Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Jørgen Urdal

Model Predictive Control of multi-rotor wind turbine

Master's thesis in Industrial Cybernetics

Supervisor: Leif Erik Andersson, Lars Struen Imsland

June 2020

NTNU
Norwegian University of Science and Technology

Jørgen Urdal

Model Predictive Control of multi-rotor wind turbine

Master's thesis in Industrial Cybernetics
Supervisor: Leif Erik Andersson, Lars Struen Imsland
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

Scaling conventional wind turbines to meet the increasing power demand comes with a variety of challenges. By up-scaling wind turbines, the power increases by the square while the weight increases by the cube of the rotor radius. An alternative to the conventional single rotor wind turbine is the multi-rotor wind turbine, where combining several rotors on one single support structure.

Vestas have created a multi-rotor wind turbine control challenge that encourage the implementation of advanced control systems for multi-rotors and in this challenge, two controllers are presented to serve as benchmark for others to compare. This thesis investigates the use of model predictive control for multi-rotor wind turbines. Having several turbines on the same support structure introduces torque that increases the fatigue loads. The objective is to reduce the fatigue loads with a limit on loss of power and not overuse pitch activity.

A tower structure model provided by Vestas is combined with four turbine models in Matlab with CasADi optimization. Trail-and-error approach have led to three objective functions that are presented in the results. Simulations shows that the model predictive controller overall outperforms the controllers presented in the Vestas multi-rotor control challenge. However, for some of the higher wind speeds the controller is not successful in reducing the loads below the standard of the Vestas controller. It is

believed that with proper tuning this can be improved. In addition, the computation time of the model predictive controller is also not real time applicable and further work is required.

Sammendrag

Skalering av konvensjonelle vindturbiner for å imøtekomme det økende kraftbehovet kommer med en rekke utfordringer. Ved å skalere vindturbiner øker effekten med kvadratet, mens vekten øker med kuben til rotorradiusen. Et alternativ til den konvensjonelle enkeltrotor-vindturbinen er multirotor-vindturbinen, der man kombinerer flere rotorer på en enkelt struktur.

Vestas har laget en multi-rotor vindturbin kontroll-utfordring som oppmuntrer implementering av avanserte kontrollsystemer for multirotorer, og i denne utfordringen blir to kontrollere presentert som målestokk for andre til å sammenligne. Denne avhandlingen undersøker bruken av prediktiv kontroll for vindturbiner med flere rotorer. Å ha flere turbiner på samme støttestruktur introduserer momenter som øker utmattingsbelastningen. Målet er å redusere utmattingsbelastningene med en begrensning på tap av kraft produksjon og ikke overforbruk av vinkel endring for turbinbladene.

En tårnstrukturmodell levert av Vestas er kombinert med fire turbinmodeller i Matlab med CasADi-optimalisering. Trail-and-error tilnærming har ført til tre kostfunksjoner som presenteres i resultatene. Simuleringer viser at modellprediktiv kontroller presterer bedre enn kontrollerne presentert i Vestas multirotor kontrollutfordring. For høyere vindhastighetene lykkes imidlertid ikke kontrolleren med å redusere belastningene under standarden til Vestas-kontrolleren. Det antas at med

riktig vekting kan dette forbedres. I tillegg er beregningstiden til modellprediktiv kontroller heller ikke sanntids anvendelig og videre arbeid er nødvendig.

Contents

Abstract **i**

Sammendrag **iii**

Preface **xii**

1 Introduction **1**

1.1 Motivation 1
1.2 Objective and considerations 2
1.3 Previous work 2
1.4 Contributions 3
1.5 Outline 4

2 Literature review **5**

2.1 Wind energy 5
2.2 Wind turbines 7
2.3 Multi-rotor wind turbines 8
2.4 Wind turbine control 11
2.5 Benchmark controller and Vestas controller 12

3 Mathematical model **15**

3.1 Turbine model 17

3.1.1 Aerodynamic . . . 18
3.1.2 Drive train . . . 20
3.1.3 Generator . . . 20
3.1.4 Pitch servo . . . 21
3.2 Tower structure model . . . 22
3.2.1 Load calculation . . . 26
3.3 Combining the models . . . 28
3.3.1 Constraints . . . 30
3.4 Wind field . . . 32

4 Methodology **33**
4.1 Control system . . . 33
4.1.1 Introduction to basic optimization and feedback control . . . 34
4.1.2 Model predictive control . . . 36
4.1.3 Control objective . . . 38
4.2 Optimizer and integration method . . . 40
4.2.1 Collocation method . . . 40

5 Results **45**
5.1 Performance criteria . . . 46
5.2 Simulation results . . . 47
5.2.1 Case number 1 . . . 48
5.2.2 Case number 2 . . . 51
5.2.3 Case number 3 . . . 54

6 Discussion **57**
6.1 Power optimization, DEL reduction and pitch activity . . . 58
6.1.1 Power optimization . . . 58
6.1.2 DEL optimization . . . 60
6.1.3 Pitch activity . . . 63
6.2 Computation time . . . 65

7 Conclusions and future work **69**

A Coefficients for C_p and C_t polynomial **71**

References **73**

List of Tables

3.1 Table of MRT parameters [21]. 17
3.2 Table of variables for the turbine model [1]. 18
3.3 Table of variables for the turbine model [1]. 18
3.4 Parameter table for geometry of the structure[21]. 23
3.5 Description and parameters for the ten degrees of freedom 24
3.6 Table of moments and forces acting on the tower structure 26
3.7 Table of inputs for the full model. Consisting of power reference and pitch reference. 28
3.8 The 40 states that represents the full model. The first 20 is from the turbine model, with 5 for each of the four models($i = 1 : 4$). The 20 last states represents the tower structure. 29

5.1 Comparing DEL's between MPC and Vestas controller for the first case. 48
5.2 Comparing mean power production between MPC and Vestas controller. 49
5.3 Pitch activity shown as pitch distance traveled. Comparing MPC with the Vestas controller for the first case. 50
5.4 Average time per iteration and median time for the simulations of the first case. 50
5.5 Comparing DEL's between MPC and Vestas controller for the second case. 52

5.6 Comparing mean power production between MPC and Vestas controller for the second case. 52
5.7 Pitch activity shown as pitch distance traveled. Comparing MPC with the Vestas controller for the second case. 53
5.8 Averege time per iteration and median time for the simulations of the second case. 53
5.9 Comparing DEL's between MPC and Vestas controller for the third case. 55
5.10 Comparing mean power production between MPC and Vestas controller for the third case. 55
5.11 Pitch activity shown as pitch distance traveled. Comparing the MPC with the Vestas controller for the third case. 56
5.12 Averege time per iteration and median time for the simulations of the third case. 56

List of Figures

2.1 Illustration of airflow through a wind turbine [4]. 7
2.2 Main components of a wind turbine [16]. 8
2.3 The Vestas Multi-rotor wind turbine[21] 9
2.4 Illustration of weight, cost, and power production of a SRT. Current trends in solid lines and extrapolated in dotted lines [24]. 10
2.5 Illustration of control regions for wind turbines. 13

3.1 A block diagram of the turbine model with an illustration of the MRT. 16
3.2 Front view illustration of the MRT [21]. 16
3.3 A block diagram of the turbine model. 17
3.4 Structure model [21]. 22
3.5 Wind field setup [21]. 32

4.1 Illustration of a feasible region shown as the white area [15]. 35
4.2 Illustration of the feedback principle 36
4.3 Illustration of the MPC system 37
4.4 Illustration of the MPC principle [9]. 38
4.5 Polynomial approximation for state profile across a finite element [5]. 41

6.1 Power production for both MPC and Vestas at 10 m/s 59

6.2 Plots of the arm rotation state values for the upper section tower arms. Comparing the objective functions from the second and third case with wind speed 20 m/s. 61
6.3 Comparing loads between Vestas controller and MPC for case number two at 8 m/s wind speed. 62
6.4 Pitch activity compared for Vestas and MPC with the second case objective function at 16 m/s wind speed. 63
6.5 Comparison of pitch activity between MPC and Vestas controller for 8 m/s wind speed 64

A.1 Coefficients for C_p and C_t polynomial 72

Preface

This master's thesis is submitted as a part of the requirements for the master degree at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology. The work presented in this thesis has been carried out under the supervision of Post.Doc. Leif Erik Andersson and Prof. Lars Struen Imsland. The thesis was also partially supervised by Kim Hylling Sørensen and Jesper Sandberg Thomsen from Vestas.

This master's thesis is a continuation of the project thesis I conducted during the autumn of 2019. Vestas have supplied a multi-rotor control challenge [21] with a model and results to compare the findings and material conducted in this thesis. The implementation of the model and controller is carried out in Matlab with CasADi, an open-source tool for nonlinear optimization [2].

Unless otherwise stated, all figures and illustrations have been created by the author.

I would like express my gratitude to Leif Erik Andersson for exceptional supervision during both the project thesis and the master thesis. His support and availability has been impeccable. I would also like to thank Lars Struen Imsland for his support and feedback, Kim Hylling Sørensen and Jesper Sandberg Thomsen for all their knowledge

about wind turbines and turbine control. Their wisdom has been of great use to me.

Finally, I would like to express my gratitude to my dear friend Erik R. Rønnestad. The cooperation during 5 years of studies and especially the time spent on this master project has been a joyful experience.

Jørgen Urdal

Trondheim, June 2020

Chapter 1

Introduction

1.1 Motivation

December 2019 the European Commission unveiled the European Green Deal. This deal states that the European Union (EU) will by 2050 have no net emissions of greenhouse gases [8]. In other words, the EU will become climate neutral in 2050. Wind energy supplies 14% of the European electricity and is expected to grow to 50% in 2050 [25]. Achieving this goal means there must be a significant growth in wind farms around Europe and with this comes the challenges in up-scaling the wind turbines. Scaling up the wind turbines will increase the power generation and utilize space more efficiently compared to installing smaller wind turbines. But it also comes with a large increase in costs. An alternative to up-scaling wind turbines is the multi-rotor wind turbine (MRT). The principle behind MRT is to place several turbines on a single tower structure.

MRTs have been around for a long time, but in recent time have been given some focus with the challenges of up-scaling wind turbines. To help motivate the research

of MRTs, Vestas has created the "Multi-rotor wind turbine control challenge". This challenge will serve as a benchmark for advanced control development and motivate others to test out different control systems of MRTs [21]. The control objective is to maximize power generation, minimize the fatigue loads on the support structure, and not overuse the pitch activity.

1.2 Objective and considerations

The objective of this master thesis project is to implement a MPC to maximize power generation, limit the use of pitch activity, and minimize load fatigue on the support structure. The result will be compared with simulation results with the controller from the Vestas paper [21]. The paper presents two controllers, the benchmark controller (BMC) and the Vestas controller. It is expected that the MPC will outperform of these controllers, in both power generation and fatigue load minimization.

This thesis was written in parallel with Erik R. Rønnestad's master thesis [18]. The objective in Erik's thesis is to implement a moving horizon estimator (MHE) to estimate the model states and wind. These estimations can be used as feedback for control systems and MPC is highly dependent on good state estimations. Due to the nature of the relationship between the two theses and the collaboration on some coinciding elements, similarities might be found.

1.3 Previous work

This master thesis project is based on the project thesis [22], finished December 2019. Both these projects have the same objective and the project thesis serves as a good starting point for the master project as the same model and general methods were used. The results of the project thesis were, however, not good. The loads were several magnitudes worse than the Vestas controller. However, the results was not compared over a wider range of wind speeds and no conclusion could be drawn. Given the poor results from the optimization, it was believed to be an error either in the model or

somewhere in the load calculations.

The biggest challenge was the slow computation time. Simulations took much of the time, and identifying the improvement potentials became a prioritizing element in the end to get a better start in the master thesis project. Identified improvement potential included:

- Interpolation for the C_p and C_t values are slow. A polynomial made to fit the table is much faster.
- Stiffness in the tower model was a big contributor to the slow computation time. Model reduction was considered an option.
- Linearization of the turbine model might make it faster. Investigation into the accuracy of a linearized model should be performed.

Overall, the project thesis provided useful information, even though the results did not meet the goals.

1.4 Contributions

The main contributions of the work presented in this thesis are as follows:

- Combining Vestas multi-rotor wind turbine tower structure with four 5MW-NREL turbines
- Development of MPC for multi-rotor wind turbine using collocation method with an IPOPT solver.
- Comparison of performance between conventional PI controllers and MPC

The Vestas multi-rotor wind turbine tower structure was combined with four 5MW-NREL turbines in Matlab with CasADi. CasADi is an open-source tool for nonlinear optimization [2]. A MPC algorithm was implemented using collocation method as the integrator and IPOPT as solver. Interior Point Optimizer (IPOPT) is a general purpose nonlinear programming solver [5].

1.5 Outline

The report is organized as follows. In Chapter 2, a literature review of relevant topics are presented. In Chapter 3 the mathematical model is detailed to describe the MRT that us used in the simulations. Chapter 4 the methodology of the project is presented, giving insight into the control system and the optimization. The results of the simulations are presented in Chapter 5. Here the performance of the MPC is compared with the results from the Vestas simulations. In Chapter 6, the results from Chapter 5 are discussed, giving some insight into what happened with the results. The report finishes off with Chapter 7, where a conclusion is reached and future work to improve the controller is presented.

Chapter 2

Literature review

This chapter will present the literature review and introduce some important concepts in the topic of wind turbines. It begins with basic principles behind wind energy in Section 2.1, and wind turbines in Section 2.2, before an introduction of the multi-rotor wind turbine is given in Section 2.3.

Section 2.4 gives insight into control systems for wind turbines. The literature on control systems for MRT are in short supply, and the author could not find any documentation on MPC for MRT systems. The presented MPC for MRT is, therefore, novel. As there are no literature on MPC for MRT a literature review of MPC for SRT was carried out. At the end of this chapter, in Section 2.5, the two controller presented in [21] is detailed to give some insight in how they operate.

2.1 Wind energy

Wind is created by temperature differences in the air. Because of the uneven terrain on the surface of the earth, the air will be heated with varying effect. Hot air rises to

the outer layer of the atmosphere because it is lighter than cold air, and in the outer layers it will move towards the poles. At the poles, the air is cooled and drops towards the surface. This creates a return flow of cool air that is coming from the polar areas. This effect together with the Coriolis effect from the rotation of the earth, creates the air circulation that we call wind.
The wind that can be utilized for wind turbines lies in the lower levels of the atmosphere. Here turbulent wind flows occur because frictional forces and obstacles from uneven terrain, creating spikes in the wind flow called turbulence. Turbulence in wind has little effect on power production but it has a major impact on aerodynamic loads [4].

