
 - (c) No Lyapunov matrix is dependent on θ .
2. Find half of the unknown variables in Eq. (7.29) on Page 107, since this turns the matrix inequality into an LMI in the remaining unknown variables.
 - (a) Find \mathbf{X} , $\hat{\mathbf{B}}_\Delta$, $\hat{\mathbf{D}}_\Delta$, by solving the necessary condition for \mathbf{X} shown in Eq. (7.32) on Page 108.
 - (b) Find \mathbf{Y} , $\hat{\mathbf{C}}_\Delta$, $\hat{\mathbf{D}}_\Delta$, by solving the necessary condition for \mathbf{Y} shown in Eq. (7.34) on Page 108.
 - (c) All matrices are known from the initial guess obtained by calculating an AFTC with $\gamma = 1$. This γ value is chosen to ensure feasibility of Eq. (7.29).
3. Solve Eq. (7.29) while minimizing $\gamma_1 = \gamma_2$ until the desired value for γ_2 is reached. This is done by switching between having \mathbf{X} , $\hat{\mathbf{B}}_\Delta$, $\hat{\mathbf{D}}_\Delta$ or \mathbf{Y} , $\hat{\mathbf{C}}_\Delta$, $\hat{\mathbf{D}}_\Delta$ as unknown variables.
4. Solve Eq. (7.29) while minimizing γ_1 and switching between having \mathbf{X} , $\hat{\mathbf{B}}_\Delta$, $\hat{\mathbf{D}}_\Delta$ or \mathbf{Y} , $\hat{\mathbf{C}}_\Delta$, $\hat{\mathbf{D}}_\Delta$ as unknown variables.

By following this procedure the values in Table 7.3 are obtained for γ_1 and γ_2 . The performance specifications are made such that the two controllers have approximately the same pitch actuator usage.

From Table 7.3, γ_1 appears to be smallest when $\mathbf{Y}(\theta)$ is parameter-dependent. Therefore, this configuration is chosen for both implemented controllers. To implement the controllers they have to be discretized. Hence, γ_1 is slightly increased to obtain similar behaviors for the continuous and discrete controllers.

Having selected sufficiently large γ values to make the controllers functional in a sampled set-up, the obtained performance degradation is evaluated when high air content in the hydraulic oil is introduced. This is shown in Figure 7.10 that displays a subplot valid for each controller. From the figure it is clear that the performance of the controllers degrades as the air content increases, as expected.

\mathbf{X}	\mathbf{Y}	Initialization	$f_{sw} = 0$		$f_{sw} = 1$	
			γ_1	γ_2	γ_1	γ_2
$\mathbf{Y} = \mathbf{Y}_0$	$\mathbf{X} = \mathbf{X}(\theta)$	Necessary condition for X	0.635	1	0.597	1
$\mathbf{Y} = \mathbf{Y}(\theta)$	$\mathbf{X} = \mathbf{X}_0$	Necessary condition for X	0.541	1	0.571	1
$\mathbf{Y} = \mathbf{Y}_0$	$\mathbf{X} = \mathbf{X}_0$	Necessary condition for X	0.619	1	0.599	1
$\mathbf{Y} = \mathbf{Y}_0$	$\mathbf{X} = \mathbf{X}(\theta)$	Necessary condition for Y	0.658	1	0.597	1
$\mathbf{Y} = \mathbf{Y}(\theta)$	$\mathbf{X} = \mathbf{X}_0$	Necessary condition for Y	0.540	1	0.571	1
$\mathbf{Y} = \mathbf{Y}_0$	$\mathbf{X} = \mathbf{X}_0$	Necessary condition for Y	0.619	1	0.599	1
$\mathbf{Y} = \mathbf{Y}_0$	$\mathbf{X} = \mathbf{X}(\theta)$	Initial guess	0.616	1	0.596	1
$\mathbf{Y} = \mathbf{Y}(\theta)$	$\mathbf{X} = \mathbf{X}_0$	Initial guess	0.544	1	0.571	1
$\mathbf{Y} = \mathbf{Y}_0$	$\mathbf{X} = \mathbf{X}_0$	Initial guess	0.632	1	0.599	1

Table 7.3: Values of γ_1 and γ_2 for the PFTC for different selections of variables in the gain-scheduled control problem. γ_1 is the upper bound on the disturbance attenuation in the fault-free case, while γ_2 is the upper bound on the disturbance attenuation in the faulty case. Notice that the γ values for $f_{sw} = 0$ and $f_{sw} = 1$ cannot be directly compared, since different performance specifications are utilized.

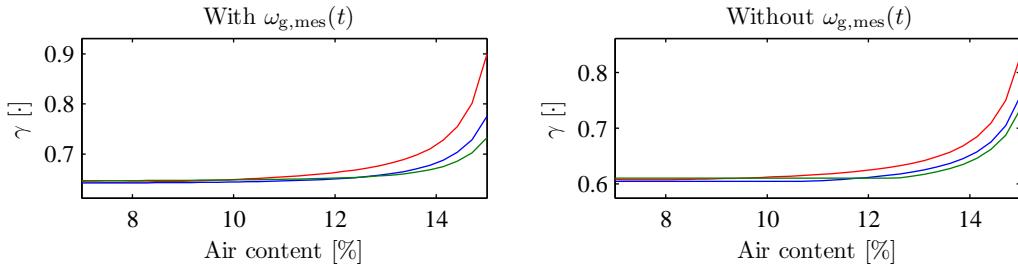


Figure 7.10: The performance level γ as a function of the air content level in the pitch system for the two passive fault-tolerant controllers. Notice that the values of γ in the two subplots should not be compared to each other since the optimization problems are solved subject to different performance specifications. The value of γ depends on the operating point and its rate, so for each controller γ is shown for $\theta_{op} = 21.5$ m/s and for $\dot{\theta}_{op} = -2$ m/s² (red), $\dot{\theta}_{op} = 0$ m/s² (blue), and $\dot{\theta}_{op} = 2$ m/s² (green).

Verification of PFTCS

To verify that the controllers are able to operate in the entire parameter space and at the rate bounds, simulations are conducted where the wind turbine model is forced between the extremes of the parameter space. The first 50 s of the simulations are shown in Figure 7.7 on Page 113, for both normal air content (blue) and high air content (red) in the hydraulic oil, to enable a comparison of the control with and without a faulty pitch system. Likewise, the consequence of not using the generator speed measurement is evaluated by comparing the left and right subplots.

From the figure it is concluded that the PFTCs are able to operate in the entire operating region specified in the design. Additionally, no significant difference is apparent on either $\beta(t)$ or $\omega_g(t)$ in the simulations due to the changing air content. In contrast, a significant performance degradation happens when $\omega_{g,mes}(t)$ is not available. This degradation is though expected due to the degradation of the available measurement information.

In the next section a comparison between the PFTCs and AFTCs is described.

7.7 Comparison of AFTCS and PFTCS

The purpose of this section is to compare the active and passive fault-tolerant control systems by means of design complexity and performance when applied to the wind turbine model.

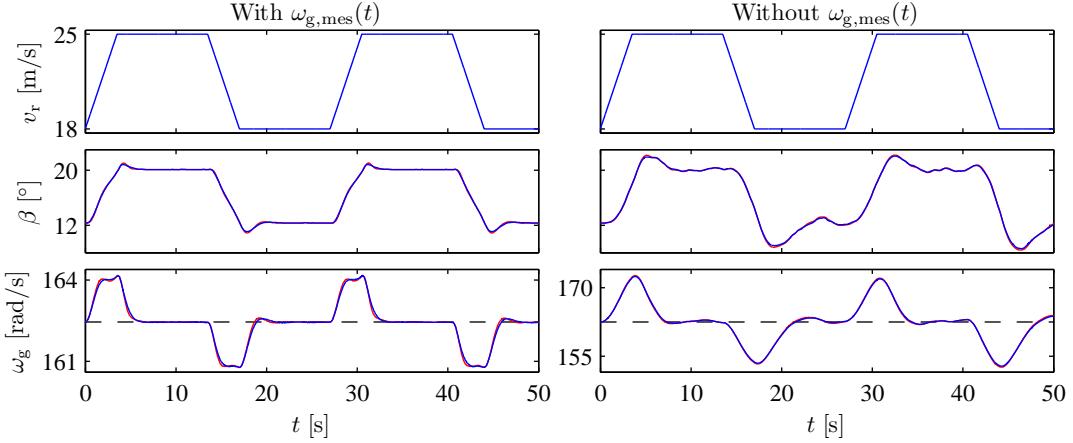


Figure 7.11: *Simulation results where the PFTC is forced between the extremes of the operational range of the wind speed and at its rate limits. The simulations are conducted at both normal air content (blue) and high air content (red).*

The design procedures of the AFTC and the PFTC are compared according to a number of comparison criteria shown in Table 7.4. Each comparison criterion is discussed further in the following text.

Comparison Criterion	AFTC	PFTC
Design problem	Convex optimization	Non-convex optimization
Conservatism	Lower than for PFTC	Higher than for AFTC
Dependent on fault diagnosis	Yes	No
Implemental complexity	Higher than for PFTC	Lower than for AFTC

Table 7.4: *Comparison of the AFTC and PFTC in terms of controller design.*

The procedures for designing active and passive fault-tolerant controllers look very similar, when comparing Eq. (7.5) on Page 102 and Eq. (7.24) on Page 106, which should be solved for realizing the controllers. Since the AFTC design is based on a convex optimization, in contrast to the non-convex optimization for the PFTC, this design problem is much easier to solve. The non-convex nature of the PFTC optimization problem implies that γ is not ensured to converge towards the global minimum, which makes the solution sensitive to the initialization of the algorithm, as indicated by Table 7.3. This makes the active fault-tolerant controller favorable in terms of solving the optimization problem.

The active fault-tolerant controller is able to schedule along the trajectory of the fault. This makes it less conservative than the passive fault-tolerant controller, which has to maintain the same controller in case of a fault. Therefore, it opens the opportunity for the active fault-tolerant controller to have superior performance compared to the passive fault-tolerant controller.

In return for the conservatism introduced in the passive fault-tolerant controller, in terms of being independent on knowledge about the fault, the risk of making wrong decisions in the fault diagnosis is eliminated. Additionally, the time spend on designing the fault diagnosis algorithms can be saved. In this light, the passive fault-tolerant controller is the favored choice, especially if the fault is difficult to diagnose.

Having compared the design procedures and the general advantages and disadvantages of the active and passive fault-tolerant controllers, the final step in the comparison is to compare simulation results obtained by the controllers and assess their performance.

Simulation Results

To verify and compare the performance of the fault-tolerant controllers, simulations of duration 5,000 s are conducted both with the normal air content level of 7% and at a level of 15%, at wind speeds ranging from 18 m/s to 25 m/s. Notice that air content levels in between the extreme values are already tested in relation with the simulations referred to in Figure 7.7 on Page 113 and Figure 7.11, but the intermediate values are not included in this comparison due to the slow evolution of the fault.

Results for the Controllers relying on the Generator Speed Measurement

The first 50 s of the simulations of the controllers relying on the generator speed measurement are shown in Figure 7.12, for both AFTC and PFTC, to enable a comparison of the two controllers. Additionally, the gray lines display the simulation results of the reference controller, which is not designed to handle the changing dynamics of the pitch system and therefore performs poorly in this case. The performance measures obtained for the 5,000 s simulations are stated in Table 7.5.

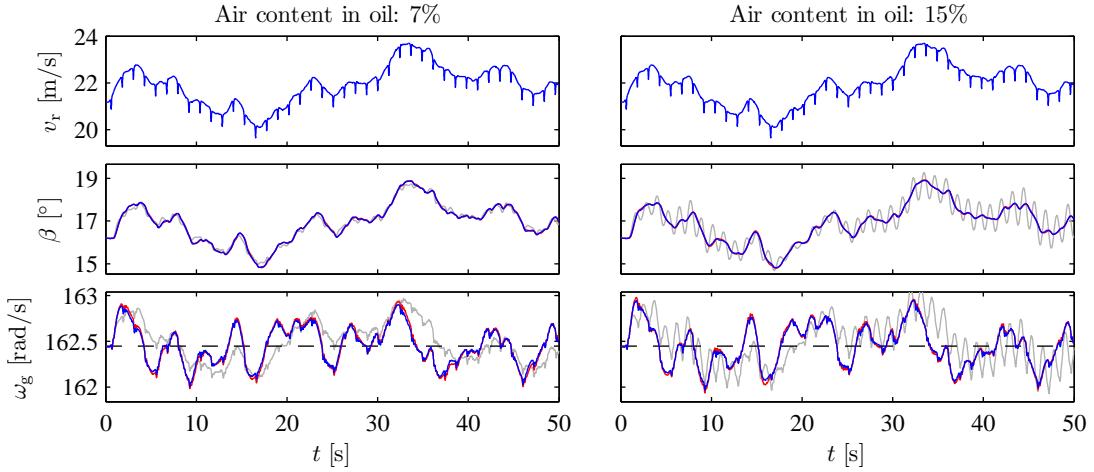


Figure 7.12: *Simulation results of the AFTC (blue) and the PFTC (red) conducted at both normal and high air content levels in the hydraulic oil. The behaviors of the fault-tolerant controllers can be compared to the operation of the reference controller (gray). The considered controllers possess fault-tolerant capabilities against high air content in the pitch system.*

Controller	Air content in oil: 7%		Air content in oil: 15%	
	$\int_0^t (\omega_{g,e}(\tau))^2 d\tau$	$\int_0^t \dot{\beta}^2(\tau) d\tau$	$\int_0^t (\omega_{g,e}(\tau))^2 d\tau$	$\int_0^t \dot{\beta}^2(\tau) d\tau$
Active	1.00	1.00	1.19 (1.00)	1.10 (1.00)
Passive	1.18	0.94	1.29 (1.08)	1.14 (1.03)
Active ($\hat{\theta}_f$ incorrect)	1.13	0.98	1.06 (0.89)	1.42 (1.29)
Reference	1.59	1.17	1.79 (1.50)	10.71 (9.67)

Table 7.5: *Speed tracking errors and pitch actuator usages obtained for simulations having duration 5,000 s for controllers, which rely on the generator speed, i.e. $f_{sw} = 0$. The results are normalized to the performance of the active fault-tolerant controller. The numbers in parentheses denote a local normalization. Note that the third controller mentioned is the active fault-tolerant controller, where the fault diagnosis algorithm is forced to make a false positive or false negative decision, to evaluate the consequence of incorrect fault estimation.*

Since the fault-tolerant controllers are designed based on the same structure and specification, they behave quite similar, which is apparent from Figure 7.12. When comparing the tracking errors of the generator speed in the last subplots in the figure, or in Table 7.5, it is seen that the active

fault-tolerant controller has better performance, especially in the fault-free case. The reason being is that this controller is less conservative, since controller reconfiguration is offered based on the fault diagnosis signal. In line with this explanation it is further concluded that the passive fault-tolerant controller has a smaller actuator usage than the active fault-tolerant controller.

In Table 7.5 the performance measures are also stated for the reference controller and for the active fault-tolerant controller when using an incorrect estimate of the fault. The first result is that the fault-tolerant controllers are superior compared to the reference controller in both performance measures, which enhances the great performance of the multi-variable LPV controller compared to the traditional PI controller. The second result is that when a wrong estimate of the fault is fed to the active fault-tolerant controller, the performance is generally degraded. This especially appears from the large pitch actuator usage in the situation of false negative diagnosis.

Results for the Controllers independent of the Generator Speed Measurement

The first 50 s of the simulations of the controllers, which do not rely on the generator speed measurement, are shown in Figure 7.13, for both AFTC and PFTC, to enable a comparison of the two controllers. The performance measures obtained for the 5,000 s simulations are stated in Table 7.6.

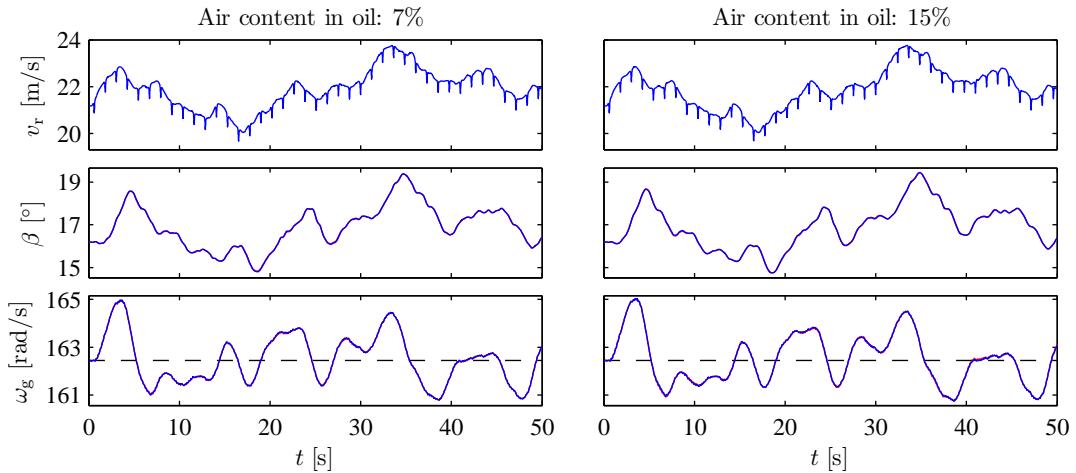


Figure 7.13: Simulation results of the AFTC (blue) and the PFTC (red) conducted at both normal and high air content levels in the hydraulic oil. The considered controllers are independent of the generator speed measurement, and possess fault-tolerant capabilities against a fixed generator speed measurement and against high air content in the pitch system.

Controller	Air content in oil: 7%		Air content in oil: 15%	
	$\int_0^t (\omega_{g,e}(\tau))^2 d\tau$	$\int_0^t \dot{\beta}^2(\tau) d\tau$	$\int_0^t (\omega_{g,e}(\tau))^2 d\tau$	$\int_0^t \dot{\beta}^2(\tau) d\tau$
Active	29.8 [1.00]	1.19 [1.00]	31.0 {1.00}	1.28 {1.00}
Passive	29.5 [0.99]	1.17 [0.98]	30.9 {1.00}	1.30 {1.01}

Table 7.6: Speed tracking errors and pitch actuator usages obtained for simulations having duration 5,000 s for controllers, which do not rely on the generator speed, i.e. $f_{sw} = 1$. The results are normalized to the performance of the nominal active fault-tolerant controller in Table 7.5. The numbers in the square parentheses and curly braces denote local normalizations.

Since the fault-tolerant controllers are designed based on the same structure and specification, they behave quite similar, as depicted by Figure 7.13 and Table 7.6. The considered situation seems to be dominated by the absence of the generator speed measurement, whereas increased air content in the pitch system has only a minor influence on the performance of the system.

As desired by the design, and confirmed by comparing Table 7.5 and Table 7.6, the pitch actuator usage is similar for the controllers using and not using the generator speed measurement. On the

other hand, the speed tracking error is dramatically increased, due to the poor quality of the rotor speed measurement compared to the measurement of the generator speed.

In addition to the performance degradation already mentioned, the drive train stress damper is deactivated when running without the generator speed measurement, consequently increasing drive train stress as discussed in Section 6.4. The reason for disabling the stress damper is that the rotor speed measurement does not possess the information about the drive train oscillations. On average, the drive train stress is increased by approximately 40%, compared to the controllers using the drive train stress damper. Hence, it should be considered if it is desired to operate the wind turbine at normal operating conditions, when the generator speed measurement is lost. Furthermore, no advantage is obtained by using the active fault-tolerant controllers for this particular scenario.

In this chapter active and passive fault-tolerant controllers have been designed for accommodating changed pitch system dynamics and the loss of the generator speed measurement in full load operation. The methods utilized for designing both the AFTC and PFTC are based on solving optimization problems. The optimization problem for the active fault-tolerant controller is the easiest to solve, and is therefore preferable. The passive fault-tolerant controller is independent on fault diagnosis; hence, it has no risk of false diagnosis of faults.

In the considered design the AFTC has a slightly better performance than the PFTC, in the situation where the generator speed measurement is available. In the situation where the generator speed measurement is not available both types of controllers show similar performance, probably because the noise on the available measurements is the limiting factor for the controllers. From this it is concluded that active fault-tolerant controllers should be utilized if performance is more important than safety, or if the fault diagnosis algorithms have small probabilities of making false decisions. In contrast, passive fault-tolerant controllers should be chosen if safety is very important, or if the fault is not the limiting factor for the controller.

The next chapter addresses an integration test, where the fault diagnosis algorithms, signal correction algorithms, and the fault-tolerant LPV controllers are combined and tested subject to multiple simultaneous faults.

8 Integration Test

The purpose of this chapter is to combine the fault diagnosis algorithms, signal correction algorithms, and the fault-tolerant LPV controllers in one fault-tolerant control system, and test it exposed to multiple simultaneous faults. Since similar results are expected when applying both the active and passive fault-tolerant LPV controllers, only the active fault-tolerant control system is used in this test. This is chosen as this controller depends on most fault diagnosis algorithms, which can then be tested at the same time.

The aim of the integration test is to confirm that it is possible to diagnose and accommodate multiple simultaneous faults. For this to happen, each diagnosis algorithm should be robust towards the other considered faults in the system to avoid making an erroneously detection or estimation, which could degrade the performance of the control system. The integration test is conducted according to Figure 8.1, which is thoroughly explained in the list below.

