Model Predictive LIDAR Control of Wind Turbines for Load Mitigation

AALBORG UNIVERSITET

Department of Electronic Systems
Electronical Engineering
Frederik Bajers vej 7a
9220 Aalborg Øst
Tlf. 99 40 99 40
http://es.aau.dk

Title: Model Predictive LIDAR Control of Wind Turbines for Load Mitigation

Theme: Master Thesis

Project period: 1/9 2010 – 6/6 2011

Project group:
1035

Members of the group:
Lars Jespersen
Marc Westring Krogen

Supervisors:
Thomas Bak
Fabiano Daher Adegas

Synopsis:

This master thesis deals with the design of a model predictive controller (MPC) for load mitigation by including LIDAR measurements in the prediction.

For simulation purposes a nonlinear model of a wind turbine is derived. The model is validated by its physical characteristics and by comparing it to another model with the same level of detail and a more complex model in the aeroelastic program LACflex. For use in the developed MPC the nonlinear model is linearized.

For use together with the developed MPC a LIDAR model is developed. The developed LIDAR model averages wind speed measurements obtained from the wind field by the LIDAR model included in LACflex.

A MPC controller that uses LIDAR measurements in the prediction is developed and tested in MATLAB and LACflex. The MPC is tested in MATLAB on the derived nonlinear model, against a benchmark controller consisting of a standard PI controller with additional damping on tower and drive train. In LACflex the MPC is tested on a more complex turbine model and the included PI controller in LACflex as benchmark.

Simulation results showed that by including LIDAR in MPC the load can be mitigated and the power can be maintained.

Number of copies: 6
Number of pages: 120
Appendixes and attachments: 2 appendixes in the report, 1 CD enclosed
Project completed: 6/6 - 2011

Preface

This thesis is the documentation of a master project within Intelligent Autonomous Systems under the department Electronic Systems at Aalborg University written by Marc Westring Krogen and Lars Jespersen. The project is written during the fall semester 2010 and the spring semester 2011.

In this master project it was chosen to work with the project ”Model predictive LIDAR control of wind turbines” which was proposed by Jacob Deleuran Grunnet from LAC engineering and Thomas Bak at Aalborg University.

During the thesis, external references are used. This is done by a number as in [4] or with a number and a page reference as in [5, p.12].

The project is divided into the following five sections. An introduction and motivation to the problem this thesis deals with. A general system description to get the reader familiar with some of the terms used during the thesis. The development of a turbine and LIDAR model. The development of a classic controller that serves as a benchmark to the in same section developed model predictive controller, before they are simulated in MATLAB. Last a test of the controller on a more advanced model which includes also an observer has to be developed.

With this report is a CD containing a digital copy of the report and some of the references.

Marc Westring Krogen Lars Jespersen

Contents

Contents **8**

1 Introduction **9**

1.1 Background and Motivation 9

1.2 Related Work 10

1.3 Problem Formulation 11

1.4 Scope 12

1.5 Objectives 12

1.6 Thesis Outline 13

2 System Description **15**

2.1 Turbine Description 15

2.2 LIDAR 17

2.3 General Control Strategy 17

2.4 LACflex 20

3 System Modeling **21**

3.1 Model Approach 21

3.2 Model Structure 22

3.3 Wind Model 23

3.4 Aerodynamic Model 24

3.5 Drive Train Model 31

3.6 Generator Model 33

3.7 Tower Model 34

3.8 Pitch and torque actuator models 36

3.9 Combined Model 37
3.10 Model Verification 38
3.11 LIDAR Sensor Model 45

4 Control Design **52**
4.1 Control Design Approach 52
4.2 Controller Test 52
4.3 Benchmark Controller 57
4.4 Model Predictive Control 70
4.5 Test of the Developed MPC Controller 81

5 Simulation in LACflex **88**
5.1 Introduction to Simulations in LACflex 88
5.2 The Turbine Model in LACflex 88
5.3 Validation of Model in LACflex 89
5.4 Design of State Observer 99
5.5 LACflex Simulation Results 100

6 Conclusion **106**

7 Perspectives **108**
7.1 Turbine Model 108
7.2 LIDAR Model 108
7.3 Controller 108

A Results for MPC Test in MATLAB **110**

B Results for MPC Test in LACflex **114**

Bibliography **117**

Chapter 1

Introduction

This chapter describes the background and motivation for making a thesis about wind turbines and LIDAR. The chapter continues with a description of related work and the hypothesis of the thesis is stated in the problem formulation. Next is a description of the scope and objectives before the content of the thesis is outlined.

1.1 Background and Motivation

Energy is the life blood of our society and plays a big role in our daily life. Daily activities such as driving around in your car, switching on the lights or heating your home they requires energy. Most of the energy today comes from fossil fuels, which plays a crucial role in the emission of greenhouse gases. In Denmark fossil fuels accounts for 80% of the energi consumption [22].

In the European Union (EU) the countries have agreed, that the emission should be reduced by 20% in 2020 and be almost independent of fossil fuels with a 80 to 95% reduction in 2050 [10]. To meet the demand of reducing the emission of greenhouse gases, investments in the order of 1 trillion euro are needed over the next ten years [10]. Another advantage of reducing the use of fossil fuels is to be independent from fossil fuel producing countries. The increasing demand from third world contries affects the market [22] together with world events. Today our society is dependent of gas from countries where political issues affects the market and prices [14]. A more reliable energy source is crucial to keep the economic growth.

The need for alternative and environmentally friendly energy increases in correlation with the substitution of fossil fuels and the increasing demand of energy. A part of the solution is to use renewable energies such as wind, solar and water, but also nuclear power plants could be a part of the solution [10]. The safety of nuclear power plants has always been an issue after the Chernobyl accident, but has in the last couple of months been an even bigger topic after the accident in Japan where an earthquake and tsunami damaged nuclear power plants. The fear for a serious nuclear disaster is spread around the world and for instance in Germany several nuclear power plants were closed. Germany is therefore in need of a new and environment friendly energy source such as wind [18] [29].

Wind energy is expected to be the largest contributer in the replacement of fossil fuels [3] and is expected to cover one-fifth of EU's electricity demand in 2020, one third in 2030 and half in 2050. To make wind energy cost competitive relies on development and research into how energy is best extracted from the wind. One challenge is the placement of the turbines. The number of turbines and the size of them makes it difficult to place them and they have to be placed further away in less favorable wind and construction conditions. Another area is optimizing the wind farm layouts and use more advanced control strategies in wind farms. One of challenges to make wind energy cost competitive is the construction of more and bigger turbines. Bigger turbines create a number of new challenges, as larger wind turbines have to withstand larger mechanical loads, leading to very heavy constructions. A possible solution to the problem is the usage of more advanced controllers, which are able to minimize the loads on the turbine and still maintain high performance. This would mean that the turbines could be constructed with less materials or it could be possible to construct bigger turbines, thus making wind energy a cheaper and even more preferable energy source [4].

More advanced controllers and/or more advanced sensors could be the solution to the problems. As the turbines gets bigger they are affected more from wind phenomenas, such as wind shear, or when the turbines is placed in wind farms there can be an influenced by the wake from other turbines and this can lead to larger fatigues. A sensor that can measure the wind in greater detail and a controller to utilize it, could possibly reduce the fatigue. A possible solution to the sensor problem is a LIDAR (Light Detection And Ranging), which can measure the wind speed at a certain distance upwind of the rotor plane. Placing a LIDAR on the turbine could provide knowledge of the mean wind, turbulence, etc. Using the LIDAR knowledge in the design of a controller and the knowledge of the future wind to predict how to react in the future may reduce the fatigue on the turbine.

1.2 Related Work

The technology within wind turbines has developed rapidly. From fixed speed, stall controlled turbines to today with variable speed and variable pitch [5, p. 56]. As new technologies are developed the size of the turbines together with the amount of produced power are increased as well. In an attempt to produce cheaper power, mitigating fatigues is an important part of the development, as lower fatigue leads to lighter and cheaper structures. In the aim of reducing the fatigue several different approaches are used. This section deals with some of the current approaches as well as ideas that are currently being tested.

Pitching the blades is one area where several approaches are developed and under way. A proper control of the blade pitch angle has proved to be an efficient way of reducing fatigue. As the rotor sizes increase so does the effect from wind shear. Individual pitch has been used successfully for mitigating the loads in [7] and [4]. In [4] it is proved that also a more advanced pitch system such as changing the pitch of the tip more than the root could reduce the load. Adding flaps on the blades as on airplains is also mentioned as an option for further load reduction. Not only the pitch system but also new designs of the blades and other materials are mentioned as ways to reduce the loads.

Mechanical changes is one way of reducing the loads. Another less expensive approach is to change the control systems. One way is to include an extra input to the control system, by using an accelerometer to measure the acceleration of the tower, which is done in [6]. The new measurement is used in the control to calculate an extra pitch contribution to the original pitch which help damp the tower motion. In the same paper the load on the drive train is reduced, by adding small ripples to the generator torque. By using the measurement of the generator speed and filtering it with a filter that amplifies the natural frequency of the drive train, an extra torque with the frequency of the drive train is added to the generator torque. The extra contribution to the torque reduces the drive train torsion.

Combining mechanical solutions with control, such as using light detection and ranging (LIDAR) to get a preview of the wind before it hits the turbine, are by many thought to have a large potential. By measuring the wind in front of the turbine and combining it with control that uses this knowledge to preemptive control the turbine the load on the turbine might be mitigated. Different papers have investigated the advantage of using LIDAR in different ways. Some have used LIDAR as a feedforward together with PI controller where others have tried more modern control theories. The LIDAR is in the different situations modeled in different ways. In some papers only a few measurements of the wind is used where in others several measurements are used. But something is the same in many models such as wind is assumed frozen, meaning the wind does not change from it is measured to it affects the turbine. Another assumption often used, is that the horizontal and vertical wind component is zero for small angles between the LIDAR and upwind [12], [23] and [30].

1.3 Problem Formulation

Based on the previous section, describing the current development within the wind turbine industry and the potential in the use of a LIDAR sensor the main hypothesis is formulated:

It is possible to use a LIDAR sensor in the control of a wind turbine, with the goal to minimize dynamic loads on the mechanical parts of the turbine while still maintaining power production.

The hypothesis can be divided into subhypotheses, that if all are accepted, the main hypothesis is accepted as well. To test the main hypothesis a controller has to be designed that uses LIDAR in the control. This is the first subhypothesis:

- It is possible to design a controller that uses LIDAR measurements in the control.

If this is not possible the LIDAR sensor is not useful. If it is possible to use LIDAR measurements in a controller, the advantage of including it is the next subhypothesis. Including LIDAR may not have a positive influence in minimizing the fatigues. Therefore the second subhypothesis is:

- It is possible to reduce the fatigue by including LIDAR measurements in the control .

It may be possible to reduce the fatigue, but perhaps the disadvantage is that the power changes. The goal of a turbine is to produce power and it is therefore important to maintain the power production.

- It is possible to maintain the power production and power quality while minimizing the fatigue.

1.4 Scope

The scope of this thesis is to determine if knowledge about the incoming wind, measured with a LIDAR sensor, can be used with a model predictive controller to minimize fatigue. The fatigues that are tried minimized are the tower fore-aft mode and the drive train torsion. The tower is except from the blades, the most expensive part on a turbine [32, p.331]. The drive train is also an expensive part and the weight of the drive train influences the tower structure.

For generating wind and simulations the aeroelastic program LACflex is used. The program is provided by the company LAC engineering, who proposed this project. The program is used for the final tests as tests on a real turbine is not possible.

Real wind measurements are not made available for this thesis, therefore the wind fields that are used in this thesis are generated by a wind model in LACflex.

No real data for a LIDAR sensor is made available. Instead LIDAR measurements are simulated from the generated wind fields, where the measurements are several points of the incoming wind field. The LIDAR measurements are in LACflex defined as a circle of point wind speeds from the simulated wind field, at a time before the wind affects the turbine. The radius and the time can be set to different values.

To manage the project and achieve the goals in the thesis an iterative, incremental, similar to scrum, approach is taken. The approach helps to ensure that both modeling and control is covered, by early in the process establish knowledge about wind turbines and control strategy and from there iterative increase the system.

1.5 Objectives

The objectives at the end of the thesis:

- Validated nonlinear model of the system suitable for simulation and as a basis for the linear model. The model is validated by its physical characteristics, by comparison to another model at the same level of detail and a more advanced nonlinear model in LACflex.
- Validated linear model suitable for control. The model is validated up against the nonlinear model.
- Validated benchmark controller with some level of fatigue damage reduction. The controller is similar in complexity to currently used by the industry and validated by its ability to follow a realistic control strategy. It may serve as a benchmark for assessment and comparison of control designs developed as part of this thesis.
- LIDAR models for use in simulation and control. The models are tested by simulating the system with the same model predictive controller but with different LIDAR models.

The damage equivalent loads for tower fore-aft mode and drive train torsion along with standard deviations for the power and pitch are calculated for all situations and compared with the result for the benchmark controller.

- Validated model predictive controller structures and designs that aim to minimize fatigue on the tower fore-aft mode and drive train by including LIDAR measurements. The validation is carried out by simulation on a simple model in MATLAB and a more advance model in LACflex. The effectiveness is assessed by calculating damage equivalent loads for tower fore-aft mode and drive train torsion along with standard deviations for the power and pitch.
- A comparison and assessment of the developed control strategies and controllers regarding the benefit of using LIDAR, together with a discussion about the used LIDAR model.

1.6 Thesis Outline

This section is an overview of the different chapters the thesis is divided into and what they include, thereby providing an overview of the project.

Chapter 2: System Description
The purpose of this chapter is to introduce some of the terms and concepts used during the thesis. A description of the parts of a turbine and the concept in controlling a turbine are explained together with an introduction to LIDAR and LACflex. The goal of this chapter is o provide a common understanding of the general system the thesis deals with, and provide the fundamentals for the later developed turbine model, controller and final test.

Chapter 3: System Modeling
In this chapter the derivation of a nonlinear model of a turbine usable for simulation purposes and the derivation of a linear model usable for control are described. Along with the linear and non linear model, is a description of a wind model and the derivation of a LIDAR model usable for simulations.

Chapter 4: Control Design
In this chapter the development and test of controllers are described. The developed controllers are a PI controller with additional damping on the tower and drive train and a model predictive controller (MPC) that uses LIDAR measurements in the prediction. The PI controller serves as the benchmark for the MPC controller. The test is a simulation in MATLAB and the controllers are compared by the damage equivalent load which also is described in this chapter.

Chapter 5: Simulation in LACflex
In this chapter is a description of the final test in the aeroelastic code LACflex which also includes the development of an observer.

Chapter 6: Conclusion
This chapter summarizes the work and results presented in this thesis.

Chapter 7: Perspectives

In this chapter is a description of some ideas and work that could be done in the future.

Chapter 2

System Description

This chapter introduces the reader to different components to get the reader familiar with some of the terms used during the thesis for a common understanding of the general system the thesis deals with. First is a description of a wind turbine and its most important components and an introduction to the LIDAR technology. This is followed by a control strategy section describing the general concept in controlling a wind turbine and a section introducing the simulation tool LACflex. This chapter describes in general the system that is worked on during the thesis.

2.1 Turbine Description

This section describes the horizontal-axis wind turbine and its components, which are referred to during the thesis. A wind turbine is a mechanical device that extracts the energy in the wind and converts it to mechanical energy, which it converts into electrical power [5, p.12]. Figure 2.1 is an illustration of a three bladed horizontal-axis wind turbine and its most important components. The different components are described based on [25] [28] and [33].

- **Tower** lifts the nacelle and the rotor (hub and rotor blades) up from the ground as the wind speed increases with height above ground, which enables higher turbines to produce more energy. The height of the tower also enables the rotor blades to rotate without touching the ground.
- **Rotor blades** and **hub** combined is called the rotor. The rotor blades capture the wind, which due to the shape of the blades generates a lift force, that makes the blade rotate. The blades are connected to the low-speed shaft in the hub, which makes the low-speed shaft rotate with the same speed as the rotor. In the hub is also the pitch mechanism, which can change the angle of the blades into or away from the wind to get the rotor to rotate faster or slower. The illustrated turbine has three blades which is the normal number of blades and is also the number of blades during this project.
- **Anemometer** measures wind speed and transmits it to the controller.

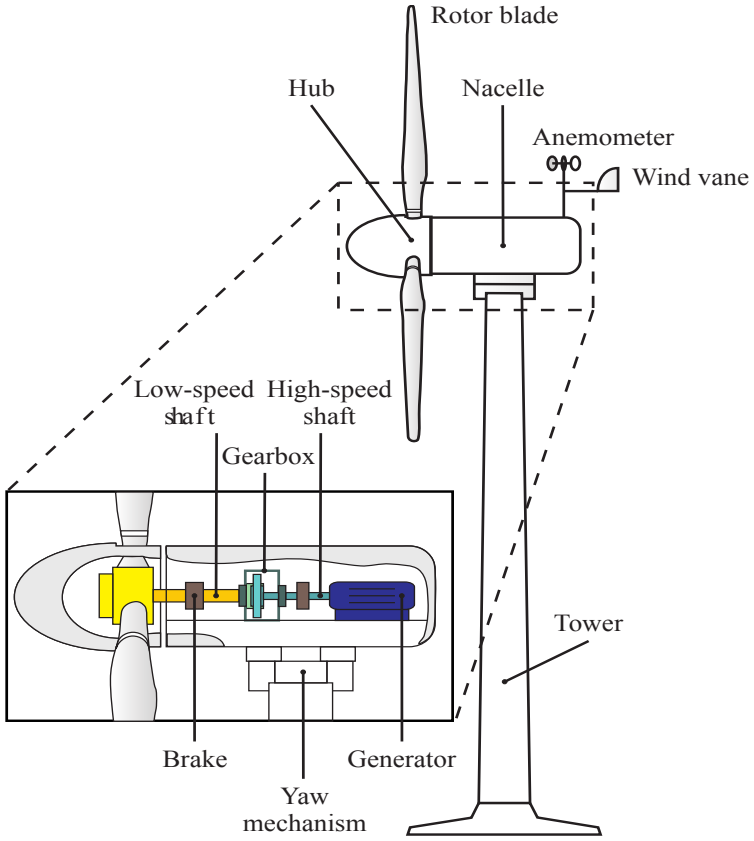


Figure 2.1: *Illustration of a wind turbine [25]*

- **Wind vane** and **Yaw mechanism** is used to orient the turbine up against the wind. The wind vane determines the wind direction and the yaw mechanism turns the nacelle until it is orientated such that the rotor disc is perpendicular to the wind.
- **Nacelle** is the house on top of the tower that protects the mechanical parts; low-speed shaft, gear, high-speed shaft, generator and brakes. It is also the nacelle that the rotor is attached to.
- **Low-speed shaft** transfers the rotational energy from the rotor to the gear box, the low-speed shaft rotates with the same speed as the rotor.
- **Gearbox** increases the rotational speed of the low-speed shaft to get a higher rotational speed at the generator.
- **High-speed shaft** is the shaft from the gearbox to the generator.
- **Generator** uses the rotational energy from the high-speed shaft to generate electricity.
- **Brake** is used to stop the rotor if a fail occurs. The pitch is used as the normal brake if the system has to shut down due to too high wind speeds.

2.2 LIDAR

Traditionally the turbine estimates the average rotor wind speed from the measured rotor speed [6], even though there often is an anemometer mounted on top of the nacelle. The anemometer is not relied on, as it is placed in the wake of the rotor disc.

In this project the idea is to use light detection and ranging (LIDAR) for measuring the wind speed. The advantage of using LIDAR, is that the LIDAR can measure the wind speed in front of the rotor disc, before it affects the turbine, where the other sensors estimates the wind that already has affected the turbine.

An illustration of a LIDAR measuring in a cone is seen in Figure 2.2 and else is the LIDAR is further described in section 3.11.

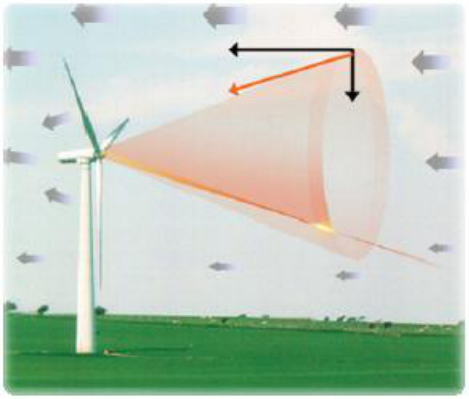


Figure 2.2: *The scan of the LIDAR when mounted on the turbine hub [27].*

2.3 General Control Strategy

This section is an analysis of the general concept in controlling a wind turbine. To be able to develop a turbine controller, it is necessary to determine what the control goal is. The main goal of a turbine is to produce power and so it is for the controller. This section describes how to control a turbine to maximize the power output.

The block diagram in Figure 2.3, illustrates a model of the general method for controlling the power production of a wind turbine. The controller is split into two controllers, a torque controller and a pitch controller.

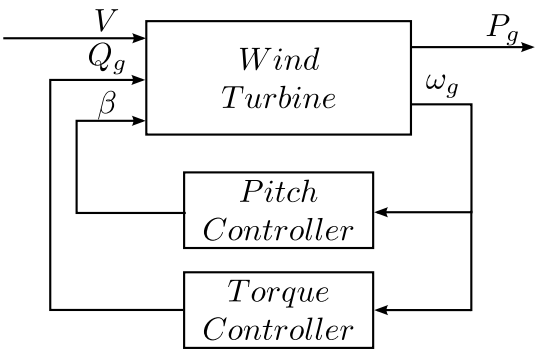


Figure 2.3: *Block diagram of the general controller setup on a turbine, where two controllers regulate the generator torque Q_g and the pitch angle β so the turbine produces the maximum possible amount of power P_g at different wind speeds V.*

The amount of power generated by a turbine is determined by generator torque Q_g and the rotational speed of the generator ω_g. Controlling the generator speed by changing the generator torque, the generator produces the maximum amount of power. This is the fundamental principal in the torque controller.

As the wind speed V increases, the torque controller will increase the load on the generator to maintain the maximum amount of power produced, until the generator load and the rotor speed will reach the design limits for the turbine. When the wind speed increases to the rated level for the turbine, the pitch controller takes over for the torque controller. The pitch controller regulates the pitch angle β to extract less energy from the wind, so the rotational speed is held at the rated value.

The explained controller setup is described in more detail in Section 4.3. From the description of the control goal given above, the values of the in- and output for the turbine can be illustrates as in Figure 2.4, where they are shown relative to different wind speeds.

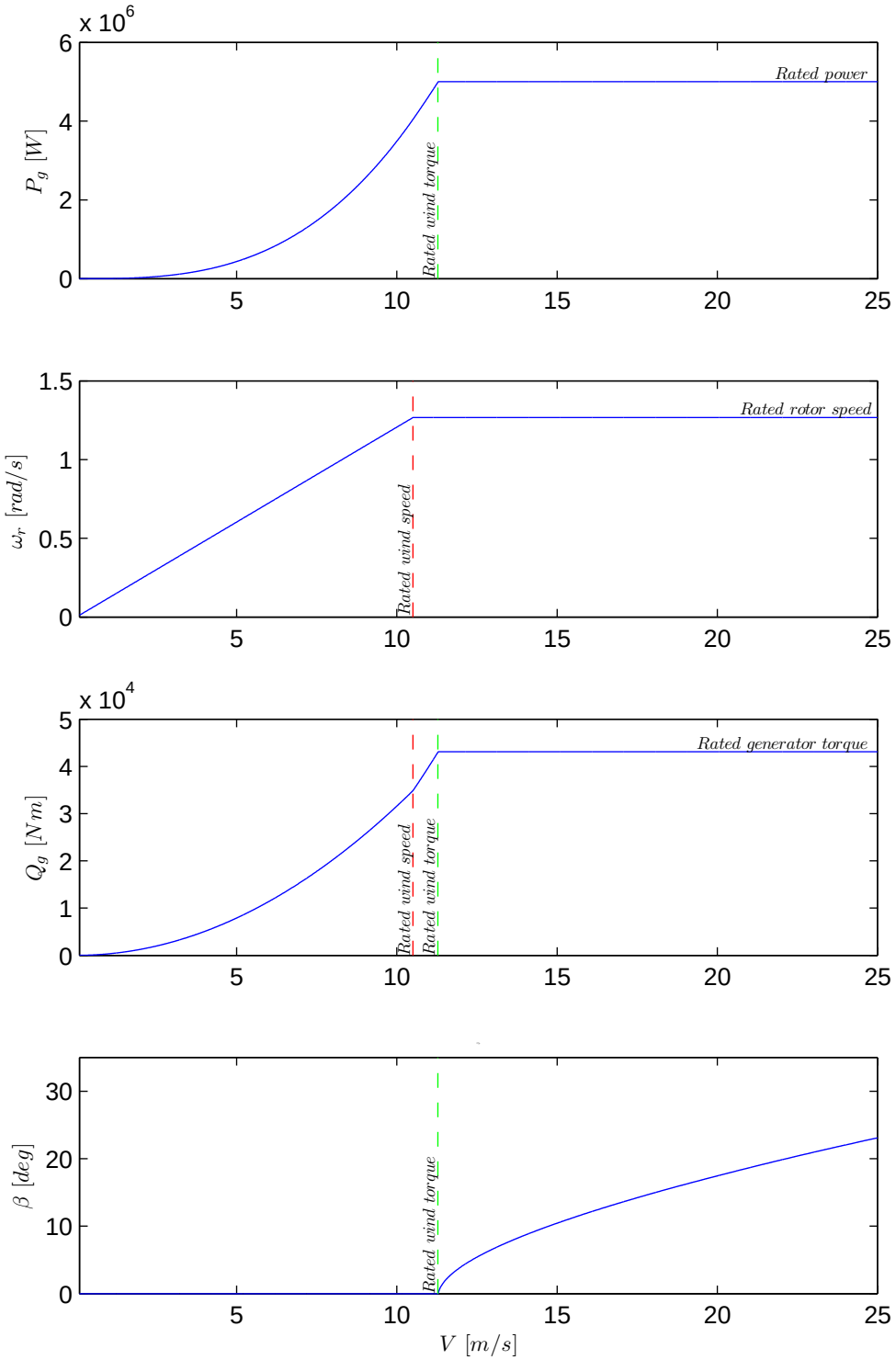


Figure 2.4: *Plot of the generator power P_g, rotor speed ω_r, generator torque Q_g and pitch β as functions of the wind. The values are calculated from the parameters for the NREL 5MW turbine.*

From the controller setup described above, the operation of the turbine can be divided into two regions, below rated wind speed and above rated wind speed. In below rated wind speeds, the

turbine has to extract as much energy as possible from the wind while above the turbine has to be maintained within its design limits.

The first region, below the rated wind speed, seen on the left side of the plots, the torque controller is regulating the generator torque Q_g, illustrated on the third plot from the top. This to keep the rotational speed proportional to the wind speed, seen on the second plot from the top, so the power produced by the generator, seen on the top plot, is at the maximum potential power at the given wind speed.The second region is when the wind speed is above the rated wind speed, seen on the right side of the plots, where the power, rotational speed and torque is held at the rated values for the turbine design. In the second region the pitch controller increases the pitch angle β to reduce the percentage of energy extracted from the wind, so the other values does not increase but stays at the rated levels.

A more detailed description of the turbines behavior at different wind speeds and changes in control input can be found in the modeling chapter, Chapter 3.

If the turbine is set to follow this strategy, the turbine will extract the maximum possible power at wind speeds below rated wind speed and maintain the power at the rated level at high wind speeds.

2.4 LACflex

LACflex is a high-fidelity aeroelastic simulation software developed and provided by LAC Engineering which proposed this project. Wind turbines in closed-loop can be simulated under different wind regimes. Therefore, these aeroelastic simulations can be used for accessing the performance of developed controllers before testing on a real wind turbine. LACflex is based on FLEX5[13] but with significant modifications and one advantage with LACflex is the user friendly GUI interface.

The turbine model in LACflex is more realistic, than the model developed in this thesis, as it has several degrees of freedom that are not included in the developed model. The difference in complexity is described further in section 5.2.

In LACflex tehre is a wind model suitable for generating 3D wind fields for simulations. The wind is generated according to different parameters such as mean wind speed and turbulence intensity.

Included in LACflex is the opportunity to do post processing on the simulated data. One opportunity for post processing is rainflow count, that calculates the number of cycles at different loads and also calculates the damage equivalent load.

Chapter 3

System Modeling

The controllers designed in this thesis are model based controllers. To be able to design the controllers and perform basic simulations, a model of the turbine is necessary. The derived model is similar to already known models, that are commonly used in various literature. In this chapter is a description of a wind model and the used wind generator, the derivation of a linear and nonlinear model of the turbine and a LIDAR model.

3.1 Model Approach

A plant model is derived for the turbine. All modeled parts of the plant are derived from an understanding of the physical laws that are governing in and around a wind turbine. Contrary to using system identification to determine the model, this method gives a better understanding of the working principles of the turbine as are assumed to help in the design and analysis of the controller.

Modeling the turbine from the physical behavior gives a non linear plant model, which can be used to run simulations of the turbine. For use in the design of a controller, a linearized model is needed as the considered controllers are linear. The model is linearized in an operating point, for simplicity only one operating point is used in this thesis.

The goal in this thesis is to try to reduce fatigue loads on the turbine. The model will reflect this in the sense, that model outputs can be used to calculate the fatigues. The primary consideration in this thesis is the fatigue on tower and drive train. These can be estimated by the movement or bending of mechanical structures, therefore these are accepted as outputs of the model. As the model is simplified, the calculated fatigues will not be an accurate estimation of the real fatigues, but for the purpose of comparing different controllers and situations the calculated fatigues are considered appropriate for a preliminary comparison, to indicate the different performances.

To test the derived model, it is simulated with a series of steps on the wind, pitch and generator torque inputs. The model is validated by analyzing the results of these simulations, to see if the model reacts to the inputs as expected according to the understanding of the physics of the turbine. To verify the values of the results, the derived model is compared to another model of

the same turbine [15].

3.2 Model Structure

The plant model is derived as several submodels that are derived separately and combined into one model afterwards. The plant model is split into Pitch and Torque actuators, Aerodynamic, Drive Train, Generator and Tower. The wind is modeled outside the plant and seen as a disturbance. A block diagram of the turbine divided into submodels and the wind is shown in Figure 3.1.

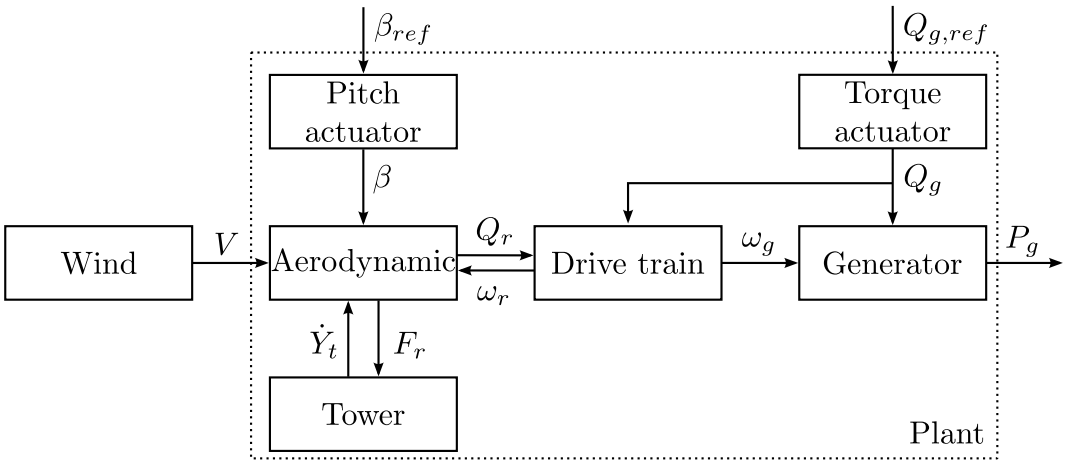


Figure 3.1: *Block diagram showing a submodel of the wind turbine and wind with in- and output signals to the separate blocks. The inputs to the system are the controllable inputs pitch* β *and generator torque* Q_g*. The output is the generator power* P_g*.*

In the derivation of the models, some simplifications are made. The simplifications are listed here as well as a short description of the different blocks.

- **Wind**
  A wind model is described in section 3.3 but is not modeled. For simulations a wind generator from LACflex is used to generate a turbulent wind field for simulations in LACflex. The wind field is for simulations in MATLAB averaged over the rotor plane to provide an average rotor wind speed for the developed nonlinear turbine model.

- **Aerodynamic**
  The aerodynamic part of the turbine is modeled according to the Blade Element Momentum (BEM) theory. It is done with the assumption of rigid blades and a collective pitch control, so all blades have the same pitch angle. This is described in section 3.4

- **Drive Train**
  The drive train is modeled as a rotating mechanical system consisting of a gear and a flexible shaft on the rotor side of the gear. The model takes the low rotor speed and high rotor torque at the rotor side and converts it to the high generator speed and low generator torque for the generator. This is described in section 3.5

- **Generator**
  The generator model describes the amount of power generated. The generator gives the torque load for the drive train. This is described in section 3.6
- **Tower**
  The tower model describes how the tower moves back and forth when the tower is affected by the wind. The tower is modeled as a single mass affected by a force from the rotor and the stiffness and damping from the tower. This is described in section 3.7
- **Pitch and torque actuators**
  The two actuators in the plant are modeled as simple 1.order systems. This is described in section 3.8

3.3 Wind Model

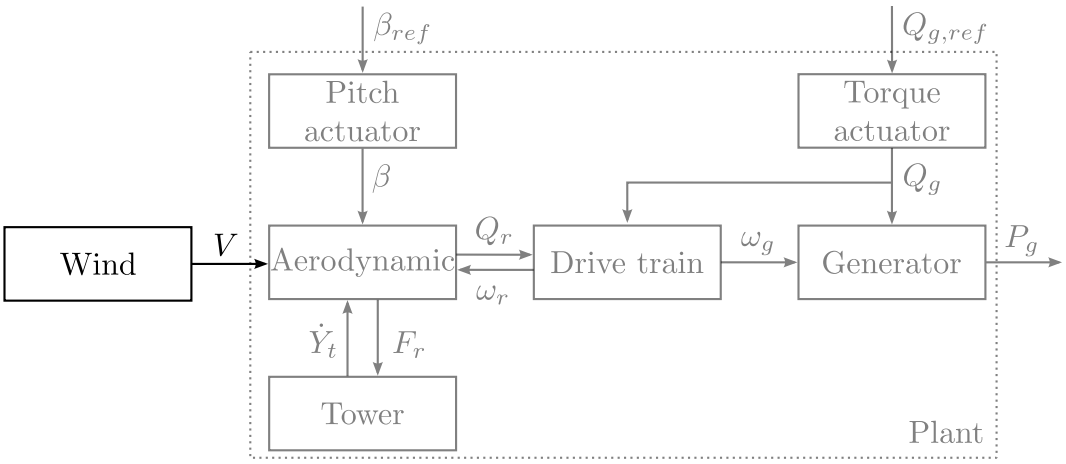


Figure 3.2: *Block diagram showing the plant model of the wind turbine, the highlighted sub-model is the wind model with in- and output signals as are described in the section.*

In this thesis a model of the wind is not derived as the simulation tool LACflex is used to generate a wind that is used for simulation purposes. Instead only the different phenomenas in the wind is explained to get an understanding of what wind is.

The wind is not a simple thing to model, as the wind varies over time and space. The location of the turbine on the globe and the more local geographical properties can have a large influence on the wind at the turbine as well as the time of year and day due to some periodical patterns or the effect of weather systems [32]. In this thesis these parts of the wind is omitted, as they change slow relative to the controller and in this way can be seen as only affecting the mean wind speed. Only fast changes in the wind speed is used.

A block diagram of a wind model [5, p. 45-48] is seen in Figure 3.3.

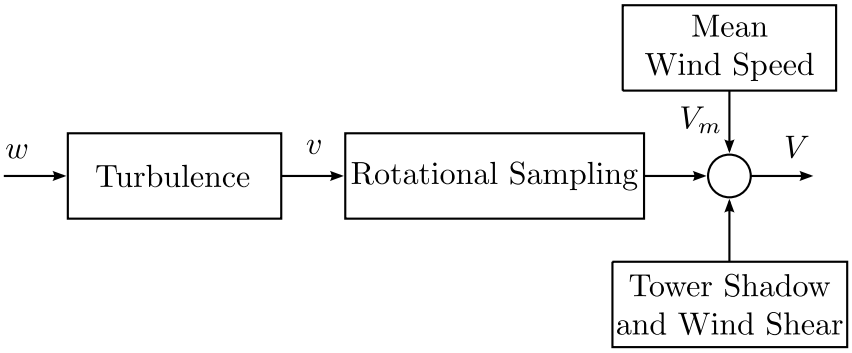


Figure 3.3: *Block diagram of a wind model.*

The wind model consists of five parts: Mean Wind Speed, Turbulence, Tower Shadow, Wind Shear and Rotational Sampling.

The Tower Shadow models how the tower influences the wind speed and thereby the blades when they pass the tower.

The Wind Shear represents that due to friction between the ground and the air, the mean wind is higher above ground and therefore higher for blades pointing upwards than blades pointing downwards.

The Turbulence is the fast changes in the wind, and are mainly coursed by topographical features such as hills and mountains and temperature variations which affects the wind. For modeling purposes, turbulence can be seen as a stochastic process. The stochastic process is driven by white noise which is filtered through a filter ex. a Kaimal spectrum to give the turbulence.