Kinetic energy P in wind flowing through an area A is expressed as

$$P = \frac{1}{2}\rho A V^3, \tag{2.1}$$

where ρ is the density of air and V is the wind speed [4].
Capturing the kinetic energy in the wind is derived from momentum theory. The mass flow rate of wind must be the same upstream as it is downstream of the turbine. Giving the equation

$$\rho A_\infty V = \rho A_D V_D = \rho A_{-\infty} V_{-\infty}, \tag{2.2}$$

where V and A_∞ is the upstream wind speed and area, V_D and A_D is the wind speed and area at the turbine, and $V_{-\infty}$ and $A_{-\infty}$ is the wind speed and area downstream. Figure 2.1 shows the principle of airflow through a wind turbine.

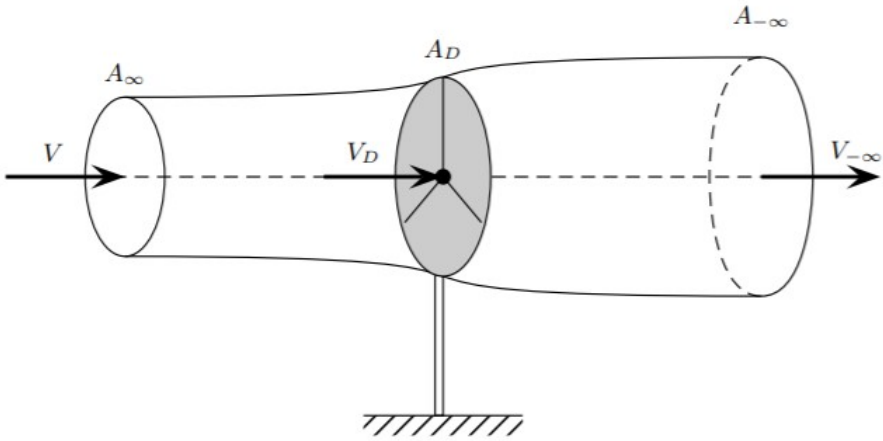


Figure 2.1: Illustration of airflow through a wind turbine [4].

2.2 Wind turbines

There are two types of wind turbines, vertical-axis and horizontal-axis wind turbines. The most common of these two are the horizontal-axis wind turbines. The reason for this is the horizontal-axis turbine can be placed on a tower utilizing larger wind speeds than the wind speed on the ground where a vertical-axis wind turbine would be placed. These advantages are only gained on larger scale wind turbines for power generation, but this is the focus in this project.

A rotor with airfoil-shaped blades capture the kinetic energy from the wind and transform it into rotational energy in a low-speed shaft. It is then transferred via a gearbox to a high speed-shaft into a generator to produce electrical power.

The pitch angle of the blades can be regulated with a pitch motor to maximize the efficiency of the power generation as well as working as a safeguard to satisfy electrical and mechanical load limits. A yaw drive and motor operate to make sure the wind turbine is facing optimally the wind direction. These functions are regulated with a controller with wind direction and wind speed as input, measured with an anemometer

and wind vane. Figure 2.2 shows the main components on the wind turbine.

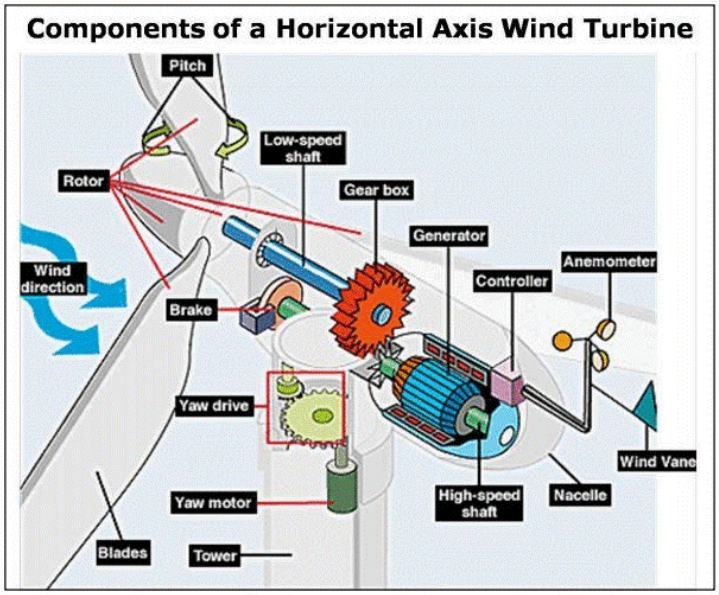


Figure 2.2: Main components of a wind turbine [16].

2.3 Multi-rotor wind turbines

In 2016, Vestas built a MRT and installed it at a test site in Roskilde, Denmark [23]. For a period of two and a half years the testing of the MRT observed several technical benefits compared to a SRT. Most notably of the benefits was the 1.5% power gain in annual energy production due to interaction between the rotors. Below in figure 2.3 the MRT can be seen at the testing site in Roskilde.

Figure 2.3: The Vestas Multi-rotor wind turbine[21]

The power gain is related to the MRT being faster in reaching the nominal power rating. It was also concluded that the wake recovery was faster. Wake recovery is the recovery in the kinetic energy of the wind behind the wind turbine. With increased wake recovery the wind turbines could be placed closer together in a wind farm.

As the size of wind turbines has grown significantly to increase the power gain in the last decades, the conventional SRT design comes with a large increase of cost in materials. An illustration of the weight, cost, and power production can be seen in the Figure 2.4.

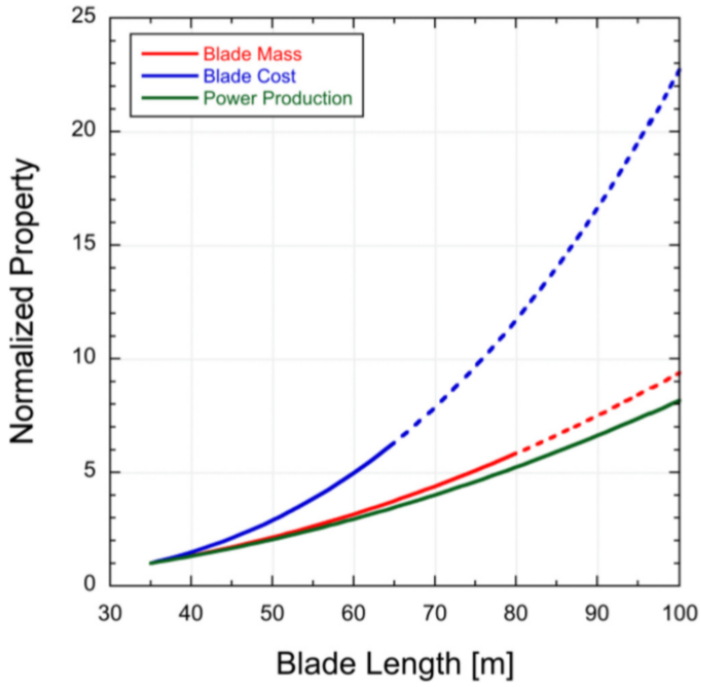


Figure 2.4: Illustration of weight, cost, and power production of a SRT. Current trends in solid lines and extrapolated in dotted lines [24].

The weight increases by the cube of the radius of the rotor, while the power increases by the square [21]. An alternative to the SRT is the MRT design. This design utilizes several rotors on the same support structure. According to the square-cube law, when the surface area increases the volume will grow faster. With n rotors on the MRT the weight can be calculated as $\frac{1}{\sqrt{n}}$ compared to the weight of a single rotor. This makes the MRT a suitable alternative to constructing large and expensive SRT. With weight reduction there will also be other advantages such as reduced transportation and installation costs. Having several turbines will create a redundancy system. Meaning that if one or more turbines must shut down, the other functioning turbines can continue producing power. This might be create higher loads on the tower structure,

but a controller can be implemented to prevent this. Another advantage with MRTs is the standardization of parts. This will bring reduction in production of parts and makes it easier to create back-up parts in large quantities.

2.4 Wind turbine control

As stated in the introduction to the literature review, the documentation on control systems for MRT is sparse. In the Vestas paper [21], a MRT control challenge was designed to motivate others to implement advanced control systems for MRT, with the control objective to minimize the fatigue on the support structure and maximize the generated power. Two controllers, the BMC and Vestas controller, was presented to act as a comparison for future control systems developed by others.

A design for linear MPC in SRT was developed in [12]. By tuning the system with trail-and-error approach, a comparison was performed comparing the conventional PID controller for above rated wind speeds and a baseline torque-based controller for below rated wind speeds. The results showed that above rated wind speeds, the MPC outperformed the PID controller, but below rated wind speeds the performance proved not be beneficial over the baseline torque-based control.

As long as the control system for SRT can handle multiple input multiple output (MIMO) systems, it should be able to control a MRT as well, even though the complexity increases.

In [11] a wind turbine is connected to an electrical storage device and the goal is to utilize MPC to maximize total energy generated while respecting the limits on the time derivative of power delivered to the grid. Here the controller is utilizing upstream wind measurements for other wind turbines to predict incoming wind, and control variables are blade pitch angle, generator torque, and charge/discharge rates for the storage device. From simulations the results demonstrates the ability to reject the disturbances from fast changes in wind speed, ensuring power gradients, with insignificant loss in

energy production.

An important part of MPC is to be able to predict what will happen in the future. As wind speed and wind turbulence influence the performance significantly, being able to predict these will create an advantage that can be utilized by the MPC. In [14], a field test investigating the use of continuous wave wind lidar (ZephIR) is used to measure the upwind approaching wind conditions. The spinner-integrated wind lidar was able to provide upwind measurements, such as:

1. Mean wind speed at hub height.
2. Vertical wind shear profiles.
3. Wind direction and yaw misalignment.
4. Power curves.
5. Lead time for warning of incoming gusts and wind direction changes, with a 5-10 s lead time depending on wind speed.

Most notably the last part about being able to have measurements on incoming gusts is a great advantage for the MPC and helps selecting a prediction horizon.

2.5 Benchmark controller and Vestas controller

As the MPC developed in this master thesis project will be compared with the Vestas results, an introduction to the two controllers is given here. The benchmark controller (BMC) is too complex to be described in this thesis, but the control strategy will give a description on how it works. The fully description on this controller can be found in [13]. The BLC is split in two different regions depending on the wind speed. Region 1 is called partial load, and region 2 is called full load. In figure 2.5 an illustration of these region can be seen.

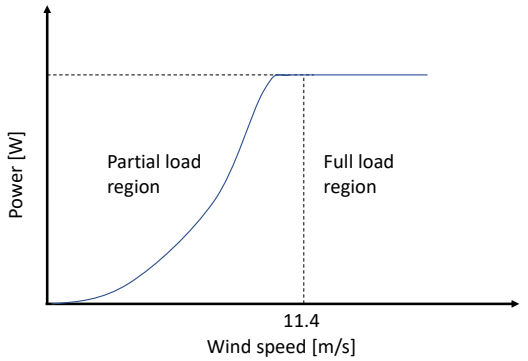


Figure 2.5: Illustration of control regions for wind turbines.

For region 1, the controller is a lookup-table for generator speed as input and generator power reference as output. To maximise power in the below rated region, the pitch angle of the blades is set to 0° at all times. In region 2 (rated region), the generator power reference is set to the rated power and the rotor speed is controlled by using the blade pitch angle. This is all controlled by a gain scheduled PI (proportional-integral) controller:

$$\beta_{ref} = K_P(\beta)\omega_{err} + K_I(\beta)\int \omega_{err}, \tag{2.3a}$$

$$K_{P/I}(\beta) = K_{P/I,0}\frac{\beta_2}{\beta_2 + \beta}. \tag{2.3b}$$

Where $\omega_{err} = \omega_{rated} - \omega$ and $K_{P/I}$ are the proportional and integral gains, $K_{P/I,0}$ is the base gain at $\beta = 0°$, and β_2 is the pitch angle where the pitch sensitivity is doubled. The controller presented in [13] will always operate at full power rating and in order to couple the two regions, the turbine control strategy has been modified by letting the dynamic power rating change the transition point between the regions.

The Vestas controller is described in [21] and is named baseline controller (BLC). This controller manipulates each individual rotor pitch reference by adding an additional pitch term. In practice this can be compared with adding a delta thrust ΔF_t to dampen the structural movements and the relation between the additional pitch term and delta thrust is

$$\Delta F_t = \frac{\partial F_t}{\partial \theta} \Delta \theta, \tag{2.4}$$

where $\Delta\theta$ is the additional pitch term. For the tower structure, using classic dynamic equation, this leads to:

$$F_t + \Delta F_t = M\ddot{y} + D\dot{y} + Ky, \tag{2.5}$$

with M,D, and K being the mass, dampening and stiffness matrix. The control action is a linear combination of tower displacement and velocities, given as

$$\Delta\theta = k_1\dot{y} + k_2 y, \tag{2.6}$$

with k_1 and k_2 as gains. More information can be found in [21].

Chapter 3

Mathematical model

This chapter will give insight in how the model for the MRT is built up. The mathematical model will be described by equations of motion, and can be divided into two sub-models, a turbine model and a tower structure model. The turbine model, which is described in Section 3.1, consists of four turbines. The turbines are connected to the tower structure, described in Section 3.2, through the force generated by the turbines. In the end of the chapter in Section 3.3, both models are combined and will give an overview of the full model.
In Figure 3.1 we can see a simplified model showing the setup for the controller, four turbines, and the tower structure. The controller gives an optimal pitch- and power-reference to the turbines which is connected to the tower structure by a force.

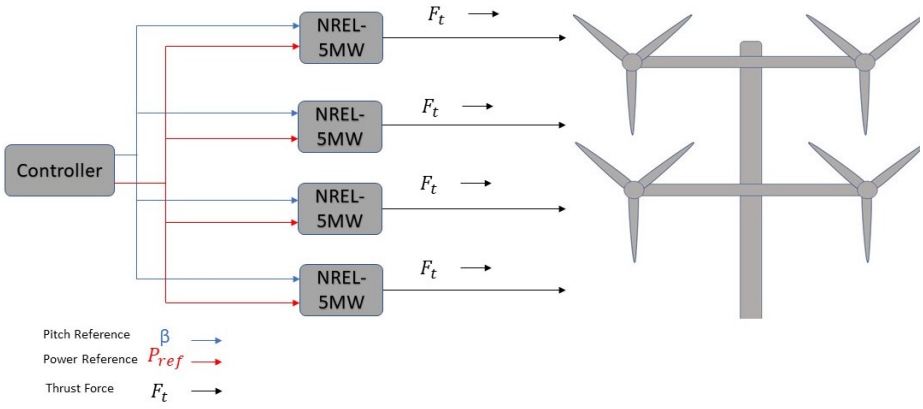


Figure 3.1: A block diagram of the turbine model with an illustration of the MRT.

