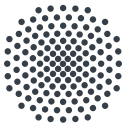


Presented at the

University of Stuttgart
Germany

Master Thesis

Development of a Model Predictive Controller for Floating Offshore Wind Turbines

Presented by Samuel Nann
at 2020-05-20

Supervisors: Wei Yu, M.Sc.
Dr.-Ing. Steffen Raach
Dr.-Ing. Frank Lemmer geb. Sandner
Prof. Dr. Po Wen Cheng

Abstract

In this work, an Economic Model Predictive Controller for a floating offshore wind turbine is presented. The classical Model Predictive Control for floating offshore wind turbines provides promising results. In addition, research on onshore wind turbines revealed the potential of the economic control method, which can improve the closed-loop behavior and simplify the control design in comparison to the classical version of this control method. The aim of this work is, to develop a novel Economic Model Predictive Controller for a floating offshore wind turbine based on these two research results.

A simplified low order model of a floating offshore wind turbine serves as a basis for the controller design. Including the disturbance preview and constraints, the controller computes optimal trajectories for the blade pitch and the generator torque. To apply the control technique to a floating offshore wind turbine two things have to be done: Firstly, the cost function is designed, to fulfill the main objectives of, maximizing the generated power and alleviating the structural fatigues. Secondly, the constraints are integrated into the control problem. After selecting a suitable solver, the controller is discretized and scaled, thus a proper implementation and smooth operation is possible. Afterwards, the successful functioning of the algorithm, a multi-objective optimization is done, to find appropriate weights to adjust the cost function for the required objectives.

Finally, the developed controller is tested with realistic wind and wave disturbances. A significant reduction of the standard deviation of the generated power can be shown, while maintaining real time capability. Furthermore, the structural fatigues of the tower and the platform are decreased.

KURZFASSUNG

In dieser Arbeit wird ein Economic Model Predictive Controller für eine schwimmende Offshore-Windturbine vorgestellt. Der klassische Model Predictive Controller für schwimmende Offshore-Windenergieanlagen liefert vielversprechende Ergebnisse. Darüber hinaus hat die Forschung an Onshore-Windturbinen das Potenzial der Economic Model Predictive Controller aufgezeigt, die das Regelverhalten verbessern und den Regelungsentwurf im Vergleich zur klassischen Version dieser Regelungsmethode vereinfachen kann. Ziel dieser Arbeit ist die Entwicklung eines auf diesen beiden Forschungsergebnissen basierenden neuartigen Economic Model Predictive Controller für eine schwimmende Offshore-Windkraftanlage.

Als Grundlage für den Reglerentwurf dient ein vereinfachtes Modell niedriger Ordnung einer schwimmenden offshore Windenergieanlage. Unter Einbeziehung der Störgroßenvorschau und der Beschränkungen berechnet der Regler optimale Trajektorien für die Rotorblattanstellung und das Generatordrehmoment. Um die Regelungstechnik auf eine schwimmende Offshore-Windturbine anzuwenden, müssen zwei Dinge getan werden. Erstens ist die Kostenfunktion so auszulegen, dass sie die Hauptziele, nämlich die Maximierung der erzeugten Leistung und die Linderung der strukturellen Ermüdungserscheinungen, erfüllt. Zweitens werden die Beschränkungen in das Reglungsproblem integriert. Nach der Auswahl eines geeigneten Solvers wird der Regler diskretisiert und skaliert, so dass eine korrekte Implementierung und ein reibungsloser Betrieb möglich ist. Nach dem erfolgreichen Funktionieren des Algorithmus wird eine Mehrzieloptimierung durchgeführt, um geeignete Gewichte zu finden, so dass die Kostenfunktion an die geforderten Ziele angepasst ist.

Schließlich wird der entwickelte Regler mit realistischen Wind- und Wellenstörungen getestet. Es zeigt sich eine signifikante Reduktion der Standardabweichung der erzeugten Leistung bei gleichzeitiger Beibehaltung der Echtzeitfähigkeit. Darüber hinaus werden auch die strukturellen Ermüdungserscheinungen des Turms und der Plattform verringert.

Declaration of Authorship

I hereby certify that the master thesis entitled “Development of a Model Predictive Controller for Floating Offshore Wind Turbines” is entirely my own work except where otherwise indicated.

Passages and ideas from other sources have been clearly indicated.

Date: 2020-05-20
Signed: _____

Samuel Nann
Author

Contents

Nomenclature

Abbreviations

ACADO	Automatic Control and Dynamic Optimization
DEL	Damage equivalent loads
DOF	Degrees of freedom
EMPC	Economic Model Predictive Controller
ENMPC	Economic Nonlinear Model Predictive Control
EOG	Extreme operating gust
FAST	Fatigue, aerodynamics, structures and turbulence
FOWT	Floating offshore wind turbine
LIDAR	Light detection and ranging
LQR	Linear-quadratic Regulator
MHE	Moving horizon estimator
MIMO	Multi-Input-Multi-Output
MPC	Model Predictive Controller
NMPC	Nonlinear Model Predictive Control
ODE	Ordinary differential equation
SISO	Single-Input-Single-Output
SLOWf	Simplified low order wind turbine, floating
SLOWfLin	Linearized simplified low order floating wind turbine model
SQP	Sequential quadratic programming
STD	Standard deviation
VSC	Variable speed control

Symbols

β_p	Platform pitch
$\check{x}$	Update of the current state
$\Delta\widetilde{x}$	Scaled and linearized variable
$\Delta\widetilde{x}_{max}$	Scaled and linearized upper border
$\Delta\widetilde{x}_{min}$	Scaled and linearized lower border
Δu_c	Linerized control input vector
Δu_w	Linearized disturbance input vector

Δx	Linearized state vector
δx	Numerical derivation
Δx_{max}, Δu_{max}	Upper bound of the linearized state and linearzed input
Δx_{min}, Δu_{min}	Lower bound of the linearized state and linearzed input
$\dot{\beta}_p$	Derivation of the platform pitch
$\dot{\theta}$	Pitch angle velocity of the blades
$\dot{x}_p$	Derivation of the platform surge
$\dot{x}_t$	Tower top velocity
$\dot{z}_p$	Derivation of the platform heave
η	Wave height
η_{el}	Electrical efficiency
λ	Tip speed ratio
ω_0	Natural frequency
Ω_r	Rotor speed
ϕ	Azimuth angle of the rotor
ρ	Air density
σ	Standard deviation
θ	Pitch angle of the blades
θ_c	Desired blade pitch
$\widehat{x}$	Scaling factors
$\widetilde{A}_d$	Discrete scaled system matrix
$\widetilde{B}_{u,d}$	Discrete scaled control input matrix
$\widetilde{B}_{w,d}$	Discrete scaled disturbance input matrix
$\widetilde{x}$	Scaled variables
ζ	Damping ratio
A	System matrix
A_d	Discrete system matrix
B	Input matrix
B_u	Control input matrix
B_w	Disturbance input matrix
$B_{u,d}$	Discrete control input matrix
$B_{w,d}$	Discrete disturbance input matrix

c_P	Power coefficient
D_u	Scaling matrix of the inputs
D_w	Scaling matrix of the disturbances
D_x	Scaling matrix of the states
f_0	Eigenfrequency
f_s	Sample rate
$F_{wave,x}$	Wave force in x direction
$F_{wave,z}$	Wave force in z direction
H_s	Significant wave height
I	Identity matrix
i_{GB}	Gear box ratio
J	Moment of inertia
k	Current time step
k_{vsc}	Torque controller constant
l	Smoothing factor
M	Mass matrix of the multi body system
M_a	Aerodynamic torque
M_g	Generator torque
$M_{wave,y}$	Wave torque around y axis
P	Velocity matrix
P_g	Generated power
Q	Position matrix
q	Degrees of freedom
q_a	Degrees of freedom of the actuation
q_e	Degrees of freedom of the elastic bodies
q_r	Degrees of freedom of the rigid bodies
q_x	Weights in the cost function of states and inputs
R	Rotor radius
T	Shifting or tuning factor
T_0	Time constant
t_f	Prediction horizon
T_p	Significant time constant

T_s	Sample time
T_{sim}	Simulation time
u	Input vector
u_c	Control input vector
$u_{c,ss}$	Steady control input vector
$u_{w,ss}$	Steady disturbance input vector
v_0	Rotor effective wind speed
w	Disturbance input vector
x_k	Variable of time step k
x_p	Platform surge
x_t	Tower top displacement
x_{org}	Original state vector
x_{max}, u_{max}	Upper bound of the state and input
x_{min}, u_{min}	Lower bound of the state and input
x_{rated}, u_{rated}	Rated value of the state and input
x_{ss}	Steady state vector
z_p	Platform heave

List of Figures

List of Tables

1 Introduction

Renewable energies play a more and more important role in the global energy supply. With the already visible negative consequences, resulting from the high carbon emissions of the conventional energy producers, the part of the renewable energies in the global energy mix has to be increased. The strongest candidate of the renewable energies is the wind energy. The on- and offshore wind energy provide 52 % of the electricity generation of the renewable energies in the year 2019 in Germany [32]. To continuously reduce the carbon emission the wind energy has to be expanded further.