The Rotational Sampling model consists of two filters. The first one is a low pass filter, this to give a fictitious wind speed, which is a scalar wind speed for the entire rotor plane. The second one is an extra filter, that incorporates the rotation of the blades by amplifying the blades rotational frequency.

When these five parts are put together, they give the average rotor wind speed as the output.

Compared to the described model that generates an average rotor wind speed, the wind generator in LACflex generates a field of turbulence and the wind shear and mean wind is added to each point in the field. This wind field is used for simulations in LACflex, but for simulations in MATLAB on the devolved nonlinear model the wind field averaged over the rotor plane is used.

3.4 Aerodynamic Model

The aerodynamic part of the model describes how the wind interacts with the turbine. Figure 3.4 illustrates the input and output of the model and which other parts of the model the aerodynamic directly interacts with.

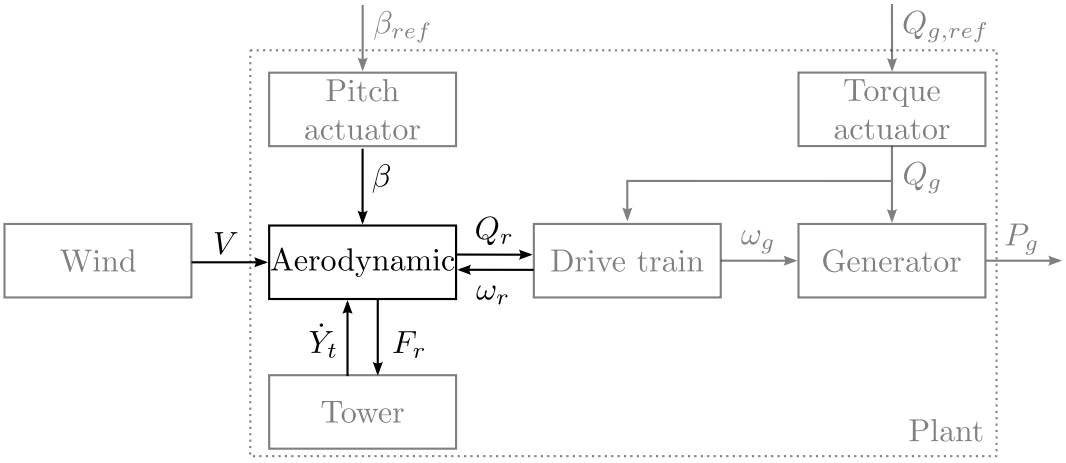


Figure 3.4: *Block diagram showing the plant model of the wind turbine, the highlighted sub-model is the aerodynamic part with in- and output signals as are described in the section.*

The wind input V, as described in Section 3.3, is an average wind speed over the entire rotor disc. It consists of a mean wind speed with turbulence.

The pitch angle β, is the angle of the blades. In this model a collective pitch is used so all three blades have the same pitch angle. The pitch system is decsribed in section 3.8.

The tower fore-aft velocity $\dot{Y}_t$, is described in section 3.7. In the aerodynamic model the tower velocity affects the wind speed seen by the rotor disc. When the tower sways into the wind, the wind speed experienced by the rotor is higher than the real wind speed and lower when the tower sways with the wind.

The thrust force F_r, from the aerodynamic to the tower is the force generated from the wind hitting the rotor disc and pushing it backwards and thereby pushing the tower backwards.

The rotational speed of the rotor ω_r, is given by the drive train and affects the aerodynamic as the amount of energy extracted from the wind depends on the ratio between the rotor speed and the wind speed, called the tip-speed-ratio, as will be described later in this section.

The rotor torque Q_r, is the torque applied from the rotor to the drive train and indirectly to the generator.

3.4.1 The Rotor Disc

An illustration of the turbine is shown in Figure 3.5. The rotor disc on the turbine is the disc that the blades forms when they rotates, in the figure the different parameters of the rotor disc is marked.

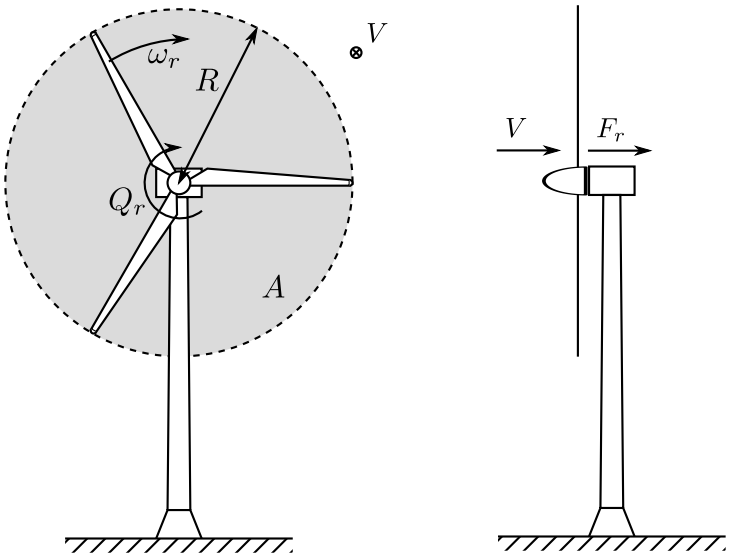


Figure 3.5: *Illustration of the turbine with parameters. ω_r is the rotor speed, R is the rotor disc radius, A is the area of the rotor disc, V is the average wind speed over the rotor disc, Q_r is the rotor torque and F_r is the force both generated by the wind on the rotor disc.*

3.4.2 Energy and Force in the Wind

Before the aerodynamic model is described in details, the properties of the wind are defined. In Figure 3.6 a wind is illustrated as a tube of air flowing in the direction of the wind as it would look in front of a turbine, before the wind is affected by the turbine, or if there is no turbine to affect it.

Figure 3.6: *The air flow seen as a uniform tube when there is no turbine to affect the wind.*

The wind speed V can be seen as an airflow with the mass flow $\dot{m}_V$ given by

$$\dot{m}_V = \rho A V, \tag{3.1}$$

where ρ is the density of the air, A is the cross section area, in this case the same area that is swept by the rotor. From fluid dynamics the force in the wind F_v, can be found as

$$F_V = \frac{1}{2}\dot{m}_V V = \frac{1}{2}\rho A V^2 \tag{3.2}$$

and works in the same direction as the wind. The last property of the wind that is defined is the

amount of kinetic energy in the wind P_V, that is given as follows

$$P_V = F_V V = \frac{1}{2}\rho A V^3. \tag{3.3}$$

The force and kinetic energy in the wind are the two properties of the wind, that are used in modeling of the aerodynamic of the turbine.

3.4.3 The Turbine in the Wind

When a turbine is placed in the wind it extracts energy from the wind and converts the energy to mechanical energy. However, the turbine can not extract all the energy in the wind, only

$$P_r = P_V C_p \tag{3.4}$$

where C_p is the power coefficient. The power coefficient can be derived from the actuator disc theory described in [32]. This explains how the extracted energy comes from a drop in the wind speed.

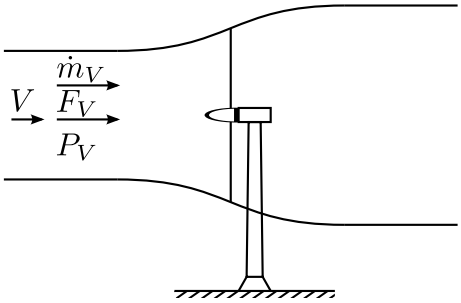


Figure 3.7: *The turbine placed in the air flow.*

The power coefficient C_p is mostly explained by two things. The first thing is when the wind speed drops, as it passes through the rotor disc, but the mass flow stays the same on both sides of the rotor disc. It is seen from equation (3.1), that the cross section area A has to grow as the wind speed drops, which introduces the notion that the air flow tube is not uniform, as illustrated in Figure 3.7. It is shown that the cross section where the wind is at free speed is smaller then the rotor area, which explains most of C_p, this also gives a maximum achievable value of 0.593 for C_p, known as the Betz Limit. The last thing that affects C_p is the aerodynamics of the rotor disc.

The outputs of the model Q_r and F_r, can be determined by the theory described earlier in this section. For a simpler notation the tip-speed-ratio λ, is introduced as

$$\lambda = \frac{\omega_r R}{V}, \tag{3.5}$$

which is a measure for the ratio between the rotor rotational speed ω_r and the wind speed V.

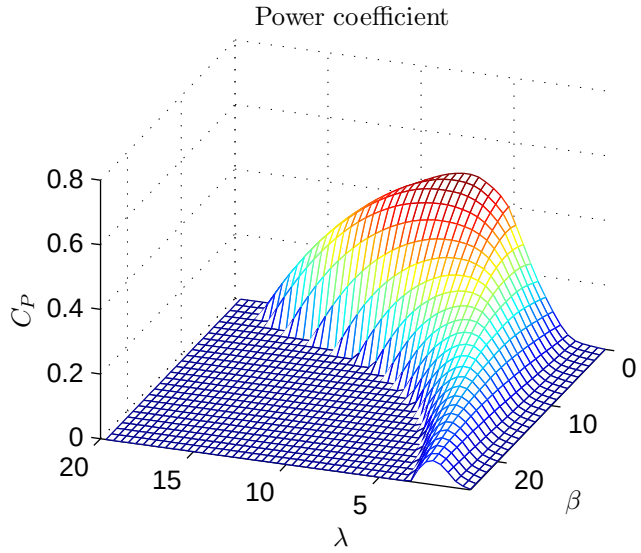


Figure 3.8: *A plot illustrating the nonlinear properties of the power coefficient where all negative values are set to zero.*

Figure 3.8 illustrates the power coefficient at different tip-speed-ratios and pitch angles, which shows that the aerodynamic properties of the turbine is nonlinear. The power coefficient on the figure is for the NREL 5MW turbine and the maximum value of C_p is found to be 0.4851, with a pitch of 0 *deg* and a tip-speed-ratio of 7.55, which is under the theoretical Betz limit. Furthermore the power coefficient C_p, is converted into a torque coefficient C_Q defined as

$$C_Q = \frac{C_P}{\lambda}. \tag{3.6}$$

The extracted power can then be expressed by inserting the equations (3.3), (3.5) and (3.6) into equation (3.4) and replacing A with πR^2 leading to the rotational energy expressed as

$$P_r = \frac{1}{2}\rho\pi R^3 V^2 \omega_r C_Q. \tag{3.7}$$

The output from the model should be the torque generated by the rotors aerodynamic, so the extracted energy is converted to torque as follows

$$Q_r = \frac{1}{\omega_r} P_r = \frac{1}{2}\rho\pi R^3 V^2 C_Q. \tag{3.8}$$

Like for the torque, the thrust force coefficient C_T, is a measure of how much of the wind force that is applied to the rotor. The thrust force applied to the rotor F_r, is given as

$$F_r = F_V C_T \tag{3.9}$$

and the force can then be expressed by inserting equation (3.2) into (3.9) which leads to the thrust force to be given as

$$F_r = \frac{1}{2}\rho A V^2 C_T. \tag{3.10}$$

The two coefficients torque and thrust, are determined by Blade Element Momentum Theory (BEM) and give nonlinear curves from the pitch β and the tip-speed-ratio λ. Furthermore the tower movement, and thus the rotor movement affects the wind seen by the rotor, as when the tower bends backwards the rotor will move with the wind and the rotor will see a lower wind speed. To include the tower movement in the aerodynamic model the wind in the model is replaced with

$$V_r = V - \dot{Y}_t, \tag{3.11}$$

where V_r is the relative wind speed seen by the rotor and $\dot{Y}_t$ is the tower speed fore-aft. This leads to the final outputs of the model which are given as

$$Q_r = \frac{1}{2}\rho\pi R^3 V_r^2 C_Q(\lambda,\beta) \tag{3.12}$$

$$F_r = \frac{1}{2}\rho A V_r^2 C_T(\lambda,\beta). \tag{3.13}$$

The found model is non-linear and for use in the design of a controller it has to be linearized.

3.4.4 Linearization of the Aerodynamic Model

The aerodynamic model, described in subsection 3.4.3 is nonlinear. The nonlinearity comes from the properties of the torque and thrust force coefficients, as are illustrated in Figure 3.9.

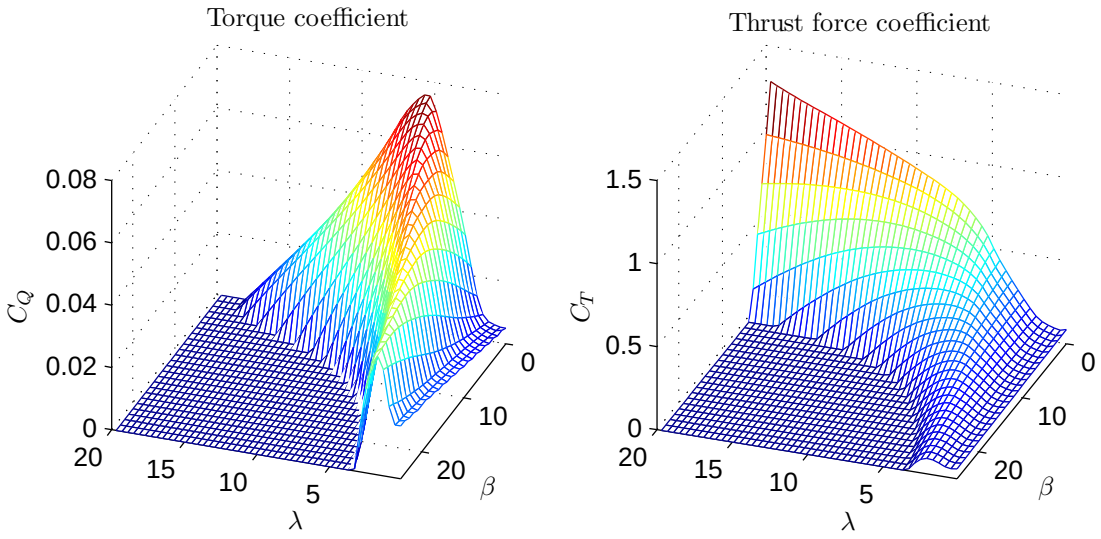


Figure 3.9: *The torque and thrust force coefficients are nonlinear parameters of the turbine model, C_Q is based on C_p, and C_p and C_t is given by the aerodynamic properties of the blades.*

Torque and thrust forces have to be linearized in order to be utilized in the design of linear controllers. The linearized model is found by a first order Taylor series expansion at an operating point. The operating point is determined from values of λ and β, which are denoted as λ_0 and β_0. λ is however dependent of ω_r and V_r therefore λ_0 is given by $\omega_{r,0}$ and $V_{r,0}$. The rotor torque at the operation point is given by

$$Q_{r,0} = \frac{1}{2}\rho\pi R^3 V_{r,0}^2 C_Q(\lambda_0, \beta_0). \tag{3.14}$$

A Taylor series expansion gives the small signal for compensating when the model moves away from the operating point. The difference between the real values and the operating point is denoted as Δ, where $\beta_\Delta = \beta - \beta_0$, leading to the rotor torque given by

$$Q_r \cong Q_{r,0} + \left.\frac{\partial Q_r}{\partial \omega_r}\right|_{\omega_{r,0}} \omega_{r,\Delta} + \left.\frac{\partial Q_r}{\partial V_r}\right|_{V_{r,0}} V_{r,\Delta} + \left.\frac{\partial Q_r}{\partial \beta}\right|_{\beta_0} \beta_\Delta. \tag{3.15}$$

Writing the small signals as a state space model gives

$$Q_{r\Delta} \cong \begin{bmatrix} \left.\frac{\partial Q_r}{\partial \omega_r}\right|_{\omega_{r,0}} & \left.\frac{\partial Q_r}{\partial V_r}\right|_{V_{r,0}} & \left.\frac{\partial Q_r}{\partial \beta}\right|_{\beta_0} \end{bmatrix} \begin{bmatrix} \omega_{r,\Delta} \\ V_{r,\Delta} \\ \beta_\Delta \end{bmatrix} \tag{3.16}$$

where the individual parts are derived as

$$\left.\frac{\partial Q_r}{\partial \omega_r}\right|_{\omega_{r,0}} = \frac{1}{2}\rho\pi R^3 V_{r,0}^2 \left.\frac{\partial C_Q}{\partial \lambda}\right|_{\lambda_0} \left.\frac{\partial \lambda}{\partial \omega_r}\right|_{\omega_{r,0}} \tag{3.17}$$

$$\left.\frac{\partial Q_r}{\partial V_r}\right|_{V_{r,0}} = \frac{1}{2}\rho\pi R^3 \left(2V_{r,0} C_{Q,0} + V_{r,0}^2 \left.\frac{\partial C_Q}{\partial \lambda}\right|_{\lambda_0} \left.\frac{\partial \lambda}{\partial V_r}\right|_{V_{r,0}} \right) \tag{3.18}$$

$$\left.\frac{\partial Q_r}{\partial \beta}\right|_{\beta_0} = \frac{1}{2}\rho\pi R^3 V_{r,0}^2 \left.\frac{\partial C_Q}{\partial \beta}\right|_{\beta_0} \tag{3.19}$$

where

$$\left.\frac{\partial \lambda}{\partial \omega_r}\right|_{\omega_{r,0}} = \frac{R}{V_{r,0}} \tag{3.20}$$

$$\left.\frac{\partial \lambda}{\partial V_r}\right|_{V_{r,0}} = -\frac{\omega_{r,0} R}{V_{r,0}^2}, \tag{3.21}$$

which can be rewritten into

$$\left.\frac{\partial Q_r}{\partial \omega_r}\right|_{\omega_{r,0}} = \frac{1}{2}\rho\pi R^4 V_{r,0} \left.\frac{\partial C_Q}{\partial \lambda}\right|_{\lambda_0} \tag{3.22}$$

$$\left.\frac{\partial Q_r}{\partial V_r}\right|_{V_{r,0}} = \rho\pi R^3 V_{r,0} C_{Q,0} - \frac{1}{2}\rho\pi R^4 \omega_{r,0} \left.\frac{\partial C_Q}{\partial \lambda}\right|_{\lambda_0} \tag{3.23}$$

$$\left.\frac{\partial Q_r}{\partial \beta}\right|_{\beta_0} = \frac{1}{2}\rho\pi R^3 V_{r,0}^2 \left.\frac{\partial C_Q}{\partial \beta}\right|_{\beta_0} \tag{3.24}$$

and inserted into the state space notation, equation (3.16), for the operating points.

A linearization is also performed for the thrust force part of the aerodynamic model. Dividing it to a operating point and small signal values gives

$$F_r \cong F_{r,0} + \left.\frac{\partial F_r}{\partial \omega_r}\right|_{\omega_{r,0}} \omega_{r,\Delta} + \left.\frac{\partial F_r}{\partial V_r}\right|_{V_{r,0}} V_{r,\Delta} + \left.\frac{\partial F_r}{\partial \beta}\right|_{\beta_0} \beta_\Delta \tag{3.25}$$

and writing this as a state space model gives

$$F_{r\Delta} \cong \begin{bmatrix} \left.\frac{\partial F_r}{\partial \omega_r}\right|_{\omega_{r,0}} & \left.\frac{\partial F_r}{\partial V_r}\right|_{V_{r,0}} & \left.\frac{\partial F_r}{\partial \beta}\right|_{\beta_0} \end{bmatrix} \begin{bmatrix} \omega_{r,\Delta} \\ V_{r,\Delta} \\ \beta_\Delta \end{bmatrix} \tag{3.26}$$

where the individual parts are

$$\left.\frac{\partial F_r}{\partial \omega_r}\right|_{\omega_{r,0}} = \frac{1}{2}\rho\pi R^3 V_{r,0} \left.\frac{\partial C_T}{\partial \lambda}\right|_{\lambda_0} \tag{3.27}$$

$$\left.\frac{\partial F_r}{\partial V_r}\right|_{V_{r,0}} = \rho\pi R^2 V_{r,0} C_{T,0} - \frac{1}{2}\rho\pi R^3 \omega_{r,0} \left.\frac{\partial C_T}{\partial \lambda}\right|_{\lambda_0} \tag{3.28}$$

$$\left.\frac{\partial F_r}{\partial \beta}\right|_{\beta_0} = \frac{1}{2}\rho\pi R^2 V_{r,0}^2 \left.\frac{\partial C_T}{\partial \beta}\right|_{\beta_0} \tag{3.29}$$

3.5 Drive Train Model

The drive train transfers the rotor torque produced by the wind hitting the rotor to the generator, where it is converted to electrical power. This section describes the model for the drive train.

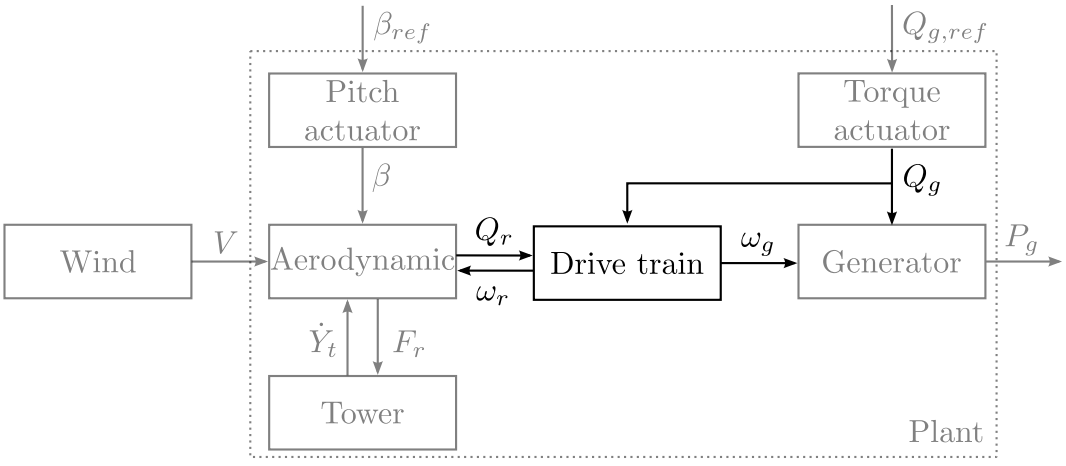


Figure 3.10: *The drive train model with in- and outputs.*

As input the drive train model takes the torques from the rotor and generator Q_r and Q_g. As output the model has the rotor speed ω_r and the generator speed ω_g. See figure 3.10.

The model of the drive train is derived from [1], [5] and [16]. Figure 3.11 illustrates the free body diagram of the drive train.

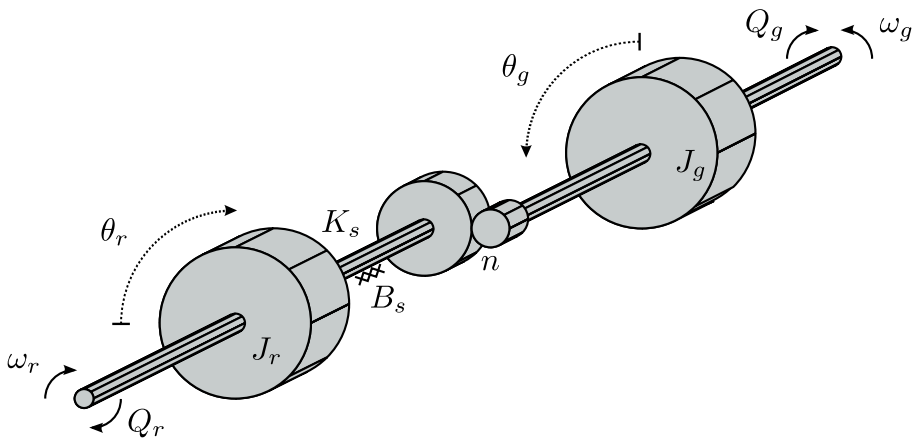


Figure 3.11: *Free Body diagram of the drive train.*

In the drive train model the inertias of the rotor and generator, respectively J_r and J_g are included. Between the two inertias there is a gearing, with the ratio n. The drive shaft between the gear and the generator is assumed to be rigid, but the other side of the gear between the gear and rotor, the drive shaft is assumed flexible represented by torsion stiffness K_s and damping B_s.

To derive the model the two inertias are modeled individually with the forces that are acting on them. The individual torques and the respective directions are illustrated in figure 3.12.

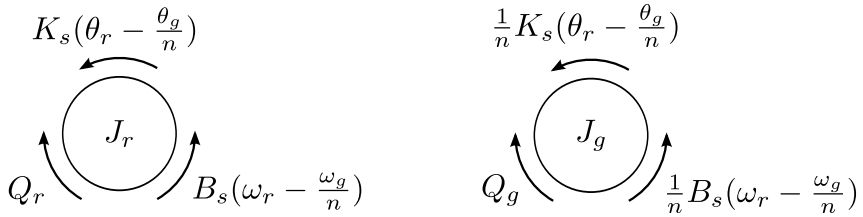


Figure 3.12: *Free Body diagram of the drive train with the individual torques and their directions.*

From analyzing the torques on the rotor inertia the dynamics of the rotor side looks like

$$J_r\dot{\omega}_r = Q_r - K_s\theta_\Delta - B_s\left(\omega_r - \frac{\omega_g}{n}\right), \tag{3.30}$$

where the torque from the aerodynamic Q_r defines the positive direction of the torques and θ_Δ is the torsion of the drive shaft introduced as

$$\theta_\Delta = \left(\theta_r - \frac{\theta_g}{n}\right).$$

There is no need for the absolute angles of the drive trains two sides, only the torsion is of interest. By the notion that the angles are of no need it is introduced that

$$\dot{\theta} = \omega$$

and the change in torsion can be written as

$$\dot{\theta}_\Delta = \left(\omega_r - \frac{\omega_g}{n}\right). \tag{3.31}$$

The same analysis is done for the generator side, where the positive direction is the direction of the generator speed or opposite the generator torque Q_g. This gives the following equation for the generator side

$$J_g\dot{\omega}_g = -Q_g + \frac{1}{n}K_s\theta_\Delta + \frac{1}{n}B_s\left(\omega_r - \frac{\omega_g}{n}\right). \tag{3.32}$$

The derived model for the drive train, equation (3.30), (3.31) and (3.32) are first order differential equations. These can be combined into a linear state space representation of the drive train, which looks like

$$\underbrace{\begin{bmatrix}\dot{\omega}_r\\ \dot{\omega}_g\\ \dot{\theta}_\Delta\end{bmatrix}}_{\dot{x}_{DT}} = \underbrace{\begin{bmatrix}-\frac{B_s}{J_r} & \frac{B_s}{nJ_r} & -\frac{K_s}{J_r}\\ \frac{B_s}{nJ_g} & -\frac{B_s}{n^2J_g} & \frac{K_s}{nJ_g}\\ 1 & -\frac{1}{n} & 0\end{bmatrix}}_{A_{DT}}\underbrace{\begin{bmatrix}\omega_r\\ \omega_g\\ \theta_\Delta\end{bmatrix}}_{x_{DT}} + \underbrace{\begin{bmatrix}\frac{1}{J_r} & 0\\ 0 & -\frac{1}{J_g}\\ 0 & 0\end{bmatrix}}_{B_{DT}}\underbrace{\begin{bmatrix}Q_r\\ Q_g\end{bmatrix}}_{u_{DT}}. \tag{3.33}$$

The found model gives the wanted outputs and it does not include any nonlinearities. The model is used in the combined model described in section 3.9.

3.6 Generator Model

The generator of the turbine converts the mechanical torque from the drive train to electrical power.

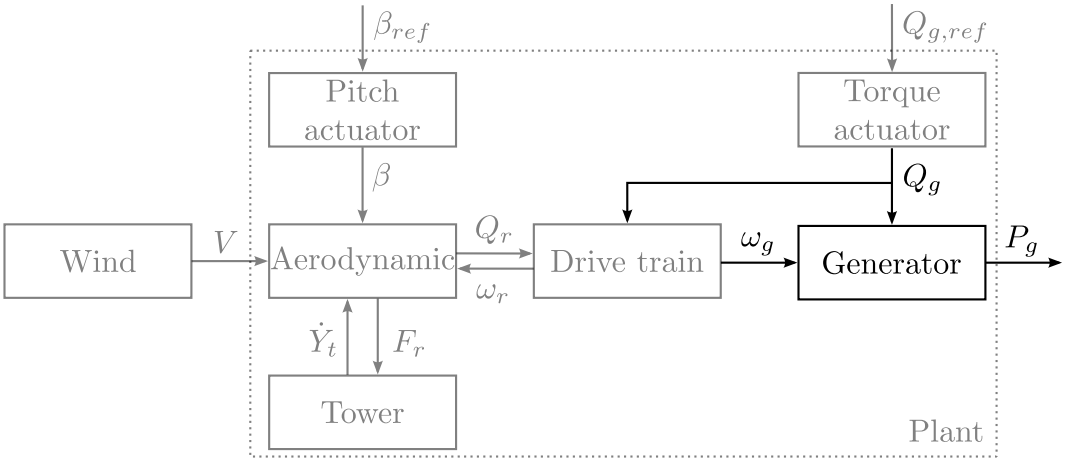


Figure 3.13: *The generator model with in- and outputs to the other submodels.*

The generator model, see Figure 3.13, takes the rotational speed from the drive train ω_g and the applied generator torque Q_g, to give the power output P_g.

The rotational energy is converted to electrical energy by the generator, but some energy is lost

in the conversion. The power generated is given by

$$P_g = \omega_g Q_g \eta_g \tag{3.34}$$

where η_g is the efficiency which is a constant for the generator. The generator torque Q_g is controllable by the electrical load put on the generator.

The power output from the generator is not used directly in the control effort but is included as a output from the model, for use in the analysis of the power production of the turbine in different situations. The generator model is non-linear and is therefore linearized.

3.6.1 Linear Generator Model

The generator model is linearized by the same principle as used for the aerodynamic model in section 3.4.4. The model is linearized around the operating point $P_{g,0}$ and is given by

$$P_g = P_{g,0} + \left.\frac{\partial P_g}{\partial \omega_g}\right|_{\omega_{g,0}} \omega_{g,\Delta} + \left.\frac{\partial P_G}{\partial Q_g}\right|_{Q_{g,0}} Q_{g,\Delta}, \tag{3.35}$$

where

$$\left.\frac{\partial P_g}{\partial \omega_g}\right|_{\omega_{g,0}} = \eta_g Q_{g,0} \tag{3.36}$$

$$\left.\frac{\partial P_G}{\partial Q_g}\right|_{Q_{g,0}} = \eta_g \omega_{g,0} \tag{3.37}$$

3.7 Tower Model

The tower model describes how the tower moves when the wind affects the rotor, which affects the wind speed seen by the rotor. Figure 3.14 illustrates the in- and outputs to the tower model from the other submodels. The aerodynamic provides the only input to the tower model, the thrust force F_r. The outputs from the tower model are the fore-aft mode displacement and the velocity of the fore-aft mode, which is the variable that affects the windspeed seen by the rotor.

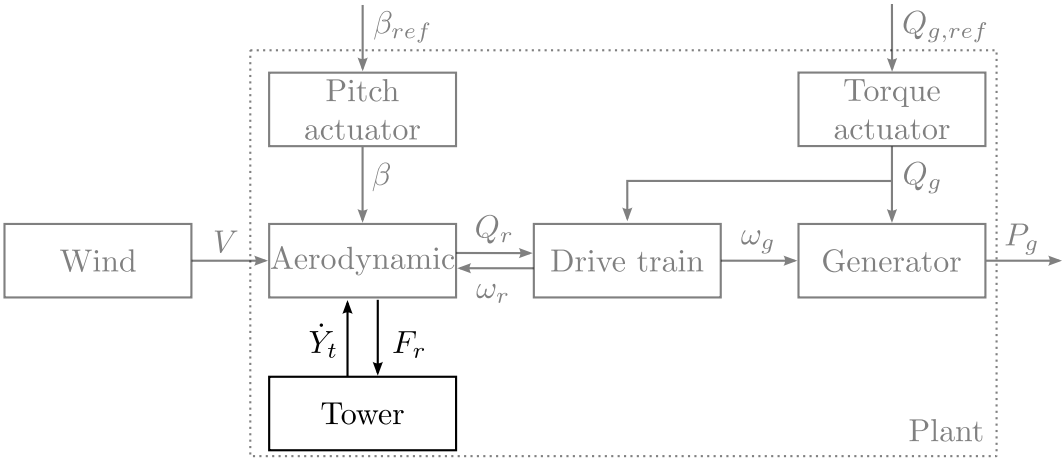


Figure 3.14: *The tower model interacts only with the aerodynamic model. It takes the force generated by the aerodynamic as input and gives the movement of the tower $\dot{Y}_t$, along with the displacement of the tower Y_t.*

As illustrated in Figure 3.15, the wind will push back the turbine with a force F_r. This force makes the nacelle move back and forth, as the tower stiffness K_t delivers a force proportional to the displacement of the tower Y_t in the other direction. Furthermore the tower has a damping effect B_t, which is affected by the rate of displacement or speed of the tower $\dot{Y}_t$.

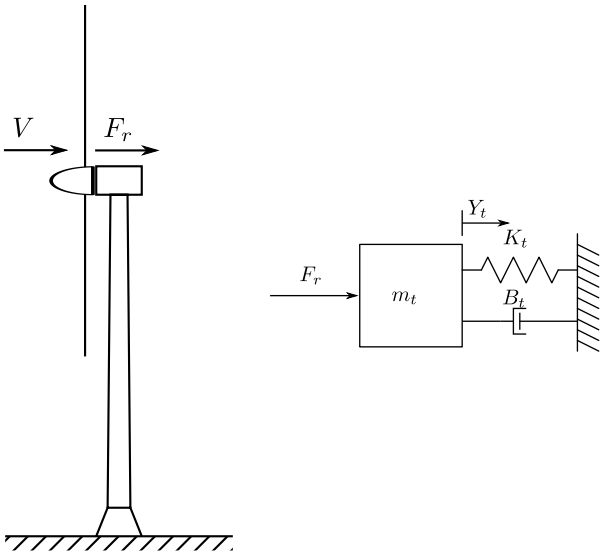


Figure 3.15: *The turbine tower seen as a 1-mass, 1-spring, 1-damper system.*

With the assumptions that the tower can be modeled as an ideal mass and that the nacelle moves back and forth without tilting. The force from the aerodynamic and the force from the tower bending is opposite direction and the model for the tower can be seen as the free body diagram in Figure 3.16.

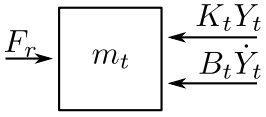


Figure 3.16: *The turbine seen as a free body diagram with the forces acting on the tower.*

From the free body diagram the model for the tower can be setup as a second order differential equation to find the tower displacement Y_t, which looks like

$$m_t \ddot{Y}_t = F_r - K_t Y_t - B_t \dot{Y}_t$$

where the force from the aerodynamic F_r, determines the positive direction and m_t is the modal mass of the tower. Rewriting the differential equation to state space gives the model for the tower as

$$\underbrace{\begin{bmatrix} \dot{Y}_t \\ \ddot{Y}_t \end{bmatrix}}_{\dot{x}_T} = \underbrace{\begin{bmatrix} 0 & 1 \\ \frac{-K_t}{m_t} & \frac{-B_t}{m_t} \end{bmatrix}}_{A_T} \underbrace{\begin{bmatrix} Y_t \\ \dot{Y}_t \end{bmatrix}}_{x_T} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m_t} \end{bmatrix}}_{B_T} \underbrace{F_r}_{u_T}. \tag{3.38}$$

3.8 Pitch and torque actuator models

The control input to the turbine plant model is not applied directly to the plant models subsystems. There is made two actuator models that describes the real turbines behavior.

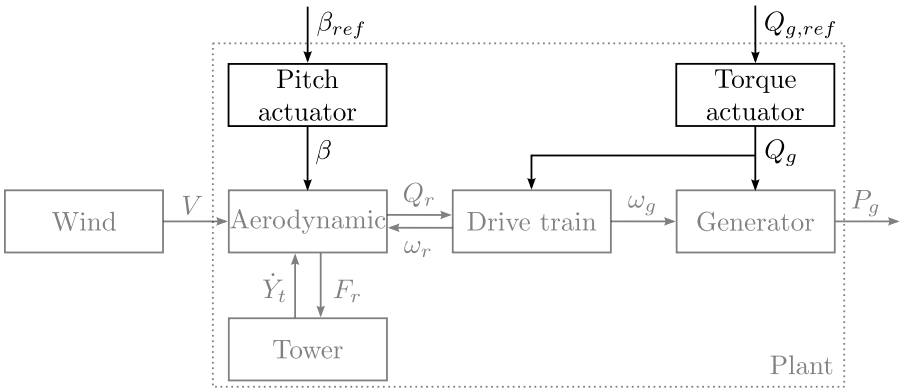


Figure 3.17: *Block diagram showing the plant model of the wind turbine, the highlighted sub-model is the aerodynamic part with in- and output signals as are described in the section.*

In a real turbine the torque will be controlled by the electrical load as are put on the generator and in most cases it is assumed the the electrical system is much faster than the mechanical systems of the turbine so in the thesis it is modeled as a fast 1. order system as in

$$\dot{Q}_g = -\frac{1}{\tau_g} Q_g + \frac{1}{\tau_g} Q_{g,ref} \tag{3.39}$$

where τ_g is the time constant for the generator.