This model is a theoretical model and the goal is to create research material for large wind turbines. Combining four 5 MW nacelles supported by a single tower structure will give a total of 20 MW rated power. This is a large construction and in Figure 3.2 an illustration on some important parameters can be seen in Table 3.1.

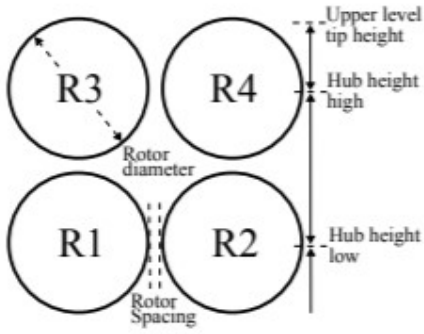


Figure 3.2: Front view illustration of the MRT [21].

Number of rotors	4
Rotor diameter	126 m
Rotor rated power	5 MW
Rated wind speed	11.4 $\frac{m}{s}$
Rotor spacing	12.6 m
Hub height low	90 m
Hub height high	228.6 m
Upper level tip height	291.6 m

Table 3.1: Table of MRT parameters [21].

3.1 Turbine model

The four turbines used for this project is based on the 5MW-NREL model is taken from SimWindFarm toolbox [1] and is described by [13] with modifications by [10]. This turbine model is built up by four different sub-models described in the sections below. These are an aerodynamic model of the turbine (Sec. 3.1.1), a drive train model (Sec. 3.1.2), a generator model (Sec. 3.1.3), and a pitch servo model (Sec. 3.1.4). To make the model run faster, some changes to has been implemented. These changes will be explained in the respective sections below. In Figure 3.3 a block diagram of the turbine model can be seen.

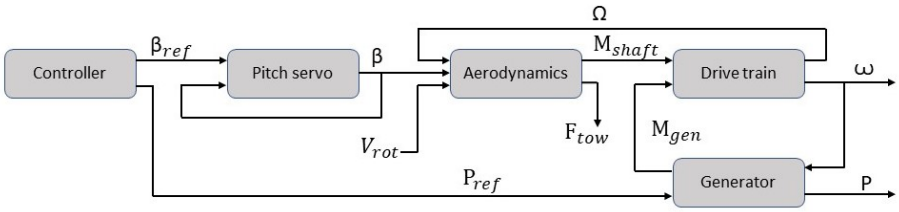


Figure 3.3: A block diagram of the turbine model.

Table 3.2 lists the important variables used in the turbine model with their respective units. The first 5 variables in the table represents the states of the turbine model. Table

3.3 lists the parameters and their value used for the turbine model.

Variable	Description	Unit
Ω	Rotor speed	$\frac{rad}{s}$
ω	Generator speed	$\frac{rad}{s}$
β	Pitch angle	degrees
M_{shaft}	Main shaft torque	Nm
M_{gen}	Generator torque	Nm
F_{tow}	Tower thrust	N
P	Power generated	W
P_{ref}	Generator power reference	W
β_{ref}	Pitch angle reference	degrees
V_{rot}	Average wind speed over the rotor	$\frac{m}{s}$

Table 3.2: Table of variables for the turbine model [1].

Parameter	Description	Value
A	Rotor area	12468 m^2
ρ	Air density	1.2231 $\frac{kg}{m^3}$
K_{shaft}	Torsion spring constant	867637000 $\frac{Nm}{rad}$
B_{shaft}	viscous friction	6215000 $\frac{Nm}{(rad/s)}$
I_{rot}	Rotor inertia	35444067 kgm^3
I_{gen}	Generator inertia	534.116 kgm^3
N	Gear ratio	97:1
R	Rotor radius	63 m
τ_{gen}	Time constant generator	0.1
τ_{β}	Time constant pitch angle	0.05

Table 3.3: Table of variables for the turbine model [1].

3.1.1 Aerodynamic

The aerodynamics of the turbine model is described by the thrust force F_{tow} and the main shaft torque M_{shaft}. These are heavily dependent on the aerodynamic coefficients

C_p and C_t which is determined by the tip speed ratio $\lambda = \frac{R\Omega}{v_{rot}}$ and the pitch angle β. The thrust force F_{tow} is calculated as

$$F_{tow} = \frac{1}{2} v_{rot}^2 \rho A C_t, \tag{3.1}$$

where v_{rot} is the average wind speed over the turbine rotor, ρ is the air density, and A is the area of the rotor. Serving as an input from the turbine to the tower structure, the thrust force is used to determine the loads on the structure. The main shaft torque M_{shaft} is calculated as

$$M_{shaft} = \frac{1}{2} v_{rot}^3 \rho A C_p \Omega^{-1}, \tag{3.2}$$

where Ω is the rotor speed. Both aerodynamic coefficients C_p and C_t are given in a look-up tables, but to make the model faster polynomials are created to represent the tables. To recreate the look-up tables as accurate as possible, the polynomials are made to fit to a smaller part of the look-up tables. This fit is based on the constraints and limits of the systems. Having a pitch angle β between $0° - 30°$ and the tip speed ratio λ between $4.5 - 17.5$ reduces the size of the table, and makes the polynomial fit more accurate. The polynomial can be seen in 3.3, the polynomial is the same for both C_p and C_t, and the coefficients can be seen in A.1.

$$\begin{aligned} C(\beta, \lambda) = {} & p_{00} + p_{10}\beta + p_{01}\lambda + p_{20}\beta^2 + p_{11}\beta\lambda \\ & + p_{02}\lambda^2 + p_{30}\beta^3 + p_{21}\beta^2\lambda + p_{12}\beta\lambda^2 \\ & + p_{03}\lambda^3 + p_{40}\beta^4 + p_{31}\beta^3\lambda + p_{22}\beta^2\lambda^2 \\ & + p_{13}\beta\lambda^3 + p_{04}\lambda^4 + p_{50}\beta^5 + p_{41}\beta^4\lambda \\ & + p_{32}\beta^3\lambda^2 + p_{23}\beta^2\lambda^3 + p_{14}\beta\lambda^4 \end{aligned} \tag{3.3}$$

This risk of using such polynomial is that the polynomial is fitted to these specific boundaries. Violating these boundaries can lead to inaccurate values.

3.1.2 Drive train

In the drive train the rotational energy is transferred from the low speed shaft, coming from the rotor, to the high speed shaft going to the generator. The low speed shaft and the high speed shaft is connected through a gearbox and the system can be represented as

$$\dot{\Omega} = \frac{1}{I_{rot}}(M_{shaft} - \phi K_{shaft} - \dot{\phi} B_{shaft}), \tag{3.4a}$$

$$\dot{\omega} = \frac{1}{I_{gen}}(-M_{gen} + \frac{1}{N}(\phi K_{shaft} + \dot{\phi} B_{shaft})), \tag{3.4b}$$

$$\dot{\phi} = \Omega - \frac{1}{N}\omega, \tag{3.4c}$$

where Ω is the rotor speed, ω is the generator speed, and ϕ is the torsion angle of the shaft. These three are the first of five states in the turbine model. K_{shaft} and B_{shaft} is torsion spring constant and viscous friction constant, while M_{gen} is the generator torque and N is the gear ratio. The drive train is connected to the generator with ω as input.

3.1.3 Generator

In the generator the electrical power is produced and the generator torque is calculated as

$$\dot{M}_{gen} = \frac{1}{\tau_{gen}}(\frac{P_{ref}}{\omega} - M_{gen}), \tag{3.5}$$

where τ_{gen} is a time constant and P_{ref} is the power reference input. The generator torque serves as the fourth state of the turbine model. Electrical power generated can be calculated as

$$P = \omega M_{gen}, \tag{3.6}$$

where P is given in Watt. This will be used at a later stage in the objective function Section 4.1.3.

3.1.4 Pitch servo

In the pitch actuator the pitch angle of the blades is controlled. This model is modified from the model used in [13] to a simpler first order system calculated as

$$\dot{\beta} = \frac{1}{\tau_\beta}(\beta_{ref} - \beta), \tag{3.7}$$

where β_{ref} is the pitch angle reference input from the controller and τ_β is a time constant. This was implemented to simplify the model without any significant influence on accuracy. Pitch angle β serves as the last of the five states in the turbine model.

This concludes the turbine model. In Section 3.3 a summary of the model combining four of these models with the tower structure model detailed in Section 3.2.

3.2 Tower structure model

In the Vestas control challenge [21] a model for the tower structure was developed. The same model is used for this project, and this section describes the model and gives some insight on how it was made. This is a simplified model with some assumptions taken:

- No gravitational loads are included due to limited effect on frequency response.
- No aerodynamic stiffening and damping from rotors.
- Only thrust forces are applied.
- Blades are not included in the model.
- No arm bending dynamics i.e. very stiff arms.

Figure 3.4 shows the geometry of the structure model with lengths, angles and masses, while Table 3.4 lists the values and units of these parameters.

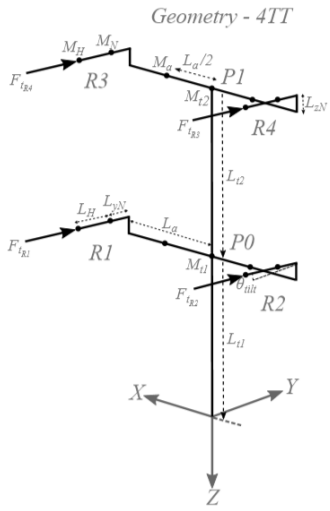


Figure 3.4: Structure model [21].

Parameter	Description	Value
L_a	Arm length	66.15 [m]
L_{zN}	Nacelle offset	1 [m]
L_{yN}	Length to nacelle CG	1.96256 [mm]
L_H	Length to hub	5.01910 [m]
L_{t1}	First tower section height	90 [m]
L_{t2}	Second tower section height	138.6 [m]
θ_{tilt}	Nacelle tilt angle	5 [deg]
M_H	Hub mass including blade mass	109930 [kg]
M_N	Nacelle mass	215000 [kg]
M_a	Arm mass	129029 [kg]
M_{t1}	First tower section mass	901690 [kg]
M_{t2}	Second tower section mass	567065 [kg]

Table 3.4: Parameter table for geometry of the structure[21].

The dynamics of the tower structure model can be described by ten degrees of freedom (DOF). These DOF, also called generalized coordinates, are used in the Lagrange equations to derive the equations of motion for the tower structure. The Lagrange equation is expressed as

$$\frac{d}{dt}\left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}_j}\right) - \frac{\partial \mathbf{T}}{\partial \mathbf{q}_j} + \frac{\partial \mathbf{V}}{\partial \mathbf{q}_j} = \mathbf{Q}_j^{(n)}, \quad j = 1, 2, ..., n, \tag{3.8}$$

where $\mathbf{T}$ is the kinetic energy, $\mathbf{V}$ is the potential energy, $\mathbf{q}$ is the generalized coordinates, and $\mathbf{Q}$ is the non-conservative generalized forces. In table 3.5 we can see the ten generalized coordinates for this structure with a description. These ten generalized coordinates along with their derivatives will be the 20 states for the tower structure.

Parameter	Description
θ_{xP0}	Lower tower section fore-aft bending
θ_{xP1}	Upper tower section fore-aft bending
u_{yP0}	Lower tower translation
u_{yP1}	Upper tower translation
θ_{zP0}	Lower tower torsion
θ_{zP1}	Upper tower torsion
θ_{xR1}	Arm torsion
θ_{xR2}	Arm torsion
θ_{xR3}	Arm torsion
θ_{xR4}	Arm torsion

Table 3.5: Description and parameters for the ten degrees of freedom

The non-conservative forces $\mathbf{Q}$ are the external forces acting on the tower structure. For this project, the only forces acting on the tower are the trust forces coming from the turbines. $\mathbf{Q}$ is calculated as

$$\mathbf{Q}_j^{(n)} = \sum_k \left(\mathbf{F}_{xk} \frac{\partial \mathbf{x}_k}{\partial \mathbf{q}_j} + \mathbf{F}_{yk} \frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_j} + \mathbf{F}_{zk} \frac{\partial \mathbf{z}_k}{\partial \mathbf{q}_j} \right), \tag{3.9}$$

where $\mathbf{F}_{xk}$, $\mathbf{F}_{yk}$ and $\mathbf{F}_{zk}$ are the external forces acting on the system, and k is the amount of forces acting on the system. Adding equations for $\mathbf{T}$, $\mathbf{V}$, and $\mathbf{Q}$ into (3.8) while linearizing around a steady state, gives the second order linear equations of motion

$$\mathbf{M}\ddot{q} + \mathbf{C}\dot{q} + \mathbf{K}q = \mathbf{F} \tag{3.10}$$

Here $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix, and $\mathbf{F}$ is the force vector. Damping matrix $\mathbf{C}$ is based on a Rayleigh type damping model. In

addition, $\mathbf{C}$ is given as

$$\mathbf{C} = \beta \mathbf{K}, \tag{3.11}$$

where β is the stiffness proportional Rayleigh parameter. $\beta = 0.12$ is selected to allow for structural and aerodynamic damping. Rewriting (3.10) on first-order form gives the state-space representation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{3.12}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \tag{3.13}$$

with $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}]$ as state vector, $\mathbf{u} = [F_{tR1}, F_{tR2}, F_{tR3}, F_{tR4}]^T$ as the input vector with the tower thrust forces. The state matrix $\mathbf{A}$ is given as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}, \tag{3.14}$$

where $\mathbf{I}$ is the identity matrix. The generalized coordinates $\mathbf{q}$ is given as the vector

$$\mathbf{q} = \left[\theta_{zP0}, \theta_{zP1}, u_{yP0}, u_{yP1}, \theta_{xP0}, \theta_{xP1}, \theta_{xR1}, \theta_{xR2}, \theta_{xR3}, \theta_{xR4}\right]^T \tag{3.15}$$

State-space representation is a good way to represent the tower structure as it is easy to use in calculations and organizes a large amount of states in a straightforwardly way. This becomes necessary as the tower structure have in total 20 states to keep track on.