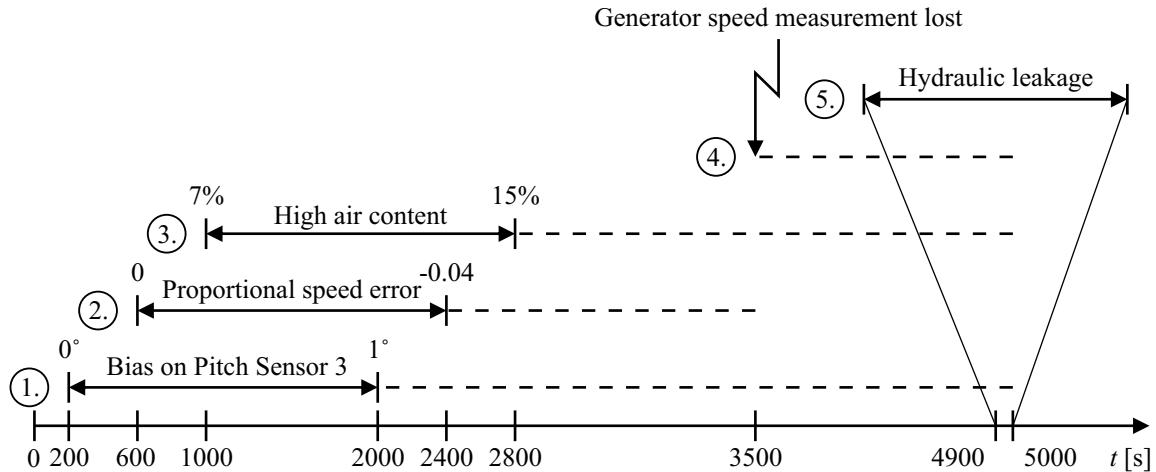


Figure 8.1: Specification of the sequence of faults in the integration test. The incipient faults are introduced where solid lines appear, while the values of the faults are held thereafter illustrated using dashed lines. The loss of the generator speed measurement is an abrupt fault; hence, it is illustrated using a lightning. Notice that the dashed line for the proportional speed error on the generator speed measurement disappears when the measurement is lost, since it makes no sense after this event.

1. A bias of 1° is introduced on the pitch sensor associated with Blade 3 from $t = 200$ s to $t = 2,000$ s. After the introduction of the fault it is held until the end of the simulation.
2. A proportional error of -0.04 is introduced on the generator speed measurement from $t = 600$ s to $t = 2,400$ s and is held constant thereafter. The fault changes the sensor gain from 1 to 0.96. From the time where the proportional error is introduced two faults exist, which should be handled by the signal correction algorithms.
3. The air content in the hydraulic oil is increased from 7% to 15% in between $t = 1,000$ s and $t = 2,800$ s. This fault should be handled by the active fault-tolerant LPV controller, which relies on the fault diagnosis algorithm estimating the dynamics of the pitch system.
4. The generator speed measurement is lost at $t = 3,500$ s; hence, a switch between the two active fault-tolerant LPV controllers should be performed. This requires that all the fault diagnosis algorithms are able to operate without the generator speed measurement.
5. A hydraulic leakage makes the pressure of the hydraulic oil drop to 50% of the normal pressure during 100 s. After the detection of this fault, the simulation is terminated.

The simulation is conducted using wind speeds ranging from 18 m/s to 25 m/s. The result of the simulation is shown in Figure 8.2.

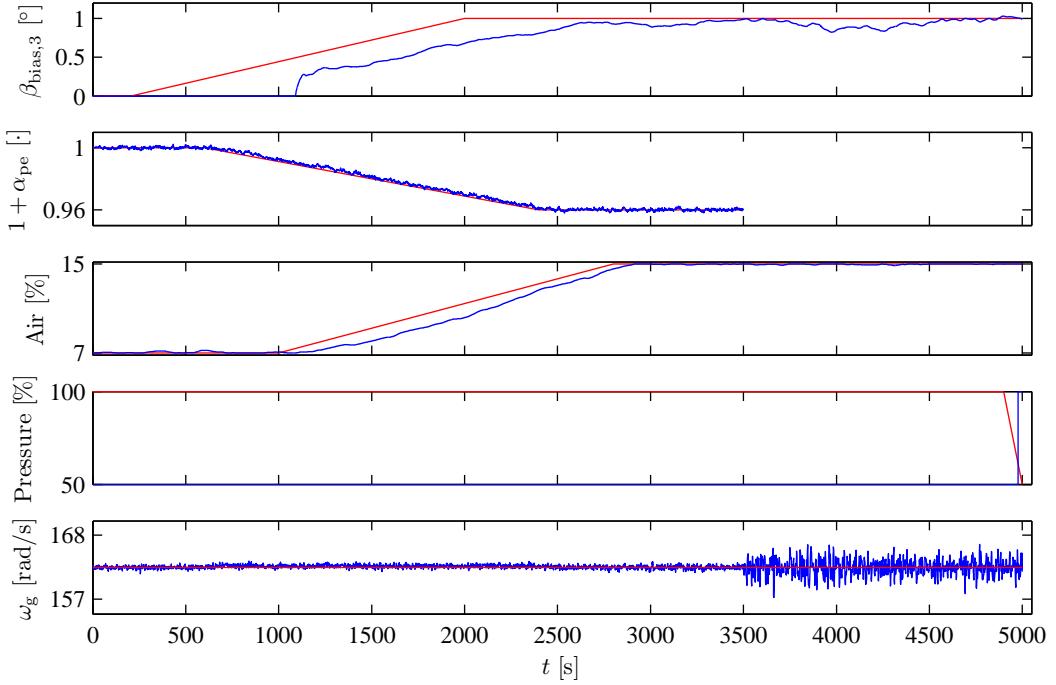


Figure 8.2: *Simulation result of the integration test. In the three upper subplots, the estimated parameters (blue) and the correct values of the parameters (red) are shown. In the fourth subplot, the detection signal for hydraulic leakage (blue) and the correct value of the pressure (red) are shown. The bottom subplot shows the generator speed (blue) and the generator speed reference (red).*

The simulation result is commented below, according to the numbering utilized in Figure 8.1.

1. The bias on Blade 3 is detected and isolated at $t = 1,189$ s. Hereafter, the bias is estimated and accommodated until the end of the simulation, using the reconfigurable extended Kalman filter. The bias is correctly estimated even though the generator speed measurement is lost at $t = 3,500$ s.
2. A proportional error on the generator speed measurement is detected at $t = 731$ s. After the detection, the sensor gain is estimated and accommodated until $t = 3,500$ s, where the generator speed measurement is lost. The accommodation of the proportional error implies that the generator speed is kept at its desired set-point throughout the simulation.
3. The air content in the hydraulic oil is estimated satisfactory throughout the simulation, and the estimate is fed to the LPV controller adapting it to the fault.
4. When the generator speed measurement is lost, a switch between the two active fault-tolerant LPV controllers is successfully accomplished. After the switch the generator speed starts fluctuating more, since the performance of the controller is degraded, in agreement with Table 7.6 on Page 119.
5. The hydraulic leakage introduced at $t = 4,900$ s is detected 76 s later. The detection time is increased compared to the detection times shown in Figure 5.17 on Page 71, since the pitch system already operates at a degraded performance when the fault is introduced.

As outlined above, the integration test has shown that the designed fault-tolerant control system is capable of diagnosing and accommodating multiple simultaneous faults. Generally, the performance of the control system is not affected visibly by the faults; the only fault that decisively affects the performance of the system is the loss of the generator speed measurement. This is expected, since the available measurement information is dramatically reduced, when this fault occurs.

9 Conclusion

In this project fault diagnosis and fault-tolerant control algorithms are developed for improving the reliability of wind turbines. The study is based on a model of a variable-speed, variable-pitch 4.8 MW wind turbine, which represents a realistic but fictitious wind turbine, to which the collaborator kk-electronic a/s has provided the parameters. The faults considered in the project are chosen based on a severity and occurrence analysis, in which the most frequent and severe faults are identified. The analysis primarily focuses on sensor and actuator faults, which are included in the model of the wind turbine.

In the diagnosis of the faults, model-based fault diagnosis algorithms are primarily developed, due to their improved resilience towards making incorrect decisions compared to signal-based approaches. Additionally, only the already available sensor information is utilized in the diagnosis of the faults. To obtain a fault-tolerant control system, different approaches are utilized dependent on the nature of the faults. For faults that affect the dynamics of the system, active and passive fault-tolerant controllers are designed and compared.

To access the performance of the designed algorithms, Monte Carlo simulations are performed to evaluate the robustness of the algorithms, where this is considered necessary.

Modeling

To facilitate a model-based approach in the design of the fault diagnosis and fault-tolerant control algorithms, a non-linear model of a variable-speed, variable-pitch wind turbine is set up. Additionally, the model acts as a simulation model for testing the designed algorithms.

The model is based on a static model of the aerodynamics, a two-mass model of the drive train, an electromechanical model of the generator, dynamic actuator models, and zero-mean Gaussian distributed measurement noises. The parameters for the wind turbine model are provided by kk-electronic a/s, similar applies for the variances of the additive measurement noises. The input to the model is generated by a wind model, which includes wind shear, tower shadow, and turbulence.

The aerodynamics of the wind turbine is non-linear and is described in form of a lookup table, where the efficiency of the aerodynamics is determined from the pitch angles of the blades and the tip-speed ratio. This part of the model is non-linear and introduces parameters that vary dependent on the operating conditions.

Fault Analysis

A fault analysis is performed in order to determine the faults which should be considered in this project. First, a number of possible component faults are chosen and their propagations through the system are determined by describing their effects on the surrounding components of the system. Subsequently, the severity of the end-effects and the occurrence rates of the faults are estimated to select the faults of highest priority.

The frequency of the faults are approximated based on statistics reported in the literature, whereas the severities of their end-effects are determined based on simulations. For conducting these simulations, a reference controller without fault-tolerant capabilities is designed based on information about an existing control system. To limit the number of faults to be handled during this project, it is decided to focus on the faults related to the pitch sensors, pitch actuators, and generator speed sensor.

The motivation behind selecting faults related to the pitch system, which e.g. cause rotor unbalance, is that these faults increase fatigue loads on the wind turbine structure. It is further seen that changed dynamics of the pitch system, caused by low pressure or high air content in the hydraulic oil, may result in an unstable closed-loop system. Finally, the main controllers in the entire operating range of the wind turbine depend solely on the measured generator speed. Hence, it is essential to diagnose and accommodate any troubles regarding this particular measurement.

Fault Diagnosis

It is decided to apply the same overall structure to all diagnosis algorithms, to enable diagnosis of multiple simultaneous faults. The fault diagnosis system consists of fault detection and isolation algorithms that determine the current state of the system and reconfigure an extended Kalman filter, which is able to provide a fault-corrected state estimate at all times.

Both abrupt and incipient faults are considered; hence, different methods are utilized in the fault diagnosis algorithms. Generally, the abrupt faults are detected using hypothesis testing based methods, while the incipient faults are diagnosed by applying parameter estimation based methods. A complete list of the methods used for the diagnosis of the selected faults is shown in Table 9.1.

Component	Fault	Fault Diagnosis Method
Pitch sensor	Biased output	Diagnosis of unbalance in rotor plane
Pitch actuator	High air content in oil	Multiple-model parameter estimation
	Pump wear	
	Hydraulic leakage	CUSUM test
	Valve blockage	Active fault diagnosis hypothesis testing (<i>PL</i>)
	Pump blockage	Passive fault diagnosis hypothesis testing (<i>FL</i>)
Generator speed sensor	Proportional error	Peak detection and estimation of 3P frequency
	Fixed output	Detection of zero measurement noise
	No output	None (<i>the control system is notified</i>)

Table 9.1: *Overview of the selected faults and the used fault diagnosis methods. Notice that different methods are used for diagnosing valve blockage and pump blockage in the partial load region (PL) and in the full load region (FL).*

Based on the performed verifications, it is concluded that it is possible to design model-based fault diagnosis algorithms for all the considered faults, even though they are applied on a parameter-varying system. This confirms that model-based fault diagnosis is relevant with application to wind turbine control. Note that the designed diagnosis algorithms can be used for condition monitoring purposes alone or in relation to an active fault-tolerant control system.

Fault-Tolerant Control

In the design of the fault-tolerant control system the faults are divided into two categories: faults that do not affect the dynamics of the system and faults that affect the dynamics of the system.

The faults that do not affect the dynamics of the system are accommodated by correcting the measurement and reference signals, based on information provided by the fault diagnosis algorithms. This enables the fault-tolerant control system to be designed independent of the controller structure and without affecting the nominal performance of the control system.

Faults that affect the dynamics of the system are accommodated using both active and passive fault-tolerant control, to enable a comparison of the two methods. The main difference between these methods is that the active fault-tolerant controller depends on the fault diagnosis algorithms, while the passive fault-tolerant controller is independent of these algorithms. Both fault-tolerant controllers are LPV controllers, which are based on a common LPV system description, accounting for the parameter-varying nature of the wind turbine.

A complete list of the fault-tolerant control methods is shown in Table 9.2. From the table it is clear that not all faults are accommodated. The effect introduced by pump wear can easily be handled by applying the method used for accommodating high air content in the hydraulic oil, where the parameter variations are considered in the design procedure of an active or a passive fault-tolerant controller. The effects of hydraulic leakage, valve blockage, and pump blockage are not accommodated, since they compromise the controllability of the wind turbine; hence, the wind turbine should be shut down.

Verifications of the fault-tolerant controllers show that the signal correcting algorithms allow the control system to regain nominal performance after fault accommodation. The active and passive

Component	Fault	Fault Accommodation Method
Pitch sensor	Biased output	Signal correction of measurement and reference signals
Pitch actuator	High air content in oil	Active and passive fault-tolerant LPV control
	Pump wear	None
	Hydraulic leakage	
	Valve blockage	
	Pump blockage	
Generator speed sensor	Proportional error	Signal correction of measurement signal
	Fixed output	Signal correction of measurement signal (<i>PL</i>)
	No output	Active and passive fault-tolerant LPV control (<i>FL</i>)

Table 9.2: *Overview of the selected faults and the used fault-tolerant control methods. Notice that different methods are used for accommodating a fixed or no output from the generator speed sensor in the partial load region (*PL*) and in the full load region (*FL*).*

fault-tolerant LPV controllers show improved performance compared to the reference controller in both the fault-free and faulty scenarios, considering a fixed generator speed measurement and increased air content in the hydraulic oil of the pitch system.

It is considered that the active fault-tolerant controller is suited for controlling the wind turbine in case of high air content in the oil, but that the passive fault-tolerant controller is preferable when the generator speed measurement is unavailable, since this effect dominates the control problem resulting in similar performance of both controllers. Generally, the active fault-tolerant controller approach shows slightly better performance, since it is adapted to the condition of the system, by exploiting information from the fault diagnosis algorithms.

Overall Conclusion

Based on the results obtained in the project, it is concluded that the reliability of wind turbines can be improved by applying fault diagnosis and fault-tolerant control. This is possible since the fault-tolerant capabilities allow wind turbines to continue operation in cases of sensor and actuator faults. Especially for incipient faults, the developed fault-tolerant control system benefits from its model-based approach, which allows faults to be accommodated based on parameter estimates. This improves the performance of the wind turbine control system in terms of minimizing mechanical loads and increasing efficiency.

With respect to handling multiple simultaneous faults, the integration test in Chapter 8 confirms that the control system is capable of controlling the wind turbine even though it is exposed to multiple simultaneous faults. This is a consequence of the common structure used for the fault diagnosis algorithms, which allows the diagnosis algorithms to maintain operation by exploiting a fault-corrected state estimate.

Limitations and Recommendations for Future Work

In order to make the developed algorithms applicable on real wind turbines, some further work have to be conducted, to guarantee that the algorithms perform satisfactorily. This work should include an investigation of the robustness of the algorithms towards the unmodeled phenomena. These are expected to originate primarily from the unmodeled aerodynamics and the simplified blade model, since a static aerodynamic model is used and since the blades are assumed to be stiff. These simplifications may affect the designed diagnosis algorithms negatively, since the unmodeled modes may disturb the fault diagnosis system.

For future work, the recommendation is to exploit model-based approaches for fault diagnosis purposes. Model-based fault diagnosis enables accommodation of incipient faults, in contrast to

Conclusion

the present signal-based approaches, since parameter estimation is possible. This may allow wind turbines to continue operation in situations where they are currently being shut down.

With respect to fault-tolerant control, the choice between active and passive fault-tolerant approaches is problem-specific. It is recommended to utilize an active fault-tolerant controller, when this allows significantly improved performance compared to a passive fault-tolerant controller. This is expected to happen in situations where a fault causes severe changes in the dynamics of the system and where a reliable diagnosis system exists. Otherwise, it is recommended to apply passive fault-tolerant controllers, since they have no risk of making wrong decisions.

Bibliography

- [Aalborg University and RISØ National Laboratory, 2005] Aalborg University and RISØ National Laboratory. Wind Turbine Blockset, 2005.
http://www.iet.aau.dk/Research/research_prog/wind_turbine/Projects/SimPlatformPrj/download/wind_turbine_beta.zip.
- [Andersen and Pedersen, 2007] Palle Andersen and Tom S. Pedersen. Modeldannelse. Lecture notes, February 2007.
- [Apkarian and Adams, 1998] Pierre Apkarian and Richard J. Adams. Advanced Gain-Scheduling Techniques for Uncertain Systems. *IEEE Transactions on Control Systems Technology*, 6(1):21–32, January 1998.
- [Apkarian *et al.*, 1995] Pierre Apkarian, Pascal Gahinet, and Greg Becker. Self-Scheduled Hinfin Control of Linear Parameter-Varying Systems: A Design Example. *Automatica*, 31(9):1251–1261, September 1995.
- [Bak, 2000] Thomas Bak. Lecture Notes - Estimation and Sensor Information Fusion. Lecture notes, November 2000.
- [Bianchi *et al.*, 2007] Fernando D. Bianchi, Hernán De Battista, and Ricardo J. Mantz. Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design. Springer, 2007. ISBN: 1-84628-492-9.
- [Blanke *et al.*, 2006] Mogens Blanke, Michel Kinnaert, Jan Lunze, and Marcel Staroswiecki. Diagnosis and Fault-Tolerant Control. Springer, 2. edition, 2006. ISBN: 3-540-35652-5.
- [Campbell and Nikoukhah, 2004] Stephen L. Campbell and Ramine Nikoukhah. Auxiliary Signal Design for Failure Detection. Princeton University Press, 2004. ISBN: 0-691-09987-1.
- [Dixit and Suryanarayanan, 2005] Amit Dixit and Shashikanth Suryanarayanan. Towards Pitch-Scheduled Drive Train Damping in Variable-Speed, Horizontal-Axis Large Wind Turbines. In *Proceedings of 44th IEEE Conference on Decision and Control*, pages 1295–1300, Seville, Spain, December 2005.
- [Dolan and Lehn, 2006] Dale S. L. Dolan and Peter W. Lehn. Simulation Model of Wind Turbine 3p Torque Oscillations due to Wind Shear and Tower Shadow. *IEEE Transactions on Energy Conversion*, 21(3):717 – 724, September 2006.
- [Donders, 2002] Stijn Donders. Fault Detection and Identification for Wind Turbine Systems: a closed-loop analysis. Master's thesis, University of Twente Faculty of Applied Physics Systems and Control Engineering, 2002.
- [DOWEC team, 2002] DOWEC team. Estimation of Turbine Reliability figures within the DOWEC project. Technical report, DOWEC, 2002.
- [East and Lantz, 2005] William East and Brian Lantz. Notch Filter Design. Technical report, Laser Interferometer Gravitational-Wave Observatory, August 2005.
- [Esbensen *et al.*, 2008] Thomas Esbensen, Christoffer Eg Sloth, Michael Odgaard Niss, and Brian Thorarinns Jensen. Joint Power and Speed Control of Wind Turbines. Technical report, Aalborg University, Department of Electronic Systems, 2008.

