

Model Predictive Control of a Wind Turbine

Aleksander Gosk

Kongens Lyngby 2011
IMM-M.Sc-2011-63

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-M.Sc: ISSN 0909-3192

Summary

In the era of growing interest in limiting CO_2 emission and our dependence on fossil fuels renewable energy sources receive the biggest attention ever. It is predicted [1] that by the year 2035 the use of this kind of energy will triple and wind energy will be the main source of this increase. This work focuses on one of the most common wind energy conversion systems: horizontal axis wind turbine

It's efficiency and longevity relies heavily on the quality of the control approach used. Controller designers are aiming for maximizing the produced electric power for some range of wind speeds and keeping it constant for others. At the same time they have to ensure that the control isn't too aggressive and that it is honouring other constraints that wind turbine is subject to. Those objectives often prove to be opposite in nature and a golden mean - optimal solution - is need.

This work presents the control technique that, by it's nature, enables optimal solution of a control problem while honouring constraints that have been imposed upon us by wind turbine's designer - Model Predictive Control.

Since control objectives are different for different wind speeds the way in which the controller operate has to change too. Gain and weight scheduling techniques, that will enable smooth shifting between those, so called, operation regions, will be introduced. This approach has the benefit of possibly lowering the stresses that the wind turbine systems are subject too, in comparison with e.g. simple switching between controllers what can be one of the causes of reduced longevity.

Preface

This thesis was prepared at Informatics Mathematical Modelling department, at the Technical University of Denmark in partial fulfilment of the requirements for acquiring the MSc. degree in engineering. It have been supervised by Niels Kjølstad Poulsen, Hans Henrik Niemann, Mahmood Mirzaei from IMM, DTU and Peter Fogh Odgaard from KK-Electronic.

Thesis deals with the issues of modelling and control of a wind turbine. The main focus is put on the use of the Model Predictive Control technique. FAST simulator has been used for simulation purposes and partially for obtaining the linearized model of the plant.

Lyngby, August 2011

Aleksander Gosk

Acknowledgements

I'd like to express my gratitude to Niles Kjølstad Poulsen and Hans Henrik Niemann for their guidance and support. Thank you for keeping me on the schedule, sharing your ideas and experience and helping me in setting reasonable goals for my work. I'm also grateful to Peter Fogh Odgaard from KK-Electronic for his feedback and turning my attention to some matters that I might have overlooked otherwise.

Special thanks goes to Mahmood Mirzaei and Lars Christian Henriksen for their invaluable input, support, motivation and all that time that we've spent discussing technical matters linked to my project. Thank you so much.

Contents

Summary	i
Preface	iii
Acknowledgements	v
1 Introduction	1
1.1 Wind Turbine	2
1.2 Control	3
1.3 Tools used	4
1.4 Modern control methods	4
1.5 Scope of the thesis	5
1.6 Thesis structure	5
I Modelling	7
2 Introduction to wind turbine modelling	9
2.1 Wind power extraction	10
2.2 Modes of operation of a wind turbine	15
2.3 Chapter summary	17
3 Wind turbine subsystems and the wind model	19
3.1 Generator	19
3.2 Flexible drivetrain shaft	20
3.3 Generator torque actuator	21
3.4 Collective blade pitch actuator	22
3.5 Wind	22
3.6 FAST	23

3.7	Chapter summary	25
4	Affine model with constraints	27
4.1	Complete model	27
4.2	Affine model	28
4.3	Constraints	28
4.4	Chapter summary	29
II	Control methods	31
5	Offset-free control and wind estimation	35
5.1	Disturbance modelling	36
5.2	Disturbance estimation with the use of Kalman filter	36
5.3	Chapter summary	37
6	Model Predictive Control	39
6.1	Output prediction	41
6.2	Cost function and quadratic programming	44
6.3	Constraints	47
6.4	Chapter summary	52
7	Gain and weights scheduling	53
7.1	Gain scheduling	53
7.2	Weight scheduling	54
7.3	Chapter summary	57
III	Implementation and simulation	59
8	Wind Turbine type	61
8.1	Wind turbine	61
8.2	Wind turbine actuators	61
8.3	Chapter summary	64
9	Model analysis	65
9.1	System matrices	66
9.2	Eigenvalues	68
9.3	Observability and controllability	70
9.4	Simulations	72
9.5	Chapter summary	73

10 MPC	81
10.1 Offset free control	81
10.2 Tuning	82
10.3 Constraints implementation	85
10.4 Stability	85
10.5 Simulations	86
10.6 Chapter summary	114
IV Conclusions	117
11 Conclusions	119
11.1 Modelling	119
11.2 Controller	120
11.3 Implementation	120
A FAST Linearization setup	123

CHAPTER 1

Introduction

According to the World Energy Outlook 2010 [4] (by International Energy Agency, IEA) the energy consumption will most likely increase by approximately 36 % between the year 2008 and 2035. At the same time the concern about the impact that the industry has on the climate change and our dependency on limited deposits of fossil fuels, is getting larger than ever before. During the United Nations conference on climate change, that has been held in the December of 2009 in Copenhagen, a non-binding objective of limiting the increase in the average global temperature to $2^{\circ}C$ above the levels in the pre-industrial era has been worked out. Reaching this goal is heavily dependent on limiting the CO₂ emission which is, among others, the by-product of utilization of fossil fuels. This points out the necessity of shifting the focus of global electric energy production policy from fossil fuels to renewable energy sources. In fact, the same report predicts that the use of this kind of "clean" electricity will triple by 2035 when it's share of the global electrical energy production will reach one third in comparison to 19 % in 2008. Wind energy exploitation, next to hydro power, will be the main source of this increase.

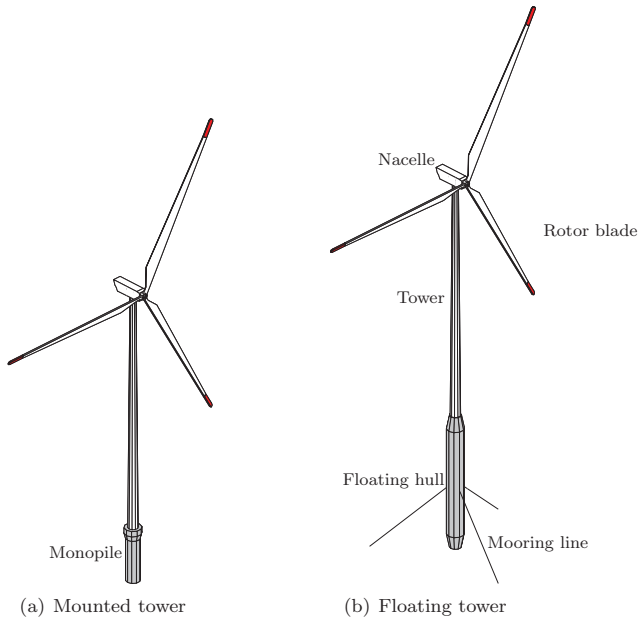


Figure 1.1: Horizontal axis wind turbines [6]

1.1 Wind Turbine

Horizontal Axis Wind Turbines (HAWT) are the most common Wind Energy Conversion Systems (WECS). They can be designed to work either on land or in the water. Two different offshore HAWT are shown on figure 1.1. The main difference between them is the way in which they are deployed. The traditional approach is to use a tower that would be mounted to the bottom with the use of a monopile. This limits the area at which it can be built to shallow water only. Another, relatively new and still rare, approach is to deploy a wind turbine in a much deeper water in a floating hull and mount it to the bottom with mooring lines. The later is much more complex in terms of technology but can yield greater reward in terms of produced power since the areas further from the land are richer in unobstructed steady wind.

Onshore HAWT are easier to control since their dynamics aren't as complicated as in a floating type. This thesis focuses on a 3 blade version of this kind of wind turbine.

Figure 1.2 shows the main components of the HAWT in more details. While wind is passing the rotor blades it creates a lift force that causes rotor to move.

Rotational power is passed to the gearbox by the low speed shaft (LSS). The momentum is transported further to the generator with the help of high speed shaft (HSS). Here mechanical power is transformed into an electrical one. Figure doesn't show the yaw drive which is used to turn the nacelle in the direction of wind. It will be disregarded in this thesis. Other means by which wind turbine can be controlled include varying generator torque and changing of the blade pitch.

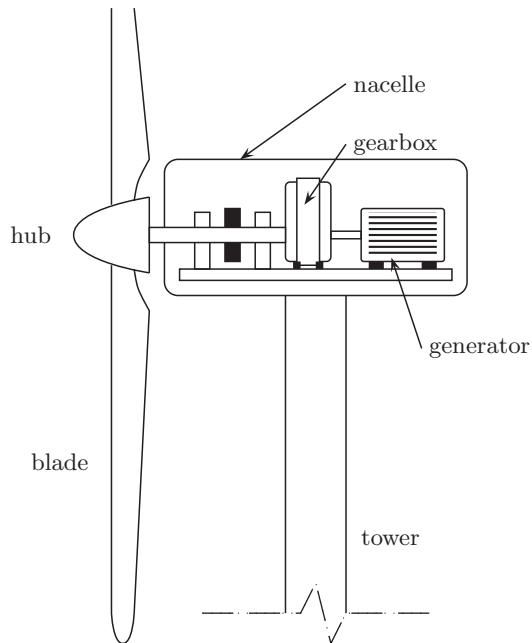


Figure 1.2: Nacelle of a horizontal axis wind turbine.[2]

The biggest factor that has an impact on the amount of energy that can be produced (provided that HAWT has been built in an optimal environment) by a wind turbine is its mechanical structure. Wind speed increases with height (so called wind shear) so higher tower means working with wind that carries, on average, more power. Longer blades translate to bigger lift force which in the end gives the same effect - more electrical power on the output. More examples of this kind could be provided.

1.2 Control

Although the mechanical design of the HAWT has the most significant impact on the amount of power produced, control is also a very important issue in terms of the efficiency and longevity of WECS. In order to maximize the power production when the wind speed is below its rated value both blade pitch and rotor's rotational speed should be kept optimal. This task can be quite complex since wind turbine exhibits some non-linear behaviour in this region. Additionally while switching between operation with above rated wind speed to operation with wind speed value below rated it is fairly easy to expose wind turbine to big mechanical stresses and vibrations which, if cannot be avoided, should at least be kept below certain limits provided by the constructor. Maximizing the power output, while at the same time enabling steady work of the actuators and the whole wind turbine can prove to be a hard task as those goals are often opposing and the golden mean - the optimal solution - has to be found.

Those and other considerations were the main reason for choosing Model Predictive Control (MPC) approach for controller design as the main idea behind this technology is to combine many performance indexes, such as error between plants outputs and their set points or control signal aggressiveness into cost function which is then minimized. Furthermore it allows to specify constraints on those indexes thus allowing e.g. to keep the the pitch of the blades below maximal and above minimal value.

1.3 Tools used

Wind Turbine will be simulated with FAST (Fatigue, Aerodynamics, Structures and Turbulence) simulator. It will be used both for testing the designed controller and partially for deriving the linearised model of HAWT. FAST has an interface that lets it connect to Simulink and this feature will be exploited here.

1.4 Modern control methods

Modern control strategies used today include some kind of optimal control. [18] for example discusses an LQG individual blade pitch control, [21] is concerned about a Non-linear Model Predictive Controller (NMPC) and [8] introduces the idea of coupling an MPC controller with a LIDAR (Light Detection And Ranging) sensor that enables measurement of future wind speed values.

In order to obtain good control, with a linear model of a wind turbine, in

the whole spectrum of operational wind speeds some kind of gain scheduling algorithm is needed. Author of [6] proposes designing one controller for each operation mode and then switching between them while the wind turbine shifts from one mode to another. In [19], on the other hand, so called Linear Parameter Varying Control (LPV) is being used to achieve this transition.

1.5 Scope of the thesis

The aim is to design an MPC controller that will be able to control the given wind turbine in all of its operation modes and transition between them in a smooth way while honouring its constraints. Additionally an effort will be put to maximize the power production while keeping the control action as mild as possible.

The chosen wind turbine (see its definition in chapter 8) employs variable-speed and changing-collective-blade-pitch control strategies and will be modelled as a linear system. An uncommon linearization method will be used. It will consist of deriving a first principle model and supplementing it with parameters acquired through numerical linearization with the use of FAST

Smooth transition between operation modes will be achieved with the use of simple gain scheduling combined with weights scheduling technique that will enable reshaping (adjusting weights) of the MPC's cost function depending on the current wind speed.

The MPC formulation will assume direct connection between plants input and output (non-zero "D" matrix) and will employ soft constraints approach in order to remedy a possible infeasibility of the control problem that might occur in case of using hard ones.

1.6 Thesis structure

This thesis is essentially divided into three parts:

Part I: Modelling In this part the fundamentals of the wind energy extraction with a HAWT will be addressed. Next, its main subsystems and the means by which they are linearized will be discussed. This part will end with a short discussion of the constraints that will be implemented in the controller and with the affine model concept which will be used as the ground for MPC design in the later part.

Part II: Control methods Here, the techniques used for control purposes will be discussed. The offset-free regulation will be discussed first then the concept of Model-Based Predictive Control will be introduced together with techniques used for ensuring proper control in different operation modes - the gain and weight scheduling techniques.

Part III: Implementation and simulation The way in which the model and the controller presented in previous two parts is implemented will be discussed here. The control plant will be defined first. Afterwards models derived in the second part will be analyzed. In the end software realization of the designed controller will be addressed. Results of various simulations that will have been carried out with the designed control system will be presented here as well.

Part IV: Conclusions In this part the conclusions regarding the previous sections will be drawn.

Part I

Modelling

CHAPTER 2

Introduction to wind turbine modelling

There are many different wind turbine subsystems that can be taken into consideration when deriving their dynamic model. The most important of them include:

- generator
- flexible drivetrain shaft
- flexible tower
- nacelle yaw
- bending of the blades

This work will focus only on the first two: the generator and the flexible drive train. Model with that degree of detail depth is denoted as WT1. Additionally it will be augmented with the blade pitch and generator torque actuators together with a wind model. It will have two inputs: generator torque Q_g and blade pitch θ , and two outputs: electric power P_e and rotational speed of the rotor Ω_r . Wind speed change with respect to the linearization point will be

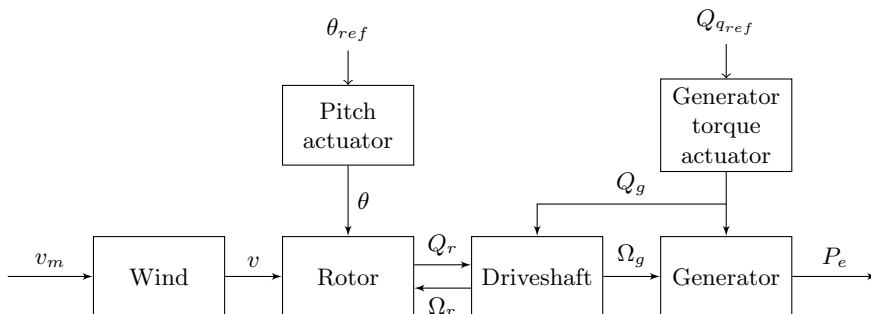


Figure 2.1: Dynamic model of the wind turbine.

treated as a disturbance. Figure 2.1 shows the block diagram of the model that will be derived in this chapter.

The model of this non-linear system will have a linear character and will be derived from physical equations describing the system. At the same time FAST linearization tool will be used in order to build another model which will be used to acquire some of the key parameters needed. The definition of the model that will be obtained in this part of the thesis is summarized in table 2.1. The reader shouldn't be concerned if he is unfamiliar with some of the terms used there. Following chapters will elaborate further on subjects concerning them.

The following two sections address the matters linked with the fundamentals of the wind energy extraction. The concept of the C_p curve which is an important issue when wind turbine's efficiency or control is being discussed. Next the variation of the modelling approach with respect to different wind speed intervals - modes of operation - will be shortly described.

2.1 Wind power extraction

As described in [4] power available from the wind flowing through the area of the *rotor disc* (area covered by the rotating blades), in the absence of the actual wind turbine P_w is given by

$$P_w = \frac{1}{2} \rho \pi R^2 v^3$$

HAWT model definition	
outputs	electric power P_e , rotor's rotational speed Ω_r
inputs	generator torque Q_g , collective blade pitch θ
disturbances	wind speed v
modelling depth	WT1
other subsystems included	generator torque actuator, blade pitch actuator, wind
model type	linear, affine
linearization approach	main structure derived from physical equations; key parameters obtained from FAST linearization tool

Table 2.1: HAWT model definition.

where ρ is the air density, v is the speed of the wind and R is the radius of the rotor disc (length of the blades). At the same time power extracted by the rotor is equal

$$P_r = 2\rho\pi R^2 v^3 a(1-a)^2, \quad a = \frac{\Delta v}{v}$$

where a is called the *axial flow induction factor* and it represents the drop in wind speed just at the rotor of the wind turbine Δv relative to the wind speed far away upstream from the rotor v .

The ratio between the power extracted at the rotor P_r and the power available in the wind P_w is called the *power coefficient* C_p .

$$C_p = \frac{P_r}{P_w} = \frac{2\rho\pi R^2 v^3 a(1-a)^2}{\frac{1}{2}\rho\pi R^2 v^3} = 4a(1-a)^2$$

Rotor is extracting the most power from the wind when the C_p curve is at its maximum. Namely, when

$$\frac{dC_p}{da} = 12a^2 - 16a + 4 = 0$$

which gives $a = \frac{1}{3}$. Hence, the maximum value of the C_p coefficient is

$$C_{p_{max}} = \frac{16}{27} \approx 0.593$$

This value is known as the *Betz limit* and it represents the theoretical limit for the wind turbine's efficiency with respect to wind-to-mechanical power conversion.

The C_p coefficient is a function of the pitch angle of the blades θ and so called *tip speed ratio* λ - factor representing the ratio between the velocity of the tip of the blade and the actual velocity of the wind.

$$\lambda = \frac{\Omega_r R}{v} \quad (2.1)$$

Plots representing relations between those values are called C_p curves. Figure 2.2 and figure 2.3 show a typical C_p curve. It's maximum is, as expected, far lower than the *Betz limit*.

The ratio between rotor power P_r and the speed at which rotor is rotating Ω_r is equal to the aerodynamic torque Q_r .

$$Q_r = \frac{P_r}{\Omega_r} = \frac{\frac{1}{2}\rho\pi R^2 v^3 C_p(\lambda, \theta)}{\Omega_r} \quad (2.2)$$

As mentioned in chapter 1 the power is transferred to the generator with the use of *low speed shaft* (LSS), *gearbox* (where the rotational speed is raised and torque lowered) and *high speed shaft* (HSS). If there were no losses in the system one could assume that the power extracted by the rotor P_r and the electric power generated by the generator P_e are equal. This is not the case in reality. Efficiency factor η is introduced in order to compensate for those losses (occurring among others in the bearings, generator etc).

$$P_e = \eta P_r \quad (2.3)$$

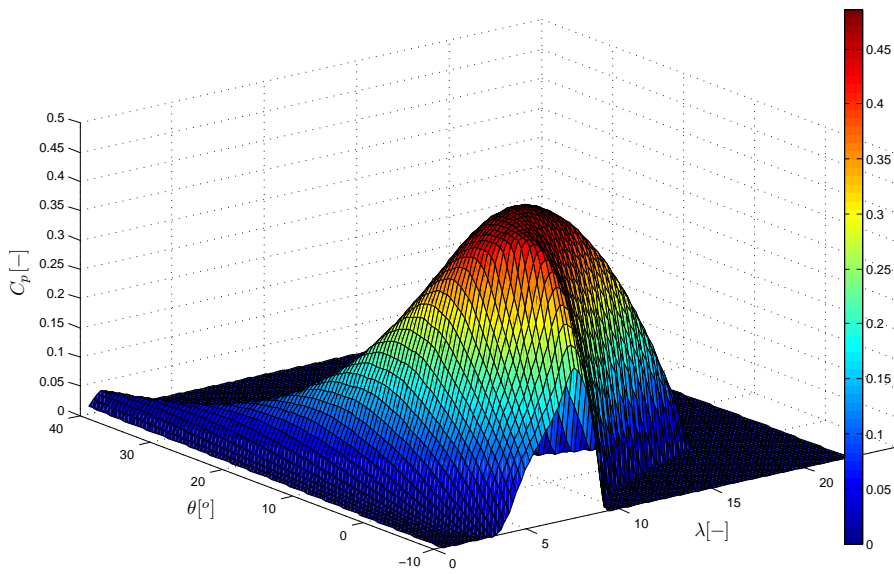


Figure 2.2: Typical C_p curve with the maximum value of 0.486.

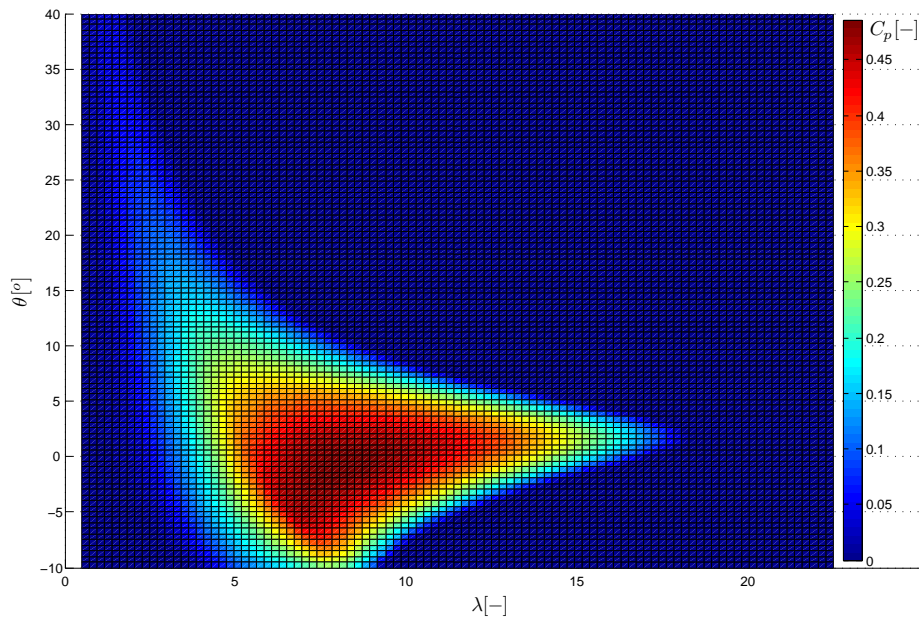


Figure 2.3: Typical C_p curve with the maximum value of 0.486 - top view.

In our case a no-loss transmission will be assumed, which is equivalent to $\eta = 1$ and $P_e = P_r$.

2.2 Modes of operation of a wind turbine

HAWT can work in four different operation modes. The knowledge of where they lie in relation to the wind speed v is important, in our case, both from the controller design (see chapter 7 for more details) and turbine's model derivation point of view (see section 3.6).

Current operating mode depends on the speed of the wind v and the constraints concerning generated electric power P_e and rotor's rotational velocity Ω_r . Four regions of operation are distinguished. They are depicted on figure 2.4. Short characterization follows with v_1 through v_4 being the border wind speeds.

- **top region** (IV, above v_4) - both rotational speed of the rotor Ω_r and the generated power P_e are at their upper limits (nominal power $P_{e,nom}$ and rated rotational speed $\Omega_{r,max}$). The torque Q_g in this region is kept constant while the blade's pitch θ is changing together with the speed of the wind v what results in a proportional change of the power coefficient C_p thus compensating for the variation in wind power. This in turn keeps the generated power P_e at it's nominal level what in consequence allows the rotational speed of the rotor Ω_r to stay constant as well (see (2.2)).
- **high region** (III, between v_3 and v_4) - rotational speed of the rotor Ω_r is at it's upper limit, while the generated power P_e is below it's nominal value. Since it is quite narrow it is treated as a transition between much wider top and mid region. In our case it will be assumed that it extends the mid region if it comes to control strategy.
- **mid region** (II, between v_2 and v_3) - both rotational speed of the rotor Ω_r and the generated power P_e are below their upper and above lower, in case of Ω_r , limits. The pitch of the rotor blades is kept in an optimal position θ_{opt} that enables maximization of the power coefficient C_p , provided that the controller will keep the tip speed ratio (see (2.1)) at it's optimal value by varying the generator torque Q_g while the wind speed is changing.
- **low region** (I, between v_1 and v_2)- the rotational speed of the rotor Ω is at it's lowest allowed level. The control strategy taken in this work assumes keeping the pitch at the level that is optimal in the mid region and continuing to control the system with the generator torque Q_g .

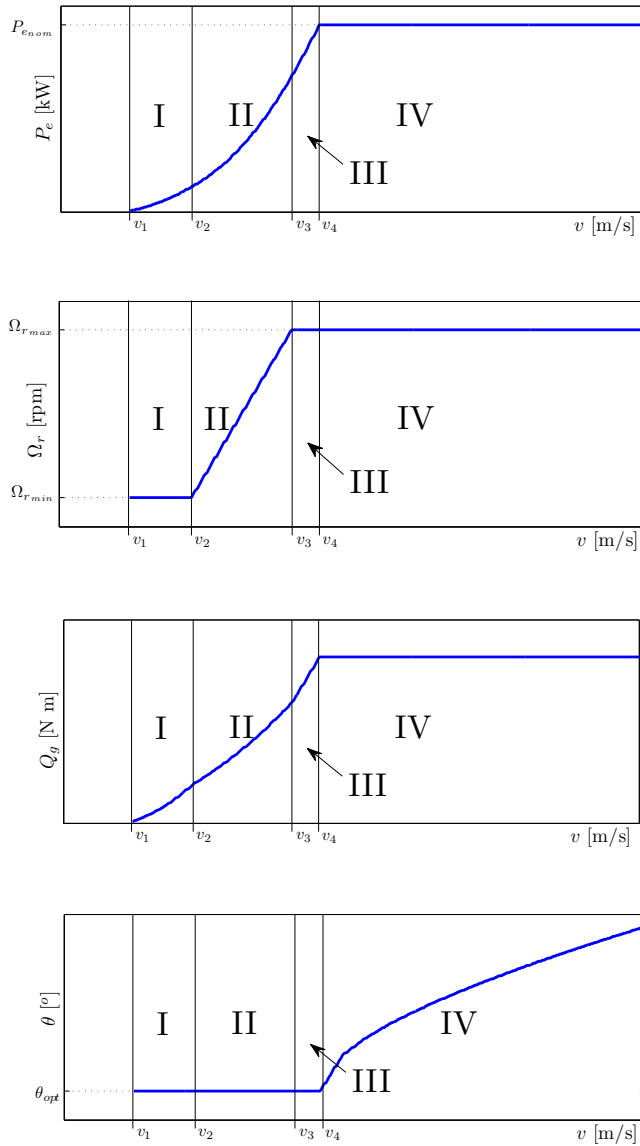


Figure 2.4: Wind turbine operating modes with respect to the wind speed.

Below the low region (below cut-in wind speed v_1) the wind turbine is shut down because of economical reasons - energy production is not affordable at that point. It is also shut down above the top region, where the forces working against it would be inducing too much stress on its construction.

2.3 Chapter summary

In this chapter the definition of the model that will be obtained, together with the basic physics behind the wind extraction and the concept of dividing the operation of the wind turbine into wind speed dependent regions has been shortly presented.

Both matters will be further addressed in the following chapter where the derivation of the wind turbine's model will be discussed.

CHAPTER 3

Wind turbine subsystems and the wind model

In this chapter the derivation of the linear model of the wind turbine will be presented. Its basic subsystems and their transformation into a regular state space description will be discussed first. Afterwards FAST linearization tool will be utilized in order to supplement the model derived with missing parameters. The knowledge obtained in the previous chapter will be used here since the linearization performed with the use of FAST is different for each region of operation and the way of setting is up is dependent on the shape of the C_p curve.