The pitch actuator is relative slow in comparison to the torque actuator. On a real turbine this

actuator is either a hydraulic system or an electrical motor and is subject to some nonlinearities and constraints. In this thesis the actuator is approximated by a 1.order system given as

$$\dot{\beta} = -\frac{1}{\tau_p}\beta + \frac{1}{\tau_p}\beta_{ref} \tag{3.40}$$

where τ_p is the time constant for the pitch system.

From the two differential equation a state space model for the actuators can be written as

$$\underbrace{\begin{bmatrix}\dot{\beta}\\ \dot{Q}_g\end{bmatrix}}_{\dot{x}_a} = \underbrace{\begin{bmatrix}-\frac{1}{\tau_p} & 0\\ 0 & -\frac{1}{\tau_g}\end{bmatrix}}_{A_a}\underbrace{\begin{bmatrix}\beta\\ Q_g\end{bmatrix}}_{x_a} + \underbrace{\begin{bmatrix}\frac{1}{\tau_p} & 0\\ 0 & \frac{1}{\tau_g}\end{bmatrix}}_{B_a}\underbrace{\begin{bmatrix}\beta_{ref}\\ Q_{g,ref}\end{bmatrix}}_{u_a}. \tag{3.41}$$

3.9 Combined Model

The different parts of the turbine model have to be combined into one model, as the models are all represented as state space models, combining them is straight forward. The models from sections 3.4, 3.5, 3.6, 3.7 and 3.8 are combined into

$$\underbrace{\begin{bmatrix}\dot{\omega}_r\\ \dot{\omega}_g\\ \dot{\theta}_\Delta\\ \dot{Y}_t\\ \ddot{Y}_t\\ \dot{\beta}\\ \dot{Q}_g\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}
-\frac{B_s}{J_r} + \frac{1}{J_r}\frac{\partial Q_r}{\partial \omega_r}\Big|_{\omega_{r,0}} & \frac{B_s}{nJ_r} & -\frac{K_s}{J_r} & 0 & -\frac{1}{J_r}\frac{\partial Q_r}{\partial V}\Big|_{V_0} & \frac{1}{J_r}\frac{\partial Q_r}{\partial \beta}\Big|_{\beta_0} & 0\\
\frac{B_s}{nJ_g} & -\frac{B_s}{n^2 J_g} & \frac{K_s}{nJ_g} & 0 & 0 & 0 & -\frac{1}{J_g}\\
1 & -\frac{1}{n} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0\\
\frac{1}{m_t}\frac{\partial F_r}{\partial \omega_r}\Big|_{\omega_{r,0}} & 0 & 0 & \frac{-K_t}{m_t} & \frac{-B_t}{m_t} - \frac{1}{m_t}\frac{\partial F_r}{\partial V}\Big|_{V_0} & \frac{1}{m_t}\frac{\partial F_r}{\partial \beta}\Big|_{\beta_0} & 0\\
0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_p} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_g}
\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}\omega_r\\ \omega_g\\ \theta_\Delta\\ Y_t\\ \dot{Y}_t\\ \beta\\ Q_g\end{bmatrix}}_{x}$$

$$+ \underbrace{\begin{bmatrix}
0 & 0 & \frac{1}{J_r}\frac{\partial Q_r}{\partial V}\Big|_{V_0}\\
0 & 0 & 0\\
0 & 0 & 0\\
0 & 0 & 0\\
0 & 0 & \frac{1}{m_t}\frac{\partial F_r}{\partial V}\Big|_{V_0}\\
\frac{1}{\tau_p} & 0 & 0\\
0 & \frac{1}{\tau_g} & 0
\end{bmatrix}}_{B}\underbrace{\begin{bmatrix}\beta_{ref}\\ Q_{g,ref}\\ V\end{bmatrix}}_{u} \tag{3.42}$$

$$\underbrace{\begin{bmatrix} \omega_r \\ \omega_g \\ \theta_\Delta \\ Y_t \\ \dot{Y}_t \\ \beta \\ Q_g \\ P_g \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \eta_g Q_{g,0} & 0 & 0 & 0 & \eta_g \omega_{g,0} & 0 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} \omega_r \\ \omega_g \\ \theta_\Delta \\ Y_t \\ \dot{Y}_t \\ \beta \\ Q_g \end{bmatrix}}_{x}. \tag{3.43}$$

3.10 Model Verification

Before working further with the model, the model is validated to check if it behaves as expected according to physic. This is done in three test, where a step is provided on the different inputs; wind speed, generator torque and pitch. The tests are made around an operating point of a wind speed of $15 m/s$, generator torque at rated $43.1 \cdot 10^3 Nm$ and a pitch of 10.44 deg. The input that is stepped is changed from ten percent below its operating point value to ten percent above. Table 3.1 shows the different values for wind, generator torque and pitch for the three test.

Test	**Wind** (V)	**Generator torque** (Q_g)	**Pitch** (β)
Wind step	13.5-16.5	$43.1 \cdot 10^3$	10.44
Generator torque step	15	$38.8 \cdot 10^3 - 47.4 \cdot 10^3$	10.44
Pitch step	15	$43.1 \cdot 10^3$	9.4 - 11.48

Table 3.1: *The different tests and the test values for wind, generator torque and pitch.*

The simulation time for the test is 500 seconds and the step is provided at 200 seconds. Data is plotted for the step input either wind, generator torque or pitch together with rotor speed ω_r, rotor torque Q_r, power coefficient C_p and the generator power P_g. These data are chosen as it is assumed that these can show if the turbine behaves as expected according to physics.

Furthermore a second test is performed, as the first test does not validate if the dynamics and steady state values are correct. To do this the model is held up against another NREL model [15]. The test description is the same as for the first test where a step in the wind is provided see wind step in Table 3.1. In this second test also the linear model is tested to see if it fits properly. The linear model is calculated in the operating point.

3.10.1 Test 1.1 wind speed step

The first test simulates what happens with a step in the wind speed. The simulation result for the step in the wind speed is shown in Figure 3.18.

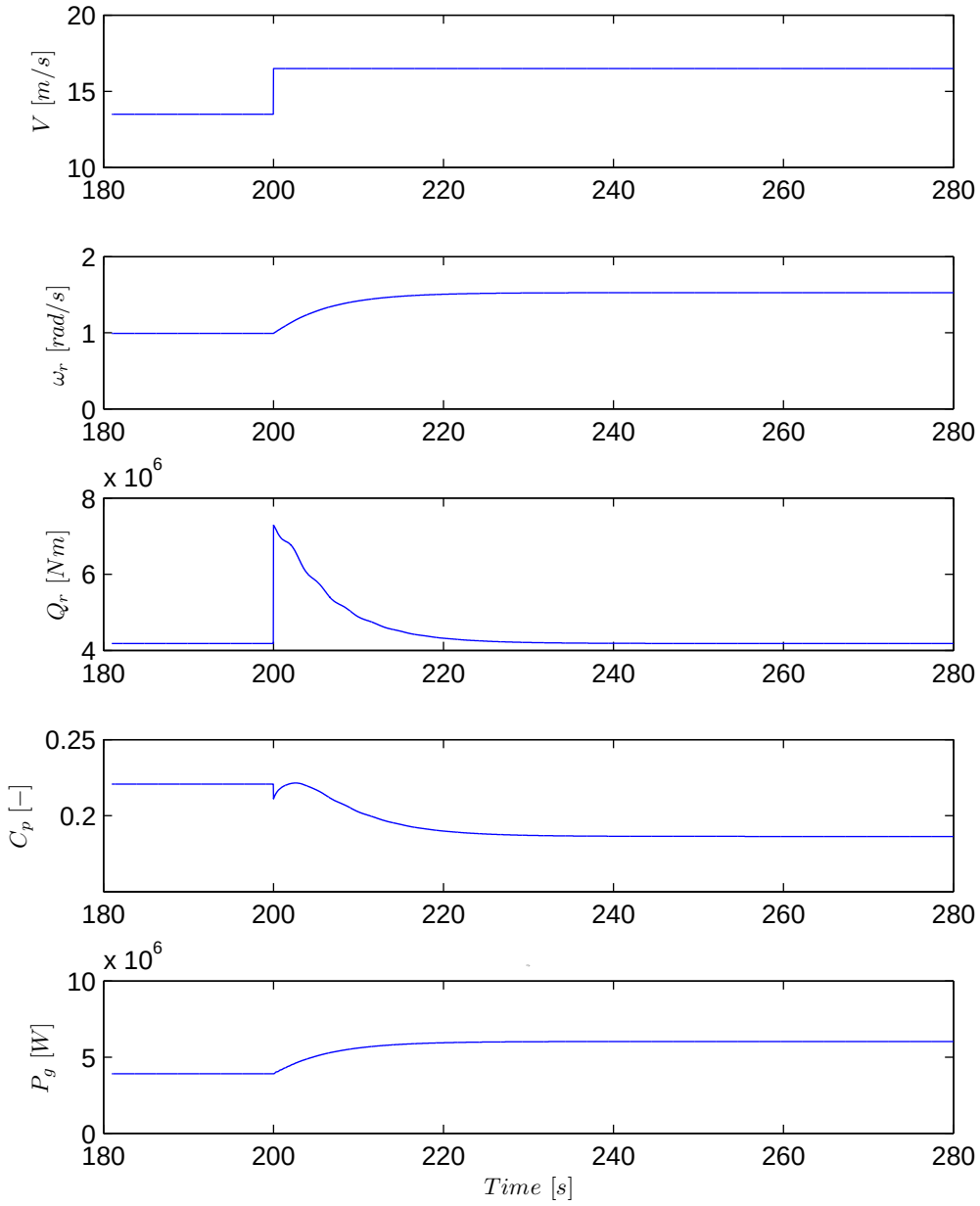


Figure 3.18: *The figure shows the plots of the wind speed V, rotor speed ω_r, rotor torque Q_r, power coefficient C_p and the generator power P_g when a step in the wind speed from 13.5 to 16.5 m/s is added.*

As seen in the figure, the wind speed V steps from 13.5 to 16.5 m/s at time 200. At the same time the rotor torque Q_r becomes higher before it stabilizes at the same level again. The rotor torque

becomes higher according to

$$Q_r = \frac{1}{2}\rho\pi R^3 V^2 C_Q(\lambda,\beta). \tag{3.8}$$

When the rotor torque becomes larger than the generator torque, the rotor speed will accelerate. It will accelerate until the torques again are equal. They will become equal again because the tip speed ratio changes as the rotor speed changes and thereby changes the C_p value. The generator power increases because the rotor speed increases which will increase the generator speed and the power will increase according to

$$P_g = \omega_g Q_g \eta_g. \tag{3.34}$$

3.10.2 Test 1.2 generator torque step

The second test simulates what happens with a generator torque step. The simulation result is seen in Figure 3.19.

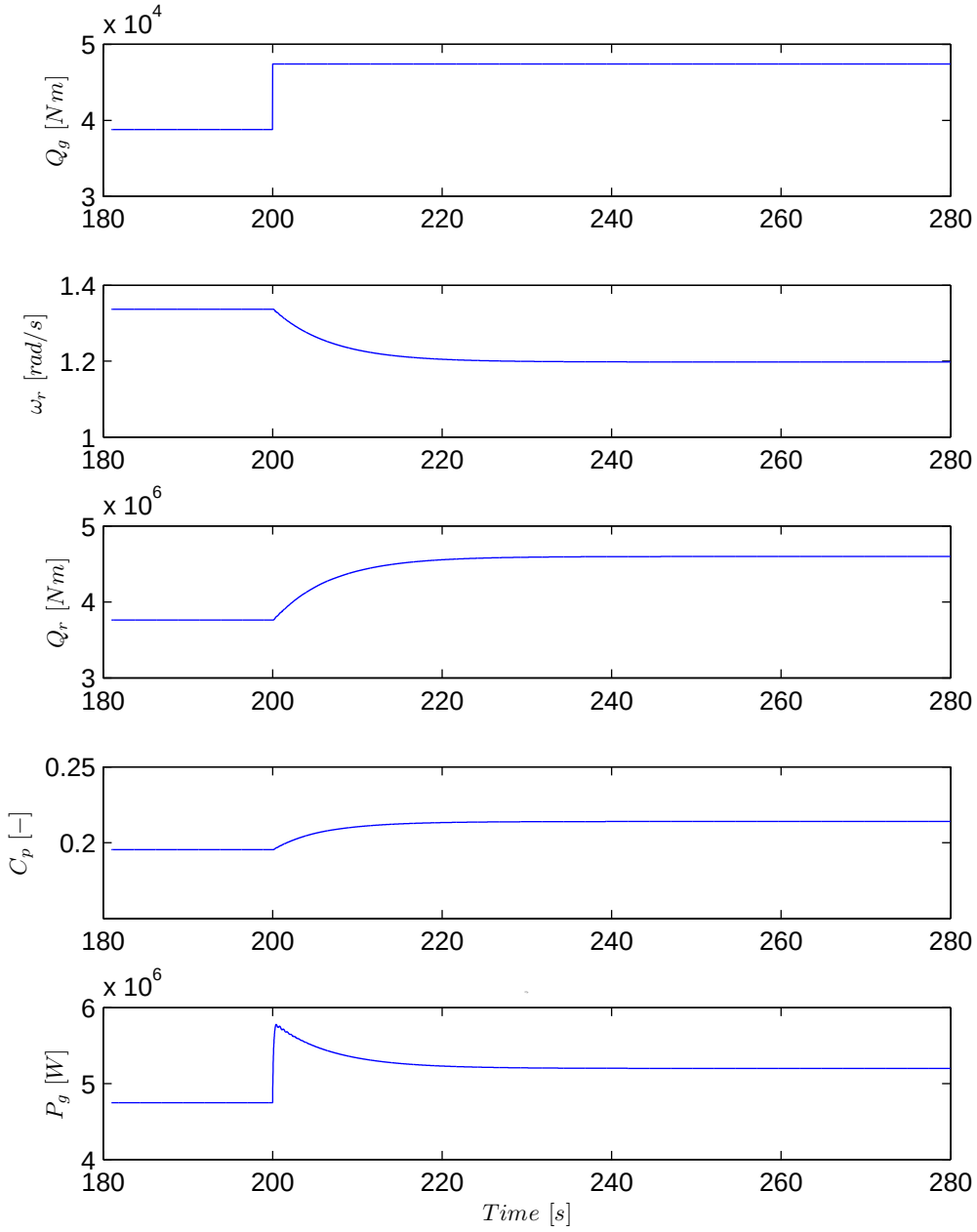


Figure 3.19: *The figure shows the plots of the generator torque Q_g, rotor speed ω_r, rotor torque Q_r, power coefficient C_p and the generator power P_g with a step in the generator torque from $38.8 \cdot 10^3$ to $47.4 \cdot 10^3 Nm$ as input.*

The figure shows that the rotor speed ω_r drops as the generator torque Q_g increases. A change in the rotational speed changes the tip speed ratio which increases C_p. This leads to an increase in the rotor torque until the rotor torque and the generator torque finds an equilibrium. The generator power given as equation (3.34) increases due to the raise in generator torque even

though the generator speed drops because of the drop in rotor speed.

3.10.3 Test 1.3 pitch step

Third test simulates what happens with a pitch step from 9.4 to 11.48deg. The simulation result is seen in Figure 3.20

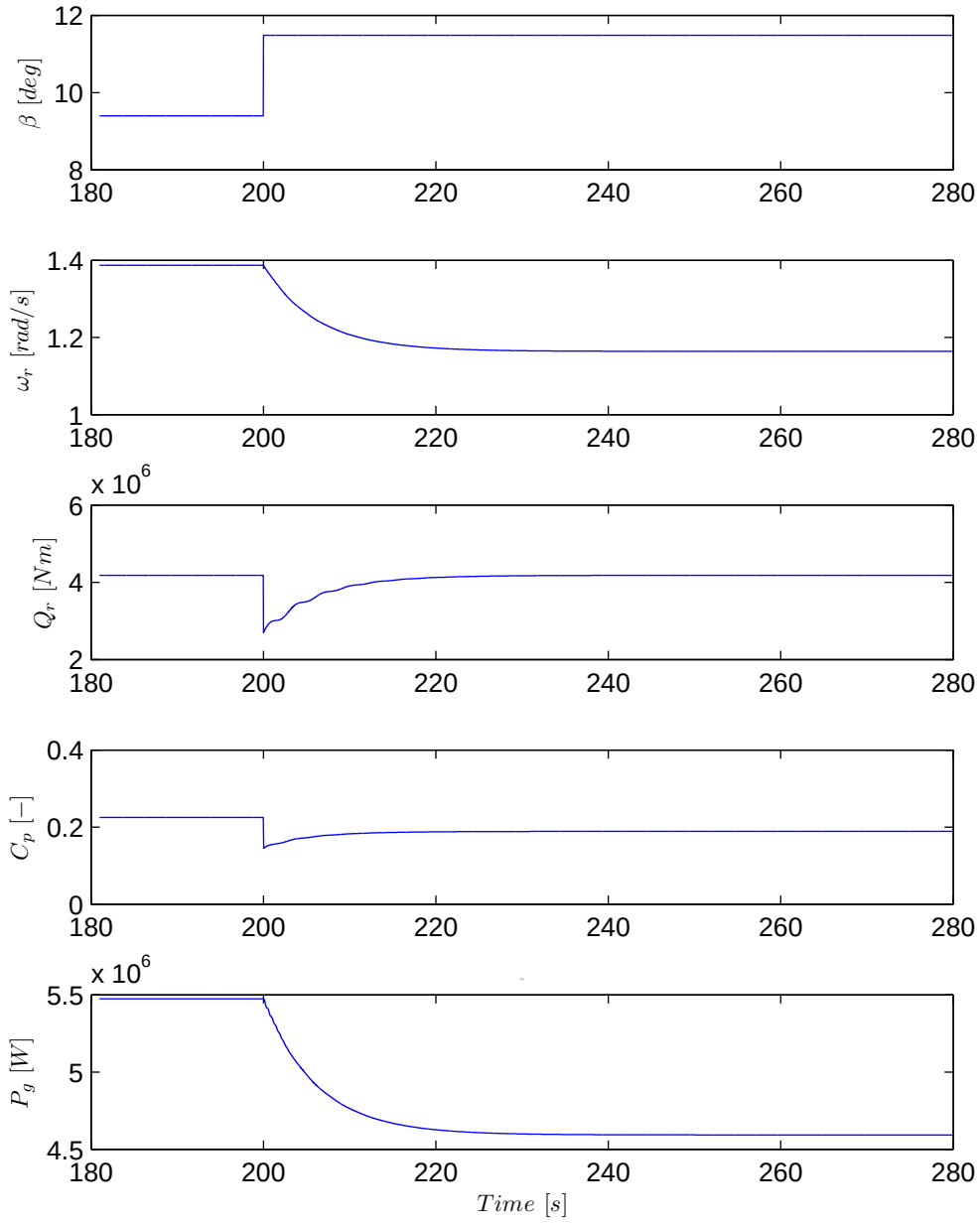


Figure 3.20: *The figure shows the plots of the pitch* β*, rotor speed* ω_r*, rotor torque* Q_r*, power coefficient* C_p *and the generator power* P_g *with a step in the pitch from* 9.4 *to* 11.48deg.

The figure shows that the rotor speed ω_r drops as the pitch β is increased, this is due do the power coefficient C_p drops and thereby drops the rotor torque as well. When the rotor torque drops it leads to a drop in the rotor speed. A drop in rotor speed, leads to a change in the power coefficient. The drop in rotor speed also leads to a drop in generator speed which leads to a drop in the generator power P_g.

3.10.4 Test 2 nonlinear model against linear model and another NREL model

To validate the dynamics and values of the model, the model is compared to another NREL model. The result is seen in Figure 3.21. The figure also shows the linear model which is tested to see if it fits the nonlinear model.

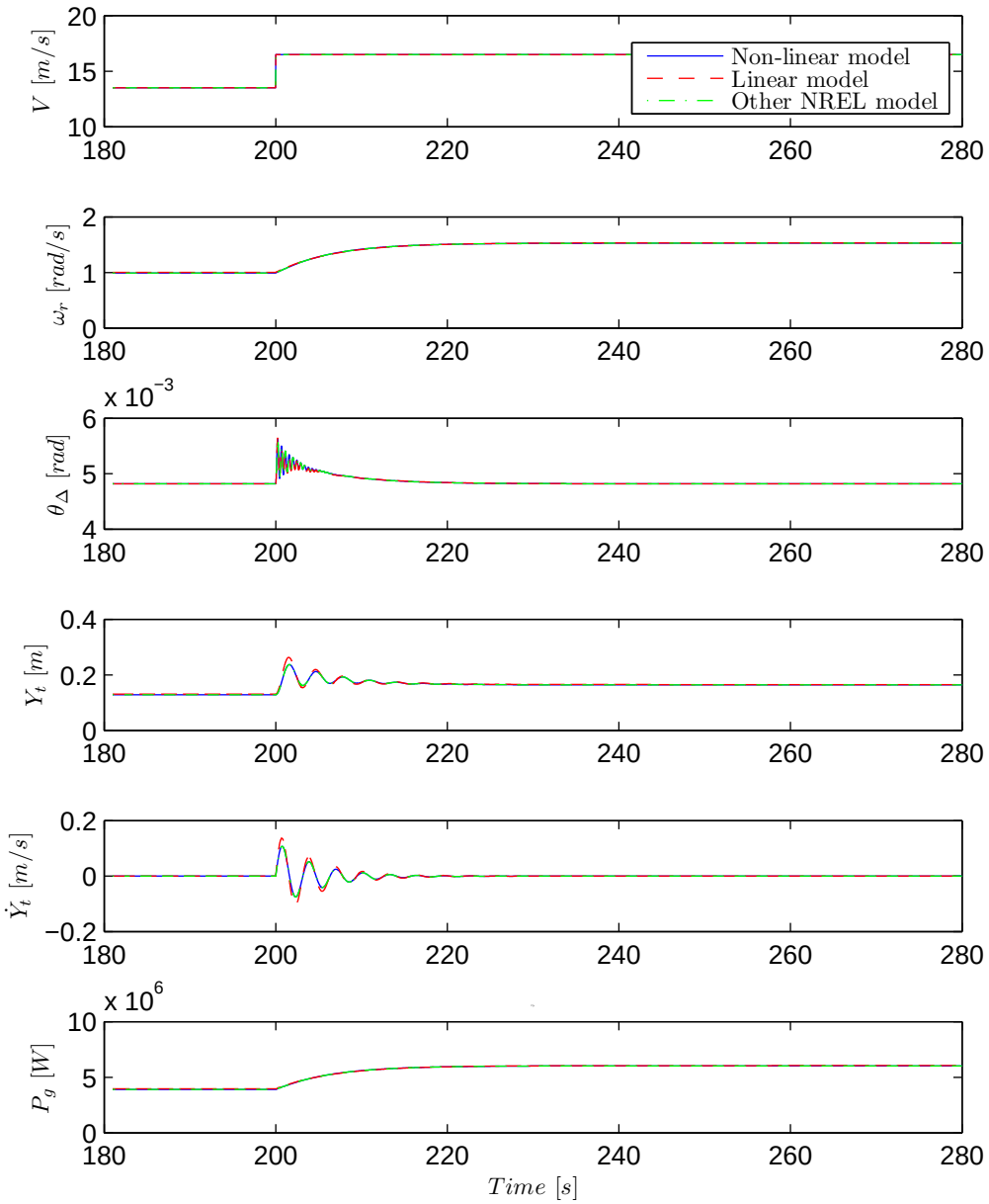


Figure 3.21: *Validation of nonlinear model vs. linear model with operating point at a wind speed of 15 m/s simulated with a step in the wind from 13.5 to 16.5m/s.*

As seen in the figure, the responses from the three models are almost identical. The nonlinear and linear model are therefore assumed to be suitable for simulations and design of a controller.

3.11 LIDAR Sensor Model

Light Detection And Ranging (LIDAR) was briefly introduced in the System Description in section 2.2. In this section the LIDAR technology is analyzed for the purpose of developing a simulation model of a LIDAR system. No real LIDAR system was available for this thesis so the model is developed from a general analyses of the technology. The section contains an analyses of the LIDAR functionality, a description of the assumptions made for the simulation model and finally a description of the final implementation.

3.11.1 LIDAR Functionality

The idea in this project is to use light detection and ranging (LIDAR) to measure the wind speed in front of the rotor disc. The LIDAR works by splitting a laser beam into two beams. One of the beams is sent out of the LIDAR device and reflected by the particles in the air, which moves with the wind speed. This is called backscattering. When the particles that reflects the laser beam is in motion, the frequency of the beam is changed, called a doppler shift. The LIDAR detects the reflected beam and by comparing it to the original beam, the speed of the particles and thereby the wind speed, can be calculated [26]. An illustration is seen in Figure 3.22.

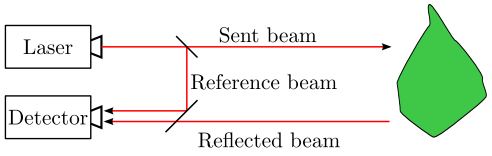


Figure 3.22: *The essential function of the LIDAR. The wind speed is measured by measuring the doppler shift of the send laser beam*

The LIDAR can measure the wind speed at all points along the beam, the maximum measurable distance is determined by the placement of the optics in the LIDAR setup and referred to as the focal point. The optic setup is considered to be outside the focus of this thesis and it is assumed the chosen distance for the focal point is within the possibilities of a real system.

Along with a fixed focal point, a fixed placement and direction of the LIDAR is chosen. There are existing models of LIDAR, with the feature to change the direction of the beam, it is however chosen to use the fixed focus in this thesis, which seems like the most common setup.

Considering the placement of the LIDAR, there are several possibilities. One placement could be on the nacelle as in [26]. With this setup the LIDAR would measure in one fixed direction. Furthermore a placement on the nacelle would mean, that the rotor blades would move in front of the LIDAR periodically and block it, this could however reasonably simple be filtered away.

Another possible placement is on the ground near the turbine [26]. This option would cause a problem, as the LIDAR has to measure in all directions around the turbine, as the turbine changes

orientation when the wind direction changes. The last considered placement, and the one used in this thesis, is in the hub. If the LIDAR is placed in the hub, the laser beam will have a clear line of sight upwind from the turbine, as no blades comes in front of the beam. There is also no problems in change of wind direction, because it is mounted on the turbine it moves with the nacelle when it turns upwind.

By placing the LIDAR in the hub of the turbine, it can be set to point directly upwind, like the center line on Figure 3.23. It would then measure the point wind speed along the beam and this would be the wind speed in the center of the rotor plane. This could be useful for estimating the mean wind speed but give no additional information about the wind field.

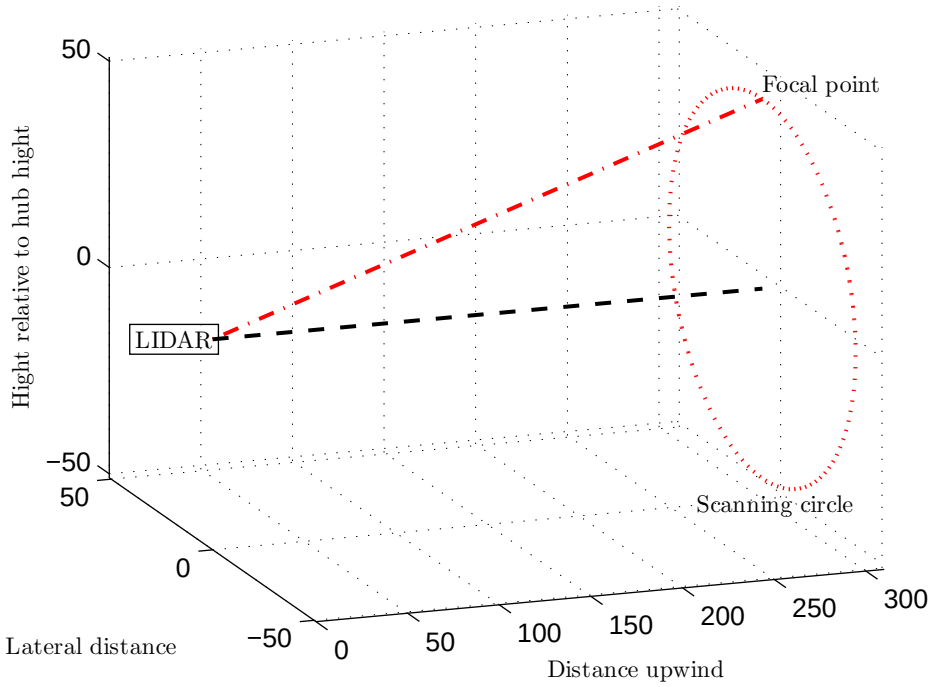


Figure 3.23: *Illustration of a LIDAR scanning direct upwind and in a circle.*

If the LIDAR is placed at an angle to direct upwind, the beam follows the red line on Figure 3.23 and as the hub rotates the focal point would move around in a circle in the wind field, the scanning circle on Figure 3.23. Including that the measured wind moves over time towards the turbine, a helix scan pattern emerges, shown in Figure 3.24

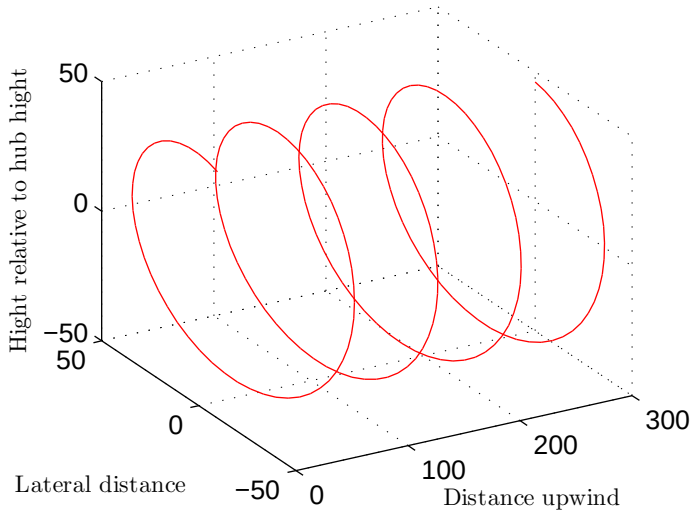


Figure 3.24: *A LIDAR measuring at an angle generates a helix pattern over time.*

Putting the LIDAR at an angle and taking advantage of the possibility of measuring several points along the LIDAR beam, several circles can be found from the LIDAR data which provide more information of the wind field. These other scanning circles have a smaller radius, and as they are measured closer to the LIDAR, the prediction time is decreasing. This is illustrated in Figure 3.25.

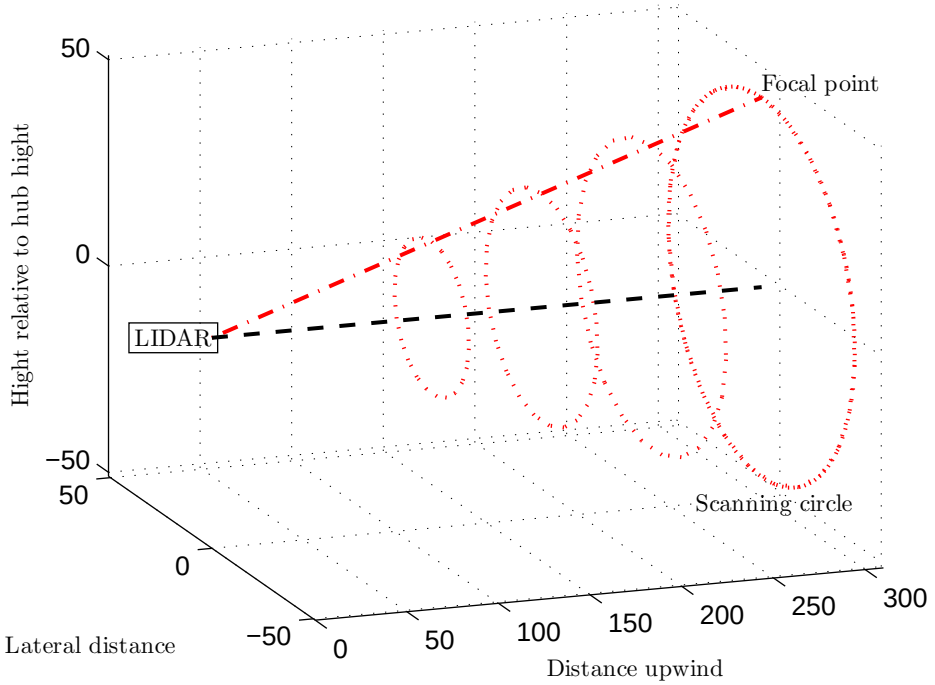


Figure 3.25: *Illustration of a LIDAR scanning several circles.*

Putting the LIDAR at an angle gives more measurement points in the wind field, but it also suffers from one of the limitations of the LIDAR, that the LIDAR can only measure the wind speed in the direction of the laser beam. This is not a problem if the LIDAR points directly upwind, but when placed in an angle, the measurements are influenced by the vertical and horizontal components of the wind.

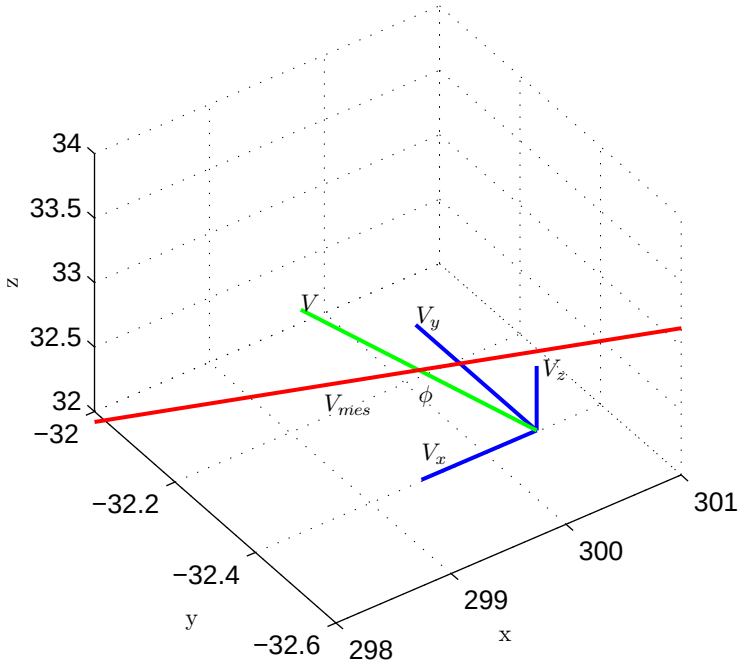


Figure 3.26: *Illustration of the LIDAR measuring the wind speed at an angle of ϕ between the LIDAR and the wind direction.*

Figure 3.26 illustrates the problem with pointing the LIDAR in another direction, than the wind direction that needs to be measured. On the figure the point wind speed that is measured V, is the resulting wind speed from the wind component in the upwind direction V_x, horizontal direction V_y and vertically direction V_z. The wind speed is then according to [23] given by

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2} \tag{3.44}$$

where the measured wind speed, V_{mes}, can be determined by

$$V_{mes} = V cos(\phi) \tag{3.45}$$

where ϕ is the angle between the LIDAR beam and the resulting wind speed. Seeing upwind from the turbine, it is the wind speed in the direction of the turbine, the wind component V_x that affects the turbine. From the geometric properties of the LIDAR measurements, it is seen that the influence from the horizontal and vertical wind speeds, V_y and V_z, is increased with the angle between the LIDAR beam and directly upwind.

This leads to a trade off between the area covered and measuring angle. The trade off can be

minimized by increasing the distance at which the focal point is placed, when the focal point is set longer away, a larger radius is obtainable with a smaller angle.

3.11.2 Assumptions and Implementation of the LIDAR Model in a Simple Simulation

As no LIDAR measurements are available a model of a LIDAR has to be used instead, in this case a simplified LIDAR model in LACflex is used. The LIDAR in LACflex is one that measures in a circle, as in Figure 2.2. The LIDAR model provides the point measurements in a circle, from the wind field generated in LACflex, at a given radius and time. The point measurement in each point is given in the three directions V_x, V_y and V_z, which is used as the LIDAR measurements, see figure 3.26.

A simple implementation of LIDAR is used for simulations together with the derived nonlinear model of the turbine. A simple model is a good starting model, as it enables testing of the potential of LIDAR in a controlled way where disturbances are removed. If it does not work on a simplified model it probably will not work on a more complex one. If it do work, the complexity of the model can gradually be increased to a more realistic model. The derived nonlinear model of the turbine is described in section 3.4, and uses the wind speed in the upwind direction, V_x on Figure 3.26. For a similar implementation of the LIDAR only the upwind speeds are used as LIDAR measurements. This is not a realistic implementation as in reality the LIDAR would have to be placed directly in front of each measuring point to gain these measurements, but at small angles the other two wind components can be assumed zero.

Simulated LIDAR data from LACflex is used in the simulations. Figure 3.27 illustrates how the LIDAR data is obtained in LACflex. The green points illustrates the wind field, as are at the turbine at time k and the red points are the LIDAR readings at time k of the wind field that will reach the turbine at time $k + Hp$.