Currently, in Germany the further development of wind energy does not increase like a couple of years ago. Two exemplary reasons for this behavior is on the one hand the partly low public acceptance and on the other hand passed laws, which restrict the area where new wind turbines are allowed to be build. Out of this perspective, offshore wind turbines are more attractive. They take advantage of the higher and constant wind speeds. However they have to be built quite close to the shore, because the bottom fixed turbines cannot be installed anymore, as soon as the sea bed is too deep or has a too rough surface. Also, the costs are higher, in comparison to the onshore wind turbines. Therefore, floating offshore wind turbines (FOWTs) are more attractive. The turbines can be placed in far distance from the shore and in high water depths. Hence, the visual pollution is not existing anymore, the wind speeds are even more constant and higher. Additionally, a main advantage is the potential decreasing cost of FOWT [21]. Thus, there is a high potential in FOWT to provide cheap, clean and environmental friendly energy for the future. With floating offshore wind turbines new challenges are ahead, the dynamic of the turbines increases and get more complex. Out of this reason, new stabilizing and efficient energy generating wind turbines with the corresponding control system are required and have to be developed. This work presents a new control system for floating offshore wind turbines to tackle these new challenges.

1.1 Motivation and challenges

Set up offshore wind turbines on floating platforms comes along with additional, difficult to handle, dynamics. The loss of a bottom-fixed tower introduces new unwanted dynamics, which can lead to instabilities, if an inappropriate controller is used. The potential instability is visualized in figure 1. With increasing wind speed, the blade pitch will be increased, which leads to a decrease of the thrust. The resulting forward movement of the top of the wind turbine increases again the relative wind speed at the rotor. Therefore, an unstable oscillation can occur. This avoidable phenomenon is the result of a too aggressive controller. Because the system has a non-minimum phase or in other words a right half plane zero. Hence the controller is limited in the bandwidth.

Another difficulty is the integration of the disturbances. The first disturbance, namely the wind, is the energy source of wind turbines. Without this disturbance no turbine would generate energy. The second disturbance are the waves. Both disturbances can only be measured at the wind turbine itself and thus, without additional sensors, no disturbance preview is possible.

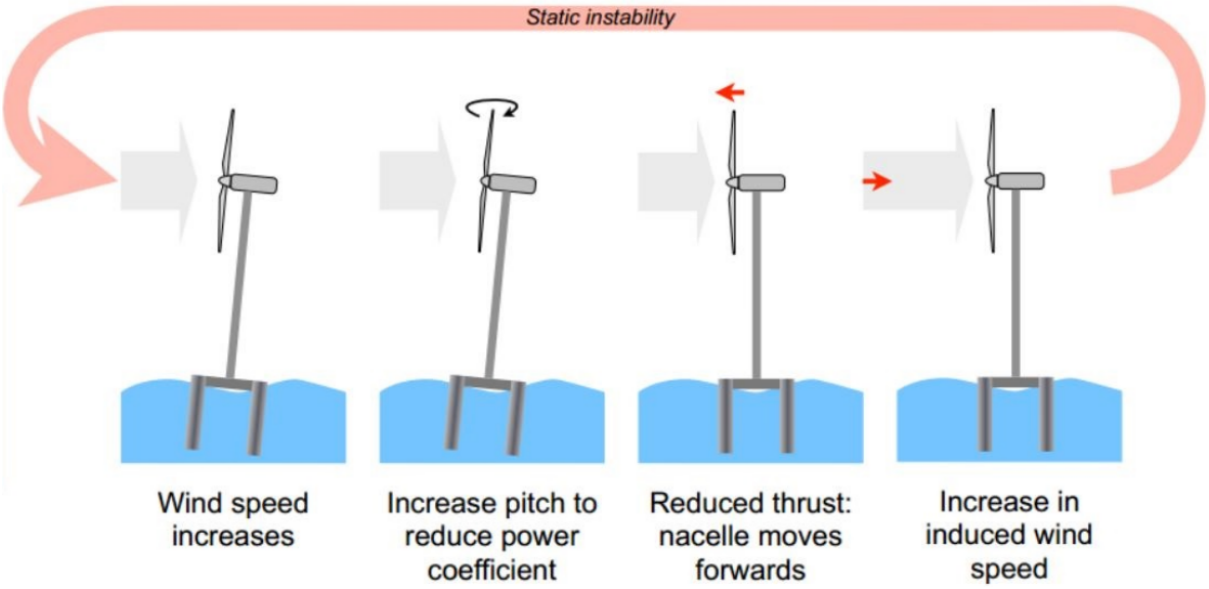


Figure 1: Instability problem of floating offshore wind turbine [33].

An additional challenge is the compliance of the constraints. Like in most control systems, there are hardware limitations, which has to be considered. For FOWTs the parts in the nacelle, like the generator, have to be protected. Also, the structural loads of the tower and the rotor blades should be under a certain limit. Additionally, there could be introduced stability constraints for certain motions. All these challenges have to be fulfilled of the controller. In the next part the usual controller and their properties are explained.

The standard controller for FOWT is made of a nonlinear torque controller below the rated wind speed and above rated of a PI controller. Especially, the PI controller has to be detuned [8] to avoid the negative damping problem (figure 1). This means, that the eigenfrequency of the certain wind turbine dynamic has to be chosen lower than the eigenfrequency of the platform pitch. So far, the presented controller is a Single-Input-Single-Output (SISO) controller. Another approach to tackle the instability problem is to use Multi-Input-Multi-Output (MIMO). Thus, not only a single loop, like from the rotor speed as measurement to the blade pitch as actuator, is applied, but several loops. Therefore the controller is able to control both inputs, the generator torque and the blade pitch, simultaneously. With more than one measurement the controller decides the control actuation on the basis of a coupled dynamic.

Also advanced control concepts are already applied by other researches on floating offshore wind turbines. One strategy is to use optimal control, like the Linear Quadratic Regulator (LQR), which is presented in [18]. Also robust controller like H_2 and H_∞ approaches are tested [29].

This work is focused on a specific type of a Model Predictive Controller (MPC). Thus, this special research field is examined more closely in the next paragraph.

1.2 State of the art

MPC is a strong candidate for floating offshore wind turbines. With MPC, the above stated challenges can be tackled. The integration of a disturbances preview is possible and the con-

straints can be considered with this control scheme. In [26], a Nonlinear Model Predictive Control (NPMC) for floating wind turbines is presented. Perfect preview and knowledge about the full state is assumed. The results are promising, significant reductions in the variations of the rotor speed and the generated power are obtained. Also loads concerning the tower and on the blades are decreased. Additionally, with NMPC, the nonlinearities can be treated directly. It is possible to implement the model unlinearized. However, MPC requires a high computational effort, which results in a non-real time-capable controller in this paper. In [5], a MPC with a state estimator is presented. The disturbance preview is done with Light detection and ranging (LIDAR). A closer look on the disturbance preview with LIDAR for FOWTs is described in [25]. NMPC can also be combined with individual pitch control, which shows a significant blade fatigue load reduction. However, the so far presented classical MPC and classical NMPC is restricted to a quadratic cost function. The formulation of the maximization of the generated energy is, with this restriction, only in an indirect way possible. Also the general control design is made up of a lot tuning parameters and thus arbitrarily hard to find appropriate parameters for the desired performance.

With a special type of MPC, namely Economic Model Predictive Control (EMPC), the just now described downsides, are easier to handle. EMPC is presented and introduced as a strong candidate to control offshore WT in [9]. Additional to the presented benefits of NMPC, the extension of NMPC to EMPC involves further advantages like, that the cost functions are directly measures of the system performance, thus it maximizes the performance directly and potentially increases the closed loop performance. Secondly, the possibility of integrating the power maximization objective directly as a cost function as well. These points ease the difficulty of tuning the weights.

The successive works [10] and [11] treat the problem of real time feasibility with the interpolation of the aerodynamic coefficient, e.g. with B-splines. Another potential is also presented in [11]. With Economic Nonlinear Model Predictive Control (ENMPC) it is possible to develop monetary models of actuator activity or tower fatigue. This allows a formal and direct comparison to the energy generation and would yield to a truly tuning-free control. However, it has not yet been developed. This is a very strong potential, which is possible with EMPC. Also, robustness analysis is done. Errors in the tower main eigenfrequency, does not impact significantly the performance of the closed loop scheme.

Another approach to handle the nonlinearities of the aerodynamic coefficient, is to use state transformation, so that the resulting description is convex. This concept is also for EMPC for onshore wind turbines. The same goal of maximize the total generated energy while considering constraints in the variation of the delivered power to the grid, is described in [14]. It is used the turbine inertia as an additional electrical storage device. The nonlinear system dynamic, the non-convex constraints and objectives describe a difficult to solve problem. A transformation of the problem to one with linear dynamics and convex constraints is presented. The trick is to rewrite the formulas with power flows and energies. Simulations reveal an ability to reject disturbances, ensuring the variation of the delivered power to the grid and keeping the amount of produced power very close to nominal.

A LIDAR-assisted ENMPC framework is presented in [27], where the goals are again the maximization of the energy capture and mitigating the tower fatigue loads. The formulation of a convex optimal control problem is introduced, with linear dynamics and convex constraints that

can be solved globally, like in [14]. The trick is again to work with power flows and energies. E.g. the available power, which depends on the power coefficient, can be approximated by concave functions depending in the kinetic energy and the wind velocity. Thus, there exists a function for each wind speed with relatively small error. The overall approximation works with a linear interpolation of two concave functions. Regarding the tower dynamic, at each time-step, the nonlinear thrust force will be replaced with a linear approximation. In [28] there is an extension of his previous work with an adaptive algorithm to compensate plant mismatches and provide the optimal tip-speed ratio.

All the EMPC concepts describe the methodology for onshore wind turbines. So far, with the best knowledge of the author, no EMPC for FOWT is developed so far.