3.1 Generator

Electrical generator is a machine that converts rotational-mechanical power P_m into electrical power P_e . In our case $P_m = P_r$ so with the use of (2.2) and (2.3) and with the assumption that the efficiency factor $\eta = 1$ we have

$$P_m = P_r = P_e = \Omega_r Q_g$$

To get a linear representation of this relation first order Taylor series expansion is used.

$$\begin{aligned}
P_e - \bar{P}_e &= \Delta P_e \cong \left. \frac{\partial P_e}{\partial \Omega_g} \right|_{\bar{\Omega}_g} (\Omega_g - \bar{\Omega}_g) + \left. \frac{\partial P_e}{\partial Q_g} \right|_{\bar{Q}_g} (Q_g - \bar{Q}_g) \quad (3.1) \\
&= \left. \frac{\partial P_e}{\partial \Omega_g} \right|_{\bar{\Omega}_g} \Delta \Omega_g + \left. \frac{\partial P_e}{\partial Q_g} \right|_{\bar{Q}_g} \Delta Q_g \\
&= \bar{Q}_g \Delta \Omega_g + \bar{\Omega}_g \Delta Q_g
\end{aligned}$$

where "–" above a variable denotes a linearization point.

3.2 Flexible drivetrain shaft

The generator is receiving power from the rotor through a system of two shafts and a gearbox. One of the shafts is (LSS - on the rotor side) is considered to be flexible while the other one (HSS - on the generator side) is considered to be rigid. In steady state the effects of this are negligible and so

$$\Omega_r - \frac{\Omega_g}{N_g} = 0 \qquad \phi_r - \frac{\phi_g}{N_g} = 0$$

where N_g is the gearbox ratio and ϕ_g and ϕ_r are the angular positions of the generator shaft and the rotor shaft respectively. In the transient state however those relations change due to occurring torsion of one of the drivetrain shafts. The above equations take the form

$$\Omega_r - \frac{\Omega_g}{N_g} = \Omega_\Delta \qquad \phi_r - \frac{\phi_g}{N_g} = \phi_\Delta \quad (3.2)$$

where

$$\Omega_\Delta \neq 0 \qquad \phi_\Delta \neq 0 \qquad \Omega_\Delta = \dot{\phi}_\Delta$$

According to [14] the equations relating those quantities are as follows

$$\dot{\Omega}_r = \frac{Q_r}{J_r} - \frac{K_s}{J_r} \phi_\Delta - \frac{D_s}{J_r} \Omega_r + \frac{D_s}{J_r N_g} \Omega_g \quad (3.3a)$$

$$\dot{\Omega}_g = -\frac{Q_g}{J_g} + \frac{K_s}{J_g N_g} \phi_\Delta + \frac{D_s}{J_g N_g} \Omega_r - \frac{D_s}{J_g N_g^2} \Omega_g \quad (3.3b)$$

where J_r and J_g are the moments of inertia of rotor and generator respectively, D_s is the dampening constant of the drivetrain and K_s is it's spring constant.

From (2.2) it is known that Q_r is a non-linear function of Ω_r , θ and v . linearization with the use of the Taylor series gives

$$Q_r - \bar{Q}_r = \Delta Q_r \cong \left. \frac{\partial Q_r}{\partial \Omega_r} \right|_{\bar{\Omega}_r} \Delta \Omega_r + \left. \frac{\partial Q_r}{\partial \theta} \right|_{\bar{\theta}} \Delta \theta + \left. \frac{\partial Q_r}{\partial v} \right|_{\bar{v}} \Delta v \quad (3.4)$$

Connecting the above equation with (3.2) and (3.3) gives the representation of the drive train system which can be used as a part of our model. Keeping in mind that it should have ΔQ_g and $\Delta \theta$ as it's inputs and wind change Δv as it's disturbance the below state space form can be derived

$$\Delta \dot{x}_{dt} = \begin{bmatrix} \Delta \dot{\Omega}_r \\ \Delta \dot{\Omega}_g \\ \Delta \dot{\phi}_\Delta \end{bmatrix} = \overbrace{\begin{bmatrix} \frac{1}{J_r} \left. \frac{\partial Q_r}{\partial \Omega_r} \right|_{\bar{\Omega}_r} & -\frac{D_s}{J_r} & \frac{D_s}{J_r N_g} & -\frac{K_s}{J_r} \\ \frac{D_s}{J_g N_g} & -\frac{D_s}{J_g N_g^2} & \frac{K_s}{J_g N_g} & 0 \\ 1 & \frac{1}{N_g} & 0 & 0 \end{bmatrix}}^{A_{dt}} \begin{bmatrix} \Delta \Omega_r \\ \Delta \Omega_g \\ \Delta \phi_\Delta \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \frac{1}{J_r} \left. \frac{\partial Q_r}{\partial \theta} \right|_{\bar{\theta}} \\ -\frac{1}{J_g} & 0 \\ 0 & 0 \end{bmatrix}}_{B_{dt}} \begin{bmatrix} \Delta Q_g \\ \Delta \theta \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{J_r} \left. \frac{\partial Q_r}{\partial v} \right|_{\bar{v}} \\ 0 \\ 0 \end{bmatrix}}_{B_{v_{dt}}} \Delta v \quad (3.5)$$

The J_r , J_g , N_g , K_s and D_s values can be easily obtained but the partial derivatives of Q_r are different for every linearization point. They will be recovered with the help of FAST in section 3.6.

Connecting the above model with the generator (see (3.1)) yields the output part of the system.

$$\begin{bmatrix} P_e \\ \Omega_r \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & \bar{Q}_g & 0 \\ 1 & 0 & 0 \end{bmatrix}}^{C_{dt}} \begin{bmatrix} \Delta \Omega_r \\ \Delta \Omega_g \\ \Delta \phi_\Delta \end{bmatrix} + \overbrace{\begin{bmatrix} \bar{\Omega}_g & 0 \\ 0 & 0 \end{bmatrix}}^{D_{dt}} \begin{bmatrix} \Delta Q_g \\ \Delta \theta \end{bmatrix}$$

3.3 Generator torque actuator

According to [14] generator torque actuator can be approximated by a first order system

$$\frac{dQ_g}{dt} = \frac{Q_{g_{ref}} - Q_g}{\tau_g}$$

where τ_g is the time constant of the generator and $Q_{g_{ref}}$ is the reference value for the actuator's output. It is equivalent to a relative state space representation

of the form

$$\Delta \dot{x}_{Q_g} = \Delta \dot{Q}_g = \underbrace{\begin{bmatrix} -1 \\ \tau_g \end{bmatrix}}_{A_{Q_g}} \Delta Q_g + \underbrace{\begin{bmatrix} 1 \\ \tau_g \end{bmatrix}}_{B_{Q_g}} \Delta Q_{gref}$$

3.4 Collective blade pitch actuator

According to [14] collective blade pitch actuator can be approximated by a second order system

$$\frac{d^2\theta}{dt^2} = \omega_n^2 \theta_{ref} - \frac{d\theta}{dt} 2\omega_n \zeta - \omega_n^2 \theta$$

where ω_n is the natural frequency of the actuator and ζ is it's damping constant. State space representation, in a relative form, of this equation is

$$\Delta \dot{x}_\theta = \begin{bmatrix} \Delta \dot{\theta} \\ \Delta \ddot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}}_{A_\theta} \begin{bmatrix} \Delta \theta \\ \Delta \dot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}}_{B_\theta} \Delta \theta_{ref}$$

3.5 Wind

Wind can be modelled as a complicated system but in this work it will approximated as second order model as proposed in [6]. It assumes that there are two two components of the actual wind speed v :

- v_m - slow varying mean wind speed modelled as having no relevant dynamics
- v_t - fast varying turbulent wind speed

A broad introduction to the topic of geographical and long term variations of the wind speed is given in [4] and won't be addressed in this thesis.

By adopting the wind model from [6] we can write

$$v = v_t + v_m$$

where,

$$v_t = \frac{k}{(p_1 s + 1)(p_2 s + 1)} e, \quad e \in N(0, 1) \quad (3.6)$$

with k , p_1 and p_2 being functions of the mean wind speed v_m . Their values in respect to v_m are shown on figure 3.1.

It has to be noted that the model in [6] was based on a more complex one presented in [20] and [14]. Relation (3.6) can also be transformed into a state

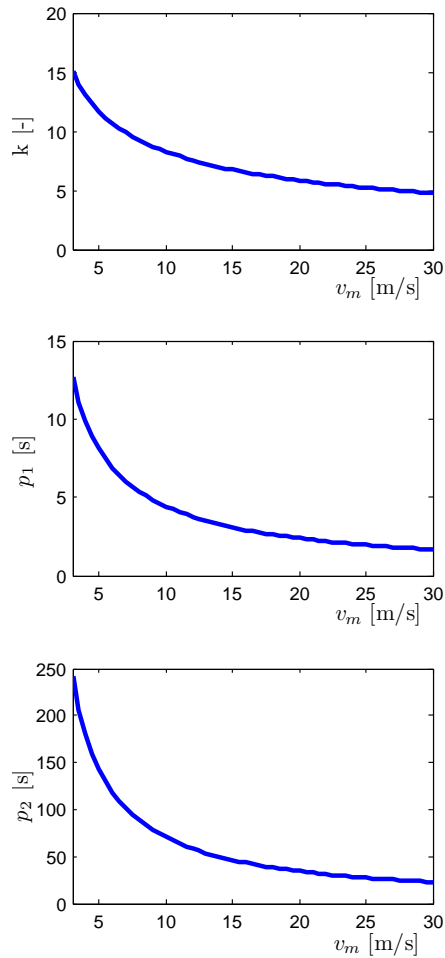


Figure 3.1: Wind model parameters as a function of the mean wind speed v_m

space representation

$$\Delta \dot{x}_v = \begin{bmatrix} \Delta \dot{v}_t \\ \Delta \ddot{v}_t \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 \\ -\frac{1}{p_1 p_2} & -\frac{p_1 + p_2}{p_1 p_2} \end{bmatrix}}^{A_v} \begin{bmatrix} \Delta v_t \\ \Delta \dot{v}_t \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ \frac{k}{p_1 p_2} \end{bmatrix}}^{B_v} e$$

It is assumed here that the wind speed is uniformly distributed over the whole area of the rotor disc and that the *wind shear* (the increase of the wind speed together with height) doesn't take place.

3.6 FAST

One of the tools that are integral part of FAST is the numerical linearization module. It is used in order to derive parameters that together with the state space representations of the wind turbine, presented in the previous sections, will provide us with a fully functional model:

- $\left. \frac{\partial Q_r}{\partial \Omega_r} \right|_{\bar{\Omega}_r}, \left. \frac{\partial Q_r}{\partial \theta} \right|_{\bar{\theta}}, \left. \frac{\partial Q_r}{\partial v} \right|_{\bar{v}}$
- linearization-operation points for inputs, outputs and states

The rest of the parameters needed (e.g. damping and spring constants) for the model can be easily obtained from the reference document of the particular HAWT such as [11]. Furthermore FAST itself has to be provided with some of those parameters in order for it to be able to simulate given wind turbine type with satisfactory precision.

Below specific actions that have to be taken in order to acquire useful, in our case, linear models from FAST (from which the given parameters will be extracted) will be presented. FAST configuration parameters, that are the most important in this process will be shortly addressed in appendix A. The whole procedure of configuring FAST for linearization purposes is presented in [10].

3.6.1 Choosing the proper model

The partial derivatives of Q_r , mentioned above, are parts of the drivetrain model derived in section 3.2. Unfortunately trying to acquire this model with the help

of FAST will result in a 4th order system with states

$$\Delta x = \begin{bmatrix} \Delta\phi_r \\ \Delta\phi_g \\ \Delta\Omega_r \\ \Delta\Omega_g \end{bmatrix} \quad \text{instead of} \quad \Delta x = \begin{bmatrix} \Delta\phi_\Delta \\ \Delta\Omega_r \\ \Delta\Omega_g \end{bmatrix}$$

Although this system is useful for simulation purposes with the use of FAST it would be complicated to design a controller for it e.g. because it (FAST) is zeroing the indicators of their shafts azimuths every 360 [°] while in a linear model they would rise to infinity.

Instead a 1st order system with generator's angular velocity Ω_g as a state will be utilized. In this case the assumption is that the drivetrain is perfectly stiff and so from (3.3) we have

$$J_r \dot{\Omega}_r = Q_r - K_s \phi_\Delta - D_s \Omega_r + \frac{D_s}{N_g} \Omega_g \quad (3.7)$$

$$J_g N_g \dot{\Omega}_g = -N_g Q_g + K_s \phi_\Delta + D_s \Omega_r - \frac{D_s}{N_g} \Omega_g \quad (3.8)$$

Adding those two equations together and remembering that generators inertia viewed from the rotor side is $J_{g|rotor} = J_g N_g^2$ and $\Omega_r = \frac{\Omega_g}{N_g}$

$$(J_r + J_{g|rotor}) \dot{\Omega}_r = Q_r - N_g Q_g$$

By denoting $J_t = J_r + J_{g|rotor}$ (total rotating inertia) and using (3.4) we get

$$\dot{\Omega}_r = \frac{1}{J_t} \left. \frac{\partial Q_r}{\partial \Omega_r} \right|_{\bar{\Omega}_r} \Delta\Omega_r + \frac{1}{J_t} \left. \frac{\partial Q_r}{\partial \theta} \right|_{\bar{\theta}} \Delta\theta + \frac{1}{J_t} \left. \frac{\partial Q_r}{\partial v} \right|_{\bar{v}} \Delta v - N_g Q_g \quad (3.9)$$

FAST model The linear model of the wind turbine, with only generator's degree of freedom enabled, obtained from FAST yields the second order system with the state vector of

$$\Delta x = \begin{bmatrix} \Delta\phi_r \\ \Delta\Omega_r \end{bmatrix}$$

Notice that those generator states are viewed with respect to the rotor side of the gearbox. If we would remove the first state we would end up with the following model

$$\dot{\Omega}_r = a\Omega_r + b_1 Q_g + b_2 \theta + b_v v$$

By comparing it with (3.9) we get the sought after partial derivatives of Q_r .

$$\left. \frac{\partial Q_r}{\partial \Omega_r} \right|_{\bar{\Omega}_r} = a J_t \quad \left. \frac{\partial Q_r}{\partial \theta} \right|_{\bar{\theta}} = b_2 J_t \quad \left. \frac{\partial Q_r}{\partial v} \right|_{\bar{v}} = b_v J_t$$

Those parameters vary with wind speed and the way in which they are obtained is slightly different in each of the operation modes. Hence there is a need for deriving a set of models that would cover the whole operational wind speed spectrum (from cut-in to cut-out wind speed). The procedure of setting up the FAST linearization tool and deriving a collection of linear models for a given range of wind speeds is described in appendix A.

3.7 Chapter summary

In this chapter the derivation of the WT1 linearized model has been discussed. It has to be kept in mind that a whole set of those models for different values of wind speed v will be needed in order to be able to control the wind turbine in all of the operation regions. In the next chapter HAWT model will be transformed into a form more suitable for the whole wind speed range control purpose - an affine model. Furthermore constraints, that will have to be taken into consideration while designing the controller in part II, will be introduced.

Affine model with constraints

Here, the models of the wind turbine's subsystems, derived in the previous chapter, will be put together and then transformed into an affine system representation. Furthermore the concept of constraint's will be introduced in the last section.

4.1 Complete model

In order to obtain a complete model from those that have been defined in chapter 3 systems presented there will be combined into one model.

$$\Delta \dot{x} = \begin{bmatrix} \Delta \dot{x}_{dt} \\ \Delta \dot{x}_{Q_g} \\ \Delta \dot{x}_\theta \\ \Delta \dot{x}_{v_t} \end{bmatrix} = \overbrace{\begin{bmatrix} A_{dt} & B_{dt_1} & B_{dt_2} & 0 & B_{dt_v} & 0 \\ 0 & A_{Q_g} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_v & 0 \end{bmatrix}}^{A_c} \begin{bmatrix} \Delta x_{dt} \\ \Delta x_{Q_g} \\ \Delta x_\theta \\ \Delta x_{v_t} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ B_{Q_g} & 0 \\ 0 & B_\theta \\ 0 & 0 \end{bmatrix}}_{B_c} \begin{bmatrix} \Delta Q_g \\ \Delta \theta \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ B_v \end{bmatrix}}_{E_c} e \quad (4.1a)$$

$$y = \begin{bmatrix} P_e \\ \Omega_r \end{bmatrix} = \underbrace{\begin{bmatrix} C_{dt} & [D_{dt} & 0] & 0 \end{bmatrix}}_{C_c} \begin{bmatrix} \Delta x_{dt} \\ \Delta x_{Q_g} \\ \Delta x_\theta \\ \Delta x_{v_t} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D_c} \begin{bmatrix} \Delta Q_g \\ \Delta \theta \end{bmatrix} \quad (4.1b)$$

It's linearization points are

$$\bar{x} = \begin{bmatrix} \bar{x}_{dt} \\ \bar{x}_{Q_g} \\ \bar{x}_\theta \\ \bar{x}_{v_t} \end{bmatrix}, \quad \bar{y} = \begin{bmatrix} \bar{P}_e \\ \bar{\Omega}_r \end{bmatrix}, \quad \bar{u} = \begin{bmatrix} \bar{Q}_g \\ \bar{\theta} \end{bmatrix} \quad (4.2)$$

4.2 Affine model

The model that have been presented in the previous section is a relative (incremental) one what is the most common approach in control. In this thesis however, due to the fact that the controller, that will be designed in the following part of the work, will be working in all of the operation modes, it is beneficial to transform the model that will be used into an affine one.

Continuing with the results from section 4.1 it can be written that

$$\left. \begin{array}{l} \Delta \dot{x} = A_c \Delta x + B_c \Delta u + E_c e \\ \Delta y = C_c \Delta x + D_c \Delta u \end{array} \right\} \Rightarrow \begin{array}{l} \dot{x} - \bar{x} = A_c(x - \bar{x}) + B_c(u - \bar{u}) + E_c e \\ y - \bar{y} = C_c(x - \bar{x}) + D_c(u - \bar{u}) \end{array}$$

and further

$$\begin{aligned} \dot{x} &= A_c x + B_c u + E_c e + (-A_c \bar{x} - B_c \bar{u} + \bar{x}) \\ y &= C_c x + D_c u + (-C_c \bar{x} - D_c \bar{u} + \bar{y}) \end{aligned}$$

by denoting

$$\delta_c = -A_c \bar{x} - B_c \bar{u} + \bar{x} \quad \gamma_c = -C_c \bar{x} - D_c \bar{u} + \bar{y}$$

One can rewrite the relative model as a function of global variables x , u and y

$$\begin{aligned} \dot{x} &= A_c x + B_c u + E_c e + \delta_c \\ y &= C_c x + D_c u + \gamma_c \end{aligned} \quad (4.3)$$

where δ_c and γ_c are functions of the linearization points.

4.3 Constraints

As mentioned in the introduction the tool that will be used for control purposes - Model Predictive Control - is perfect for handling constraints what is one of it's

biggest advantages. Below the constraints that will be included in the control problem are presented .

- Output constraints

- for produced power P_e

$$0 \leq P_e \leq P_{e,nom}$$

- for rotors rotational speed Ω_r

$$\Omega_{r,min} \leq \Omega_r \leq \Omega_{r,max}$$

- Input constraints

- for generator torque Q_g

$$0 \leq Q_g \leq Q_{g,max}$$

$$\dot{Q}_{g,min} \leq \dot{Q}_g \leq \dot{Q}_{g,max}$$

- for collective blade pitch θ

$$\theta_{min} \leq \theta \leq \theta_{max}$$

$$\dot{\theta}_{min} \leq \dot{\theta} \leq \dot{\theta}_{max}$$

4.4 Chapter summary

In this chapter the complete relative model of the WT1 wind turbine has been put together and transformed into an affine form which will be used in the next part of the thesis, in the MPC controller design. It will also include the constraints that have been introduced here.

Part II

Control methods

Part II introduction

In this part the main ideas of Model Predictive Control (MPC) and its use in HAWT control will be presented. As stated in the introduction this thesis is focussing on controlling the wind turbine in all of the operation regions. Furthermore our goal is to achieve a smooth transition between them, contrary to a different approach of switching from one controller to another when the operation mode changes, as it is the case in [6].

In order to be able to perform control in the whole spectrum of wind speeds *gain scheduling* technique will be presented. Additionally *weight scheduling* approach will be implemented in order to further improve performance in different modes of operation.

Modelling errors, unmeasured disturbances and the necessity of acquiring proper value of the wind speed will be addressed in the section dedicated to offset free control.

It will be assumed here that the model (4.3) derived in the previous section have become discretized with the sampling period of T_s yielding

$$\begin{aligned}
 x_{k+1} = \begin{bmatrix} x_{dt} \\ x_{Q_g} \\ x_{\theta} \\ x_{v_t} \end{bmatrix}_{k+1} &= \overbrace{\begin{bmatrix} A_{dt,d} & B_{dt1,d} & B_{dt2,d} & 0 & B_{dtv,d} & 0 \\ 0 & A_{Q_g,d} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{\theta,d} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{v,d} & 0 \end{bmatrix}}^{A_d} \begin{bmatrix} x_{dt} \\ x_{Q_g} \\ x_{\theta} \\ x_{v_t} \end{bmatrix}_k + \\
 &\quad \underbrace{\begin{bmatrix} 0 & 0 \\ B_{Q_g,d} & 0 \\ 0 & B_{\theta,d} \\ 0 & 0 \end{bmatrix}}_{B_d} \begin{bmatrix} Q_g \\ \theta \end{bmatrix}_k + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ B_{v,d} \end{bmatrix}}_{E_d} e_k + \delta_d \quad (\text{II.0})
 \end{aligned}$$

$$y_k = \begin{bmatrix} P_e \\ \Omega_r \end{bmatrix}_k = \underbrace{\begin{bmatrix} C_{dt,d} & [D_{dt,d} & 0] & 0 \end{bmatrix}}_{C_d} \begin{bmatrix} x_{dt} \\ x_{Qg} \\ x_\theta \\ x_{vt} \end{bmatrix}_k + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D_d} \begin{bmatrix} Q_g \\ \theta \end{bmatrix}_k + \gamma_d$$

Where

$$\delta_d = -A_d \bar{x} - B_d \bar{u} + \bar{x} \qquad \gamma_d = -C_d \bar{x} - D_d \bar{u} + \bar{y}$$

and "d" subscript for the internal matrices ($A_{dt,d}$ for example) represents their version obtained after discretizing the whole continuous system. In the following sections $E_d e_k$ will be represented, with the abuse of notation, simply as e_k .

CHAPTER 5

Offset-free control and wind estimation

In order to achieve offset-free control one must compensate for unmeasured disturbances, such as modelling errors. Furthermore without the assumption that the wind is uniformly distributed over the rotor disc area achieving it's reliable measurement would also be quite difficult. Due to these considerations we will attempt to estimate both the unmeasured disturbances and effective wind speed value.

First the augmentation of the discrete version of the model derived in the previous chapter (II.0) with the models of unmeasured disturbances will be presented. The concept of the estimating them with the use of kalman filter will be discussed next.

5.1 Disturbance modelling

Unmeasured disturbances will be modelled, as having an integrating character, in a way proposed in [16]

$$\begin{aligned}d_{k+1} &= d_k + e_k \\ p_{k+1} &= p_k + e_k\end{aligned}$$

Augmenting system (II.0) with them yields

$$\begin{aligned}x_{e,k+1} = \begin{bmatrix} x_d \\ d \\ p \end{bmatrix}_{k+1} &= \overbrace{\begin{bmatrix} A_d & B_{dp} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{A_e} \begin{bmatrix} x_d \\ d \\ p \end{bmatrix}_k + \overbrace{\begin{bmatrix} B_d \\ 0 \\ 0 \end{bmatrix}}^{B_e} u_k + e_k + \delta_e \\ y_{e,k} = \underbrace{\begin{bmatrix} C_d & D_{dp} \end{bmatrix}}_{C_e} \begin{bmatrix} x_d \\ d \\ p \end{bmatrix}_k + \underbrace{\begin{bmatrix} D_d \end{bmatrix}}_{D_e} u_k + w_k + \gamma_e\end{aligned}\quad (5.1)$$

Where w_k , representing the measurement noise, is being added for the sake of generalizing the problem.

5.2 Disturbance estimation with the use of Kalman filter

In order to estimate the values of d and p disturbances as well as the wind speed an optimal observer is built in a form of a kalman filter

$$\begin{aligned}\hat{x}_{e,k|k} &= \hat{x}_{e,k|k-1} + L[y_m - (C_e \hat{x}_{e,k|k-1} + D_e u_k + \gamma_e)] \\ \hat{x}_{e,k+1|k} &= A_e \hat{x}_{e,k|k} + B_e u_k + \delta_e\end{aligned}$$

where " $\hat{\cdot}$ " denotes estimated value, y_m measurement and value on the right hand side of " $|$ " in $(k|k-i)$ represents the iteration at which the estimation has been carried out. L is so called *kalman gain matrix* which is calculated by solving a Riccati equation which is a function of state and measurement noise covariance matrices, $Q_e = E\{ee^T\}$ and $R_e = E\{ww^T\}$ respectively. In practice those matrices are used as a tuning parameter for the kalman filter. A broad discussion on fundamentals of using this tool is presented in [5] and it goes beyond the scope of this work.

According to [3] where the same setup for the disturbance augmented model is

used system (5.1) is observable if and only if the pair of matrices (C_d, A_d) of the original system is observable and

$$\begin{bmatrix} A_d - I & B_{dp} \\ C_d & D_{dp} \end{bmatrix} \text{ has a full column rank} \quad (5.2)$$

With the condition on (C_d, A_d) matrices being observable this basically means that the augmented system has to guarantee that the deduction of unique values of the disturbances, provided that the output measurements are available, will be possible in steady state.

Offset free control is achieved by using the augmented system (5.1) instead of (II.0) as a base for the controller setup. The estimates of the unmeasured disturbances included in the model in such way will allow to compensate for, among others, modelling errors. Furthermore, due to the fact that the model used is an affine one the values of the disturbances will also include the errors in the determination of the linearization points. This, provided that the noise covariance matrices Q_e and R_e are chosen correctly, will allow us to get not only a zero offset response in the steady state but also determine the actual effective speed of the wind.

5.3 Chapter summary

Unmeasured disturbances and effective wind speed estimation has been discussed in this chapter. Model Predictive Controller will be using them in order to perform an offset-free control. It will be discussed in the next chapter.