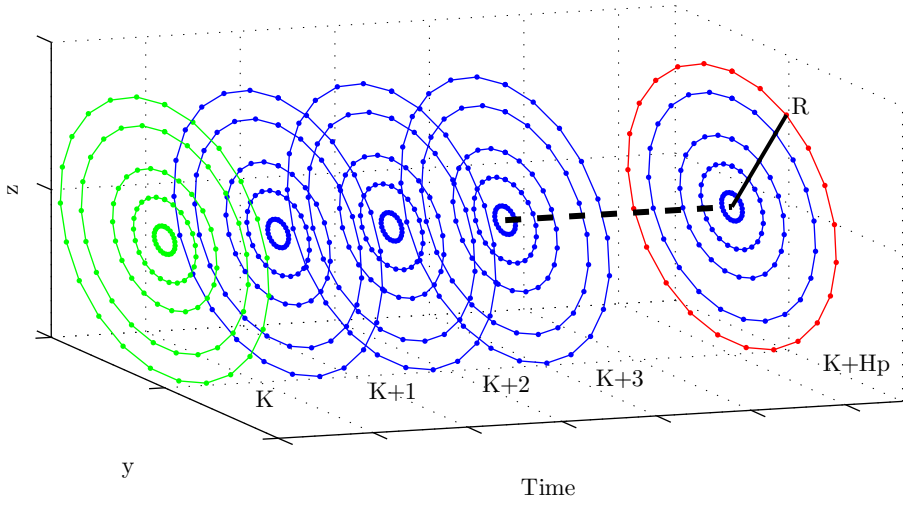


Figure 3.27: *The figure illustrates graphical what LACflex does. The generated wind fields which in LACflex is represented as circles with different radius. The LIDAR measures around a circle, with radius R at Hp Time steps in the future.*

In this thesis the distance to the focal point in chosen to be 300 meters and the radius of the scanning circle is set to 100% of the rotor radius. The chosen distance is based on the chosen prediction time used in the implementation of the model predictive controller, descriped in section 4.4 which is set to 20 seconds, the gives a distance of 300 meters with a mean wind speed of 15 m/s.

For the LIDAR signal to be used together with the model, the LIDAR measurements has to be converted into a single wind speed. This is done by calculating the average of all the V_x measurements at each sample, which gives a LIDAR signal consisting of an average of the measurements generated from the wind field at each sample.

The LIDAR signal is measured at some distance upwind from the turbine and as the wind changes over time as it moves towards the rotor disc, the LIDAR measurements should be propagated with a model for how wind changes over time. As Taylors frozen turbulence is assumed for the wind, meaning that the wind does not change as it moves, the LIDAR measurements are not propagated. Another assumption included in the LACflex LIDAR model, is the assumption that the LIDAR can measure a full circle at each wind sample.

Because the LIDAR signal is measured at some time in the future, it has to be delayed the time it is measured in the future. The time input that the LIDAR model uses in LACflex, is used instead of distance as it is more convenient to use than distance. In reality the time would have to be calculated, from the mean wind speed and the distance to where the LIDAR measures. For instance if the LIDAR measures at 300 meters and the mean wind speed is 15 m/s, it takes the wind 20 seconds to move the 300 meters to the rotor disc. This delay function is made, by including what works like a first in first out (FIFO) buffer. This means that at every new measurement, the oldest measurement is removed and the new is included, an illustration is seen in Figure 3.28. This would in reality be more difficult as the mean wind speed is not defined and has to be calculated. This could give some problems that the wind, when it is at the rotor disc does not fit with the LIDAR measurements, because the delay is a little off and the controller reacts wrong because of the LIDAR measurements is not the ones to the respective wind.

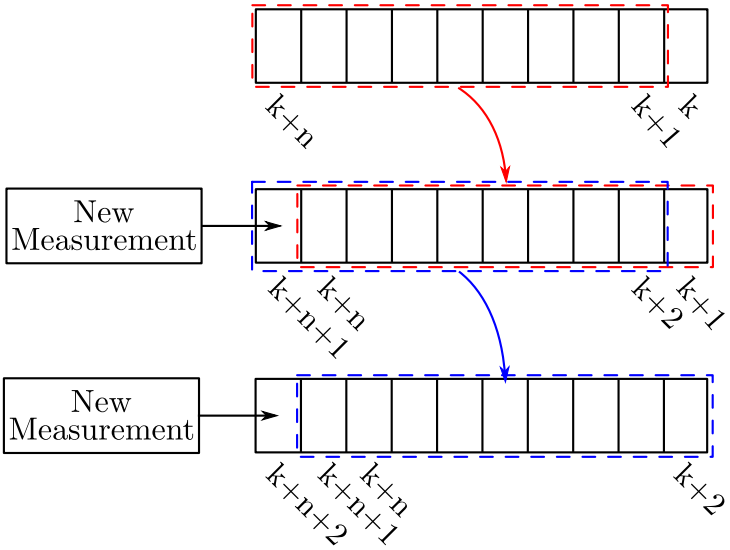


Figure 3.28: *Illustration of the LIDAR memory that works like a FIFO buffer. The illustration shows two steps with new measurement data from the LIDAR. Each time step the data on the second space and forward is delayed once and the new measurement is put into the array of data.*

In the LIDAR model all the outer factors from the environment are removed. Removing the outer factors is the first step in testing if LIDAR can have a positive influence on the fatigue of the tower and drive train. If the test in a simplified controlled environment does not seem to work, it probably will not work either in reality where uncertainties and disturbances influences the result.

Chapter 4

Control Design

In this chapter the damage equivalent load used for testing the developed controllers is described along with the developed controllers. The developed controllers are a model predictive controller that uses LIDAR in the prediction and a PI controller that serves as benchmark to the model predictive controller.

4.1 Control Design Approach

To begin with, a standard controller is designed that serves as a benchmark to the later designed model predictive controller. The standard controller that serves as a benchmark is a PI controller, that is based on literature. The benchmark controller is divided into two controllers, a torque controller for low wind speeds and a pitch controller for high wind speeds. Added to the standard controller is damping of the tower and drive train to make it a fair comparison between the model predictive controller and the benchmark controller.

To minimize the tower and drive train fatigues, compared to the benchmark controller, the knowledge a LIDAR provides about the wind is used in the control design. The LIDAR information is used in the designed model predictive controller where the information from the LIDAR is used in the prediction model. This should make the controller able to better predict and react according to what happens in the future, when it have knowledge of the incoming wind.

To test which controller is best, the benchmark or the model predictive controller, the damage equivalent load is calculated for the tower and drive train. The loads are calculated for both controllers and compared. The test is carried out with three different winds A, B and C each with 14% 16% and 18% turbulence intensity accoding to IEC61400-1 [31].

4.2 Controller Test

To be able to test which controller is best, it is necessary to have some way to compare them, as it is not always easy to see which one is best. The test of the controller should reflect the goal for

the controller, as it is always a trade off between for instance, the use of pitch and performance. The goal for the controller is to decrease the tower and drive train fatigue without sacrificing the power. One way of testing is to calculate the damage equivalent load for the tower and drive train and use that as a representation of the controller performance. The controller with the lowest values is the best controller at the given test criteria.

As previously described there is always a trade off. To include this trade off in the test result and better be able to judge the controllers and compare them, also the standard deviations of the power and pitch are calculated. This to check if the power suffers and to check the use of the pitch when the fatigue is lowered.

4.2.1 Damage Equivalent Load

This section is based on [2] and [11] and describes the theory of damage equivalent load. The damage a material is exposed to depends on the number of cycles and how large the amplitudes of the stress are. For different materials SN curves are made that shows how many cycles for a given stress value a material can withstand before it fails. An illustration of a SN curve is seen in Figure 4.1.

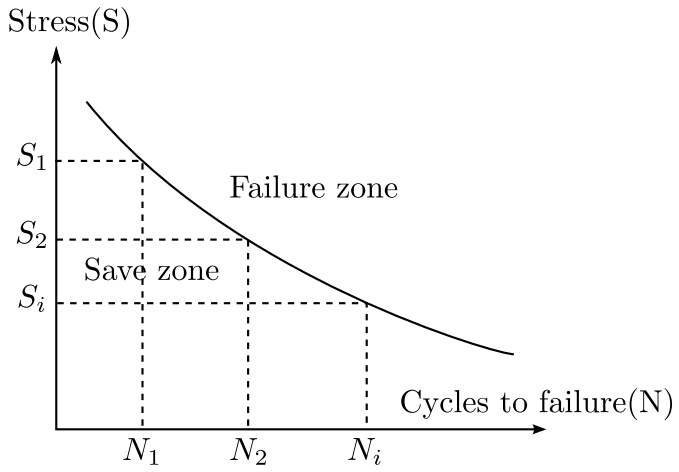


Figure 4.1: *Illustration of a SN curve. The material is save if it is below the curve and fails above.*

The equation for the SN curve is given as

$$N_i S_i^k = K \tag{4.1}$$

where N_i is number of cycles for breakdown at a given stress range S_i, k and K are material parameters. Using equation (4.1) the fractional damage a material has been exposed to can be calculated as

$$D_i = \frac{n_i S_i^k}{N_i S_i^k} = \frac{n_i}{N_i} \tag{4.2}$$

where n_i is the number of cycles at the stress amplitude S_i and $n_i < N_i$. This can according to the

Palmgren-Miner rule, that states that the total stress can be calculated as the sum of the individual fractional damages, be rewritten to

$$D = \sum_{i=1}^{j} \frac{n_i}{N_i} \tag{4.3}$$

where i is one of j stress values. Breakdown is said to be reached for the the total damage $D \geq 1$. If the SN curve and the material parameters are unknown the damage can be calculated as

$$D = \sum_{i=1}^{j} n_i S_i. \tag{4.4}$$

To calculate the damage the software Mcrunch is used [9]. Mcrunch calculates the damage as

$$D = \left(\sum_{i=1}^{j} {n_i}^m S_i \right)^{\frac{1}{m}}. \tag{4.5}$$

where m, is a material constant which for tower is set to 4 and for drive train to 6. To calculate the damage it is necessary to know how many cycles and the stress range the material has been exposed to. To calculate these values rainflow counting is an option which also Mcrunch use.

4.2.2 Rainflow Counting

To be able to calculate the damage the number of cycles must be known. The number of cycles is easy to calculate if the amplitude is constant as illustrated in Figure 4.2, where it is easy to count that there is 5 cycles with an amplitude of 5 and 4 cycles with an amplitude of 10. The following section is based on [2].

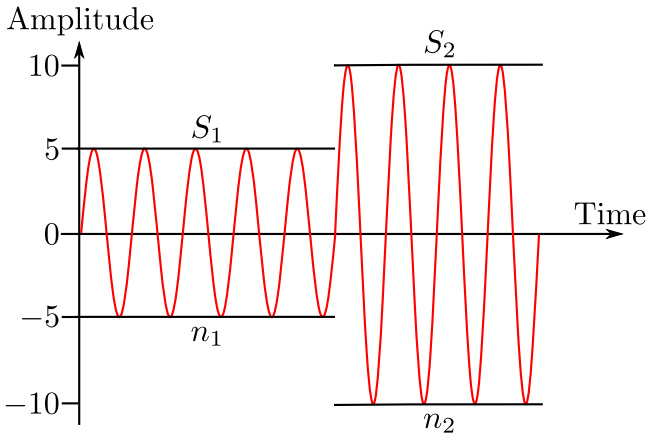


Figure 4.2: *Illustration of a uniform signal where there is no need for rainflow counting. It is easy to see there is $n_1 = 5$ cycles at stress range $S_1 = 5$ and $n_2 = 4$ cycles at a stress range $S_2 = 10$.*

If the signal instead looks like the one in Figure 4.3, it is not as easy to calculate the number of cycles and the stress range.

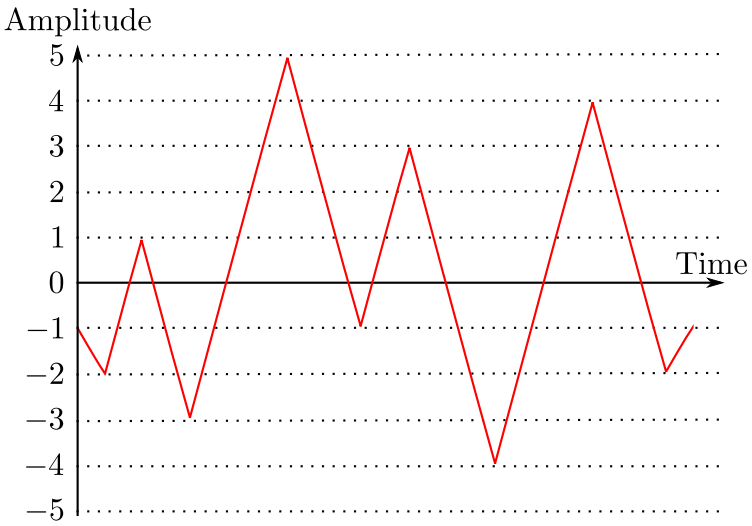


Figure 4.3: *Random amplitude signal where the number of cycles and the stress has to be calculated with rainflow counting.*

Instead a method called rainflow counting can be used, which converges a signal into number of cycles for given stress ranges. The principle in rainflow counting is as follows: the signal is converted into peaks and valleys and is turned 90 deg clockwise. It now looks like a pagoda which is a tower construction with multiple eaves where water is imagined to flow from each peak and valley and drips down the pagoda as seen in Figure 4.4.

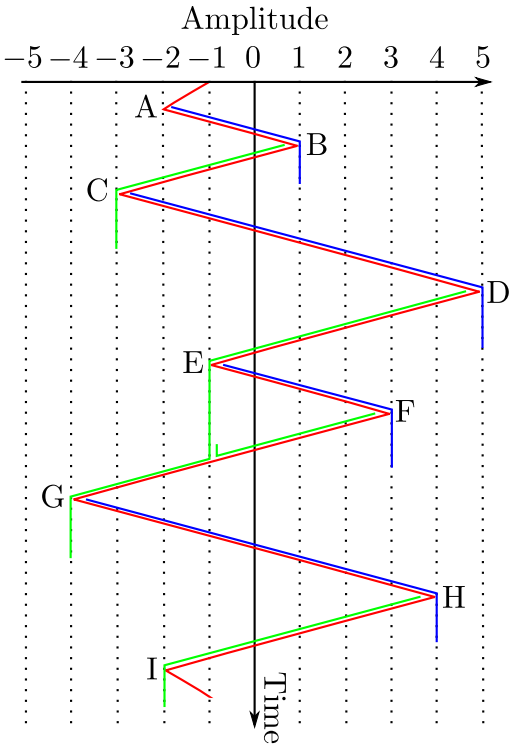


Figure 4.4: *The signal turned 90 deg and with illustration of rainflow.*

The water flows down the pagoda according to the following rules [2, p.18]:

1. A rainflow path starting at a valley will continue down the pagoda roofs, until it encounters a valley that is more negative than the origin. From the figure, the path that starts at A will end at B as C is more negative than A.

2. A rainflow path is terminated when it encounters flow from a previous path. For example, the path that starts at F is terminated as it flows into a path from above started at D.

3. A new path is not started until the path under consideration is stopped.

4. Valley generated half cycles are defined for the entire record. For each cycle, the stress range Si is the vertical excursion of a path.

5. The process is repeated in reverse with peak generated rainflow paths. For a sufficiently long record, each valley generated half cycle will match a peak-generated half cycle to form a whole cycle.

Using the above rules gives the number of cycles with the stress ranges as in table 4.1.

Stress(S_i)	**Cycles**(n_i)	**Events**
9	0.5	D-G
8	1	C-D,G-H
6	0.5	H-I
4	1.5	E-F,B-C,F-E
3	0.5	A-B

Table 4.1: *The calculated stress ranges and number of cycles for the random amplitude signal.*

The original signal is calculated into a number of cycles and stress ranges which represents the signal. A new representation of the signal from the values gives a signal as in Figure 4.5. If the material was exposed to the new signal it would cause the same damage as the original signal.

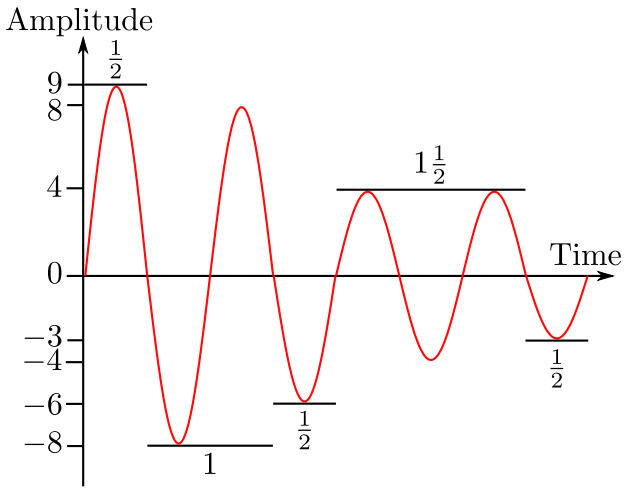


Figure 4.5: *Illustration of a signal that gives the same damage as the signal where rainflow count is performed. It has one half cycle of an amplitude of 9, one cycle with an amplitude of 8, one half cycle with amplitude 6, one and a half half cycle with amplitude 4 and one half cycle with an amplitude of 3.*

4.3 Benchmark Controller

A standard PI controller for regulating the power is developed to serve as a benchmark for the later developed model predictive controller. The development of the controller is based on [20] and [6]. The approach for regulating the power is described in control strategy in section 2.3. The approach is to design two different controllers, one below and one above rated wind speed, with two different purposes. Below rated wind speed, the purpose of the torque controller is to extract as much energy out of the wind as possible. As the wind reaches above rated wind speed, the purpose of the second controller, the pitch controller, is to keep the turbine at rated power. This is done by changing the pitch of the blades to keep the generator and rotor at rated speed.

4.3.1 Torque Control

The torque controller works below rated wind speed. Below rated wind speed the goal for the controller is to maximize the output power by extracting as much energy as possible out of the wind. The extracted energy depends on the power coefficient C_p, which must be kept at maximum value to extract the highest possible energy out of the wind. The value of C_p depends on the tip speed ratio λ and the pitch angle β. To keep C_p at maximum value the tip speed ratio has to be kept the same, as the pitch is constantly zero at below rated wind speed which is seen in Figure 2.4 in the control strategy section. To keep the tip speed ratio the same, the generator torque is controlled, which changes the rotor speed and thereby keeping the tip speed ratio at the same optimal value.

The calculated generator torque as function of the generator speed is shown in Figure 4.6. The generator torque is divided into five regions as done in [20]. In the first region the torque is zero because there is not enough wind to drive the turbine. The region $1\frac{1}{2}$ between region 1 and 2 is a start up region between start up generator speed and 30% above, where the torque changes from zero uptil the calculated value for torque to give maximum Cp. The start is calculated from the minimum rotor speed of 6.9 rpm and with a gear ratio of 97 the minimum generator speed is 70 rad/s. In the second region the torque is calculated to track the optimal C_p curve. The torque is caluclated as

$$Q_g = \frac{\pi \rho R^5 C_p}{2\lambda^3 n^3} \omega_g^2 \tag{4.6}$$

where Q_g is generator torque, ρ is the air density, R is the rotor radius, C_p is the power coefficient, λ is the tip speed ratio, n is the gear ratio and ω_g is the generator speed. Region $2\frac{1}{2}$, starts from 99% of rated generator speed to rated genertor speed, where the torque is changed from the value at 99% to rated torque. In the last region the torque is held constant at rated generator speed. The torque is implented as a table that gives the torque dependent of the generator speed.

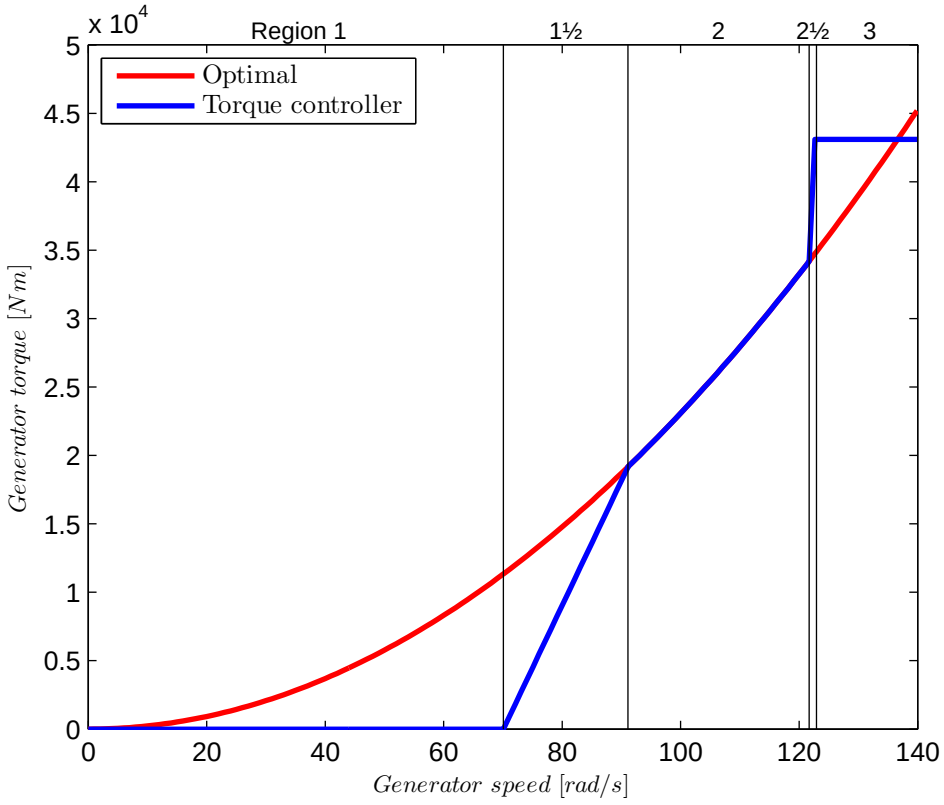


Figure 4.6: *The generator torque as function of the generator speed. The red curve is the optimal torque for tracking C_p and is given as equation* (4.6) *and the blue curve is the torque controller.*

4.3.2 Pitch Control

In above rated wind speeds the goal is to maintain the turbine at rated power. This is done by the pitch controller that changes the pitch of the blades. The generator torque is kept constant in above rated wind speeds. Changing the pitch changes how much energy is extracted from the wind, as the power coefficient drops below its maximum value. Extracting less energy from the wind enables the controller to keep the rotor at rated speed. Keeping the rotor at rated speed keeps the generator at rated speed as well and the generated power at rated power. The pitch is controlled with a PI controller as illustrated in Figure 4.7, where the gains are calculated according to [21].

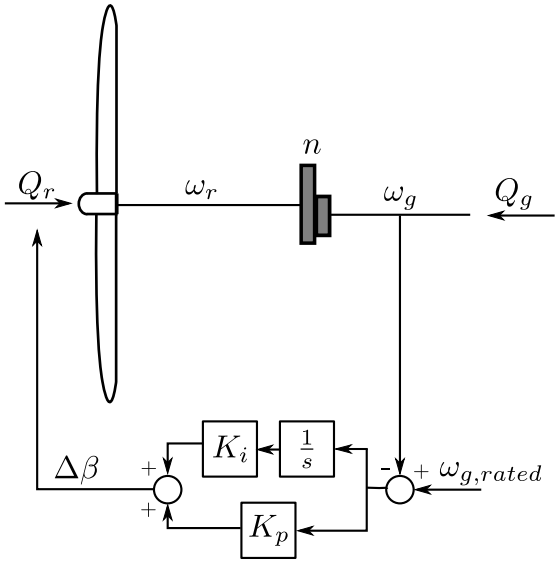


Figure 4.7: *Illustration of the pitch controller which is a PI controller and a simple illustration of a wind turbine.*

If the drive train is assumed rigid, the rotor speed can be expressed as

$$J\dot{\omega}_r = Q_r - nQ_g, \tag{4.7}$$

where the rotor speed ω_r is dependent of the difference between the rotor torque Q_r and generator torque Q_g. The generator torque is transfered to the rotor side by the gear ratio n. Replacing the rotor torque with a first order Taylor approximation yields

$$J\dot{\omega}_r = \bar{Q}_r + \frac{dQ_r}{d\beta}\Delta\beta - nQ_g \tag{4.8}$$

and given that $\bar{Q}_r = nQ_g$ in steady state, the change in rotor speed can be expressed as

$$J\dot{\omega}_r = \frac{dQ_r}{d\beta}\Delta\beta. \tag{4.9}$$

Expressing the change in pitch as

$$\Delta\beta = \underbrace{(\omega_{g,rated} - \omega_g)}_{-\dot{\phi}}\left(K_p + K_i\frac{1}{s}\right) \tag{4.10}$$

where $\dot{\phi}$ is defined as the change in generator speed $(\omega_g - \omega_{g\_rated})$. Substituting the equation

into (4.9) gives the equation

$$J \underbrace{\dot{\omega}_r}_{\frac{1}{n}\ddot{\phi}} = \frac{-dQ_r}{d\beta}\left(K_p\dot{\phi} + K_i\phi\right) \tag{4.11}$$

$$J\frac{1}{n}\ddot{\phi} + \frac{dQ_r}{d\beta}K_p\dot{\phi} + \frac{dQ_r}{d\beta}K_i\phi = 0 \tag{4.12}$$

$$J\ddot{\phi} + \frac{dQ_r}{d\beta}nK_p\dot{\phi} + \frac{dQ_r}{d\beta}nK_i\phi = 0. \tag{4.13}$$

Rewriting equation (4.13) to the standard form

$$J\ddot{\phi} + D\dot{\phi} + K\phi = 0 \tag{4.14}$$

where

$$J = J_r + n^2 J_g \tag{4.15}$$

$$D = \frac{dQ_r}{d\beta}nK_p \tag{4.16}$$

$$K = \frac{dQ_r}{d\beta}nK_i. \tag{4.17}$$

The system can be reformulated as a function of the damping ζ and the natural frequency ω_n as

$$\ddot{\phi} + 2\zeta\omega_n\dot{\phi} + {\omega_n}^2\phi = 0 \tag{4.18}$$

$$\omega_n = \sqrt{\frac{K}{J}} \Longleftrightarrow K = J{\omega_n}^2 \tag{4.19}$$

$$\zeta = \frac{D}{2J\omega_n} \Longleftrightarrow D = 2\zeta J\omega_n. \tag{4.20}$$

which leads to the two control gains

$$K_i = \frac{J{\omega_n}^2}{\frac{dQ_r}{d\beta}n} \tag{4.21}$$

$$K_p = \frac{2\zeta J\omega_n}{\frac{dQ_r}{d\beta}n}. \tag{4.22}$$

According to [20] as a rule of thumb the natural frequency is $\omega_n = 0.6\ rad/s$ and the damping is $\zeta = 0.7$. The equations for the gains depends on the pitch sensitivity $\frac{dQ_r}{d\beta}$, which changes with the pitch angle β as seen in figure 4.8. As seen in the figure there is a linear relationship between the pitch angle and the pitch sensitivity.

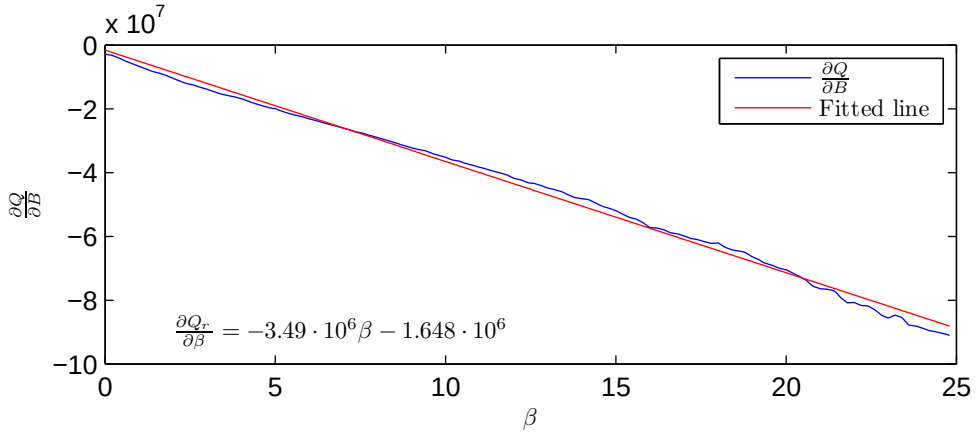


Figure 4.8: *How the pitch sensitivity changes along with the pitch.*

Instead of calculating different control gains for different pitch angles, the linear relationship is used to calculate the gains as follows

$$K_i = \frac{J{\omega_n}^2}{n(a\beta + b)} \tag{4.23}$$

$$K_p = \frac{2\zeta J\omega_n}{n(a\beta + b)}. \tag{4.24}$$

where a is the slope and b is the base for the fitted line. The equations for the gains is rewritten into a constant gain at zero pitch and a gain scheduling factor depending on the pitch as follows

$$K_i = \frac{J{\omega_n}^2}{n(a\beta + b)} \tag{4.25}$$

$$K_i\Big|_{\beta=0} = \frac{J{\omega_n}^2}{nb} \tag{4.26}$$

$$K_i = \frac{J{\omega_n}^2}{nb}\frac{1}{\frac{a}{b}\beta + 1}, \tag{4.27}$$

and for K_p

$$K_p = \frac{2\zeta J\omega_n}{nb}\frac{1}{\frac{a}{b}\beta + 1}. \tag{4.28}$$

The first fraction is the constant gain when the pitch is zero and the second fraction

$$GS = \frac{1}{\frac{a}{b}\beta + 1} \tag{4.29}$$

is the gain scheduling GS, which is the same for both K_p and K_i. The gain scheduling depends on the pitch and is seen in Figure 4.9.

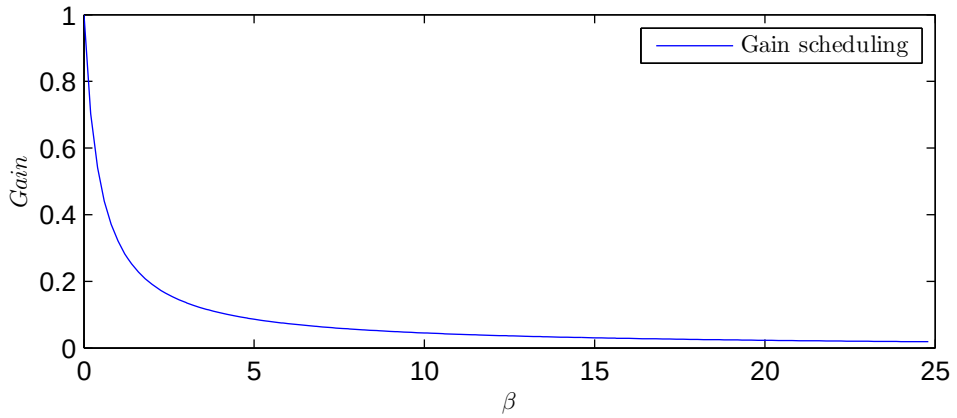


Figure 4.9: *The gain scheduling as function of the pitch angle.*

The control gains calculated with the linear gain scheduling function and with the nonlinear derivative $\frac{dQ_r}{d\beta}$ are seen in Figure 4.10.

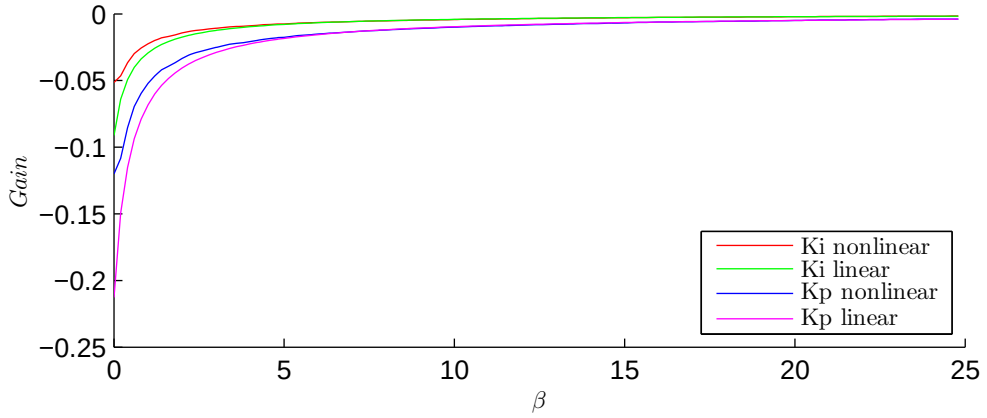


Figure 4.10: *The control gains Ki and Kp calculated with the linear gains scheduling and the nonlinear derivative.*

4.3.3 Verification of the Controller

To verify the controller, the test is divided into three parts, one that tests only the torque controller, one that tests the pitch controller and one that test both. In the tests it is checked if the system variables has the expected values.

Verification torque controller

The system is simulated with a ramp input with mean wind speeds from zero to rated wind speed, to see if the torque has the calculated torque values calculated in subsection 4.3.1. The value of the power coefficient C_p, should be at maximum in region 2. Figure 4.11 is the result from the simulation, and it is seen that the torque values are the same as the calculated ones. It is

also seen, that the power coefficient C_p is at it's maximum value in region 2, approximately the interval from 90 to 120 rad/s.

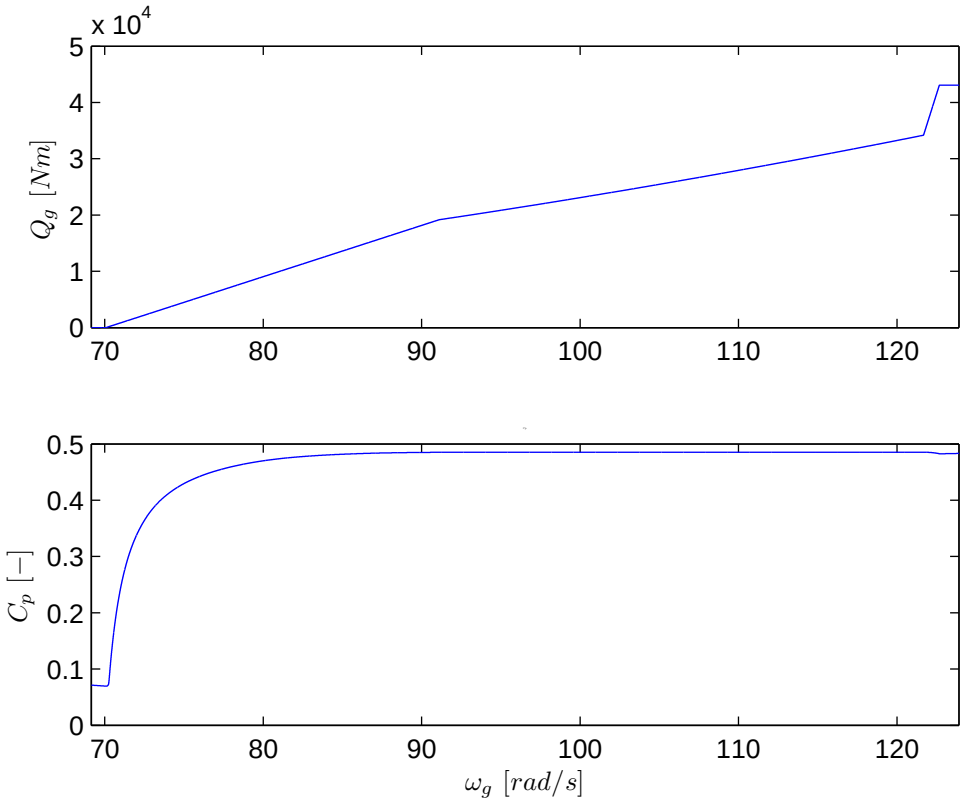


Figure 4.11: *The generator torque Q_g and the power coefficient C_p as function of the generator speed. It is seen that C_p is max in region 2, approximately the interval from 90 to 120rad/s*

Verification of Pitch Controller

The pitch controller is tested with a ramp as input to the wind speed. The result is seen in Figure 4.12, where the pitch and generator speed is plotted as function of the wind speed.

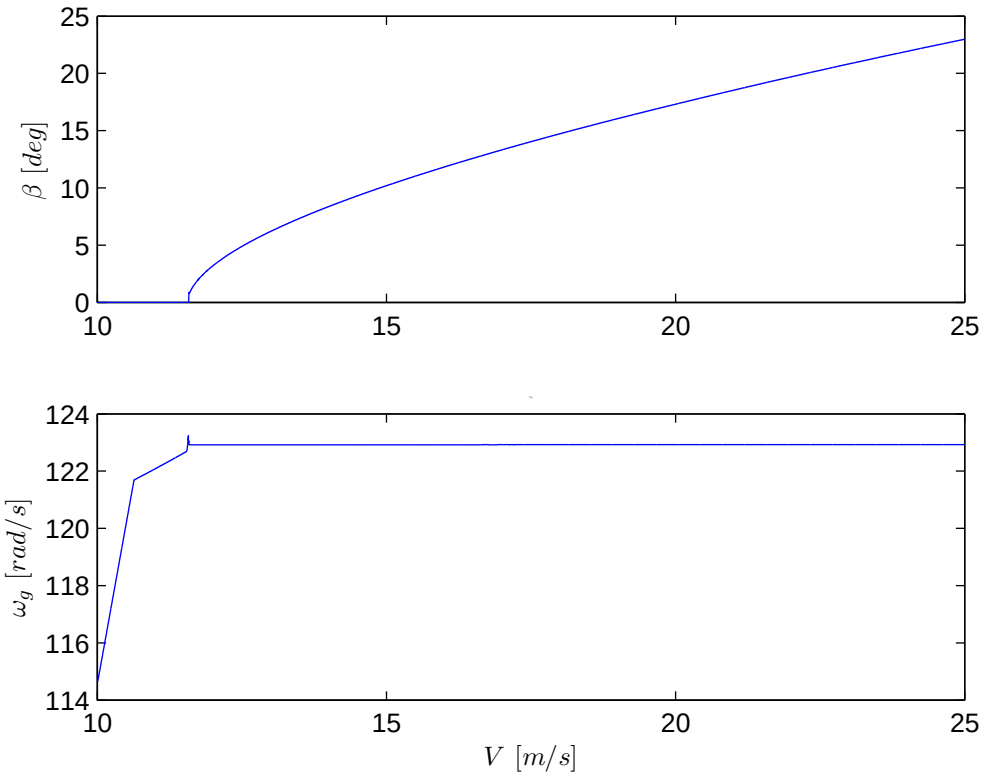


Figure 4.12: *The pitch β and generator speed as function of the wind speed V. It is seen that when the pitch controller is active it keeps the generator speed at the rated value.*

It is seen that the pitch angle is similar to the one calculated in control strategy, and it is seen that the change in pitch holds the generator speed ω_g at the rated value.