Bibliography

- [Gill, 2008] George Gill. Maintaining the wind turbine revolution. *Lube Report*, 8(34), August 2008.
- [Grewal and Andrews, 2001] Mohinder S. Grewal and Angus P. Andrews. Kalman Filtering: Theory and Practice Using MATLAB. John Wiley & Sons, 2. edition, 2001. ISBN: 0-47139-254-5.
- [GWEC, 2009] GWEC. Global Wind 2008 Report. Technical report, Global Wind Energy Council, 2009.
- [Hallouzi, 2008] Redouane Hallouzi. Multiple-Model Based Diagnosis for Adaptive Fault-Tolerant Control. PhD thesis, Delft University of Technology, 2008.
- [Hameed *et al.*, 2009] Z. Hameed, Y.S. Hong, Y.M. Cho, S.H. Ahn, and C.K. Song. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable Energy Reviews*, 13(1):1–39, January 2009.
- [Hammerum, 2006] Keld Hammerum. A Fatigue Approach to Wind Turbine Control. Master's thesis, Technical University of Denmark, 2006.
- [Hassanpour *et al.*, 2004] Hamid Hassanpour, Mostefa Mesbah, and Boualem Boashash. EEG spike detection using time-frequency signal analysis. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004*, volume 5, pages V–421 – V–424, Montreal, QC, May 2004.
- [How Stuff Works, 2006] How Stuff Works. Horizontal-axis Turbine, 2006. <http://static.howstuffworks.com/gif/wind-power-horizontal.gif>.
- [Hristov *et al.*, 2000] Tihomir S. Hristov, Scott D. Miller, and Carl A. Friehe. Linear Time-Invariant Compensation of Cup Anemometer and Vane Inertia. *Boundary-Layer Meteorology*, 97(2):293–307, November 2000.
- [Ireson *et al.*, 1996] William Grant Ireson, Clyde F. Coombs, and Richard Y. Moss. Handbook of Reliability Engineering and Management. McGraw-Hill Professional, 2. edition, 1996. ISBN: 0-07-012750-6.
- [Izadi-Zamanabadi, 1999] Roozbeh Izadi-Zamanabadi. Fault-tolerant Supervisory Control - System Analysis and Logic Design. PhD thesis, Aalborg University Department of Control Engineering, 1999.
- [Izadi-Zamanabadi, 2004] Roozbeh Izadi-Zamanabadi. Fault-tolerant Control. Lecture slides, 2004.
- [Jabbari, 1997] Faryar Jabbari. Output Feedback Controllers for Systems with Structured Uncertainty. *IEEE Transactions on Automatic Control*, 42(5):715–719, May 1997.
- [Johnson *et al.*, 2006] Kathryn E. Johnson, Lucy Y. Pao, Mark J. Balas, and Lee J. Fingersh. Control of Variable-Speed Wind Turbines: Standard and Adaptive Techniques for Maximizing Energy Capture. *IEEE Control Systems Magazine*, 26(3):70–81, June 2006.
- [Kithil, 2008] Richard Kithil. Case Study of Lightning Damage to Wind Turbine Blade. Technical report, National Lightning Safety Institute, June 2008.
- [Krohn, 2002] Soren Krohn. Danish Wind Turbines: An Industrial Success Story. Technical report, Danish Wind Industry Association, February 2002.
- [Köse and Jabbari, 1997] I. Emre Köse and Faryar Jabbari. Control of LPV Systems with Partly-Measured Parameters. In *Proceedings of the 36th Conference on Decision and Control*, pages 972–977, San Diego, California USA, December 1997.
- [Liao *et al.*, 2003] Fang Liao, Jian Liang Wand, and Guang-Hong Yan. Reliable H_2 Static Output-Feedback Tracking Control Against Aircraft Wing/Control Surface Impairment. In *Proceedings of International Conference on Physics and Control 2003*, pages 112–117, St. Petersburg, Russia, August 2003.

- [Niemann and Stoustrup, 2005a] H. Niemann and J. Stoustrup. An Architecture for Fault Tolerant Controllers. *International Journal of Control*, 78(14):1091–1110, September 2005.
- [Niemann and Stoustrup, 2005b] H. Niemann and J. Stoustrup. Passive fault tolerant control of a double inverted pendulum - A case study. *Control Engineering Practice*, 13(8):1047–1059, August 2005.
- [Østergaard *et al.*, 2007] K.Z. Østergaard, P. Brath, and J. Stoustrup. Estimation of effective wind speed. *Journal of Physics: Conference Series*, 75, August 2007.
- [Peeters, 2006] Joris Peeters. Simulation of dynamic drive train loads in a wind turbine. PhD thesis, Katholieke Universiteit Leuven, 2006.
- [Quality Associates International, 2008] Quality Associates International. Process FMEA reference card, 2008.
<http://www.quality-one.com/files/downloads/PFMEArefcard.pdf>.
- [Ribrant, 2006] Johan Ribrant. Reliability performance and maintenance - A survey of failures in wind power systems. Master's thesis, KTH School of Electrical Engineering, 2006.
- [The Encyclopedia of Alternative Energy and Sustainable Living, 2005] The Encyclopedia of Alternative Energy and Sustainable Living. Wind Turbine, 2005.
http://www.daviddarling.info/encyclopedia/W/AE_wind_turbine.html.
- [Thiringer and Petersson, 2005] Torbjörn Thiringer and Andreas Petersson. Control of a Variable-Speed Pitch-Regulated Wind Turbine. Technical report, Chalmers University of Technology, 2005.
- [van der Hooft and van Engelen, 2004] E.L. van der Hooft and T.G. van Engelen. Estimated Wind Speed Feed Forward Control for Wind Turbine Operation Optimisation. Technical report, Energy research Centre of the Netherlands, November 2004.
- [Vijayaraghavan, 2003] Giridharan Vilangadu Vijayaraghavan. A Taxonomy of E-commerce Risks and Failures. Master's thesis, Florida Institute of Technology, 2003.
- [Wei and Verhaegen, 2008] Xiukun Wei and Michel Verhaegen. Fault Detection of Large Scale Wind Turbine Systems: A Mixed H_∞/H_- Index Observer Approach. In *Proceedings of 16th Mediterranean Conference on Control and Automation*, pages 1675–1680, Ajaccio, France, June 2008.
- [Wind Energy News, 2007] Wind Energy News. Wind Energy is the World's Fastest Growing Energy Source, 2007.
<http://windenergynews.blogspot.com/2007/02/wind-energy-is-worlds-fastest-growing.html>.
- [Zhang and Jiang, 2003] Youmin Zhang and Jin Jiang. Bibliographical Review On Reconfigurable Fault-Tolerant Control Systems. In *Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, pages 265–276, Washington, D.C., USA, June 2003.

A

Severity and Occurrence Analysis

In this appendix severity and occurrence indices are provided for the failures, which can happen to the sub-models introduced in Figure 4.2 on Page 26. The appendix therefore provides the background for the selected component faults in Section 4.3. The occurrence indices are derived from the surveys presented in [Ribrant, 2006] and [DOWEC team, 2002], while severity indices are determined based on the simulation results included in the following sections.

In this appendix each sub-model is analyzed separately. Consistently, plots in two colors are utilized, where blue lines indicate simulation results of the fault-free scenario and red lines represent results with injection of faults. To utilize the same time axis at the shown simulations, the faults are injected appropriately fast, such that unrealistic phenomena do not occur even when incipient faults are considered. Notice that no fault handling exists in the nominal controller used in the simulations. Furthermore, only a selected number of simulation results are provided in this appendix to limit the extent of the analysis.

A.1 Severity and Occurrence Analysis of Rotor Sub-Model

The end-effects for the faults in the rotor sub-model are shown in Table A.1 with associated severity and occurrence indices. It should be remarked that it is difficult to assign a certain occurrence index to component faults such as a biased sensor output, dirt on blades, and pump wear, since they are always present to some extent. Similar concerns exist in connection to assigning severity numbers. For instance, a small bias on a sensor is very likely but has only a minor influence on the system, whereas a larger bias on the sensor is less likely but is more severe. The tables in this appendix however only state a single value for both the occurrence index and the severity index for a given fault, which should therefore be related to the fault sizes injected in the simulations utilized in this appendix.

Effect	Fault	O	S
Unbalanced rotation	Pitch sensor (1,2,3) - Biased output	3	6
	Pitch sensor (1,2,3) - Fixed output	2	9
	Pitch sensor (1,2,3) - Random output	1	8
	Pitch sensor (1,2,3) - No output	3	9
	Blade (1,2,3) - Damaged blade tip	3	6
Changed efficiency	Blade (1,2,3) - Dirt on blades	10	3
Changed dynamics	Pitch actuator (1,2,3) - Pump wear	4	5
	Pitch actuator (1,2,3) - Hydraulic leakage	3	8
	Pitch actuator (1,2,3) - High air content in oil	5	5
Out of control	Pitch actuator (1,2,3) - Valve blockage	3	8
	Pitch actuator (1,2,3) - Pump blockage	2	9

Table A.1: Assignment of severity (S) and occurrence (O) indices for the rotor sub-model.

According to Table A.1 the severity of one end-effect may depend heavily on the component fault which causes the failure: a particular end-effect may have multiple degrees of severity. In the case of an unbalanced rotation of the rotor, the severity depends on the mutual difference between the blades and their individual aerodynamic properties. In this case a biased output from a pitch sensor is likely to cause a small pitch offset on a blade, whereas a random output or no output of the sensor cause an irregular pitching of one particular blade and therefore introduces a much

larger unbalance in the rotor plane. From the simulations in the following subsections the specific severity indices are estimated and inserted in the table.

As most of the faults considered in connection with the rotor sub-model are related to the pitching mechanism, simulation results are only conducted for the full load region, since the pitch system is only activated in this region. Therefore, simulations used in the next subsections are conducted using a mean wind speed of 16 m/s.

Unbalanced Rotation

Blade damage and incorrect pitching of a blade cause asymmetry in the forces affecting the rotor; hence, introducing an unbalanced rotation, which is assumed to increase mechanical loads on the rotor and tower. Potentially, an unbalanced rotation can bring the system to a hazardous situation due to the increased loads on the structure.

Biased output from a pitch sensor equal to -2° is simulated to obtain the results shown in Figure A.1, where the bias is injected linearly between $t = 30$ s and $t = 40$ s. From the simulation it is noticed that power production is not affected much, while the tracking of the generator speed is affected a little due to the altered operation area of the aerodynamics. Furthermore, the asymmetric stress caused by different forces affecting the blades is assumed to increase structural loads on the rotor significantly. This is especially apparent from the tower acceleration shown in Figure A.1.

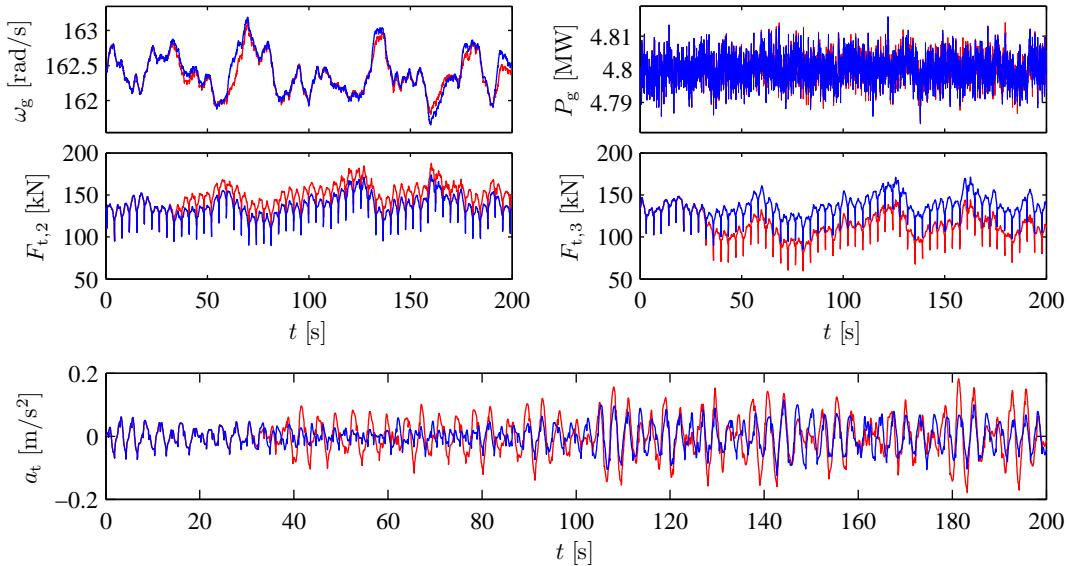


Figure A.1: Evaluation of a biased pitch angle with linear injection of the fault taking place between $t = 30$ s and $t = 40$ s, whereupon a bias of -2° is added to the measurement of the pitch angle of Blade 3.

Fixed output from a pitch sensor is the basis of the simulation results shown in Figure A.2. A fixed output propagates to the pitch controller that outputs a wrong signal to the actuator. Dependent on the values of the reference signal and the fixed output of the sensor, this makes the pitch angle pitch either towards its maximum or minimum position. The conclusion of the simulation is that structural loads are dangerously increased, which is obvious from the tower acceleration plotted in the figure. Eventually, it could cause severe damages to the wind turbine and the surroundings.

Random output from a pitch sensor results in a pitch angle that moves towards the upper or lower limits of its operational area, dependent on the reference signal to the controller in relation to the average value of the measurement. This causes a high-severity end-effect, since the mutual difference in the pitch angles rapidly increases and introduces unbalance to the rotor.

No output from a pitch sensor causes behavior of the system identical to the case of a fixed output from the sensor.

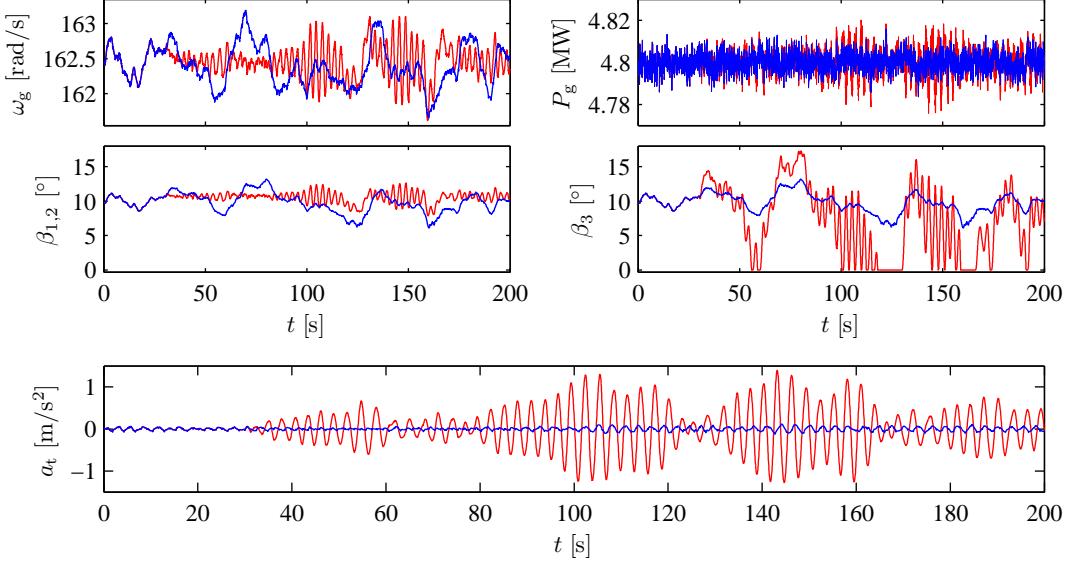


Figure A.2: Evaluation of fixed value from a pitch sensor with fault injection initiated at $t = 30$ s on the pitch sensor associated with Blade 3.

Damaged blade tip can happen e.g. due to a lightning stroke, according to [Kithil, 2008]. A damaged blade changes its aerodynamic properties, since the blade is shortened while the broken blade tip increases drag and turbulence. A simplification of these effects, utilized in this analysis, is to decrease the length of the blade in the simulation and maintain the fault-free aerodynamic properties. Therefore, the results in Figure A.3 are obtained from a simulation where the length of a blade is instantaneously reduced to 90% of its original length at $t = 30$ s.

The simulation shows that power production is only negligibly affected, but that the asymmetric load of the rotor is assumed to increase structural loads, since different forces act on the blades and since the tower acceleration is increased. The conclusion is similar to the case of a biased output from a pitch sensor, although the effect of a damaged blade tip is less likely.

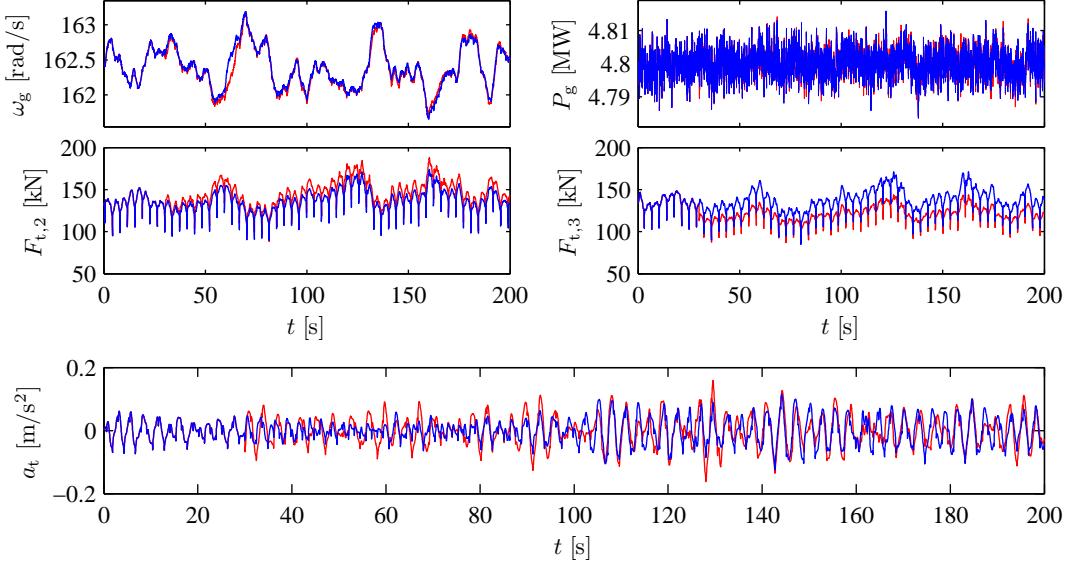


Figure A.3: Evaluation of damaged blade tip with fault injection initiated at $t = 30$ s where the length of Blade 3 is reduced by 10%.

Changed Efficiency

Research has shown that the C_p -surface of a wind turbine changes over time [Johnson *et al.*, 2006]. The consequence is that less power is produced in the partial load region due to decreased efficiency of the aerodynamics. Moreover, the optimal point of the power coefficient curve may change as well, resulting in even lower efficiency of the wind turbine. The changes are caused by debris building up on the blades and depend on the weather and the location of the wind turbine. For this reason the blades are cleaned occasionally to reduce the effect of the debris.

It seems inappropriate to specify a certain occurrence level of debris building up on the blades in Table A.1 on Page 131, since debris on the blades are always present to some extent. In return, the severity is considered minor, as the wind turbine is able to perform its objectives, but at a lower efficiency. Note however that decreased efficiency of the wind turbine may be considered a serious problem by the owner of the wind turbine.

Changed Dynamics

The dynamics of the pitch system can change due to wear of its components and may cause decreasing pressure in the hydraulic system caused by pump wear or hydraulic leakage. Furthermore, the air content of the hydraulic oil can increase when actuating the pitch system, which changes its transient response. Notice that the effects of increased air content and low pressure are evaluated by studying the effects on the phase and gain margins of the closed-loop system, while the worst-case scenarios are simulated at last.

Pump wear is an incipient fault, which is assumed to cause the pressure to decrease towards a level of 75% over a period of approximately 20 years.

Hydraulic leakage results in a decreased hydraulic pressure which, once occurred, worsens as time passes and eventually disables the possibility of using the pitch actuators.

High air content in oil causes overshoot in the transient response due to the higher elasticity of the hydraulic oil.

Step responses of the pitch system in certain conditions are illustrated in Figure A.4 with model parameters provided in Table A.2 supplied by kk-electronic a/s.

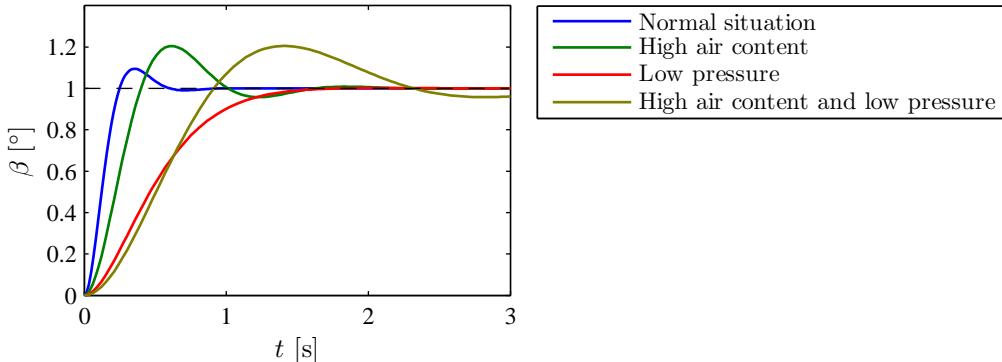


Figure A.4: Step responses of hydraulic pitch model under different conditions.

Air content \ Pressure	Normal (100%)	Low (50%)
Normal (7%)	$\omega_n = 11.11 \text{ rad/s}, \zeta = 0.6$	$\omega_n = 3.42 \text{ rad/s}, \zeta = 0.9$
High (15%)	$\omega_n = 5.73 \text{ rad/s}, \zeta = 0.45$	$\omega_n = 2.5 \text{ rad/s}, \zeta = 0.45$

Table A.2: Parameters of the pitch system under different conditions.

In Figure A.4 the significance of the changed parameters of the pitch system is evaluated for the open loop system. When closing the loop, by applying the reference controller described in Appendix C, stability of the system is no longer guaranteed due to the altered parameter values.

In order to evaluate stability, the gain and phase margins of the controllers are evaluated in the extremity of their regions at different conditions of the pitch system. These margins appear in Table A.3.

Controller	Wind Speed	Normal air content Normal pressure	High air content Normal pressure	Normal air content Low pressure
Speed Controller 1	12 m/s	Gm = 14.8 dB Pm = 76.8°	Gm = 8.50 dB Pm = 70.1°	Gm = 10.0 dB Pm = 43.5°
	15 m/s	Gm = 6.03 dB Pm = 50.0°	Gm = -0.25 dB Pm = -2.21°	Gm = 1.29 dB Pm = 5.39°
Speed Controller 2	15 m/s	Gm = 13.3 dB Pm = 71.2°	Gm = 6.98 dB Pm = 62.36°	Gm = 8.09 dB Pm = 33.7°
	25 m/s	Gm = 6.02 dB Pm = 50.0°	Gm = -0.25 dB Pm = -2.29°	Gm = 1.30 dB Pm = 5.48°

Table A.3: *Gain and phase margins of the control system evaluated in the extremity of its regions, at different conditions of the pitch system.*

According to the table the closed-loop system becomes unstable at certain wind speeds due to high air content in the hydraulic oil combined with the parameter values of the aerodynamic model at these wind speeds. At low pressure in the pitch system the controlled wind turbine appears to have very small stability margins, and it can be concluded that even minor model uncertainties or parameter deviations can bring the system in an unstable state, which results in a very high severity number.