Model Predictive Control

Model-Based Predictive Control strategy will be discussed in this chapter. First its basic principles will be introduced. Next, prediction of the outputs and definition of the cost function will be carried out. Since those two elements of the MPC problem are treated in a slightly different way from the standard approach they will be presented in details. Reformulation of the constraints into an MPC usable form will follow. In that section the aspect of softening the output constraints will be presented in details as well. In the last section all of the above concepts will be put together yielding a softly-constrained quadratic programming problem whose solution is the main task of the MPC controller. The basic idea standing behind Model Predictive Control is depicted on figure 6.1. Having defined a reference trajectory for the output of the plant $r(t)$ we want to track it in an optimal way. Namely, we want to balance between the index referring to the tracking error and other performance indexes such as the aggressiveness of the control action. In the current sampling instant an output prediction trajectory $z(t|k)$ is being calculated. It represents the fashion in which the reference trajectory $r(t)$ should be reached by the output signal $y(t)$. It is defined over a certain number of future samples known as the *prediction horizon* P . The plant's inputs, which will be responsible for driving the system along the prediction trajectory $z(t|k)$, are assumed to be changing over a certain number of samples, called the *control horizon* M , and stay constant afterwards. With the knowledge of the plant's dynamics, in a form of a model, a formula describing system's outputs evolution, over a given prediction horizon P is de-

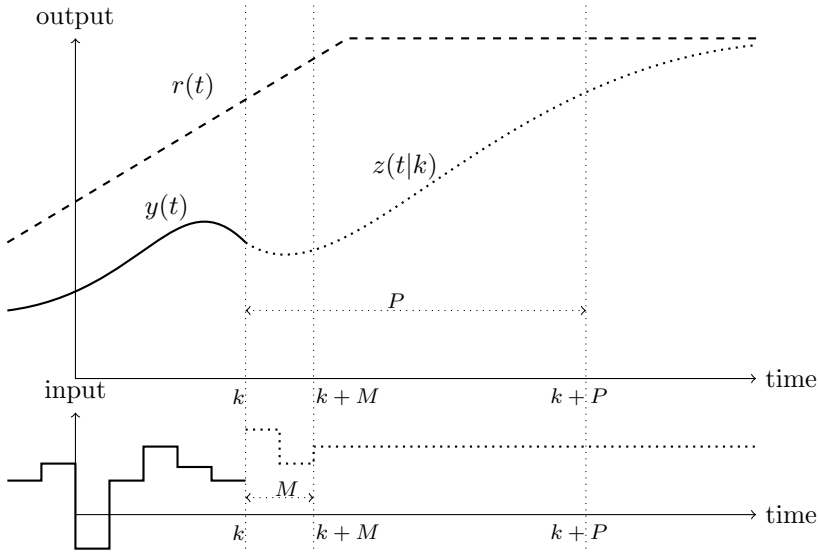


Figure 6.1: The basic idea behind Model Predictive Control

rived. Next a cost function, describing the optimal, from the designers point of view, balance between certain characteristics of the plant's behaviour, is defined. It is often not an easy task since the individual control objectives which are reflected there are often opposite in nature. The value of the *decision vector* that minimizes the cost function is, in theory, the one that will allow the system to follow the prediction trajectory $z(t|k)$ in the way that is most satisfactory. It typically consists of the values of the current and future inputs to the systems $\Delta\mathcal{U}(k)$ but can also contain other variables too (like constraints *violation margin* - see 6.3.4).

Once a set of future inputs to the system has been computed only the first one is applied to the plant. In the next iteration of the algorithm (at time $k + 1$) the cycle will repeat. Both prediction P and control M horizons will be shifted forward in time (but will preserve their length) by one sample, new set of future inputs will be obtained and again only the first one will be used. This approach is called the *receding horizon strategy*.

In the model that will be used throughout this chapter the wind speed v and unmeasured integrating disturbances, d and p , are treated as external disturbances with unknown dynamics and only the influence that they have on the rest of the system, in the form of $B_{d,m}$ and $D_{d,m}$ matrices, is being considered.

Namely,

$$\begin{aligned}
 x_{k+1} = \begin{bmatrix} x_{dt} \\ x_{Qg} \\ x_{\theta} \end{bmatrix}_{k+1} &= \overbrace{\begin{bmatrix} A_{dt,d} & B_{dt_1,d} & [B_{dt_2,d} & 0] \\ 0 & A_{Qg,d} & 0 \\ 0 & 0 & A_{\theta,d} \\ 0 & 0 & 0 \end{bmatrix}}^{A_m} \begin{bmatrix} x_{dt} \\ x_{Qg} \\ x_{\theta} \end{bmatrix}_k + \\
 &\underbrace{\begin{bmatrix} 0 & 0 \\ B_{Qg,d} & 0 \\ 0 & B_{\theta,d} \end{bmatrix}}_{B_m} \begin{bmatrix} Q_g \\ \theta \end{bmatrix}_k + \underbrace{[[B_{dt_v,d} & 0] & B_{dp}]}_{B_{d,d}} \underbrace{\begin{bmatrix} x_{v_t} \\ d \\ p \end{bmatrix}_k}_{d_k} + \delta_m \quad (6.1a)
 \end{aligned}$$

$$\begin{aligned}
 y_k = \begin{bmatrix} P_e \\ \Omega_r \end{bmatrix}_k &= \underbrace{[C_{dt,d} & [D_{dt,d} & 0]]}_{C_m} \begin{bmatrix} x_{dt} \\ x_{Qg} \\ x_{\theta} \end{bmatrix}_k + \underbrace{[0]}_{D_m} \begin{bmatrix} Q_g \\ \theta \end{bmatrix}_k + \underbrace{[[0 & 0] & D_{dp}]}_{D_{d,m}} \underbrace{\begin{bmatrix} x_{v_t} \\ d \\ p \end{bmatrix}_k}_{d_k} + \gamma_m \\
 &\quad (6.1b)
 \end{aligned}$$

where

$$\delta_m = -A_m \bar{x} - B_m \bar{u} + \bar{x} \quad \gamma_m = -C_m \bar{x} - D_m \bar{u} + \bar{y}$$

This approach allows to treat the disturbances independently from the system what will be the case in this work. It gives a choice of choosing what source of disturbance estimates will be used and in theory enables the possibility of including the values of the future wind speeds into the disturbance trajectory (see $\mathcal{D}(\|)$ in (6.3)) if only they are known.

6.1 Output prediction

System's response prediction (evolution) formula will be derived on the following pages. Contrary to the approach usually presented in the literature (see [17], [13]), where no direct influence of the inputs on the system outputs is being assumed, the assumption here is that the system that is being considered is not strictly proper i.e. $D_m \neq 0$. The below calculations extend the the strictly proper case presented in [13].

$$\begin{aligned}
 x_{k+1} &= A_m x_k + B_m u_k + B_{d,m} d_k + \delta_m \\
 y_k &= C_m x_k + D_m u_k + D_{d,m} d_k + \gamma_m
 \end{aligned} \quad (6.2)$$

Consider the system presented in (6.2). Expressions for the future states of this system can be derived as follows

$$\begin{aligned}
 x_{k+2} &= A_m x_{k+1} + B_m u_{k+1} + B_{d,m} d_{k+1} + \delta_m = \\
 &\quad A_m (A_m x_k + B_m u_k + B_{d,m} d_k + \delta_m) + \\
 &\quad B_m u_{k+1} + B_{d,m} d_{k+1} + \delta_m = \\
 &\quad A_m^2 x_k + A_m B_m u_k + A_m B_{d,m} d_k + A_m \delta_m + \\
 &\quad B_m u_{k+1} + B_{d,m} d_{k+1} + \delta_m
 \end{aligned}$$

$$\begin{aligned}
 x_{k+3} &= A_m x_{k+2} + B_m u_{k+2} + B_{d,m} d_{k+2} + \delta_m = \\
 &\quad A_m^3 x_k + A_m^2 B_m u_k + A_m^2 B_{d,m} d_k + A_m^2 \delta_m + \\
 &\quad A_m B_m u_{k+1} + A_m B_{d,m} d_{k+1} + A_m \delta_m + \\
 &\quad B_m u_{k+2} + B_{d,m} d_{k+2} + \delta_m
 \end{aligned}$$

hence, if future outputs $y(k+i)$ would be denoted as output predictions z_k they would become

$$\begin{aligned}
 z_{k+1} &= C_m x_{k+1} + D_m u_{k+1} + D_{d,m} d_{k+1} + \gamma_m = \\
 &\quad C_m (A_m x_k + B_m u_k + B_{d,m} d_k + \delta_m) + \\
 &\quad D_m u_{k+1} + D_{d,m} d_{k+1} + \gamma_m = \\
 &\quad C_m A_m x_k + C_m B_m u_k + C_m B_{d,m} d_k + C_m \delta_m + \\
 &\quad D_m u_{k+1} + D_{d,m} d_{k+1} + \gamma_m
 \end{aligned}$$

$$\begin{aligned}
 z_{k+2} &= C_m x_{k+2} + D_m u_{k+2} + D_{d,m} d_{k+2} + \gamma_m = \\
 &\quad C_m A_m^2 x_k + C_m A_m B_m u_k + C_m A_m B_{d,m} d_k + \\
 &\quad C_m A_m \delta_m + C_m B_m u_{k+1} + C_m B_{d,m} d_{k+1} \\
 &\quad + C_m \delta_m + D_m u_{k+2} + D_{d,m} d_{k+2} + \gamma_m
 \end{aligned}$$

$$\begin{aligned}
 z_{k+3} &= C_m x_{k+3} + D_m u_{k+3} + D_{d,m} d_{k+3} + \gamma_m = \\
 &\quad C_m A_m^3 x_k + C_m A_m^2 B_m u_k + C_m A_m^2 B_{d,m} d_k + \\
 &\quad C_m A_m^2 \delta_m + C_m A_m B_m u_{k+1} + C_m A_m B_{d,m} d_{k+1} + \\
 &\quad C_m A_m \delta_m + C_m B_m u_{k+2} + C_m B_{d,m} d_{k+2} + \\
 &\quad C_m \delta_m + D_m u_{k+3} + D_{d,m} d_{k+3} + \gamma_m
 \end{aligned}$$

The above results can be presented in a matrix form. Additionally the outputs

$$\begin{aligned}
 \overbrace{\begin{bmatrix} z_{k+1} \\ z_{k+2} \\ z_{k+3} \\ \vdots \\ z_{k+P} \end{bmatrix}}^{\mathcal{Z}(k)} &= \overbrace{\begin{bmatrix} C_m A_m \\ C_m A_m^2 \\ C_m A_m^3 \\ \vdots \\ C_m A_m^P \end{bmatrix}}^{\Psi} x_k + \\
 &\overbrace{\begin{bmatrix} C_m B_m & D_m & 0 & 0 & \dots & 0 \\ C_m A_m B_m & C_m B_m & D_m & 0 & \dots & 0 \\ C_m A_m^2 B_m & C_m A_m B_m & C_m B_m & D_m & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ C_m A_m^{P-1} B_m & C_m A_m^{P-2} B_m & \dots & \dots & C_m B_m & D_m \end{bmatrix}}^{\Theta'} \overbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ u_{k+3} \\ \vdots \\ u_{k+M-1} \end{bmatrix}}^{\mathcal{U}(k)} + \\
 &\overbrace{\begin{bmatrix} C_m B_{d,m} & D_{d,m} & 0 & 0 & \dots & 0 \\ C_m A_m B_{d,m} & C_m B_{d,m} & D_{d,m} & 0 & \dots & 0 \\ C_m A_m^2 B_{d,m} & C_m A_m B_{d,m} & C_m B_{d,m} & D_{d,m} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ C_m A_m^{P-1} B_{d,m} & C_m A_m^{P-2} B_{d,m} & \dots & \dots & C_m B_{d,m} & D_{d,m} \end{bmatrix}}^{\Xi} \overbrace{\begin{bmatrix} d_k \\ d_{k+1} \\ d_{k+2} \\ d_{k+3} \\ \vdots \\ d_{k+P-1} \end{bmatrix}}^{\mathcal{D}(k)} + \\
 &\overbrace{\begin{bmatrix} C_m \\ C_m A_m + C_m \\ C_m A_m^2 + C_m A_m + C_m \\ \vdots \\ \sum_{i=0}^{P-1} C_m A_m^i \end{bmatrix}}^{\Phi} \delta_m + \overbrace{\begin{bmatrix} I_z \\ I_z \\ I_z \\ \vdots \\ I_z \end{bmatrix}}^{I_p} \gamma_m \quad (6.3)
 \end{aligned}$$

where $I_z \in \mathbb{R}^{n_z \times n_z}$ for n_z being the number of outputs that are being predicted. Furthermore,

$$u_k = u_{k-1} + (u_k - u_{k-1}) = u_{k-1} + \Delta u_k \quad (6.4)$$

$$u_{k+1} = u_{k-1} + (u_k - u_{k-1}) + (u_{k+1} - u_k) = u_{k-1} + \Delta u_k + \Delta u_{k+1} \quad (6.5)$$

$$u_{k+2} = u_{k-1} + (u_k - u_{k-1}) + (u_{k+1} - u_k) + (u_{k+2} - u_{k+1}) = \quad (6.6)$$

$$= u_{k-1} + \Delta u_k + \Delta u_{k+1} + \Delta u_{k+2} \quad (6.7)$$

$$\vdots \quad (6.8)$$

What in a matrix form is

$$\begin{array}{c} \overbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ u_{k+3} \\ \vdots \\ u_{k+M-1} \end{bmatrix}}^{\mathcal{U}(k)} = \overbrace{\begin{bmatrix} I_u \\ I_u \\ I_u \\ \vdots \\ I_u \end{bmatrix}}^{I_m} u_{k-1} + \overbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \ddots & 0 \\ 1 & 1 & 1 & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 1 & 1 & \dots & \dots & 1 \end{bmatrix}}^{H_I} \overbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \Delta u_{k+3} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix}}^{\Delta \mathcal{U}(k)} \end{array} \quad (6.9)$$

where $I_u \in \mathbb{R}^{n_u \times n_u}$ for n_u being the number of systems inputs. This allows us to write $\Theta' \mathcal{U}(k)$ as

$$\begin{aligned} \Theta' \mathcal{U}(k) &= \Theta' (I_m u_{k-1} + H_I \Delta \mathcal{U}(k)) = \\ &= \overbrace{\begin{bmatrix} C_m B_m + D_m \\ C_m A_m B_m + C_m B_m + D_m \\ C_m A_m^2 B_m + C_m A_m B_m + C_m B_m + D_m \\ \vdots \\ \sum_{i=0}^{P-1} (C_m A_m^i B_m) + D_m \end{bmatrix}}^{\Gamma} u_{k-1} + \\ &+ \overbrace{\begin{bmatrix} \Gamma(1) & D_m & 0 & 0 & \dots & 0 \\ \Gamma(2) & \Gamma(1) & D_m & 0 & \ddots & 0 \\ \Gamma(3) & \Gamma(2) & \Gamma(1) & D_m & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ \Gamma(P) & \Gamma(P-1) & \dots & \dots & \Gamma(1) & D_m \end{bmatrix}}^{\Theta} \overbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \Delta u_{k+3} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix}}^{\Delta \mathcal{U}(k)} \end{aligned} \quad (6.10)$$

Where $\Gamma(P) = \sum_{i=0}^{P-1} (C_m A_m^i B_m) + D_m$ is the P th element (matrix element) of the Γ matrix.

Combining (6.3) and (6.10) together we get

$$\mathcal{Z}(k) = \Psi x_k + \Gamma u_{k-1} + \Theta \Delta \mathcal{U}(k) + \Xi \mathcal{D}(k) + \Phi \delta_m + I_p \gamma \quad (6.11)$$

6.2 Cost function and quadratic programming

The means by which the optimal value of the inputs will be calculated is the minimization of a cost function J . As will be shown below, in case of MPC, it

can be defined as a *quadratic programming* problem.

The cost function that will be the object of minimization is chosen to be

$$J(k) = \frac{1}{2} \sum_{i=1}^P \|z_{k+i|k} - r_{k+1}\|_{Q_i}^2 + \frac{1}{2} \sum_{i=1}^{M-1} \|\Delta u_{k+i|k}\|_{R_i}^2 + \frac{1}{2} \sum_{i=1}^{M-1} \|u_{k+i|k} - \bar{u}\|_{W_i}^2 \quad (6.12)$$

The first two terms in the above formula represent the most common representation of a cost function used in MPC. As proposed in [13] the controller will be operating on the moves of the system inputs instead of their absolute values ($\Delta u_{k+i|k}$ instead of $u_{k+i|k}$). The last term in the cost function is an extension of the standard cost function formulation and it describes the deviation of the calculated inputs from the linearization point of the model. The Q_i , R_i and W_i parameters are the weights that are used as importance indicators. If the issue of different magnitudes (in a non-scaled system) of the system variables would be neglected it could be said that the higher the value of the weight is the bigger impact the respective norm (performance index) has on the value of the cost function and hence will be minimized the strongest (and the most aggressively). Cost function (6.12) can be rewritten in a matrix form

$$J(k) = \frac{1}{2} \|\mathcal{Z}(k) - \mathcal{T}(k)\|_Q^2 + \frac{1}{2} \|\Delta \mathcal{U}(k)\|_R^2 + \frac{1}{2} \|\mathcal{U}(k) - I_m \bar{u}\|_W^2 \quad (6.13)$$

where

$$\mathcal{T}(k) = \begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+P} \end{bmatrix} \quad Q = \begin{bmatrix} Q_{k+1} & 0 & \dots & 0 \\ 0 & Q_{k+2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_{k+P} \end{bmatrix}$$

$$R = \begin{bmatrix} R_k & 0 & \dots & 0 \\ 0 & R_{k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R_{k+M-1} \end{bmatrix} \quad W = \begin{bmatrix} W_k & 0 & \dots & 0 \\ 0 & W_{k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W_{k+M-1} \end{bmatrix}$$

The term that is the subject of the first norm in (6.13) can be rewritten as

$$\begin{aligned} \mathcal{Z}(k) - \mathcal{T}(k) = \\ \Psi x_k + \Gamma u_{k-1} + \Theta \Delta \mathcal{U}(k) + \Xi \mathcal{D}(k) + \Phi \delta_m + I_p \gamma - \mathcal{T}(k) = \\ \Theta \Delta \mathcal{U}(k) - (\mathcal{T}(k) - (\Psi x_k + \Gamma u_{k-1} + \Xi \mathcal{D}(k) + \Phi \delta_m + I_p \gamma)) = \\ \Theta \Delta \mathcal{U}(k) - \mathcal{E}(k) \end{aligned} \quad (6.14)$$

where

$$\mathcal{E}(k) = \mathcal{T}(k) - (\Psi x_k + \Gamma u_{k-1} + \Xi \mathcal{D}(k) + \Phi \delta_m + I_p \gamma)$$

Furthermore since the controller will be working on increments of the input signals $\mathcal{U}(k)$, (6.13) has to be represented as a function of $\Delta\mathcal{U}(k)$. Using the results from (6.9) it can be written that

$$\begin{aligned} \mathcal{U}(k) - I_m \bar{u} &= H_I \Delta\mathcal{U}(k) + I_m u_{k-1} - I_m \bar{u} = \\ &= H_I \Delta\mathcal{U}(k) - \overbrace{I_m (\bar{u} - u_{k-1})}^{c_k} = H_I \Delta\mathcal{U}(k) - c_k \end{aligned} \quad (6.15)$$

By taking (6.14) and (6.15) into consideration (6.13) can be rewritten as

$$J(k) = \frac{1}{2} \|\Theta \Delta\mathcal{U}(k) - \mathcal{E}(k)\|_Q^2 + \frac{1}{2} \|\Delta\mathcal{U}(k)\|_R^2 + \frac{1}{2} \|H_I \Delta\mathcal{U}(k) - c_k\|_W^2 \quad (6.16)$$

what can be further expanded in the following way

$$\begin{aligned} J(k) &= \frac{1}{2} \|\Theta \Delta\mathcal{U}(k) - \mathcal{E}(k)\|_Q^2 + \frac{1}{2} \|\Delta\mathcal{U}(k)\|_R^2 + \frac{1}{2} \|H_I \Delta\mathcal{U}(k) - c_k\|_W^2 = \\ &= \frac{1}{2} [\Delta\mathcal{U}(k)^T \Theta^T - \mathcal{E}(k)^T] Q [\Theta \Delta\mathcal{U}(k) - \mathcal{E}(k)] + \frac{1}{2} \Delta\mathcal{U}(k)^T R \Delta\mathcal{U}(k) + \\ &\quad \frac{1}{2} [\Delta\mathcal{U}(k)^T H_I^T - c_k^T] W [H_I \Delta\mathcal{U}(k) - c_k] = \\ &= \frac{1}{2} \Delta\mathcal{U}(k)^T [\Theta^T Q \Theta + R + H_I^T W H_I] \Delta\mathcal{U}(k) - [\Theta^T Q \mathcal{E}(k) + H_I^T W c_k]^T \Delta\mathcal{U}(k) + \\ &\quad \frac{1}{2} (\mathcal{E}(k)^T Q \mathcal{E}(k) + c_k^T W c_k) \end{aligned}$$

What can be finally written in a form

$$J(k) = \frac{1}{2} \Delta\mathcal{U}(k)^T \mathcal{H} \Delta\mathcal{U}(k) + \mathcal{G}^T \Delta\mathcal{U}(k) + const. \quad (6.17)$$

where

$$\begin{aligned} \mathcal{H}(k) &= \Theta^T Q \Theta + R + H_I^T W H_I \\ \mathcal{G}(k) &= -\Theta^T Q \mathcal{E}(k) - H_I^T W c_k \end{aligned}$$

Problem of minimizing the cost function given by (6.17) while it is subject to constraints (see section 6.3) is called *quadratic programming* (QP). It's solution requires the use of special algorithms what is beyond the scope of this thesis. This subject is discussed in details in [15]. In a case when no constraints are defined however, the values of $\Delta\mathcal{U}(k)$ that minimize (6.17) $\Delta\mathcal{U}(k)_{opt}$ are fairly easy to find by simply taking it's gradient and equating it to zero.

$$\nabla_{\Delta\mathcal{U}(k)} J = \mathcal{H}(k) \Delta\mathcal{U}(k)_{opt} + \mathcal{G}(k) = 0$$

The optimal set of future moves is then given by

$$\Delta\mathcal{U}(k)_{opt} = -\frac{\mathcal{G}(k)}{\mathcal{H}(k)}$$

As explained in the beginning of this chapter, the receding horizon strategy will be used. This implies that only the inputs for the current instant (Δu_k), from the vector of the future inputs $\Delta \mathcal{U}(k)_{opt}$, will be applied to the plant.

6.3 Constraints

The way of including constraints into the problem formulated above will be addressed next.

The constraints that will be implemented in the MPC controller are defined below

- Output constraints

$$z_{min} \leq z_{k+i} \leq z_{max} \quad , \text{ where } i = 1, 2, \dots, P$$

- Input constraints

$$\begin{aligned} u_{min} &\leq u_{k+j} \leq u_{max} \\ \Delta u_{min} &\leq \Delta u_{k+j} \leq \Delta u_{max} \end{aligned} \quad , \text{ where } j = 0, 1, \dots, M-1$$

The quadratic programming problem formulation requires that the constraints are expressed in the form of

$$L\mathcal{X}(k) \leq l \tag{6.18}$$

where L is the constraints matrix, l is the constraints vector and $\mathcal{X}(k)$ the decision vector. Typically $\mathcal{X}(k) = \Delta \mathcal{U}(k)$

The following formulation of hard constraints is based on [13]. The method of softening the output constraints is independent of that work however.

6.3.1 Output constraints

Having output constraints in the form of

$$z_{min} \leq z_{k+i} \leq z_{max} \quad , \text{ where } i = 1, 2, \dots, P \tag{6.19}$$

and stretching them over the whole prediction horizon P we obtain

$$I_p z_{min} \leq \mathcal{Z}(k) \leq I_p z_{max}$$

yielding

$$\begin{bmatrix} -I_z \\ I_z \end{bmatrix} \mathcal{Z}(k) \leq \begin{bmatrix} -\mathcal{Z}_{min} \\ \mathcal{Z}_{max} \end{bmatrix} \quad (6.20)$$

where I_p is defined as in (6.3).

Remembering that the constraints have to be in a form of (6.18) and that $\mathcal{Z}(k)$ is defined as in (6.11) we can write

$$\begin{aligned} \mathcal{Z}(k) = \Psi x_k + \Gamma u_{k-1} + \Theta \Delta \mathcal{U}(k) + \Xi \mathcal{D}(k) + \Phi \delta_m + I_p \gamma = \\ \mathcal{Z}(k)' + \Theta \Delta \mathcal{U}(k) \end{aligned}$$

Substitution into (6.20) yields

$$\begin{bmatrix} -I_z \\ I_z \end{bmatrix} (\mathcal{Z}(k)' + \Theta \Delta \mathcal{U}(k)) \leq \begin{bmatrix} -\mathcal{Z}_{min} \\ \mathcal{Z}_{max} \end{bmatrix} \quad (6.21)$$

$$\underbrace{\begin{bmatrix} -I_z \\ I_z \end{bmatrix} \Theta \Delta \mathcal{U}(k)}_{L_z} \leq - \underbrace{\begin{bmatrix} -I_z \\ I_z \end{bmatrix} \mathcal{Z}(k)'}_{l_z} + \begin{bmatrix} -\mathcal{Z}_{min} \\ \mathcal{Z}_{max} \end{bmatrix} \quad (6.22)$$

6.3.2 Input constraints

Having input constraints in the form of

$$u_{min} \leq u_{k+j} \leq u_{max} \quad , \text{ where } j = 0, 1, \dots, M-1$$

and stretching them over the whole control horizon M we obtain

$$I_m u_{min} \leq \mathcal{U}(k) \leq I_m u_{max}$$

yielding

$$\begin{bmatrix} -I_u \\ I_u \end{bmatrix} \mathcal{U}(k) \leq \begin{bmatrix} -\mathcal{U}_{min} \\ \mathcal{U}_{max} \end{bmatrix} \quad (6.23)$$

where I_m is defined as in (6.9).

Remembering that the constraints have to be in a form of (6.18) and that $\mathcal{U}(k)$ is defined as in (6.9) we can write

$$\mathcal{U}(k) = H_I \Delta \mathcal{U}(k) + I_m u_{k-1}$$

Substitution into (6.23) yields

$$\begin{bmatrix} -I_u \\ I_u \end{bmatrix} (H_I \Delta \mathcal{U}(k) + I_m u_{k-1}) \leq \begin{bmatrix} -\mathcal{U}_{min} \\ \mathcal{U}_{max} \end{bmatrix} \quad (6.24)$$

$$\underbrace{\begin{bmatrix} -I_u \\ I_u \end{bmatrix} H_I \Delta \mathcal{U}(k)}_{L_U} \leq - \underbrace{\begin{bmatrix} -I_u \\ I_u \end{bmatrix} I_m u_{k-1}}_{l_U} + \begin{bmatrix} -\mathcal{U}_{min} \\ \mathcal{U}_{max} \end{bmatrix} \quad (6.25)$$

6.3.3 Input move constraints

Having input move constraints in the form of

$$\Delta u_{min} \leq \Delta u_{k+j} \leq \Delta u_{max} \quad , \text{ where } j = 0, 1 \dots M - 1$$

and stretching them over the whole prediction horizon M we obtain

$$I_m \Delta u_{min} \leq \Delta \mathcal{U}(k) \leq I_m \Delta u_{max}$$

yielding

$$\begin{bmatrix} -I_u \\ I_u \end{bmatrix} \Delta \mathcal{U}(k) \leq \begin{bmatrix} -\Delta \mathcal{U}_{min} \\ \Delta \mathcal{U}_{max} \end{bmatrix} \quad (6.26)$$

where I_m is defined as in (6.9).

6.3.4 Softening of the output constraints

If the constraints are too restrictive a situation may occur where QP problem won't have a solution. It is then said that the problem is *infeasible*. In the case of MPC controlled wind turbine it means that neither generator torque actuator nor pitch actuator will get reference values for their outputs what can result in serious complications. Hence, the issue of ensuring that the problem will be *feasible* is of high importance. A practical way of doing this is to introduce *soft constraints* on one (or more) of the variables. In the case of the wind turbine analysed in this work soft constraints will be introduced on the plants outputs. The result of it will be that MPC will be allowed to violate them in a situation that in other case would render QP problem infeasible. The cost of this happening should be handled in an optimal way what naturally leads to reformulation of not only constraint matrices L and l in (6.18) but also of the cost function (6.13).