Verification of Both Controllers

Simulation with different mean wind speeds from 0-25m/s with turbulence is carried out. The simulated power for all cases, is held up against the optimal power as seen in figure 4.13.

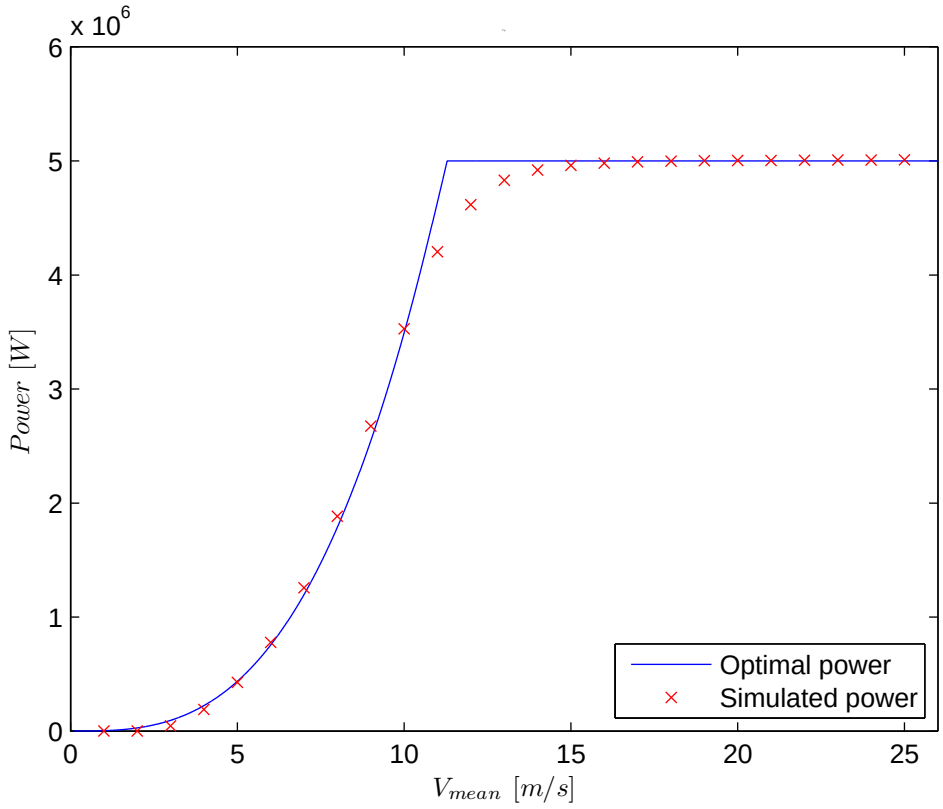


Figure 4.13: *The average power in a simulation compared to the optimal power over the entire operating area of the turbine.*

It is seen that the produced power is consistent with the optimal power when well belove and well above the rated wind speed. Around the rated wind speed the simulated power levels drops belove the optimal, this is most likely coursed by the logic that is used to change between the torque and pitch controller. But also when the wind drops below rated wind it is not possible to generate the rated power and when the wind is above the rated power it is kept constant. This also leads to it is not possible to produce the optimal power when the wind sometimes drops below rated wind speed.

4.3.4 Tower Damping

According to [6] the tower movement can be damped with an additional damping coefficient D, if a change in the pitch according to the tower velocity is included. To begin with the tower movement can be described as

$$M\ddot{Y}_t + B\dot{Y}_t + KY_t = \Delta\beta\frac{\partial F}{\partial\beta} + F, \tag{4.30}$$

and if $\Delta\beta$ is substituted with $G\dot{Y}_t$, it gives

$$M\ddot{Y}_t + B\dot{Y}_t + KY_t = G\dot{Y}_t\frac{\partial F}{\partial\beta} + F. \tag{4.31}$$

Subtracting the operating point and rearranging the equation yields

$$M\ddot{Y}_t + \left(B - G\frac{\partial F}{\partial \beta}\right)\dot{Y}_t + KY_t = 0, \tag{4.32}$$

and by simplifying the equation by defining G as the gain $-D$ divided by the derivative of the force, the gain D is the additional damping of the tower

$$G = \frac{-D}{\frac{\partial F}{\partial \beta}} \tag{4.33}$$

$$M\ddot{Y}_t + \left(B - \left(\frac{-D}{\frac{\partial F}{\partial \beta}}\right)\frac{\partial F}{\partial \beta}\right)\dot{Y}_t + KY_t = 0. \tag{4.34}$$

$$M\ddot{Y}_t + (B+D)\dot{Y}_t + KY_t = 0. \tag{4.35}$$

The tower damping

$$\Delta\beta = \frac{-D}{\frac{\partial F}{\partial \beta}}\dot{Y}_t = G\dot{Y}_t. \tag{4.36}$$

is included in the system as seen in Figure 4.14.

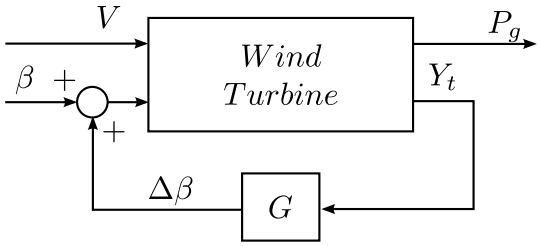


Figure 4.14: *Diagram of how the tower damping is included in the system.*

The tower damping is tuned at different wind conditions and a gain of $G = 4$ is chosen, as it was the value where the tower was damped the most and affecting the drive train and power less. $G = 4$ corresponds to an additional damping coefficient $D = 263.04 \cdot 10^3$ compared to the towers own damping coefficient $B = 225.07 \cdot 10^3$. The calculated damping ζ is for the tower 0.08 and with the extra damping 0.1735. A step on the tower is seen in figure 4.15.

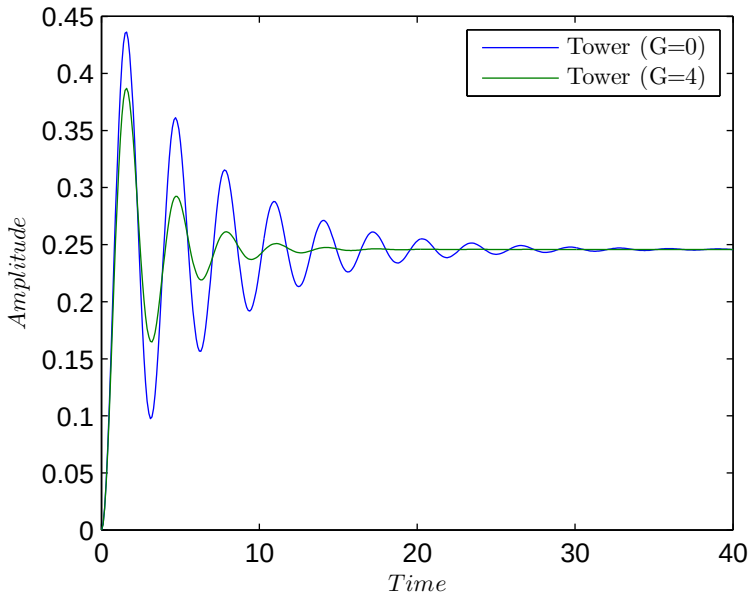


Figure 4.15: *Step on the tower with and without damping.*

To test how the standard controller with tower damping performs compared to the standard controller a test at 16% turbulence intensities and mean wind 15 m/s is carried out. The result is seen in Figure 4.16 which shows a plot of the simulation and Figure 4.17 showing the performance in percent compared to the standard controller.

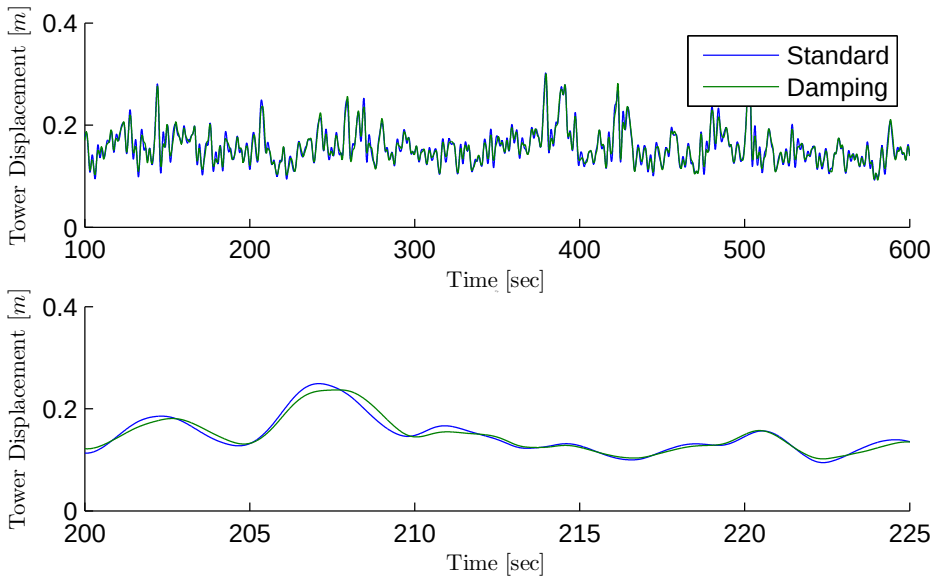


Figure 4.16: *Time series plot for the tower fore-aft mode displacement with and without tower damping included. The bottom figure is a slice of the top figure.*

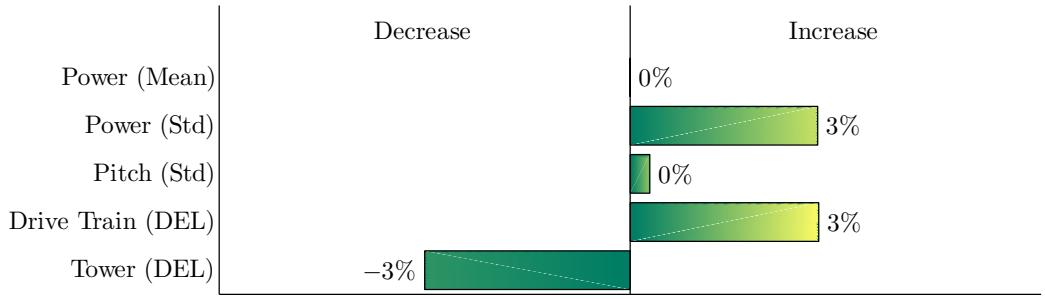


Figure 4.17: *The difference between the standard controller with tower damping compared with the standard controller as reference. The test is carried out with a wind with a turbulence intensity of 16%.*

4.3.5 Drive Train Damping

A way to damp the drive train oscillations is to add a filter [6]. The filter damps the oscillations by adding a small variations with the frequency of the drive train to the generator torque, which is constant above rated wind speed. The filter is given as

$$\frac{\Delta Q_g}{\omega_g} = G_{DT}\frac{2\zeta\omega_n s(1+s\tau)}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{4.37}$$

where ω_n is the natural frequency of the drive train, ζ is how broad the filter is and G_{DT} is the gain. Giving the natural frequency is 14 rad/s the filter is tuned in different wind conditions. The chosen filter has a gain G_{DT} of 500 and a ζ of 5. A bode plot of the filter is seen in Figure 4.18.

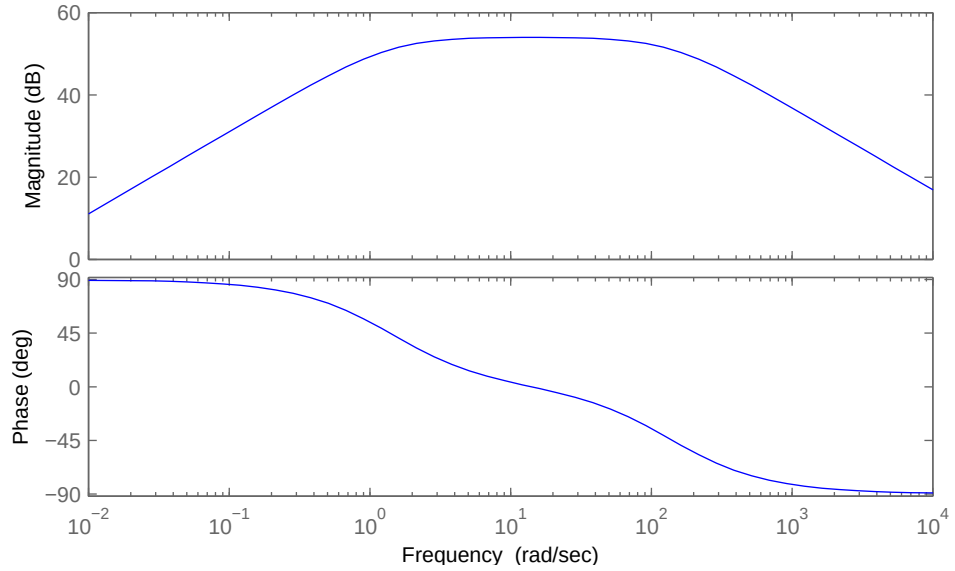


Figure 4.18: *Bode plot of the drive train filter, with* $\omega_n = 14\ rad/s$, $G_{DT} = 500$ *and* $\zeta = 5$.

The filter is included in the system as showed in Figure 4.19. It is seen that the input to the filter is the generator speed and the output is added to the generator torque.

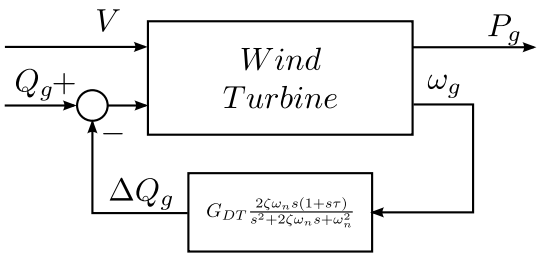


Figure 4.19: *Diagram of how the drive train filter is included in the system.*

To test the performance of the standard controller with drive train damping compared to the standard controller, both are tested with a wind speed with 16% turbulence intensity and mean wind speed 15 m/s. The result is seen in Figure 4.20 which shows a plot of the simulation and Figure 4.21 showing the performance in percent compared to the standard controller.

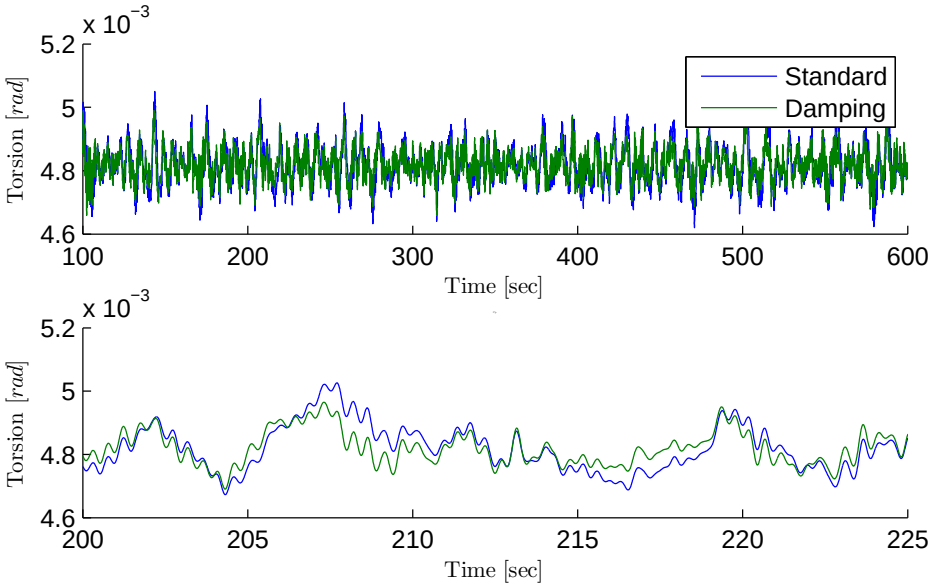


Figure 4.20: *Time series plot of the drive train with and without damping. The bottom figure is a slice of the top figure.*

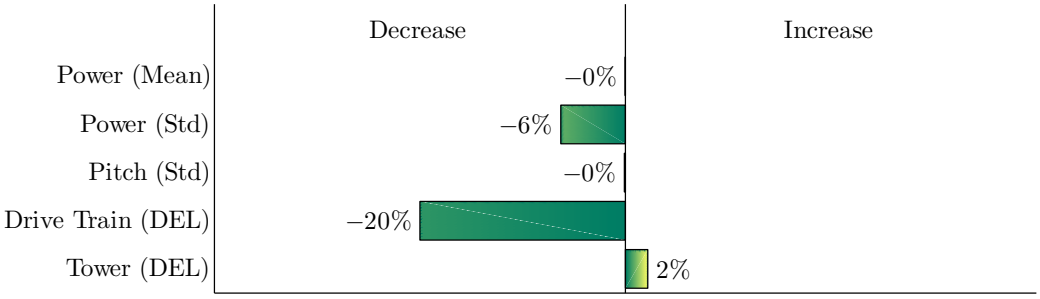


Figure 4.21: *Standard controller including the drive train damping compared with the standard controller as reference at a wind turbulence intensity of 16%.*

4.3.6 Tower and Drive Train Damping

In the previous sections the tower and drive train damping are individually tested against the standard controller. In this test both the tower and drive train are included to the standard controller. The result is a controller that is used as benchmark for the model predictive controller. How the benchmark controller performs compared to the standard controller is seen in Figure 4.22.

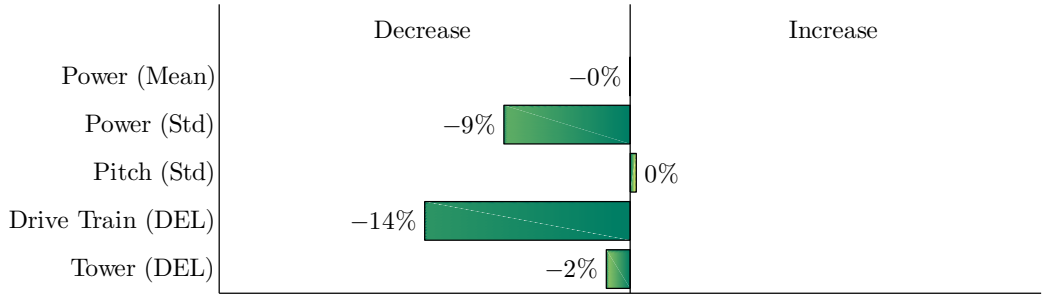


Figure 4.22: *Standard controller with tower and drive train damping compared with the standard controller as reference, simulated with a wind turbulence intensity of 16%.*

The standard PI controller including tower damping and drive train damping is seen to perform better than the standard controller alone and is the one chosen to serve as the benchmark for later developed model predictive controllers.

4.4 Model Predictive Control

In this section the general theory and setup of a model predictive controller (MPC) is described. The MPC is derived with the inclusion of LIDAR measurements. Before the implementation of the MPC is described.

The MPC controller is set up like and ordinary feedback controller, as takes the outputs from the plant $y[k]$, and sends it back to the input of the plant $u[k]$, as shown in Figure 4.23.

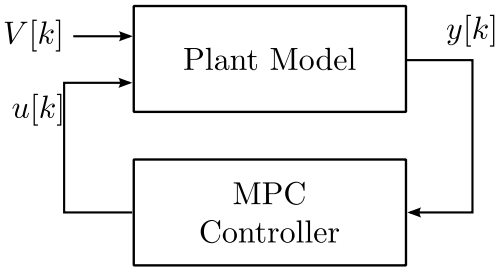


Figure 4.23: *Basic MPC setup*

Described in Chapter 3 the inputs to the plant model are the rotor average wind speed $V[k]$, the pitch angle β and the generator torque Q_g. Of the three inputs only two of them are controllable, β and Q_g, so the developed model is rearranged as in Figure 4.24.

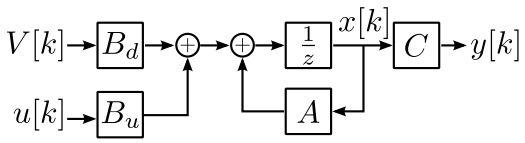


Figure 4.24: *Rearranging of the model, the wind $V[k]$ is seen as a disturbance and is separated from the controlled input $u[k]$*

A and C are the model matrixes from equation (3.42) and the input matrix B_u and B_d are given by the input matrix B in (3.42) such that

$$B = \begin{bmatrix} B_u & B_d \end{bmatrix} \tag{4.38}$$

Rearranging the model makes the state-space equation look like

$$\begin{aligned} x[k+1] &= Ax[k] + B_u u[k] + B_d V[k] \\ y[k] &= C_y x[k] \end{aligned} \tag{4.39}$$

where the wind $V[k]$ is seen as a disturbance, B_d determines how it affects the states and y is the output from the model.

MPC utilizes the model of the plant to predict the states for the near future. A plant model is therefore needed for the controller, and the MPC is applied with the derived state-space model shown in Equation (4.39).

4.4.1 Receding Horizons

The basic idea in MPC is that the controller is given the current states of the system and a setpoint for each output. Within a time horizon called the prediction horizon, the controller predicts the response of the system for a given set of input changes. The number of input changes is called the control horizon. The goal of the controller is find the optimal input changes that gives the optimal output. This is done by solving a cost function on the states and choosing the future inputs that gives the least cost.

Figure 4.25 illustrates the principal in MPC, where the controller at the time k, uses the states at time k to estimate the model output over a horizon. Two predictions are made by the MPC. The first is the free response $\hat{y}_f(t)$, which is the output of the model if no input changes are made. The second prediction is part of a optimization process where the MPC predicts the optimal input $u[k]$, that will give the output, $\hat{y}(t)$, which is optimized so the output is moved to the setpoint, $s(t)$. In some cases a reference signal $r(t)$, can be introduced. This is used if the outputs should try to follow at predetermined path to the setpoint. The reference signal is not implemented in this thesis, and will not be used in the following descriptions. Instead the MPC is made so $s(t)$ and $r(t)$ are the same and the outputs are driven to the setpoint as fast as possible.

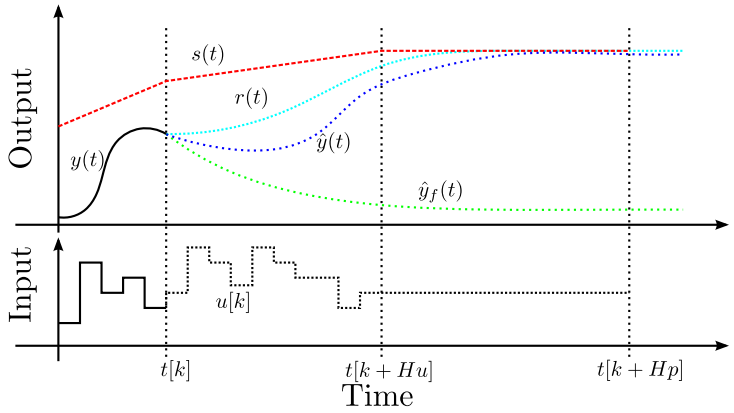


Figure 4.25: *The principal in prediction, from the states at time k the MPC predicts the future output of the plant. MPC predictes the free response of the plant $\hat{y}_f(t)$ which is the response if there in made no input changes. From the free response the MPC predicts the optimal input $u[k]$ as will make the optimal response $\hat{y}(t)$ be as close to the reference $r(t)$.*

The term receding horizon comes from that the prediction and control horizon moves along with the time, so when the time changes from $[k]$ to $[k+1]$ the prediction horizon moves to $[k+1+H_p]$, so it is always H_p long. The control horizon moves in the same manor, just with H_u as the horizon length.

The length of the prediction and the control horizons are tuning parameters for the controller and are described in section 4.4.7.

The MPC predicts and optimizes with H_u input changes, but only the first input change $u[k]$ is applied to the plant. Then the prediction and optimization is performed again at the next time step $[k+1]$ the first input $u[k+1]$ is chosen as the next input and so it continues.

4.4.2 Cost Function

As briefly mentioned in the previous section, the MPC controller performs an optimization on the predicted output of the plant. For the optimization a cost function is defined where the different outputs and control inputs can be weighted to decide the objectives of the controller. The cost function is given by

$$\mathcal{J}[k] = \sum_{i=1}^{H_p} \|\hat{y}[k+i|k] - r[k+i]\|^2_{Q(i)} + \sum_{i=0}^{H_u-1} \|\Delta\hat{u}[k+i|k]\|^2_{\mathcal{R}(i)}\,, \tag{4.40}$$

where $\hat{y}[k+i|k]$ is the predicted output from the plant, $r[k+i]$ is the reference for the output and $\Delta\hat{u}[k+i|k]$ is the predicted control input. $(\hat{y}[k+i|k] - r[k+i])$ can be seen as the tracking error. It is this error that is weighted and minimized in the optimization. The weight on the tracking error is determined by Q. The last part of the cost function is the calculation of the cost for the control input, which is weighted by $\mathcal{R}$.

The weights Q and $\mathcal{R}$ are tuning parameters for the controller and can be used to determine the objectives for the controller. The tuning of the weights are described in section 4.4.7.

4.4.3 MPC Controller Design

As described in the previous section, the MPC controller uses a prediction of the future outputs to give the optimal input changes. This gives the two primary functions of a MPC controller which are prediction and optimization. In this thesis it is proposed to use a LIDAR to measure the future wind for the turbine, this is included in the MPC controller as a measured disturbance and is used in the prediction to make a more accurate estimation of the future plant outputs. The LIDAR setup is described in section 3.11 and the LIDAR model is included as a part of the plant where the described buffer function is introduced as a part of the controller. The leads to the controller design shown in Figure 4.26.

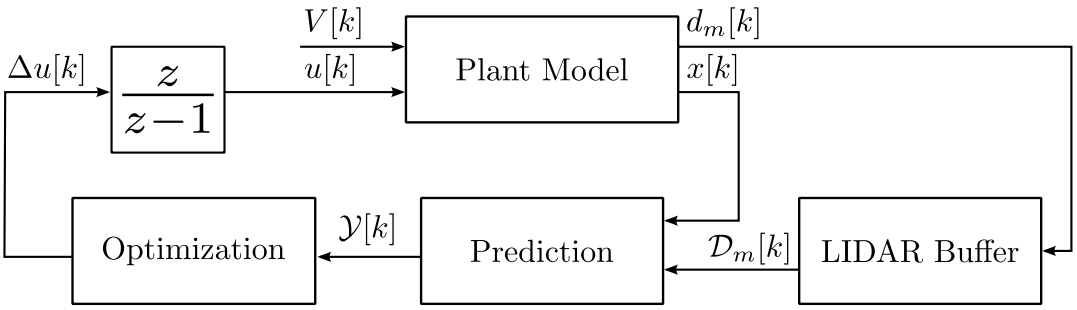


Figure 4.26: *The structure of the MPC controller is split into the three functions in the controller design. The LIDAR buffer handles the LIDAR measurements, the prediction estimates the output over the prediction horizon and the optimization gives the optimal input. The MPC naturally gives the output change instead of the real output so an integrator is placed between the controller and the plant.*

The LIDAR model is included in the plant model and gives the output, $d_m[k]$ which is the measured wind included in the controller as a measured disturbance. The MPC controller is split into three parts. The first part is the LIDAR buffer which stores the LIDAR measurements and gives $\mathcal{D}_m[k]$ which is the $d_m[k]$ values delayed so they cover the prediction horizon. The second part is the prediction that uses the states $x[k]$, and the measured disturbances $\mathcal{D}_m[k]$ to predict the outputs over the prediction horizon $\mathcal{Y}[k]$. The third and last part is the cost optimization that predicts the optimal input changes $u[k]$.

In this basic controller design it is assumed that there is full state measurements and thus that there is no need for an observer, which otherwise would have been needed for estimating the states of the plant model. This is not the case when the controller is implemented in LACflex, where not all states can be measured, and an observer is added for estimating the states. This is described in section 5.4.

From the cost function it is seen that the MPC controller optimizes on the control change $\Delta u[k]$ instead of the input value $u[k]$. Therefore an integrator is added between the controller and the plant so for the rest of the MPC description the Plant takes $\Delta u[k]$ as input.

For the model to fit the new definition, the real input at time k can be found as

$$u[k] = u[k-1] + \Delta u[k] \tag{4.41}$$

which will change the state space formulation from equation (4.39) to

$$\begin{aligned} x[k+1] &= Ax[k] + B_u u[k-1] + B_u \Delta u[k] + B_d V[k] \\ y[k] &= C_y x[k] \end{aligned} \tag{4.42}$$

which is the model used in the rest of the MPC description.

4.4.4 Prediction Model

The MPC controller uses the linear model equation (4.42), to predict the plant states and output over the prediction horizon. In the prediction the model uses the measured disturbance d_m, instead of the the wind which is used in the plant model. The measured disturbance is the LIDAR measurements of the wind, which have to be delayed for use in the model. This makes the state estimation for the prediction model to look like

$$\hat{x}[k+1|k] = Ax[k] + B_u u[k-1] + B_u \Delta\hat{u}[k|k] + B_d d_m[k-H_p]. \tag{4.43}$$

This can then be used to predict the states by propagating the model over the prediction horizon, where the second step can be estimated by

$$\hat{x}[k+2|k] = A\hat{x}[k+1|k] + B_u\hat{u}[k|k] + B_u\Delta\hat{u}[k+1|k] + B_d d_m[k-H_p+1] \tag{4.44}$$

$$\begin{aligned} \hat{x}[k+2|k] =& A^2 x[k] + (AB_u + B_u)\, u[k-1] + (AB_u + B_u)\, \Delta\hat{u}[k|k] + B_u\Delta\hat{u}[k+1|k] \\ &+ AB_d d_m[k-H_p] + B_d d_m[k-H_p+1] \end{aligned} \tag{4.45}$$

and the third step can be estimated by

$$\hat{x}[k+3|k] = A\hat{x}[k+2|k] + B_u\hat{u}[k+1|k] + B_u\Delta\hat{u}[k+2|k] + B_d d_m[k-H_p+2] \tag{4.46}$$

$$\begin{aligned} \hat{x}[k+3|k] =& A^3 x[k] + \left(A^2B_u + AB_u + B_u\right) u[k-1] \\ &+ \left(A^2B_u + AB_u + B_u\right) \Delta\hat{u}[k|k] + (AB_u + B_u)\, \Delta\hat{u}[k+1|k] + B_u\Delta\hat{u}[k+2|k] \\ &+ A^2 B_d d_m[k-H_p] + AB_d d_m[k-H_p+1] + B_d d_m[k-H_p+2]. \end{aligned} \tag{4.47}$$

The estimation is continued over the entire prediction horizon. The prediction of the plant states can be divided into three groups, based on where the contribution comes from. The first contribution is based on known values $u[k-1]$ and $x[k]$, there contribution can be expressed on matrix

form like

$$\mathcal{X}_{known}[k] = \underbrace{\begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{Hp} \end{bmatrix}}_{\Psi_x} x[k] + \underbrace{\begin{bmatrix} B_u \\ AB_u + B_u \\ A^2B_u + AB_u + B_u \\ \vdots \\ \sum_{i=0}^{Hp-1} A^i B_u \end{bmatrix}}_{\Upsilon_x} u[k-1] \tag{4.48}$$

The second contribution comes from the predicted input moves, which are given by

$$\mathcal{X}_{predicted}[k] = \underbrace{\begin{bmatrix} B_u & 0 & 0 & \cdots & 0 \\ AB_u + B_u & B_u & 0 & \cdots & 0 \\ A^2 + AB_u + B_u & AB_u + B_u & B_u & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{Hp-1} A^i B_u & \sum_{i=0}^{Hp-2} A^i B_u & \cdots & \cdots & \sum_{i=0}^{Hp-Hu} A^i B_u \end{bmatrix}}_{\Theta_x} \underbrace{\begin{bmatrix} \Delta\hat{u}[k|k] \\ \Delta\hat{u}[k+1|k] \\ \Delta\hat{u}[k+2|k] \\ \vdots \\ \Delta\hat{u}[k+Hu-1|k] \end{bmatrix}}_{\Delta\mathcal{U}[k]} \tag{4.49}$$

The contribution from the past and the predicted would give the prediction where there are not used a LIDAR to provide the measured disturbance. The prediction can be found by

$$\mathcal{X}[k] = \begin{bmatrix} \hat{x}[k+1] \\ \hat{x}[k+2] \\ \hat{x}[k+3] \\ \vdots \\ \hat{x}[k+Hp] \end{bmatrix} = \underbrace{\Psi_x x[k] + \Upsilon_x u[k-1]}_{\mathcal{X}_{known}[k]} + \underbrace{\Theta_x \Delta\mathcal{U}[k]}_{\mathcal{X}_{predicted}[k]}. \tag{4.50}$$

In this MPC there is included measurements of the future wind by a LIDAR. The future wind is included in the prediction as a measured disturbance. The measured disturbance gives the contribution

$$\mathcal{X}_{dm}[k] = \underbrace{\begin{bmatrix} B_d & 0 & 0 & \cdots & 0 \\ AB_d & B_d & 0 & \cdots & 0 \\ A^2B_d & AB_d & B_d & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{(Hp-1)}B_d & A^{(Hp-2)}B_d & A^{(Hp-3)}B_d & \cdots & B_d \end{bmatrix}}_{\Xi_x} \underbrace{\begin{bmatrix} d_m[k-H_p] \\ d_m(k-H_p+1) \\ d_m(k-H_p+2) \\ \vdots \\ d_m(k) \end{bmatrix}}_{\mathcal{D}[k]} \tag{4.51}$$

The MPC prediction of the states can now be written as

$$\mathcal{X}[k] = \begin{bmatrix} \hat{x}[k+1] \\ \hat{x}[k+2] \\ \hat{x}[k+3] \\ \vdots \\ \hat{x}[k+Hp] \end{bmatrix} = \underbrace{\Psi_x x[k] + \Upsilon_x u[k-1]}_{\mathcal{X}_{known}[k]} + \underbrace{\Theta_x \Delta\mathcal{U}[k]}_{\mathcal{X}_{predicted}[k]} + \underbrace{\Xi_x \mathcal{D}[k]}_{\mathcal{X}_{dm}[k]}. \tag{4.52}$$

In the implementation $\mathcal{D}[k]$ is the output from the LIDAR buffer function, where the LIDAR measurements are stored so they fit into the prediction mode.

The prediction of the states could be changed to predict y. Instead of of predicting y the controlled output z, is introduced. z is given as the output of the prediction model (4.43) like

$$\hat{z}[k] = C_z x[k] + D_{u,z} u[k-1] + D_{u,z} \Delta\hat{u}[k|k].$$

where C_z is the output matrix for the controlled output and $D_{u,z}$ are the direct feedtrough from the input to the controlled output.

In the prediction it is often assumed that there is no direct influence from the input to the output so the D term from the equations is omitted. The case in this thesis a direct term from the control input to the controlled output is needed, because of the chosen control output described in section 4.4.7, so the direct term is added to the prediction. The direct term is added as $D_{u,z}$ and $D_{\Delta u,z}$, to keep the notation for the prediction, which makes the prediction of the controlled output look like.

$$\mathcal{Z}[k] = C_z \mathcal{X}[k] + \mathcal{D}_{u,z} u[k-1] + \mathcal{D}_{\Delta u,z} \Delta\mathcal{U}[k] \tag{4.53}$$

where $\mathcal{D}_{u,z}$ gives the direct feed trough from the last input and $\mathcal{D}_{\Delta u,z}$ handles it for the predicted input changes. Adding the direct term to the prediction yields the prediction matrixes as

$$\underbrace{\begin{bmatrix} D_{u,z} \\ D_{u,z} \\ \vdots \\ D_{u,z} \end{bmatrix}}_{\mathcal{D}_{u,z}} u[k-1] + \underbrace{\begin{bmatrix} D_{u,z} & D_{u,z} & 0 & 0 & 0 & \dots & 0 \\ D_{u,z} & D_{u,z} & D_{u,z} & 0 & 0 & \dots & 0 \\ D_{u,z} & D_{u,z} & D_{u,z} & D_{u,z} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ D_{u,z} & D_{u,z} & D_{u,z} & D_{u,z} & D_{u,z} & \dots & D_{u,z} \end{bmatrix}}_{\mathcal{D}_{\Delta u,z}} \underbrace{\begin{bmatrix} \Delta\hat{u}[k|k] \\ \Delta\hat{u}[k+1|k] \\ \Delta\hat{u}[k+2|k] \\ \vdots \\ \Delta\hat{u}[k+Hu-1|k] \end{bmatrix}}_{\Delta\mathcal{U}[k]} \tag{4.54}$$

where $D_{u,z}$ is the direct term from the control input, u, to the controlled output z. The implemented control output is described in section 4.4.7. This makes the final prediction of the control output look like

$$\mathcal{Z}[k] = C_z \Psi_x x[k] + C_z \left(\Upsilon_x + D_{u,z}\right) u[k-1] + C_z \left(\Theta_x + D_{u,z}\right) \Delta\mathcal{U}[k] + \Xi_x \mathcal{D}[k] \tag{4.55}$$

Defining

$$\Psi = C_z \Psi_x \tag{4.56}$$
$$\Upsilon = C_z (\Upsilon_x + D_{u,z}) \tag{4.57}$$
$$\Theta = C_z (\Theta_x + D_{u,z}) \tag{4.58}$$

the prediction can be written as

$$\mathcal{Z}[k] = \Psi x[k] + \Upsilon u[k-1] + \Theta \Delta \mathcal{U}[k] + \Xi \mathcal{D}[k]. \tag{4.59}$$

This leads to a controller setup as shown in Figure 4.27 which is the one used for the simulations in MATLAB and LACflex. Additional an observer described in section 5.4 is added for simulations in LACflex as not all states are measured.