A simulation of the wind turbine system imposed by high air content in the hydraulic oil is shown in Figure A.5. The considered air content level is equal to 15% and is expected to happen seldom, whereas it is expected that the air content often exceeds the nominal value of 7%, marginally. It is clear from the figure that the system tends to oscillate at $t = 125$ s which increases tower accelerations.

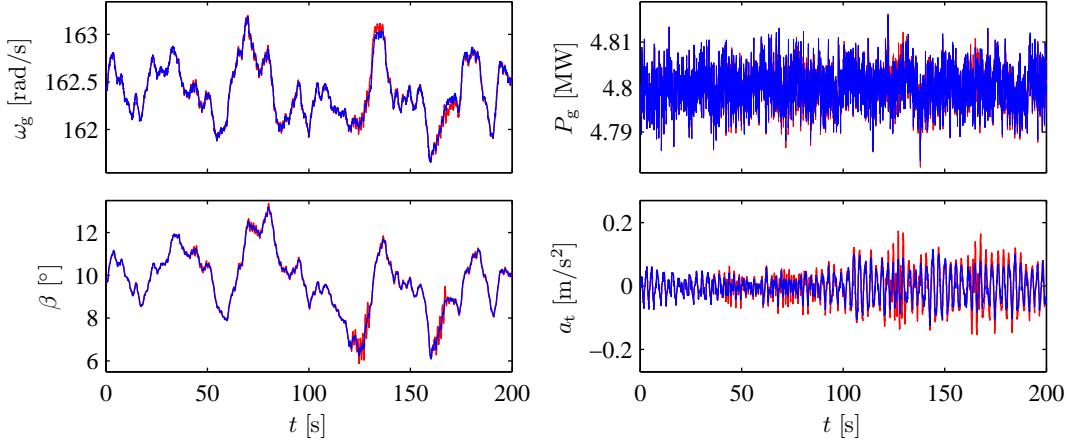


Figure A.5: *Evaluation of high air content in hydraulic oil of pitch system with fault injected linearly between $t = 30$ s and $t = 40$ s, whereupon the parameters in entry (1, 2) of Table A.2 are utilized in the model of the pitch system.*

The simulation result for a low pressure in the pitch system is shown in Figure A.6. It is assumed that the simulated pressure level of 50% can only arise from a leakage in the system and is therefore assumed to occur seldom. By comparison, it is assumed that a worn pump can only reduce the pressure level to about 75% during the lifetime of the wind turbine, according to kk-electronic a/s. The figure shows that the system invokes behaviors similar to those stated above for high air content, where in this case oscillations of the pitch angle appear around $t = 150$ s.

The severe acceleration of the tower indicates that structural loads are increased. Furthermore, the tracking ability of the generator speed reference is degraded.

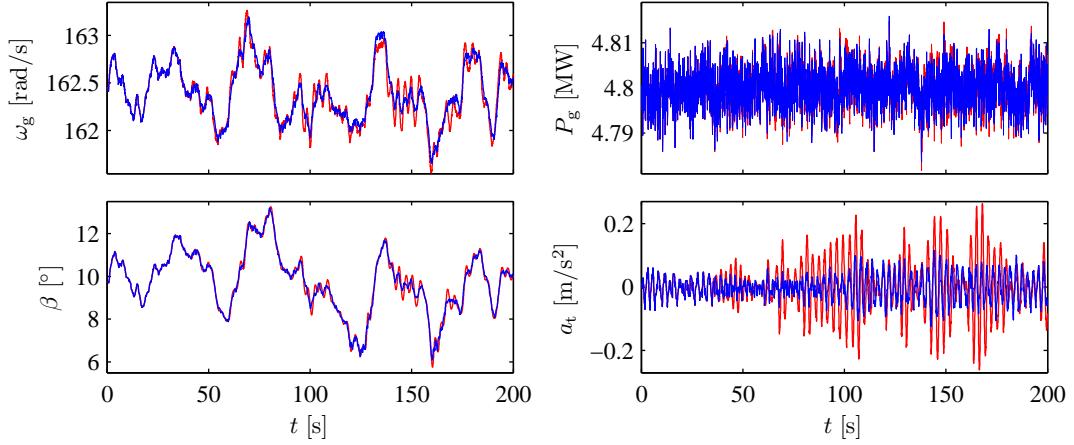


Figure A.6: Evaluation of low pressure of pitch system with fault injection initiated at $t = 30$ s where parameters in entry (2,1) of Table A.2 are utilized in the model of the pitch system.

Out of Control

In the case of a valve blockage the associated pitch actuator will be inoperable, whereas the entire pitch system will fail due to a pump blockage.

Valve blockage brings a pitch actuator in a locked position with effects shown in Figure A.7. The fault corresponds to a blockage of Valve 3 in Figure 3.14 on Page 20. As the mutual offset between the pitch angles increases, so does the mutual difference between the forces on the blades, which affects the hub and tower. Therefore, the presence of the fault is safety critical and has a very high severity. For the same reason kk-electronic a/s shut down a wind turbine, using an independent accumulator, if the mutual pitch angle offset exceeds 5° .

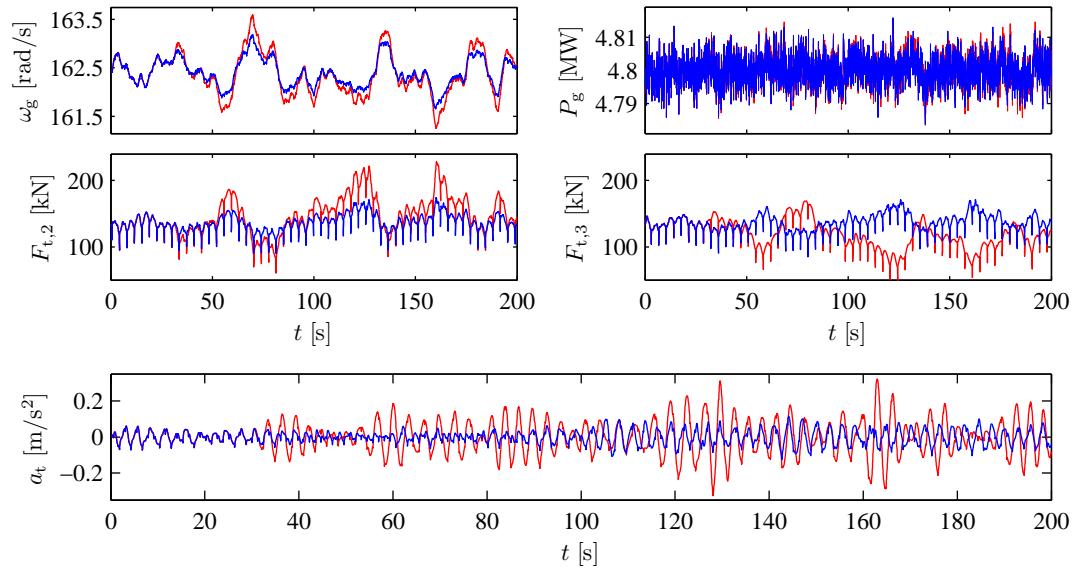


Figure A.7: Evaluation of valve blockage with fault injection initiated at $t = 30$ s where the pitch angle of Blade 3 is fixed.

Pump blockage prevents the blades from pitching, which is the situation simulated in Figure A.8. The fault corresponds to a blockage of Valve 4 in Figure 3.14 on Page 20. The simulation

reveals overspeed and increased power fluctuations in the faulty case. As the wind speed decreases the rotor is decelerated and the power output goes beneath 4.8 MW due to the limited generator torque and the inefficient aerodynamics.

The outcome of the fault is critical as the trajectory followed by the wind turbine is not decided by the controller, but left in the hands of the stochastic wind speed. This leads to a dangerous situation when the wind speed is comparatively high, while the blades are not pitched sufficiently to keep the aerodynamic torque at the nominal value.

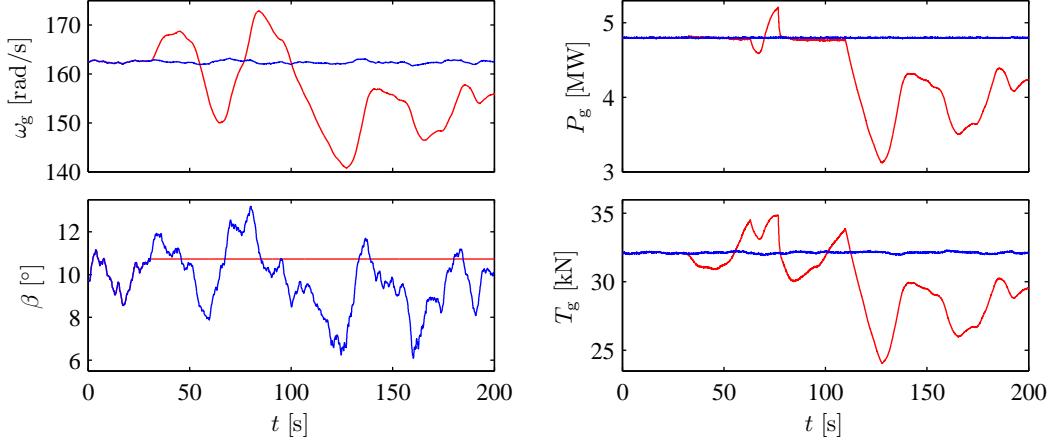


Figure A.8: *Evaluation of pump blockage with fault injection initiated at $t = 30$ s where all pitch actuators become inoperable.*

A.2 Severity and Occurrence Analysis of Drive Train Sub-Model

The end-effects for the component faults in the drive train sub-model are shown in Table A.4 with determined severity and occurrence indices.

Effect	Fault	O	S
Non-uniform rotation	High-speed shaft - Misalignment	2	4
	Low-speed shaft - Misalignment	2	4
	Gearbox - Tooth wear	2	5
Decreased efficiency	High-speed shaft - Bearing wear	5	2
	Low-speed shaft - Bearing wear	6	2
Speed offset	Generator speed sensor - Proportional error	4	4
High speed	Generator speed sensor - Fixed output	2	9
	Generator speed sensor - No output	3	9
Low speed	Generator speed sensor - Fixed output	2	8
	Generator speed sensor - No output	3	8
Random speed	Generator speed sensor - Random output	1	9

Table A.4: *Assignment of severity (S) and occurrence (O) indices for the drive train sub-model.*

The fault analysis of the drive train sub-model considers both mechanical faults in the drive shafts and the gearbox, and faults associated with the generator speed sensor that measures the angular velocity of the high-speed shaft. In this analysis, however, only faults in the generator speed sensor are simulated, as the modeling of mechanical faults in the gearbox and bearings require models of higher detail than the one provided in this thesis. Moreover, it should be pointed out that diagnostic systems already exist, which are able to detect and identify problems from

vibration sensors located around the gearbox and the bearings. However, these methods are signal-based rather than model-based. For a descriptions of how to model the dynamic behavior of a complex drive train in a wind turbine consult [Peeters, 2006]. The severity indices for the faults that are not simulated are rough guesses, since the model gives no details about it.

The simulations referred to throughout the next subsections are conducted using a mean wind speed of 10 m/s when analyzing end-effects in the partial load region while a mean wind speed of 16 m/s is utilized in the simulations for the full load region.

Non-Uniform Rotation

Misalignment in the mechanical structure may happen in the assembling process of the transmission or arise during operation, e.g. may the generator sink over time due to aging of its vibration damper [Ribrant, 2006, p. 55]. Misalignments could bring the system to run incorrectly and will generate vibrations that may result in a breakdown of the mechanical system. Gears and bearings seldom break down abruptly, but are subjects to a wear process [Ribrant, 2006, p. 56].

Decreased Efficiency

Damaged bearings often come as a direct result of improper lubrication and lack of routine maintenance and cause increased friction, mechanical oscillations, and decreased efficiency of the drive train. [Gill, 2008].

Speed Offset

Smudge on the disc surface of the speed encoder may introduce an error proportional to the rotational speed, due to some tracks on the disc being covered by dust and dirt. On the other hand, high-frequency mechanical vibrations may result in some tracks being counted several times during a single rotation. This creates an offset in the measured speed from the encoder, which is assumed to be proportional to the absolute speed, as mechanical vibrations increase with the rotational speed. However, the latter is typically managed internally in a fault correction scheme in the sensor.

A proportional error of the generator speed sensor has different effects in the partial load and full load regions, and further depends on whether the error causes a too low or too high reading. The simulation results related to the partial load region are found in Figure A.9, while simulation results for the full load region follow in Figure A.10. In both cases fault injection happens at $t = 30$ s where the gain of the speed sensor is linearly reduced from 1 to 0.96 during a minute to reflect that some tracks are not being counted.

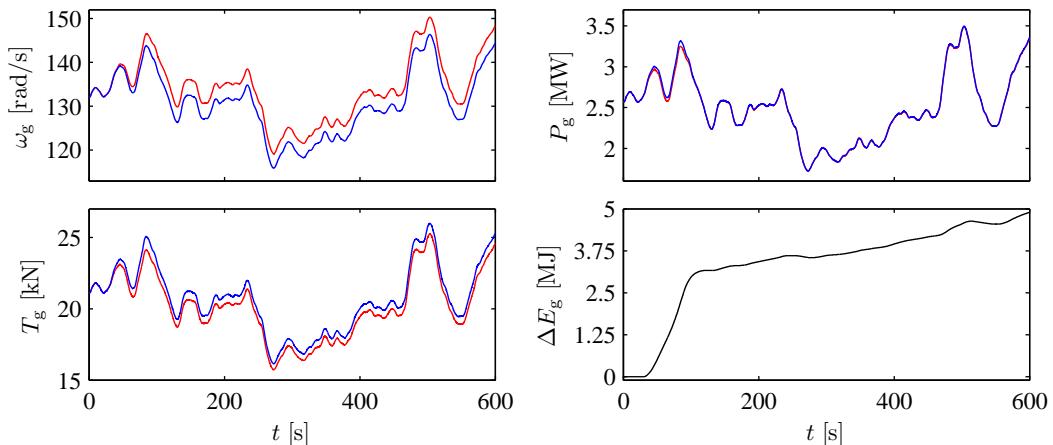


Figure A.9: Evaluation of proportional error of the generator speed sensor in the partial load region with fault injected linearly between $t = 30$ s and $t = 90$ s, whereupon the gain of the speed sensor is reduced from 1 to 0.96. Note that $\Delta E_g = \int_0^t (P_{g,nominal}(\tau) - P_{g,faulty}(\tau)) d\tau$.

In the partial load region the fault propagates to the set-point value of the generator torque controller, causing a deviation from the nominal operating trajectory. This involves no safety risks but leads to lower efficiency of the wind turbine, which is evaluated in the lower right subplot of Figure A.9, where the loss in energy production is illustrated.

The considered example of a faulty speed measurement in the full load region results in overspeed of the generator shaft. This shrinks the safety speed margin and makes the rotational speed approach the critical speed limit, which results in a shutdown of the wind turbine when reached.

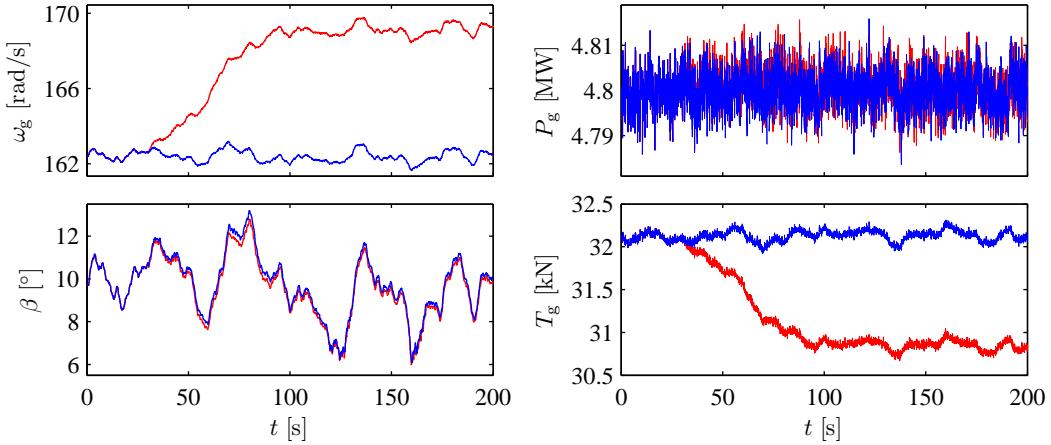


Figure A.10: *Evaluation of proportional error of the generator speed sensor in the full load region with fault injected linearly between $t = 30$ s and $t = 90$ s, whereupon the gain of the speed sensor is reduced from 1 to 0.96.*

High and Low Speed

Too high or too low rotational speed comes as a result of a fixed output or no output from the generator speed sensor.

Fixed output from the generator speed sensor is evaluated for the partial load region in Figure A.11 while results for the full load region follow in Figure A.12.

In the partial load region the fixed measurement of the generator speed implies a constant generator torque, as seen in Figure A.11. If the wind speed increases, once the fault is injected, the speed of the rotor increases without any influence of control. However, if the wind speed decreases, the rotor will decelerate and come to a stop if the wind speed becomes low enough. Hence, the wind turbine is inoperable.

Under full load operation a fixed output propagates to the speed controller and causes improper pitching of the blades. Dependent on the fixed value from the sensor, in comparison to the constant set-point value to the generator speed, the pitch angle either converges towards the minimum or maximum pitch angle, or it stays at a constant value. This situation is critical to the safety of the wind turbine due to the possibility of overspeed.

No output from the generator speed sensor implies similar effect as mentioned above, although a notification of the fault is available to the control system, since the sensor does not provide an output.

Random Speed

If the generator speed sensor outputs a random output, the system is brought into a hazardous situation since the wind turbine becomes out of control; high loads are introduced by a fluctuating pitch angle in the full load region or fluctuating generator torque in the partial load region.

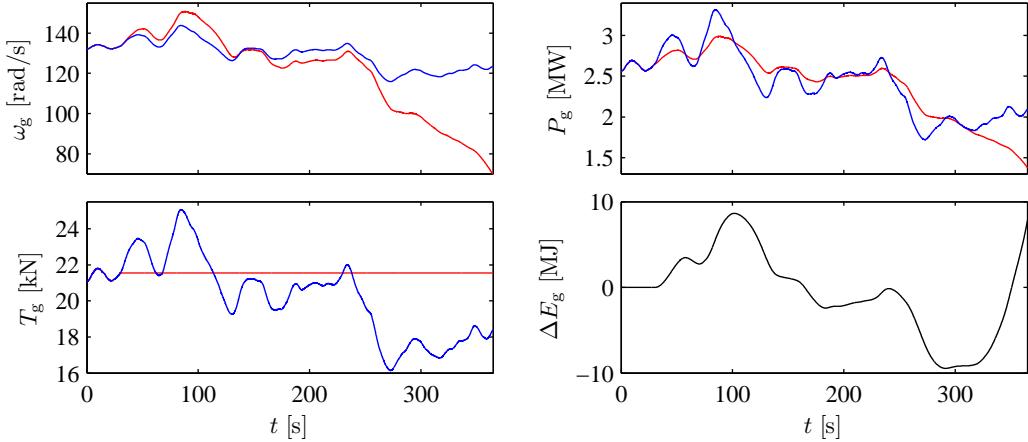


Figure A.11: Evaluation of fixed value from generator speed sensor in the partial load region with fault injection initiated at $t = 30$ s where the output of the speed sensor is fixed. Note that $\Delta E_g = \int_0^t (P_{g,\text{nominal}}(\tau) - P_{g,\text{faulty}}(\tau)) d\tau$.

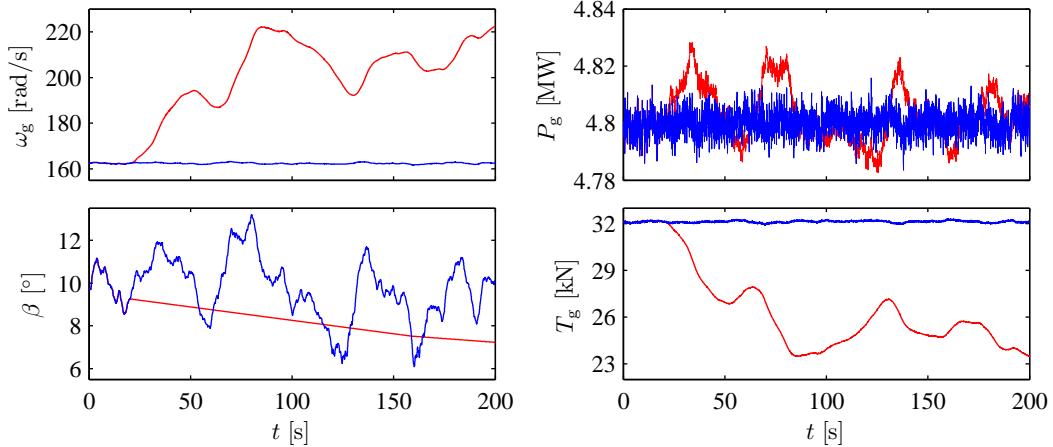


Figure A.12: Evaluation of fixed value from generator speed sensor in the full load region with fault injection initiated at $t = 30$ s where the output of the speed sensor is fixed at a value equal to the reference value to the speed controller.