Constraints matrices Relaxation of the output constraints is performed by introduction of a *violation margin* ϵ in their formulation. By its inclusion in (6.19) we get

$$z_{min} - \epsilon \leq z_{k+i} \leq z_{max} + \epsilon \quad , \text{ where } \begin{array}{l} i = 1, 2, \dots, P \\ \epsilon \geq 0 \end{array} \quad (6.27)$$

and for the whole prediction horizon P it becomes

$$I_p(z_{min} - \epsilon) \leq \mathcal{Z}(k) \leq I_p(z_{max} + \epsilon)$$

what can be written as

$$\begin{bmatrix} -I_z \\ I_z \end{bmatrix} \mathcal{Z}(k) - \begin{bmatrix} I_p \\ I_p \end{bmatrix} \epsilon \leq \begin{bmatrix} -z_{min} \\ z_{max} \end{bmatrix}$$

In order to be able to optimize the constraints violation the violation margin ϵ has to be included in the decision vector. By doing this and utilizing the results obtained in (6.21) at the same time we get

$$\begin{aligned} \begin{bmatrix} -I_z \\ I_z \end{bmatrix} (\mathcal{Z}(k)' + \Theta \Delta \mathcal{U}(k)) - \begin{bmatrix} I_p \\ I_p \end{bmatrix} \epsilon &\leq \begin{bmatrix} -z_{min} \\ z_{max} \end{bmatrix} \\ \begin{bmatrix} -I_z \\ I_z \end{bmatrix} \Theta \Delta \mathcal{U}(k) - \begin{bmatrix} I_p \\ I_p \end{bmatrix} \epsilon &\leq - \begin{bmatrix} -I_z \\ I_z \end{bmatrix} \mathcal{Z}(k)' + \begin{bmatrix} -z_{min} \\ z_{max} \end{bmatrix} \\ \underbrace{\begin{bmatrix} -I_z \\ I_z \end{bmatrix} \Theta}_{L_{z\epsilon}} \underbrace{- \begin{bmatrix} I_p \\ I_p \end{bmatrix}}_{\mathcal{X}} \underbrace{\begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix}}_{\mathcal{X}} &\leq - \underbrace{\begin{bmatrix} -I_z \\ I_z \end{bmatrix} \mathcal{Z}(k)'}_{l_{z\epsilon}} + \begin{bmatrix} -z_{min} \\ z_{max} \end{bmatrix} \end{aligned} \quad (6.28)$$

Violation margin ϵ won't have any influence on the constraints that concern inputs u_{k+i} and their moves Δu_{k+i} . From (6.24) we can then write

$$\underbrace{\begin{bmatrix} -I_u \\ I_u \end{bmatrix} H_I}_{L_{U\epsilon}} \underbrace{0}_{l_{U\epsilon}} \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix} \leq - \begin{bmatrix} -I_u \\ I_u \end{bmatrix} I_m u_{k-1} + \begin{bmatrix} -u_{min} \\ u_{max} \end{bmatrix} \quad (6.29)$$

and from (6.26)

$$\underbrace{\begin{bmatrix} -I_u \\ I_u \end{bmatrix}}_{L_{\Delta U\epsilon}} \underbrace{0}_{l_{\Delta U\epsilon}} \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix} \leq \begin{bmatrix} -\Delta u_{min} \\ \Delta u_{max} \end{bmatrix} \quad (6.30)$$

Additionally from (6.27) we know that $\epsilon \geq 0 \implies -\epsilon \leq 0$ what can be stated in a form of (6.18) as

$$\underbrace{\begin{bmatrix} 0 \\ -I_z \end{bmatrix}}_{L_\epsilon} \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{l_\epsilon} \quad (6.31)$$

Cost function In order to get an optimal solution of the QP problem with soft constraints cost function has to be build in a way that includes a factor that is referring to the importance of the constraints violation margin ϵ in relation to other performance indexes. It can be done by expanding (6.16) to yield

$$J(k) = \frac{1}{2} \|\Theta \Delta \mathcal{U}(k) - \mathcal{E}(k)\|_Q^2 + \frac{1}{2} \|\Delta \mathcal{U}(k)\|_R^2 + \frac{1}{2} \|H_I \Delta \mathcal{U}(k) - c_k\|_W^2 + \frac{1}{2} \|\epsilon\|_S^2 \quad (6.32)$$

6.3.5 Definition of a QP problem with soft constraints

By gathering together the results of a few last sections a quadratic programming problem with soft constraints can be defined. It's solution will yield the control reference signals for both blade pitch and generator torque actuators.

To rewrite (6.32) in a form of (6.17) with $\mathcal{X}_E = \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix}$ being the decision vector we can write

$$J(k) = \frac{1}{2} \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix}^T \underbrace{\begin{bmatrix} \mathcal{H} & 0 \\ 0 & S \end{bmatrix}}_{\mathcal{H}_E} \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix} + \underbrace{\begin{bmatrix} \mathcal{G} \\ 0 \end{bmatrix}}_{\mathcal{G}_E}^T \begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix} + const. \quad (6.33)$$

Joining (6.28), (6.29), (6.30) and (6.31) together yields constraints definition in the form required by a softly constrained QP problem. Namely,

$$\underbrace{\begin{bmatrix} \begin{bmatrix} -I_z \\ I_z \end{bmatrix} \Theta & - \begin{bmatrix} I_p \\ I_p \end{bmatrix} \\ \begin{bmatrix} -I_u \\ I_u \end{bmatrix} H_I & 0 \\ \begin{bmatrix} -I_u \\ I_u \end{bmatrix} & 0 \\ 0 & -I_z \end{bmatrix}}_{L_E} \underbrace{\begin{bmatrix} \Delta \mathcal{U}(k) \\ \epsilon \end{bmatrix}}_{\mathcal{X}_E} \leq \underbrace{\begin{bmatrix} - \begin{bmatrix} -I_z \\ I_z \end{bmatrix} \mathcal{Z}(k)' + \begin{bmatrix} -\mathcal{Z}_{min} \\ \mathcal{Z}_{max} \end{bmatrix} \\ - \begin{bmatrix} -I_u \\ I_u \end{bmatrix} I_m u_{k-1} + \begin{bmatrix} -\mathcal{U}_{min} \\ \mathcal{U}_{max} \end{bmatrix} \\ \begin{bmatrix} -\Delta \mathcal{U}_{min} \\ \Delta \mathcal{U}_{max} \end{bmatrix} \\ 0 \end{bmatrix}}_{l_E}$$

Finally the QP problem with soft constraints can be defined as

$$J(k) = \frac{1}{2} \mathcal{X}_E^T \mathcal{H}_E \mathcal{X}_E + \mathcal{G}_E^T \mathcal{X}_E + const. \quad (6.34)$$

subject to the constraints in the form of

$$L_E \mathcal{X} \leq l_E \quad (6.35)$$

6.4 Chapter summary

In this chapter the concept of Model-Based Predictive Control has been introduced. Both cost function and special form of constraints representation have been discussed. Finally the output constraints have been softened and joined together with the QP programming form of the cost function yielding a Quadratic Programming problem with soft constraints that is solved by the MPC controller.

In the next chapter its tuning, together with the strategy of controlling the plant in the whole spectrum of wind speeds, will be discussed.

Gain and weights scheduling

In order for the MPC controller, that have been presented in the previous chapter, to work in all of the operation modes a gain scheduling strategy will be implemented. It will be introduced here. Next, weight scheduling approach, that will allow for further increase in control performance, will be discussed.

7.1 Gain scheduling

As mentioned before the wind turbine that is the focus of this thesis will be controlled in all of its operation modes all the way from low region, through mid, high and finishing on the top one. The properties of the plant are different between regions and they change together with the change in the wind speed v . Having a different model for every wind speed would be an optimal solution that would minimize errors coming from imprecise modelling. In practice it is carried out by employing so called *gain scheduling* that assumes obtaining a finite number of models for different values of *scheduling variable* (wind speed v in this case) and switching between them when it changes. In this thesis one of the goals is to design a control system that would allow to shift between control regions in a smooth way. This means that the transition between models should also be smooth. This is accomplished by element wise interpolation between

two models closest to the desired wind speed value. Both the model for the MPC controller (see (6.1)) and for the kalman filter (see (5.1)) are subject to this procedure.

The interpolation is carried out by calculating a *relevance coefficient* α which is the measure of where does the given wind speed v_0 fall between two other wind speeds that have a real model associated with them. It is shown on figure 7.1 where v_H is the wind speed that is higher than given v_0 and v_L is the lower one. Next the elements of the matrices of the interpolated system are computed.

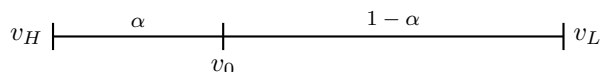


Figure 7.1: The idea behind the relevance coefficient α .

Each of those elements $element_0$ is given by

$$element_0 = element_H \times (1 - \alpha) + element_L \times \alpha$$

where $element_H$ and $element_L$ are the elements of the system matrices of the model associated with the higher and lower wind speed respectively.

It should be remembered that interpolating with a cubic method would most probably render better results, due to it's smoothness, but for the sake of simplicity it wasn't implemented here. It is important to note that only the matrices directly connected to the model are the subject of interpolation. Prediction matrices needed to construct the cost function will not be interpolated in the approach taken in this thesis. Instead they will be calculated afterwards from the interpolated (new) model.

7.2 Weight scheduling

Together with plants model it's reaction to weights is changing together with wind speed as well. Furthermore in the mid region, where one of controller's objectives is to keep the wind turbine's power coefficient C_p at it's maximum, linear model of the plant is accurate only in the close vicinity of the linearization point what forces a different control approach than in the top region. This is illustrated on Fig 7.2 where a C_p curve for a fixed value of tip speed ratio λ is shown. The straight line is representing linearized relation between C_p coefficient and the pitch of the blades θ . It is important to note that since the aim of the operation in the mid region is to maximize the power production the controller should ensure that wind turbine is working close to, and ideally at, the top of the C_p curve what translates to very low sensitivity to change in the

pitch angle ($\left. \frac{\partial C_p}{\partial \theta} \right|_{\theta_0} \approx 0$). This is true only very locally. However, linear model

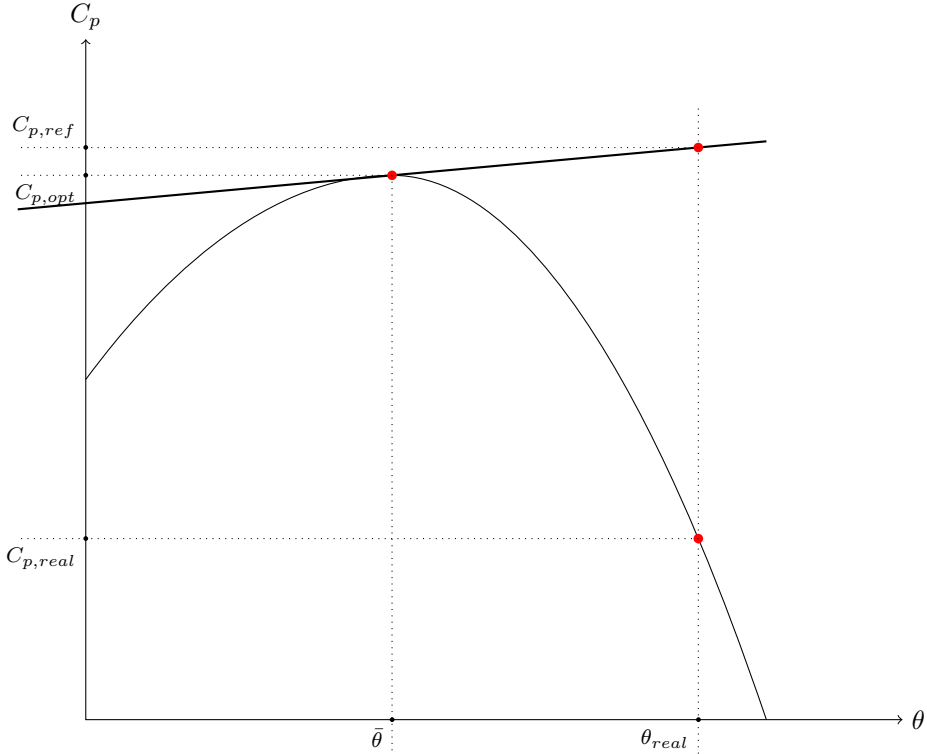


Figure 7.2: Illustration of the problem with controlling a HAWT with blade pitch angle change in the mid region - C_p curve for a fixed value of tip speed ratio λ

extends this dependence onto the whole spectrum of input and output values. It can be observed that power P_e (which is proportional to C_p coefficient) reference tracking in this case might result in a dangerous behaviour of the blade pitch actuator. In a scenario where an increase in power P_e output is expected controller might try to achieve this by raising the value of the C_p coefficient to a new value $C_{p,ref}$ by setting the pitch θ to a new value θ_{real} that would be far higher from its linearization point $\bar{\theta}$. In a linear case this action would cause the power to rise. However in reality, power coefficient C_p will drop to a value far lower than expected $C_{p,real}$ what can be interpreted by the controller as a need for even stronger control action that would lower the power output P_e even further what would result in an unstable behaviour in case no constraints are implemented.

This issue is the main motivation for introduction of *weight scheduling* strategy

that would involve not only the MPC controller weights (Q , R , W , and S) but also the ones connected with the kalman filter. It is used mainly to shape the cost function (by weights adjustment) in a way that would guarantee that for certain wind speeds some performance indexes would be penalized very strongly (their violation would be very costly) in respect to others.

It can be implemented in the same way as gain scheduling, i.e. by linear interpolation (or cubic for better results) with respect to wind speed between tuning parameters prepared beforehand.

7.2.1 Scheduling guidelines

Below guidelines for tuning of the MPC controller weights for each operation mode will be discussed. Kalman filter must be tuned beforehand, but only roughly. After obtaining suboptimal control action with weight scheduling of the MPC parameters it should be fine-tuned in order to minimize the estimation errors and optimize their dynamics.

7.2.1.1 Mid region

In this work the issue with turbine's control in the mid region is addressed by:

- Ensuring that the pitch of the blades will be held at it's linearization point $\bar{\theta}$ which, at the same time, is the optimal value θ_{opt} (value giving the highest C_p coefficient value when tip speed ratio λ is at it's optimum as well)

This is done by:

- setting part of the W (see (6.13)) weight that is linked to the blade pitch very high.

- Instead of tracking both maximal available electric power P_e and rotor rotational speed Ω_r the focus is put on tracking only Ω_r . The reference value $\Omega_{r,opt}$ is chosen to be one giving optimal value of tip speed ratio λ_{opt} (see (2.1)) which together with optimal blade pitch θ_{opt} maximizes the value of power coefficient C_p , what is equivalent to maximizing output power P_e .

This is done by:

- setting part of the Q (see (6.13)) weight that is linked to the generated power P_e close to zero.

- setting part of the Q (see (6.13)) weight that is linked to rotor speed Ω_r much higher than the one associated with P_e .

7.2.1.2 Low and high region

Tuning in the low region should be carried out in a similar fashion as in the mid one. The only difference is that the reference rotor speed Ω_r will be fixed at the lower limit $\Omega_{r,min}$ what might require some minor changes in the weight balance.

The same situation takes place in the high region where rotor speed Ω_r reaches the opposite limit $\Omega_{r,max}$. However here, the tuning parameters should be chosen in a way that would help smoothing out the transition between tuning strategy of minimizing the use of the blade pitch as a control signal and tracking only rotor's rotational speed (mid region), and strategy of prioritizing pitch as the active control signal and balancing the importance of Ω_r and P_e as the system's outputs (top region).

7.2.1.3 Top region

The tuning strategy here is different from the one in low through high region. Collective blade pitch is the main control signal in this operation but unlike in the mid region mismatch between linear model and non-linear wind turbine is not that troublesome so there is no need of suppressing the other (control signal) one. Furthermore output tracking cannot involve only rotor's speed Ω_r and power output P_e has to be also taken into consideration here.

This is done by:

- setting W weight for both blade pitch and generator torque to zero.
- setting Q weight in a way that would more or less balance the importance of both rotor speed Ω_r and power output P_e tracking.

7.3 Chapter summary

Gain and weight scheduling has been introduced in this chapter. Also guidelines for tuning the MPC controller has been given. Those methods will have a substantial influence on it's behaviour. In the next part simulations illustrating that fact will be carried out.

Part III

Implementation and simulation

Wind Turbine type

8.1 Wind turbine

The HAWT type that is being considered in this work is a wind turbine model widely known as "NREL offshore 5-MW baseline wind turbine". It was developed by National Renewable Energy Laboratory (NREL) of United States of America. It's detailed definition is based on design informations obtained from various wind turbine manufactures in order to establish a specification that would be representative for a utility-scale multimegawatt wind turbine suitable for deployment either on land or in shallow or deep water. Table 8.1 lists the parameters that are the most significant with respect to this thesis. It's detailed specification is presented in [11].

8.2 Wind turbine actuators

The generator torque and collective blade pitch actuators weren't defined in wind turbine's specification [11]. As it was mentioned in section 3 they are modelled as first and second order system respectively following the guidelines in [14]. In [6] the same models are used and parameters proposed by the author

NREL offshore 5-MW baseline wind turbine			
parameter	symbol	units	value
nominal power	$P_{e,nom}$	[kW]	5000
rated rotor speed	$\Omega_{r,max}$	[rpm]	12.1
cut-in rotor speed	$\Omega_{r,min}$	[rpm]	6.9
drivetrain spring const.	K_s	[N · m/rad]	867.637×10^6
drivetrain dampening const.	D_s	[N · m/(rad/s)]	6.215×10^6
generator inertia (w.r.t. HSS ¹)	J_g	[kg · m ²]	534.116
rotor inertia (w.r.t LSS ²)	J_r	[kg · m ²]	3.8768×10^7
total inertia (w.r.t LSS ²)	J_t	[kg · m ²]	4.3792×10^7
blade radius	R	[m]	63
number of blades	-	[-]	3
gear ratio	N_g	[-]	97
generator efficiency ³	η	[-]	1
maximum blade pitch	θ_{max}	[deg]	90
minimum blade pitch ³	θ_{min}	[deg]	-1
maximum blade pitch rate	$\Delta\theta_{max}$	[deg/s]	8
minimum blade pitch rate	$\Delta\theta_{min}$	[deg/s]	-8
maximum generator torque	$Q_{g,max}$	[N · m]	47402.97
minimum generator torque	$Q_{g,min}$	[N · m]	0
maximum generator torque rate	$\Delta Q_{g,max}$	[N · m/s]	15000
minimum generator torque	$\Delta Q_{g,min}$	[N · m/s]	-15000
control approach:	variable blade-pitch-to-feather-controlled, variable speed generator		

¹with respect to high speed shaft²with respect to low speed shaft³changed with respect to original NREL's definition

Table 8.1: NREL offshore 5-MW baseline wind turbine specification

Parameters of torque and pitch actuators			
parameter	symbol	units	value
time const. of the gen. torque. act.	τ_g	[s]	0.1
natural frequency of the blade pitch act.	ω_n	[rad/s]	0.88
damping of the blade pitch act.	ζ	[-]	0.9

Table 8.2: Generator torque and collective blade pitch actuator parameters.

are used here. These are: Figures 8.1 and 8.2 show the step response of each of the actuators with the given parameters .

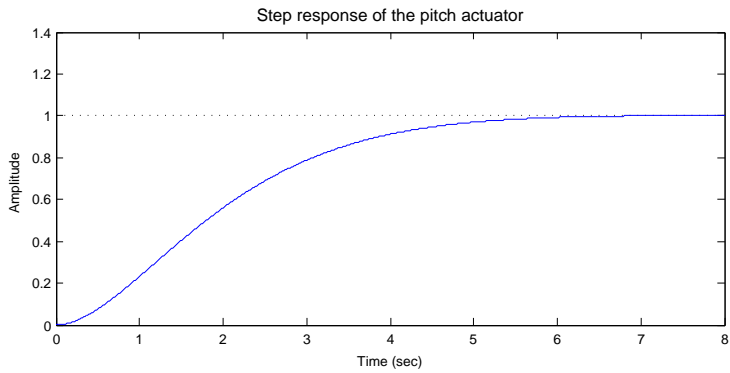


Figure 8.1: Step response of the blade pitch actuator.

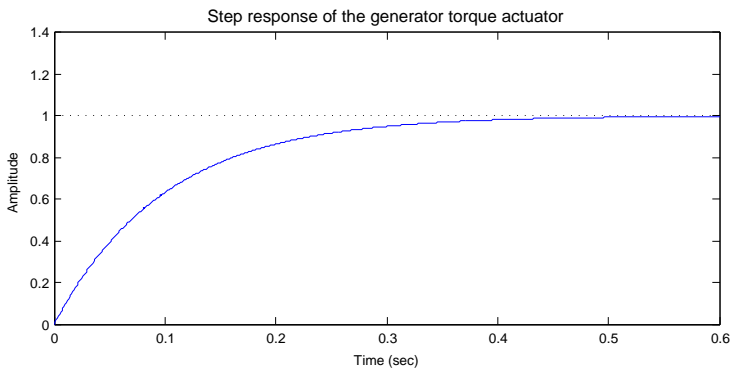


Figure 8.2: Step response of the generator torque actuator.

8.3 Chapter summary

This chapter presented the wind turbine type that this thesis is focused on. It's main parameters has been listed and it's actuators has been has been shortly analysed.

In the next chapter the models representing those systems are tested and their implementation is investigated.

Model analysis

Based on the guidelines described in the previous chapters one model for each wind speed has been created. Only one model from each operation mode will be analysed here. Namely, models associated with the wind speeds of 18, 11, 8, 4 m/s . The focus will be put only on the wind turbine dynamics. Neither wind speed v , nor integrating unmeasured disturbances d and p will be considered here. Their implementation will be presented in later sections. The model will be of 6th order and its state vector will contain: rotor's rotational speed Ω_r , generator's rotational speed Ω_g , torsion of the drivetrain ϕ_Δ , generator torque Q_g , collective blade pitch θ and its first derivative $\dot{\theta}$ (see (9.1)).

The notation used here is the same as in part I.

$$\Delta \dot{x}' = \begin{bmatrix} \Delta \dot{x}_{dt} \\ \Delta \dot{x}_{Q_g} \\ \Delta \dot{x}_\theta \end{bmatrix} = \overbrace{\begin{bmatrix} A_{dt} & B_{dt_1} & B_{dt_2} & 0 \\ 0 & A_{Q_g} & 0 & \\ 0 & 0 & A_\theta & \end{bmatrix}}^{A'_c} \begin{bmatrix} \Delta x_{dt} \\ \Delta x_{Q_g} \\ \Delta x_\theta \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 \\ B_{Q_g} & 0 \\ 0 & B_\theta \end{bmatrix}}^{B'_c} \begin{bmatrix} \Delta Q_g \\ \Delta \theta \end{bmatrix} + \overbrace{\begin{bmatrix} B_{vdt} \\ 0 \\ 0 \end{bmatrix}}^{B'_{v,c}} \Delta v \quad (9.1a)$$

$$\Delta y' = \begin{bmatrix} P_e \\ \Omega_r \end{bmatrix} = \underbrace{\begin{bmatrix} C_{dt} & [D_{dt} & 0] \end{bmatrix}}_{C'_c} \begin{bmatrix} \Delta x_{dt} \\ \Delta x_{Q_g} \\ \Delta x_\theta \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{D'_c} \begin{bmatrix} \Delta Q_g \\ \Delta \theta \end{bmatrix} \quad (9.1b)$$

9.1 System matrices

Below the system matrices of the models obtained for the above mentioned wind speeds v are presented.

9.1.1 Model linearized at 4 m/s

$$A'_c = \begin{bmatrix} -0.18078 & 0.0016528 & -22.381 & 0 & 0.029586 & 0 \\ 119.96 & -1.2367 & 16747 & -0.0018723 & 0 & 0 \\ 1 & -0.010309 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.77440 & -1.5840 \end{bmatrix}$$

$$B'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 0 \\ 0 & 0.77440 \end{bmatrix} \quad B'_{v,c} = \begin{bmatrix} 0.00702 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C'_c = \begin{bmatrix} 0 & 2.6421 & 0 & 0.070082 & 0 & 0 \\ 9.5493 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (9.2)$$

9.1.2 Model linearized at 8 m/s

$$A'_c = \begin{bmatrix} -0.21969 & 0.0016528 & -22.381 & 0 & 0.069020 & 0 \\ 119.96 & -1.2367 & 16747 & -0.0018723 & 0 & 0 \\ 1 & -0.010309 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.77440 & -1.5840 \end{bmatrix}$$

$$B'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 0 \\ 0 & 0.77440 \end{bmatrix} \quad B'_{v,c} = \begin{bmatrix} 0.01937 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (9.3)$$

$$C'_c = \begin{bmatrix} 0 & 18.836 & 0 & 0.099231 & 0 & 0 \\ 9.5493 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (9.4)$$

9.1.3 Model linearized at 11 m/s

$$A'_c = \begin{bmatrix} -0.23043 & 0.0016528 & -22.381 & 0 & 0.066682 & 0 \\ 119.96 & -1.2367 & 16747 & -0.0018723 & 0 & 0 \\ 1 & -0.010309 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.77440 & -1.5840 \end{bmatrix}$$

$$B'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 0 \\ 0 & 0.77440 \end{bmatrix} \quad B'_{v,c} = \begin{bmatrix} 0.02644 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (9.5)$$

$$C'_c = \begin{bmatrix} 0 & 40.011 & 0 & 0.12290 & 0 & 0 \\ 9.5493 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (9.6)$$

9.1.4 Model linearized at 18 m/s

$$A'_c = \begin{bmatrix} -0.44216 & 0.0016528 & -22.381 & 0 & -1.3589 & 0 \\ 119.96 & -1.2367 & 16747 & -0.0018723 & 0 & 0 \\ 1 & -0.010309 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.77440 & -1.5840 \end{bmatrix}$$

$$B'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 0 \\ 0 & 0.77440 \end{bmatrix} \quad B'_{v,c} = \begin{bmatrix} 0.03111 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C'_c = \begin{bmatrix} 0 & 40.683 & 0 & 0.12290 & 0 & 0 \\ 9.5493 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D'_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (9.7)$$

It can be noticed that only four values in the above matrices are changing. They are the parameters that have been obtained with the help of FAST linearization tool (see 3.6). Namely,

$$\frac{1}{J_r} \frac{\partial Q_r}{\partial \Omega_r} \Big|_{\bar{\Omega}_r}, \quad \frac{1}{J_r} \frac{\partial Q_r}{\partial \theta} \Big|_{\bar{\theta}}, \quad \frac{1}{J_r} \frac{\partial Q_r}{\partial v} \Big|_{\bar{v}}, \quad \bar{Q}_g, \quad \bar{\Omega}_g$$

Figure 9.1 shows how they change with the wind speed v .