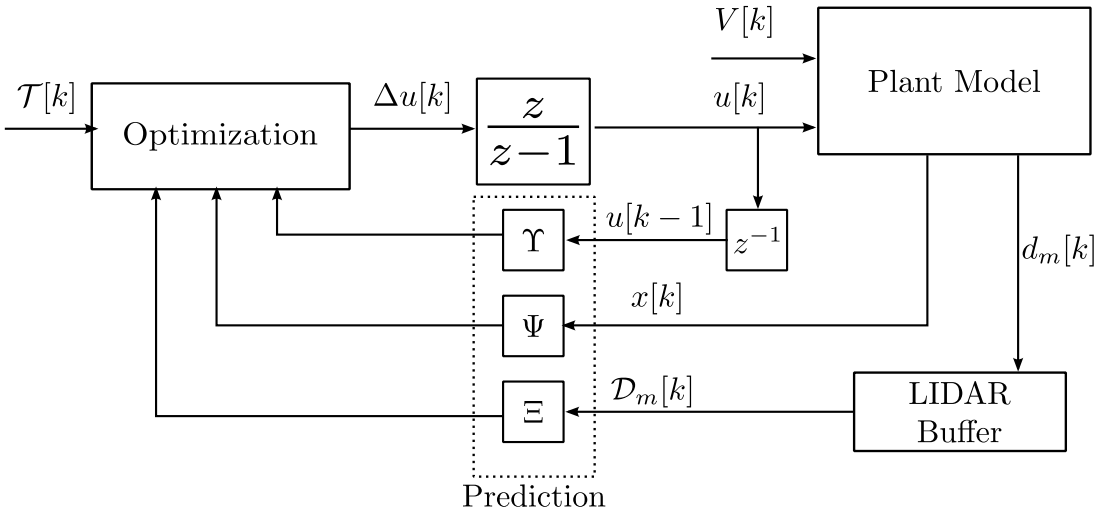


Figure 4.27: *Controller implementation*

4.4.5 Constraints

In MPC constraints can be implemented for the control input, control rate and the controlled output. The constraints has to be formulated as

$$E \begin{bmatrix} \Delta \mathcal{U}[k] \\ 1 \end{bmatrix} \leq 0 \qquad F \begin{bmatrix} \mathcal{U}[k] \\ 1 \end{bmatrix} \leq 0 \qquad G \begin{bmatrix} \mathcal{Z}[k] \\ 1 \end{bmatrix} \leq 0 \tag{4.60}$$

where the control input, control rate and the controlled output are constrained by E, F and G respectively. For the use in the MPC optimization all the constraints have to be written into linear inequalities for Δu.

The constraints E on the control rate $\Delta\mathcal{U}$ can be written as

$$\underbrace{\begin{bmatrix} \frac{1}{\Delta u_{min,k}} & 0 & \dots & 0 & -1 \\ \frac{1}{\Delta u_{max,k}} & 0 & \dots & 0 & -1 \\ 0 & \frac{1}{\Delta u_{min,k+1}} & \dots & 0 & -1 \\ 0 & \frac{1}{\Delta u_{max,k+1}} & \dots & 0 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{1}{\Delta u_{min,k+Hu-1}} & -1 \\ 0 & 0 & \dots & \frac{1}{\Delta u_{max,k+Hu-1}} & -1 \end{bmatrix}}_{E} \begin{bmatrix} \Delta\hat{u}[k|k] \\ \Delta\hat{u}[k+1|k] \\ \vdots \\ \Delta\hat{u}[k+H_u-1|k] \\ \\ 1 \end{bmatrix} \leq 0 \tag{4.61}$$

where $\Delta u_{min,k}$ and $\Delta u_{max,k}$ are the minimum and maximum constraints for $\Delta u[k]$. E is structured such that the constraints are applied for all the control moves $\Delta\mathcal{U}$ in the control horizon. The constraints in F and G are structured the same way as E.

The constraints F on the control input $\mathcal{U}$ have to be rearranged so they depend on $\Delta\mathcal{U}$ for the use in the MPC optimization. Suppose that F has the form

$$F = [F_1, F_2, \dots, F_{H_u}, f] \tag{4.62}$$

and the inequalities can be written as

$$\begin{aligned} &\sum_{j=1}^{H_u} F_j \Delta u[k|k] + \sum_{j=2}^{H_u} F_j \Delta u[k+1|k] + \ldots + F_{H_u} \Delta u[k+H_u-1|k] \\ &+ \sum_{j=1}^{H_u} F_j u[k-1] + f \leq 0. \end{aligned} \tag{4.63}$$

Defining

$$\mathcal{F}_i = \sum_{j=i}^{H_u} F_j \tag{4.64}$$

and

$$\mathcal{F} = \begin{bmatrix} \mathcal{F}_1 & \mathcal{F}_2 & \dots & \mathcal{F}_{H_u} \end{bmatrix} \tag{4.65}$$

the inequalities are then given by

$$F\Delta\mathcal{U}[k] \leq -\mathcal{F}_1 u[k-1] - f \tag{4.66}$$

The last set of constraints G are on the controlled output $\mathcal{Z}$. Like for the control input the inequalities have to be reformulated as constraints for $\Delta\mathcal{U}$. This is done by inserting the prediction model from equation (4.59) into the constraints in equation (4.60) which gives

$$G \begin{bmatrix} \Psi x[k] + \Upsilon u[k-1] + \Theta\Delta\mathcal{U}[k] + \Xi\mathcal{D}[k] \\ 1 \end{bmatrix} \leq 0. \tag{4.67}$$

Setting $G = [\Gamma, g]$ where g is the last column of G this can be written as

$$\Gamma(\Psi x[k] + \Upsilon u[k-1] + \Xi \mathcal{D}[k]) + \Gamma\Theta\Delta\mathcal{U}[k] + g \leq 0. \tag{4.68}$$

With all three sets of constrains reformulated as inequalities of $\Delta\mathcal{U}$ they can be combined into one single inequality

$$\underbrace{\begin{bmatrix} \mathcal{F} \\ \Gamma\Theta \\ \mathcal{W} \end{bmatrix}}_{\Omega} \Delta\mathcal{U} \leq \underbrace{\begin{bmatrix} -\mathcal{F}_1 u[k-1] - f \\ \Gamma(\Psi x[k] + \Upsilon u[k-1] + \Xi\mathcal{D}[k]) - g \\ w \end{bmatrix}}_{\omega}. \tag{4.69}$$

where Ω and ω are known at the time k.

4.4.6 Solving the Optimization Problem

The optimization problem takes the predicted controlled output from the prediction model and finds the optimal input changes so the predicted output is driven towards the reference. The cost function given in equation (4.40) is reformulated to use the derived prediction model in equation (4.59) and looks like

$$\mathcal{J}[k] = \|\mathcal{Z}[k] - \mathcal{T}[k]\|_Q^2 + \|\Delta\mathcal{U}[k]\|_R^2 \tag{4.70}$$

where

$$\mathcal{Z}[k] = \begin{bmatrix} z[k+1|k] \\ \vdots \\ \hat{z}[k+H_p|k] \end{bmatrix} \quad \mathcal{T}[k] = \begin{bmatrix} r[k+1|k] \\ \vdots \\ \hat{r}[k+H_p|k] \end{bmatrix} \quad \Delta\mathcal{U}[k] = \begin{bmatrix} \Delta u[k|k] \\ \vdots \\ \Delta\hat{u}[k+H_u-1|k] \end{bmatrix} \tag{4.71}$$

The tracking error in the prediction can be defined as

$$\varepsilon[k] = \mathcal{T}[k] - (\Psi\hat{x}[k] + \Upsilon u[k-1] + \Xi\mathcal{D}[k]) \tag{4.72}$$

which gives the difference between the reference and the predicted output if there are no changes to the input. The cost function can be reformulated as

$$\mathcal{J}[k] = \|\Theta\Delta\mathcal{U}[k] - \varepsilon[k]\|_Q^2 + \|\Delta\mathcal{U}[k]\|_{\mathcal{R}}^2 \tag{4.73}$$

$$\mathcal{J}[k] = (\Theta\Delta\mathcal{U}[k] - \varepsilon[k])^T \mathcal{Q}(\Theta\Delta\mathcal{U}[k] - \varepsilon[k]) + \Delta\mathcal{U}[k]^T\mathcal{R}\Delta\mathcal{U}[k] \tag{4.74}$$

$$\mathcal{J}[k] = \Delta\mathcal{U}[k]^T\left(\Theta^T\mathcal{Q}\Theta + \mathcal{R}\right)\Delta\mathcal{U}[k] - 2\Delta\mathcal{U}[k]^T\Theta^T\mathcal{Q}\varepsilon[k] + \varepsilon[k]^T\mathcal{Q}\varepsilon[k]. \tag{4.75}$$

This can be written as

$$\mathcal{J}[k] = \Delta\mathcal{U}[k]^T\mathcal{H}\Delta\mathcal{U}[k] - \mathcal{G}^T\Delta\mathcal{U}[k] + const[k] \tag{4.76}$$

where

$$\begin{aligned} \mathcal{G}[k] =& 2\Theta^T Q\varepsilon[k] \\ \mathcal{H}[k] =& \Theta^T Q\Theta + \mathcal{R} \\ const[k] =& \varepsilon[k]^T Q\varepsilon[k] \end{aligned}$$

The cost function in equation (4.76) is optimized subject to the inequalities in equation (4.69). The cost function is a Quadratic programming problem which is a standard optimization problem given as

$$min \frac{1}{2}\Theta^T \Phi\Theta + \phi^T \Theta$$

subject to

$$\Omega\Theta \leq \omega.$$

When the cost function is inserted into the optimization problem it looks like

$$minimize \qquad \Delta\mathcal{U}[k]^T \mathcal{H}\Delta\mathcal{U}[k] - \mathcal{G}^T \Delta\mathcal{U}[k]$$

subject to

$$\begin{bmatrix} \mathcal{F} \\ \Gamma\Theta \\ \mathcal{W} \end{bmatrix} \Delta\mathcal{Q} \leq \begin{bmatrix} -\mathcal{F}_1 u[k-1] - f \\ \Gamma(\Psi x[k] + \Upsilon u[k-1] + \Xi\mathcal{D}[k]) - g \\ w \end{bmatrix}$$

4.4.7 Implementation of the MPC Controller

Prediction and Control Horizons

The prediction and control horizons have to be determined. In an attempt to keep the system within all constraints, the prediction horizon have to be longer than it takes for the system to find steady state. To determine this, the step response for the system is analyzed and it is found that the model reaches steady state around 15 seconds after a step, so the prediction horizon is chosen to be 20 seconds. The control horizon is at this point chosen to be 10 seconds. The choice of the control horizon is made from simulation with different values, and 10 seconds was chosen as are reasonable trade off between performance and computational time.

Control Output

In the MPC prediction it was introduced that the MPC controller predicted the controlled output z. The goal of the controller as given in the problem statement, is to minimize fatigue while still producing maximum power and maintaining power quality. From these goals the controlled output are chosen.

The power produced and the quality of it, is determined by the generator speed and the applied

generator torque, described in section 3.6, so the generator speed ω_g, and generator torque Q_g, is included in the controlled output. The goal of minimizing fatigue on the drive train and tower can be addressed by the assumption that the fatigue has a strong connection to the dynamic and thus the oscillations of the drive train and tower. For this reason the tower velocity $\dot{Y}_t$, and the change in torsion of the drive train $\dot{\Theta}$, is chosen as controlled output as well.

It is chosen only to implement constraints on the actuators why the pitch angle β, pitch change rate $\dot{\beta}$, generator torque Q_g and generator change rate $\dot{Q}_g$ are added to the controlled outputs.

The controlled outputs are then

$$z = \begin{bmatrix} \omega_g \\ \dot{\Theta} \\ \dot{Y}_t \\ Q_g \\ \dot{Q}_g \\ \beta \\ \dot{\beta} \end{bmatrix} \tag{4.77}$$

Weights

The weights Q and $\mathcal{R}$ will not be described in detail, but the general properties of the tuning process is described here.

The values in Q decides the priority of which of the outputs that are the most important as there will be a trade off between them, The trade off can be explained by the number of control inputs that are available and the number of outputs that there are weight on. If there are more output weights then there are inputs there is no guarantee that all outputs can be at the reference [19].

The values in $\mathcal{R}$ have the same function as the values in Q except that it is the input that are prioritized. In this thesis where the control input are the pitch angle and the generator torque, $\mathcal{R}$ decides which input will be used to control the output.

The relative size of Q and $\mathcal{R}$ determines the overall amount of control that are used to reach the set points. If $\mathcal{R}$ is very small relative to Q the controller will do a lot of control to reach the set points and if opposite that $\mathcal{R}$ is much larger then Q the controller will not use the controller as much.

For the actual values of the weights Brysons rule[17] was used as a start guess and then manually tunned to achieve the controller goals.

4.5 Test of the Developed MPC Controller

To test the controller, three different wind fields with a mean wind speed of $15 m/s$, seed A,B and C are generated in LACflex. The fields are generated with turbulence intensities of 14%, 16%

and 18%. The three wind signals at different intensities are seen in Figure 4.28, 4.29 and 4.30.

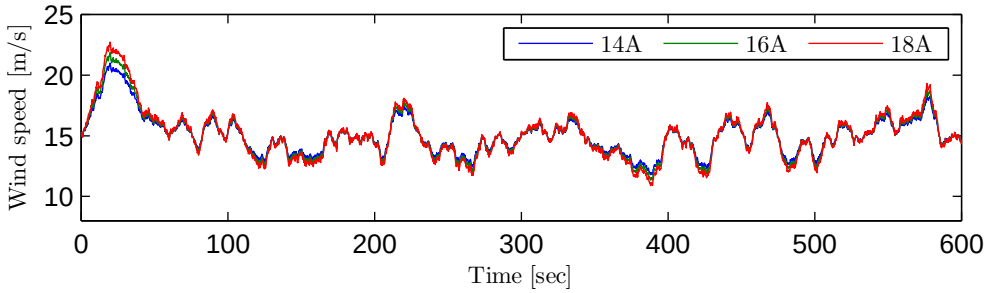


Figure 4.28: *Wind speed seed A with 14%, 16% and 18% turbulence intensity.*

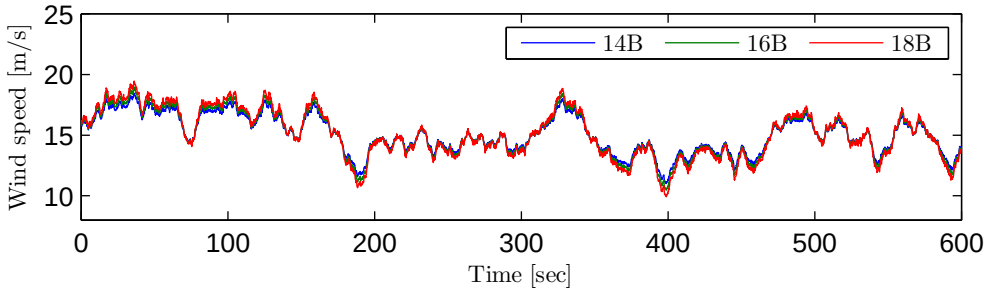


Figure 4.29: *Wind speed seed B with 14%, 16% and 18% turbulence intensity.*

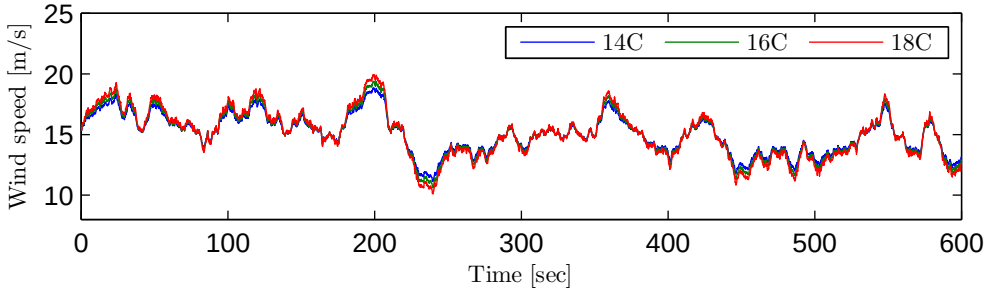


Figure 4.30: *Wind speed seed C with 14%, 16% and 18% turbulence intensity.*

600 seconds simulations with the different winds are carried out for the controller with different LIDAR models, The LIDAR models are:

- OFF: No LIDAR is used
- REAL: An average of the measurements generated with the LACflex LIDAR model, is used as the LIDAR signal.
- FILTER: The REAL signal from above is filtered with a low pass filter.
- IDEAL: The average rotor wind speed is used as the LIDAR signal.

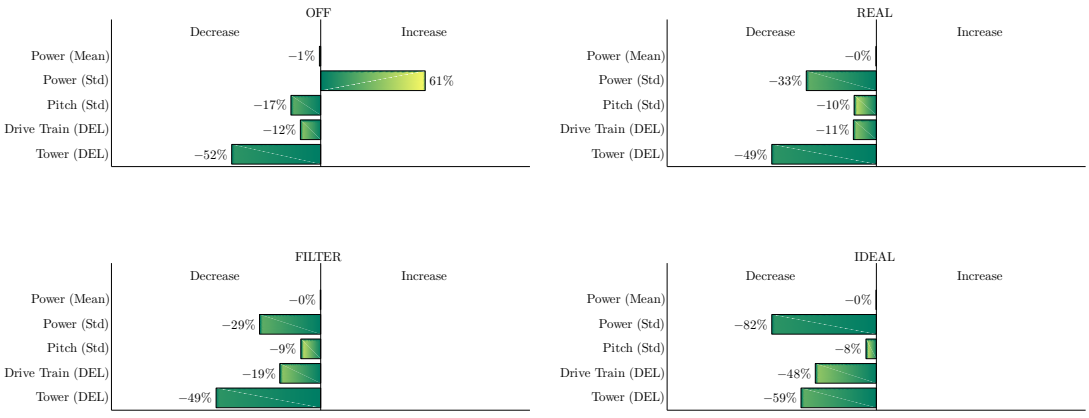
The data from the simulation is post processed. The mean and standard deviation for the power are calculated along with the damage equivalent load, calculated for the tower fore-aft mode and drive train torsion.

The results from seed A, at the turbulence intensity 14%, 16% and 18% are seen below. The Results from seed B and C are in Appendix A. They are similar to the results from seed A.

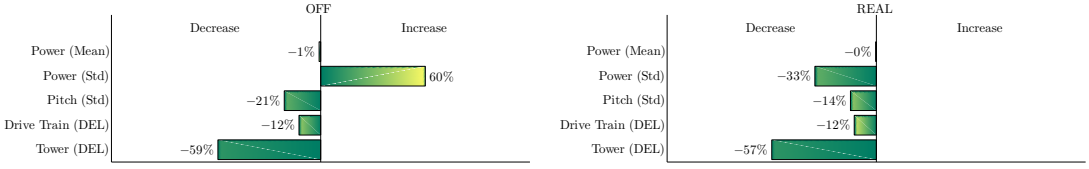
The results show how the MPC with the four different LIDAR setup performs compared to the benchmark controller, which as described in section 4.3 is a standard PI controller for power tracking with added damping on the tower and drive train. The benchmark controller serves as a benchmark for the damage level that has to be decreased. For testing if the LIDAR has a positive effect the OFF case where no LIDAR is used together with the model predictive controller is important. If there is a mitigation of the loads it could be because of the use of model predictive control, but by having both cases, with and without LIDAR the effect of the LIDAR is possible to test.

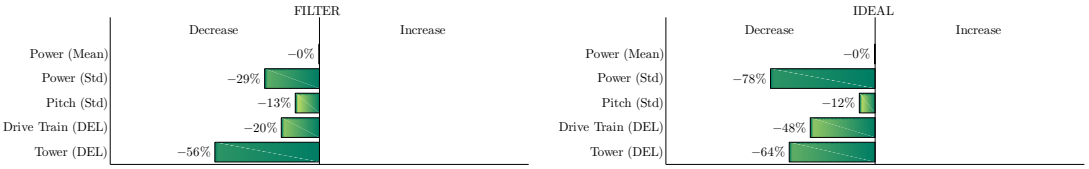
The simulation results are given in bar plots, showing in percent how the values either increase or decrease compared to the benchmark. The best possible result is if power mean is kept the same as for the PI controller and all the other values decreases as much as possible. The results from seed A for the three turbulence intensities are:

Seed A turbulence intensity 14%

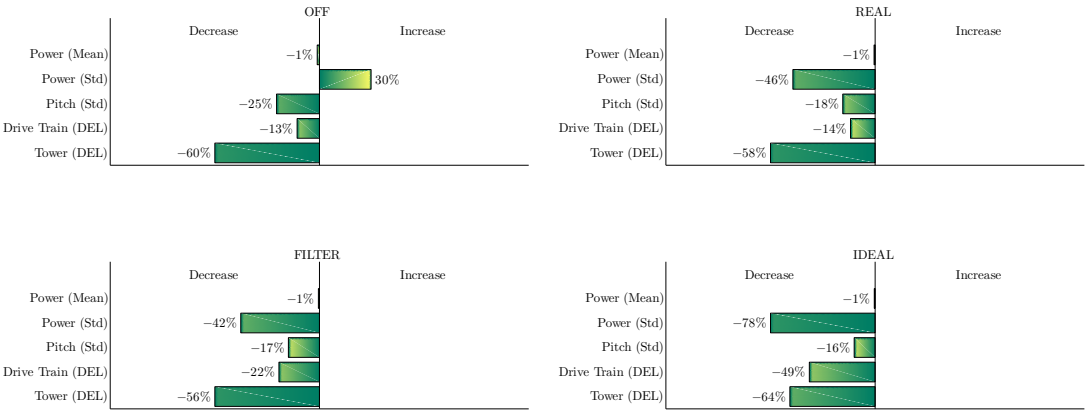


Seed A turbulence intensity 16%





Seed A turbulence intensity 18%



The results from the three different turbulence intensities for seed A, shows in general the damage of the tower and drive train are minimized. There is however a difference in the results depending on the used LIDAR model.

The OFF case, where no LIDAR is used, the damage of the tower and drive train are decreased, but there is an increase in the standard deviation of the power. As the turbulent intensity rises, the standard deviations for the power and pitch along with the damages for the tower and drive train decreases compared to the lower intensity.

The REAL case where a LIDAR signal is used, generated from an average of measurements from the LACflex LIDAR model. The results shows, as in the OFF case, when the turbulence rises the values decreases. The differences compared to the OFF case are, that the standard deviation for power is decreased where it for OFF was increased and the damages are not decreased as much as for the OFF case.

The FILTER case where REAL is filtered with a low pass filter shows almost the same as REAL. The tower is damped not quite as much and the standard deviation of the power is not as low but the drive train is however deceased more.

For the IDEAL case, where the LIDAR signal is the same as the wind signal, the damage of the tower and drive train and the standard deviation of the power and pitch are decreased more than for the other cases and as for the other cases they are decreased further as the intensity is increased.

Timeplots of the power, pitch, tower and drive train are seen below for seed A with a turbulence intensity of 16%. Figure 4.31 shows the power. It is seen that the power for benchmark and OFF varies more than the other cases which also the bar plot showed.

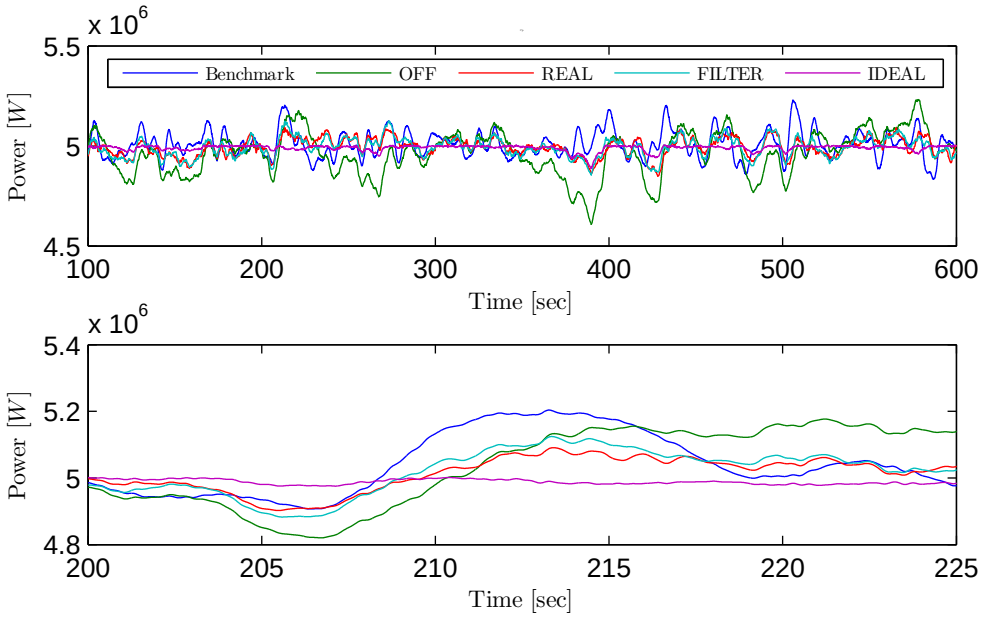


Figure 4.31: *Time plot of the power. The top plot is for the entire simulation time of 500 sec and the bottom plot is a close up on a part of the simulation*

Figure 4.32 shows the pitch. The pitch value is almost the same for all the controllers, except from benchmark that varies a little more.

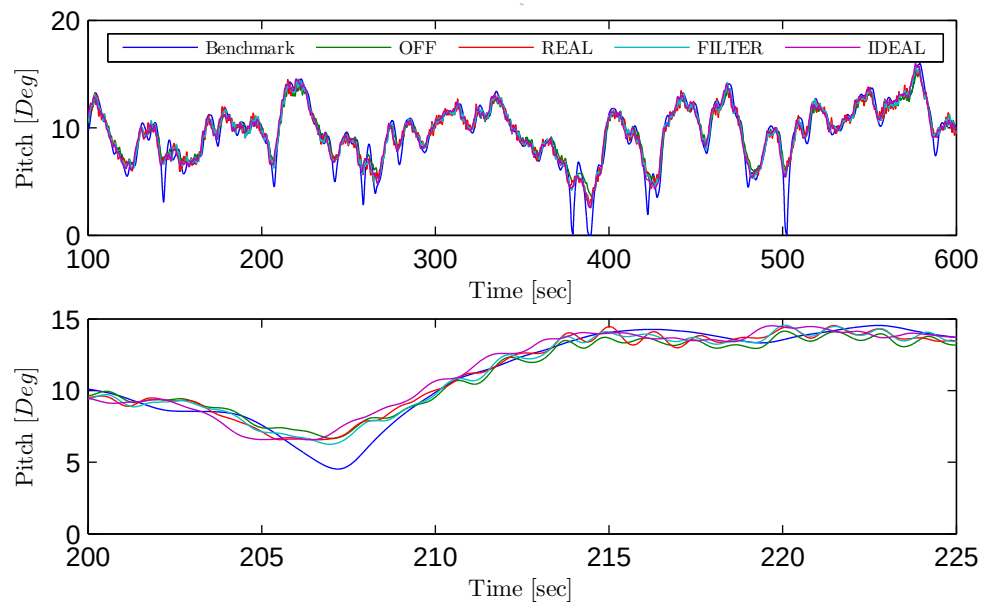


Figure 4.32: *Time plot of the pitch. The top plot is for the entire simulation time of 500 sec and the bottom plot is a close up on a part of the simulation*

Figure 4.33 shows the torsion of the drive train. Benchmark is the one with the largest variations.

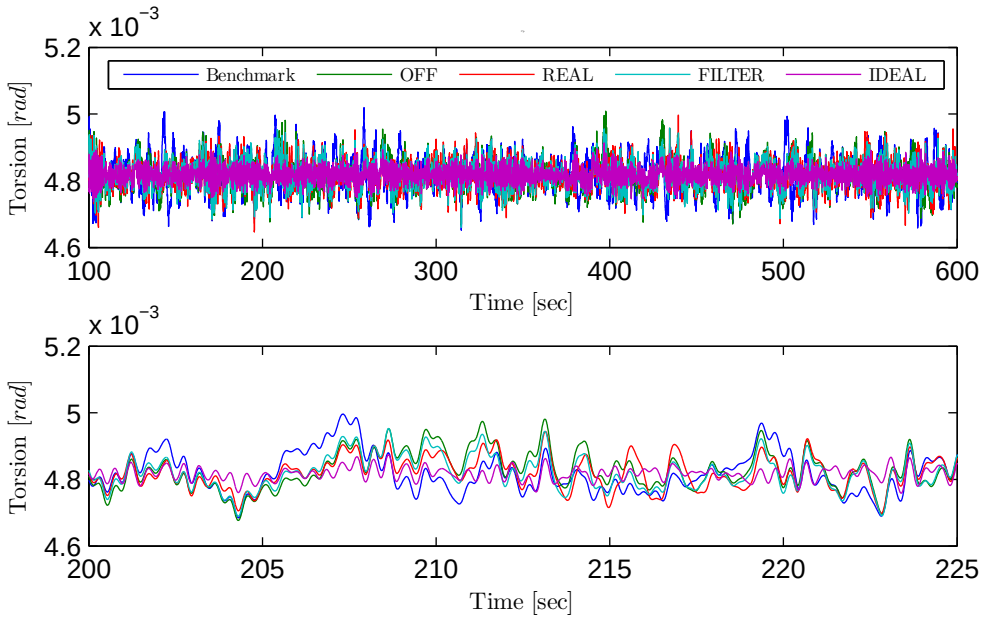


Figure 4.33: *Time plot of the drive train torsion. The top plot is for the entire simulation time of 500 sec and the bottom plot is a close up on a part of the simulation*

Figure 4.34 shows the tower fore-aft mode. Especially the benchmark stands out with large variations compared to the others.

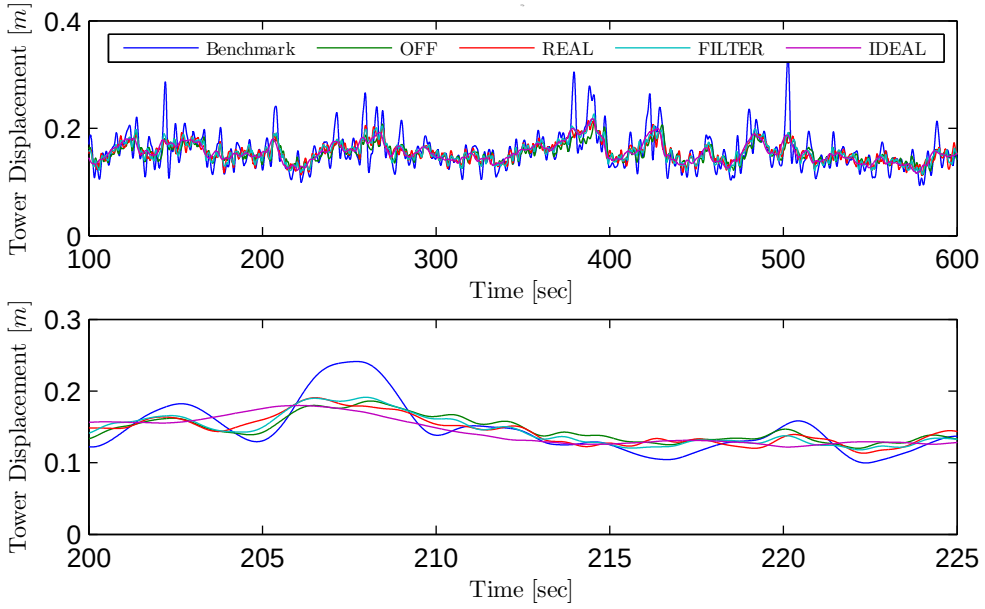


Figure 4.34: *Time plot of the tower fore-aft displacement.The top plot is for the entire simulation time of 500 sec and the bottom plot is a close up on a part of the simulation*

4.5.1 Discussion of the Results

The LIDAR model that performs best is the IDEAL case. This is not a surprise as the LIDAR signal is the same as the wind, and there is therefore no unknown disturbances in that case. Except from the IDEAL case it is judged that the FILTER is better than the REAL case and the OFF case where no LIDAR is used.

Before further discussion of the reason for choosing FILTER as the best LIDAR model, it is important to discuss the mean power. The goal for the controller was to minimize the fatigue on the tower and drive train but still maintain the same power quality. The results however showed a decrease of 1% in some cases which is unaccaptable as the main purpose for a turbine is to produce power. But by looking at the plot for the power, see Figure 4.31, it is seen that it is the benchmark controller that does not keep the rated power. The benchmark has several spikes above the rated power compared to the other controllers except from the OFF case. It is therefore judged that a one percent decrease in power mean compared to the benchmark controller is accaptable.

The controller without LIDAR, the OFF case, do minimize the fatigue but does not keep the standard deviation lower than the deviation for the benchmark, why it is considered not to perform as good as the other controllers.

The REAL and FILTER case are quite similar. The REAL case minimizes the tower and standard deviation a little more than the FILTER, but the FILTER lowers the drive train several percent more than the REAL does, why it is judged that the FILTER is best. If a more steady power and more damping on the tower is wanted REAL is the best one.

From the results it is seen that with a model predictive controller, the tower and drive train can be damped. It is however a trade OFF as the power variation increases. Using LIDAR not only removes this trade off but lowers the standard deviation and maintain the damping on tower and drive train.

The results show that by using model predictive control and LIDAR it has been possible to reduce the drive train DEL by approximately 20%, the tower DEL with 50% and the power standard deviation with 29%. Different tunings of the tower and drive train damping for the PI controller could perhabs reduce these values, but also different tunings of the MPC would change the results. The different wind signals and the intensities is also an important factor. Sometimes a tuning works perfect for one wind situation but for another it does the inverse.

Chapter 5

Simulation in LACflex

In this chapter is a description of the simulations in the aeroelastic code LACflex. This includes a description of the model in LACflex and a validation of the developed model against the LACflex model. As not all states are measured an observer is designed, before the model predictive controller is simulated in LACflex.

5.1 Introduction to Simulations in LACflex

The developed controller have been tested in MATLAB on the developed nonlinear turbine model. For a more realistic simulation the controller is tested on the more realistic turbine model in LACflex.

The turbine model in LACflex is different from the one used in MATLAB. It has not been possible to implement the NREL turbine, which is used for the development of the MPC, in LACflex as not enough data about the parameters was available. The simulations in LACflex are therefore done with a model for the LACflex turbine, which is a 2MW turbine with other dynamics than the 5MW used in MATLAB. Using a different turbine includes changing the tuning of the MPC. Changing the tuning and the use of two different turbine parameters, makes it difficult to make a complete comparison of the results from MATLAB and LACflex, it is however tried in the best possible way to compare the two results.

5.2 The Turbine Model in LACflex

The turbine model implemented in LACflex is a 2MW turbine, that is modeled with several degrees of freedom, that are not included in the developed model. The LACflex model includes the bending of the rotor blades, this is included for 1st and 2nd bending mode both flap and edgewise where in the developed model the blades are assumed rigid and does not represent the dynamics of the blades. For the tower LACflex also includes the 1st and 2nd bending model, here only the 1st mode is represented in the developed model. Furthermore the developed model only describes the fore-aft mode of the tower where LACflex also used the lateral movement of

the tower. For the drive train the torsion is represented in both models, but LACflex also includes the rotation of the drive train around all three axes were the developed model only give rotation around the x axis.

For using the developed linear model or controllers, the model parameters and the operational point have to be determined for the turbine used in LACflex. The parameters are extracted from LACflex and the operational point is determined by a simulation, with the in LACflex provided controller at the same wind speed as used for operating point for the developed model in MATLAB. The values for pitch, torque, generator speed and tower displacement, that are used as operating point are assumed to be brought to steady state, when using a constant wind by the controller.

5.3 Validation of Model in LACflex

Before any simulations are performed in LACflex with the derived controller, the nonlinear model, derived in Chapter 3, is validated. This is done to see if the derived model is a reasonable representation of the complex model in LACflex that have more degrees of freedom, and is more realistic than the derived nonlinear model. In the comparison the different parts of the derived model is compared to the different parts of the LACflex model, by taking the input for the submodels from the LACflex simulated data. All tests are performed with the same simulation data obtained from LACflex from a simulations of a simple laminar wind.