3.2.1 Load calculation

The loads acting on the tower structure are calculated as

$$\mathbf{F} = \mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x}, \tag{3.16}$$

where **M** is the mass matrix and **K** is the stiffness matrix of the tower structure. The result of this calculation is a load vector **F** with two forces and eight moments that can be seen in table 3.6.

Parameter	Description	Unit
$M_{yaw,z,1}$	Yaw moment Mz lower tower	Nm
$M_{yaw,z,2}$	Yaw moment Mz upper tower	Nm
F_1	Force F lower tower	N
F_2	Force F upper tower	N
$M_{x,1}$	Moment Mx lower tower	Nm
$M_{x,2}$	Moment Mx upper tower	Nm
$M_{arm1,x}$	Arm 1 root moment Mx	Nm
$M_{arm2,x}$	Arm 2 root moment Mx	Nm
$M_{arm3,x}$	Arm 3 root moment Mx	Nm
$M_{arm4,x}$	Arm 4 root moment Mx	Nm

Table 3.6: Table of moments and forces acting on the tower structure

An important part of the optimization is to minimize these loads. To evaluate the fatigue on the structure these loads contributes, a parameter of high interest is named tower root moment can be calculated as

$$M_{TwrR} = F_1 L_{t1} + F_2(L_{t1} + L_{t2}) + M_{x,1} + M_{x,2} + M_{arm1,x} + M_{arm2,x} + M_{arm3,x} + M_{arm4,x} \tag{3.17}$$

This parameter is of high importance as it captures a lot of the tower dynamics and represents critical mechanical properties. Along with the parameters in Table 3.6, these parameters are used to calculate the Damage Equivalent Loads (DEL). These DEL's are used to evaluate the fatigue loads on the tower structure, and the goal is to keep these as low as possible while maintaining power production. DEL's are calculated using a rainflow-counting algorithm that was provided by Vestas. The concept is based on calculating the stress coming from cycles of varying loads on the structure but going into detail about this is not part of the scope of this project.

3.3 Combining the models

To create the full model of the multi-rotor wind turbine these models must be combined. Combining four of the turbine models along with the tower structural model makes up the full model. This gives a total of 40 states listed in Table 3.8. The 20 first states is the four turbine models with 5 states each, and the last 20 states represents the tower structure. Each turbine has two inputs, giving a total of 8 inputs for the full model, and can be seen in Table 3.7. As the turbine model is a nonlinear model on the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ and the tower structure is on state space form, they are combined as

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{f}(\mathbf{x}, \mathbf{u}) \end{bmatrix}. \tag{3.18}$$

When combining the linear tower structure model and the nonlinear turbine model, it will result in a nonlinear model. The nonlinear model is more complex than a linear model and will require a longer run time to simulate.

Nr.	Variable	Description	Nr.	Variable	Description
1	$P_{ref,1}$	Power reference	5	$P_{ref,3}$	Power reference
2	$\beta_{ref,1}$	Pitch reference	6	$\beta_{ref,3}$	Pitch reference
3	$P_{ref,2}$	Power reference	7	$P_{ref,4}$	Power reference
4	$\beta_{ref,2}$	Pitch reference	8	$\beta_{ref,4}$	Pitch reference

Table 3.7: Table of inputs for the full model. Consisting of power reference and pitch reference.

Nr.	Variable	Description
1	Ω_i	Rotor speed
2	ω_i	Generator speed
3	ϕ_i	Shaft torsion angle
4	$M_{gen,i}$	Generator torque
5	β_i	Pitch angle
21	$\theta_{z,P0}$	Tower torsion (lower section)
22	$\theta_{z,P1}$	Tower torsion (upper section)
23	$u_{y,P0}$	Translation (lower section)
24	$u_{y,P1}$	Translation (upper section)
25	$\theta_{x,P0}$	Rotation (lower section)
26	$\theta_{x,P1}$	Rotation (upper section)
27	$\theta_{x,R1}$	Arm rotation
28	$\theta_{x,R2}$	Arm rotation
29	$\theta_{x,R3}$	Arm rotation
30	$\theta_{x,R4}$	Arm rotation
31	$\dot{\theta}_{z,P0}$	Angular velocity z (lower section)
32	$\dot{\theta}_{z,P1}$	Angular velocity z (upper section)
33	$\dot{u}_{y,P0}$	Velocity (lower section)
34	$\dot{u}_{y,P1}$	Velocity (upper section)
35	$\dot{\theta}_{x,P0}$	Angular velocity x (lower section)
36	$\dot{\theta}_{x,P1}$	Angular velocity x (upper section)
37	$\dot{\theta}_{x,R1}$	Angular velocity of arm
38	$\dot{\theta}_{x,R2}$	Angular velocity of arm
39	$\dot{\theta}_{x,R3}$	Angular velocity of arm
40	$\dot{\theta}_{x,R4}$	Angular velocity of arm

Table 3.8: The 40 states that represents the full model. The first 20 is from the turbine model, with 5 for each of the four models($i = 1 : 4$). The 20 last states represents the tower structure.

3.3.1 Constraints

Constraints are added to the model to make sure the system behaves realistically. This is important as the simulation might not give valid results if able to behave freely. Adding these constraints have two intentions:

1. Making sure the turbine model does not break down by exceeding the limit on how much it can handle.
2. Making sure the turbine model does not give unrealistic values.

Unrealistic values are for example negative rotor speed or negative generator torque. By giving constraints this behavior can be prevented. Constraints are given to both input and states on the form:

$$x_{min} < x < x_{max}, \tag{3.19}$$

for states x and for inputs u

$$u_{min} < u < u_{max}. \tag{3.20}$$

Where x_{min},x_{max} are the lower and upper limits for the state values, u_{min} and u_{max} is the lower and upper limits for the input values. For the turbine states there are constraints on the rotor speed, generator speed, generator torque and pitch angle. Although the pitch angle can vary between $-5°$ and $90°$, in practice it lies between $0°$ and $30°$. This specific constraint also keeps the pitch angle inside the range of the fitted polynomial for the C_p and C_t. This gives:

$$x_{min,turb} = \begin{bmatrix} 0 & 0 & -Inf & 0 & 0 \end{bmatrix} \tag{3.21}$$

$$x_{max,turb} = \begin{bmatrix} 1.2671 & 122.9096 & Inf & 4.745 \cdot 10^5 & 30 \end{bmatrix}, \tag{3.22}$$

for each of the four turbines. The shaft torsion angle is the only turbine state that is not restricted. There are also no constraints on the tower states, as these are freely to move in any direction. In practice this means to make the constraints go from minus infinity to infinity, allowing any value.

There also limits to how fast the system can respond to changes, with constraints given to the rate of change for the generator torque and pitch angle. Given as:

$$\dot{x}_{min,turb} = \begin{bmatrix} -1 \cdot 10^{10} & -1 \cdot 10^{10} & -1 \cdot 10^{10} & -15000 & -8 \end{bmatrix} \tag{3.23}$$

$$\dot{x}_{max,turb} = \begin{bmatrix} 1 \cdot 10^{10} & 1 \cdot 10^{10} & 1 \cdot 10^{10} & 15000 & 8 \end{bmatrix}, \tag{3.24}$$

where the change of rate is restricted to 8° and 15000 Nm per second. To avoid any limitations on the other turbine states and tower states, $-1 \cdot 10^{10}$ and $1 \cdot 10^{10}$ are used. These are considered large enough to not limit the rate of change for any of the states.

The input constraints are there to make sure the controller always give realistic inputs to both the power reference and pitch angle reference. This gives u_{min} with all zeros and

$$u_{max} = \begin{bmatrix} Power_{rated}, & 30, & Power_{rated}, & 30, & Power_{rated}, & 30, & Power_{rated}, & 30 \end{bmatrix}, \tag{3.25}$$

where $Power_{rated}$ is the maximum rated power, given as $5.2966 \cdot 10^6$ Watt.

This concludes the chapter on the mathematical model and next section will give some insight into the wind field used in the simulations.

3.4 Wind field

To get the same conditions used in the Simulink model from Vestas, the same wind field is used in the simulations. The wind field is generated from functions taken from [20] and is based on Taylor's frozen turbulence hypothesis method. To be able to use the wind field generator, the wind turbines were placed in a row with 10% of the rotor diameter distance between each other and this can be seen in Figure 3.5. In reality this will not represent the layout of the multi-rotor and a wind shear contribution of 15% is added to the top rotors to represent the height difference between the upper and lower rotors, as the upper rotors will experience a higher wind speed.

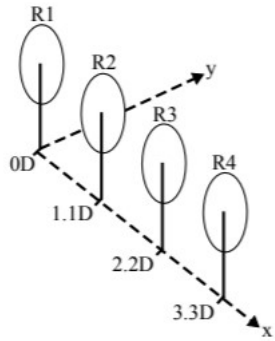


Figure 3.5: Wind field setup [21].

Simulating with several different wind speeds, a more comprehensive evaluation of the model can be made. The simulation will be performed with wind fields in the range from 4 to 20 m/s. It is also assumed that the wind is known in advance, meaning that future wind is used in the prediction horizon. To predict this wind, a lidar system can be utilized, but details of this type of system is not part of the scope in this project and more can be read in [19] and [14]. Predicting the wind is expected to increase the performance of the MPC in terms of reducing the loads. Wind turbulence greatly affect the fatigue loads, and knowing when these gusts will appear makes it easier to counteract.

Chapter 4

Methodology

This chapter will go into detail on how the problem was solved and what methods where used. In order to understand these methods, some important principles must be explained. The chapter will first start with describing how the control system works and how it is designed for this particular problem in Section 4.1. Section 4.2 will explain which optimizer is used and some details on how it works.

4.1 Control system

There are many ways to control a system to achieve the desired result. The methods used to control a system is dependent on the complexity of the system and the control objective. One approach is model predictive control (MPC). MPC is an optimization tool based on utilizing a model to predict the outcome, and give the best input to control the system. This section will give insight to what model predictive control is and how it's built up. First, basic optimization and feedback control will be introduced in Section 4.1.1. These are important parts of how MPC works. In section 4.1.2, an introduction

to MPC will be explained, before going more into detail about the objective function, in Section 4.1.3.

4.1.1 Introduction to basic optimization and feedback control

Optimization is used in a large variety of applications. Optimization is based on an algorithm trying to give the best possible result with the possibility to change certain variables inside the optimization problem, while maintaining some restrictions given [15]. There are three main ingredients in an optimization problem:

1. An objective function
2. Decision variables
3. Constraints

The objective function defines what should be minimized or maximized in the optimization problem. Another name for the objective function is cost function. It is a scalar function that represents the property we want to minimize. There can be several elements to optimize in a cost objective function, and these can be assigned a weight parameter to decide the value each element will affect the solution.

Decision variables is the variables that the optimization algorithm can vary to achieve the optimum solution. These are both the state variables and the input variables.

Constraints are used to restrict the possible solutions to the optimization problem. Constraints are divided into equality constraints and inequality constraints. These constraints define a region where a solution might lie, also called feasible region. The optimization problem is written on mathematical form as:

$$\min_{z \in \mathbb{R}} f(z) \tag{4.1}$$

subject to

$$c_i(z) = 0, \quad i \in \epsilon \qquad \text{(equality constraint)}, \tag{4.2}$$

$$c_i(z) \geqslant 0, \quad i \in \mathcal{I} \qquad \text{(inequality constraint)}, \tag{4.3}$$

where $f(z)$ is the objective function that will be minimized, and $c_i(z)$ is the constraints shown as both equality and inequality form [15]. In Figure 4.1 an illustration of this principle is shown. Here the constraints c_1 and c_2 defines the feasible region for the optimum solution x^*.

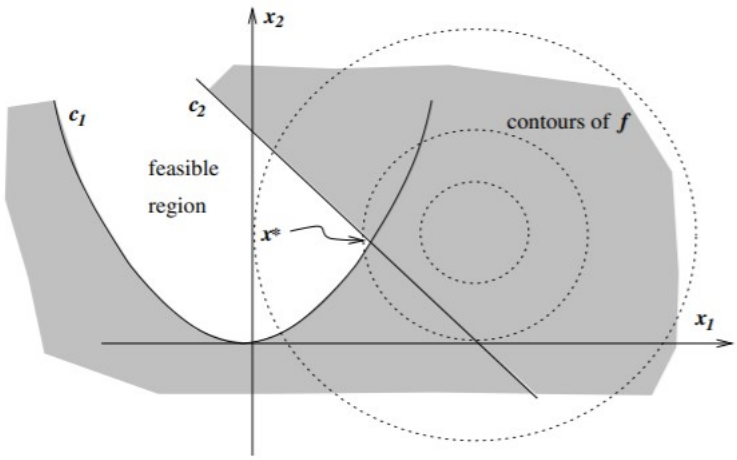


Figure 4.1: Illustration of a feasible region shown as the white area [15].

Feedback control is used where the systems output is measured and given as a feedback to the controller. In the controller, the measurements are compared to a given reference in which the difference is given as an error signal. The controller then corrects this error by giving the correct input. In Figure 4.2 an illustration of the feedback control principle is shown.

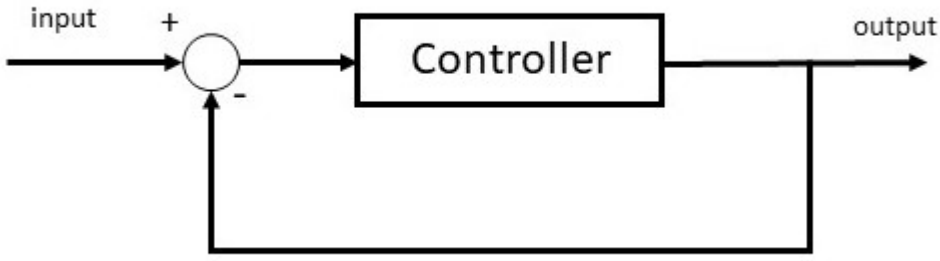


Figure 4.2: Illustration of the feedback principle

4.1.2 Model predictive control

Model predictive control is a method that combines feedback control with dynamic optimization [9]. When a model is dynamic it means that the decision variables are functions of time. Dynamic optimization is useful when dealing with system subject to frequent changes. MPC is a robust method that have several advantages compared to other control methods. It can handle multiple input and multiple output (MIMO) systems and can also handle constraints to the objective function. This is a good fit with the control problem given for this project, as it is a MIMO system and has several constraints in the objective function. In Figure 4.3 an overview of how the MPC works is shown.