1.3 Objectives

In the last section the state-of-the art for standard SISO controller, for MIMO controllers and advanced optimal control types like MPC are described. Finally, and most important for this work, the current EMPC state of research is presented. However, the depicted EMPC elaboration is designed for onshore wind turbines. In this work, the main objective is to develop and analyze an EMPC and the corresponding control design for FOWTs. The focus lies on first implementations and a selection of an appropriate solver for the controller. Thus, the controller of this work is not designed for the whole operation range, but rather for one operation point and the associated linearization. The difficulties of the nonlinearities of the aerodynamic coefficients and possible solutions do not occur in this work.

The first adaption of a model for FOWT to a solver environment is done. Further modifications of the model, to tackle occurring problems concerning the feasibility and the convergence of the algorithm, are shown. Cost functions for different use cases are stated and compared to each other, regarding to defined objectives. Each cost function has a different focus on respective objectives. The results are compared to a standard baseline controller. Thus, the potential and usability of the EMPC control concept for floating wind turbines is explored. It is also tried to achieve real time capability with the developed optimization problem and the chosen solver. The simulations are done with the simplified low order model.

1.4 Outline

In the next chapter, the modeling part is presented. The reference turbine is defined and a short introduction of model with its individual modeling disciplines are given. The required linearization for the later internal control model is presented. Also, the cases with the individual disturbances are defined. Finally, the control objectives by which the result of the controllers are measured.

The third chapter introduces the standard baseline controller with its benefits and downsides. This controller is later used for the scaling of the internal model and is taken as the reference controller to compare with. Then the Linear Quadratic Regulator (LQR) as basic optimal controller is presented. This controller is on the one hand an introduction of the later following controller and on the other hand a benchmark example, to show the potential of optimal controller. After this, the main control class, MPC is introduced, again with its advantages and disadvantages. In the end, the EMPC, where the focus lies on, is presented. The EMPC is adapted

to the FOWT, thus the cost function and the constraints are with respect the floating offshore wind turbines models defined. The optimization problem is also adjusted for the linear model.

In the fourth chapter different solver types are presented. Also, the necessary discretization of the model is conducted. Then the required scaling of the model is described. This serves the purpose of the feasibility and convergence of the respective algorithm.

The fifth chapter covers the results of the simulations. Firstly, the general results are presented, analyzed and compared with the baseline controller. Secondly, imperfect disturbance preview is tested as well.

The last chapter concludes this work and gives an outlook of different topics to research on.

2 Modeling

To develop a new control type, in this case a specific MPC, an *internal* model for the controller itself and a *simulation* model is required. The model is made up of a wind turbine and a platform, or rather floater, where the wind turbine is being build. The model has to consider all the different dynamics acting on the floating wind turbine. Beginning with the hydrodynamic of the floater, where the waves and the mooring lines have an impact, over the structural dynamic of the tower and the rotor blades, to the aerodynamic of the wind and the servo dynamic in the nacelle. In this work a reduced model is used, which is sufficient for controller design.

In this chapter, the used reference wind turbine inclusive the floater is defined. Secondly, an introduction and description into the reduced conceptual model used in this work is given. Thirdly, the two disturbances, wind and waves, are defined. Fourthly and lastly, the comparison between the linearized and the origin model is done.

2.1 Definition of the floating wind turbine

This section introduces the used wind turbine and the used platform with the individual design parameter.

2.1.1 Wind turbine

The wind turbine model is the DTU 10 MW Reference Wind Turbine from the DTU Wind Energy, Technical University of Denmark [2]. The rated power of the turbine is 10 MW. It is a usual upwind and three-bladed turbine. The rotor has a diameter of 178.3 m and the hub height is 119 m. The rated wind speed amounts 11.4 m/s, where the rotor speed runs with 9.6 RPM. These described design values and a few more are collected in table 1.

Table 1: The DTU 10 MW reference wind turbine design values.

Description	Unit	Value
Rated power	[MW]	10
Rotor diameter	[m]	178.3
Hub height	[m]	119
Cut-in wind speed	[m/s]	4
Rated wind speed	[m/s]	11.4
Cut-out wind speed	[m/s]	25
Cut-in rotor speed	[RPM]	6
Rated rotor speed	[RPM]	9.6
Rotor mass	[tons]	229
Nacelle mass	[tons]	446
Tower mass	[tons]	605
Electrical efficiency η_{el}	[-]	0.94
Gear box ratio i_{GB}	[-]	1/50

2.1.2 Platform

The present floater is the LIFE50+ OO-Star Wind Floater Semi 10 MW [34]. The semi-submersible star-shaped base pontoon, which is made up of concrete, connects three outer columns with the inner column, where the wind turbine is placed on. This pontoon is connected to the seabed with three catenary mooring lines. The concept is shown in figure 2. A summary of the design values is listed in tabel 2 [22].

Figure 2: The OO-Star Wind Floater Semi 10 MW concept [1].

Table 2: Summary design values of the OO-Star floater.

Description	Unit	Value
Water depth	[m]	130
Mooring length	[m]	703
Mass incl. ballast	[tons]	21 709
Distance outer column to center column	[m]	37

2.2 Reduced order nonlinear model

The described model is called SLOWf, which is an abbreviation for *Simplified low order wind turbine, floating*. It is an already developed model [17], which is, as the name suggests, a reduced conceptual model. The cases of application are simple and fast simulations or controller design, like it is required in this work. The following introduction describes the general set up and the individual dynamics.

2.2.1 General set up of the model

The floating wind turbine is divided into three rigid bodies, namely the platform, the nacelle and the rotor. Consequently one assumption is, that the rotor is modeled as a rigid disk. Additionally to these rigid bodies, there is one flexible body, the tower. The focus of SLOWf is on the most significant degrees of freedom (DOF). Also the model is developed in 2D, in the x-z-plane. Where the positive x-axis is defined downwind and the z-axis points upwards. Thus the first four DOFs q_r results from the rigid bodies as platform surge x_p, platform heave z_p, platform

pitch β_p and the rotor azimuth ϕ. The fifth DOF q_e describes the tower top position x_t of the elastic tower and the sixth one q_a is the blade pitch of the pitch actuator θ, see figure 3. All combined are defined by

$$q = \begin{pmatrix} x_p \\ z_p \\ \beta_p \\ \phi \\ x_t \\ \theta \end{pmatrix}, \quad q_r = \begin{pmatrix} x_p \\ z_p \\ \beta_p \end{pmatrix}, \quad q_e = \begin{pmatrix} x_t \end{pmatrix}, \quad q_a = \begin{pmatrix} \theta \end{pmatrix}. \tag{2.1}$$

The state space results in the combination of the all the DOFs and their derivatives

$$x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} = (x_p,\, z_p,\, \beta_p,\, \phi,\, x_t,\, \theta,\, \dot{x}_p,\, \dot{z}_p,\, \dot{\beta}_p,\, \Omega_r,\, \dot{x}_t,\, \dot{\theta})^T, \tag{2.2}$$

where the rotor speed is denoted with $\dot{\phi} = \Omega_r$. Thus the model has 12 states.

2.2.1.1 Control and disturbance inputs

The system has two control inputs. The first one is the generator torque M_g and the second one the desired blade pitch θ_c, which is not the measured blade pitch θ. There is an additional dynamic between θ_c and θ, described in more detail below. Thus the resulting control input vector reads

$$u_c = (M_g,\, \theta_c)^T. \tag{2.3}$$

Furthermore, there are two disturbances, or rather four disturbance inputs. The first one is the one dimensional rotor effective wind speed v_0, which is the simplification of a complex three dimensional wind field. The other three disturbances are a result of the wave height η, which is only an indirect disturbance. The wave height η leads to three direct disturbance inputs, two generalized forces and one generalized torque, affecting the platform. They are all dependent on the wave height η through a wave disturbance model. The resulting disturbance input vector follows to

$$w = (v_0,\, \eta)^T = (v_0,\, F_{wave,x}(\eta),\, F_{wave,z}(\eta),\, M_{wave,y}(\eta))^T. \tag{2.4}$$

Figure 3 illustrates the floating wind turbine, with its floater, the elastic tower, the nacelle and the rotor. Also marked are the six degrees of freedom q, the two control inputs $u_c = (M_g,\, \theta_c)^T$ drawn in red and the two disturbances $w = (v_0,\, \eta)^T$ marked in blue.

The overall input vector u is merged of the control input vector u_c and the disturbance one w

$$u = (u_c,\, w)^T. \tag{2.5}$$

2.2.2 Modeling

The reduced order model is designed by using the Euler-Newton equations, which lead to ordinary differential equations (ODE). This system of equations can be transformed into the state space formalism

$$\dot{x} = \frac{\partial x}{\partial t} = \begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ M^{-1}(p - k) \end{pmatrix} = f(x, u_c, w), \tag{2.6}$$

where the matrix M denotes the mass matrix of the multi body system. More details regarding the modeling can be found in [17].