A.3 Severity and Occurrence Analysis of Power System Sub-Model

The end-effects for the faults in the power system sub-model are shown in Table A.5 with the respective severity and occurrence indices.

As discussed in Section 3.10 the converter torque and output power are not measured directly, but are estimated based on measurements of flux, current, and voltage inside each converter. Therefore, it is assumed that these estimates are coherent, such that a fault in one of the estimates results in a fault in the other estimate as well. For example, a biased generator torque measurement of a converter is assumed to imply a biased measurement of the output power for this particular converter. Notice that there are four converters as discussed in Section 3.6.

Power Offset

In the partial load region, a biased measurement of the converter torque has similar consequences as an offset in the generator speed evaluated in Figure A.9. The effect is that the wind turbine does not operate along the nominal operating trajectory; hence, the efficiency of the wind turbine is decreased.

Effect	Fault	O	S
Power offset	Converter torque sensor (1,2,3,4) - Biased output Power sensor (1,2,3,4) - Biased output	3	4
High power	Converter torque sensor (1,2,3,4) - Fixed output Power sensor (1,2,3,4) - Fixed output	2	7
	Converter torque sensor (1,2,3,4) - No output Power sensor (1,2,3,4) - No output	3	7
Low power	Converter (1,2,3,4) - Converter failure	2	7
	Converter torque sensor (1,2,3,4) - Fixed output Power sensor (1,2,3,4) - Fixed output	2	7
Random power	Converter torque sensor (1,2,3,4) - No output Power sensor (1,2,3,4) - No output	3	7
	Converter torque sensor (1,2,3,4) - Random output Power sensor (1,2,3,4) - Random output	1	7
Changed dynamics	Converter (1,2,3,4) - Converter failure	2	1

Table A.5: Assignment of severity (*S*) and occurrence (*O*) indices for the power system sub-model.

In the full load region, a negative bias on the measured output power induces a biased output power, which is larger than the rated output power and will put higher loads on the wind turbine components.

High and Low Power

In contrast to the power offset described above, a too high or too low output power is different, since it results in a completely different output power, and not just an offset compared to the fault-free scenario. If the failure is caused by a fixed output or no output from a converter torque sensor and power sensor, the result will be a fluctuating power. Also, failure of a single converter can limit the possible output power due to the limited converter torque while operating without one of the four converters.

Fixed outputs from a converter torque sensor and power sensor is evaluated in Figure A.13 for the partial load region, while results corresponding to operation in the full load region are shown in Figure A.14. The conclusion is that the power production is compromised by the oscillatory torque from the converter with the faulty sensor. Furthermore, it is assumed that fatigue loads in the drive train are increased significantly, as the changing generator torque creates larger torsion in the gearbox.

No outputs from a converter torque sensor and power sensor implies a similar effect as mentioned above, although a notification of the fault is available to the control system, since no measurements are outputted.

Failure of a converter makes its output drop to zero. By simulation, the fault is injected progressively during a five seconds interval starting at $t = 30$ s obtaining the results shown in Figure A.15. The conclusion is that in the full load region the rated output of the wind turbine cannot be reached, even though the three remaining converters deliver their maximum torque.

At the lower wind speeds in the partial load region, where the reduced maximum converter torque is not an issue, the situation in the faulty case does not differ significantly from the fault-free case. Only in the case where the three converters reach their slew rate, since they are heavily loaded, the output is different from the situation where every converter functions satisfactory.

Random Power

Random output from the sensors of a converter lead to a fluctuating generator torque which increases torsion in the drive train.

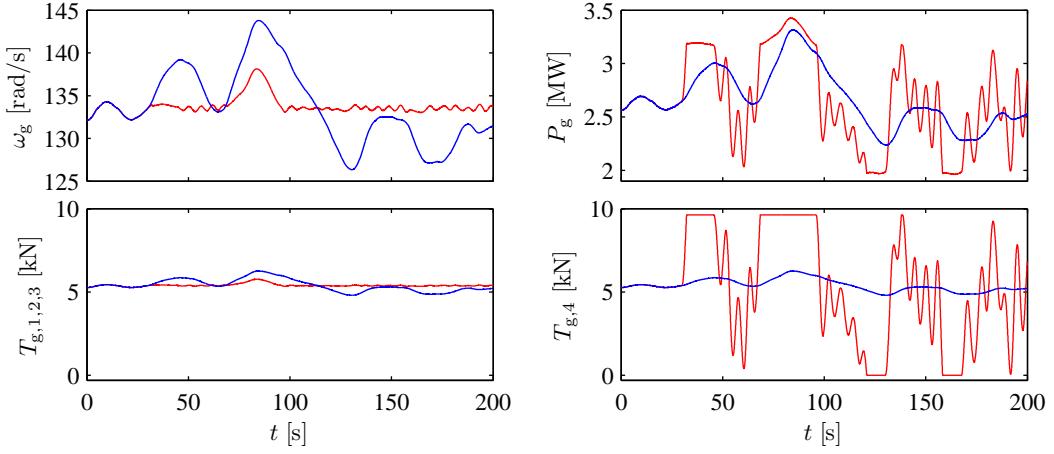


Figure A.13: Evaluation of fixed outputs from the sensors of a converter in the partial load region with fault injection initiated at $t = 30$ s where the outputs of the converter torque sensor and power sensor are fixed for Converter 4.

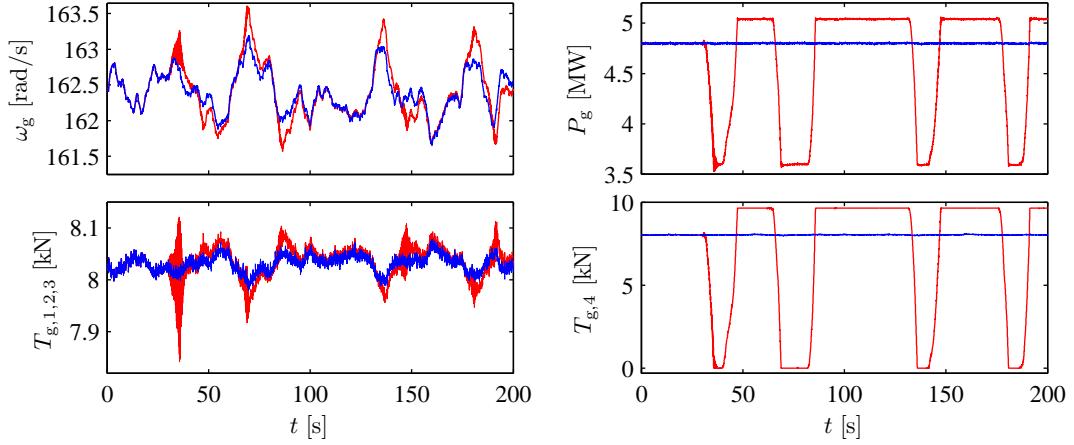


Figure A.14: Evaluation of fixed outputs from the sensors of a converter in the full load region with fault injection initiated at $t = 30$ s where the outputs of the converter torque sensor and power sensor are fixed for Converter 4.

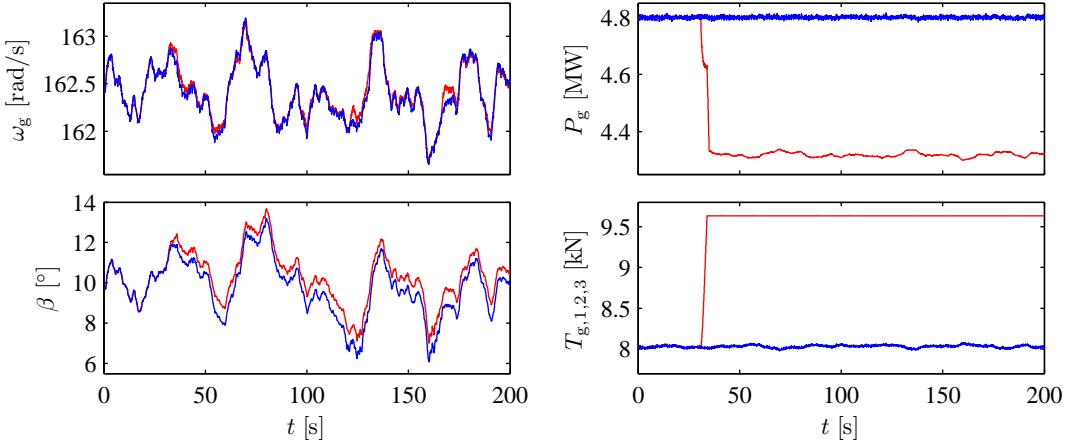


Figure A.15: Evaluation of failed converter with fault injected smoothly between $t = 30$ s and $t = 35$ s, whereupon the output of the failed converter drops to zero.

Changed Dynamics

Changed dynamics refers to the fact that when a single converter fails, the dynamics of the converters as a single unit change. This results in slower dynamics due to a reduced slew rate of the joint converter system. However, when utilizing the considered parameters of the converters, the problem seems to be rather theoretical, since the variation in the generator torque comes no way near the slew rate.

B Linearization

Linearization of the plant is performed to make the use of linear design methods possible in this thesis. Even the extended Kalman filters utilized in this project, which exploit the non-linear model directly, also use linearized models to propagate the covariance matrices. The linearization procedure is performed using the steps shown below [Andersen and Pedersen, 2007, p. 60].

1. Determine the operating point equation by solving the non-linear equation in a stationary point.
2. Replace the variables with operating point values plus small signal values, i.e. $x(t) = \bar{x} + x^\Delta(t)$, and approximate the non-linear expressions with first order Taylor approximations.
3. Subtract the operating point equation from the Taylor approximation. The result is a linear equation in small signal values.

In order to set up a linear model of the wind turbine system, two equations for the aerodynamics have to be linearized. Furthermore, the output power equation is linearized in order to be included in an output equation of the system. Linearizing the expressions is accomplished in the next three sections. Notice that this appendix is inspired by [Esbensen *et al.*, 2008, pp. 121-123]

B.1 Aerodynamic Torque Applied to the Rotor

The purpose of this section is to linearize the equation of the aerodynamic torque that is applied to the rotor. For convenience the equation describing the torque acting on the rotor, Eq. (3.9) on Page 15, is repeated below.

$$T_a(t) = \frac{1}{2\omega_r(t)} \rho A v_r^3(t) C_p(\lambda(t), \beta(t)) \quad [\text{Nm}] \quad (\text{B.1})$$

1. To set up a linear expression of the torque, firstly, the operating point equation is derived in a stationary point. A bar indicates the operating point value of the variable:

$$\bar{T}_a = \frac{1}{2\bar{\omega}_r} \rho A \bar{v}_r^3 C_p(\bar{\lambda}, \bar{\beta}) \quad [\text{Nm}] \quad (\text{B.2})$$

2. In order to form linearized expressions, a Taylor approximation of each non-linear part of Eq. (B.1) is derived and each linear part is replaced with an operating point value plus a small signal value. The equations are set up below.

$$\begin{aligned} T_a(t) &= \bar{T}_a + T_a^\Delta(t) \\ &= \bar{T}_a + \frac{\partial T_a}{\partial \omega_r} \omega_r^\Delta(t) + \frac{\partial T_a}{\partial v_r} v_r^\Delta(t) + \frac{\partial T_a}{\partial \beta} \beta^\Delta(t) \\ \frac{\partial T_a(t)}{\partial \omega_r} &\approx \frac{1}{2} \rho A \bar{v}_r^3 \left(-\frac{1}{\bar{\omega}_r^2} C_p(\bar{\lambda}, \bar{\beta}) + \frac{1}{\bar{\omega}_r} \frac{\partial C_p(\lambda(t), \beta(t))}{\partial \omega_r} \right) \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \\ \frac{\partial T_a(t)}{\partial v_r} &\approx \frac{1}{2\bar{\omega}_r} \rho A \left(3\bar{v}_r^2 C_p(\bar{\lambda}, \bar{\beta}) + \bar{v}_r^3 \frac{\partial C_p(\lambda(t), \beta(t))}{\partial v_r} \right) \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \\ \frac{\partial T_a(t)}{\partial \beta} &\approx \frac{1}{2\bar{\omega}_r} \rho A \bar{v}_r^3 \frac{\partial C_p(\lambda(t), \beta(t))}{\partial \beta} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \end{aligned}$$

By combining the previous equations, a new equation for $T_a(t)$ can be set up as shown in Eq. (B.3).

$$\begin{aligned}
 T_a(t) = & \frac{1}{2} \rho A \bar{v}_r^3 \left(-\frac{1}{\bar{\omega}_r^2} C_p(\bar{\lambda}, \bar{\beta}) + \frac{1}{\bar{\omega}_r} \frac{\partial C_p(\lambda(t), \beta(t))}{\partial \omega_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \right) \omega_r^\Delta(t) \\
 & + \frac{1}{2\bar{\omega}_r} \rho A \left(3\bar{v}_r^2 C_p(\bar{\lambda}, \bar{\beta}) + \bar{v}_r^3 \frac{\partial C_p(\lambda(t), \beta(t))}{\partial v_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \right) v_r^\Delta(t) \\
 & + \frac{1}{2\bar{\omega}_r} \rho A \bar{v}_r^3 \frac{\partial C_p(\lambda(t), \beta(t))}{\partial \beta} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \beta^\Delta(t) \\
 & + \frac{1}{2\bar{\omega}_r} \rho A \bar{v}_r^3 C_p(\bar{\lambda}, \bar{\beta}) \quad [\text{Nm}] \tag{B.3}
 \end{aligned}$$

3. To obtain a linear system equation the operating point equation, Eq. (B.2), is subtracted from Eq. (B.3). The result shown in Eq. (B.4) is a first order differential equation, which is used in the state space model.

$$\begin{aligned}
 T_a^\Delta(t) = & \frac{1}{2} \rho A \bar{v}_r^3 \left(-\frac{1}{\bar{\omega}_r^2} C_p(\bar{\lambda}, \bar{\beta}) + \frac{1}{\bar{\omega}_r} \frac{\partial C_p(\lambda(t), \beta(t))}{\partial \omega_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \right) \omega_r^\Delta(t) \\
 & + \frac{1}{2\bar{\omega}_r} \rho A \left(3\bar{v}_r^2 C_p(\bar{\lambda}, \bar{\beta}) + \bar{v}_r^3 \frac{\partial C_p(\lambda(t), \beta(t))}{\partial v_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \right) v_r^\Delta(t) \\
 & + \frac{1}{2\bar{\omega}_r} \rho A \bar{v}_r^3 \frac{\partial C_p(\lambda(t), \beta(t))}{\partial \beta} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \beta^\Delta(t) \quad [\text{Nm}] \tag{B.4}
 \end{aligned}$$

B.2 Thrust Acting on the Rotor

The purpose of this section is to linearize the equation describing the thrust acting on the rotor. For convenience the equation describing the thrust, Eq. (3.10) on Page 15, is repeated below.

$$F_t(t) = \frac{1}{2} \rho A v_r^2(t) C_t(\lambda(t), \beta(t)) \quad [\text{N}] \tag{B.5}$$

1. To set up a linear expression of the thrust, firstly, the operating point equation is derived in a stationary point:

$$\bar{F}_t = \frac{1}{2} \rho A \bar{v}_r^2 C_t(\bar{\lambda}, \bar{\beta}) \quad [\text{N}] \tag{B.6}$$

2. In order to form linearized expressions, a Taylor approximation of each non-linear part of Eq. (B.5) is derived and each linear part is replaced with an operating point value plus a small

signal value. The equations are set up below.

$$\begin{aligned}
 F_t(t) &= \bar{F}_t + F_t^\Delta(t) \\
 &= \bar{F}_t + \frac{\partial F_t}{\partial \omega_r} \omega_r^\Delta(t) + \frac{\partial F_t}{\partial v_r} v_r^\Delta(t) + \frac{\partial F_t}{\partial \beta} \beta^\Delta(t) \\
 \frac{\partial F_t(t)}{\partial \omega_r} &\approx \frac{1}{2} \rho A \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial \omega_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \\
 \frac{\partial F_t(t)}{\partial v_r} &\approx \frac{1}{2} \rho A \left(2\bar{v}_r C_t(\bar{\lambda}, \bar{\beta}) + \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial v_r} \right) \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \\
 \frac{\partial F_t(t)}{\partial \beta} &\approx \frac{1}{2} \rho A \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial \beta} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}}
 \end{aligned}$$

By combining the previous four equations, a new equation for $F_t(t)$ can be set up as shown in Eq. (B.7).

$$\begin{aligned}
 F_t(t) &= \frac{1}{2} \rho A \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial \omega_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \omega_r^\Delta(t) \\
 &+ \frac{1}{2} \rho A \left(2\bar{v}_r C_t(\bar{\lambda}, \bar{\beta}) + \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial v_r} \right) \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} v_r^\Delta(t) \\
 &+ \frac{1}{2} \rho A \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial \beta} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \beta^\Delta(t) \\
 &+ \frac{1}{2} \rho A \bar{v}_r^2 C_t(\bar{\lambda}, \bar{\beta}) \quad [N] \tag{B.7}
 \end{aligned}$$

3. To obtain a linear system equation the operating point equation, Eq. (B.6), is subtracted from Eq. (B.7). The result shown in Eq. (B.8) is a first order differential equation, which is used in the state space model.

$$\begin{aligned}
 F_t^\Delta(t) &= \frac{1}{2} \rho A \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial \omega_r} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \omega_r^\Delta(t) \\
 &+ \frac{1}{2} \rho A \left(2\bar{v}_r C_t(\bar{\lambda}, \bar{\beta}) + \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial v_r} \right) \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} v_r^\Delta(t) \\
 &+ \frac{1}{2} \rho A \bar{v}_r^2 \frac{\partial C_t(\lambda(t), \beta(t))}{\partial \beta} \Bigg|_{\substack{\beta(t) = \bar{\beta} \\ \omega_r(t) = \bar{\omega}_r \\ v_r(t) = \bar{v}_r}} \beta^\Delta(t) \quad [N] \tag{B.8}
 \end{aligned}$$

B.3 Output Power

The purpose of this section is to linearize the power produced by the generator. For convenience the equation describing the output power, Eq. (3.28) on Page 20, is repeated below.

$$P_g(t) = \eta_g \omega_g(t) T_g(t) \quad [\text{W}] \quad (\text{B.9})$$

1. To set up a linear expression, firstly, the operating point equation is derived in a stationary point:

$$\bar{P}_g = \eta_g \bar{\omega}_g \bar{T}_g \quad [\text{W}] \quad (\text{B.10})$$

2. In order to form linearized expressions, a Taylor approximation of each non-linear part of Eq. (B.9) is derived and each linear part is replaced with an operating point value plus a small signal value. The equations are set up below.

$$\begin{aligned} P_g(t) &= \bar{P}_g + P_g^\Delta(t) \\ \eta_g \omega_g(t) T_g(t) &\approx \eta_g \bar{\omega}_g \bar{T}_g + \eta_g \bar{\omega}_g T_g^\Delta(t) + \eta_g \bar{T}_g \omega_g^\Delta(t) \end{aligned}$$

By combining the previous two equations, a new equation for $P_g(t)$ can be set up as shown in Eq. (B.11).

$$\bar{P}_g + P_g^\Delta(t) = \eta_g \bar{\omega}_g \bar{T}_g + \eta_g \bar{\omega}_g T_g^\Delta(t) + \eta_g \bar{T}_g \omega_g^\Delta(t) \quad [\text{W}] \quad (\text{B.11})$$

3. To obtain a linear system equation the operating point equation, Eq. (B.10), is subtracted from Eq. (B.11). The result shown in Eq. (B.12) is a first order differential equation, which is used in the state space model.

$$P_g^\Delta(t) = \eta_g \bar{\omega}_g T_g^\Delta(t) + \eta_g \bar{T}_g \omega_g^\Delta(t) \quad [\text{W}] \quad (\text{B.12})$$

Reference Controller

In this appendix the design of a reference controller for the wind turbine is described. The controller is supposed to approximate the configuration of an existing control system, and is designed based on classical principles. Note that the structure of the reference controller is specified in cooperation with kk-electronic a/s and that this appendix is based on [Esbensen *et al.*, 2008, pp. 45-59].

As stated in Section 2.2 the nominal operating trajectory of the wind turbine is created to satisfy different demands below and above a certain wind speed. Since the classical control approach deals only with SISO transfer functions, and because several references exist, the control task is split into the design of multiple separate controllers.

For an overview of the structure of the classical controller refer to Figure 2.4 on Page 8. The design of the classical controller is divided into four main design steps, corresponding to the names of the sections in this appendix. These are listed below to provide an overview.

- **Controller Operating in Partial Load Operation:** Describe the design and verification of the generator torque controller. This controller operates in the partial load region and should maximize the energy production while minimizing mechanical stress and actuator usage.
- **Controller Operating in Full Load Operation:** Describe the design and verification of the speed controller and power controller. These controllers operate in the full load region and should track the rated generator speed and limiting the output power.
- **Drive Train Stress Damper:** Describe the design and verification of the drive train stress damper. The purpose of the drive train stress damper is to dampen drive train oscillations.
- **Bumpless Transfer:** Describe the design and verification of the mechanism which eliminates bumps on the control signals, when switching between the controllers in the partial load and full load regions.

Notice that the transfer functions derived throughout this appendix have been discretized to allow implementation of the controllers and filters.

C.1 Controller Operating in Partial Load Operation

At low wind speeds, i.e. in partial load operation, variable-speed control is implemented to track the optimum point on the C_p -surface for maximizing power output. The speed of the generator is controlled by regulating the torque on the generator through the generator torque controller. The purpose of this section is to go through the design of the generator torque controller. Finally, it is evaluated by simulation.