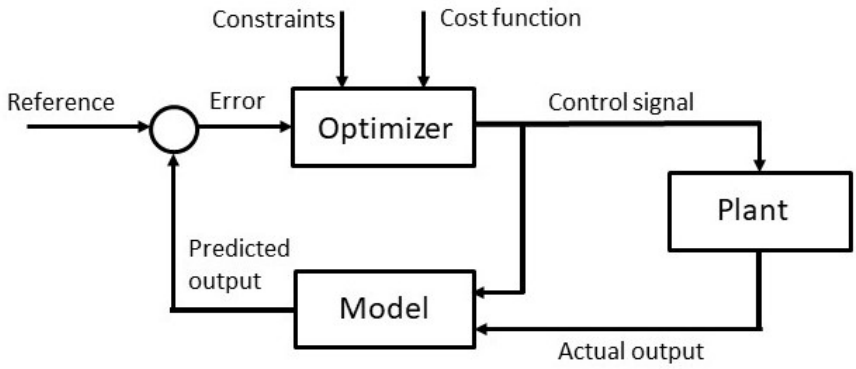


Figure 4.3: Illustration of the MPC system

A good explanation of the MPC principle is given by [7]:

Model predictive control is a form of control in which the current control action is obtained by solving, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant.

This principle is shown in Figure 4.4. The lower part illustrates the plant, where the blue dots and black lines represents the measured states and control inputs of the past, respectively. Each time a state is measured, a model of the plant simulates an optimization problem, predicting what will happen in a given time horizon, and finds the optimal input. This can be seen in the upper part of Figure 4.4, where the blue dots and red lines represents the predicted states and predicted input respectively. When the optimization is complete, the first input from the model is used as input in the plant, seen as the red line in the lower part. This repeats until the optimization is completed.

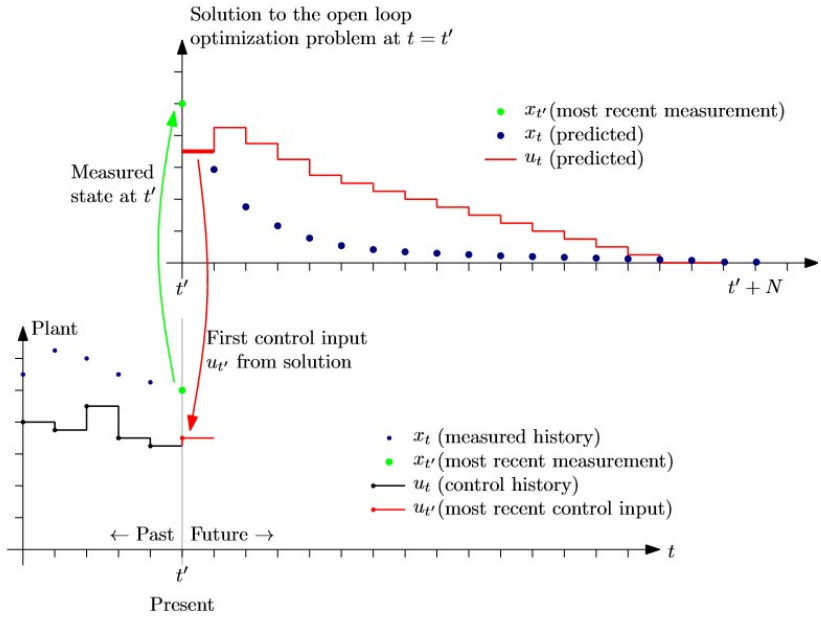


Figure 4.4: Illustration of the MPC principle [9].

4.1.3 Control objective

Arguably the most important part of the optimization is the control objective. The control objective states what is going to be optimized. For this project, the objective is to reduce load and pitch activity while following a power reference. The standard form for the objective function is

$$\min_{z \in \mathbb{R}^n} f(z) = \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^T Q_{t+1} x_{t+1} + \frac{1}{2} u_t^T R_t u_t, \tag{4.4}$$

where x is the states, u is the input, Q and R are weighting matrices. Minimizing this equation will result in the optimal solution. Selecting the most efficient objective function can be hard for complex systems, as there are many ways to represent the

objective. To fit with the control objective, this equation is rewritten as

$$\min_{z \in \mathbb{R}^n} f(z) = \sum_{t=0}^{N-1} q_1(P_{ref} - P)^2 + q_2(Loads)^2, \tag{4.5}$$

where P_{ref} is the power reference and P is the actual power produced, containing the power from the four turbines combined. *Loads* is the tower loads taken from (3.16), while q_1 and q_2 is weighting parameters. Each of the terms in the objective function is squared to get positive numbers. By expanding (3.6), the total power production P can be calculated as

$$P = \omega_1 M_{gen,1} + \omega_2 M_{gen,2} + \omega_3 M_{gen,3} + \omega_4 M_{gen,4} \tag{4.6}$$

Loads in (4.5) is on the form of a vector. As the result from this equation is scalar, a reshape of *Loads* is necessary. This is solved by taking the sum of all the loads.

The weight parameters q_1 and q_2 are used to tune the system. By increasing a weight parameter, more effort will go to reduce this term in the objective function. This is a trade-off between the terms in the objective function. By increasing q_1 compared to q_2 will result in focus to follow the power reference, and focus less on reducing loads. By simulations, these parameters will be tuned to find the best result. The objective function (4.5) served as a starting point in trying to achieve the best results. Several changes to the objective were implemented trying to improve the results. Different objective functions highly influence the results and computation time. Details on this will be presented in the results sections in Chapter 5.

4.2 Optimizer and integration method

The nonlinear solver fmincon used in the project thesis turned out to be slow and to handle this problem, CasADi is implemented. CasADi is an open-source tool for nonlinear optimization with a symbolic framework that facilitates rapid and efficient implementation of numerical optimal control methods [2]. Combining the model described in Chapter 3 with the constraints stated in Section 3.3.1 gives a non linear program (NLP) that needs to be optimized. The NLP solver used for this project is called Interior Point Optimizer (IPOPT). This is an open-source primal-dual interior point method that implements an interior point line search filter method to converge at a local solution to the NLP [15].

4.2.1 Collocation method

In the project thesis [22] the ode15s was chosen as the integration method trying to counter the stiffness of the model. Although this was better then some other ode functions like ode45, it was still slow. To speed things up, collocation method was chosen for this project.

Collocation methods can be presented as a special class of implicit Runge-Kutta (IRK) methods. The collocation method is based on approximating the state trajectories with polynomials. By dividing up the interval between state point z_i to z_{i+1} in the trajectory into a finite number of collocation points, polynomials can be fitted between each of these collocation points. This will then represent the trajectory of the system. Figure 4.5 shows a polynomial approximation for a state point.

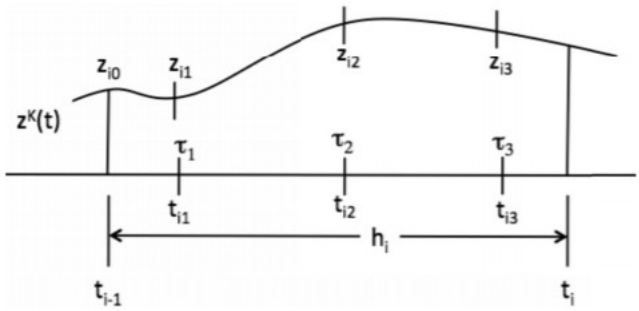


Figure 4.5: Polynomial approximation for state profile across a finite element [5].

To understand the collocation method we consider an ODE:

$$\frac{dz}{dt} = f(z(t), t),\ z(0) = z_0, \tag{4.7}$$

and develop the collocation method for this. A polynomial approximation of order $K + 1$ for the state $z(t)$, based on Lagrange interpolation polynomials, is presented as:

$$\left.\begin{array}{l} t = t_{i+1} + h_i\tau, \\ z^K(t) = \sum_{j=0}^{K} L_j(\tau) z_{ij}, \end{array}\right\} t\ \epsilon[t_{i-1}, t_i],\ \tau\ \epsilon[0, 1], \tag{4.8}$$

where

$$L_j(\tau) = \prod_{k=0,\neq}^{K} \frac{(\tau - \tau_k)}{(\tau_j - \tau_k)}, \tag{4.9}$$

for $\tau_0 = 0$, $\tau_j < \tau_{j+1}$, $j = 0, ..., K$ and h_i is the length of element i. This gives the property:

$$z^K(t_{ij}) = z_{ij}, \tag{4.10}$$

where

$$t_{ij} = t_{i-1} + \tau_j h_i. \tag{4.11}$$

With the same approach, the time derivative of the state is also represented with a Lagrange polynomial with K interpolation points. Leading to a Runga-Kutta representation of the differential state:

$$z^K(t) = z_{i-1} + h_i \sum_{j=1}^{K} \Omega_j(\tau)\dot{z}_{ij}, \tag{4.12}$$

where z_{i-1} is a coefficient representing the differential state at the start of element i, $\dot{z}_{ij}$ is the time derivative, and $\Omega_j(\tau)$ is a polynomial of order K. Ultimately this leads to the collocation equations:

$$\sum_{j=0}^{K} z_{ij} \frac{dL_j(\tau_k)}{d\tau} = h_i f(z_{ik}, t_{ik}),\ k = 1, ..., K, \tag{4.13}$$

for the Lagrange polynomial and for the Runga-Kutta method

$$\dot{z}_{ik} = f(z_{ik}, t_{ik}),\ k = 1, ..., K \tag{4.14a}$$

$$z_{ik} = z_{i-1} + h_i \sum_{j=1}^{K} \Omega_j(\tau_k)\dot{z}_{ij}, k = 1, ..., K. \tag{4.14b}$$

These collocation equations are now algebraic equations and can be merged with the NLP formulations and solved. More on this topic can be found in [5].

The collocation method script was not developed by the author, but was implemented by modifying a code given by the supervisor of this project. An example of direct collocation can be found in CasADi example pack [6].

Chapter 5

Results

This chapter will present the results from simulations performed on the model. It will start with listing the performance criteria, in Section 5.1, before presenting the simulation results in Section 5.2. The results will not be discussed in this chapter, but rather just state how the performance of the MPC is doing compared to the Vestas controller. The results are heavily dependent on selecting the right objective function, and the results of several different objective functions will be presented. In Chapter 6, a discussion of the result will be presented.

5.1 Performance criteria

To evaluate the performance of the system, a set of criteria must be made. These criteria are based on the paper from Vestas [21] and are listed below.

1. Follow power reference.
2. Reduce structural loads.
3. Pitch activity.
4. Run-time of simulations.

As with any power producing systems, the goal of the MRT is to produce as much power as possible. That is why it is important follow the power reference as close as possible. In the Vestas paper [21] it is stated that a power loss of maximum 3% is allowed compared to the base line controller.

Allowing the power production to fall with 3% is meant to allow a larger reduction in loads on the structure. Reducing the structural loads will increase the lifetime of the MRT and a longer lifetime will lead to an overall reduction in costs. In the Vestas paper [21] seven important moments are identified that will be used to evaluate the loads on the structure. These are the tower root moment (3.17), the yaw moments of the upper and lower tower, and the four bending moments of the arms. As mentioned in Section 3.2.1 these loads are presented as Damage Equivalent Loads (DEL).

Pitch activity given in accumulated pitch distance is also seen as a parameter to evaluate the lifetime of the turbines. In the Vestas paper [21] it is stated that an increase of 10% is allowed to achieve the control objective.

These three parameters will be compared with the results from the Vestas paper. The last parameter is the run-time of the optimization. Real-time optimizing requires fast run time. Taking note of the mean value for how long each iteration in the simulation takes, will give an indication on how fast the optimization is performed. To achieve real-time each iteration should stay below one second.

5.2 Simulation results

This section will present the results from the simulations. As the goal of this project is to compare with the results made by Vestas, these results will be presented in a similar way as in the paper [21]. The results will be presented in tables to give overview of the performance in a range of different wind speeds, and tables for each of the evaluation criteria will be shown. Plots of the results will not be shown in this section, but rather in the next chapter where the results will be discussed. All the results are based on a 200 second simulation time. The simulation time is selected to make the results comparable with the Vestas paper [21]. The same initial conditions from the Vestas simulations are also used. These are:

$$x_{init,turb} = \Big[0.5155,\ 50,\ 0,\ 0,\ 0,\Big] \tag{5.1}$$

and all zeros for the tower states. To get a more thorough evaluation of the performance, simulations are performed for nine different wind speeds, above and below the rated wind speed.

Three cases will be presented where different objective functions are tested. These are listed below and will be presented in the respective order. All the results are compared with the performance of the Vestas controller.

1. Power reference tracking and minimization of input activity.
2. Power reference tracking, minimize translation of the upper tower section, minimization of thrust force between turbines, and minimization of input activity.
3. Power reference tracking, minimize rotation of arm 3 and 4 (upper) and translation of the upper tower, minimization of thrust force between turbines, and minimization of input activity.

The process of selecting an objective function was based on a trail-and-error approach with the goal of improving the results step by step. For all simulations, a prediction horizon is selected to be 10 seconds. In [14] it is stated that lidar can give upwind

measurements on incoming gusts up to 10 seconds depending on the wind speed. Each of the result sections below will present the objective function used for the particular cases.

5.2.1 Case number 1

For the first case, the objective function was kept simple to serve as a basis to be able to pinpoint where the optimization is struggling to find the best end result. The objective function for these results are

$$L = q_1(P_{ref} - P)^2 + Q_1(U_k - U_{k-1})^2. \tag{5.2}$$

Here the optimization will mainly care about tracking the power reference and also limit the use of input. To evaluate the performance of the MPC, a comparison is presented in a table showing a percentage compared to the performance from the Vestas controller. As there are four evaluation criteria, there are also four tables representing the performance of the controller. Table 5.1 displays the DEL's, Table 5.2 shows the power production, Table 5.3 shows the pitch activity, and Table 5.4 details the computation time.

Damage Equivalent Loads (DEL)

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total**
Tower root moment	0.27	0.30	0.34	0.31	0.99	0.47	0.51	1.66	2.04	0.80
Yaw moment 1	0.19	0.28	0.28	0.56	0.82	0.39	0.40	0.81	1.16	0.58
Yaw moment 2	0.22	0.32	0.31	0.16	1.51	0.35	0.25	1.28	1.42	0.54
Arm moment 1	0.16	0.21	0.27	0.52	0.74	0.43	0.43	0.67	1.07	0.58
Arm moment 2	0.15	0.24	0.26	0.45	0.86	0.43	0.42	0.82	1.02	0.60
Arm moment 3	0.17	0.25	0.24	0.17	1.04	0.44	0.40	1.68	1.84	0.70
Arm moment 4	0.15	0.22	0.19	0.20	0.85	0.49	0.44	1.51	1.82	0.69

Table 5.1: Comparing DEL's between MPC and Vestas controller for the first case.