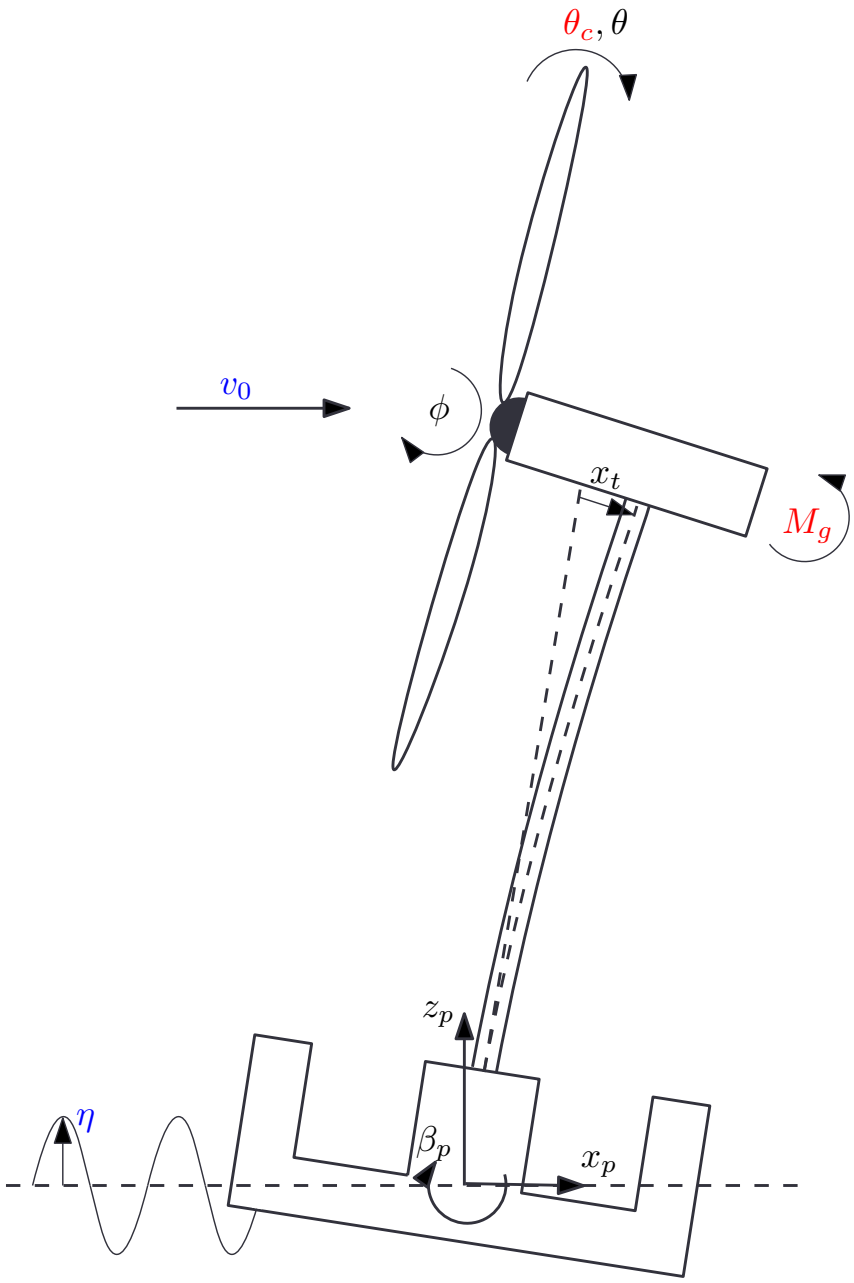


Figure 3: Floating offshore wind turbine model with six DOFs $q = (x_p, z_p, \beta_p, \phi, x_t, \theta)^T$ plus two inputs $u_c = (M_g, \theta_c)^T$ and two disturbances $w = (v_0, \eta)^T$.

Table 3: Identified parameter of the 2nd-order system, which models the pitch actuator dynamics.

Parameter	Unit	Value
f_0	[Hz]	1.6
$\omega_0 = 2\pi f_0$	[rad/s]	10.053
ζ	[-]	0.8

The aerodynamics are modeled through a quasi-static representation of the power and thrust coefficient, a two dimensional look up table for each coefficient, depending on the tip-speed ratio λ and the blade pitch θ. Also concerning the aerodynamics, like above mentioned, the disturbance input is only a scalar rotor effective wind speed, as a simplification of a more complex wind field.

Each of the three catenary mooring lines affects the platform with a horizontal and a vertical force as a function of the floater position. The hydrostatic is calculated via potential flow. There is no radiation damping applied, instead a lumped linear damping matrix.

2.2.3 Pitch actuator

The actuator dynamic of the rotor pitch is modelled as a 2nd-order system. The property of the dynamic is a delayed and oscillating behaviour as it is observed on the real rotor pitch actuator. This element is described by the natural frequency ω_0 and the damping ratio ζ. The generic differential equation of such an element looks like

$$\begin{aligned} \frac{1}{\omega_0^2}\ddot{\theta} + \frac{2\zeta}{\omega_0}\dot{\theta} + \theta &= \theta_c \\ \Leftrightarrow \ddot{\theta} + 2\zeta\omega_0\dot{\theta} + \omega_0^2\theta &= \omega_0^2\theta_c. \end{aligned} \tag{2.7}$$

The identified parameters for the pitch actuator dynamic are listed in table 3.

2.3 Linearization

The linearized model is needed, on the one hand for different analyses of the system itself and on the other hand for the later presented control concept. The analysis and identifications are around an exemplary operation point with a wind speed v_0 of 16 m/s. With this set point, the wind turbine is in the above rated wind speed region, thus in the full load case. The steady states of this operation point are shown in table 4. These values are the result of a closed loop simulation with SLOWf as simulation model and the default baseline controller. This controller is so far the state of the art controller and is often used as a reference controller. It is described later in depth. All derivatives $\dot{q}$ are equal zero, except of the rotor speed Ω_r, which is also in the steady state case unequal zero. The steady state of all twelve states are combined in the vector x_{ss} , the two control inputs in $u_{c,ss}$ and the four steady states of the disturbance inputs are merged in $u_{w,ss}$. With these three shifting vectors the absolute variables are separated into

$$x = \Delta x + x_{ss}, \quad u_c = \Delta u_c + u_{c,ss} \quad w = \Delta w + u_{w,ss}, \tag{2.8}$$

where Δx, Δu_c and Δu_w are the new vectors of the linearized state around the operation point. The above described state space formulation in equation 2.6 can be linearized, resulting in the

Table 4: Steady states of the chosen operating point of v_0 = 16 m/s.

State/Input	Unit	Value
x_p	[m]	12.4824
z_p	[m]	0.0179
β_p	[deg]	3.1355
ϕ	[deg]	-
x_t	[m]	0.1193
θ	[deg]	12.6923
$\dot{x}_p$	[m/s]	0
$\dot{z}_p$	[m/s]	0
$\dot{\beta}_p$	[deg/s]	0
Ω_r	[rpm]	9.5989
$\dot{x}_t$	[m/s]	0
$\dot{\theta}$	[deg/s]	0
M_g	[Nm]	2.1164×10^5
θ_c	[deg]	12.6923
v_0	[m/s]	16
$F_x(\eta), F_z(\eta),$	[N]	0
$M_y(\eta)$	[Nm]	0

linear differential equations of the multi body system in the state-space formulation

$$\Delta\dot{x} = \underbrace{\begin{pmatrix} 0 & I \\ -M^{-1}Q & -M^{1}P \end{pmatrix}}_{A} \Delta x + B\Delta u \tag{2.9}$$

where I represents the identity matrix, Q the position matrix, P the velocity matrix, A the system matrix and B the complete input matrix. The input matrix B can be split into the control inputs and into the disturbance inputs like

$$\Delta\dot{x} = A\Delta x + B_u\Delta u_c + B_w\Delta w, \tag{2.10}$$

where B_u denotes the control input matrix and B_w the disturbance input matrix. The resulting linear model is called SLOWfLin. With this model, it is possible to calculate eigenvalues, natural frequencies and do further researches on the system dynamic. Table 5 summarizes all the system properties of the linearized system model SLOWfLin. The resulting pole of every DOF is written in the second column, the corresponding natural frequency ω_0 , the damping ratio ζ, the eigenfrequency f_0 and the time constant T_0 are presented in the remaining columns. The slowest dynamic has the platform surge and the pitch actuation has the fastest.

2.4 Load cases

There are two different disturbances, which affect the FOWT. The first one is the wind and the second one are the waves. In the following subsections, the load cases are introduced and described.

2.4.1 Wind

The wind is a one dimensional rotor effective wind vector. In this work there are two different types of wind models used. The first one is a stochastic one and the second a deterministic

Table 5: System properties of the linear model SLOWf. The poles of each DOF q, the natural frequency ω_0, the damping ratio ζ, the eigenfrequency f_0 and the time constant T_0.

DOF q	Pole	ω_0 [rad/s]	ζ	f_0 [Hz]	T_0 [s]
Surge x_p	-0.0024 + 0.0405i	0.0406	0.0593	0.0065	154.6849
Heave z_p	-0.0115 + 0.3019i	0.3021	0.0380	0.0481	20.7965
Pitch β_p	-0.0317 + 0.1670i	0.1700	0.1864	0.0271	36.9570
Rotations ϕ	-0.1652 + 0.0000i	0.1652	1.0000	0.0263	38.0313
Tower x_t	-0.1053 + 4.4492i	4.4505	0.0237	0.7083	1.4118
Blade pitch θ	-8.0425 + 6.0319i	10.0531	0.8000	1.6000	0.6250

wind disturbance model.