In partial load operation it is chosen to operate the wind turbine at $\beta = 0^\circ$ since the maximum power coefficient is obtained at this pitch angle. This means that the highest efficiency is achieved for:

$$\lambda_{\text{opt}} = \frac{\omega_{r,\text{opt}}(t)R}{v_r(t)} \quad [:] \quad (\text{C.1})$$

where:

- R is the rotor radius [m]
 $v_r(t)$ is the rotor effective wind speed [m/s]
 λ_{opt} is the tip-speed ratio maximizing the C_p -value for $\beta = 0^\circ$ [:]
 $\omega_{r,\text{opt}}(t)$ is the optimum rotor speed [rad/s]

In order to obtain the optimal tip-speed ratio a method is used, which suggests applying a certain generator torque as a function of the generator speed. This is described in [Johnson *et al.*, 2006, p. 74] and is referred to as the standard control law for operating variable-speed turbines. The benefit of this approach is that only the measurement of the rotor speed or generator speed is

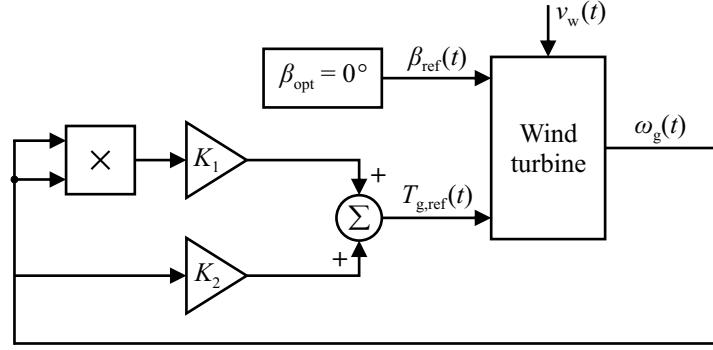


Figure C.1: Generator torque controller for operation in the partial load region, i.e. below the rated wind speed.

required. When utilizing this approach, the controller structure for partial load operation becomes as illustrated in Figure C.1, which will become apparent from the following text.

The principle of the standard control law is to isolate the wind speed in the definition of the tip-speed ratio, Eq. (3.7) on Page 14, and insert it into the expression for the aerodynamic torque, Eq. (3.9) on Page 15. Hence, an equation can be obtained expressing the required generator torque based on the maximum power coefficient and the optimal tip-speed ratio. The equation is now going to be derived. First, the wind speed is isolated:

$$v_r(t) = \frac{\omega_r(t)R}{\lambda(t)} \quad [\text{m/s}] \quad (\text{C.2})$$

Secondly, this expression is inserted into Eq. (3.9) describing the aerodynamic torque:

$$\begin{aligned} T_a(t) &= \frac{1}{2\omega_r(t)} \rho A v_r^3(t) C_p(\lambda(t), \beta(t)) \\ T_a(t) &= \frac{1}{2} \rho A \frac{R^3}{\lambda^3(t)} C_p(\lambda(t), \beta(t)) \omega_r^2(t) \quad [\text{Nm}] \end{aligned} \quad (\text{C.3})$$

Since the wind turbine includes a transmission system, the gear ratio and friction components of the drive train have to be considered when determining the generator torque corresponding to a certain aerodynamic torque. This is however not included in [Johnson *et al.*, 2006, p. 74], and in order to describe the generator torque only as function of the generator speed, the system has to be assumed in steady-state, where $\dot{\omega}_r(t) = 0 \text{ rad/s}^2$, $\dot{\omega}_g(t) = 0 \text{ rad/s}^2$, and $\omega_g(t) = N_g \omega_r(t)$. By doing so, Eq. (Eq. (3.19)) and Eq. (3.20) on Page 17 can be rewritten into:

$$\begin{aligned} 0 &= T_a(t) - K_{dt}\theta_\Delta(t) - B_r\omega_r(t) \quad [\text{Nm}] \\ 0 &= \frac{K_{dt}}{N_g}\theta_\Delta(t) - B_g\omega_g(t) - T_g(t) \quad [\text{Nm}] \end{aligned} \quad (\text{C.4})$$

These equations can be combined into:

$$T_a(t) = N_g T_g(t) + \left(\frac{B_r}{N_g} + N_g B_g \right) \omega_g(t) \quad [\text{Nm}] \quad (\text{C.5})$$

Inserting Eq. (C.5) into Eq. (C.3) in an optimal point for the power coefficient leads to the control law dictated by Eq. (C.6). Notice that the equation is based on a steady-state consideration so that actuator dynamics can also be disregarded by setting $T_{g,\text{ref}}(t) = T_g(t)$.

$$T_{g,\text{ref}}(t) = \underbrace{\frac{1}{2} \rho A \frac{R^3}{N_g^3 \lambda_{\text{opt}}^3} C_{p,\text{max}} \omega_g^2(t)}_{K_1} - \underbrace{\left(\frac{B_r}{N_g^2} + B_g \right) \omega_g(t)}_{K_2} \quad [\text{Nm}] \quad (\text{C.6})$$

where:

K_1 is the gain of the standard control law [Nm/(rad/s)²]

K_2 is a gain compensating for frictions in the drive train [Nm/(rad/s)]

Evaluation of Controller Operating at Partial Load

Figure C.2 shows how the generator torque controller performs. It should be noted from the figure that in the first part of the simulation the output power becomes larger than the theoretically $P_{g,max}(t)$, because kinetic energy from the rotor shaft is converted into electrical energy produced by the generator. Similar, $P_{g,max}(t)$ is above the produced power towards the end of the time series, since the inertia of the rotor has to be accelerated before $P_{g,max}(t)$ can be matched. This phenomenon is caused by ignoring dynamics in the calculation of the theoretical maximum power output.

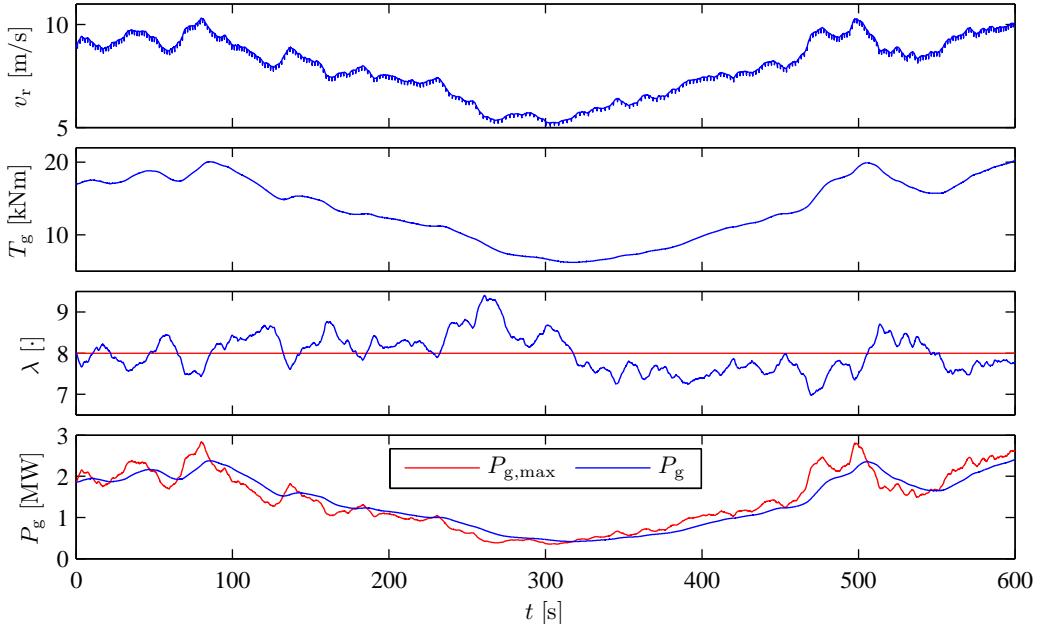


Figure C.2: Simulation of the generator torque controller operating below rated wind speed. The lower subplot compares the power produced by the generator to the theoretical maximum power output of the wind turbine given the instant wind speed.

The obvious disadvantage of the standard control law is that the optimum point of the C_p -surface must be known precisely. Even if it is assumed that this information is correct, whenever the turbine is installed, power optimization still degrades over time as a result of a changing power coefficient. According to kk-electronic a/s, the maximum power coefficient of the wind turbine blades reduces in amplitude while the relationship between tip-speed ratio, pitch angle, and C_p -value also changes over time, e.g. due to debris building up on the blades.

In the next section a description of the controller design for the operation at full load is provided.

C.2 Controller Operating in Full Load Operation

For the high wind speeds, i.e. in full load operation, the desired operation of the wind turbine is to keep the rotor speed and the generated power at constant values, as described in Section 2.2. The main idea is to use the pitch system to control the efficiency of the aerodynamics while applying the rated generator torque. However, in order to improve tracking of the power reference and cancel steady-state errors on the output power, a power controller is also introduced. Therefore, the purpose of this section is to go through the design of the speed controller and the power controller.

The structure of the controllers operating above the rated wind speed is shown in Figure C.3.

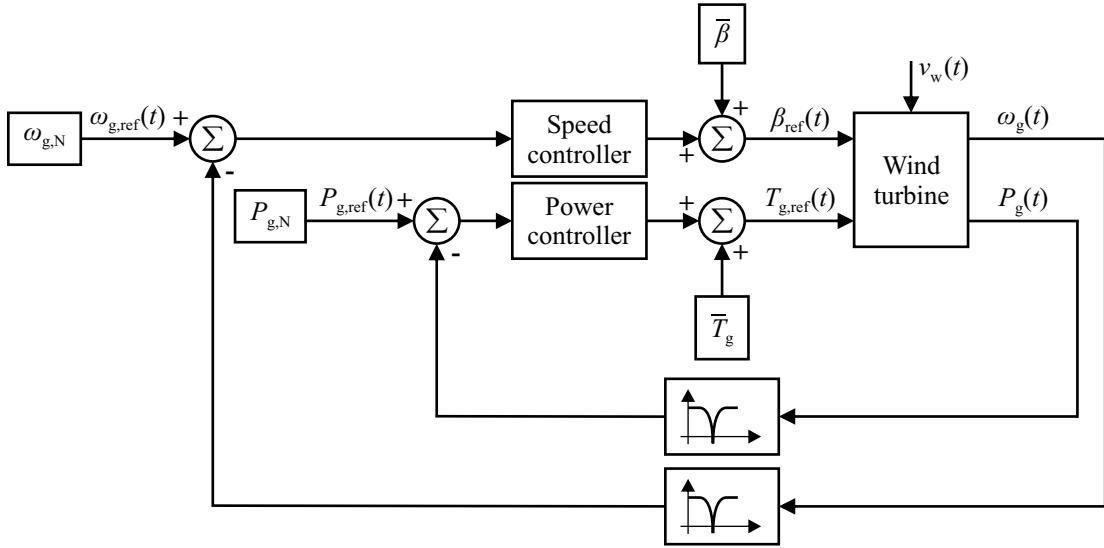


Figure C.3: Speed controller and power controller for operation in the full load region, i.e. above the rated wind speed.

The wind speed is considered the disturbance input to the system. However, higher frequency components such as the resonant frequency of the drive train are also apparent on the measured generator speed. Therefore, the measured generator speed is band-stop filtered before it is fed to the controller, to remove the drive train eigenfrequency from the measurement. This solution is also found in other wind turbine control schemes to mitigate the effects of drive train oscillations. The applied notch filter appears from Eq. (C.7) [East and Lantz, 2005, p. 3]. Notice that for implementation the filter is discretized using zero-order hold.

$$H_n(s) = \frac{s^2 + 2\frac{d}{c}\omega_0 s + \omega_0^2}{s^2 + 2\frac{1}{c}\omega_0 s + \omega_0^2} \quad [\cdot] \quad (\text{C.7})$$

where:

c and d are filter coefficients; $c = 0.1$ and $d = 5$ $[\cdot]$

$H_n(s)$ is the notch filter $[\cdot]$

ω_0 is the notch frequency of the filter, which must be close to the resonant frequency to be damped $[\text{rad/s}]$

In the following subsection the design of the speed controller is described and evaluated. Afterwards, in the succeeding subsection, the power controller is presented.

Speed Controller

The speed controller is implemented as a PI controller that is able to track the speed reference and cancel possible steady-state errors on the generator speed. The speed controller is on the form:

$$D_s(s) = K_{ps} \left(1 + \frac{1}{T_{is} \cdot s} \right) \quad [^\circ / (\text{rad/s})] \quad (\text{C.8})$$

where:

$D_s(s)$ is the PI controller to ensure that the generator speed follows the speed reference $[\circ / (\text{rad/s})]$

K_{ps} is the proportional gain of $D_s(s)$ $[\circ / (\text{rad/s})]$

T_{is} is the reset rate of $D_s(s)$ $[\text{s}]$

Figure C.4 shows how the partial derivative of $T_a(t)$ with respect to β varies along the nominal operating trajectory. As a consequence, a gain scheduling scheme is applied to the speed controller, dividing the full load region into two control regions.

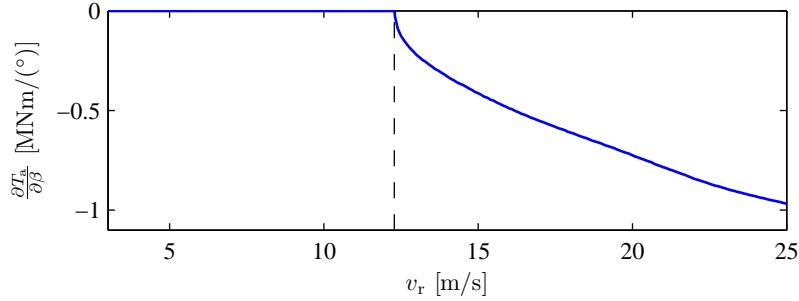


Figure C.4: Partial derivative of $T_a(t)$ with respect to β along the nominal operating trajectory. The vertical dashed line represents the rated wind speed.

According to Figure C.4 pitching the blades will have a larger influence on the aerodynamic torque at higher wind speeds. For this reason the gain of the speed controller should be large near the rated wind speed but smaller at higher wind speeds, which is also shown in [Hammerum, 2006, p. 30].

The optimal gain of the speed controller associated with a certain wind speed can make the system become unstable at higher wind speeds due to the increasing gain of the system. Therefore, the speed controller is configured with one set of parameters in the region corresponding to stationary wind speeds in the interval 12–15 m/s, while a smaller gain is utilized for the region covering wind speeds of 15–25 m/s. Although the system has different gains in these two regions, it is possible to design the controllers so that similar transient responses of the closed-loop system are obtained.

The design procedure of finding appropriate parameters for the speed controller is initiated by determining the reset rate based on a root locus of the transfer function $H_n(s) \cdot D_s(s) \cdot \omega_g(s)/\beta_{ref}(s)$, which is the transfer function arising by opening the loop prior to the speed controller. Figure C.5 sketches an excerpt from the pole-zero map. The pole in zero originates from the integral action of the PI controller, $\omega_{pole,1}$ denotes a pole of $\omega_g(s)/\beta_{ref}(s)$, and the zero is determined by the reset rate of the speed controller. The value of $\omega_{pole,1}$ changes according to the operating point, and to obtain similar responses in the two cases, the zero in both cases are placed just to the right of $\omega_{pole,1}$. This is done to let the loci stay on the real axis. For the relevant case, the following has been utilized:

$$T_{is} > \frac{1}{\omega_{pole,1}}$$

$$T_{is} = 1.065 \frac{1}{\omega_{pole,1}} \quad [\text{s}] \quad (\text{C.9})$$

Having determined the reset rates of the speed controllers, the proportional gains are selected in order to provide phase margins of 50° . This is verified in Figure C.6.

The parameters for the speed controllers are written in Table C.1.

Speed Controller 1		Speed Controller 2	
$K_{ps,1}$	$T_{is,1}$	$K_{ps,2}$	$T_{is,2}$
$-6.89^\circ/(\text{rad/s})$	25 s	$-2.95^\circ/(\text{rad/s})$	6.02 s

Table C.1: Parameters of the speed controllers. Speed Controller 1 are designed to operate at wind speeds in the region 12 – 15 m/s, while Speed Controller 2 are designed to operate at wind speeds in the region 15 – 25 m/s.

In order to evaluate the designed controllers in the extremity of their regions, the minimum and maximum parameters of the linearized model are applied. Hence, the step responses in Figure C.7

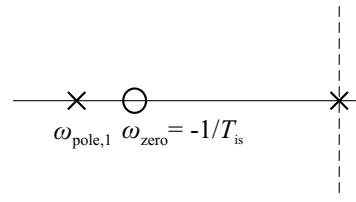


Figure C.5: Excerpts from the root locus of the transfer function $H_n(s) \cdot D_s(s) \cdot \omega_g(s)/\beta_{ref}(s)$.

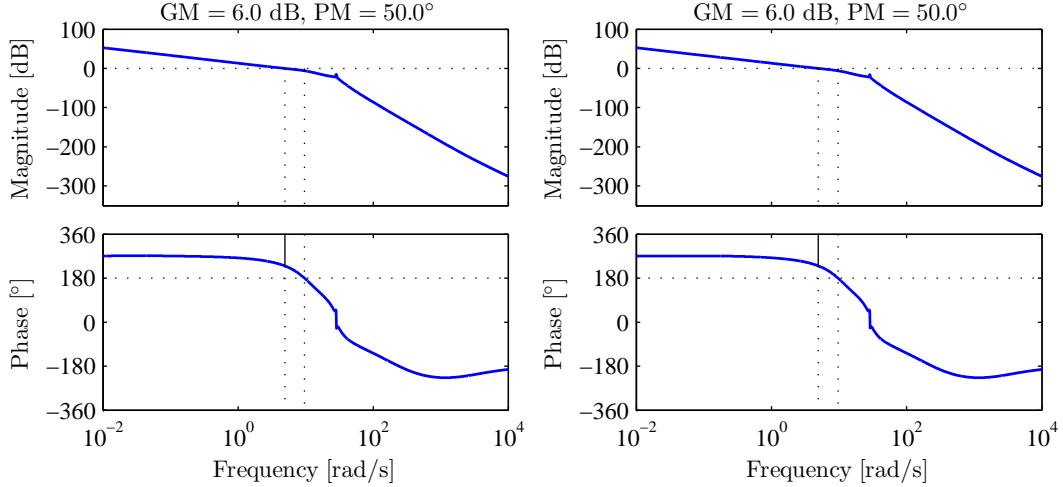


Figure C.6: *Bode plots of $H_n(s) \cdot D_s(s) \cdot \omega_g(s)/\beta_{ref}(s)$. The results in the left subplots are for Speed Controller 1 at an operating point wind speed of 15 m/s, while the right subplots display results for Speed Controller 2 at an operating point wind speed of 25 m/s.*

appear. Notice that the blue line in the left subplot corresponds to a stationary wind speed of 12.8 m/s, which is utilized since the gain of the system becomes zero for a wind speed equal to 12 m/s, when operating along the nominal operating trajectory. This is obvious from Figure C.4. According to Figure C.7 each controller gives rise to an overshoot at the higher wind speed of its operating range, where it is further verified that the two controllers behave equally, as dictated by the design. Furthermore, it becomes obvious that slow transient responses are provided at lower wind speeds.

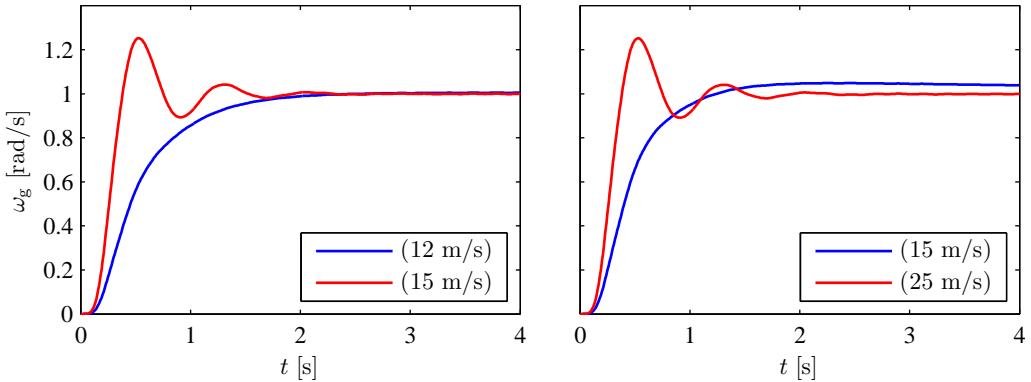


Figure C.7: *Step responses of the speed controllers plotted in the extremes of the operating area.*

So far the two speed controllers have been considered separately. However, at this point the conditions for making transitions between the two controllers are revealed; hereby, finalizing the gain scheduling approach. As the wind speed is considered unknown, the pitch angle is used as an indicator of the current wind speed. This approach is justified by the integral action in the PI controller, as this ensures asymptotic convergence towards the desired operating point for a given wind speed [Hammerum, 2006, p. 30]. Scheduling between the two speed controllers will then happen according to the conditions defined in Table C.2, using the method presented in Section C.4. A hysteresis is introduced through ϵ to avoid oscillations between the two speed controllers.

Evaluation of Speed Controller

Figure C.8 shows how the speed controller performs when keeping the generator torque constant. By doing so, the output power is proportional to the generator speed according to Eq. (3.28) on

Switch	Switching condition
$1 \rightarrow 2$	$\beta(t) \geq \beta_{12} + \epsilon$
$1 \leftarrow 2$	$\beta(t) \leq \beta_{12} - \epsilon$

Table C.2: *Switching conditions, where $\beta_{12} = 7.98^\circ$ and $\epsilon = 0.5^\circ$.*

Page 20. This means that the output power depends directly upon the correctness of the model of the power system and of the performance of the speed controller. To circumvent this dependency, a power controller is introduced in the next subsection in order to improve the tracking of the power reference based on a measurement of the power.