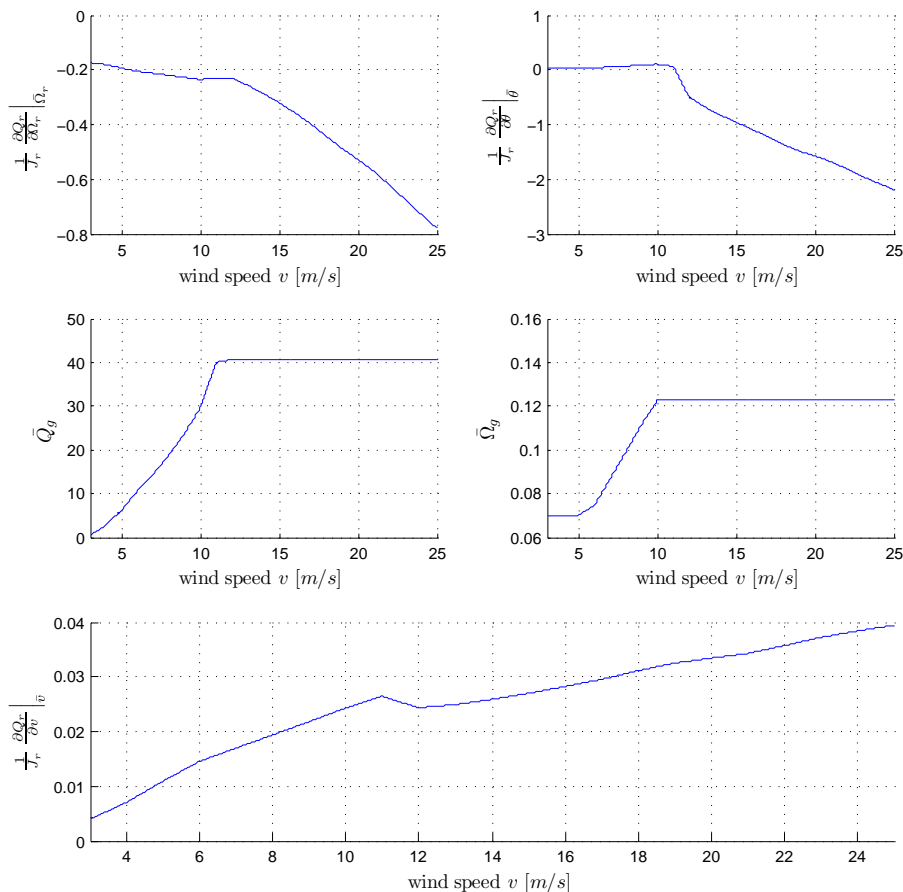


Figure 9.1: The way in which wind turbine model's parameters are changing with respect to the wind speed v .

9.2 Eigenvalues

In table 9.1 the eigenvalues s_i for the above mentioned models are presented. As it can be seen all of the eigenvalues are negative what is indicating that

s_i	Eigenvalues of models linearized at...			
	... 4 [m/s]	... 8 [m/s]	... 11 [m/s]	... 18 [m/s]
s_1	-0.01811	-0.05256	-0.0621	-0.24951
s_2	-0.69968 + 13.948i	-0.70191 + 13.948i	-0.70252 + 13.948i	-0.71467 + 13.947i
s_3	-0.69968 - 13.948i	-0.70191 - 13.948i	-0.70252 - 13.948i	-0.71467 - 13.947i
s_4	-0.792 + 0.38358i	-0.792 + 0.38358i	-0.792 + 0.38358i	-0.792 + 0.38358i
s_5	-0.792 - 0.38358i	-0.792 - 0.38358i	-0.792 - 0.38358i	-0.792 - 0.38358i
s_6	-10	-10	-10	-10

Table 9.1: Eigenvalues of models linearized at different operation regions.

the model of the wind turbine that this thesis is focused on is inherently stable. The sampling period T_s is chosen to be two times smaller than the inverse of the eigenvalue with the highest real part, what corresponds to

$$T_s = \frac{1}{2\zeta\omega_n} \quad (9.8)$$

In this case the eigenvalue with the highest real part is $s_6 = -10$. This gives the sampling period of $T_s = 0.05$.

Judging from the fact that only few parameters were changing in the system matrices, as it was shown in the previous section, it could be predicted that only some eigenvalues will be changing with the wind speed. In fact only the first one do it in a significant way. It's trajectory is shown on the figure 9.2.

9.3 Observability and controllability

Controllability and observability of a dynamic system is discussed in details in [5]. Here only the basic theorems will be utilized.

It is known that the given system is observable in the time interval spanning from the current instant to infinity if and only if it's observability gramian W_o

$$W_o = \int_0^{\infty} e^{A^T \tau} C^T C e^{A \tau} d\tau \quad (9.9)$$

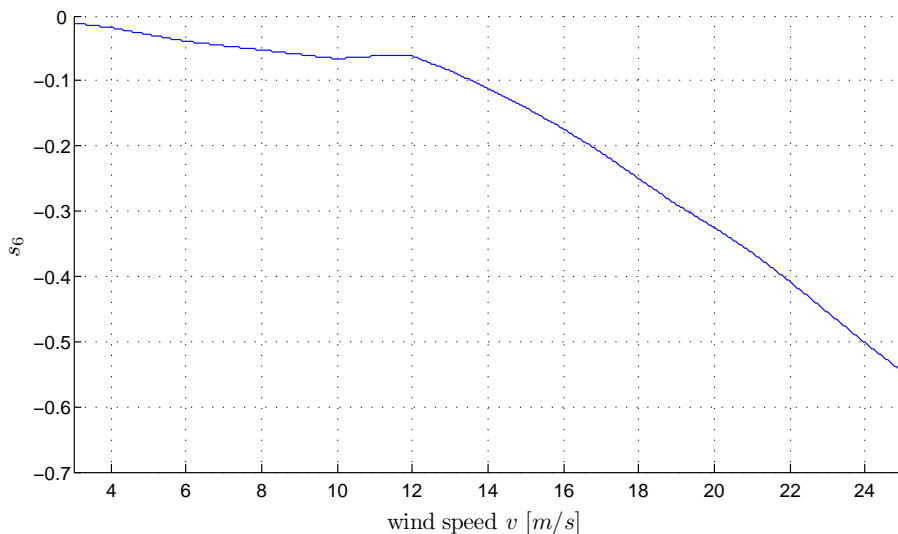


Figure 9.2: The way in which eigenvalue s_6 changes with the wind speed.

is of full rank. It is also known that it is controllable in the same time interval if and only if it's controllability gramian W_c

$$W_c = \int_0^{\infty} e^{A\tau} B B^T e^{A^T \tau} d\tau \quad (9.10)$$

is of full rank as well.

The maximum rank that both W_o and W_c can have is equal to the one of the A'_c matrix. Hence the given system is controllable and/or observable if it's respective gramian is of 6th rank.

Every model that is being analysed has been tested for both observability and controllability with the above tests. It has been found that both observability and controllability gramians for every model checked were of 6th rank. This leads to a conclusion that all of them are both observable and controllable.

9.4 Simulations

The below simulations are showing the reaction of the given models to the step changes of the control signal references for the actuators. Each of them is changing by a factor of 10% with respect to the linearization point of the given model. They were carried out with the use of FAST as the simulator. The degrees of freedom (DOF), available in FAST, for the simulated wind turbine together with their state during the simulation are presented in table 9.2. The

Degrees of freedom available in FAST		
name	state	parameter name
First flapwise blade mode	on	FlapDOF1
Second flapwise blade mode	on	FlapDOF2
First edgewise blade mode	on	EdgeDOF
Rotor-teeter	off	TeetDOF
Drivetrain rotational-flexibility	on	DrTrDOF
Generator	on	GenDOF
Yaw	off	YawDOF
First fore-aft tower bending-mode	on	TwFADOF1
Second fore-aft tower bending-mode	on	TwFADOF2
First side-to-side tower bending-mode	on	TwSSDOF1
Second side-to-side tower bending-mode	on	TwSSDOF2

Table 9.2: Degrees of freedom of a HAWT available in FAST and their use in model simulation.

results of the simulation are presented on figures 9.3 through 9.6. They show the values of input references and outputs of both FAST and the models tested.

It can be noticed that there is a steady state error between FAST and the model linearised for wind speeds falling into mid and high regions. In order to check if the source of the error lies in the mixed modelling technique described in section I a model, whose core (generator and drivetrain - without actuators) had been created entirely with the use of FAST, was used for comparison. The results of the simulation are shown on figure 9.7.

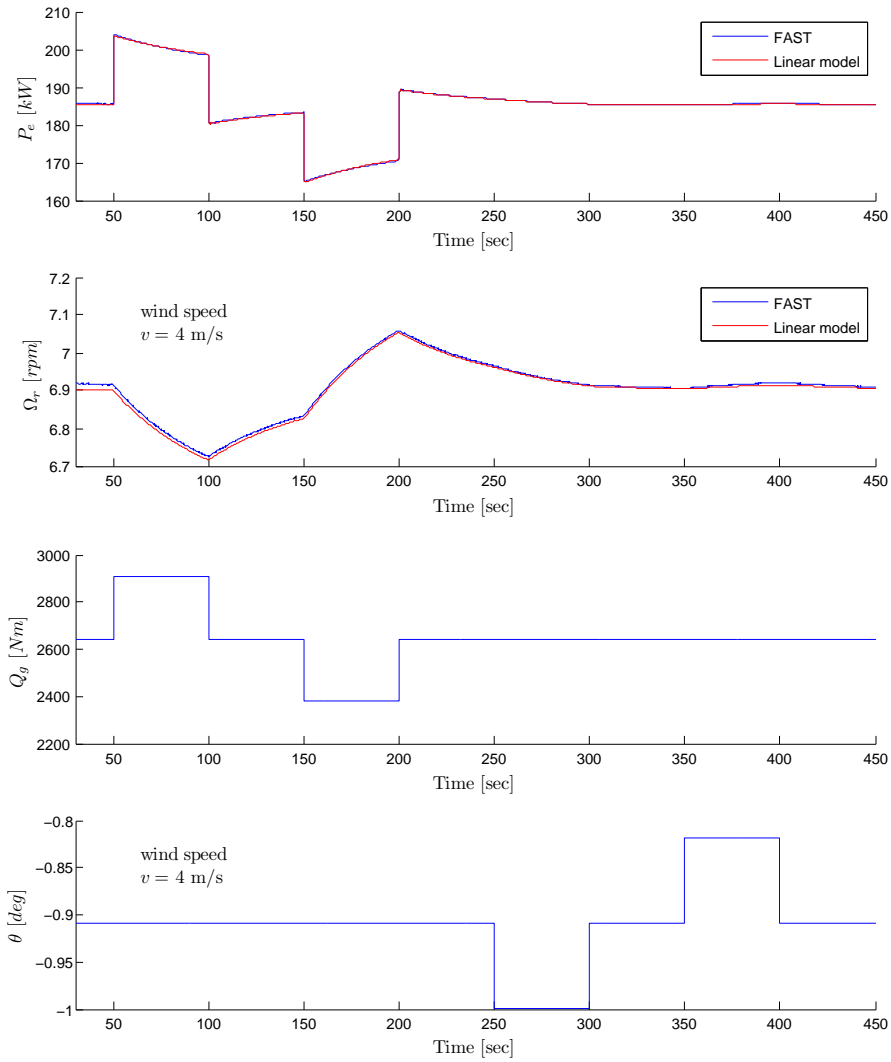
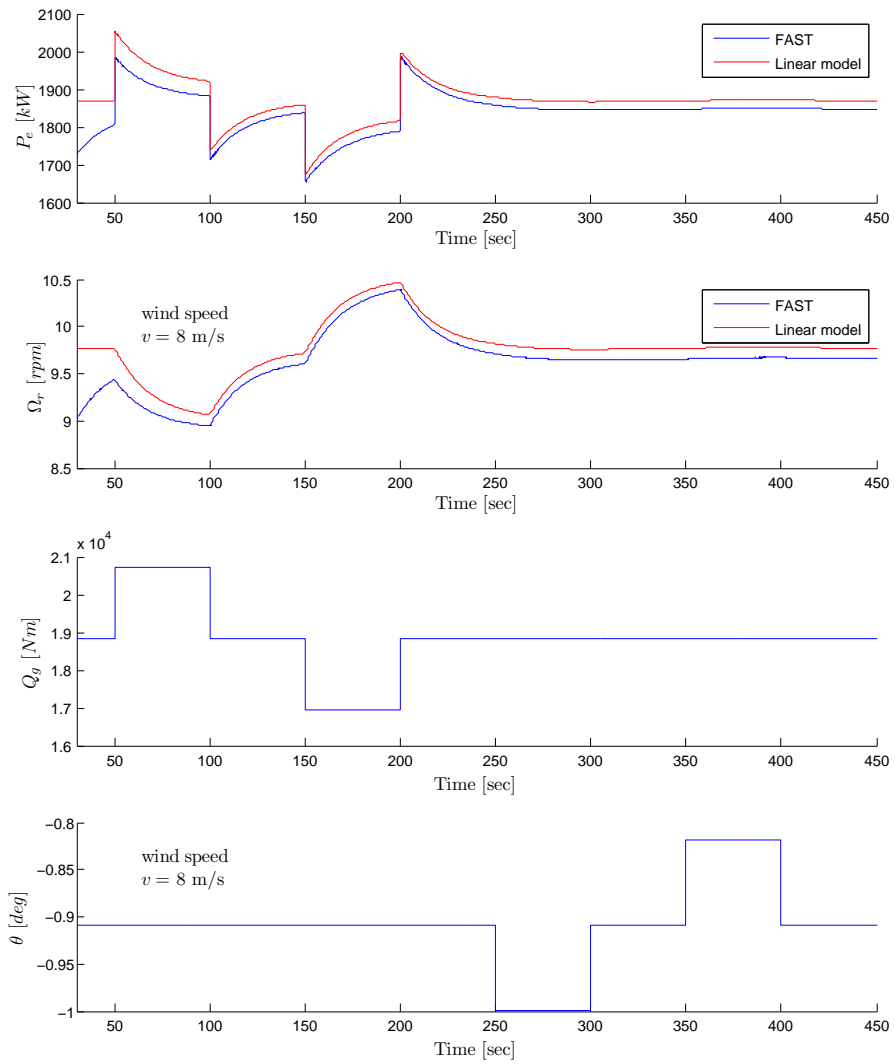


Figure 9.3: Linear model simulation for the wind speed v of 4 m/s.

Figure 9.4: Linear model simulation for the wind speed v of 8 m/s .

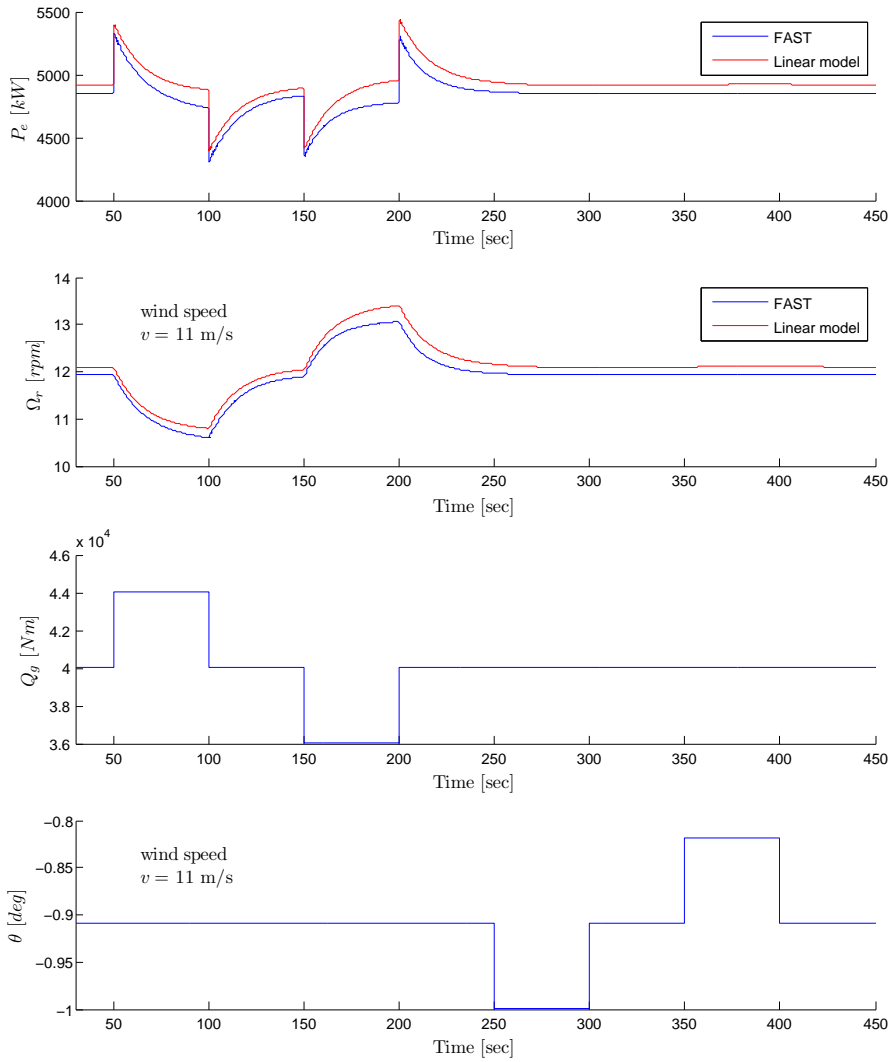


Figure 9.5: Linear model simulation for the wind speed v of 11 m/s.

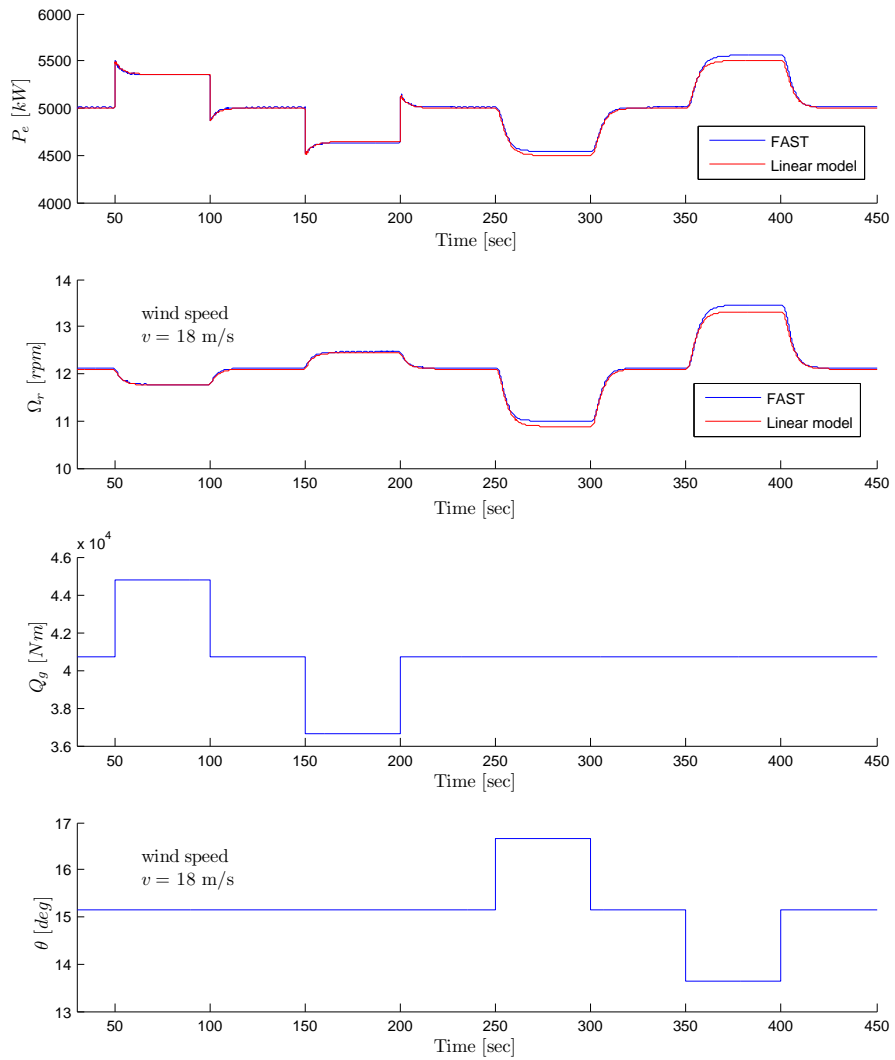


Figure 9.6: Linear model simulation for the wind speed v of 18 m/s .

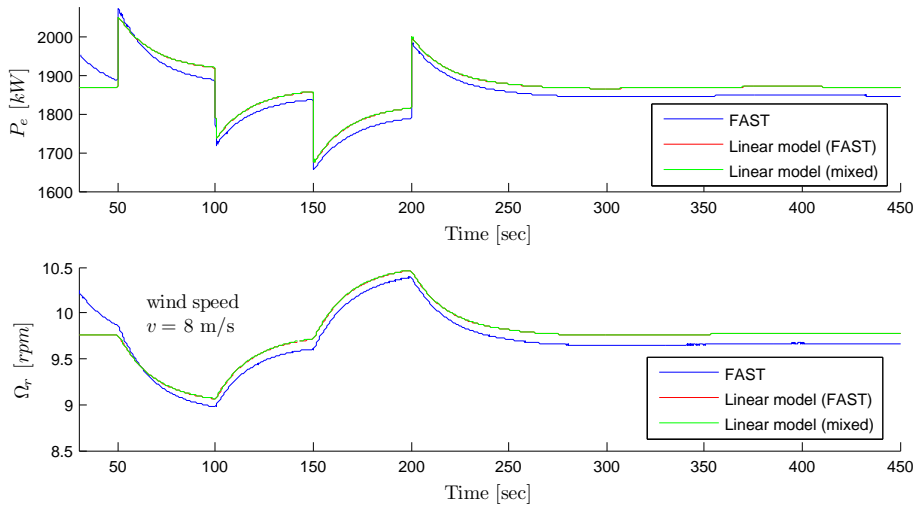


Figure 9.7: Comparison of the responses of FAST, with 9 degrees of freedom (DoF) enabled, and both mixed (3rd order) and FAST computed (4th order) models for the wind speed v of 8 m/s .

9.4.1 Simulation results discussion

With respect to figure 9.7 It can be noted that the tested model is essentially the same in response to the given control signals as the one obtained straight from FAST. They both have an identical offset with respect to FAST's non-linear representation of the wind turbine. Furthermore, for top and low operation regions the error is negligible and the only elements of the model matrices that are changing from one wind speed to another are the parameters calculated by the FAST linearization tool.

Taking the above into consideration it can be concluded that the relatively big modelling mismatch is caused by wrong or imprecise computation of the parameters that were calculated with the use of FAST linearization tool.

9.5 Chapter summary

The chapter presented the aspects linked to the implementations of the model of the wind turbine that this thesis is focused on. Next it's properties were investigated and the model was simulated in the end. The conclusions are that the model (set of models) is inherently stable, observable and controllable and

that there is a noticeable modelling mismatch for the models laying in the mid and high regions. This error is most probably caused by the FAST linearization tool.

In the next chapter the implementation of the MPC controller based on the model tested here will be analysed.

CHAPTER 10

MPC

In this section the implementation of the Model Predictive Controller controlling a FAST simulated wind turbine will be discussed. Topic relevant issues like steady state error rejection and choice of weights will also be addressed here. As it was mentioned in section II two versions of one model are used: one for kalman filter, where all considered states are present, and one for MPC controller where models of the integrating disturbances and wind are not included (during the simulation their evolution is evaluated independently).

10.1 Offset free control

As it was discussed in section 5, in order to be able to obtain offset free control in the steady state the system has to be augmented with the model of unmeasured integrating disturbances and then attempt to estimate them with the use of kalman filter. This requires it (the system) to be observable. A condition for this to be true is (as mentioned in section (5.2)) that the pair of matrices (C_d , A_d) of the original system is observable and that

$$\begin{bmatrix} A_d - I & B_{dp} \\ C_d & D_{dp} \end{bmatrix} \text{ has a full column rank}$$

Test for the observability of the continuous time model representing the wind turbine systems has been already performed in section 9.3 but here it's discrete representation augmented with the wind model is being analysed (see (II.0)).

The B_{dp} , D_{dp} matrices that guarantee that the above condition will be met for all wind speeds are

$$B_{dp} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad D_{dp} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

10.2 Tuning

As was mentioned in section 7.2 one of the means by which smooth transition between operation regions will be achieved is the weight scheduling technique which calls for choosing weights for a number of predefined values of wind speed, putting them into an array and interpolating between them when the wind speed changes. Example values of those weights are presented in tables 10.1 and 10.2. Additionally prediction horizon P and control horizon M are set to

$$P = 5sec \quad M = 2sec \quad (10.1)$$

Those values give acceptable response and don't pose overly big computational burden for the processing unit.

MPC const function weigths				
wind speed	Q	R	W	S
1	[1e-3 1e2]	[1e3 1e4]	[0 1e4]	[1e8 1e8]
2	[1e-3 1e2]	[1e3 1e4]	[0 1e4]	[1e8 1e8]
3	[1e-8 1e4]	[1e5 1e5]	[0 1e10]	[1e8 1e8]
4	[1e-8 1e4]	[1e5 1e5]	[0 1e10]	[1e8 1e8]
5	[1e-8 1e4]	[1e6 1e5]	[0 1e10]	[1e8 1e8]
6	[1e-8 1e3]	[1e5 1e3]	[0 1e10]	[1e8 1e8]
7	[1e-8 1e3]	[1e5 1e3]	[0 1e10]	[1e8 1e8]
8	[1e-8 1e3]	[1e5 1e3]	[0 1e10]	[1e8 1e8]
9	[1e-8 1e3]	[1e5 1e5]	[0 1e10]	[1e8 1e8]
10	[1e-8 1e3]	[1e5 1e5]	[0 1e10]	[1e8 1e8]
11	[1e-8 1e4]	[1e5 1e5]	[0 1e10]	[1e8 1e8]
12	[1e-4 1e3]	[3e4 1e4]	[0 1e0]	[1e8 1e8]
13	[1e-3 1e3]	[1e3 1e5]	[0 0]	[1e8 1e8]
14	[1e-3 1e3]	[1e3 1e5]	[0 0]	[1e8 1e8]
15	[1e-3 1e3]	[1e3 1e5]	[0 0]	[1e8 1e8]
16	[1e-3 1e3]	[1e3 1e5]	[0 0]	[1e8 1e8]
17	[1e-3 1e3]	[1e3 1e5]	[0 0]	[1e8 1e8]
18	[1e-3 1e3]	[1e3 1e5]	[0 0]	[1e8 1e8]
19	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]
20	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]
21	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]
22	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]
23	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]
24	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]
25	[1e-3 1e3]	[1e3 1e4]	[0 0]	[1e8 1e8]

Table 10.1: Tuning parameters for the MPC cost function with respect to the wind speed.

Kalman tunings										
wind speed	Ω_r	Ω_r	ϕ_Δ	Q_g	θ	$\dot{\theta}$	v	\dot{v}	d	p
1	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
2	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
3	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
4	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
5	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
6	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
7	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
8	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
9	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
10	1e-3	1e-4	1e-4	1e-2	1e-2	1e-3	1e-6	1e-6	1e-8	1e-8
11	1e-3	1e-4	1e-4	1e-4	1e-4	1e-3	1e-6	1e-6	1e-8	1e-8
12	1e-3	1e-4	1e-4	1e-4	1e-4	1e-3	1e-6	1e-6	1e-8	1e-8
13	1e-3	1e-4	1e-4	1e-4	1e-4	1e-3	1e-6	1e-6	1e-8	1e-8
14	1e-3	1e-4	1e-4	1e-4	1e-3	1e-3	1e-4	1e-4	1e-8	1e-8
15	1e-3	1e-4	1e-4	1e-4	1e-3	1e-3	1e-4	1e-4	1e-8	1e-8
16	1e-3	1e-4	1e-4	1e-4	1e-3	1e-3	1e-4	1e-4	1e-8	1e-8
17	1e-3	1e-4	1e-4	1e-4	1e-3	1e-3	1e-4	1e-4	1e-8	1e-8
18	1e-3	1e-4	1e-4	1e-4	1e-3	1e-3	1e-4	1e-4	1e-8	1e-8
19	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
20	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
21	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
22	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
23	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
24	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4
25	1e-3	1e-1	1e-1	1e-2	1e-3	1e-3	1e-4	1e-4	1e-4	1e-4

Table 10.2: Tuning parameters for the kalman filter with respect to the wind speed.

10.3 Constraints implementation

In this work constraints on the values of outputs, inputs and input rates will be implemented . They are presented in table 10.3.