Firstly, the stochastic wind disturbance is described. The used wind disturbance is a turbulent wind field of class C with an average rotor effective wind speed of $v_0 = 16\,\mathrm{m/s}$. The disturbance data [15] are from the LIFES50+ project [20], funded by the European Union Horizon2020 program. In figure 4 the time behaviour and the corresponding power spectral density (PSD) are shown. It is clearly visible, that the most significant power density is at $f = 0\,\mathrm{Hz}$, which repre-

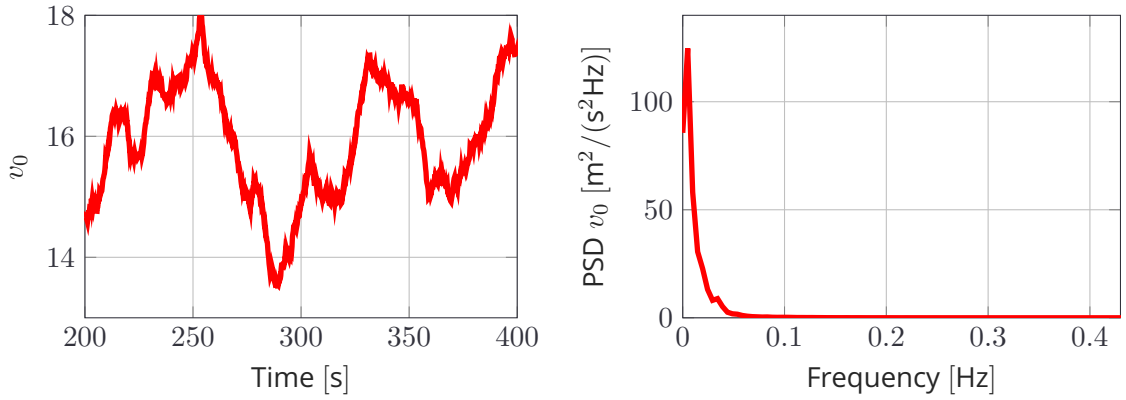


Figure 4: Exemplary time behaviour and power spectral density (PSD) of the wind disturbance with v_0 = 16 m/s

sents the constant part of the wind disturbance. The power density for higher frequencies is continuously decreasing.

Secondly, the deterministic wind disturbance is introduced. A deterministic wind model is a predefined wind field. The chosen parameter are e.g. the mean wind velocity and its direction, altitude profile, tower shadow or wind shear. An example for such models is the wind gust, also called extreme operating gust (EOG) , which is used as the deterministic model in this work. The used EOG is plotted in figure 5. The used average wind speed is 16 m/s. Before and after the main peak, which has a top wind speed of almost 21 m/s, the speed decreases to approximately 14.5 m/s.

2.4.2 Waves

The used wave disturbances are irregular waves with a wave height $H_s = 3.534\,\mathrm{m}$ and a peak spectral period $T_p = 7.289\,\mathrm{s}$. The data also comes from the LIFES50+ project, where a Pierson-Moskowitz spectrum is used [22]. These chosen values fit to the described wind speed of $v_0 =$

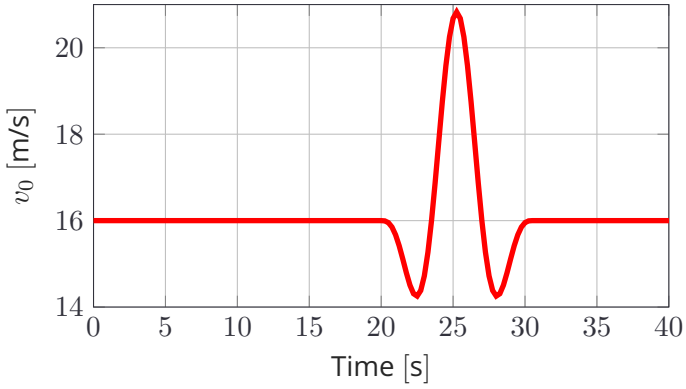


Figure 5: Exemplary deterministic wind model: Extreme operating gust with v_0= 16 m/s and start time at 20s.

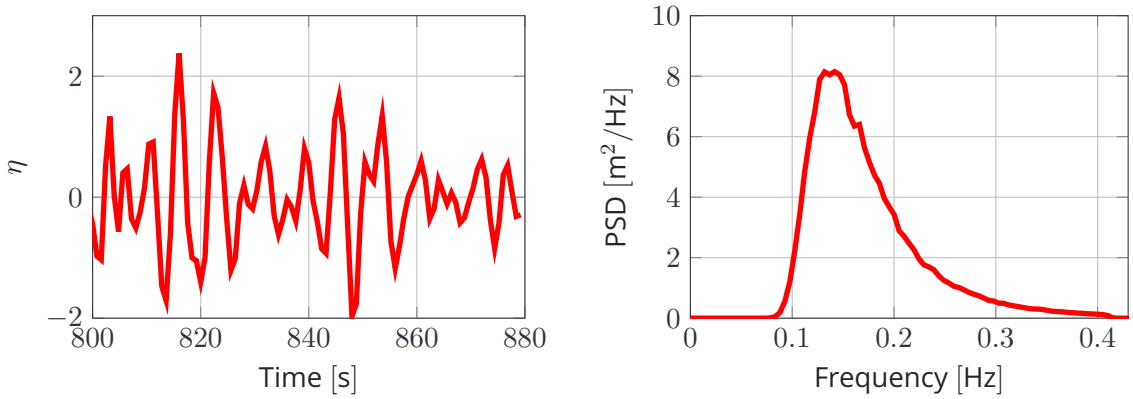


Figure 6: Exemplary time behaviour and power spectral density (PSD) of the wave disturbance with H_s = 3.534 m and T_p = 7.289 s.

$16\,\mathrm{m/s}$. Wind and waves are correlated. Figure 6 shows the time series and the corresponding PSD. On the left hand side, the time series is plotted, whose range is mostly between plus and minus 2 m wave height. The above mentioned time constant leads to a natural frequency of $f_p = 0.137\,\mathrm{Hz}$, which is clearly visible in the right plot of the figure.

2.5 Comparison SLOWfLin and SLOWf

To verify the derived linear SLOWfLin model, both simulation models are tested in closed loop with the same default baseline controller, see section 3.1.

Two different test simulation are shown. For the first test, it is used a deterministic wind gust, an EOG, like described above. This is the only disturbance, there is no wave disturbance acting on the FOWT. The second test contains stochastic wind and stochastic waves as disturbances, also like mentioned above.

2.5.1 Simulation configuration

Both following simulations are done with MATLAB and Simulink 2018a (64bit). The sample rate f_s is chosen to 40 Hz, which is a sample time T_s of 0.025 s. The length of the simulation differs

and is described below.

2.5.2 Deterministic disturbance model: wind gust

The first case is an EOG as deterministic wind model and no waves are active. The simulation time T_{sim} amounts 400 s. The result of this comparison is shown in figure 7. The upper two diagrams display the disturbances, which are, as expected, identical. The next three plots, three to five, shows the three DOFs of the platform, the surge x_p, the heave z_p and the pitch β_p. The heave and the pitch time course are almost the same, the eigenfrequency, in which the individual variable is decreasing, fits. The same holds for the platform surge. During the first 80 s is no mentionable difference. In the remaining progress, the linear simulation has a lower surge, whereas the oscillations are the same. Thus, the surge has only a small offset of a couple of centimeters. The next two diagrams, plot six and seven, display the flexible tower top position x_t and the rotor speed Ω_r. The main dynamic occurs during the first 50 s, where the two time courses fit together, despite the fast changes. After 100 s there is almost no motion left, the steady states are also identical. The third- and second-last two plots show the two control inputs. The generator torque M_g is constant, because the FOWT is operated above rated. The blade pitch θ_c is also identical, however it is not surprisingly, when all the previous states of the system are similar, the computed control inputs cannot be different. The last plot shows the generated power P_g. In summary, the linear model SLOWfLin contains all the important dynamics and provides almost identical simulation results as the nonlinear SLOWf model in account of deterministic wind disturbances.

2.5.3 Stochastic disturbance model

The second test, to verify the linear model, is done with both disturbances, wind and wave. Both disturbances are stochastic. The simulation time is chosen to one hour (3600 s). The length is chosen so that it is possible to analyze whether there are long time phenomenon like slow drift or not. However, for the reason of clarity only a time section of 200 s, started at 700 s, is shown in the figure 8. The order of the plots are the same as in the previous test. Firstly, two disturbances plots, which are of course identical for both simulation models. Secondly, the three DOFs of the platform, where all three time courses match almost perfectly to the corresponding simulation result. Small differences in the platform surge and pitch, but especially the heave motion is identical. Then, the tower top displacement, there is also no visible distinction. The same applies to the rotor speed. Of course the generator torque is for both simulations constant with no difference. The linear controlled blade pitch follows also most of the course the nonlinear simulation result, only small differences are visible. In summary, also the stochastic disturbances test contains all the important dynamics and provides nearly identical simulations results. Also with longer simulation times, no results are drifting apart of each other.

2.5.4 Result

The simulations for both load cases provide almost identical results. The differences are not significant. Thus the linear SLOWfLin model is a good surrogate model of the nonlinear SLOWf model. Consequently it is a good candidate as a simulation model for later closed loop simulations and also a suitable internal model for any controller. These results hold only for a certain range around the linearized operation point of $v_0 = 16\,\mathrm{m/s}$. Both previous test cases are run with disturbances around the operation point. The EOG, the stochastic wind and waves are