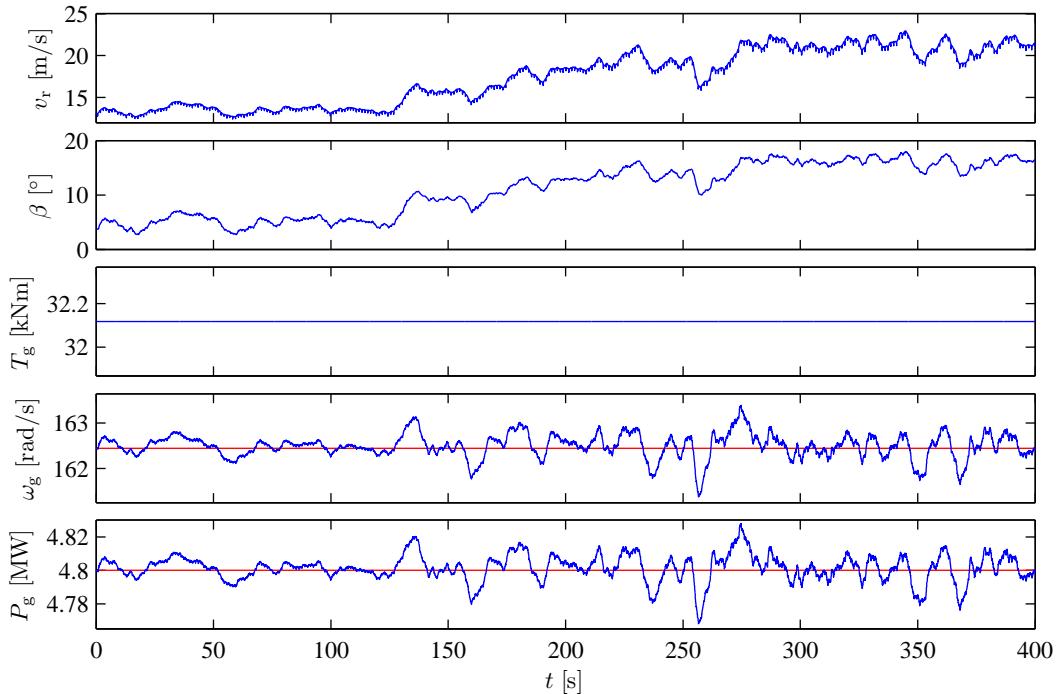


Figure C.8: *Simulation of the speed controller operating above the rated wind speed. Speed Controller 1 is active at time $0 \text{ s} \leq t < 70 \text{ s}$, while Speed Controller 2 operates the wind turbine the rest of the time. Note that the red lines are references.*

Power Controller

The power controller is implemented in order to cancel possible steady-state errors on the output power. This suggests using slow integral control for the power controller, as this will eventually cancel steady-state errors on the output power without interfering with the speed controller. However, it may be beneficial to make the power controller faster to improve accuracy in the tracking of the rated power. To follow this approach the power controller is realized as a PI controller on the form:

$$D_p(s) = K_{pp} \left(1 + \frac{1}{T_{ip} \cdot s} \right) \quad [\text{Nm/W}] \quad (\text{C.10})$$

where:

$D_p(s)$ is the PI controller ensuring that the generated power is close to the rated power [Nm/W]

K_{pp} is the proportional gain of $D_p(s)$ [Nm/W]

T_{ip} is the reset rate of $D_p(s)$ [s]

Applying the measured output power directly is an issue since the measurement is very noisy, as depicted by Section 3.10. This means that measurement noise has to be accounted for in the

design and yields that the proportional gain has to be sufficiently small. The proportional gain is chosen using a trial and error approach while the reset rate is selected large enough to avoid overshoot on the step response. The resulting parameters are shown in Table C.3.

K_{pp}	T_{ip}
$447 \cdot 10^{-6} \text{ Nm/W}$	0.031 s

Table C.3: *Parameters for the power controller.*

Evaluation of Power Controller

By adding the power controller to the controller structure, the reference controller operating in the full load region behaves as shown in Figure C.9.

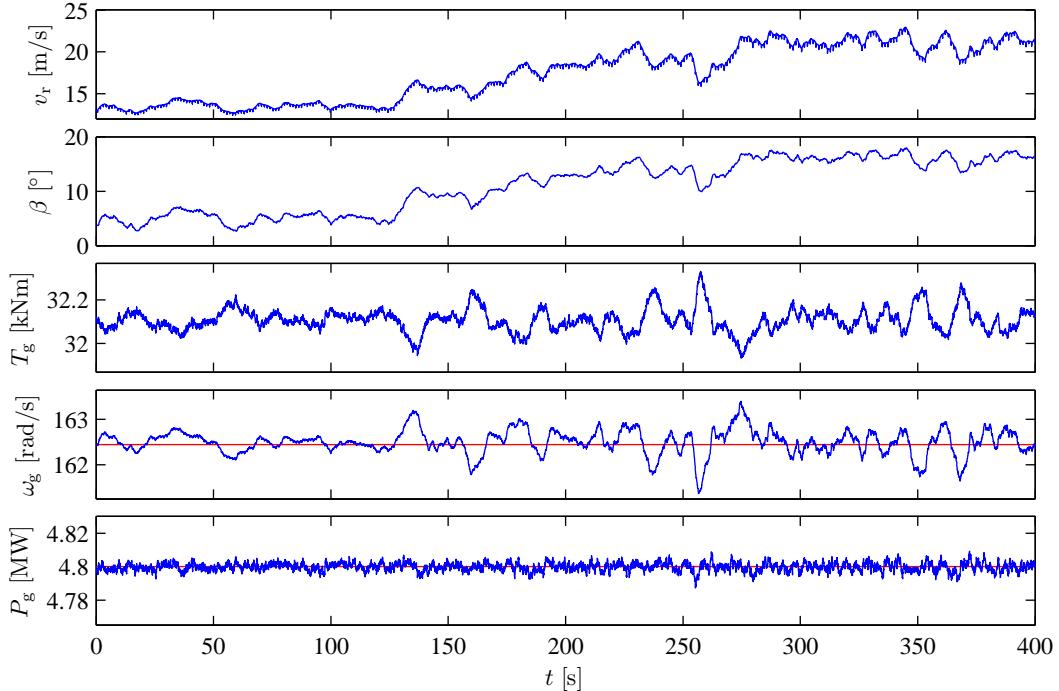


Figure C.9: *Simulation of the cooperative power controller and speed controller operating above the rated wind speed.*

When comparing Figure C.9 to Figure C.8 it is obvious that introducing the power controller results in improved tracking of the power reference. Notice that the differences obtained in the pitch angle and generator speed are neglectable. This states that the power controller improves tracking of the power reference without interfering with the speed controller.

The performance measures evaluating drive train stress and power fluctuations of the two configurations are shown in Table C.4. The results reveal that drive train stress is not increased much by introducing the power controller.

Active controllers	$\int_0^t \dot{\theta}_\Delta^2(\tau) d\tau$	$\int_0^t (P_{g,N} - P_g(\tau))^2 d\tau$
Speed Controller	3.57 nrad ² /s	25.8 GW ² s
Speed Controller and Power Controller	3.60 nrad ² /s	2.55 GW ² s

Table C.4: *Performance in terms of drive train stress and power fluctuations evaluated during the simulations shown in Figure C.8 and Figure C.9.*

At this stage both controllers for operating in partial load and full load have been designed. The next section presents the design of a filter which can be implemented in order to minimize the drive train stress.

C.3 Drive Train Stress Damper

Active drive train damping solutions are deployed in large horizontal-axis wind turbines to mitigate fatigue damage due to drive train oscillations. The idea is to add a component to the generator torque to compensate for the oscillations in the drive train. This signal should have a frequency equal to the eigenfrequency of the drive train, which can be found by filtering the measurement of the generator speed. When the output from the filter is added to the generator torque, the phase of the filter must be zero at the resonant frequency to achieve the desired damping effect.

The drive train damper is implemented to add a compensating torque as shown in Figure C.10.

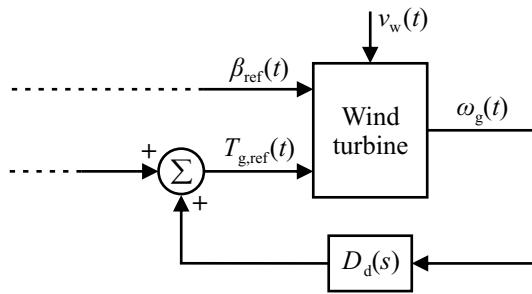


Figure C.10: Illustration of how the drive train stress damper can be implemented. The dashed lines indicates the control signals coming from the reference controller.

The following filter structure for the drive train damper is proposed and can be applied to dampen the eigenfrequency of the drive train [Dixit and Suryanarayanan, 2005, p. 1297]:

$$D_d(s) = K_d \frac{2\zeta_d \omega_{nd} s (1 + \tau_d s)}{s^2 + 2\zeta_d \omega_{nd} s + \omega_{nd}^2} \quad [\text{Nm}/(\text{rad/s})] \quad (\text{C.11})$$

where:

$D_d(s)$ is the band-pass filter $[\text{Nm}/(\text{rad/s})]$

K_d is the gain of the filter $[\text{Nm}/(\text{rad/s})]$

ω_{nd} is the undamped eigenfrequency of the filter $[\text{rad/s}]$

τ_d is the time constant which can be used for compensate for time lags in the system $[\text{s}]$

ζ_d is the damping ratio $[\cdot]$

The time constant τ_d introduces a zero in the filter, and can be used to compensate for time lags in the system. Therefore, τ_d is selected to equal the time delay of the converter system, $t_{g,d}$.

The spread of the peak in the amplitude of the frequency response is determined by the damping ratio ζ_d , and can be chosen small to give a narrow peak if the eigenfrequency is known precisely. A damping ratio of 0.25 gives a bode plot as shown in Figure C.11.

To determine the gain of the filter, K_d , a root locus is plotted for the transfer function from $T_{g,ref}(s)$ to $\omega_g(s)$ with the filter coupled in a positive feedback loop. The root locus plot is shown in Figure C.12. It is apparent from the figure that if the filter gain becomes too large the system becomes unstable due to the non-minimum phase behavior, introduced by the Padé approximation that resembles the communication delay of the actuator.

Due to the higher loads at higher wind speeds, it is favorable if the filter gain depends on the point of operation. A simple way of fulfilling this property is to apply different gains in the partial and full load configurations of the reference controller; these are 500 $\text{Nm}/(\text{rad/s})$ at partial load operation and 2,500 $\text{Nm}/(\text{rad/s})$ at full load operation.

By enabling the drive train stress damper in the simulations shown in Figure C.2 on Page 151 and Figure C.9 the drive train stress, evaluated as $\int_0^t \theta_\Delta^2(\tau) d\tau$, is reduced by 50% and 11%, respectively. Hereby, the drive train stress damper is successful in both regions. However, it increases power

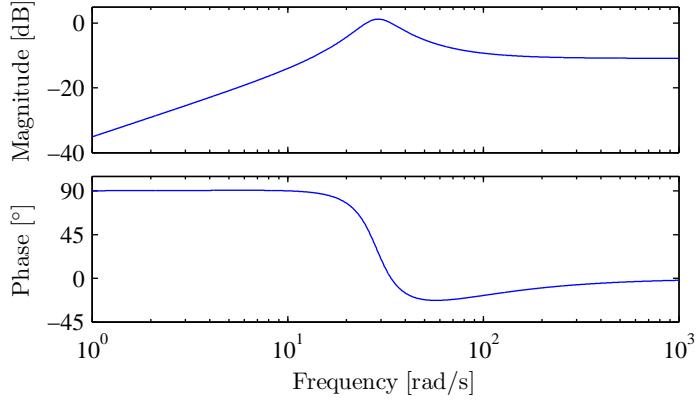
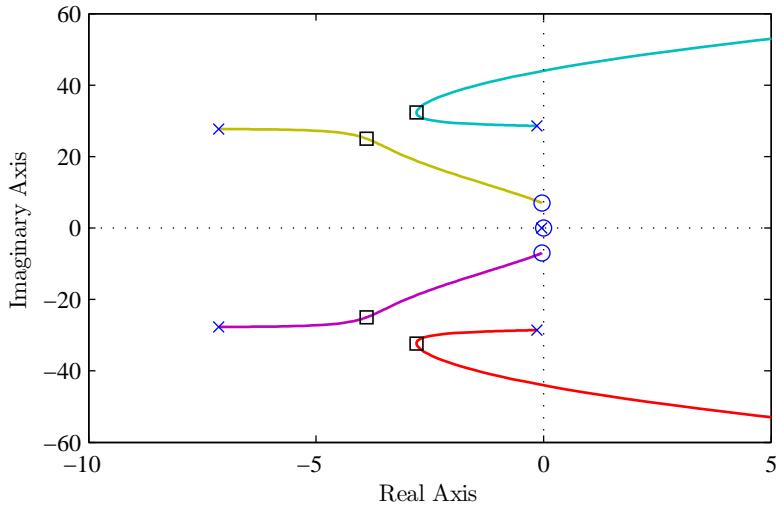

 Figure C.11: *Bode plot of the band-pass filter.*


Figure C.12: *Root locus plot of the designed band pass filter and the transfer function from $T_{g,\text{ref}}(s)$ to $\omega_g(s)$. The poles are marked with crosses and the zeros are marked with circles. The squares represent the location of the closed loop poles when applying the gain chosen for the full load configuration.*

fluctuation in full load operation by working against the power controller to some extent, although it is still beneficial to use the drive train stress damper and power controller in terms of drive train stress and power fluctuations, compared to maintaining the rated generator torque.

The next section addresses the bumpless transfer, which must ensure that no bumps exist on the control signal in the switch between two different controllers.

C.4 Bumpless Transfer

The purpose of this section is to explain how the bumpless transfer mechanism is designed; i.e. how and when to activate the switch illustrated in Figure 2.4 on Page 8. The considered transition is the one that brings the control system from partial load operation to full load operation, and vice versa.

When the control system switches from partial load to full load operation it is important that this transition is not apparent on the control signals; being the generator torque and pitch angle. This is known as bumpless transfer and is important because two controllers may not agree upon the magnitude of the control signal at the time that the transition happens. If a switch between two controllers is undertaken without bumpless transfer, a bump in the control signal may trigger

oscillations between the two controllers, making the system unstable.

The transition from partial to full load operation must happen as the wind speed becomes sufficiently large. For stationary wind speeds this happens at 12.29 m/s, when assuming a single aerodynamic model and absence of tower shadow and wind shear effects. However, it is not convenient to apply the wind speed as the switching condition, since the large inertia of the rotor causes the generator speed and output power to follow significantly later than a rise in the wind speed. Besides, the wind speed is poorly known. Therefore, it is more appropriate to utilize the generator speed as switching condition.

The switching conditions appear in Table C.5. Notice that ϵ is a small number that introduces hysteresis to ensure a minimum time between each transition.

Switch	Switching conditions
partial → full	$\omega_g(t) \geq \omega_{g,N}$
full → partial	$(\beta(t) \leq \beta_{opt}) \ \& \ (\omega_g(t) \leq \omega_{g,N} - \epsilon)$

Table C.5: *Switching conditions where $\omega_{g,N} = 162.45$ rad/s, $\beta_{opt} = 0^\circ$, and $\epsilon = 0.2$ rad/s.*

Due to the switching condition on $\beta(t)$, and because the output of the speed controller is saturated not to move below 0° , the transition already fulfills bumpless transfer for this control signal. For the generator torque signal a bumpless transfer is assured by adjusting an integral state in such a way that the generator torque will not change abruptly. The compensation torque is calculated using Eq. (C.12), and applies for the transition from Controller 1 to Controller 2; a similar equation applies for the reverse transition.

$$\begin{aligned} T_{g,1}(k) + T_{g,comp}(k-1) &= T_{g,2}(k) + T_{g,comp}(k) \\ T_{g,comp}(k) &= T_{g,1}(k) - T_{g,2}(k) + T_{g,comp}(k-1) \quad [\text{Nm}] \end{aligned} \quad (\text{C.12})$$

where:

$T_{g,1}(k)$ and $T_{g,2}(k)$ are the torque output from Controller 1 and 2 respectively [Nm]

$T_{g,comp}(k)$ is the compensation torque which ensures a bumpless transfer [Nm]

The compensation torque is not important when operating above the rated wind speed, because the power controller has integral action. When operating below rated wind speed the compensation torque is discharged to zero, as it otherwise would result in the optimal tip-speed ratio not being followed.

Derivation of Reliable LPV Controller Design Procedure

The purpose of this appendix is to derive the equations necessary to design a reliable LPV controller. The derivation is inspired by [Bianchi *et al.*, 2007, pp. 159-171].

The Bounded Real Lemma for LPV System with Parameter-Dependent Lyapunov Matrices

In this subsection the plant, controller, and closed-loop system matrices are set up. From these matrices the bounded real lemma is set up, assuming parameter-dependent Lyapunov matrices.

The LPV system is defined as shown below, where $D_{22}(\theta, \Delta) = 0$.

$$\dot{x}(t) = A(\theta, \Delta)x(t) + B_1(\theta, \Delta)w(t) + B_2(\theta, \Delta)u(t) \quad (\text{D.1a})$$

$$z(t) = C_1(\theta, \Delta)x(t) + D_{11}(\theta, \Delta)w(t) + D_{12}(\theta, \Delta)u(t) \quad (\text{D.1b})$$

$$y(t) = C_2(\theta, \Delta)x(t) + D_{21}(\theta, \Delta)w(t) + D_{22}(\theta, \Delta)u(t) \quad (\text{D.1c})$$

The following shorthand notation is defined for convenience:

$$A(\theta, \Delta) \triangleq A_0 + A^\theta(\theta) + A^\Delta(\Delta) \quad (\text{D.2})$$

where:

A^θ is the part of $A(\theta, \Delta)$ which is dependent on θ

A^Δ is the part of $A(\theta, \Delta)$ which is dependent on Δ

The LPV controller defined below is only dependent on the measured parameter vector, θ , not the unmeasured parameter vector, Δ , since it should be robust towards these parameter variations.

$$\dot{x}_c(t) = A_c(\theta)x_c(t) + B_c(\theta)y(t) \quad (\text{D.3a})$$

$$u(t) = C_c(\theta)x_c(t) + D_c(\theta)y(t) \quad (\text{D.3b})$$

By combining the system and the controller, the following closed-loop matrices can be set up:

$$A_{cl}(\theta, \Delta) = \begin{bmatrix} A(\theta, \Delta) + B_2(\theta, \Delta)D_c(\theta)C_2(\theta, \Delta) & B_2(\theta, \Delta)C_c(\theta) \\ B_c(\theta)C_2(\theta, \Delta) & A_c(\theta) \end{bmatrix} \quad (\text{D.4a})$$

$$B_{cl}(\theta, \Delta) = \begin{bmatrix} B_1(\theta, \Delta) + B_2(\theta, \Delta)D_c(\theta)D_{21}(\theta, \Delta) \\ B_c(\theta)D_{21}(\theta, \Delta) \end{bmatrix} \quad (\text{D.4b})$$

$$C_{cl}(\theta, \Delta) = [C_1(\theta, \Delta) + D_{12}(\theta, \Delta)D_c(\theta)C_2(\theta, \Delta) \quad D_{12}(\theta, \Delta)C_c(\theta)] \quad (\text{D.4c})$$

$$D_{cl}(\theta, \Delta) = D_{11}(\theta, \Delta) + D_{12}(\theta, \Delta)D_c(\theta)D_{21}(\theta, \Delta) \quad (\text{D.4d})$$

In the following the uncertainty is assumed only to exist in $A(\theta, \Delta)$. This implies that B_{cl} , C_{cl} , and D_{cl} become independent of Δ . Hereby, the closed-loop system can be written as follows, called T_{zw} :

$$\dot{x}(t) = A_{cl}(\theta, \Delta)x(t) + B_{cl}(\theta)w(t) \quad (\text{D.5a})$$

$$z(t) = C_{cl}(\theta)x(t) + D_{cl}(\theta)w(t) \quad (\text{D.5b})$$

The following definition is borrowed from [Bianchi *et al.*, 2007, p. 163]:

Given the compact set Θ and the hypercube \mathcal{V} , the continuous function $A_{\text{cl}}(\cdot)$ is parametrically-dependent quadratically (PDQ) stable if there exists a continuously differentiable symmetric function $\mathcal{X}(\cdot)$ such that $\mathcal{X}(\theta) > 0$ and

$$A_{\text{cl}}^T(\theta)\mathcal{X}(\theta) + \mathcal{X}(\theta)A_{\text{cl}} + \dot{\mathcal{X}} < 0 \quad (\text{D.6})$$

where

$$\dot{\mathcal{X}} = \sum_{i=1}^{n_\theta} \dot{\theta}_i \frac{\partial \mathcal{X}(\theta)}{\partial \theta_i} \quad (\text{D.7})$$

for all $(\theta, \dot{\theta}) \in \Theta \times \mathcal{V}$.

When the function $A_{\text{cl}}(\cdot)$ is PDQ stable, it is said that the LPV system in Eq. (D.5) is PDQ stable.