Implemented constraints				
min	\leq	variable	\leq	max
0 <i>kW</i>		P_e		5000 <i>kW</i>
6.9 <i>rpm</i>		Ω_r		12.1 <i>rpm</i>
0 <i>N · m</i>		Q_g		47402.97 <i>N · m</i>
-15000 <i>N · m/s</i>		\dot{Q}_g		15000 <i>N · m/s</i>
-1 <i>deg</i>		θ		90 <i>deg</i>
-8 <i>deg/s</i>		$\dot{\theta}$		8 <i>deg/s</i>

Table 10.3: Constraints imposed on the system variables.

10.4 Stability

The issue of stability of a MPC controller is more complex than in case of classical control since it's main purpose is to enable the use of constraints and when they are active classical linear stability theorems, such as pole placement, are not valid since they are based on linear analysis.

According to [17] there are three conditions that guarantee that a constrained MPC algorithm is stable. These are

- The future inputs trajectory in the current sampling instant $\Delta\mathcal{U}(k)$ contain the optimal inputs trajectory calculated in the previous sampling instant (so called *tail*).
- The output prediction horizon P is infinite in size.
- The control problem in the presence of active constraints has to be feasible.

Those conditions imply that the cost function of the MPC controller is Lyapunov, what in essence means that the cost function will decrease from one sampling instant to the next one.

In practice the above conditions are obtained with the use of so called *dual mode control* approach. A detailed discussion of this subject is presented in [17] and

it won't be elaborated here.

In industrial applications instability is often avoided (but not guaranteed) by introducing sufficiently long input and output horizons [17] and this is the case in this work. Hence, typical stability analysis will not be carried out. Stability is concluded solely from the system's behaviour during simulations.

10.5 Simulations

Below simulations of the MPC controlled wind turbine are carried out. Since the weight scheduling approach is used here the tuning process might become very time consuming. In this case one set of weights is prepared for each integer value of wind speed v , in the range that is of interest to us (from 3 to 25 m/s). There are 8 MPC controller weights and 10 kalman filter weights in each set. This gives a large number of tuning parameters what in turn makes fine tuning a process that might require substantial amounts of time to complete. In this work tuning hasn't been treated as a priority hence although an effort has been put to achieve proper control performance the reader should be aware that there still might exist a room for improvement. It is assumed, in all of the simulations, that the wind speed that is used for gain scheduling is known (measured with perfect accuracy).

10.5.1 Step changes in wind speed

Below results of simulating an MPC controlled wind turbine for a step change in wind speed v is being discussed. Two simulations has been carried out. The first one treats the case when the wind speed v is increasing by 1 m/s every 100 seconds starting at 3 and finishing at 25 m/s . In the second one the wind speed v is changing in the opposite direction. Moreover the constraints are not active. It is important to keep in mind that the reference values for the outputs are calculated with the use of plant's C_p curve and their (outputs') rated values ($P_{e,nom}$ etc.). Hence, as discussed in 7.2.1, the emphasis in the mid region is put on the tracking of rotor's rotational speed Ω_r .

Another important thing to remember is that in reality a pure step change in wind speed wouldn't be possible. A continuous change in wind speed would have to be a very rapid one in order to be perceived by a discrete system, with a relatively small sampling period T_s , as a step change.

10.5.1.1 Step changes in wind speed from 3 to 25 m/s

The results of this simulations are presented on figures from 10.1 to 10.6.

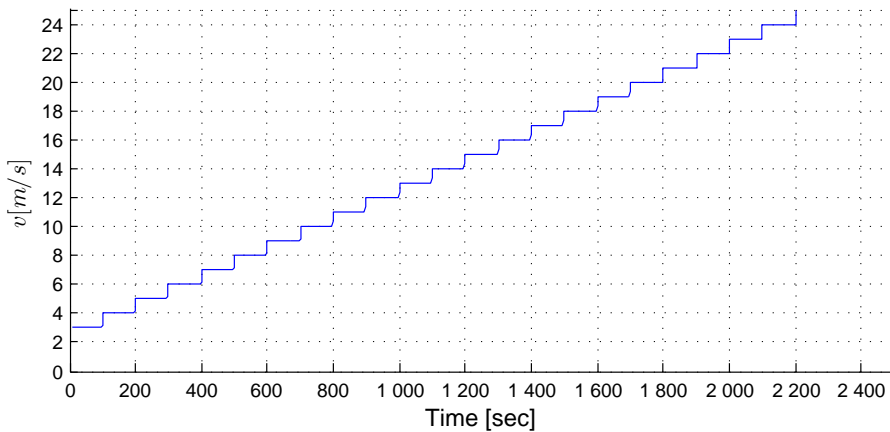


Figure 10.1: Wind speed v change

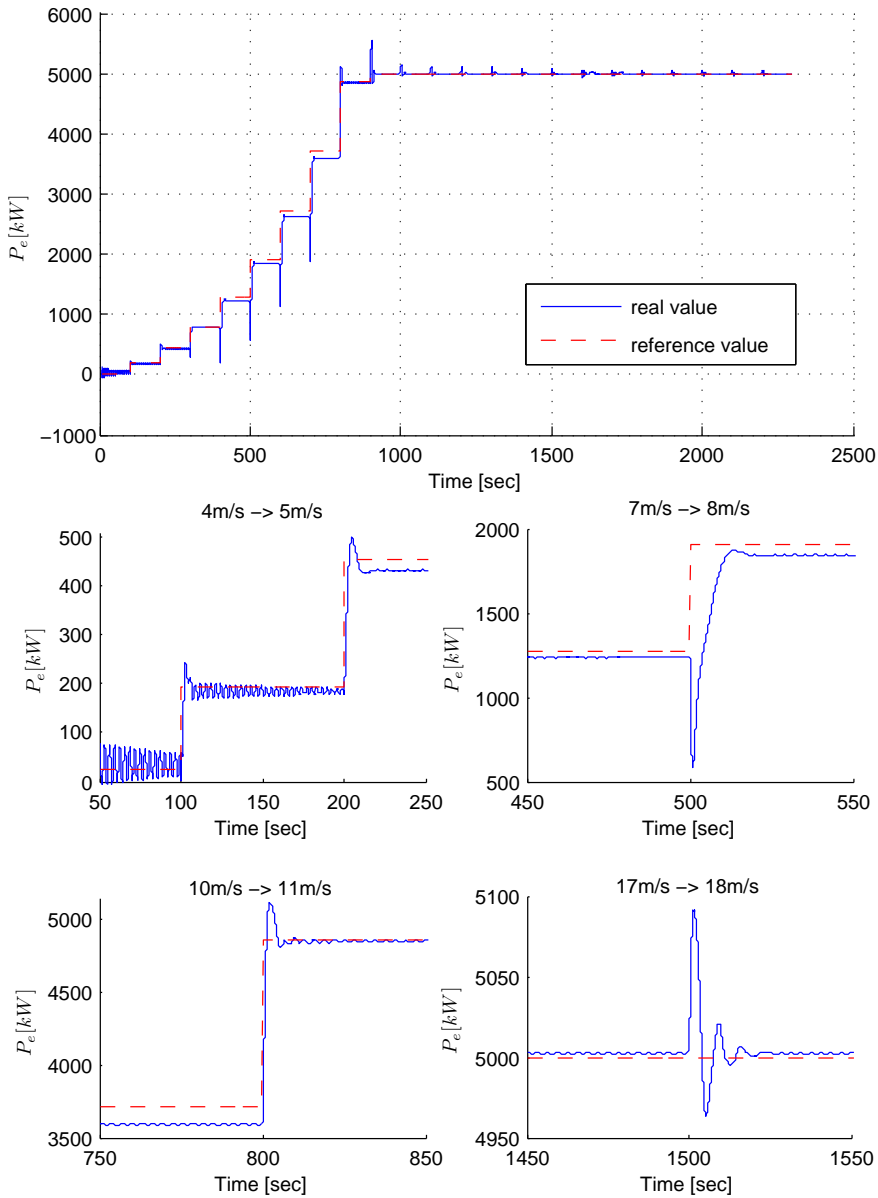


Figure 10.2: P_e reference tracking during wind speed v step changes.

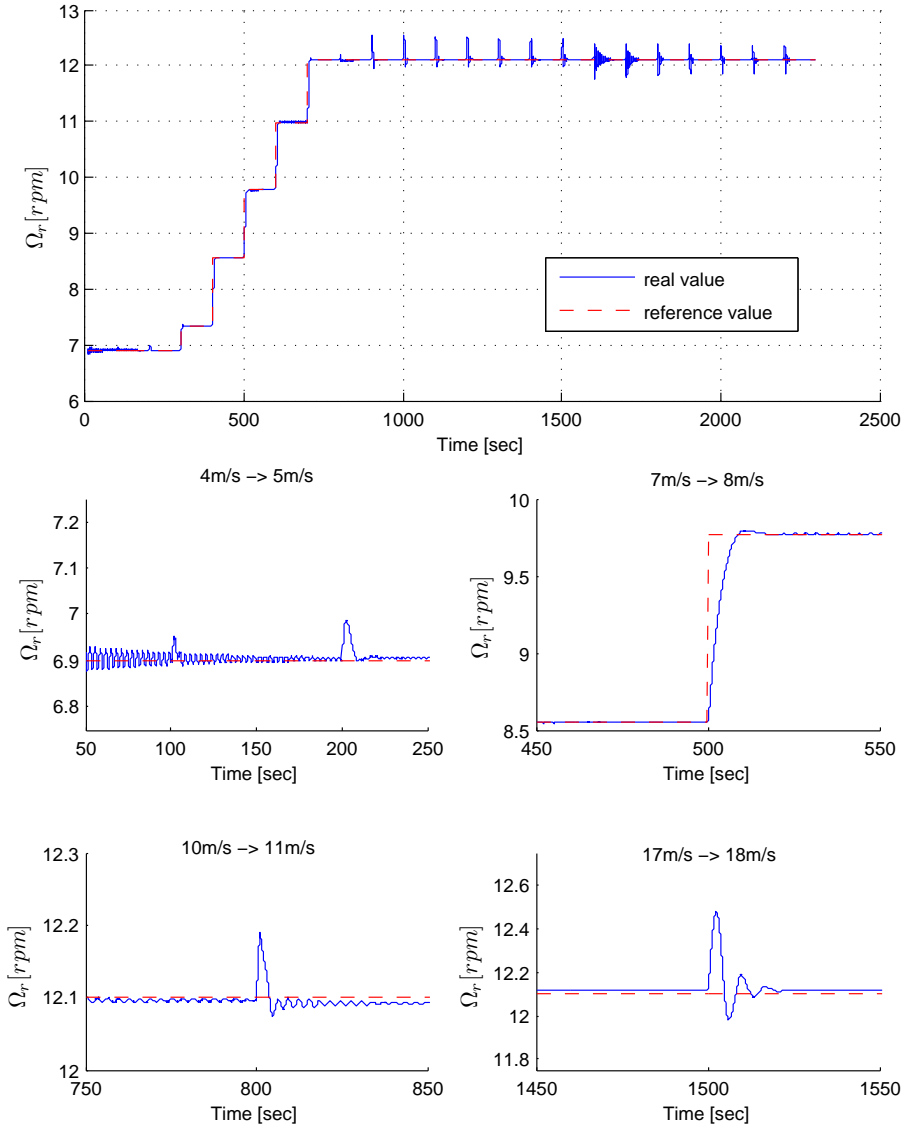


Figure 10.3: Ω_r reference tracking during wind speed v step changes.

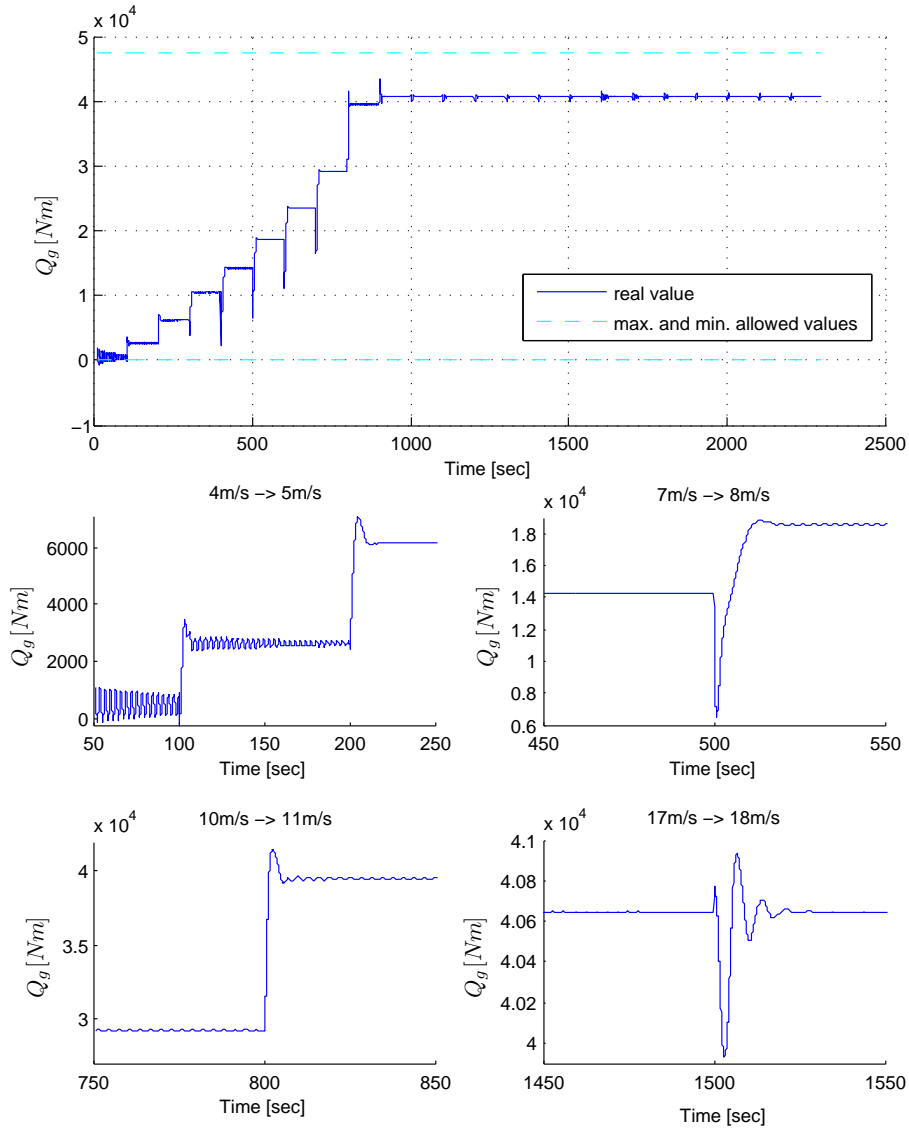


Figure 10.4: Q_g behaviour during wind speed v step changes.

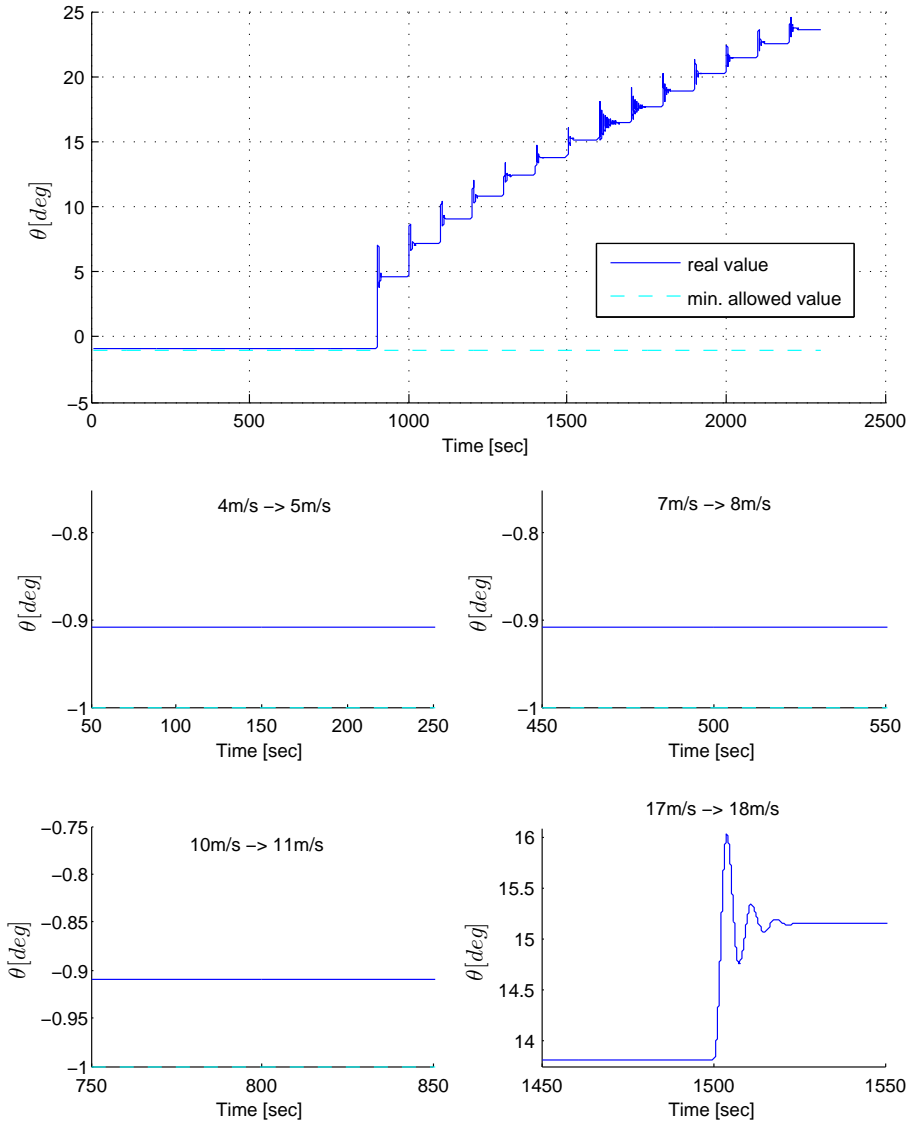


Figure 10.5: θ behaviour during wind speed v step changes.

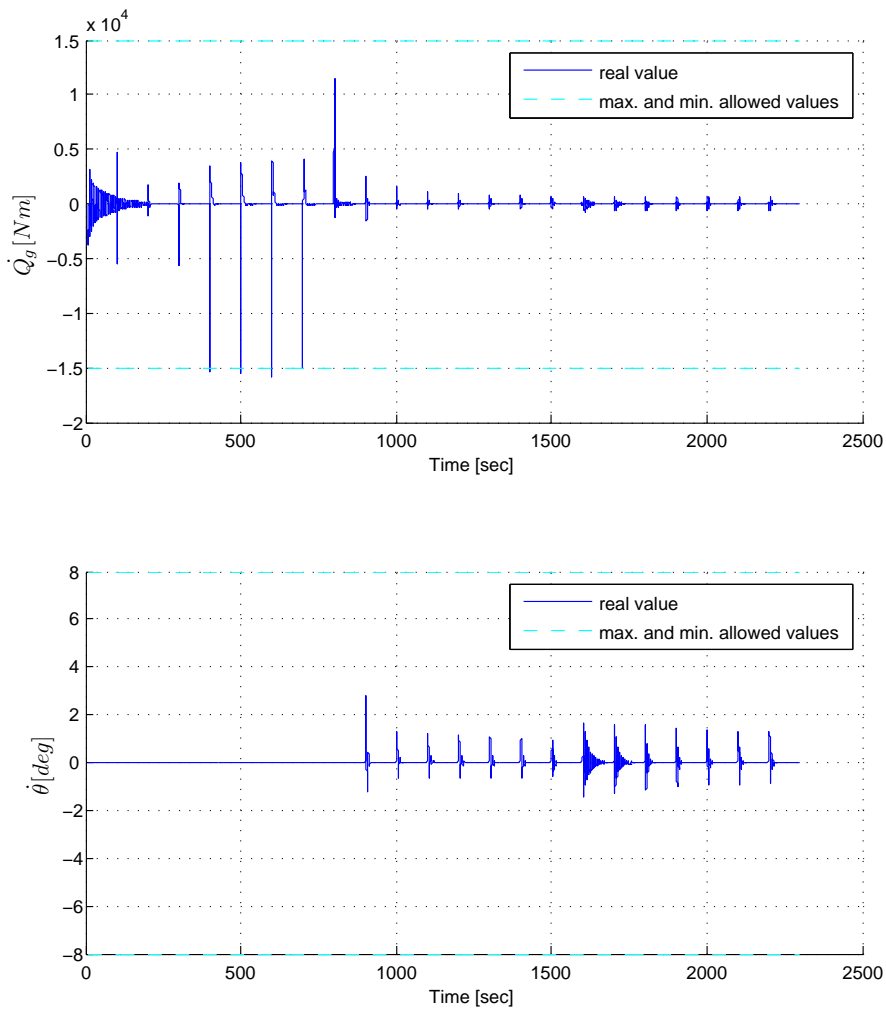


Figure 10.6: Rates of change of the control signals during wind speed v step changes.

10.5.1.2 Step changes in wind speed from 25 to 3 m/s

The results of this simulations are presented on figures from 10.7 to 10.12.

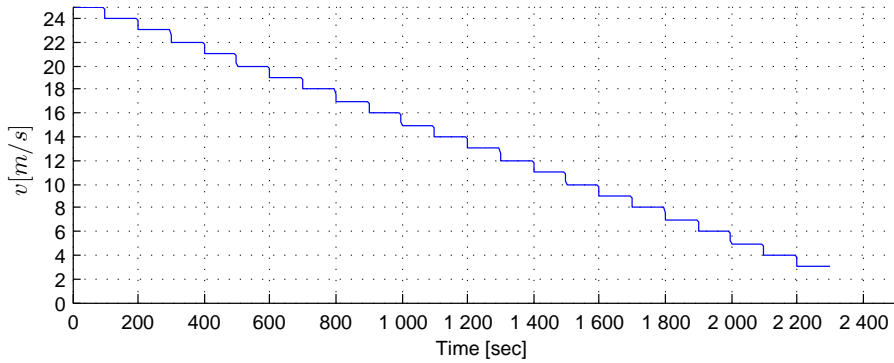


Figure 10.7: Wind speed v change

10.5.1.3 Simulation results discussion

Both simulations prove that the system is working properly. The rate of change of the generator torque is a slightly exceeding it's maximum and minimum allowed values. As mentioned in the beginning of this section a wind speed change that would be perceived by a controller as a step change would be quite unusual in case of a relatively small sampling period T_s . This is the case here, hence it can be anticipated that a situation where the generator torque would get close to it's limits would be quite uncommon.

A significant off-set in the electric power P_e reference tracking in the mid region can be noticed. This is due to the fact that the main focus is put on tracking of the rotor rotational speed Ω_r . In the mid region produced power P_e is closely linked with the rotor rotational speed Ω_r . When the blade pitch value θ is at it's optimum (and it's fixed at this value in this case) the optimum value of Ω_r (calculated with a C_p curve) should give an optimal value of the electric power P_e (through optimal value of the C_p coefficient). Although it's tracked quite accurately this is not the case here. This is most probably an effect of imperfect way in which C_p curve is reflecting the interdependence between those values, what in turn could be caused by, among others, the fact that it is representing only static relations, it's modelling might not be sufficiently accurate etc.

There is a minor steady state error in the tracking of Ω_r . It is most probably caused by imprecise tuning of the kalman filter which might be losing some

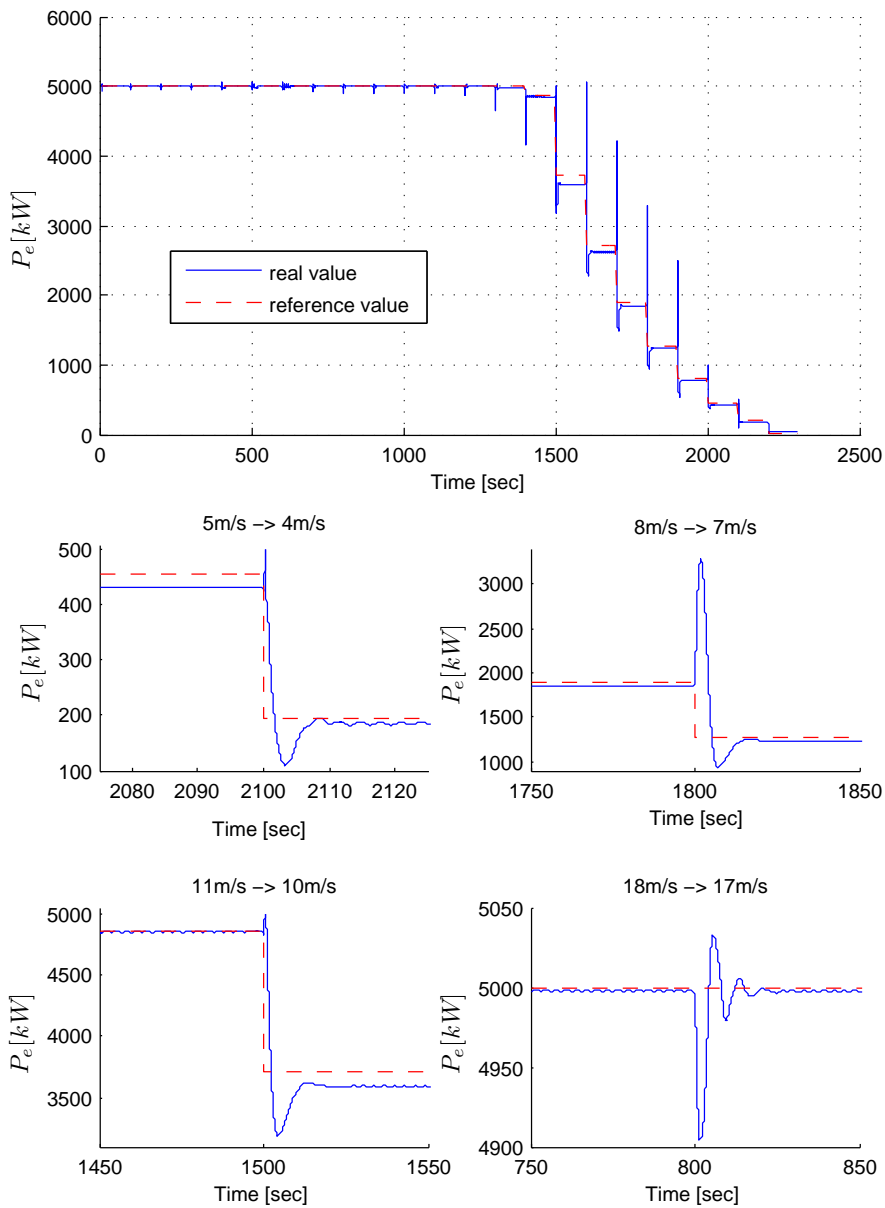


Figure 10.8: P_e reference tracking during wind speed v step changes.