From the DEL's in Table 5.1 we can see that the MPC outperforms the Vestas controller for the lower wind speeds from 4 m/s to 10 m/s and middle wind speeds of 12 m/s to 16 m/s. There are some exceptions for the 12 m/s, both the yaw moment 2 and arm moment 3 is worse. For the higher wind speeds of 18 m/s and 20 m/s, we can see that the MPC is not performing well. For 20 m/s all the DEL's are above values of the Vestas controller, and the tower root moment is more than twice as large. For the power production, seen in Table 5.2, the overall performance is well above what the Vestas controller achieves. However, if we look at the power production for 10 m/s, it is below the maximum power loss allowed and only produce 89% power compared with the Vestas controller.

Mean power production

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total**
Power 1	1.18	0.84	1.10	0.96	1.07	1.06	1.10	1.14	1.14	1.08
Power 2	1.16	0.81	1.14	0.84	0.99	1.07	1.10	1.14	1.14	1.05
Power 3	1.75	1.14	1.04	0.90	1.24	1.16	1.13	1.11	1.11	1.11
Power 4	0.86	1.42	1.10	0.88	1.45	1.16	1.11	1.12	1.11	1.13

Table 5.2: Comparing mean power production between MPC and Vestas controller.

The pitch activity is presented as pitch distance traveled (PDT). From the results we see that the overall usage of pitch is below the Vestas controllers usage. There are however exceptions here as well. The wind speeds 8 m/s to 12 m/s all experience higher pitch activity than the Vestas controller.

Pitch distance traveled (PDT) [Degrees]										
	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total sum (ratio)**
PDT 1	0.32	0.24	0.80	3.35	1.19	0.75	0.44	0.61	0.70	874.98 (0.86)
PDT 2	0.17	0.07	1.56	1.70	1.02	0.69	0.46	0.56	0.47	810.24 (0.73)
PDT 3	0.03	0.06	1.59	0.27	1.58	0.42	0.17	0.38	0.44	599.60 (0.42)
PDT 4	0.06	0.10	1.38	0.40	0.81	0.53	0.25	0.41	0.49	554.91 (0.41)

Table 5.3: Pitch activity shown as pitch distance traveled. Comparing MPC with the Vestas controller for the first case.

Achieving real time optimization is an important part of the controller. Looking at the average time per iteration we can see that the lower wind speeds 4 m/s to 8 m/s and the higher wind speeds 16 m/s to 20 m/s the computation time is close to the threshold of 1 second per iteration. However, the middle wind speeds 10 m/s to 14 m/s have a much higher computation time, and are not close to this threshold. When we look at the median time, we can see that for the wind speeds that have high computation time the median is lower than the average. Meaning that there are some iterations that are slow that affects the entire run time. This case is especially valid for 14 m/s, where the average time is almost 30 seconds but the median time is 2.48 seconds.

Computation time									
	Wind speed [m/s]								
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
Average time per iteration [seconds]	3.22	3.00	3.50	22.56	40.80	29.64	5.83	3.10	2.51
Median time [seconds]	2.53	2.06	1.67	16.08	24.59	2.48	2.92	1.98	1.83

Table 5.4: Average time per iteration and median time for the simulations of the first case.

Overall the performance of the MPC is promising, and improvement potentials are identified. The next step is to modify the objective function, trying to improve the results.

5.2.2 Case number 2

Looking the the results in the previous section, it is clear that the yaw moment of the upper and lower sections of the structure had values that where too high for wind speeds 18 m/s and 20 m/s. Modifying the objective function trying to reduce the yaw moment 1 and 2 can be solved several ways which will be discussed in Chapter 6, and the new objective function resulted in:

$$\begin{aligned} L = q_1(P_{ref} - P)^2 + q_2 x_{24}^T x_{24} + q_3(F_{tow,1} - F_{tow,2})^2 \\ +q_3(F_{tow,3} - F_{tow,4})^2 + Q_1(U_k - U_{k+1})^2. \end{aligned} \tag{5.3}$$

Where x_{24} is the translation of the upper tower section, $F_{tow,1}$ and $F_{tow,2}$ are the thrust forces for the lower tower section, $F_{tow,3}$ and $F_{tow,4}$ are the thrust forces for the upper tower section, and q_2, q_3 are the new weight parameters. The idea behind minimizing the difference in thrust force between the opposite turbines is to reduce the yaw moment for the upper and lower section, and also reducing the variance for these values, leading to fewer cycles. By also adding state number 24, the translation of the upper tower section, to the objective function, the yaw moment should be reduced by quite a bit.

In Table 5.5 we can see the DEL comparison between the MPC with the new objective function and Vestas controller. For wind speeds from 4 m/s to 16 m/s we can see that most of the DEL's have been reduced. Especially the Yaw moment 2 for 12 m/s, which is reduced from 1.51 to 0.24. DEL reduction is also achieved for wind speeds 18 m/s and 20 m/s. The yaw moments for the upper and lower sections are reduced significantly and are well below the results from the Vestas controller. Similarly we can see that the tower root moment are also reduced, but not yet below the desired threshold. We can also see a reduction in the arm moments for these wind speeds. Although, the threshold for the upper section arms are still not good enough, they have been reduced. A further reduction in these loads is preferable, and in theory, the tower root moment

will also follow the trend of these reductions.

Damage Equivalent Loads (DEL)

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total**
Tower root moment	0.28	0.31	0.34	0.19	0.67	0.43	0.42	1.44	1.97	0.69
Yaw moment 1	0.16	0.22	0.22	0.11	0.22	0.12	0.19	0.45	0.57	0.58
Yaw moment 2	0.19	0.25	0.24	0.09	0.24	0.10	0.10	0.36	0.55	0.18
Arm moment 1	0.14	0.19	0.25	0.14	0.55	0.30	0.37	0.56	0.69	0.41
Arm moment 2	0.15	0.23	0.24	0.15	0.70	0.27	0.27	0.73	0.81	0.44
Arm moment 3	0.16	0.24	0.23	0.13	0.33	0.34	0.29	1.16	1.43	0.49
Arm moment 4	0.15	0.19	0.18	0.14	0.39	0.34	0.32	1.07	1.56	0.51

Table 5.5: Comparing DEL's between MPC and Vestas controller for the second case.

In Table 5.6 the mean power production comparison between the MPC and the Vestas controller is displayed. Looking at the results we can see that the power production had an overall decrease of 6% for the first turbine and 2% for the other turbines. This is to be expected as the objective function is now more focused on reducing loads. The issue with low power production for wind speed 10 m/s is still present.

Mean power production

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total**
Power 1	1.16	0.82	1.09	0.92	0.88	0.96	1.10	1.13	1.13	1.02
Power 2	1.18	0.81	1.11	0.85	1.02	0.95	1.07	1.14	1.13	1.03
Power 3	1.69	1.15	1.05	0.90	1.19	1.11	1.12	1.11	1.10	1.09
Power 4	0.86	1.37	1.07	0.88	1.35	1.11	1.11	1.11	1.10	1.11

Table 5.6: Comparing mean power production between MPC and Vestas controller for the second case.

Table 5.7 is showing the pitch activity compared to the Vestas controller. Although the overall pitch activity has decreased, some major changes has occurred. Specifically 8 m/s wind speed has seen a large increase and 10 m/s wind speed has seen a large decrease.

Pitch distance traveled (PDT) [Degrees]

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total sum (ratio)**
PDT 1	0.75	0.87	2.10	1.20	1.02	0.75	0.46	0.53	0.39	701.75(0.69)
PDT 2	0.31	0.93	3.24	0.54	0.92	0.61	0.45	0.72	0.45	693.99(0.62)
PDT 3	0.19	0.16	1.60	0.36	0.46	0.15	0.28	0.28	0.50	447.08(0.31)
PDT 4	0.06	0.80	2.57	0.40	0.61	0.57	0.11	0.39	0.58	504.44(0.37)

Table 5.7: Pitch activity shown as pitch distance traveled. Comparing MPC with the Vestas controller for the second case.

In Table 5.8 we can see that the computation time has increased for all the wind speed simulations, and are now far from being real time optimization. Increasing the complexity of the objective function has lead to longer calculations. Looking at the median time we can see that for most of the wind speeds it correlates, meaning that the average time does give a more correct picture of the time total computation time.

Computation time

	Wind speed [m/s]								
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
Average time per iteration [seconds]	20.33	22.70	27.17	29.14	45.51	53.88	16.91	8.92	3.81
Median time [seconds]	16.51	23.55	24.66	24.00	33.15	30.85	2.70	1.91	2.40

Table 5.8: Averege time per iteration and median time for the simulations of the second case.

Based on the performance of the MPC, we can see that the results are better in terms of load minimizing. However, there are still some loads in the higher wind speeds that needs to be addressed. A new objective function meant to address these load values is presented in the next section. Computation time has also increased for all the simulations, and are thought to come from the increased complexity in the objective function.

5.2.3 Case number 3

For this case a new objective function that will try to handle the high loads for the wind speeds 18 m/s and 20 m/s. Looking at the values, we can see that both the arm moment 3 and 4 gives high values, and a reduction in these is preferable. A new objective function is proposed as:

$$\begin{aligned} L = q_1(P_{ref} - P)^2 + q_4 x_{29}^T x_{29} + q_5 x_{30}^T x_{30} + \\ q_2 x_{24}^T x_{24} + q_3(F_{tow,1} - F_{tow,2})^2 + \\ q_3(F_{tow,3} - F_{tow,4})^2 + Q_1(U_k - U_{k+1})^2. \end{aligned} \tag{5.4}$$

By adding the states for both of the upper arm rotations (state x_{29} and x_{30}), this will minimize the movement in these arms and thus reduce the moment. In Table 5.9 we can see the results of the DEL's. For the wind speeds up to, and including 16 m/s there is not much change except for 12 m/s where most of the DEL's have been reduced. The tower root moment for 20 m/s have been reduced by almost 30%, and this reduction seems to originate in the reduction of the arm moment 3 and 4. However, the DEL's for 18 m/s did not experience the same reduction, but rather an increase in these arm moments.

Damage Equivalent Loads (DEL)

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total**
Tower root moment	0.27	0.31	0.34	0.23	0.40	0.52	0.45	1.64	1.68	0.69
Yaw moment 1	0.16	0.23	0.22	0.24	0.10	0.15	0.20	0.39	0.77	0.25
Yaw moment 2	0.19	0.25	0.23	0.11	0.22	0.19	0.08	0.53	0.69	0.22
Arm moment 1	0.15	0.19	0.25	0.21	0.19	0.35	0.35	0.57	0.72	0.38
Arm moment 2	0.15	0.23	0.24	0.31	0.22	0.34	0.29	0.68	0.83	0.41
Arm moment 3	0.16	0.24	0.23	0.14	0.27	0.42	0.30	1.38	1.23	0.50
Arm moment 4	0.15	0.19	0.18	0.16	0.33	0.39	0.30	1.25	1.43	0.51

Table 5.9: Comparing DEL's between MPC and Vestas controller for the third case.

Looking at the power generation in Table 5.10 there are marginally changes to the power. We can see that for 20 m/s, the lower turbines experience a loss of 2% and the upper turbines has a loss of 1% in power production. These correlates to the theory about the trade-off between power and load.

Mean power production

	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total**
Power 1	1.16	0.82	1.09	0.90	0.92	0.95	1.10	1.13	1.11	1.02
Power 2	1.18	0.81	1.11	0.89	0.89	0.94	1.07	1.14	1.11	1.01
Power 3	1.69	1.15	1.04	0.90	1.19	1.08	1.12	1.11	1.09	1.09
Power 4	0.86	1.37	1.06	0.88	1.36	1.07	1.11	1.11	1.09	1.10

Table 5.10: Comparing mean power production between MPC and Vestas controller for the third case.

In Table 5.11 the pitch activity is presented. We can see that the overall pitch activity is still low compared to the Vestas controller, and has only marginally changed with the new objective function. There are some changes within certain wind speeds e.g. 10 m/s have gained an increase for both of the lower turbines.

Pitch distance traveled (PDT) [Degrees]										
	Wind speed [m/s]									
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**Total sum (ratio)**
PDT 1	0.74	0.94	2.10	1.68	0.39	0.84	0.46	0.52	0.43	613.9(0.60)
PDT 2	0.31	1.17	3.06	1.13	0.35	0.73	0.47	0.57	0.63	683.2(0.61)
PDT 3	0.15	0.16	1.61	0.30	0.41	0.59	0.17	0.36	0.36	495.7(0.34)
PDT 4	0.06	0.80	2.54	0.40	0.54	0.61	0.12	0.35	0.44	488.4(0.36)

Table 5.11: Pitch activity shown as pitch distance traveled. Comparing the MPC with the Vestas controller for the third case.

Table 5.12 the computation time is shown. Overall the computation time has not changed much compared to the previous case. It is still far from real time optimization for the majority of the wind speeds.

Computation time									
	Wind speed [m/s]								
	4	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
Average time per iteration [seconds]	18.65	27.45	28.31	38.10	48.07	36.11	20.84	3.18	4.95
Median time [seconds]	14.80	27.32	27.23	29.69	39.84	28.61	2.94	1.90	1.71

Table 5.12: Averege time per iteration and median time for the simulations of the third case.

This concludes the result sections, and the next chapter will go more into detail about some of the more interesting results.

Chapter 6

Discussion

This chapter will discuss the result presented in the previous chapter. The three cases will be discussed and compared to give a better understanding on the changes that were made to improve the results.

Selecting the objective function that gave the best results was a time-consuming progress. Many of the candidates for objective functions turned out to have a long computation time and tuning these trying to achieve good results also increased the time spent on it.

The chapter will start with a discussion on power optimization, DEL reduction and pitch activity in Section 6.1. Before a discussion on the computation time in Section 6.2.

6.1 Power optimization, DEL reduction and pitch activity

In this section, more details on the results will be discussed. In the introduction it is stated that the MPC is expected to perform better than the Vestas controller, and this section will try to determine if this assumption holds or not. As the results are only presented in tables up until now, it is hard to determine what happens for the different wind speeds. Some cases will now be presented with plots that will be discussed. Having a range of wind speeds to compare the results will lead to many similar plots. Seeing all these plots will increase the length of the thesis but provide little useful information. Therefore only presenting plots from interesting results is necessary. Of the four evaluation criteria stated in Section 5.1, three of them will be discussed in this chapter, and the computation time will be discussed in the next section.