Using the parameter-dependent Lyapunov matrices, the bounded real lemma can be written on matrix form as shown below, based on the parameter-dependent Lyapunov matrices and the closed-loop system matrices. Notice that the Lyapunov matrices are independent of Δ , since this is the unmeasured parameters.

Given the LPV system $(A_{\text{cl}}(\theta, \Delta), B_{\text{cl}}(\theta), C_{\text{cl}}(\theta), D_{\text{cl}}(\theta))$ with $(\theta, \Delta, \dot{\theta}) \in \Theta \times \mathcal{D} \times \mathcal{V}$. Suppose that there exists a differentiable symmetric function $\mathcal{X}(\theta)$ such that $\mathcal{X}(\theta) > 0$ and

$$\begin{bmatrix} \dot{\mathcal{X}}(\theta) + A_{\text{cl}}^T(\theta, \Delta)\mathcal{X}(\theta) + \mathcal{X}(\theta)A_{\text{cl}}(\theta, \Delta) & \mathcal{X}(\theta)B_{\text{cl}}(\theta) & C_{\text{cl}}^T(\theta) \\ B_{\text{cl}}^T(\theta)\mathcal{X}(\theta) & -\gamma I & D_{11}^T(\theta) \\ C_{\text{cl}}(\theta) & D_{11}(\theta) & -\gamma I \end{bmatrix} < 0 \quad (\text{D.8})$$

for all $(\theta, \Delta, \dot{\theta}) \in \Theta \times \mathcal{D} \times \mathcal{V}$. Then,

1. the function $A_{\text{cl}}(\cdot)$ is PDQ stable over $\Theta \times \mathcal{D}$,
2. there exists a scalar δ with $0 \leq \delta < \gamma$ such that $\|T_{zw}\|_{i,2} \leq \delta$.

From these inequalities the reliable LPV controller can be designed, when the inequalities in Eq. (D.8) have been appropriately rearranged.

Transformation of the Bounded Real Lemma

In order to utilize the bounded real lemma for design it has to be transformed, to include the controller and system matrices directly. These transformations are shown in this subsection.

Two symmetric matrices are defined as:

$$\Pi_1 = \begin{bmatrix} \mathbf{Y} & I \\ M^T & 0 \end{bmatrix} \quad \Pi_2 = \begin{bmatrix} I & \mathbf{X} \\ 0 & N^T \end{bmatrix} \quad (\text{D.9})$$

A congruence transformation of Eq. (D.8) is performed with $\text{diag}(\Pi_1, I, I)$:

$$\begin{bmatrix} \Pi_1^T \dot{\mathcal{X}}(\theta) \Pi_1 + \Pi_1^T A_{\text{cl}}^T \mathcal{X} \Pi_1 + \Pi_1^T \mathcal{X} A_{\text{cl}} \Pi_1 & \Pi_1^T \mathcal{X} B_{\text{cl}} & \Pi_1^T C_{\text{cl}}^T \\ B_{\text{cl}}^T \mathcal{X} \Pi_1 & -\gamma I & D_{11}^T \\ C_{\text{cl}} \Pi_1 & D_{11} & -\gamma I \end{bmatrix} < 0 \quad (\text{D.10})$$

The Lyapunov function \mathcal{X} and its inverse can be partitioned as shown below.

$$\mathcal{X} = \begin{bmatrix} \mathbf{X} & N \\ N^T & \Upsilon_1 \end{bmatrix} \quad \mathcal{X}^{-1} = \begin{bmatrix} \mathbf{Y} & M \\ M^T & \Upsilon_2 \end{bmatrix} \quad (\text{D.11})$$

This partitioning is useful, since the following relations can be used when rewriting the equations:

$$\begin{aligned} \mathcal{X}\mathcal{X}^{-1} &= I \\ \begin{bmatrix} \mathbf{X} & N \\ N^T & \Upsilon_1 \end{bmatrix} \begin{bmatrix} \mathbf{Y} & M \\ M^T & \Upsilon_2 \end{bmatrix} &= I \\ \begin{bmatrix} \mathbf{XY} + NM^T & \mathbf{XM} + N\Upsilon_2 \\ N^T\mathbf{Y} + \Upsilon_1 M^T & N^T M + \Upsilon_1 \Upsilon_2 \end{bmatrix} &= I \end{aligned} \quad (\text{D.12})$$

The following relation is used later in the derivation:

$$\begin{aligned} \mathcal{X}\Pi_1 &= \begin{bmatrix} \mathbf{X} & N \\ N^T & \Upsilon_1 \end{bmatrix} \begin{bmatrix} \mathbf{Y} & I \\ M^T & 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{XY} + NM^T & \mathbf{X} \\ N^T\mathbf{Y} + \Upsilon_1 M^T & N^T \end{bmatrix} \\ &= \begin{bmatrix} I & \mathbf{X} \\ 0 & N^T \end{bmatrix} \\ &= \Pi_2 \end{aligned} \quad (\text{D.13})$$

To finalize the congruent transformation, the elements in Eq. (D.10) are rewritten as shown below.

$$\begin{aligned} \Pi_1^T \mathcal{X} A_{\text{cl}} \Pi_1 &= \Pi_2^T A_{\text{cl}} \Pi_1 \\ &= \begin{bmatrix} I & 0 \\ \mathbf{X} & N \end{bmatrix} \begin{bmatrix} A + B_2 D_c C_2 & B_2 C_c \\ B_c C_2 & A_c \end{bmatrix} \begin{bmatrix} \mathbf{Y} & I \\ M^T & 0 \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ \mathbf{X} & N \end{bmatrix} \begin{bmatrix} (A + B_2 D_c C_2)\mathbf{Y} + B_2 C_c M^T & A + B_2 D_c C_2 \\ B_c C_2 \mathbf{Y} + A_c M^T & B_c C_2 \end{bmatrix} \\ &= \begin{bmatrix} (A + B_2 D_c C_2)\mathbf{Y} + B_2 C_c M^T & \\ \mathbf{X}(A + B_2 D_c C_2)\mathbf{Y} + \mathbf{X}B_2 C_c M^T + N B_c C_2 \mathbf{Y} + N A_c M^T & \\ A + B_2 D_c C_2 & \\ \mathbf{X}(A + B_2 D_c C_2) + N B_c C_2 & \end{bmatrix} \end{aligned} \quad (\text{D.14})$$

$$\begin{aligned} \Pi_1^T \mathcal{X} B_{\text{cl}} &= \Pi_2^T B_{\text{cl}} \\ &= \begin{bmatrix} I & 0 \\ \mathbf{X} & N \end{bmatrix} \begin{bmatrix} B_1 + B_2 D_c D_{21} \\ B_c D_{21} \end{bmatrix} \\ &= \begin{bmatrix} B_1 + B_2 D_c D_{21} \\ \mathbf{X}(B_1 + B_2 D_c D_{21}) + N B_c D_{21} \end{bmatrix} \end{aligned} \quad (\text{D.15})$$

$$\begin{aligned} C_{\text{cl}} \Pi_1 &= [C_1 + D_{12} D_c C_2 \quad D_{12} C_c] \begin{bmatrix} \mathbf{Y} & I \\ M^T & 0 \end{bmatrix} \\ &= [(C_1 + D_{12} D_c C_2)\mathbf{Y} + D_{12} C_c M^T \quad C_1 + D_{12} D_c C_2] \end{aligned} \quad (\text{D.16})$$

$$\begin{aligned} \Pi_1^T \mathcal{X} \Pi_1 &= \Pi_1^T \Pi_2 \\ &= \begin{bmatrix} \mathbf{Y} & M \\ I & 0 \end{bmatrix} \begin{bmatrix} I & \mathbf{X} \\ 0 & N^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Y} & \mathbf{YX} + MN^T \\ I & \mathbf{X} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Y} & I \\ I & \mathbf{X} \end{bmatrix} \end{aligned} \quad (\text{D.17})$$

$$\begin{aligned}
 \dot{\Pi}_1^T \mathcal{X} \Pi_1 + \Pi_1^T \dot{\mathcal{X}} \Pi_1 + \Pi_1^T \mathcal{X} \dot{\Pi}_1 &= \begin{bmatrix} \dot{\mathbf{Y}} & 0 \\ 0 & \dot{\mathbf{X}} \end{bmatrix} \\
 \Pi_1^T \dot{\mathcal{X}} \Pi_1 &= \begin{bmatrix} \dot{\mathbf{Y}} & 0 \\ 0 & \dot{\mathbf{X}} \end{bmatrix} - (\dot{\Pi}_1^T \mathcal{X} \Pi_1 + \Pi_1^T \mathcal{X} \dot{\Pi}_1) \\
 &= \begin{bmatrix} \dot{\mathbf{Y}} & 0 \\ 0 & \dot{\mathbf{X}} \end{bmatrix} - \left(\begin{bmatrix} \dot{\mathbf{Y}} & \dot{M} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & \mathbf{X} \\ 0 & N^T \end{bmatrix} + \Pi_1^T \mathcal{X} \dot{\Pi}_1 \right) \\
 &= \begin{bmatrix} \dot{\mathbf{Y}} & 0 \\ 0 & \dot{\mathbf{X}} \end{bmatrix} - \left(\begin{bmatrix} \dot{\mathbf{Y}} & \dot{\mathbf{Y}}\mathbf{X} + \dot{M}N^T \\ 0 & 0 \end{bmatrix} + \Pi_1^T \mathcal{X} \dot{\Pi}_1 \right) \\
 &= \begin{bmatrix} \dot{\mathbf{Y}} & 0 \\ 0 & \dot{\mathbf{X}} \end{bmatrix} - \begin{bmatrix} 2\dot{\mathbf{Y}} & \dot{\mathbf{Y}}\mathbf{X} + \dot{M}N^T \\ \mathbf{X}\dot{\mathbf{Y}} + N\dot{M}^T & 0 \end{bmatrix} \\
 &= \begin{bmatrix} -\dot{\mathbf{Y}} & -\dot{\mathbf{Y}}\mathbf{X} + \dot{M}N^T \\ -\mathbf{X}\dot{\mathbf{Y}} + N\dot{M}^T & \dot{\mathbf{X}} \end{bmatrix} \tag{D.18}
 \end{aligned}$$

Substitution of Variables in the Bounded Real Lemma

In order to reduce the number of multiplications of unknown variables in the optimization problem, auxiliary matrices are defined and substituted into the transformed bounded real lemma.

The auxiliary matrices used for the substitution are shown below.

$$\hat{\mathbf{A}}_\Delta = NA_c M^T - \mathbf{X}\dot{\mathbf{Y}} - N\dot{M}^T + \mathbf{X}(A_0 + A^\theta + B_2 D_c C_2)\mathbf{Y} + \mathbf{X}B_2 C_c M^T + NB_c C_2 \mathbf{Y} \tag{D.19a}$$

$$\hat{\mathbf{B}}_\Delta = NB_c + \mathbf{X}B_2 D_c \tag{D.19b}$$

$$\hat{\mathbf{C}}_\Delta = C_c M^T + D_c C_2 \mathbf{Y} \tag{D.19c}$$

$$\hat{\mathbf{D}}_\Delta = D_c \tag{D.19d}$$

Substituting these matrices into the inequalities of the transformed bounded real lemma, the following appears:

$$\begin{bmatrix} -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2 \hat{\mathbf{C}}_\Delta + (*) & \hat{\mathbf{A}}_\Delta^T + \mathbf{Y} A^{\Delta T} \mathbf{X} + A + B_2 \hat{\mathbf{D}}_\Delta C_2 & * & * \\ * & \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}_\Delta C_2 + (*) & * & * \\ (B_1 + B_2 \hat{\mathbf{D}}_\Delta D_{21})^T & (\mathbf{X}B_1 + \hat{\mathbf{B}}_\Delta D_{21})^T & -\gamma I_{n_w} & * \\ C_1 \mathbf{Y} + D_{12} \hat{\mathbf{C}}_\Delta & C_1 + D_{12} \hat{\mathbf{D}}_\Delta C_2 & D_{11} + D_{12} \hat{\mathbf{D}}_\Delta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{D.20}$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \tag{D.21}$$

To make this optimization problem look like the notation found in e.g. [Bianchi *et al.*, 2007], the following matrix is utilized in an congruent transformation:

$$\Pi_3 = \begin{bmatrix} 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{D.22}$$

Given the open-loop LPV system governed by Eq. (D.1) on Page 161. Suppose that there exists two parameter-dependent symmetric matrices \mathbf{X} , \mathbf{Y} and four parameter-dependent matrices $\hat{\mathbf{A}}_\Delta$,

$\hat{\mathbf{B}}_\Delta$, $\hat{\mathbf{C}}_\Delta$ and $\hat{\mathbf{D}}_\Delta$ such that for all $(\theta, \Delta, \dot{\theta}) \in \Theta \times \mathcal{D} \times \mathcal{V}$,

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}_\Delta C_2 + (*) & * & * \\ \hat{\mathbf{A}}_\Delta^\top + \mathbf{Y}A^{\Delta\top}\mathbf{X} + A + B_2\hat{\mathbf{D}}_\Delta C_2 & -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2\hat{\mathbf{C}}_\Delta + (*) & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}_\Delta D_{21})^\top & (B_1 + B_2\hat{\mathbf{D}}_\Delta D_{21})^\top & -\gamma I_{n_w} \\ C_1 + D_{12}\hat{\mathbf{D}}_\Delta C_2 & C_1\mathbf{Y} + D_{12}\hat{\mathbf{C}}_\Delta & D_{11} + D_{12}\hat{\mathbf{D}}_\Delta D_{21} -\gamma I_{n_z} \end{bmatrix} < 0 \quad (\text{D.23})$$

$$\begin{bmatrix} \mathbf{X} & I \\ I & \mathbf{Y} \end{bmatrix} > 0 \quad (\text{D.24})$$

Then, there exists a controller of the form in Eq. (D.3) such that

1. the closed-loop system in Eq. (D.5) is PDQ stable over $\Theta \times \mathcal{D}$ and,
2. the induced \mathcal{L}_2 -norm of the operator T_{zw} is bounded by $\gamma > 0$ (i.e. $\|T_{zw}\|_{i,2} < \gamma$).

Necessary Conditions for the Bounded Real Lemma

Finding a feasible point in Eq. (D.23) is not trivial, since it is a BMI. Therefore, two necessary conditions are set up for two sets of variables, and when inserting one of these sets into Eq. (D.23) it becomes a convex optimization problem.

The necessary conditions can be obtained by applying the projection lemma, defined in [Bianchi et al., 2007, p. 156], on Eq. (D.23). Both necessary conditions are shown below:

$$\begin{bmatrix} \dot{\mathbf{X}} + \mathbf{X}A + \hat{\mathbf{B}}_\Delta C_2 + (*) & * & * \\ (\mathbf{X}B_1 + \hat{\mathbf{B}}_\Delta D_{21})^\top & -\gamma I_{n_w} & * \\ C_1 + D_{12}\hat{\mathbf{D}}_\Delta C_2 & D_{11} + D_{12}\hat{\mathbf{D}}_\Delta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (\text{D.25})$$

for all $\theta \in \theta_{\text{vex}}$, $\Delta \in \Delta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$, and

$$\begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * \\ B_{1,i}^\top \mathbf{X}_i & 0 & * \\ 0 & 0 & 0 \end{bmatrix} \geq 0 \quad (\text{D.26})$$

for $i = 1 \dots n_\theta$ and $\Delta \in \Delta_{\text{vex}}$.

$$\begin{bmatrix} -\dot{\mathbf{Y}} + A\mathbf{Y} + B_2\hat{\mathbf{C}}_\Delta + (*) & * & * \\ (B_1 + B_2\hat{\mathbf{D}}_\Delta D_{21})^\top & -\gamma I_{n_w} & * \\ C_1\mathbf{Y} + D_{12}\hat{\mathbf{C}}_\Delta & D_{11} + D_{12}\hat{\mathbf{D}}_\Delta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \quad (\text{D.27})$$

for all $\theta \in \theta_{\text{vex}}$, $\Delta \in \Delta_{\text{vex}}$, $\dot{\theta} \in \dot{\theta}_{\text{vex}}$, and

$$\begin{bmatrix} A_i \mathbf{Y}_i + (*) & * & * \\ 0 & 0 & * \\ C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \geq 0 \quad (\text{D.28})$$

for $i = 1 \dots n_\theta$ and $\Delta \in \Delta_{\text{vex}}$.

The necessary conditions can be utilized to find one half of the unknown variables in Eq. (D.23) such that it becomes an LMI.

Affine Case

In the general case explained in the previous subsection, designing an LPV controller involves solving infinitely many inequalities. The number of inequalities can be reduced by assuming that the unknown matrices in Eq. (D.23) has an affine dependence of θ and Δ , as in:

$$\mathbf{X}(\theta) = \mathbf{X} + \sum_{i=1}^{n_\theta} \theta_i \mathbf{X}_i \quad \mathbf{Y}(\theta) = \mathbf{Y}_0 + \sum_{i=1}^{n_\theta} \theta_i \mathbf{Y}_i \quad \hat{\mathbf{A}}_\Delta(\theta) = \hat{\mathbf{A}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{A}}_{\Delta i} \quad (\text{D.29a})$$

$$\hat{\mathbf{B}}_\Delta(\theta) = \hat{\mathbf{B}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{B}}_{\Delta i} \quad \hat{\mathbf{C}}_\Delta(\theta) = \hat{\mathbf{C}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{C}}_{\Delta i} \quad \hat{\mathbf{D}}_\Delta(\theta) = \hat{\mathbf{D}}_{\Delta 0} + \sum_{i=1}^{n_\theta} \theta_i \hat{\mathbf{D}}_{\Delta i} \quad (\text{D.29b})$$

In the following, the inequality shown in Eq. (D.23) is rewritten to be a function of the vertices of the parameter space. First, the matrix written below is defined to shorten the notation in the next part.

$$\Phi(\theta, \Delta, \dot{\theta}) \triangleq \begin{bmatrix} \dot{\mathbf{X}}(\theta) + \mathbf{X}(\theta)A(\theta, \Delta) + \hat{\mathbf{B}}_\Delta(\theta)C_2 + (*) \\ \hat{\mathbf{A}}_\Delta^\top(\theta, \Delta) + \mathbf{Y}(\theta)A^{\Delta\top}\mathbf{X}(\theta) + A(\theta, \Delta) + B_2\hat{\mathbf{D}}_\Delta(\theta)C_2 \\ (\mathbf{X}(\theta)B_1(\theta) + \hat{\mathbf{B}}_\Delta(\theta)D_{21})^\top \\ C_1(\theta) + D_{12}\hat{\mathbf{D}}_\Delta(\theta)C_2 \\ * & * & * \\ -\dot{\mathbf{Y}}(\theta) + A(\theta, \Delta)\mathbf{Y}(\theta) + B_2\hat{\mathbf{C}}_\Delta(\theta) + (*) & * & * \\ (B_1(\theta) + B_2\hat{\mathbf{D}}_\Delta(\theta)D_{21})^\top & -\gamma I_{n_w} & * \\ C_1(\theta)\mathbf{Y}(\theta) + D_{12}\hat{\mathbf{C}}_\Delta(\theta) & D_{11} + D_{12}\hat{\mathbf{D}}_\Delta(\theta)D_{21} & -\gamma I_{n_z} \end{bmatrix} \quad (\text{D.30})$$

The matrix $\Phi(\theta, \Delta, \dot{\theta})$ can be written for any value, θ_* , in the parameter space $(\theta, \Delta, \dot{\theta}) \in \Theta \times \mathcal{D} \times \mathcal{V}$ as a function of the vertices of the parameter space and an additional term. This expression is shown below, where θ is a scalar (i.e. $i = 1$).

$$\Phi(\theta_*) = \alpha\Phi(\theta_{v_1}) + (1 - \alpha)\Phi(\theta_{v_2}) + (\alpha^2 - \alpha)(\theta_{v_1} - \theta_{v_2})^2 \begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * & * \\ \mathbf{Y}_i A^{\Delta\top} \mathbf{X}_i & A_i \mathbf{Y}_i + (*) & * & * \\ B_{1,i}^\top \mathbf{X}_i & 0 & 0 & * \\ 0 & C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \quad (\text{D.31})$$

This implies that if $\Phi(\theta_{v_1}) < -\epsilon$ and $\Phi(\theta_{v_2}) < -\epsilon$, then:

$$\begin{bmatrix} \mathbf{X}_i A_i + (*) & * & * & * \\ \mathbf{Y}_i A^{\Delta\top} \mathbf{X}_i & A_i \mathbf{Y}_i + (*) & * & * \\ B_{1,i}^\top \mathbf{X}_i & 0 & 0 & * \\ 0 & C_{1,i} \mathbf{Y}_i & 0 & 0 \end{bmatrix} \geq \frac{-4\epsilon}{(\theta_{v_1} - \theta_{v_2})^2} \quad (\text{D.32})$$

$$\epsilon \geq 0 \quad (\text{D.33})$$

In [Bianchi *et al.*, 2007, pp. 159-171] ϵ is zero, but this restricts the structure of \mathbf{X}_i and \mathbf{Y}_i as these should be in the null space of $B_{1,i}^\top$ and $C_{1,i}$, respectively.

In this appendix a method for designing a passive fault-tolerant controller has been derived, which are based on parametric uncertainties. The method is derived for the general case and in the case of affine parameter dependencies. Additionally, a numerical trick for relaxing the restrictions on the structure of the parameter-dependent Lyapunov matrices has been presented.

Contents of Attached DVD

- Bibliography with internet sources as PDFs.
- Thesis in digital form.
- Model parameters.
- Tuning parameters.
- MATLAB files.
- Simulink models.