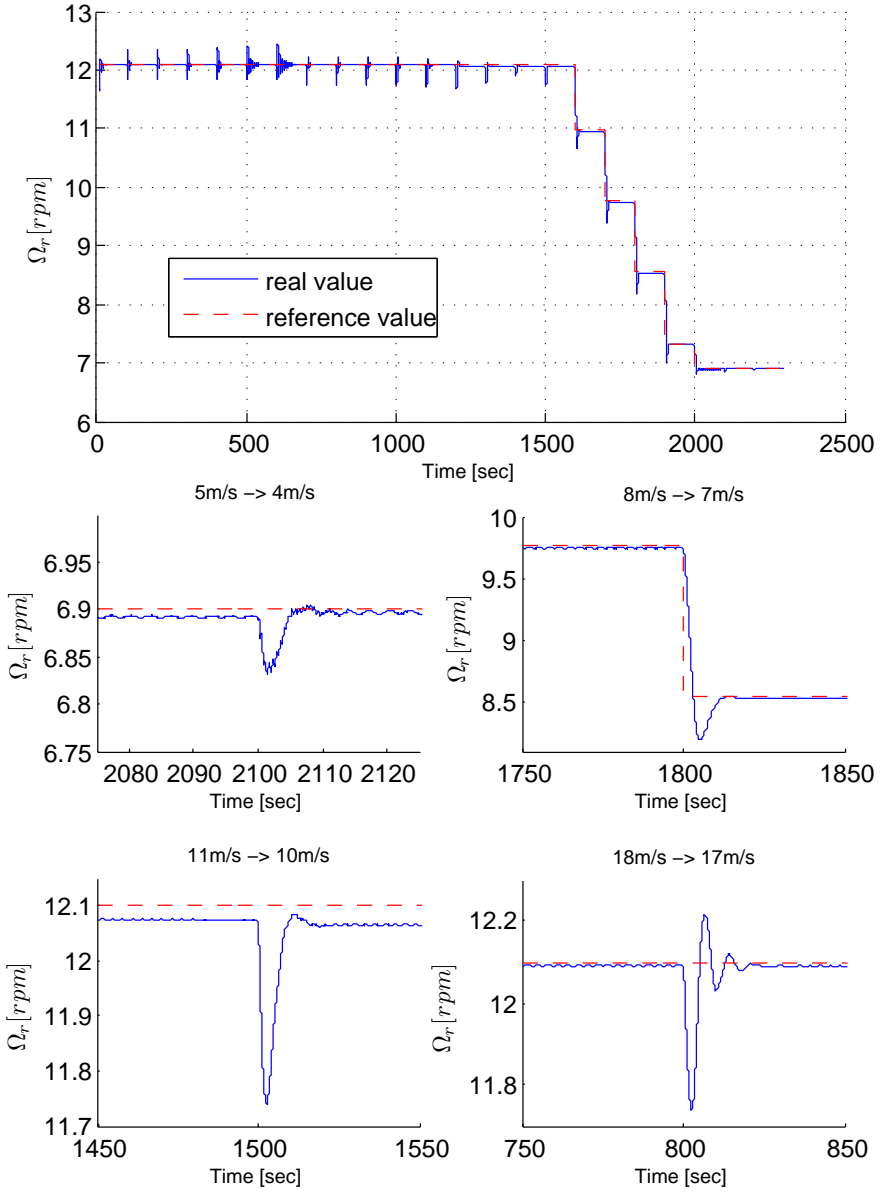


Figure 10.9: Ω_r reference tracking during wind speed v step changes.

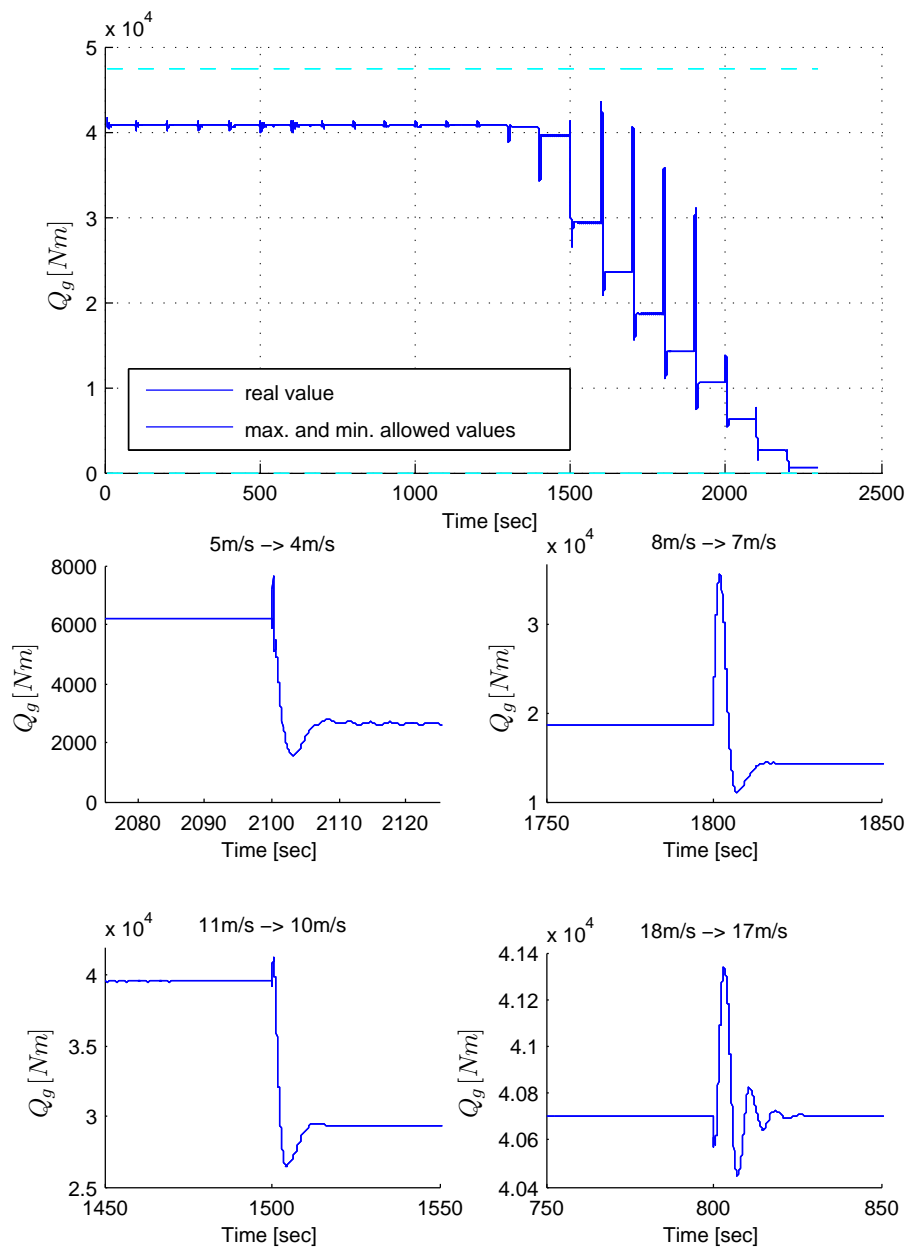


Figure 10.10: Q_g behaviour during wind speed v step changes.

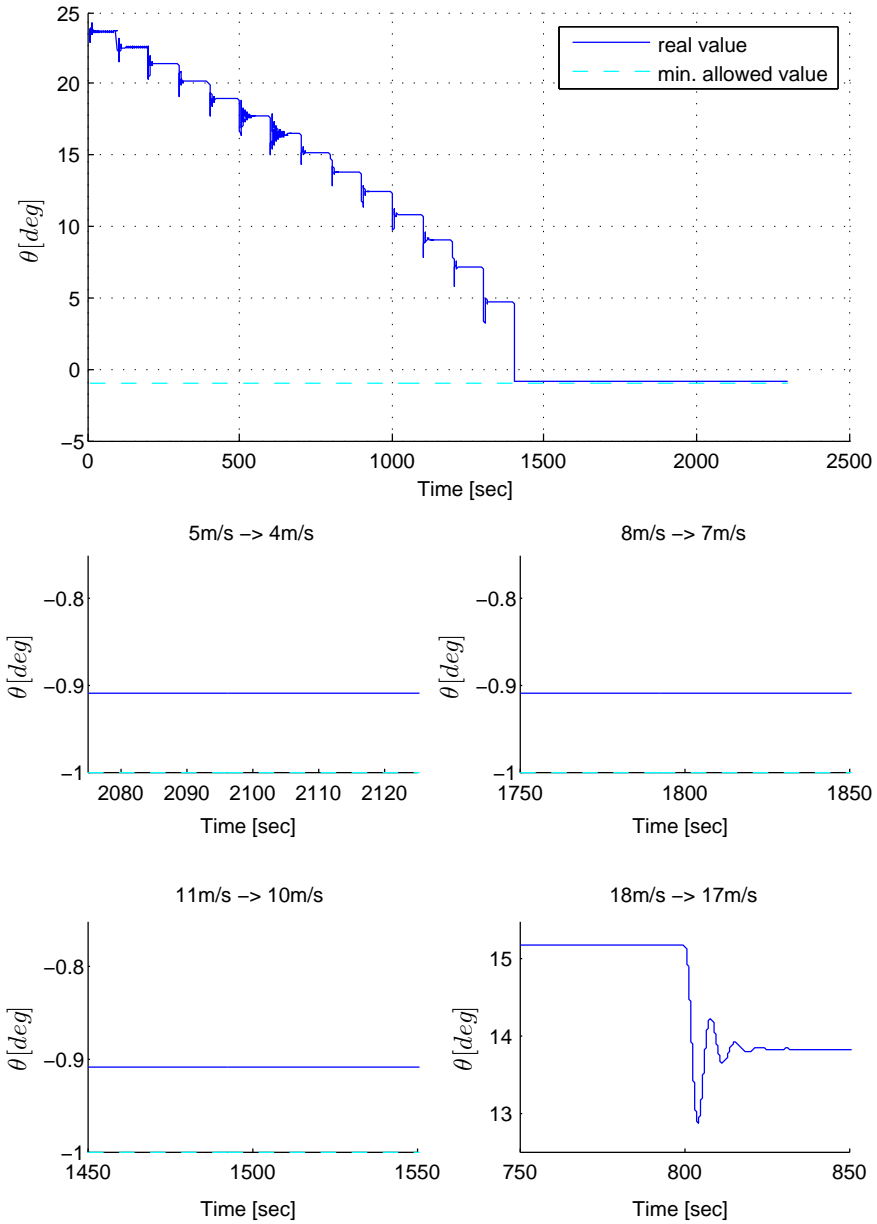


Figure 10.11: θ behaviour during wind speed v step changes.

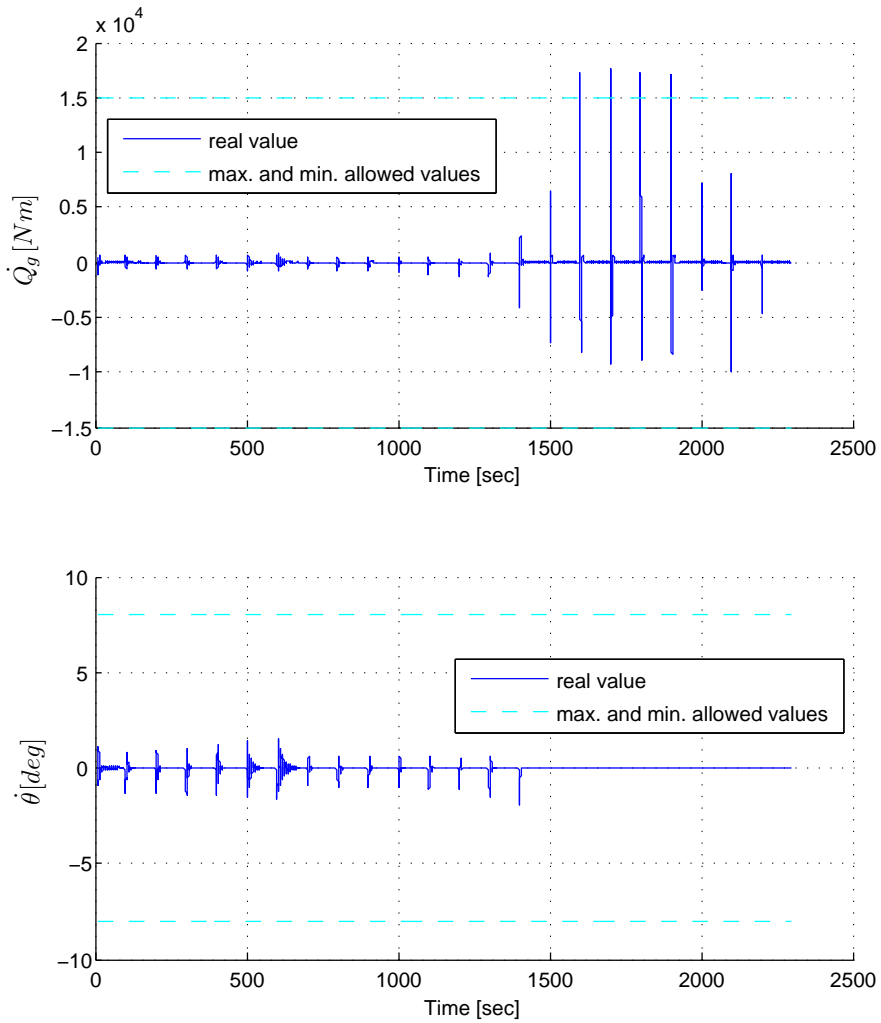


Figure 10.12: Rates of change of the control signals during wind speed v step changes.

information, thus giving biased estimates. Performing fine-tuning should allow to get rid of this offset.

10.5.2 Turbulent wind simulation for a mean wind speed of $v_m = 17[m/s]$

The purpose of the simulation whose results are presented on figures 10.13 through 10.16 is to test the MPC controlled FAST simulated HAWT behaviour under a wind that has a non-zero turbulent component. The wind time series for the simulation were created with the use of TurbSim software. It is a numerical, turbulent wind simulator, developed by NREL (see [9]), that can use a stochastic turbulence model to compute the needed wind inputs. IEC (International Electrotechnical Commission) Kaimal turbulence model has been used in this case. It won't be discussed in more details in this work. [4] presents this model in more details.

The mean wind speed value for which the turbulence have been computed is $v_m = 17[m/s]$. It was chosen so high in order to keep the wind turbine in the top region where both reference values for P_e and Ωr are constant.

The real wind dynamics are much different from those represented by an artificial step change in e.g. the previous simulation. In order to obtain good control of this kind of signal a retuning of the controller in the whole top region is needed. For reasons mentioned in the beginning of this section only a rough tuning has been carried out.

10.5.2.1 Simulation results discussion

The simulation shows that the control of the wind turbine with the use of MPC in the presence of stochastic wind turbulences is carried out in a way that gives much poorer performance than in case of the step change of the wind speed v . It is not a surprise since step signal doesn't present any dynamic in any instant beside the one that the step actually take place in. Stochastic disturbance with high dynamic are much more difficult to control. It should be noted that the controller was tuned only roughly for this simulation and that reasonably better results in terms of performance should be possible to achieve.

Simple analysis of the time series of the system outputs has been performed and their results are as follows

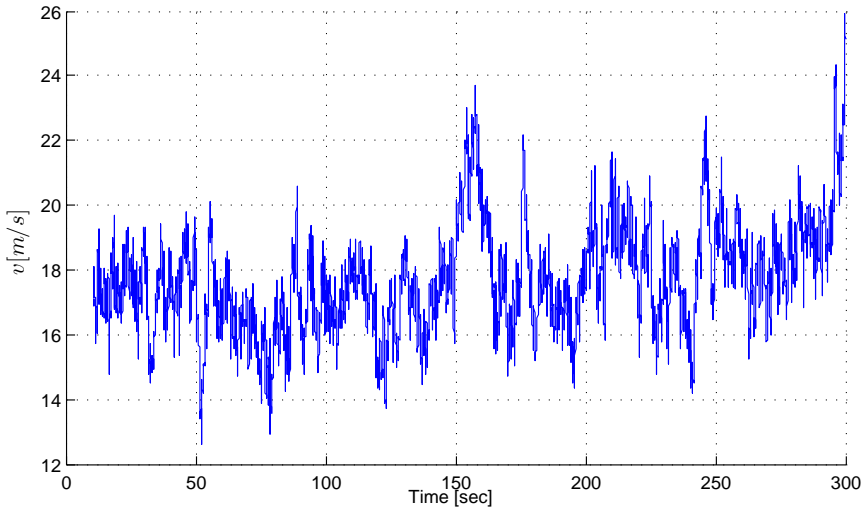


Figure 10.13: Wind turbulence created with IEC Kaimal model for the mean wind speed value of $v_m = 17m/s$.

Electric power P_e

mean value $\mu_{P_e} = 4995.20kW$

standard deviation $\sigma_{P_e} = 25.91kW$

Rotor speed Ω_r

mean value $\mu_{\Omega_r} = 12.093rpm$

standard deviation $\sigma_{\Omega_r} = 0.3399rpm$

From basic statistics it is known that approximately 68% of values of the given data set, provided that it is normally distributed, lay within one standard deviation σ from its mean value and approximately 95% within two. The distribution of the data set samples for the two outputs is shown on figure 10.17.

From this we can say that both of our time series display normal distribution and further that

- Approximately 95% of the electric power P_e time series samples is different by not more than

$$\frac{2\sigma_{P_e}}{P_{e,nom}} 100\% = \frac{2 \cdot 25.91kW}{5000kW} 100\% = 1.0366\% \quad (10.2)$$

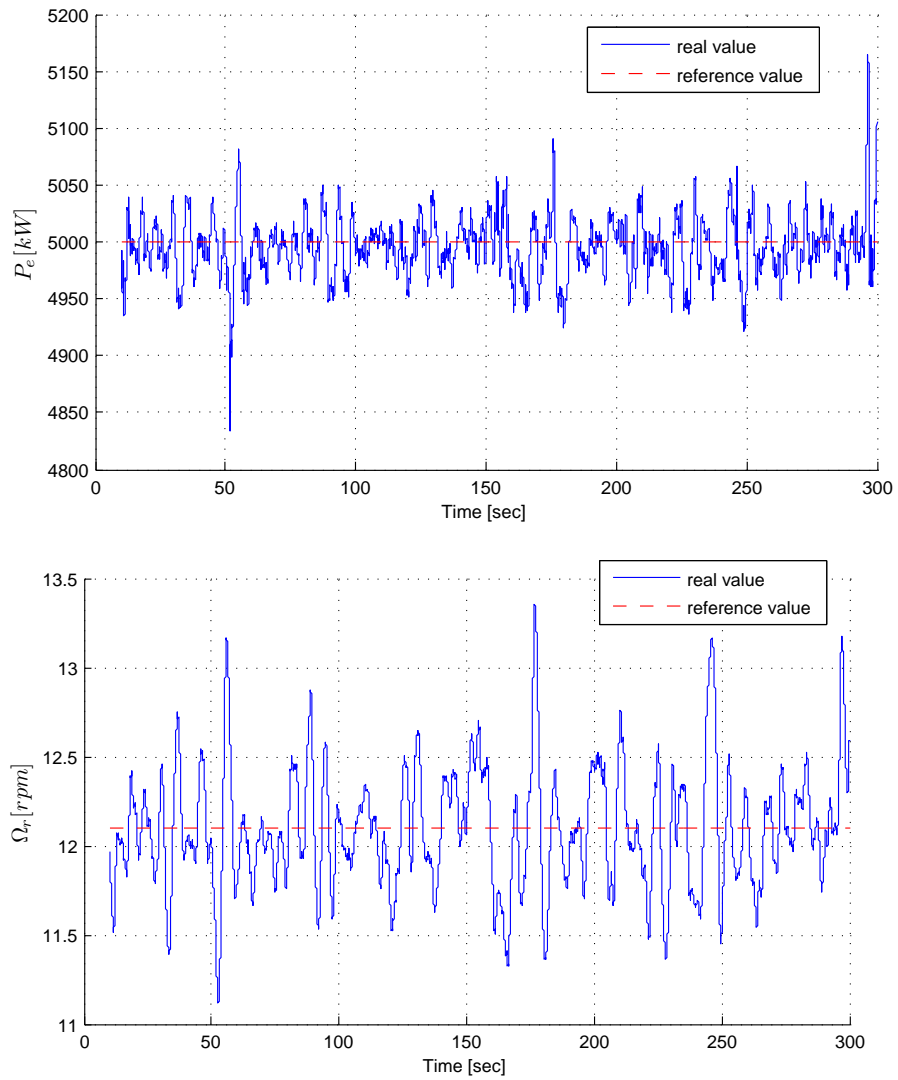


Figure 10.14: Electric power P_e and Rotor rotational speed Ω_r in wind turbine simulation with stochastic turbulences consistent with IEC Kaimal model.

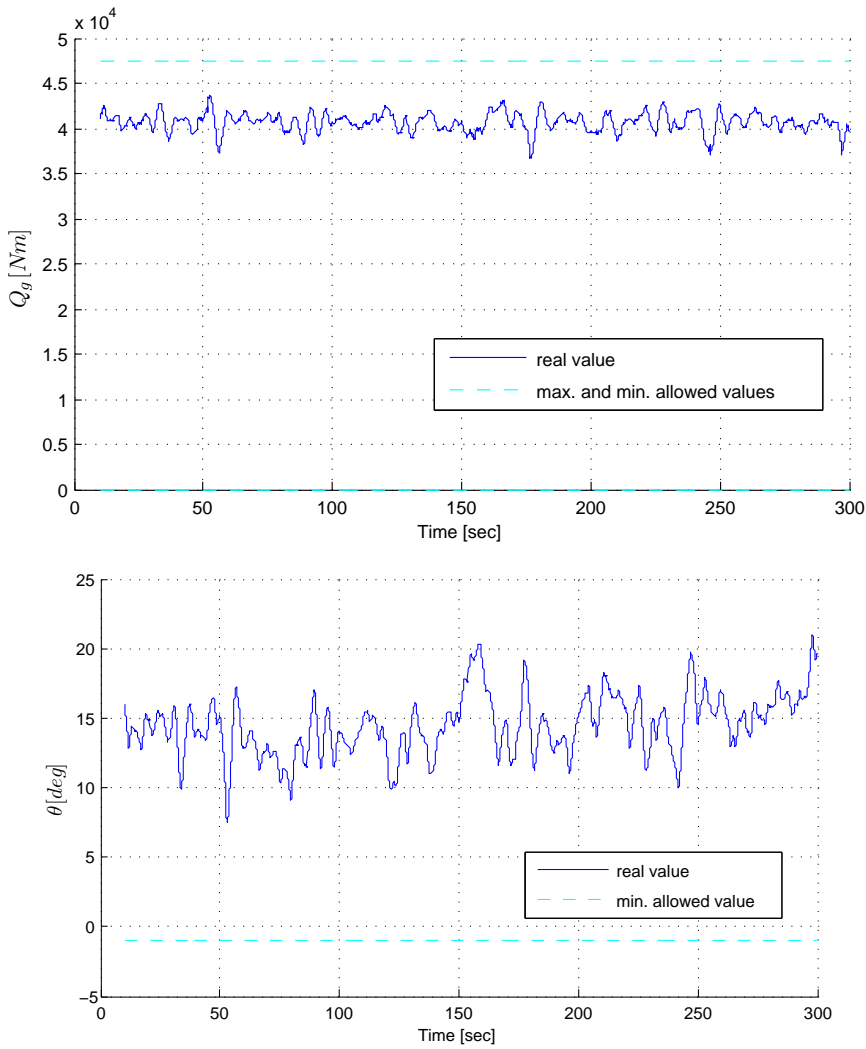


Figure 10.15: Generator torque Q_g and collective blade pitch θ in wind turbine simulation with stochastic turbulences consistent with IEC Kaimal model.

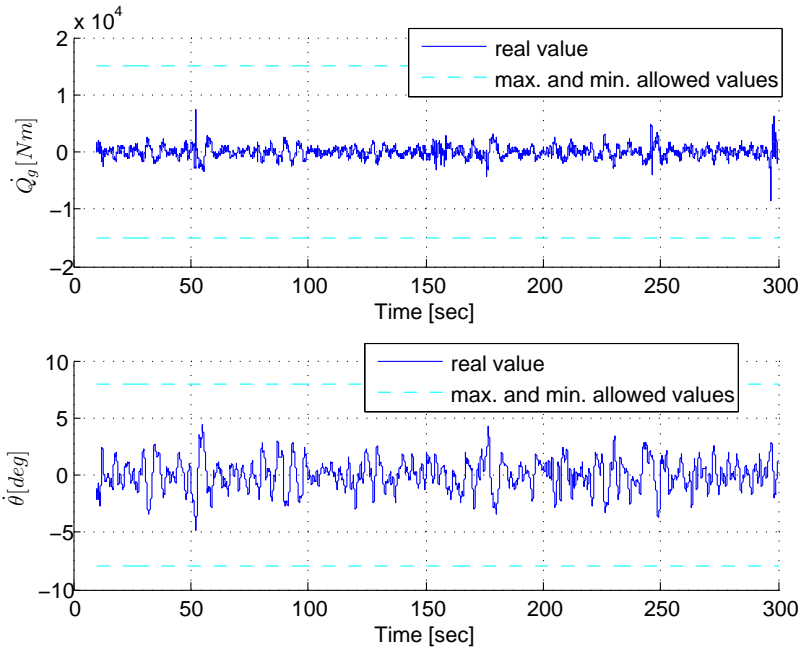


Figure 10.16: Rates of the wind turbine’s control signals in wind turbine simulation with stochastic turbulences consistent with IEC Kaimal model.

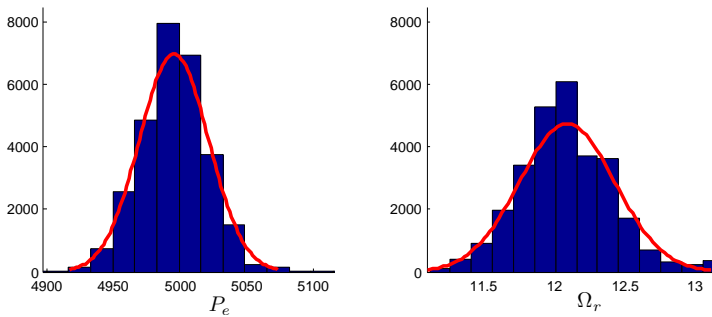


Figure 10.17: Distribution of the system outputs for the stochastic wind turbulence simulation.

from it’s nominal value $P_{e,nom}$

- Approximately 95% of the rotor speed Ω_r time series samples is different by

not more than

$$\frac{2\sigma_{\Omega_r}}{\Omega_{r,max}} 100\% = \frac{2 \cdot 0.3399rpm}{12.1rpm} 100\% = 2.809\% \quad (10.3)$$

from it's nominal value $\Omega_{r,max}$

The above calculations consider samples at and after the 50th second of the simulation. The data have been limited in this way in order to eliminate the influence of the startup transient on the results.

10.5.3 Simulation of a wind turbine controlled with a constrained MPC

Below, results of simulations, with MPC constraints enabled, are shown. The basic setup is the same for all them. The wind speed in each is changing from the value of 13 to the value of 18 m/s in a step manner. It is depicted on figure 10.18. This is a purely abstract situation (in reality wind changes don't exhibit such a high values of acceleration) having nothing in common with real life application and was used here with the sole purpose of illustrating the work of the constrained MPC. Also the values of constraints were chosen arbitrarily in order to simplify the simulations. The test carried out are:

- Operation with no constraints enabled
- Operation with constraints for the maximum value of the blade pitch velocity $\dot{\theta}$ enabled
- Operation with constraints for the maximum value of the blade pitch θ enabled
- Operation with constraints for the maximum value of the produced power P_e enabled
- Operation with constraints for the maximum value of the blade pitch θ and softened constraints for maximum value of the produced power P_e and rotors speed Ω_r enabled

The simulations concern only the constraints for maximum values of one of the outputs, one of the control signals and it's velocity. The minimum value constraints, constraints for another control signal and for another output are implemented in the same way.

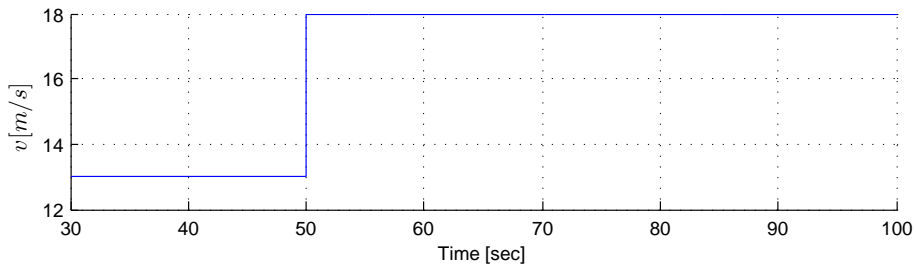


Figure 10.18: Wind speed change for the simulations concerning constrained MPC.