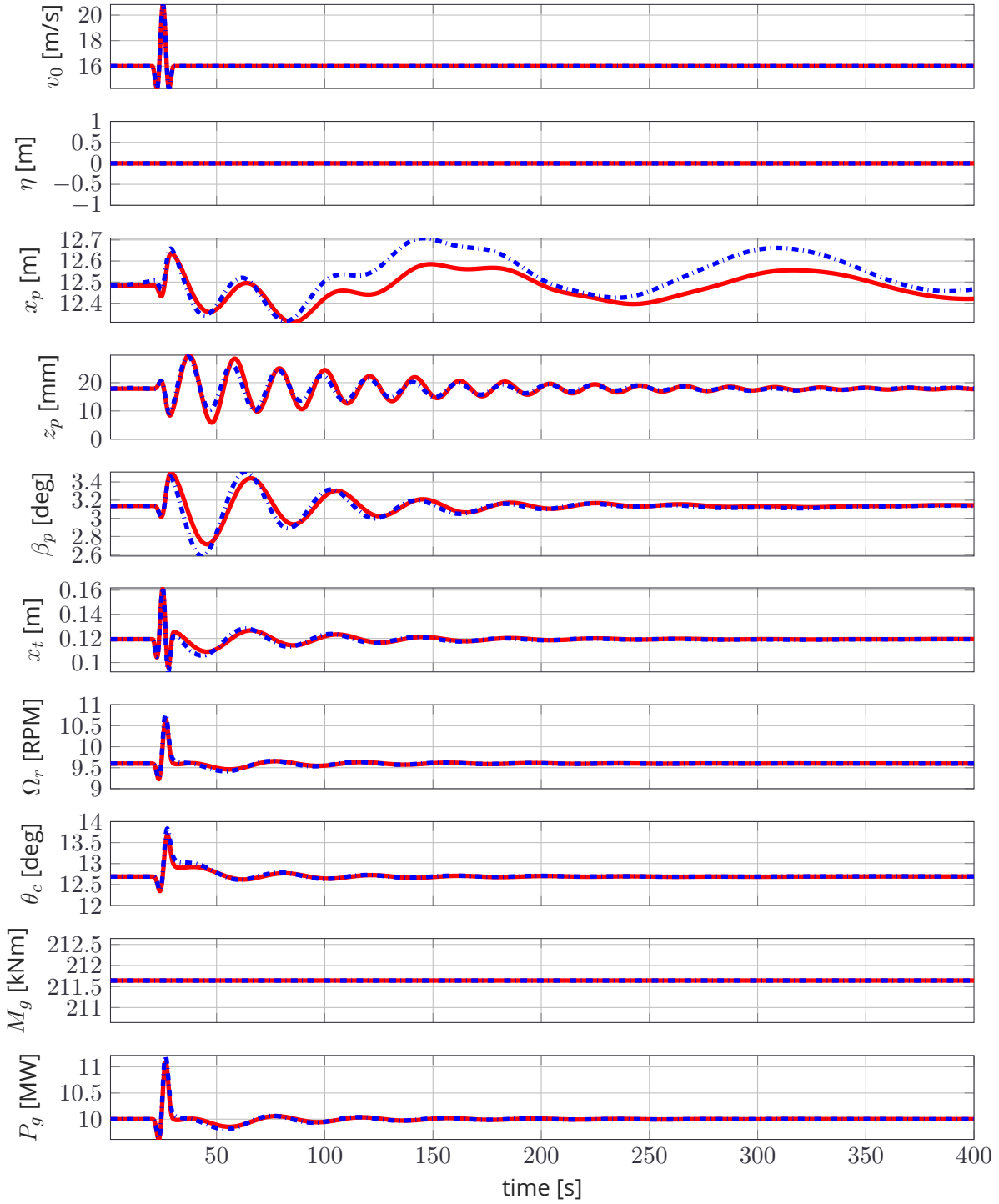


Figure 7: Comparison of the SLOWfLin and the SLOWf model with a wind gust as disturbance. The response of the linear SLOWfLin model illustrated with (—) and the nonlinear model response with (- -).

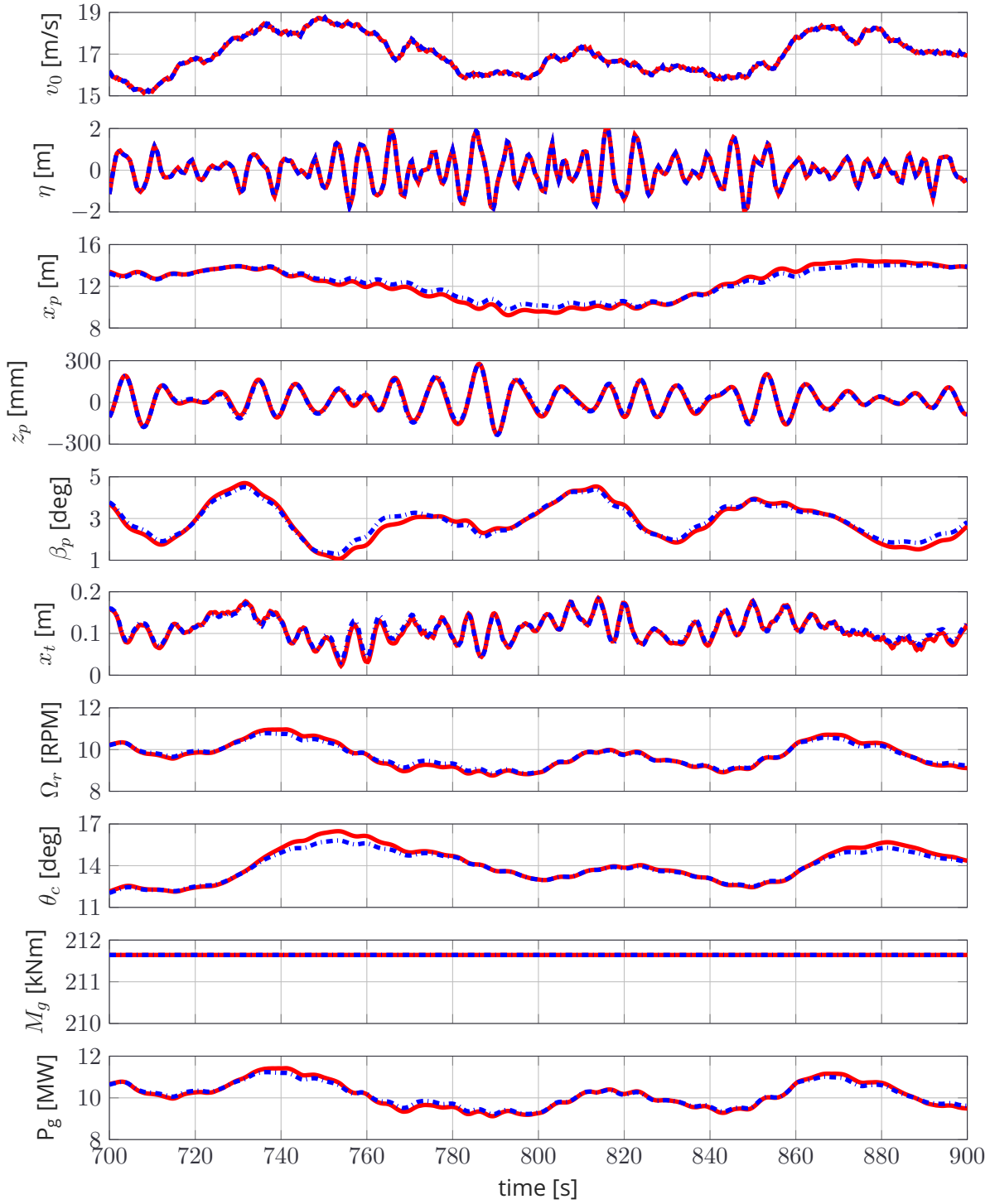


Figure 8: Comparison of the SLOWfLin and the SLOWf model with stochastic wind and wave disturbances. The response of the linear SLOWfLin model illustrated with (—) and the nonlinear model response with (- -).

Table 6: Performance targets

Objective	Unit	Formula
Maximize generated power	[MW]	$mean(P_g)$
Minimize power variation	[MW]	$\sigma(P_g)$
Minimize tower top variation	[m]	$\sigma(x_t)$
Minimize platform pitch variation	[deg]	$\sigma(\beta_p)$
Minimize generator torque actuation	[kNm]	$\sigma(M_g)$
Minimize blade pitch actuation	[deg]	$\sigma(\theta_c)$

Table 7: Performance requirements

Requirements	Unit	Formula
Restrict overrated power	[MW]	$P_g \leq P_{g,max}$
Restrict overrated rotor speed	[RPM]	$\Omega_r \leq \Omega_{r,max}$
Protects the generator	[Nm]	$M_g \leq M_{g,max}$

all designed for the mentioned operation point. Thus the linear model is only valid for these disturbances. All further simulations are done with these defined disturbances.

2.6 Control objectives

To compare further simulation results in account of the controller performance and controller objectives, there have to be defined criteria first. The general objectives of a wind turbine are

1. Maximizing generated power
2. Minimize structural fatigue loads (tower top and platform fore-aft motion)
3. Minimize variation in the generated power

The transformation of these aims into mathematical formulations for the controller can be arbitrarily hard. The transformation for one specific control type is documented in this work. However, like already mentioned, to rank the controller performance, there have to be criteria, which make an evaluation possible. Table 6 shows all the targets. The first part is motivated by the energy production and stability, the second part by the reduction of the structural fatigues and the third part by the actuation load. The standard deviation (STD) σ is used to measure the individual targets. Using the two targets of the second part to measure the structural fatigues is only an indirect way, but a straight forward one. If higher dimensional models are used for the simulation, a more realistic way to compute the structural fatigue loads, is by calculate the damage equivalent loads (DEL). In this work the defined objectives, $\sigma(x_t)$ and $\sigma(\beta_p)$, are used to represent the structural loads. The third part estimates the actuation loads, whereas the focus is on the STD of the blade pitch actuation, due to the higher wear. Additionally to these performance targets there are requirements, which must be fulfilled of the closed loop control system. All these specifications are listed in table 7. The requirements are protective measures for the generator and all connected electronics. The introduced maximal values of the three

requirements are defined to 10% above the rated value to

$$\begin{aligned} P_{g,max} &= 1.1 \cdot P_{g,rated} = 11\,\mathrm{MW} \\ \Omega_{r,max} &= 1.1 \cdot \Omega_{r,rated} = 10.56\,\mathrm{RPM} \\ M_{g,max} &= 1.1 \cdot M_{g,rated} = 232.8064\,\mathrm{kNm} \end{aligned} \tag{2.11}$$

All the required models, validations, targets and requirements are now defined and ready to use for the control design, which is covered in the next chapter, and the resultant simulations.