5.3.1 Inputs Used for Validation

For the validation simulation the inputs for the two models have to be the same. As the derived nonlinear model is used for linearization in one operating point for the controller design, it is reasonable only to validate the model around the same operating point. In the validation no controller is used under the simulations, so the control inputs, pitch and generator torque, are set to the steady state values for the working point and two steps in the wind speed is applied to the models. The applied wind, V, is shown on Figure 5.1.

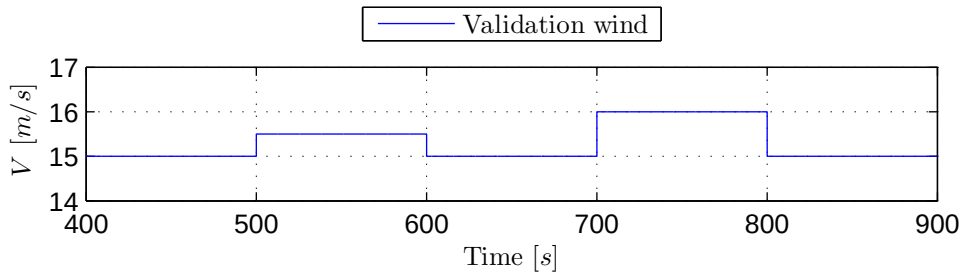


Figure 5.1: *For the validation a laminar wind with two steps is used to show the responses of the models.*

For the validation a 900 seconds simulation is carried out, where the first 400 seconds are given for the model to find steady state in the operating point. The last 500 seconds of the simulation are used for the validation. The wind speed signal is used in a simulation of the model in LACflex and the output from this simulation is used as inputs for the different submodels in the derived nonlinear model.

5.3.2 Aerodynamic Model

The first part of the derived model to be compared to the LACflex model is the aerodynamic model. The validation wind shown in Figure 5.1, the pitch angle β and the generator torque Q_g for the operational point, are applied to the LACflex simulation and the derived nonlinear model. The setup is illustrated in Figure 5.2.

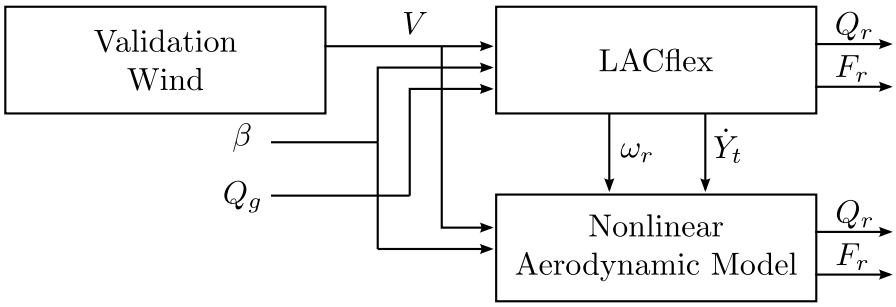


Figure 5.2: *Validation setup for the aerodynamic model.*

The last two inputs to the nonlinear model, the rotor speed ω_r and the tower speed $\dot{Y}_t$, is taken from the LACflex simulation to give an optimal comparison of the two models. For the validation of the two outputs, the rotor torque Q_r and the thrust force F_r from the two models are compared. The results from the validation of the aerodynamic model is plotted in Figure 5.3.

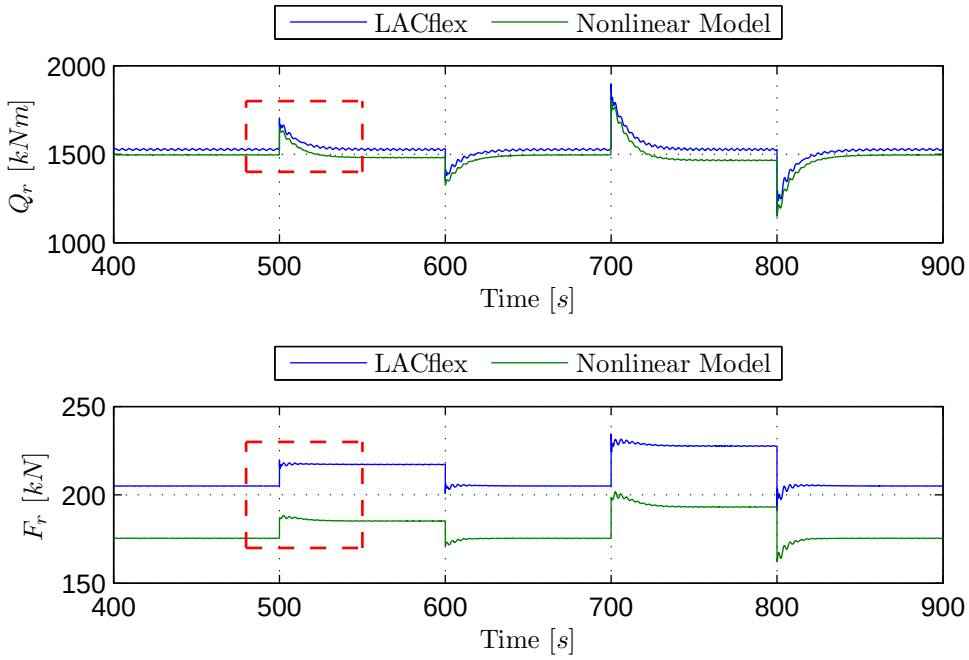


Figure 5.3: *Simulation of the aerodynamic submodel*

From the simulation data it can be seen, that there seems to be a constant offset when the model reaches steady state between the steps. The smaller dynamics from the models can be hard to see in Figure 5.3 so the step response of the models from the first step, marked by the red box is shown in Figure 5.4.

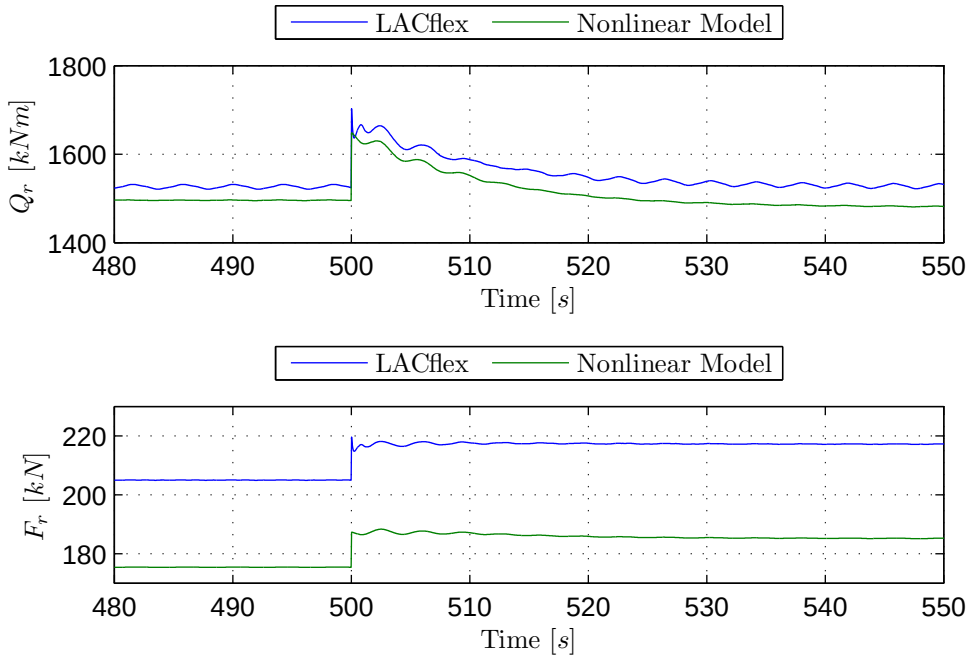


Figure 5.4: *Close-up of the box in figure 5.3, showing the rotor torque and thrust force for the simulation results of the aerodynamic validation.*

The biggest difference, except for the offset, seems to be the small oscillations seen on the LACflex outputs for torque Q_r, as might be explained by the degrees of freedoms included in the LACflex model for the blades.

5.3.3 Tower Model

The tower model is validated in the same way as the aerodynamic model, with the input for the model taken from the LACflex simulation and then comparing the output of the model with the same signals from LACflex. The setup is illustrated in Figure 5.5.

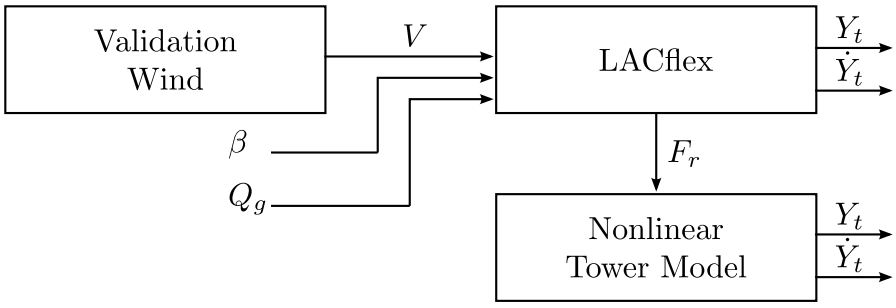


Figure 5.5: *Validation setup of the tower model.*

For the tower validation the thrust force from the LACflex simulation is used as input and the tower displacement Y_t and velocity $\dot{Y}_t$ is the signals compared for the validation. The results from

the validation are shown in Figure 5.6 .

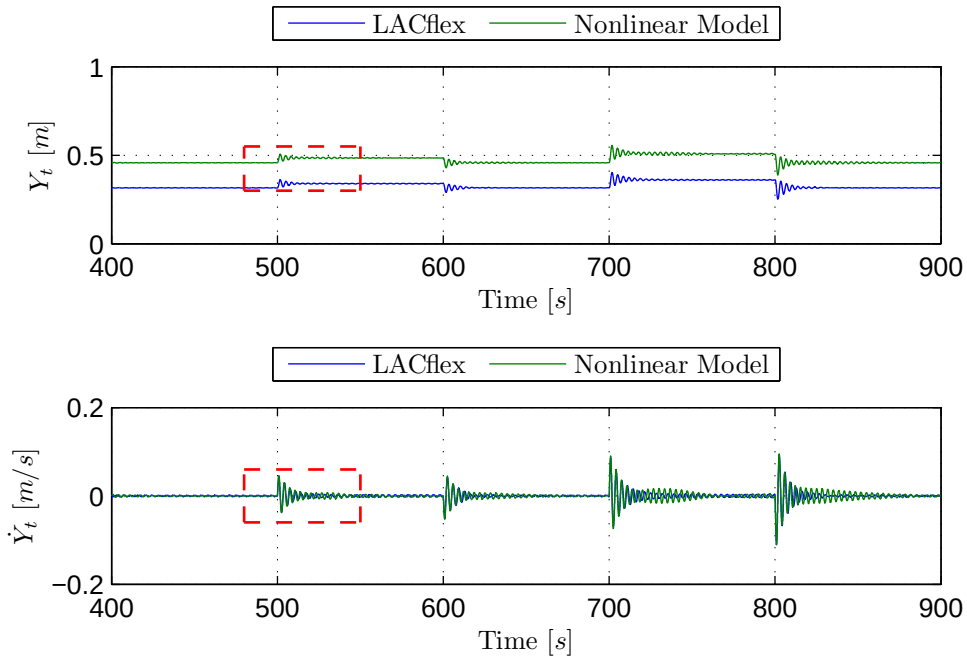


Figure 5.6: *Results from the tower validation. The compared variables for the two models are the tower fore-aft displacement and velocity.*

From the simulation results shown in Figure 5.6 it can be seen that there are an offset between the two models on the tower displacement Y_t, but that the steady state values seems to change the same from the steps. The tower speed $\dot{Y}_t$ seems to follow the same pattern, but to give a better look at the faster dynamics Figure 5.7 shows a close up of the results around the first step.

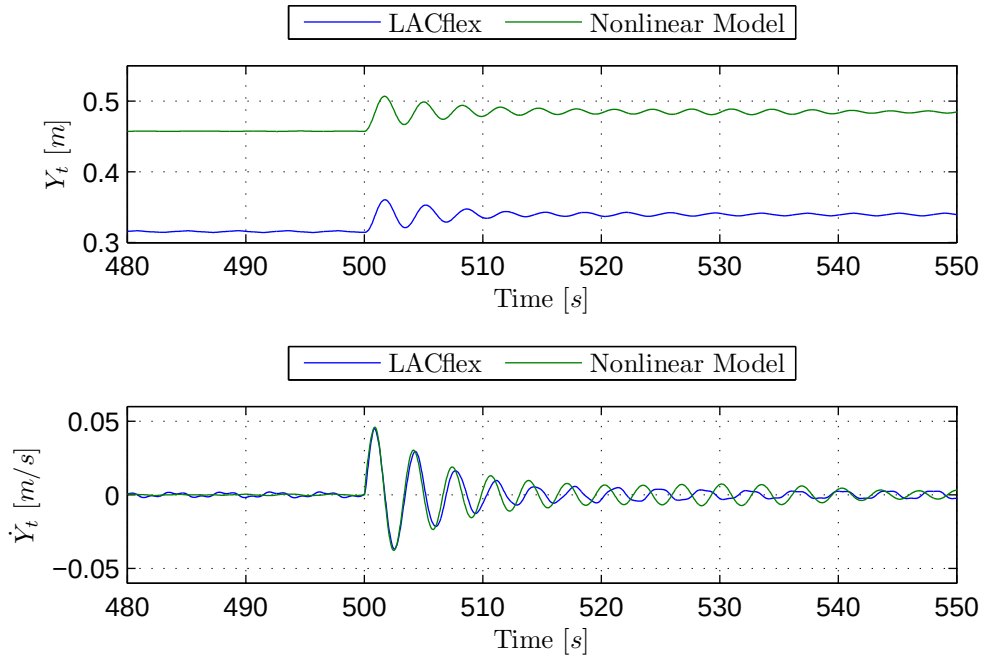


Figure 5.7: *Close-up of Figure 5.6 showing the tower fore-aft displacement and velocity for the tower validation.*

The close-up of the simulation results, shows that the step response except from offset, differs in the dynamics. It seems like the natural frequency of the two models are not completely the same.

5.3.4 Drive Train Model

The drive train model is validated with the same method as the aerodynamic and tower model. For the validation of the drive train model, it is the rotor speed ω_r, generator speed ω_g and the drive train torsion θ that are compared. As input for the nonlinear model in the simulation is despite the wind, generator torque and pitch the rotor torque Q_r from the LACflex model. The setup is shown in Figure 5.8.

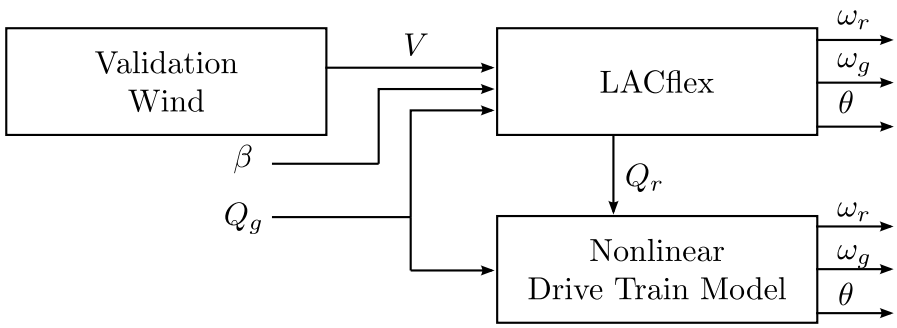


Figure 5.8: *Drive train setup for validation.*

The data from the validation is shown in Figure 5.9. It is seen, that there like the two previous

models, are an offset present on all three outputs, but the major dynamic seems to be the same.

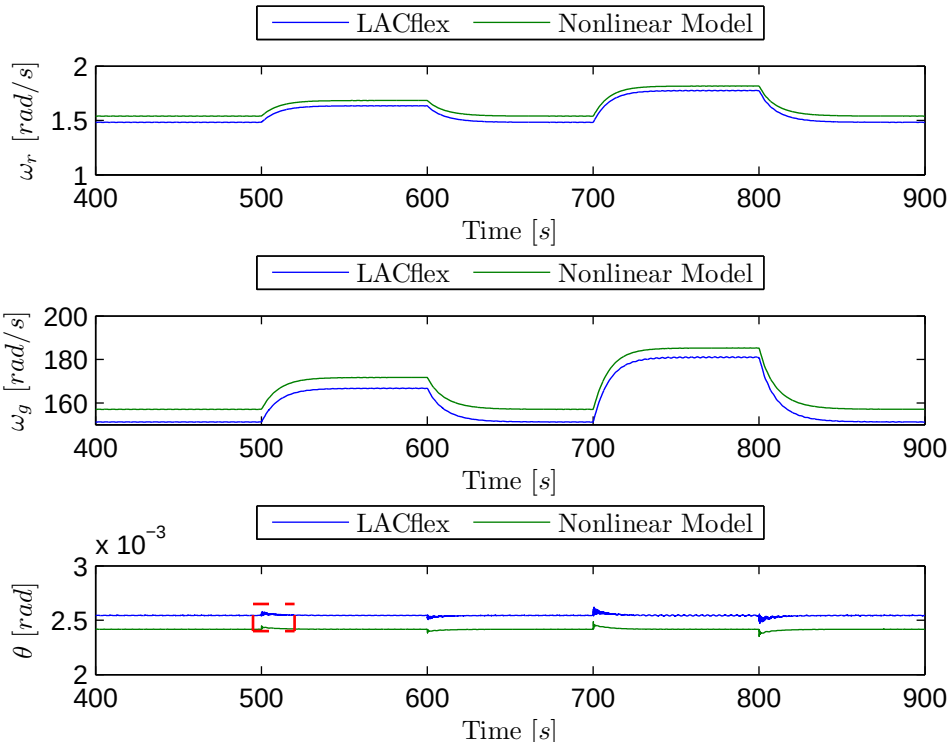


Figure 5.9: *Results from the validation of the drive train model.*

There do not seem to be any fast dynamics on ω_r and ω_g, why only a close-up of the drive train torsion is shown in Figure 5.10.

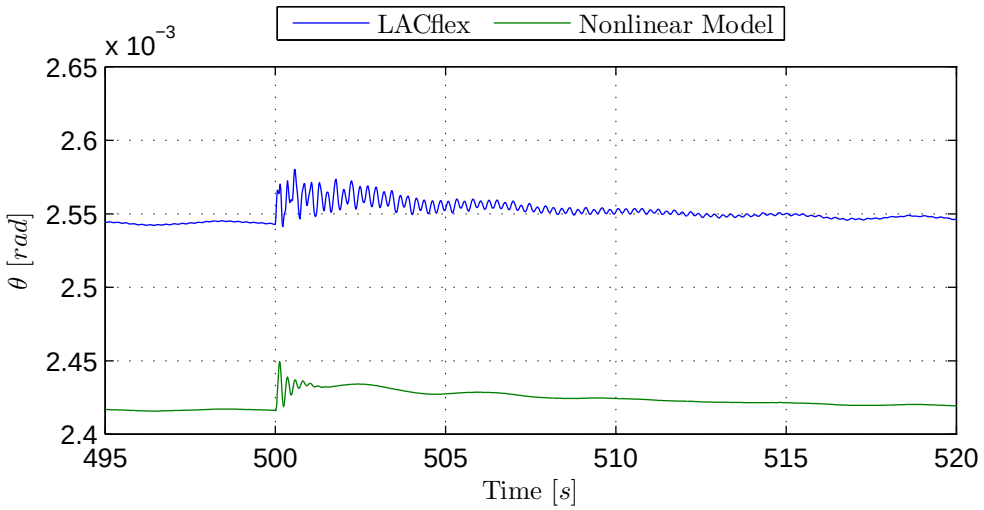


Figure 5.10: *Close-up of the drive train torsion from Figure 5.9, showing the results for the validation of the drive train model.*

The data in Figure 5.10 shows that there is more variations on the LACflex model than the nonlinear model. It can be seen that the fast dynamic that are present in the nonlinear model

right after the step are similar to the dynamics in the LACflex model at the same time, where the LACflex dynamic continues the nonlinear model settles. The dynamics seen in the LACflex model seems to be the same as the one for the rotor torque Q_r seen in Figure

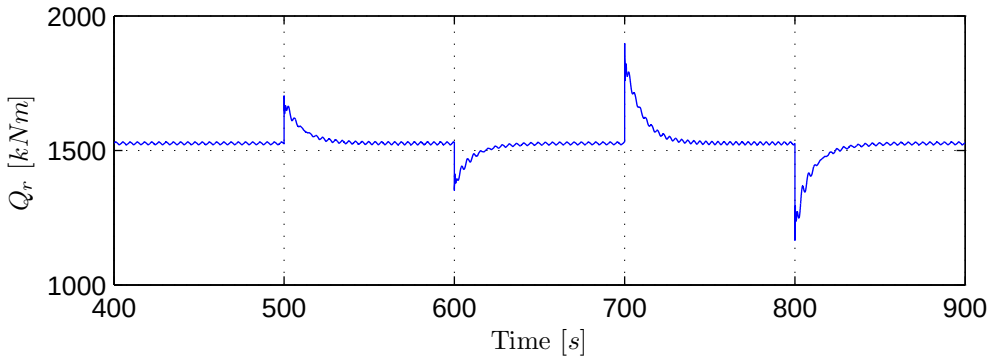


Figure 5.11: *The rotor torque.*

5.3.5 Generator Model

The generator model is the last submodel of the developed model to be compared to LACflex. The generator model is validated in the same way as the previous submodels, where the simulated data from LACflex are used as input for the nonlinear model. The test setup is shown in Figure 5.12.

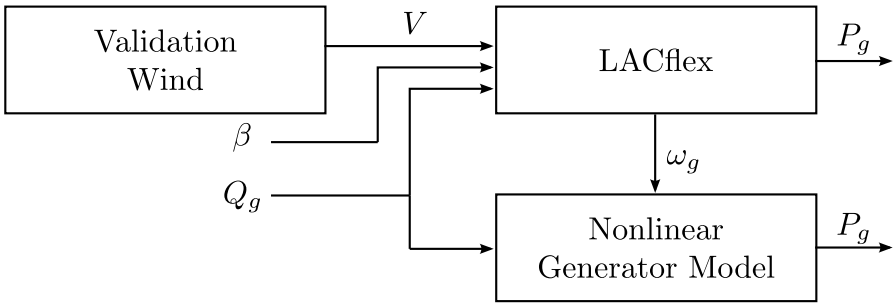


Figure 5.12: *The setup for the validation of the Generator Model.*

As input to the model the generator torque Q_g and pitch β is set to the values for the operating point and the generator speed ω_g is taken from the LACflex model. The generator is a simple model and the only output that is compared is the generator power P_g which can be seen in Figure 5.13.

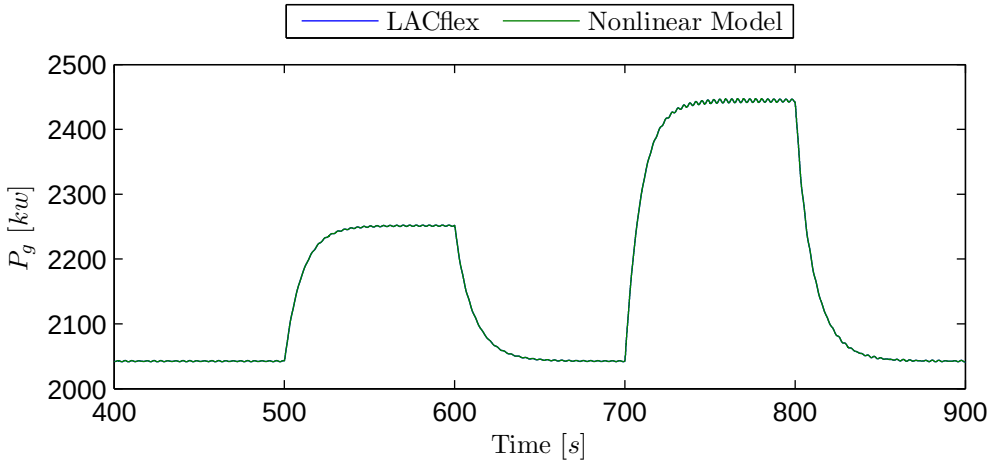


Figure 5.13: *The generator power, from the validation of the generator model.*

In the generator model it is only the dynamic of the input signals that can be seen from the data, this is the case for both the nonlinear model and the LACflex model.

5.3.6 Validation of the Combined Model

All four submodels of the developed nonlinear model have been compared to their respective part of the LACflex model. For a final validation the combined non linear model is compared to the model in LACflex. The model is validated by giving the two models the same input and comparing the outputs of the models. The setup of the validation is shown in Figure 5.14.

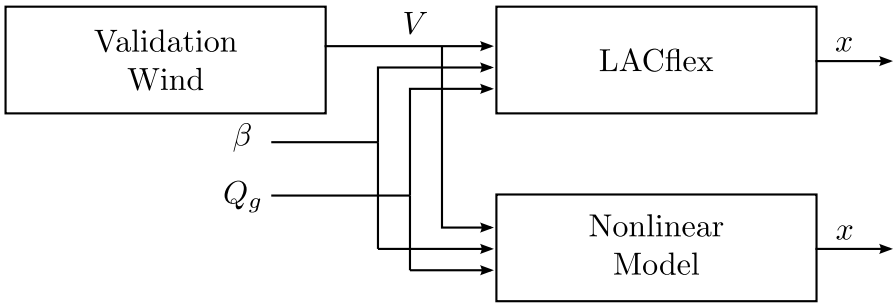


Figure 5.14: *Setup for validation of the combined Nonlinear Model.*

For the validation the same inputs are used as for the validation of the submodels, the wind is the one shown in Figure 5.1 and the pitch and generator torque are held at the operating points. In the simulation it is the states of the nonlinear model that are compared as they are responsible for the dynamics of the model. The result of the validation is shown in Figure 5.15.

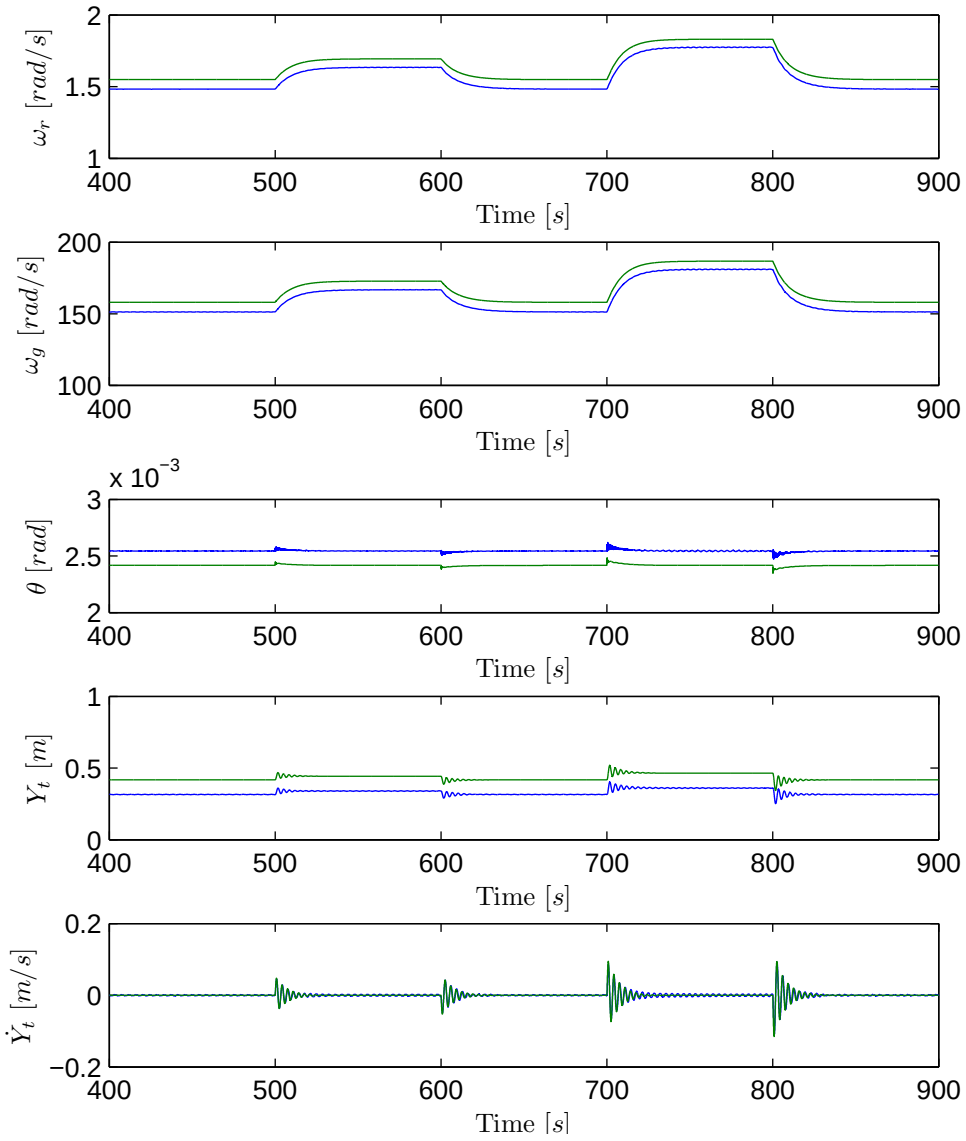


Figure 5.15: *The states from the validation of the combined model.*

From the validation for the nonlinear model and the model from LACflex the same tendencies are seen as were seen for the individual parts of the model. There seems to be some offsets on all states, but the developed nonlinear model displays most of the fundamental dynamic of the LACflex model. The developed nonlinear model is deemed to be sufficient to base a controller on for the use in simulations in LACflex.

5.4 Design of State Observer

An observer has to be designed to estimate the states as not all states are outputs from the turbine model in LACflex. If the turbine model is given as

$$x[k+1|k] = Ax[k] + Bu[k] + Gw[k] \tag{5.1}$$
$$y[k] = Cx[k] + v[k] \tag{5.2}$$

where A, B and C are the system matrixes, G is how the process noise w enter the system and v is the sensor noise. The process noise is the wind and G is given as the model B_d as in equation (4.39). Given the system the observer can be described as

$$\hat{x}[k|k] = \hat{x}[k|k-1] + L'\left(y[k] - \hat{y}[k|k-1]\right) \tag{5.3}$$
$$\hat{x}[k+1|k] = A\hat{x}[k|k] + Bu[k] \tag{5.4}$$
$$\hat{y}[k|k-1] = C\hat{x}[k|k-1]. \tag{5.5}$$

where L' is the observer gain. The block diagram of the observer [19, p. 204] is seen in Figure 5.16. The observer differs from a normal Luenberger observer by how the observer gain is added to the system. L' is added to $\hat{x}[k|k-1]$ where for the Luenberger observer the observer gain denoted L where $L = AL'$ is added to $\hat{x}[k+1|k]$. The difference is that the state at time k, is dependent of the current k as it is updated with the latest output $y[k]$, where the Luenberger predicts the state on behalf of the previously measurement.

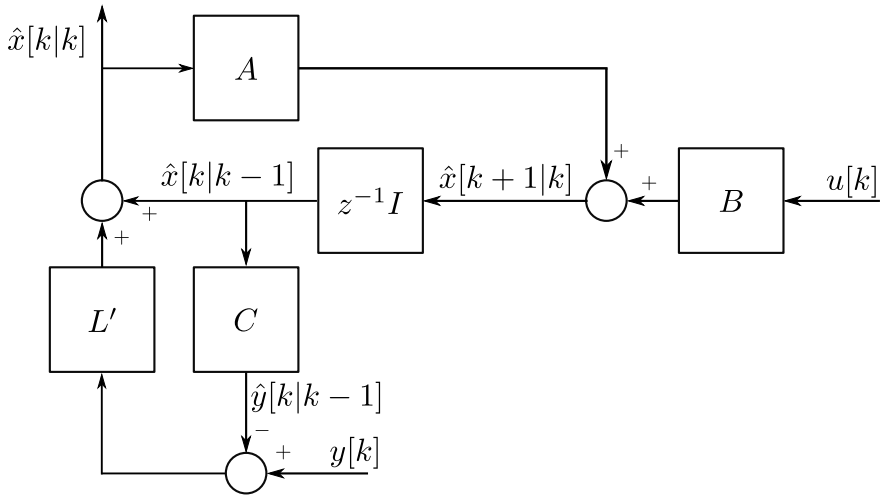


Figure 5.16: *Block diagram of state observer.*

The observer gain L' can be calculated by placing the observer poles with the MATLAB command *place*. Another way is to calculate the observer gain is by using Kalman with the MATLAB command *Kalman*. The advantage of Kalman is that it calculates the optimal gain based on the estimated disturbances on the plant.

The equations for the Kalman filter [24] are given as

Prediction:

$$\hat{x}[k|k-1] = A\hat{x}[k-1|k-1] + Bu[k-1] \tag{5.6}$$

$$\hat{y}[k|k-1] = C\hat{x}[k|k-1] \tag{5.7}$$

$$P[k|k-1] = AP[k-1|k-1]A^T + Q \tag{5.8}$$

Update:

$$L'[k] = P[k|k-1]C^T \left(CP[k|k-1]C^T + R\right)^{-1} \tag{5.9}$$

$$\hat{x}[k|k] = \hat{x}[k|k-1] + L'[k]\left(y[k] - \hat{y}[k|k-1]\right) \tag{5.10}$$

$$P[k|k] = \left(I - L'[k]C\right)P[k|k-1] \tag{5.11}$$

where P is the error covariance $E\left[(x-\hat{x})(x-\hat{x})^T\right]$, Q is the variance $E[ww^T]$ of the process noise w, R is the variance $E[vv^T]$ of the sensor noise v.

According to [19, p. 226] and [8], if the system does not depend on time and the disturbances ω and v are stationary the Kalman gain reaches a steady state given as

$$L' = PC^T \left(CPC^T + R\right)^{-1} \tag{5.12}$$

where P is the steady state error covariance $P = \lim_{t\to\infty} E\left[(x-\hat{x})(x-\hat{x})^T\right]$, and is given as the solution to the algebraic Riccati equation coming from substituting the Kalman gain equation (5.9) into equation (5.11) and afterwards substitute it into equation (5.8) which gives

$$P = APA^T - APC^T \left(CPC^T + R\right)^{-1} CPA^T + Q \tag{5.13}$$

which inserted in equation (5.12) gives the observer gain. The observer gain is calculated with the MATLAB command *Kalman* which takes the system and the variance of Q and R as input. The value of Q and R are used as tuning parameters. Q is scaler value representing the variance of the wind and R is a matrix with the variance of the different sensors on the diagonal.

5.5 LACflex Simulation Results

A simulation is carried out in LACflex which has a more advanced turbine model, than the derived one used for the MATLAB simulations. To be able to run a simulation in LACflex, a wind field is necessary. The wind generator in LACflex is used to generate three different wind fields, with different turbulence intensities according to [31]. The three different wind fields that are generated, are the same ones that was used for the MATLAB simulations, but for the MATLAB simulations the wind fields was reduced to average rotor wind speeds, seen in Figure 4.28, 4.29 and 4.30.

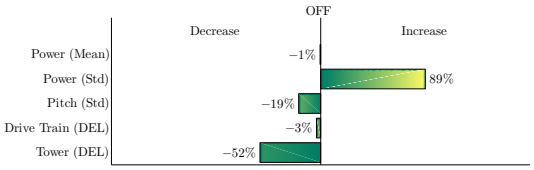
In the MATLAB simulations, four different LIDAR models were simulated. In LACflex, it is not possible to simulate the IDEAL case, as the wind field can not be used as a LIDAR signal in the model predictive controller.

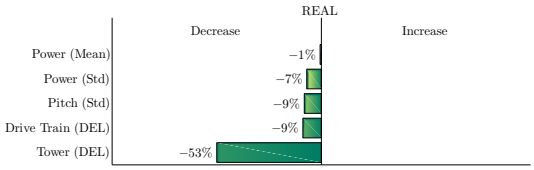
As the turbine in LACflex is a different type than the NREL and the dynamics are different, the

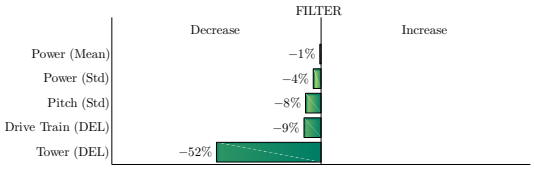
same controller can not be used in both cases and a different tuning has to be used. Different tunings and different turbine models makes it difficult to match the results with MATLAB. The benchmark controller in LACflex is also just for power tracking, no additional damping on the drive train and tower is added which also makes a difference in the comparison of the results. Testing in LACflex does however has the advantage of testing if it works with a wind field and a more complex turbine model. It is also possible to see if the controllers perform in a similar way on a different turbine and if the LIDAR has the same influence on the power variation as it had in the MATLAB simulations.

The controller is tuned, such that the OFF case is able to control the turbine and not to reduce fatigue as much as possible. The reason is, that in LACflex is a standard PI controller for power tracking. This is not a suitable benchmark controller and it is difficult to test if a potential damping is caused by LIDAR or the use of model predictive control. A model predictive controller without LIDAR is therefore necessary for comparing the effect of using LIDAR. The results for OFF, REAL and FILTER compared to the PI controller in LACflex, for seed A at the three different turbulence intensities are given below. The results for seed B and C are in Appendix B.