10.5.3.1 Operation with no constraints

In this simulation no constraints were enabled. It's purpose is to give the reader a reference point for the simulations that follows. The results are depicted on figure 10.19.

10.5.3.2 Operation with constraints for the maximum value of the blade pitch velocity $\dot{\theta}$

The setup for this simulation is the same as in the previous one. The only difference is that a constraint of 4 *deg/s* on the blade pitch velocity has been enabled. The results are presented on figure 10.20.

10.5.3.3 Operation with constraints for the maximum value of the blade pitch θ

In comparison with the previous simulation the constraint on $\dot{\theta}$ has been disabled and a constraint of 18 *deg* for the blade pitch has been introduced. The results of the simulation are depicted on 10.21

10.5.3.4 Operation with constraints for the maximum value of the produced power P_e

In this simulation all previously used constraints have been disabled and a constraint of 5350 *kW* has been enabled for the produced power P_e output. 10.22

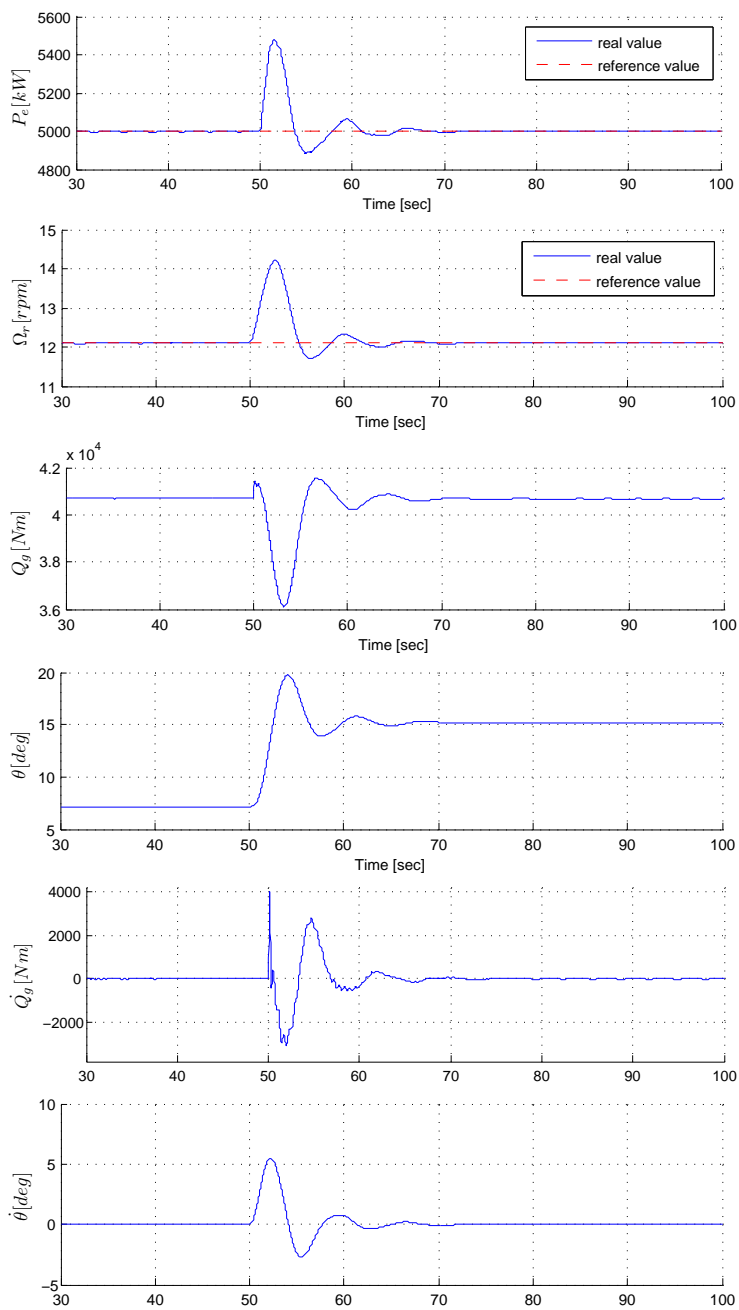


Figure 10.19: Results for the simulation with no constraints enabled.

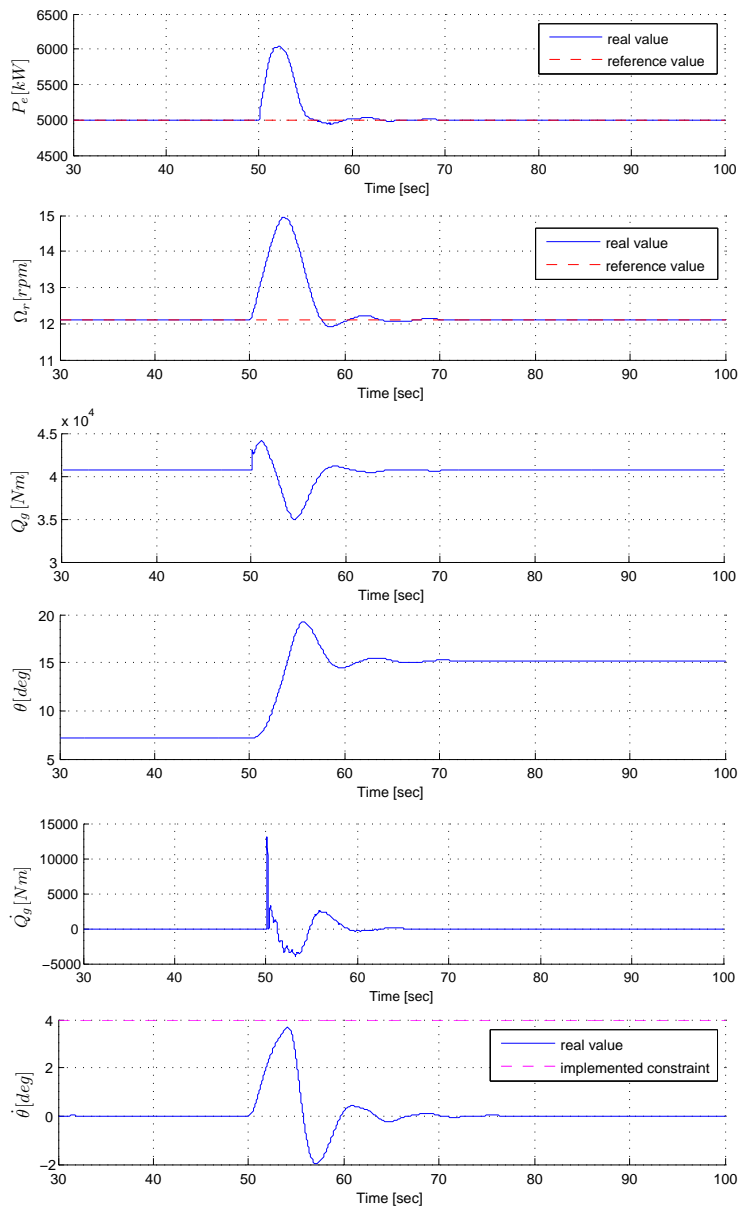


Figure 10.20: Results for the simulation with blade pitch velocity constraint enabled.

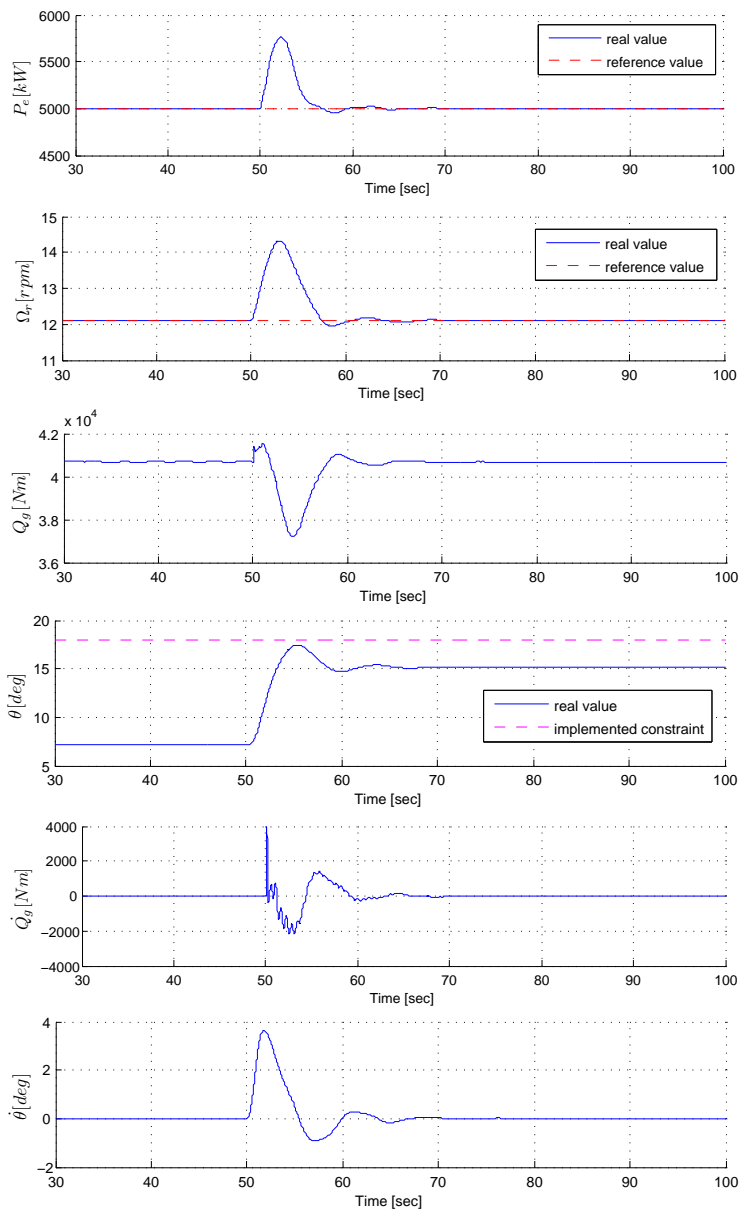


Figure 10.21: Results for the simulation with blade pitch constraint enabled.

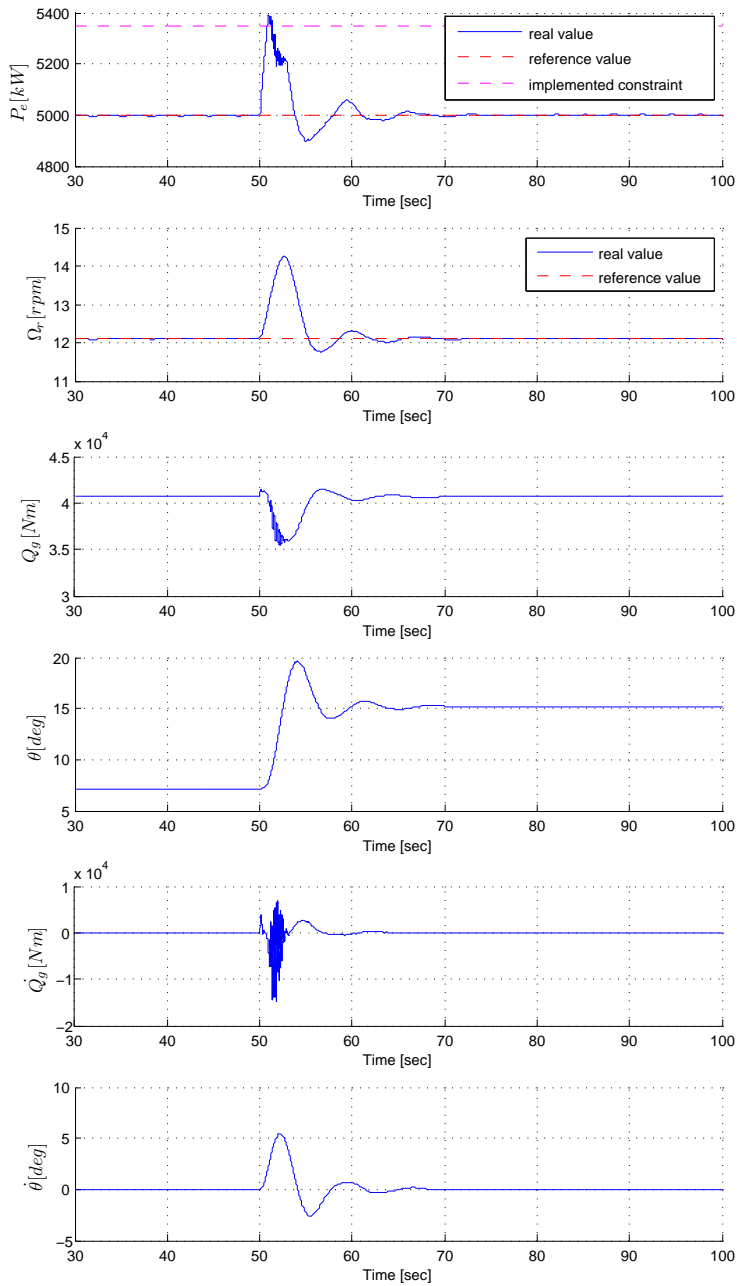


Figure 10.22: Results for the simulation with produced power constraint enabled.

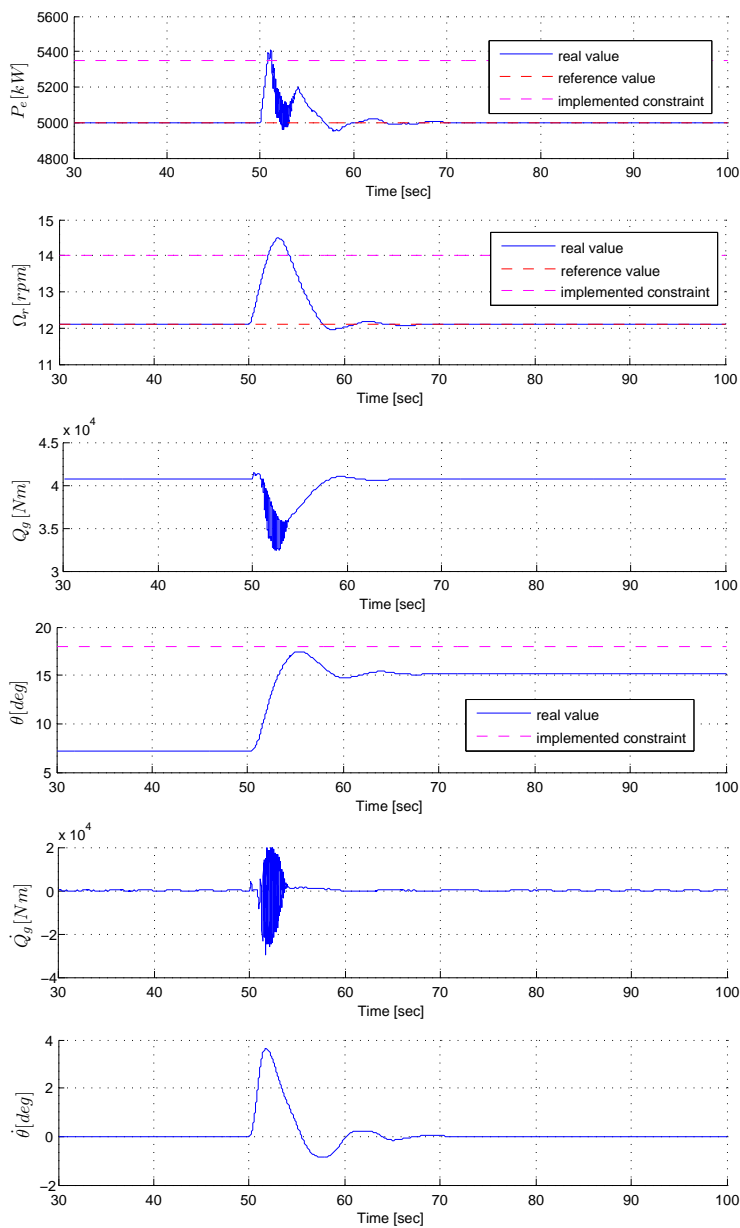


Figure 10.23: Electric power P_e and Rotor rotational speed Ω_r output for simulation with no constraints enabled.

10.5.3.5 Operation with constraints for the maximum value of the blade pitch θ and softened constraints for maximum value of the produced power P_e and rotor rotational speed Ω_r

The constraints introduced in this simulations were a combination of those from two previous ones and additional for the rotor speed Ω_r . It has a value of 14 *rpm*. The output constraints were softened in comparison with the constraints presented before. It has been done by lowering the values of the S weight in the cost function (see 6.3.4). Carrying out a simulation with this set of constraints without softening them leads to failure of the QP solving algorithm (problem is infeasible). A proper simulation addressing this situation have been carried out, still the plots of it's inputs and outputs wouldn't reveal anything significant, thus they have been omitted.

10.5.3.6 Simulation results discussion

The above simulations prove that both the constraints that were mentioned in 6.3 and the technique that was used to soften them work as intended. Comparison of the results for the non-constrained MPC with a constrained one reveal that having constraints for one or more variables active can push other towards their constraints as well. This can result in instability of the system.

It can also be noticed that during the simulation whose results were presented in section 10.5.3.4 the produced power P_e is actually exceeding it's constraint. This is due to the fact that the controlled plant doesn't have exactly the same dynamics as the model that is used by it's controller and even with disturbance estimation there are many unavoidable errors. This imperfections are smaller in case of the control signal because the dynamics that are producing them are the dynamics of the controller itself connected to the actuator models that were implemented as ideal systems. Simulation with the soft constraints enabled proved to save the QP problem from becoming infeasible - the outputs were allowed to get beyond the area of normal operation at the cost of significant increase of the value of the cost function. Simulating system with the same setup but without softening would cause the QP problem solving algorithm to fail to converge and so the input signals (reference signals for the actuators) would not be calculated.

10.6 Chapter summary

In this chapter the implementation of the control systems, that have been introduced in the previous part of this work, have been discussed. It's analysis have

been carried out and simulations have been performed. Their results have been discussed and the overall findings concerning the implementation have been presented.

Based on simulation outcome and analysis performed it can be concluded that the controller have been implemented successfully

The simulations revealed that it's tuning is one of the most significant factors with respect to the control performance. Still with the use of weight scheduling technique obtaining proper weights is a time consuming process. It's benefit is that it enables more control over control performance in the whole spectrum of wind speeds, while not increasing computational burden, in a significant way, at the same time.

Part IV

Conclusions

Conclusions

In this section conclusions, concerning the work carried out, will be drawn.

11.1 Modelling

Models representing different systems linked with wind turbine have been presented in part I. They were all first principle models - they were derived from physical equations. One of the wind turbine subsystems (generator and drivetrain) have been relinearized in a numerical way, with the use of FAST linearization tool, in the attempt to obtain certain parameters whose values were unknown. After deriving the relation between the parameters of both models they were connected into one - a mixed model. Its simulation in chapter 9 proved that this approach has been successful.

The model derived have been a relative one - the values of its variables were expressed in relation to their linearization points and not in global values. This was changed by transforming it into an affine model in chapter 4.2. Although it isn't a necessary operation it proved to simplify the process of designing the controller.

11.2 Controller

In part II the control methods used in this work have been presented. An attempt to obtain offset-free control with the use of disturbance modelling approach, discussed in chapter 5, have been made. The controller coupled with the kalman filter wasn't able to remove the off-set completely. Probably due to rough-tuning approach that have been assumed in this work.

In chapter 6 Model Predictive Controller have been introduced. It's proper implementation have been proven in simulations whose results are presented in section 10.5. It is worth of noticing that it's formulation, contrary to the standard approach that can be fined for example in [13] or [17], is extended for the case in which there is a direct influence of the input signal on plant's outputs (non-zero "D" matrix).

Simulations of MPC controlled wind turbine with the presence of step changes in wind speed (see 10.5.1) proved that the implementation of the gain scheduling and weight scheduling techniques (see chapter 7) were successful. With the use of fixed set of MPC and kalman filter weights it would be difficult to obtain stable control not to mention good performance performance. In this case however smooth transition between the operation regions can be obtained.

Judging from the simulations whose results were presented in section 10.5.3 the constraints introduced in section 6.3 and the softening of the output constraints (see section 6.3.4) have also been implemented in a proper way what not only allows to introduce limitations for certain variables in the system but also enables handling of otherwise infeasible problems.

11.3 Implementation

During the course of implementing the system it became apparent that proper scaling of the input variables was necessary in our case. Without it numerical errors would force the system to behave in an improper way.

The use of such future wind speed measuring device as LIDAR might improve systems behaviour during the stochastic wind turbulence simulation if the results would be fed into the disturbance trajectory or have been used for the gain scheduling algorithm. Although wind estimates obtained with the kalman filter weren't used for this (gain scheduling) purpose it seems that with proper fine tuning using them in this fashion would be possible.

The control performance during the stochastic wind turbulence simulation raises significantly with the increase of sampling frequency what have been concluded during simulation's preparation (not documented here). This is due to the fact that in this case controller has the chance to react faster to the wind speed changes.

It was noticed that during the simulations that would involve soft constraints activation the algorithm would significantly slow down while attempting to obtain proper solution by violating output constraints. Specialized algorithms might address this issue and their use is recommended.

FAST Linearization setup

The main FAST configuration files that are of importance in the linearization process are:

- *.fst file - the main FAST configuration file. It will be referred to as [FST] file
- *_Linear.dat - the main linearization module configuration file. It will be referred to as [.DAT] file

Below the following notation will be used:

”.DAT//MODEL LINEARIZATION//(NAzimStep)” will refer to [NAzimStep] parameter in the [MODEL LINEARIZATION] section of the [.DAT] file.

As pointed out in section 2.2 the way in which wind turbine is being controller is strictly connected to the current wind speed. Furthermore, due to the restrictions introduced by it's (wind turbine's) designers, four different operation modes are distinguished. Which of them is currently active depends also on the speed of the wind at the given instant. This emphasizes the necessity of having multiple linear models constructed around different wind speed operation points in order to be able to achieve good control.

Keeping in mind different control approaches in each region, described in section

2.2, we will set FAST parameters differently for each mode of operation. Those parameters will include:

- .FST//TURBINE CONTROL//(BIPitch(1) through BIPitch(3)) - represents the initial or fixed pitch of blades 1 through 3.
- .FST//INITIAL CONDITIONS//(RotSpeed) - represents the initial or fixed rotor speed
- .DAT//PERIODIC STEADY STATE SOLUTION//(TrimCase) - represents the control strategy for the model. It's possible settings are:
 - 1 - find nacelle yaw (not useful in our case since it is assumed that there is no yaw control)
 - 2 - find generator torque while the blade pitch is kept constant
 - 3 - find collective blade pitch while the generator torque is kept constant

The main tool for obtaining proper values of those parameters is the C_p curve. We use it for calculating the border wind speeds ($v_1...v_4$) for the operation modes, optimum blade pitch (θ_{opt}) that would give maximum power coefficient C_p (provided that tip speed ratio would be optimal as well) and rotor speed that would give optimum tip speed ratio (see (2.1)) for the given wind speed. Control strategy represented by the [TrimCase] parameter will be chosen based on the region characteristics outlined in 2.2.

Guidelines for setting those parameters for the purpose of linearization are summarized in Table A.1.

Region I (low): $v_1 \dots v_2$ $\Omega_{r_{min}}$ limit reached	
parameter	value
BiPitch(1)	optimal pitch θ_{opt}
BiPitch(2)	optimal pitch θ_{opt}
BiPitch(3)	optimal pitch θ_{opt}
RotSpeed	lower limit for rotors speed $\Omega_{r_{min}}$
TrimCase	2
Region II (mid): $v_2 \dots v_3$ no limits reached	
parameter	value
BiPitch(1)	optimal pitch θ_{opt}
BiPitch(2)	optimal pitch θ_{opt}
BiPitch(3)	optimal pitch θ_{opt}
RotSpeed	calculated with the use of C_p curve
TrimCase	2
Region III (mid): $v_3 \dots v_4$ $\Omega_{r_{max}}$ limit reached	
parameter	value
BiPitch(1)	optimal pitch θ_{opt}
BiPitch(2)	optimal pitch θ_{opt}
BiPitch(3)	optimal pitch θ_{opt}
RotSpeed	upper limit for rotors speed $\Omega_{r_{max}}$
TrimCase	2
Region IV (mid): $v_4 \dots$ $\Omega_{r_{max}}$ and $P_{e_{max}}$ limits reached	
parameter	value
BiPitch(1)	doesn't matter
BiPitch(2)	doesn't matter
BiPitch(3)	doesn't matter
RotSpeed	upper limit for rotors speed $\Omega_{r_{max}}$
TrimCase	3

Table A.1: Guidelines for choosing values of the key parameters of the FAST linearization module for different operation modes.

Bibliography

- [1] International Energy Agency. World energy outlook 2010.
- [2] F.D. Bianchi, Hernán De Battista, and R.J. Mantz. *Wind Turbine Control Systems. Principles, Modelling and Gain Scheduling Design*. Springer-Verlag, 2007.
- [3] F. Borrelli, M. Morari, and U. Maedera. Linear offset-free model predictive control. *Automatica*, 45, 2009.
- [4] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi. *Wind energy handbook*. John Wiley & Sons Ltd, 2001.
- [5] E. Hendricks, O. Jannerup, and P.H. Sørensen. *Linear Systems Control. Deterministic and Stochastic Methods*. Springer-Verlag, 2008.
- [6] L.C. Henriksen. Model predictive control of a wind turbine. Master's thesis, Technical University of Denmark, 2007.
- [7] L.C. Henriksen. *Model Predictive Control of Wind Turbines*. PhD thesis, Technical University of Denmark, 2010.
- [8] L. Jespersen and M.W. Krogen. Model predictive lidar control of wind turbines for load mitigation. Master's thesis, Aalborg University, 2010.
- [9] B.J. Jonkman. Turbsim user's guide: Version 1.50. Technical report, NREL, 2009.
- [10] J.M. Jonkman and M.L.Jr. Buhl. Fast user's guide. Technical report, NREL, 2005.

-
- [11] J.M. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical report, NREL, 2009.
 - [12] Lecture notes for model predictive control course (02619) on dtu, 2010.
 - [13] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.
 - [14] T.S. Mogensen and A.J. Larsen. Individuel pitchregulering af vindmølle. Master's thesis, Technical University of Denmark, 2006.
 - [15] J. Nocedal and S.J Wright. *Numerical Optimization*. Springer-Verlag, 2006.
 - [16] J.B. Rawlings and G. Pannocchia. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2), 2003.
 - [17] J.A. Rossiter. *Model-Based Predictive Control. A Practical Approach*. CRC Press, 2004.
 - [18] K. Selvam, S. Kanev, J. W. van Wingerden, T. van Engelen, and Verhaegen. Feedback-feedforward individual pitch control for wind turbine load reduction. *International Journal of Robust and Nonlinear Control*, 19, 2009.
 - [19] Kasper Zinck Østergaard, Jakob Stoustrup, and Per Brath. Linear parameter varying control of wind turbines covering both partial load and full load conditions. *International Journal of Robust and Nonlinear Control*, 19, 2009.
 - [20] P. Østergaard. Pitchregulering af en vindmølle. Master's thesis, Technical University of Denmark, 1994.
 - [21] S.C. Thomsen. Nonlinear control of a wind turbine. Master's thesis, Technical University of Denmark, 2006.
 - [22] L. Wang. *Model Predictive Control. System Design and Implementation Using MATLAB*. Springer-Verlag, 2009.
 - [23] A.D. Wright and L.J. Fingersh. Advanced control design for wind turbines. part i: Control design, implementation, and initial tests. Technical report, NREL, 2008.