3 Control design

This chapter starts with the usual used standard controller in wind turbine control, particularly the baseline controller, with its advantages and downsides. This controller is used for scaling and comparing other controllers in this work. After this basic controller, the proximately following controllers belong to the optimal controller class. Starting from the linear-quadratic regulator (LQR), which introduces the class of the optimal controller and forms the bases for the following classical MPC. This chapter finalizes with the Economic Model Predictive Controller (EMPC), on what the focus lies in this work.

3.1 Baseline controller

The baseline controller is build up of two parts. Firstly, the torque controller, which operates in the partial load region, in other words, when the wind is below the rated wind speed. Secondly, the pitch controller, which acts in the full load region, when the wind is above the rated wind speed. In addition to these two controllers, there exists a switching mechanism, thus that the overall controller is able to switch smoothly between these two basic controllers.

The torque control, also called variable speed control (VSC) [16], for the below rated region is a nonlinear feedback controller depending on the generator speed Ω_g and acts according to the following law

$$M_g = k_{vsc}\Omega_g^2, \tag{3.1}$$

where k_{vsc} is a constant to maximize the generated power P_g. This is reached, when the power coefficient c_P is equal to its optimum $c_{P,opt}$. Because the VSC takes place below rated, the corresponding pitch angle θ is zero. Thus, the resulting optimal generated power looks like

$$P_{g,opt} = \frac{1}{2}\rho\pi R^2 c_{P,opt}(\lambda_{opt}, 0)v_0^3\eta_{el}, \tag{3.2}$$

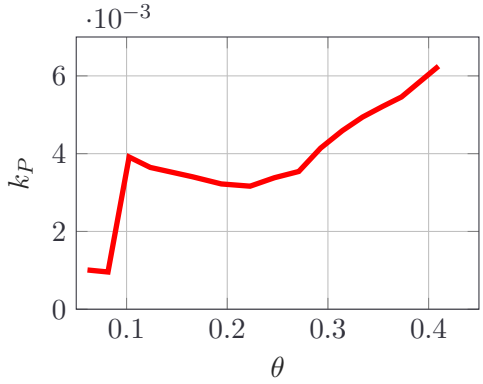
where ρ is the air density, R the rotor radius, λ is the tip speed ratio. Hence, this controller aims to track the optimal tip speed ratio λ_{opt}. The constant parameter k_{vsc} is derived of the equilibrium of the rotor speed equation

$$J\dot{\Omega}_r = M_a - \frac{M_g}{i_{GB}} = 0. \tag{3.3}$$

After inserting the equation for the aerodynamic torque M_a in the above presented equilibrium and converting it to the generated torque M_g, the parameter reads

$$k_{vsc} = \frac{1}{2}\rho\pi R^2 \frac{c_{P,opt}}{\lambda_{opt}^3} i_{GB}^3. \tag{3.4}$$

The second basic controller, which operates in the above rated wind speed region is a gain-scheduled PI anti-windup controller used for pitch actuation. The gain-scheduled part of the controller is referred to the proportional parameter k_P, which depends on the blade pitch θ. The result of this gain-scheduling is, that the pitch controller is less aggressive for higher wind speeds. The corresponding proportional factors k_P are read out of a 1D look up table, visualized in figure 9 on the left hand side. The anti-wind up part is necessary to prevent the integration of the negative error of the generated speed. This is the case, when the wind speed is below



Description	Unit	Value
T_i	[s]	9.5
k_{vsc}	[kg m^2]	112.3365
θ_{fine}	[deg]	1.0027
$\Omega_{1,1.5}$	[RPM]	5
$\Omega_{1.5,2}$	[RPM]	6
$\Omega_{2,2.5}$	[RPM]	9.6
$\Omega_{2.5,3}$	[RPM]	9.6

Figure 9: Gain-scheduling of the proportional part k_P depending on blade pitch θ (left) and baseline control parameter (right).

rated. Thus, it is a limitation of the integrator in the below rated region. Besides these two additions, the pitch controller is a usual PI controller, defined like

$$\Delta\theta = k_P(\theta)\Delta\Omega_g + k_I \int_0^t \Delta\Omega_g d\tau \tag{3.5}$$

or in the frequency domain like

$$\Delta\theta = k_P(\theta)(1 + \frac{1}{T_I s})\Delta\Omega_g, \tag{3.6}$$

with $T_I = k_P/k_I$. The summarized parameters of the baseline controller are listed in table 9 on the right hand side.

3.1.1 Benefits

The advantages of the baseline controller is the small amount of required measurements. The torque controller needs only the generator speed Ω_g and the pitch controller needs the blade pitch θ itself plus the generator speed Ω_g again. Thus, this controller does not require an observer or any additional measurements. Hence, it is quite easy to install this controller. Secondly, the complexity is manageable, the controller is built up of a nonlinear feedback and a PI element. For this very reason, there is no controller needed with a high computational power, it is directly real-time deployable. Thirdly, this controller has a high robust stability [19].

3.1.2 Downsides

The first disadvantage is regarding the constraints. The baseline controller does not include the constraints in the design process. Hence, the parameters have to be chosen, so that the limitations are not violated. This means an iterative offline tuning of the tuning parameters. And there has to be implemented hard saturations after every control output to ensure the compliance. The second downside is the separation of the controller into different regions. Out of this reason the switching from one to another region has to be smooth enough. Thirdly, there is no option to include a disturbance preview into the feedback controller. And with LIDAR systems, there exist a opportunity to get reliable wind preview data. The only option is to add a feedforward controller, in other words, implement a tracking controller of the blade pitch θ, which depends on the wind speed v_0 on the base of a steady-state relation between these two variables. So, there is potential to find better candidates for wind turbine control.

3.2 Linear-quadratic regulator (LQR)

The next controller presented in this work, is the linear-quadratic regulator (LQR) [18], especially the continuous-time infinite-horizon version. This type of controller belongs to the class of optimal controllers. It is a model based controller, the underlying dynamic model has to be linear

$$\dot{x} = Ax + Bu, \tag{3.7}$$

which is the reason for the first word in the name of the controller. In case of a given nonlinear dynamic, it has to be linearized around the desired setpoint. Thus, the chosen model in this work is the SLOWfLin model, like described in 2.3. Furthermore, every optimal controller needs a objective to optimize. Particularly, the LQR's objective is a cost function of the form

$$J = \int_0^\infty x^T Q x + u^T R u \, d\tau. \tag{3.8}$$

The cost describes the integral from zero to infinity over the sum of the weighted states plus the weighted inputs. Both summands are given in a quadratic form, this is the reason for the second word in the name of the controller. Thus, all parts of the controller are described, the resulting controller reads

$$\begin{aligned} J^* = &\min_u \int_0^\infty x^T Q x + u^T R u \, d\tau \\ &\text{s.t.} \quad \dot{x} = Ax + Bu, \end{aligned} \tag{3.9}$$

where J^* describes the minimized cost function, which is optimized with subject to the model dynamic.

The solution of the continuous time and infinity-horizon LQR is the algebraic Riccati equation [30]. The resulting matrix P is required for the control law

$$u = -Kx = -R^{-1} B^T P x. \tag{3.10}$$

This following full-feedback control law is a linear, explicit and time-invariant control law. It is depending on the chosen weighting matrices Q and R plus the underlying dynamic (A, B).

The chosen weights for the states and for the inputs can be found in [17]. The model is also a FOWT, however not the same. Nevertheless, the dynamic is similar enough to adopt the weighting parameters in this work. Because the focus of this work is not lying on the LQR, it is used as a benchmark of an optimal controller, or rather to see the potential of an optimal controller. The LQR is a predecessor of the later described MPC. Furthermore, the scaling factors are calculated with the same principle like it is used in this work, thus they are quite similar. The form of the state weighting matrix Q is defined as

$$Q = \text{diag}([0, 0, 0, q_\phi, 0, 0, 0, 0, q_{\dot{\beta}_p}, q_{\Omega_r}, q_{\dot{x}_t}, 0]). \tag{3.11}$$

and the input weighting matrix R as

$$R = \text{diag}([q_{M_g}, \, q_{\theta_c}]). \tag{3.12}$$

Every parameter of these matrices is the quotient of an individual chosen weighting factor divided by the square of the scaling factor, which is defined by twice the STD 2σ. The used parameters are listed in table 8.

Table 8: Parameters of the LQR, taken over of [17].

Description	Value
q_{ϕ}	0.0099
$q_{\dot{\beta}_p}$	64.9385
q_{Ω_r}	1.1842
$q_{\dot{x}_t}$	3.1494×10^{-4}
q_{M_g}	1.0339×10^{-11}
q_{θ_c}	0.1277

3.2.1 Benefits

The LQR provides the optimal feedback control law regarding the chosen weights in the cost function. Thus, the performance of the closed-loop control system is optimal. So, it could be used as a benchmark to other controller. Secondly, the control law is computed offline in advance, hence there is no computational power required during the deployment. It is just a linear matrix multiplication with the current state. So, the controller is for sure real time applicable.