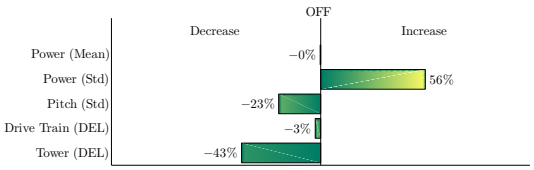
Seed A turbulence intensity 14%

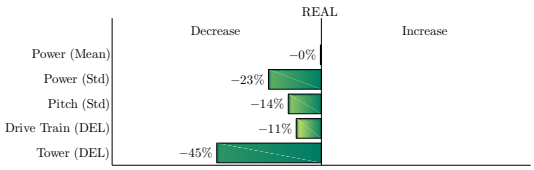


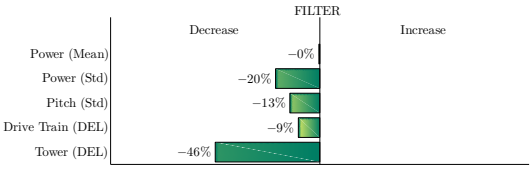




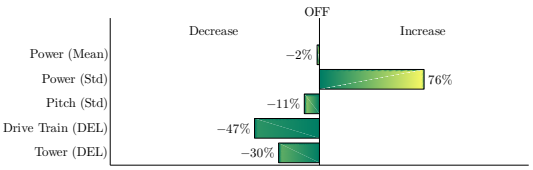
Seed A turbulence intensity 16%

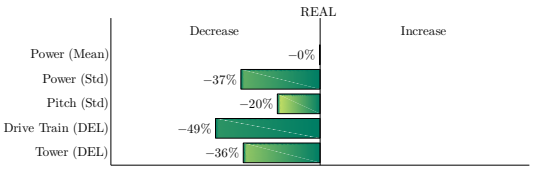


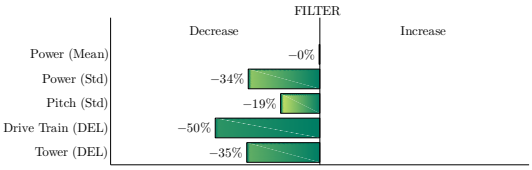




Seed A turbulence intensity 18%







As seen in the results, OFF still has higher standard deviation than the benchmark controller as it had in the MATLAB simulations. The other two cases where LIDAR is used, the standard deviations is kept under the benchmark and as the results for the MATLAB simulation, at higher turbulence intensities the controllers with LIDAR performs even better compared to the benchmark because the higher intensity does not affect the MPC controller as much as it does for the benchmark. It is seen in the bar plots that there is a significant decrease on the drive train torsion from 16 to 18%. The reason for this is seen in figure 5.17, where the top figure is the benchmark and the bottom is the MPC with the REAL LIDAR model. It is seen that the MPC performs almost the same for all turbulence intensities, where the benchmark at 18% varies more than it did for the lower intensities.

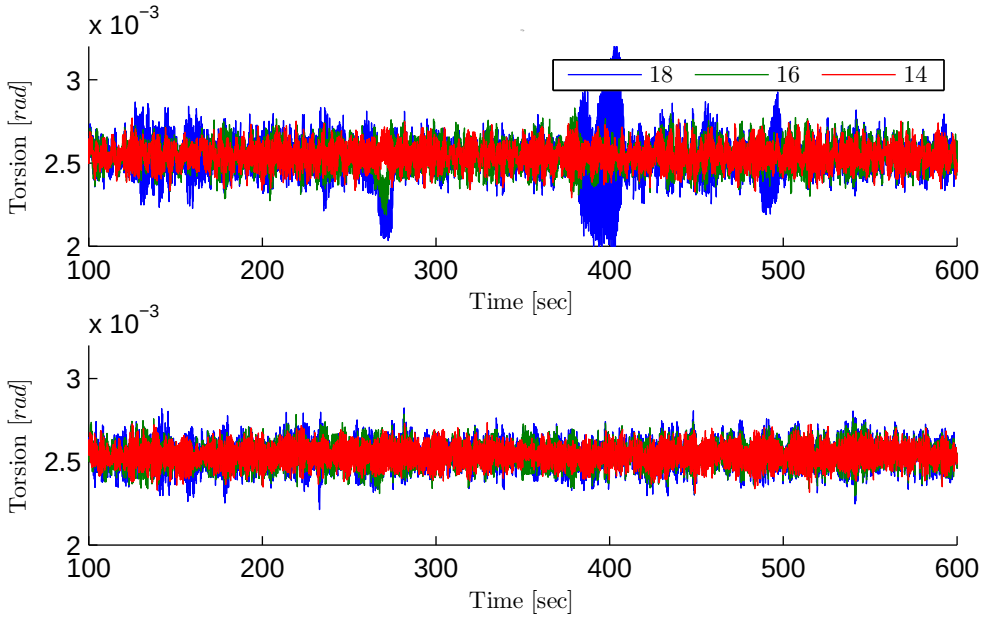


Figure 5.17: *The drive train torsion, at wind seed A at 14, 16 and 18% turbulence intensity. The top figure is the benchmark and the bottom figure is the MPC with LIDAR model REAL.*

It is not only the drive train torsion but also the power that is affected by the higher turbulence intensities, as seen in Figure 5.18. The figure shows that at higher turbulence intensities there are some big spikes on the power for benchmark, compared to the MPC with LIDAR where the higher turbulence intensities does not really affect the power.

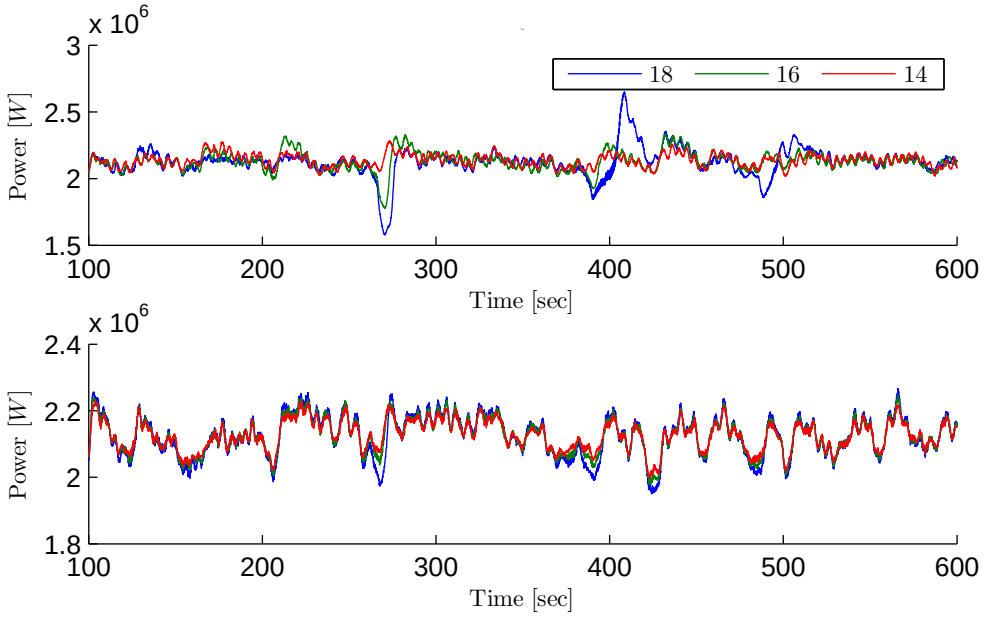


Figure 5.18: *The power, at wind seed A at 14, 16 and 18% turbulence intensity. The top figure is the benchmark and the bottom figure is the MPC with LIDAR model REAL.*

The bar plot also showed a damping of the tower. The effect of the damping is seen in Figure 5.19 that shows the tower fore-aft mode displacement for the benchmark and REAL for wind seed A at 16%. It is seen as the bar plot showed that the tower varies less for REAL than for the benchmark.

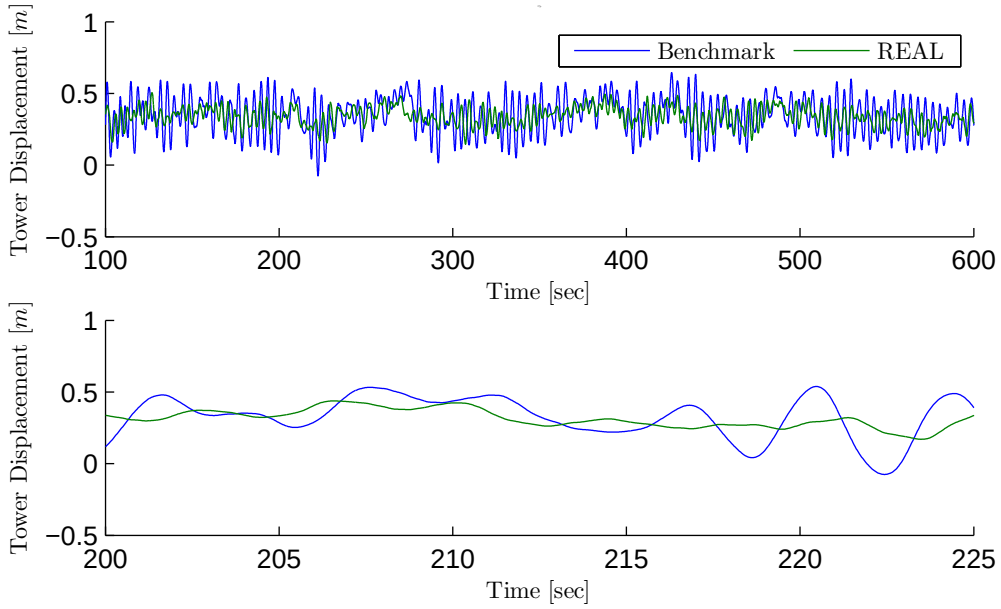


Figure 5.19: *The tower fore-aft mode for benchmark and REAL, simulated with wind seed A at 16% turbulence intensity.*

5.5.1 Discussion of LACflex Results and Comparison to MATLAB Results

As mentioned previously the turbine models in MATLAB and LACflex are different and the tunings of the controllers are different and the benchmark models are different. However the results from the simulations in LACflex, shows some of the same tendencies as the results from the MATLAB simulations, despite the differences.

It is seen that the OFF case where no LIDAR is used, damps the drive train and tower both in MATLAB and LACflex and has a higher standard deviation than the benchmark in both cases.

Adding LIDAR showed in MATLAB, that the trade off between mitigating fatigue and higher power variations, as seen in the OFF case, could not only be removed but decreased, this is also the case in the LACflex simulations.

Despite the tower in LACflex, it seems that the tendency from the simulations in MATLAB, that there is an improvement as the turbulence intensity increases, also is the case in LACflex. It is however not that clear in the LACflex simulations, as it is in MATLAB. One possible reason could be that the benchmark controller in LACflex seems to change its performance from 16 to 18% quite much which espacially is seen on the drive train. But also the tuning of the controller could affact the result as there are different weights on the tower and drive train in MATLAB and LACflex.

In the MATLAB simulations the FILTER seemed to have a positive influence on the damping of the drive train compared to the unfiltered REAL. In LACflex this does not seem to be the case, where the FILTER do no good in comparison to REAL. An explanation could be the faster dynamics of the LACflex turbine.

Chapter 6

Conclusion

The following summarizes the work and results in this thesis, with the goal to be able to accept or reject the hypothesis at the end:

It is possible to use a LIDAR sensor in the control of a wind turbine, with the goal to minimize dynamic loads on the mechanical parts of the turbine while still maintaining power production.

The work with wind turbines is motivated by the increasing demand for alternative and environmentally friendly energy in correlation with the substitution of fossil fuels and the increasing demand of energy. To make wind energy a cost competitive energy source, relies on development and research into how energy is best extracted from the wind. This includes better materials and better control systems which can reduce the loads on the turbine. Lower loads make it possible to save money on the constructions, as it does not have to withstand the full load. It does also make it possible to build bigger turbines that can produce more power.

For simulation purposes a nonlinear model of a wind turbine was derived based on physics. The model was validated by its physical characteristics and by comparing it to another turbine model at the same level of detail. The nonlinear model was also compared to the more complex model in LACflex. The results showed that the models had similar dynamics but there was an offset difference. From the nonlinear model a linear model for use in a later developed controller was derived and validated. For the simulations a wind model was described but not developed as a wind generator in LACflex was used to generate the average rotor wind.

For use together with the later developed model predictive controller a LIDAR model was developed. The developed LIDAR model did use the in LACflex included LIDAR model, that extracts measurements at a specified radius and time in front of the turbine from the generated wind field. Assuming only upwind nonzero and frozen turbulence, the measurements was averaged and delayed and used as input to the model predictive controller.

To serve as a benchmark controller for the later developed model predictive controller, a standard controller for power tracking consisting of a torque controller for below rated wind speed and a pitch controller for above rated was devolved. For a fair comparison additional damping of the

tower and drive train was added to the standard controller.

A model predictive controller (MPC) for mitigating loads on the tower fore-aft mode and drive train torsion by including LIDAR in the prediction was developed. The controller was validated against the benchmark controller and the same MPC without LIDAR. This is to determine if LIDAR have an effect or it is just the use of MPC as controller that minimizes loads.

For simulations three different wind fields was generated in LACflex with turbulence intensities of 14, 16 and 18%. The controller was tested both on the nonlinear turbine model in MATLAB and on a more advanced turbine model in LACflex for all the generated wind fields. The performance of the controllers was assessed by calculating the damage equivalent load for the tower fore-aft mode and the drive train torsion and the standard deviation of the pitch and power.

The results in MATLAB showed that by using model predictive control, the load on the tower fore-aft mode and the drive train torsion could be mitigated with an increase in the power standard deviation, compared to the developed benchmark controller. Including LIDAR in the MPC not only removed the trade off between lower fatigue and higher power standard deviation, but lowered the standard deviation and maintained the damping on tower and drive train. Filtering the LIDAR measurements with a low pass filter damped the torsion of the drive train further. It was also seen that the MPC did perform better than the benchmark at higher turbulence intensities.

Simulations in LACflex on the more advanced turbine model, showed some of the same tendencies as the simulations in MATLAB despite the differences in the turbine models and controller tunings. MPC without LIDAR did mitigate the load on tower fore-aft mode and drive train, but as in the MATLAB simulations with a trade off as the power standard deviation increased. Including LIDAR showed in MATLAB, that the trade off between mitigating fatigue and higher power variations, could not only be removed but decreased, the same was seen in the LACflex simulations. Filtering the LIDAR proved in the MATLAB simulations to have a positive affect in mitigating the drive train torsion, this was however not the case in LACflex where the MPC with the filtered LIDAR performed worse than the one where it was not filtered. In MATLAB the performance of the MPC did increase with the turbulence intensity, the same was seen in LACflex but not as clear as in MATLAB.

A more advanced setup would perhaps change the outcome, but with the setup used in this thesis, it has been possible to design a model predictive controller that uses LIDAR in the control. The controller with LIDAR has been able to reduce the tower fore-aft mode and drive train torsion as well as keeping the power quality. The hypothesis is therefore accepted under these setup conditions.

Chapter 7

Perspectives

In this chapter idea for future work is described.

7.1 Turbine Model

The implemented turbine model in this thesis includes the dynamics of the drive train and tower as this thesis focused on reducing the fatigue on these. If the general goal is to represent loads on the turbine, the dynamics of the blades could be included in the model.

The derived model showed some offsets compared to the LACflex model. This indicates that there can be model parameters that are not estimated precise enough or that there are properties of the LACflex model, which are not represented by the derived model. An example is that the LACflex model includes mechanical loss in the drive train and has tilted the rotor a little. To reduce the offsets these properties can be added to the developed model.

7.2 LIDAR Model

The LIDAR model implemented in the thesis is used to estimate the rotor average wind speed, which is a simple representation of the wind field. If the LIDAR model is expanded with more information about the wind field, a more accurate representation of the field could be estimated. If the LIDAR gives measurements that varies vertical and horizontal it might be possible to estimate the wind shears and other phenomenas in the wind. The extra information about the wind field could for example be used by a controller for further load reduction on the drive train and tower, or other parts of the turbine.

7.3 Controller

The controller is implemented for a single operating point, as future work it could be implemented to operate in multiple points, both over and under the rated wind speed. If the goal for

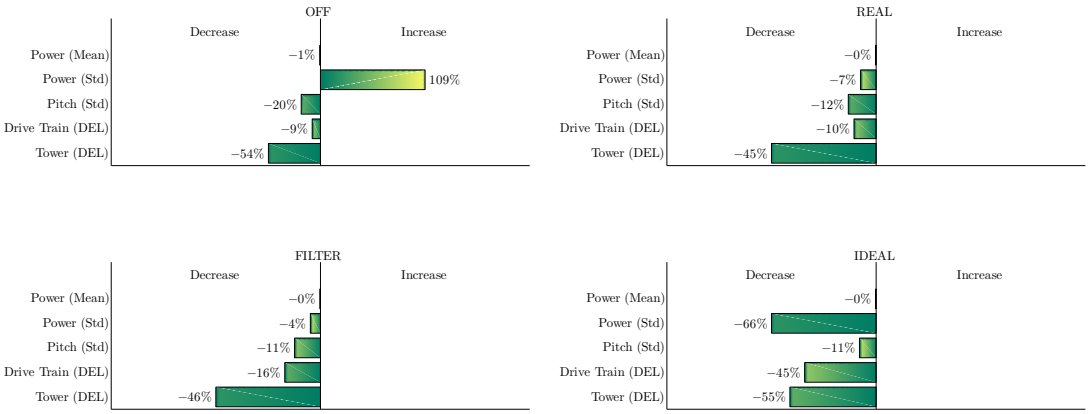
the controller is to operate in all wind speeds, then some form of startup and shutdown procedures have to be implemented. If the model is increased in complexity or the LIDAR are used to estimate a more accurate representation of the wind field other control goals can be implemented.

Appendix A

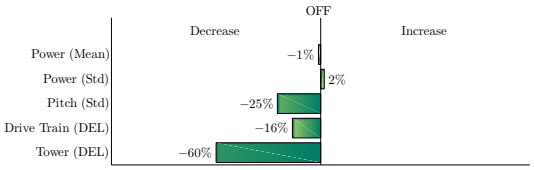
Results for MPC Test in MATLAB

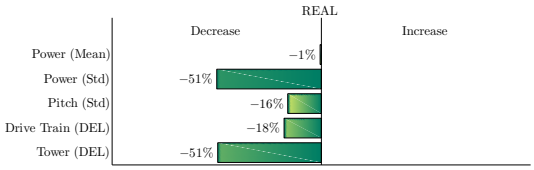
In this appendix is the rest of the results from the simulation in MATLAB, described in section 4.5. The simulation was carried out on three different wind seeds, A, B and C at three different turbulence intensities 14, 16 and 18%. The results from seed A are in the report, where seed B and C are in this appendix. The results shows in percent how the MPC for four different LIDAR models varies from the benchmark controller. The benchmark controller is a standard power tracking PI controller with added damping on tower and drive train desribed in section 4.3.

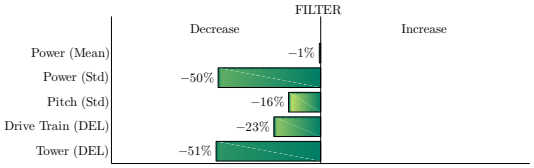
A.0.1 Seed B turbulence intensity 14%

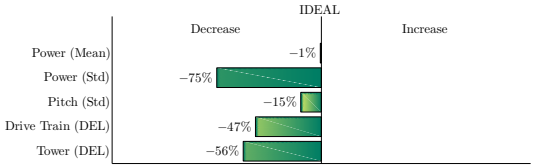


A.0.2 Seed B turbulence intensity 16%

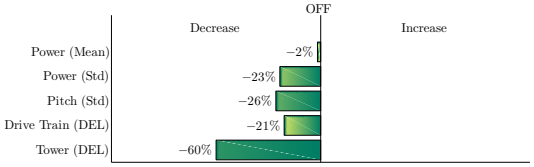


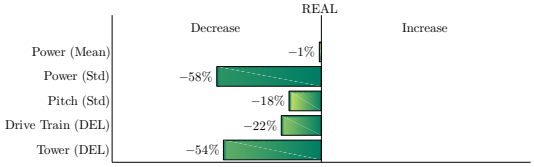


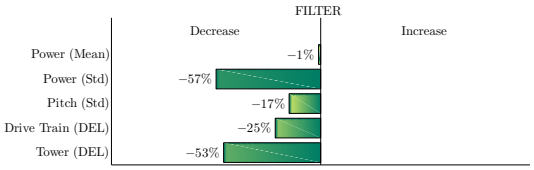


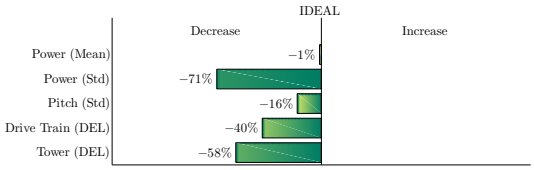


A.0.3 Seed B turbulence intensity 18%

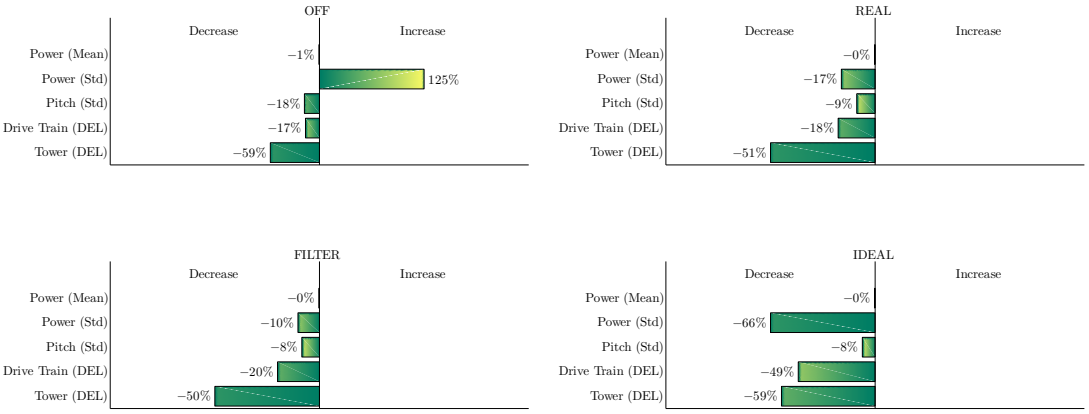




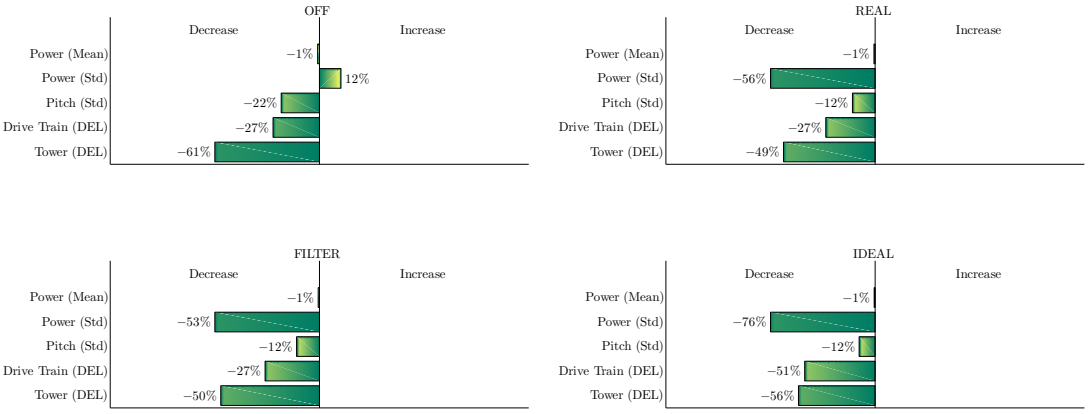




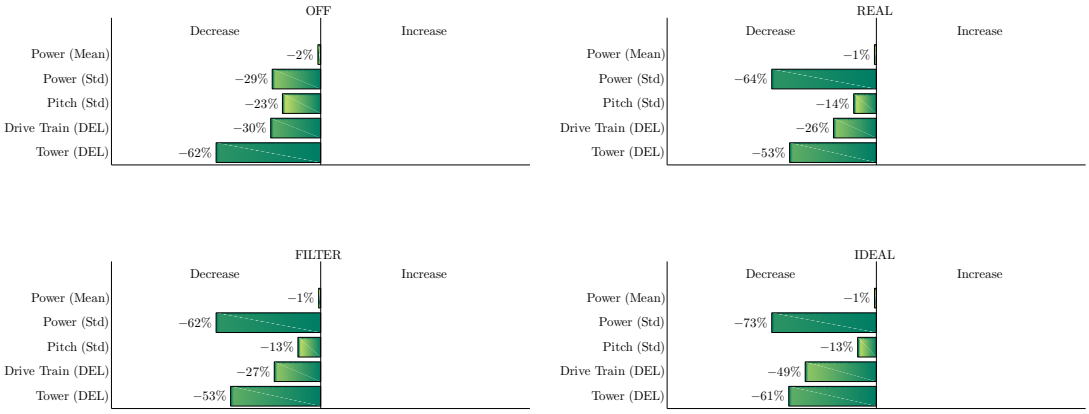
A.0.4 Seed C turbulence intensity 14%



A.0.5 Seed C turbulence intensity 16%



A.0.6 Seed C turbulence intensity 18%



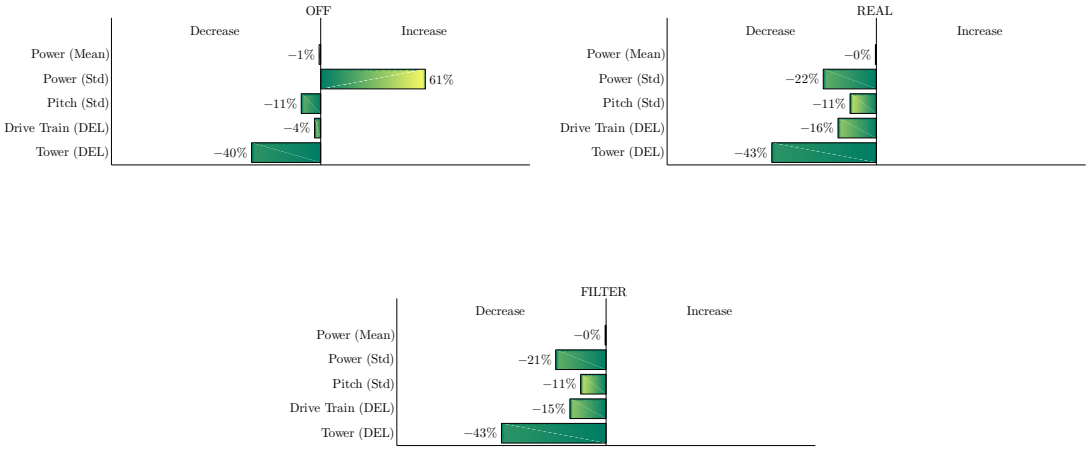
Appendix B

Results for MPC Test in LACflex

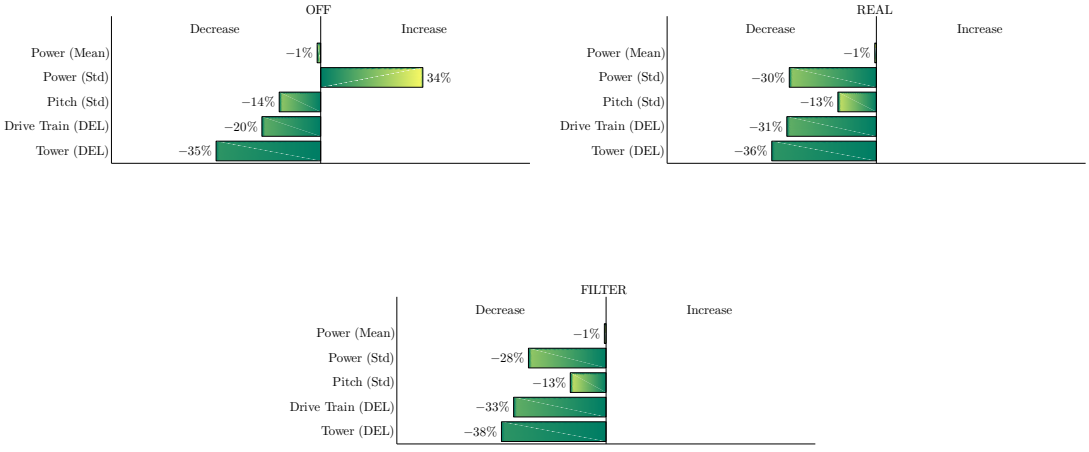
In this appendix is the rest of the results from the simulation in LACflex described in section 5.5. The simulation was carried out on three different wind seeds, A, B and C at three different turbulence intensities 14, 16 and 18%. The results from seed A are in the report, where seed B and C are in this appendix. The results shows in percent how the MPC for three different LIDAR models varies from the benchmark controller. The benchmark controller is a standard power tracking PI controller which is a part of LACflex.

B.0.7 Seed B turbulence intensity 14%

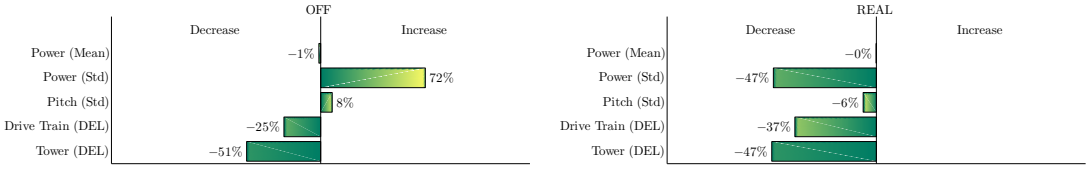
B.0.8 Seed B turbulence intensity 16%

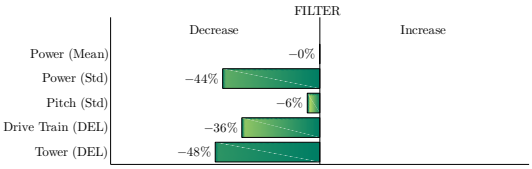


B.0.9 Seed B turbulence intensity 18%

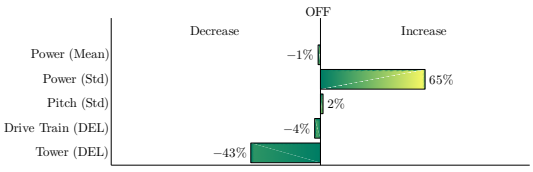


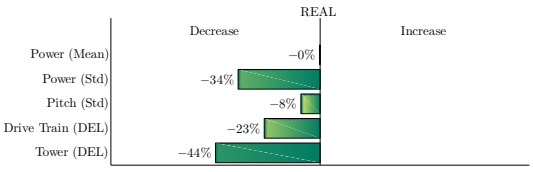
B.0.10 Seed C turbulence intensity 14%

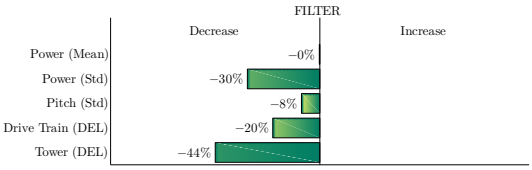




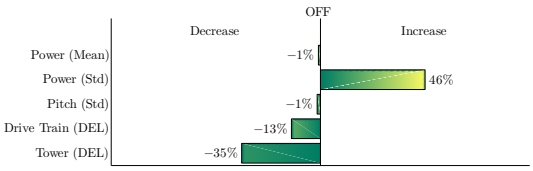
B.0.11 Seed C turbulence intensity 16%

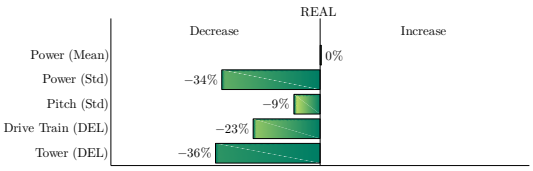


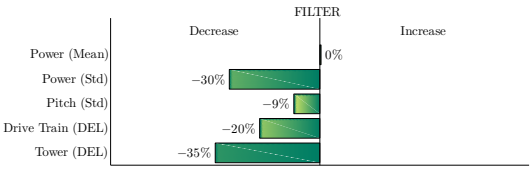




B.0.12 Seed C turbulence intensity 18%







Bibliography

[1] Palle Andersen and Tom S. Pedersen. *Modeldannelse*. Februar 2010.

[2] S. ARIDURU. Fatigue life calculation by rainflow cycle counting method. December 2004.

[3] The European Wind Energy Association. Eu energy policy to 2050 - achieving 80-95% emissions reductions. March 2011.

[4] The European Wind Energy Association. Upwind - design limits and solutions for very large wind turbines. March 2011.

[5] Fernando D. Bianchi, Hernán De Battista, and Ricardo J. Mantz. *Wind Turbine Control Systems*. Advances in Industrial Control. Springer, 2007. ISBN: 1-84628-492-9.

[6] E. A. Bossanyi. The design of closed loop controllers for wind turbines. *WIND ENERGY*, (3):149–163, 2000.

[7] E. A. Bossanyi. Individual blade pitch control for load reduction. *WIND ENERGY*, (3): 119–128, 2003.

[8] Stephen Boyd. The kalman filter. http://www.stanford.edu/class/ee363/lectures/kf.pdf, 09 2008.

[9] Marshall Buhl. Nwtc design codes. http://wind.nrel.gov/designcodes/postprocessors/crunch/.

[10] European Commission. Energy 2020 - a strategy for competitive, sustainable and secure energy. 2011.

[11] ReliaSoft Corporation. Miners rule and cumulative damage models. http://www.weibull.com/hotwire/issue116/hottopics116.htm.

[12] Fiona Dunne, Lucy Y. Pao, Alan D. Wright, Bonnie Jonkman, Neil Kelley, and Erik Simley. Adding feedforward blade pitch control for load mitigation in wind turbines: Non-causal series expansion, preview control, and optimized fir filter methods.

[13] LAC engineering. Lacflex & technology transfer. http://lacengineering.com/images/Technology/LACflex_productsheets.pdf.

[14] Kasper Frandsen. Uro i mellemøsten kan sende olieprisen op på 150 dollars. http://www.business.dk/energi-miljoe/uro-i-mellemoesten-kan-sende-olieprisen-op-paa-150-dollars, February
2011.

[15] Jacob Deleuran Grunnet, Mohsen Soltani, Torben Knudsen, Martin Kragelund, and Thomas Bak. Aeolus toolbox for dynamic wind farm model, simulation and control. In *Proc. of the 2010 European Wind Energy Conference*, 2010.

[16] Keld Hammerum. A fatigue approach to wind turbine control. 2006.

[17] João P. Hespanha. Undergraduate lecture notes on lqg/lqr controller design. note, April 2007. Adresse:.

[18] Jan Hromadko. Germany can produce 65% of its power with wind - industry group. http://www.nasdaq.com/aspx/stock-market-news-story.aspx?storyid=201104050901dowjonesdjonline000138&title=germany-can-produce-65-of-its-power-with-wind—industry-group, May
2011.

[19] J.M.Maciejowski. *Predictive Control with constraints*. Pearson Education Limited, 2002. ISBN: 978-0-2013-9823-6.

[20] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system develpment. Technical Report TP-500-38060, NREL, February 2009.

[21] Bjarne S. Kallesøe. Hawc2 course lesson 3: Control interface. *DTU*.

[22] Klimakommissionen. Grøn energi - vejen mod et dansk energisystem uden forssile brændsler. September 2010.

[23] Jason Laks, Lucy Y. Pao, Eric Simley, Alan Wright, Neil Kelley, and Bonnie Jonkman. Model predictive control using preview measurements from lidar. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[24] Jesper A. Larsen. Linear kalman filtering. http://www.control.aau.dk/ bisgaard/teaching/ias9/esif2009/mm3.pdf.

[25] Julia Layton. How wind power works. http://science.howstuffworks.com/environmental/green-science/wind-power2.htm.

[26] A. Wright M. Harris, M. Hand. Lidar for turbine control. Technical Report TP-500-39154, NREL, January 2006.

[27] T. Mikkelsen, K. Hansen, N. Angelou, M. Sjöholm, M. Harris, P. Hadley, R. Scullion, G. Ellis, and G. Vives. Lidar wind speed measurements from a rotating spinner. *ewec 2010 Warsaw*, April 2010.

[28] Encyclopedia of Renewable Energy and Sustainable Living. wind turbine. http://www.daviddarling.info/encyclopedia/W/AE_wind_turbine.html.

[29] P3Nyheder. Grøn strøm boomer i tyskland.
http://www.dr.dk/P3/P3Nyheder/2011/04/12/071736.htm, April 2011.

[30] David Schlipf, Simone Schuler, Patrick Grau, Frank Allgöver, and Martin Kühn.
Look-ahead cyclic pitch control using lidar.

[31] WAsP the Wind Atlas Analysis and Application Program. Turbulence modelling.
http://www.wasp.dk/Products/wat/WAThelp/WATmodelling.htm.

[32] Nick Jenkins Tony Burton, David Sharpe and Ervin Bossanyi. *Wind Energy Handbook*.
John Wiley and Sons Ltd, 2001. ISBN: 0-471-48997-2.

[33] Danmarks Vindmølleforening. Sådan fungerer en vindmølle.
http://www.dkvind.dk/fakta/pdf/T1.pdf, 05 2001.