6.1.1 Power optimization

Starting off with the first case, we can see from Section 5.2.1 that this objective function is doing well overall compared to the Vestas controller. For the majority of the wind speeds the power production is high. There are, however, an exceptions to this. As mentioned in Section 5.2.1, the power production for 10 m/s wind speed is lower than expected and this was seen for all the three cases. An explanation for this can be that at 10 m/s the wind speed is close to the rated power, and if we look at the term $(P_{ref} - P)^2$ in the objective function, an interesting behavior can be found here. If the wind speed exceeds the rated power, the controller can easily reach the rated power and P is close to P_{ref}. A consequence of this is that this power term becomes very small and is no longer an important part of the objective. Once the wind speed drops below the rated wind speed the value of this term will then increase rapidly, and this will give large differences in the value of the objective function making it difficult to find the optimal solution. In Figure 6.1 we can see this happens with the MPC between 60 to 90 seconds for the first turbine.

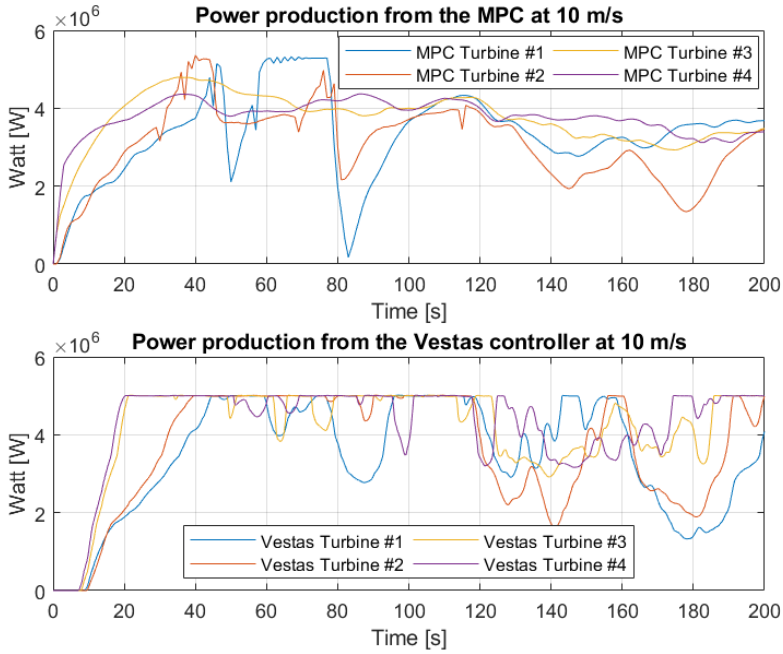


Figure 6.1: Power production for both MPC and Vestas at 10 m/s

This can be seen as a weakness in the objective function, and an alternative is to just optimize power rather than trying to follow a power reference. However, the solver was struggling to find a feasible solution, and the results were not good. No other solution to this problem has been implemented, and this problem will require further study.

It should be noted that for all the other wind speeds, the MPC has a superior power production compared to the Vestas controller regardless on the objective function presented in the result sections. It is worth noticing that not all the individual turbines meets the criteria of keeping 97% power production listed in the evaluation criteria in Section 5.1, but this is accomplished by the collective sum of power.

6.1.2 DEL optimization

Several different methods have been tried to achieve good DEL's results. Looking at the second and third cases in the result sections, we can see that the MPC achieves good results overall with only a few exceptions for the wind speeds of 18 m/s and 20 m/s. Even with no load optimization, the MPC did rather well in most cases. The theory that there is a trade-off between power and loads gives hopes to further reduce the loads since the power is high in most cases.

Calculating the loads directly in the objective function lead to a very long simulation time and not gaining any significant reduction compared to the solution proposed in the cases presented. It is expected that the complexity of this calculation was too high and therefore it turned out to be an unacceptable solution. With direct load calculation in the objective function not viable, alternative ways to reduce the structure moments have been tested. As stated in the result sections 5.2.2 and 5.2.3, reducing the movement in the structure by adding selected tower states into the objective function worked well. Another good alternative to the direct load calculation, is to add the thrust difference between the opposite turbines.

The last proposal worked quite well and did give a reduction in the yaw moments for the tower sections. This can be seen in the results going from the first case to the second case. In Figure 6.2 we can see the state trajectory for the upper section arms. Case number two, represented in blue, does not include these states in the objective function. While case number three, represented in red, have these included in the objective function. We can see that the overall activity of the state trajectory for case 3 is lower, thus reducing the loads on the structure.

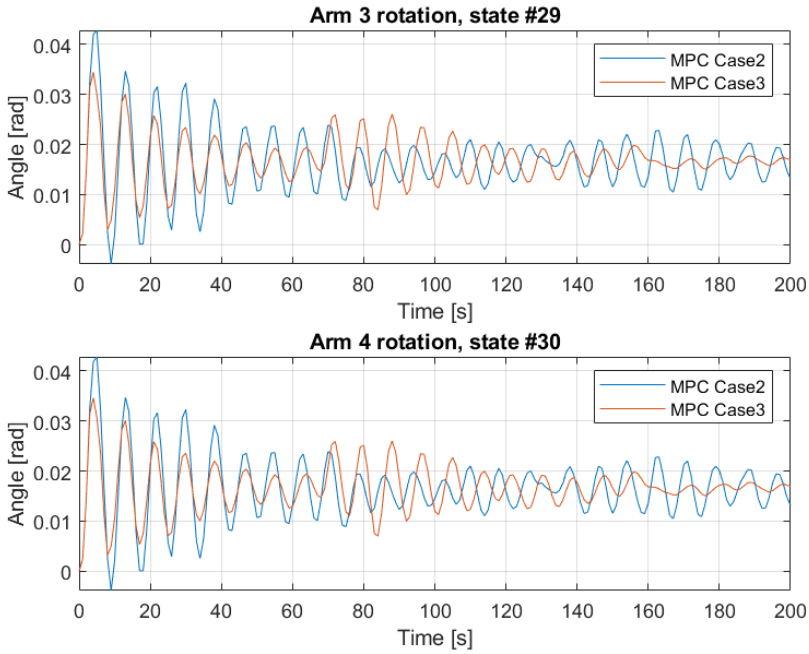


Figure 6.2: Plots of the arm rotation state values for the upper section tower arms. Comparing the objective functions from the second and third case with wind speed 20 m/s.

In Figure 6.3 we can see the moment loads on the structure, comparing the MPC with the objective function from case number two and the Vestas controller at 8 m/s wind speed. From this figure we see that the M_{yaw} activity is reduced significantly, however an increase in the moment M_x for the lower tower is introduced.

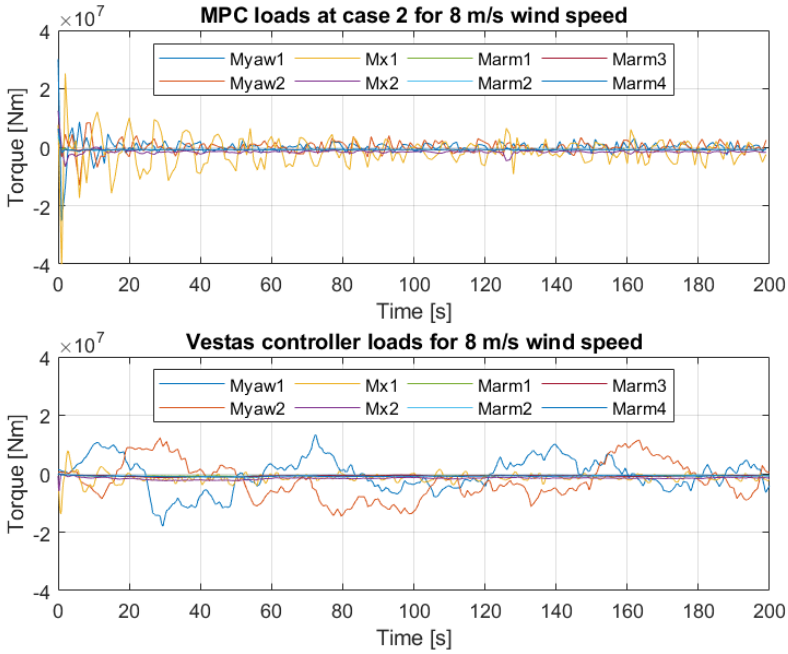


Figure 6.3: Comparing loads between Vestas controller and MPC for case number two at 8 m/s wind speed.

The DEL values for the upper arm moments are still high but the implementation of the new objective function shows that there is potential to improve this. Since the power generation is very high for these wind speeds, the potential to trade power into load reduction is present. The author is convinced that with proper tuning these values could be reduced. There is also a potential to implement gain scheduling. Gain scheduling is a method to control nonlinear system for varying operation conditions using a family of linear controllers [17]. By utilizing gain scheduling, the tuning gains can be automatically adjusted for the different wind speeds. This has previously implemented for wind turbine control in [3]. Overall the DEL's are better compared to the performance of the Vestas controller.

6.1.3 Pitch activity

A similar statement as for DEL reduction and power production can be drawn for the pitch activity, overall the pitch activity is superior to the Vestas controller. There are, however, some exceptions here as well. For both cases, number two and three, the pitch activity is significantly high for wind speeds 8 m/s and 10 m/s. For the above rated wind speeds the pitch activity exceeds the Vestas controller by a large margin.

In Figure 6.4 we can see an example where the pitch activity is far better for the MPC. This comparison is from the MPC with the second case objective function and the Vestas controller. In this example the pitch distance traveled is 67.5% less than the Vestas controller.

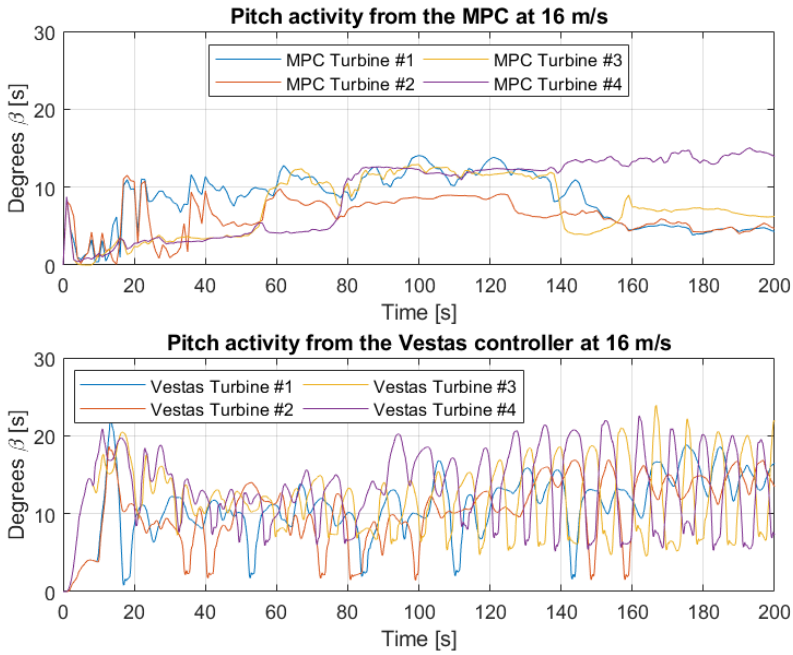


Figure 6.4: Pitch activity compared for Vestas and MPC with the second case objective function at 16 m/s wind speed.

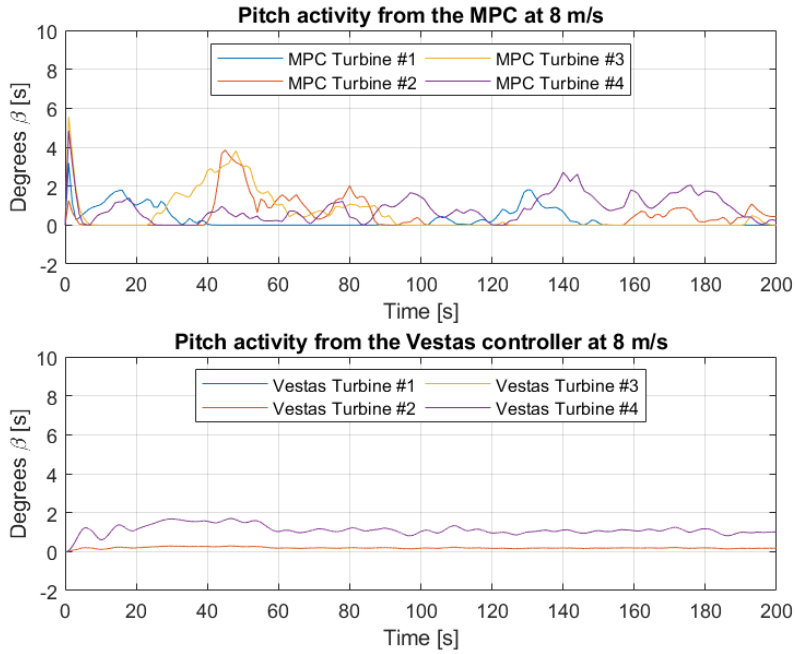


Figure 6.5: Comparison of pitch activity between MPC and Vestas controller for 8 m/s wind speed

As mentioned in the results, Section 5.2.2, the pitch activity for wind speed 8 m/s is almost 2.4 times higher than the Vestas controller. In Figure 6.5 we can see the difference in pitch activity and it shows that the MPC does have much higher activity. However, compared to the higher wind speeds, these numbers are still low. We can also look at the overall performance of the MPC for this particular wind speed and note that the DEL reduction is 75.7% better and the power production is 8% higher than the Vestas controller.

6.2 Computation time

Computation time is an important topic for control systems. All controllers need to be able to calculate fast enough to give the correct input at the correct time. MPC is a real time optimization controller that also predicts the state of the system into the future. This increases the amount of computations required to solve the optimization. To achieve real time optimization, the threshold is that each iteration must be computed in time to give the correct input. In other words, each second plus the MPC horizon must be computed faster than one second. Large models often come with high computation time, and this was a problem in the project thesis [22] that prior to the start of this master project was conducted. These results showed that the MPC was nowhere near the threshold of achieving real time capabilities.

There are many contributions to the computation time. Some have larger affect than others and some of them will be discussed here. A trade-off between accuracy and computation time is always present and selecting time step and number of collocation points will determine this. Investigation into the tower model uncovered that this model was stiff. Stiff models typically require small step size to capture the fast dynamics. As the MPC runs with a 1 Hz time step, compared to the 80 Hz for the Vestas controller, this might seem to be too large. However, changing the step size to the 1 Hz for the Vestas controller gave the same results. From these findings we can conclude that the fast dynamics of the tower model does not have a great effect on the fatigue loads.

In the tables that lists the computation time we can see information on the average time per iteration and the median time. Although the average time can be high, this does not necessarily mean that all the iterations are slow. That is why the median time is also presented. If there is a large difference in the average time and median time, it will indicate that there are some iterations that are slowing down the overall computation time. Meaning that at certain values for the states, the solver is having a hard time finding the optimal solutions. There can be several reasons for this, and one explanation can be that the region where the solver is searching is "flat". If the optimal

solution lies in a "flat" area, the differences in the objective function values are low, and this can be recognized by looking at the value for each iteration during the simulation. If the value of the objective function stays stable for some iteration and then suddenly makes a "jump" to a higher value before quickly going back to the stable value, we can recognize this as the solver is trying to find a better solution by changing the direction in the feasible region. Eventually the iteration will run out of the maximum allowed iteration given, for this case 3000, and move on to the next step. This would not be a problem for linear MPC as it would have been a convex problem, but as the MPC is not linear it might be an explanation of the slow computation time.

Looking at the computation time for the first case in Section 5.2.1 Table 5.4, simulations show that there three out the nine wind speeds have a slow computation time. These wind speeds are 10 m/s to 14 m/s. This wind speed region is close to the rated 11.4 m/s for the turbines. Due to the wind turbulence, the wind values are going from below rated to above rated region constantly. This will affect the value in the objective function as discussed in 6.1.1, making it harder for the solver to find the optimal solution and influence the computation time. The other six wind speeds show a much faster computation time and are close to the threshold of reaching real time optimization.

As the objective function for the first case is less complex compared to the other cases, it was expected that the computation time would be faster. Also worth mentioning is that this objective function does not actually have any terms that involve the tower model, and in reality the tower model could be removed from the whole optimization problem, only simulating the turbine models. This would remove half of the states making the problem significantly smaller. However, this was not implemented, and the computation time on this is only speculation as it is not tested.

In Section 5.2.2 and 5.2.3 the computation had a large increase and far from becoming real time optimizing. Several different methods trying to reduce the computation time has been tested. Worth mentioning is model reduction, where the tower model was reduced in the hopes of removing the fast dynamics. Linearization of the turbine model

was also considered, however, the accuracy was not as good as the author had hoped. Overall, the performance of the MPC in regards to DEL reduction, power generation, and pitch activity was prioritized over computation time. Further work with reducing the run time is needed to achieve real time optimization.

Selecting the horizon can influence both the computation time and the controller performance. Having a long horizon time will increase the computation time, but at the same time give the MPC more predicted information that can increase the performance. For this master thesis 10 seconds was selected as horizon length, however it would be interesting to test out a reduced horizon length to see if there is any influence.

Chapter 7

Conclusions and future work

This thesis aimed to implement a MPC on a MRT and evaluate the performance compared to the results with the Vestas controller from [21]. The objective is to reduce the fatigue loads on the support structure with a limitation on the power loss, while at the same time does not overuse the pitch activity of the turbine blades. MPC is implemented using CasADi in Matlab and three different objective functions have been presented with results. It was expected that the MPC should outperform the Vestas controller and based on the results it suggests that the MPC is overall superior to the Vestas controller on DEL reduction, power generation and pitch activity. However, the MPC does not meet the criteria for real time optimizing.

In [21] it is stated that the Vestas controller performance is almost identical to the BLC for the lower wind speeds. This cannot be said about the MPC as the performance is exceptional at lower wind speeds.

For wind speeds of 18 m/s and 20 m/s, the MPC does not manage to compete with the Vestas controller on DEL reduction. In addition, the MPC is also struggling to compete with the Vestas controller on power generation in the region close to rated wind speed and a weakness in the suggested objective function have been identified. However, in theory, it is expected to be a trade-off between power production and DEL reduction. In all these cases, either the power or DEL performance has been superior, and the author is convinced that with tuning or changing the objective function these results will improve.

Further work is needed to improve the computation time of the MPC and some methods have been considered but not implemented. These are model reduction and linearization of turbine. It is also worth mentioning that the computation time is heavily dependent on the objective function and finding a good objective function might reduce the run time significantly.

Additional time is also needed to improve on the DEL reduction results for higher wind speeds. A method that can be utilized is gain scheduling to have individual tuning over each of the wind speed conditions.

It is also worth investigating further into how much influence a reduced prediction horizon have on the performance, for computation time, DEL reduction, pitch activity, and power optimization. Changing the length of the horizon might cause large changes to the results.

Appendix A

Coefficients for C_p and C_t polynomial

```
Cpp00 = -1.718;
Cpp10 = 0.8838;
Cpp01 = 0.2794;
Cpp20 = -0.1347;
Cpp11 = -0.1138;
Cpp02 = -0.0007477;
Cpp30 = 0.009895;
Cpp21 = 0.01575;
Cpp12 = 0.00055;
Cpp03 = -0.001353;
Cpp40 = -0.0003606;
Cpp31 = -0.0008605;
Cpp22 = -0.0002366;
Cpp13 = 0.0003915;
Cpp04 = 3.502e-05;
Cpp50 = 5.059e-06;
Cpp41 = 1.691e-05;
Cpp32 = 6.773e-06;
Cpp23 = -1.919e-05;
Cpp14 = -7.491e-06;
```

```
Ctp00 = -1.733;
Ctp10 = 0.9051;
Ctp01 = 0.312;
Ctp20 = -0.132;
Ctp11 = -0.1199;
Ctp02 = -0.01086;
Ctp30 = 0.009922;
Ctp21 = 0.01507;
Ctp12 = 0.003372;
Ctp03 = -0.0005681;
Ctp40 = -0.0003669;
Ctp31 = -0.0008581;
Ctp22 = -0.0004387;
Ctp13 = 0.0001889;
Ctp04 = 2e-05;
Ctp50 = 5.311e-06;
Ctp41 = 1.678e-05;
Ctp32 = 1.554e-05;
Ctp23 = -8.839e-06;
Ctp14 = -4.621e-06;
```

Figure A.1: Coefficients for C_p and C_t polynomial

References

[1] Aeolus. Simplified nrel5mw turbine for simulink. `http://www.ict-aeolus.eu/SimWindFarm/model-turbine.html`, 2019.

[2] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, In Press, 2018.

[3] J. Bao, H. Yue, W. E. Leithead, and J. Wang. Lidar-assisted wind turbine gain scheduling control for load reduction. In *2016 22nd International Conference on Automation and Computing (ICAC)*, pages 15–20, 2016.

[4] F. Bianchi, H. De Battista, and R. J. Mantz. *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design.* 01 2007.

[5] L. T. Biegler. *Nonlinear Programming*, chapter 10. Simultaneous Methods for Dynamic Optimization, pages 287–324.

[6] CasADi. Casadi example pack. https://web.casadi.org/, 2019.

[7] D.Q.Mayne, J. Rawlings, and C. R. andP.O.M. Scokaert. *Constrained model predictive control: Stability and optimality.* Elsevier, 2000.

[8] EU. Communication from the commission to the european parliament, the european council, the council, the european economic and social committee and

the committee of the regions, the european green deal. https://ec.europa.eu/info/sites/info/files/european-green-deal-communication_en.pdf, 2019.

[9] B. Foss and T. A. N. Heirung. *Merging Optimization and Control.* 03 2016.

[10] J. D. Grunnet, M. Soltani, T. Knudsen, M. Kragelund, and T. Bak. Aeolus toolbox for dynamic wind farm model, simulation and control. In *Proc. of the 2010 European Wind Energy Conference*, 2010.

[11] T. G. Hovgaard, S. Boyd, and J. B. Jørgensen. Model predictive control for wind power gradients. *Wind Energy*, 18(6):991–1006, 2015.

[12] A. Jain, G. Schildbach, L. Fagiano, and M. Morari. On the design and tuning of linear model predictive control for wind turbines. *Renewable Energy*, 80:664 – 673, 2015.

[13] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory, 2009.

[14] T. Mikkelsen, N. Angelou, K. Hansen, M. Sjöholm, M. Harris, C. Slinger, P. Hadley, R. Scullion, G. Ellis, and G. Vives. A spinner-integrated wind lidar for enhanced wind turbine control. *Wind Energy*, 16(4):625–643, 2013.

[15] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, New York, NY, USA, second edition, 2006.

[16] L. Y. Pao and K. E. Johnson. A tutorial on the dynamics and control of wind turbines and wind farms. In *Proceedings of the 2009 Conference on American Control Conference*, ACC'09, pages 2076–2089, Piscataway, NJ, USA, 2009. IEEE Press.

[17] W. J. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401 – 1425, 2000.

[18] E. R. Rønnestad. State estimation of multi-rotor wind turbines. Master's thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics, and Electrical Engineering, Department of Engineering Cybernetics, Trondheim, 6 2020.

[19] A. Scholbrock, P. Fleming, D. Schlipf, A. Wright, K. Johnson, and N. Wang. Lidar-enhanced wind turbine control: Past, present, and future. In *2016 American Control Conference (ACC)*, pages 1399–1406, 2016.

[20] SimWindFarm. Creating a wind field. http://www.ict-aeolus.eu/SimWindFarm/windfield.html, 2017.

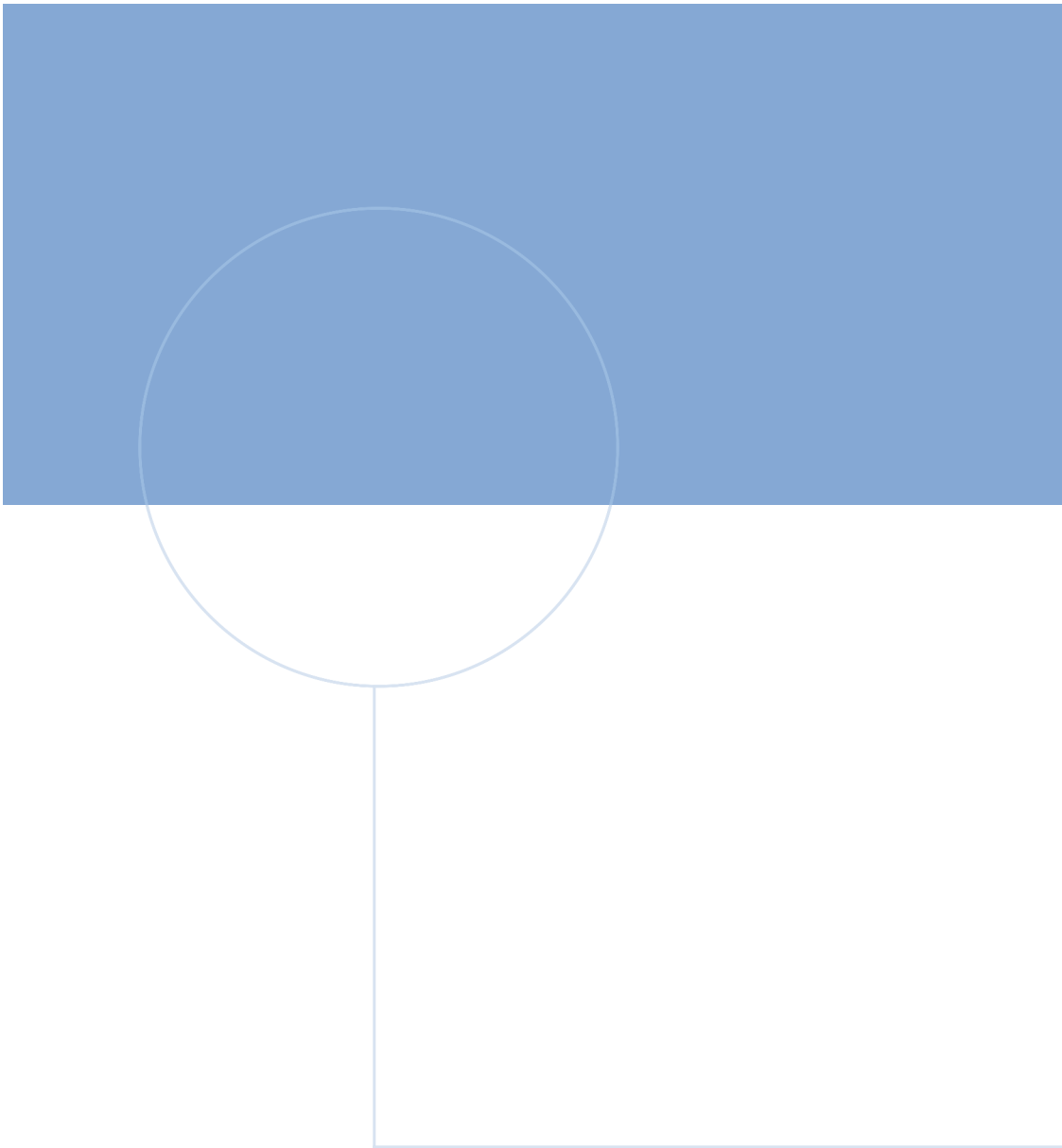
[21] K. H. Sørensen, T. Knudsen, O. T. Filsoof, T. G. Hovgaard, J. D. Grunnet, J. X. V. Neto, and R. Wisniewski. Multi-rotor wind turbine control challenge - a benchmark for advanced control development. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1615–1622, Aug 2018.

[22] J. Urdal. Model predictive control of mrt. 2019.

[23] D. Weston. Measurable power gains found in multi-rotor vestas concept. `https://www.windpowermonthly.com/article/1521072/measurable-power-gains-found-multi-rotor-vestas-concept`, 2018.

[24] D. Willis, C. Niezrecki, D. Kuchma, E. Hines, S. Arwade, R. Barthelmie, M. DiPaola, P. Drane, C. Hansen, M. Inalpolat, J. Mack, A. Myers, and M. Rotea. Wind energy research: State-of-the-art and future research directions. pages 133–154, 2018.

[25] Windeurope. Wind energy ready to help deliver europe's green deal investment plan. `https://windeurope.org/newsroom/press-releases/wind-energy-ready-to-help-deliver-europes-green-deal-investment-plan/?ref=mainbanner`, 2020.



NTNU

Norwegian University of
Science and Technology