3.2.2 Downsides

The first downside is regarding the integration of the constraints. It is the same problem as with the above described baseline controller. There is no possibility to integrate the limitations of inputs or states in the online feedback computation. In case of limitations, the controller has to be detuned offline in an iterative process, as long as the closed-loop results fulfill the constraints. Secondly, a big number of parameters has to be determined. In general, there are dim(x) + dim(u) weighting parameters to choose, in particular for the SLOWfLin model, 14 weights. Thus, there are many degrees of freedom to tune the controller and according to that difficult to find. The third disadvantage is the disturbance preview, there is also no possibility to apply the the wind and wave preview in the control system. Another decision to make, in which region should the controller takes place. One option is to place the LQR in the above rated region and the torque controller in region two. Then again, the switches between the different regions have to be sufficient smooth. Furthermore, the LQR, like the name already reveals, is a linear controller, thus it is only applicable in a certain area around the linearized setpoint. Otherwise there has to be implemented other methods, like gain-scheduled or adaptive LQR. Additionally, the controller requires a full state feedback, in other words, a observer has to be implemented. In this work it is assumed, that the states are perfectly measurable. However, if the current state has to be observed, there are high-order effects, which are not involved in the simplified linear model, like the blade structural dynamic, which could lead to wrong estimations and instability. Consequently, the robustness is not guaranteed.

Nevertheless, the LQR states a benchmark example of an optimal controller. This enables a comparison to other controller, even if only with the low order model.

3.2.3 Simulation results

The used control design model is the SLOWfLin model, the weighting matrices are in table 8. The simulation model is the nonlinear SLOWf model. Real wind and wave disturbances are active. The simulation results for the closed-loop LQR control system with an additional

Table 9: Comparison of the controller objectives between the baseline controller and the LQR.

Objective	Unit	Baseline controller	LQR	LQR/BC
$mean(P_g)$	[MW]	9.8422	9.9789	101.4%
$\sigma(P_g)$	[MW]	0.7913	0.1978	25%
$\sigma(x_t)$	[m]	0.0372	0.0309	83.1%
$\sigma(\beta_p)$	[deg]	1.0735	0.7403	69%
$\sigma(M_g)$	[kNm]	0	0	100%
$\sigma(\theta_c)$	[deg]	1.9966	2.1815	109.3%

Table 10: Comparison of the controller requirements between the baseline controller and the LQR.

Requirement	Unit	Baseline controller	LQR
$P_g \leq P_{g,max} = 11$	[MW]	11.1683	10.3523
$\Omega_r \leq \Omega_{r,max} = 10.56$	[RPM]	10.7215	9.9382
$M_g \leq M_{g,max} = 232.8064$	[kNm]	211.6422	211.6422

comparison to the baseline controller are shown in figure 10. Because the defined setpoint is around $v_0 = 16\,\mathrm{m/s}$, the simulation takes place in the above rated region. Thus, the torque control input is hold constant. The pitch actuation of the LQR is visible higher than in the baseline case, consequently the optimal controller has a higher actuation fatigue. The STDs of the pitch actuation θ_c of the LQR is 9 % higher than the STD of the baseline controller. However, the resulting rotor speed Ω_r is much smoother and varies less. The same result is visible in the generated power P_g. This is reflected in the ratio of the corresponding $\sigma(P_g)$, namely only 25 %. There are only small deviations to the rated power. The LQR has also less load fatigues, which can be seen in the STD of x_t and β_p, see table 9. Additionally, the average P_g is around 1.4 % higher, which results in more produced power. All the comparing objectives are listed in table 9. In summary, the actuation is higher, but therefore less fatigue loads arise, and a less oscillating energy generation is the result. It can be seen, that the LQR provides good results around the above rated operation point.

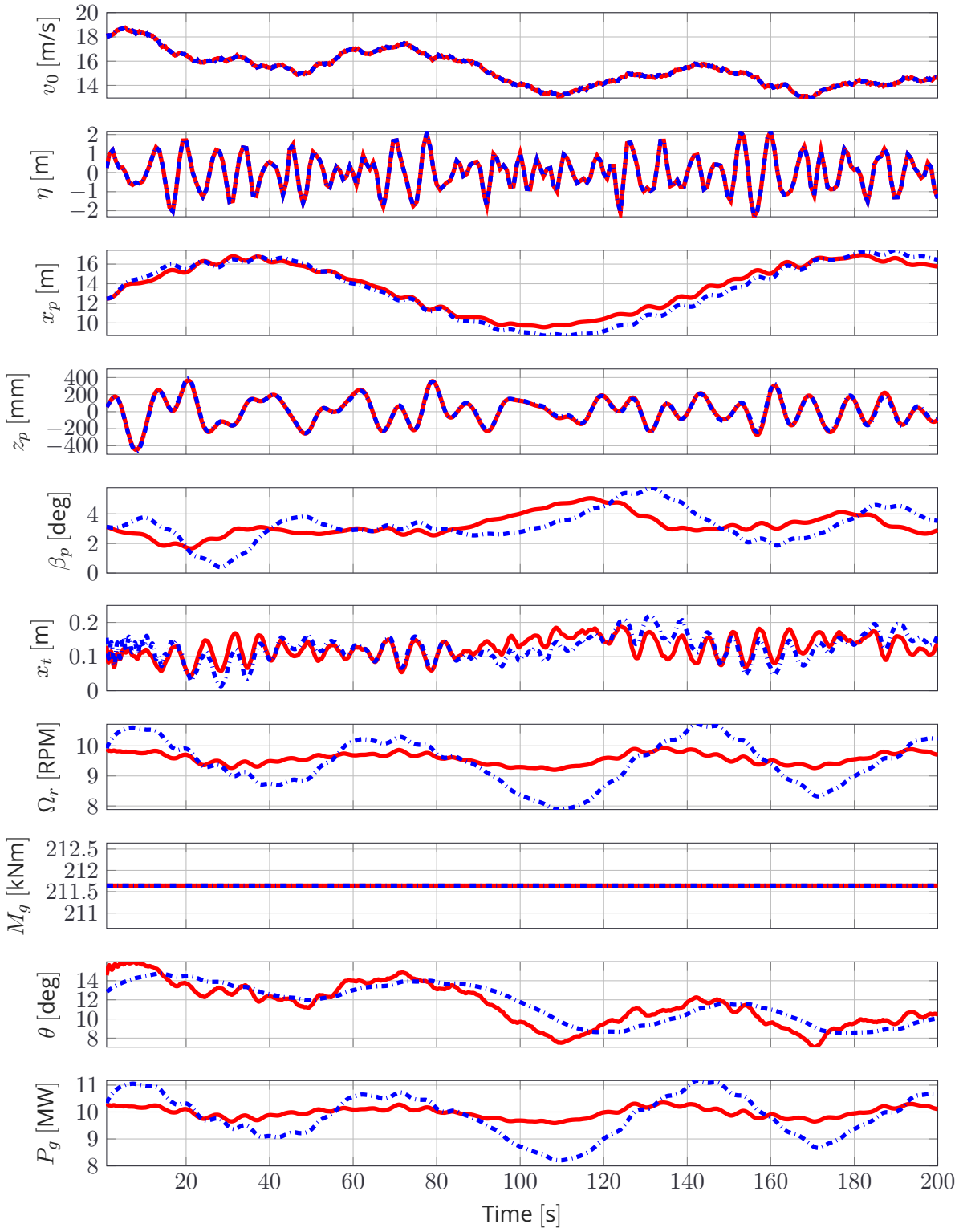


Figure 10: Comparison of the LQR (—) and the baseline controller (- -).

3.3 Model predictive control

In comparison to the last presented controller, the MPC has only a finite-horizon [23]. Still it belongs to the class of the optimal controller. Thus, the MPC needs also a cost function, which is to be optimized. The required fundamental, like it is mentioned in the name of the controller, is again a state-space model, however it has not to be linear, also nonlinear models like

$$\dot{x} = f(x, u_c, w) \tag{3.13}$$

can be included. In general, MPC is a control technique, which optimizes the defined cost function

$$J = \int_t^{t+t_f} \left(x^T Q x + u^T R u\right) d\tau + x_f^T P x_f \tag{3.14}$$

over a finite horizon, in this case from the current time t to the prediction horizon t_f. From the resulting optimal control time series

$$u^* = (u^*(t), ..., u^*(t + t_f)), \tag{3.15}$$

it is only be taken the first step $u^*(t)$ of the current time. After the application of this optimal control step, the new states of the system are measured. The new state is used as the new initial condition of the next optimization. Thus, this control method optimizes the cost function J with subject to the system dynamic and in presence of the disturbances d in every time step. This method is called receding horizon, the prediction horizon is always shifted. The method of receding horizon is important to compensate disturbances and model errors.

The cost function is like the cost function of the LQR controller, also written in a quadratic form. Beyond that, there is an additional term in the cost function, called terminal cost, because it only weights the last, or in other words, the terminal state. The aim of this term is to mimic the remaining cost, from $t + t_f$ to infinity. This can be solved by taking the result of the Riccati solution P under the use of the chosen weighting matrices Q and R. Moreover, state and/or input constraints of the form

$$h(x, u) \leq 0 \tag{3.16}$$

can be taken into account. The optimization turns now into a constrained optimization. Finally, the resulting controller set up reads

$$\begin{aligned} \min_u \int_t^{t+t_f} & x^T Q x + u^T R u \, d\tau + x_f^T P x_f \\ \text{s.t.} \quad & \dot{x} = f(x, u_c, w), \quad x = \check{x} \\ & h(x, u) \leq 0, \end{aligned} \tag{3.17}$$

where the equation $x = \check{x}$ defines the update of the state in every time step. The resulting implicit control law follows to

$$u^* = \kappa(x). \tag{3.18}$$

The calculated optimal control u^* cannot be defined a-priori. It depends on the current state x and the remaining problem set up. Every u^* will be computed online in every time step and cannot be expressed in advance explicitly.

The main part of designing a MPC is the translation of the real objective into the cost function.

3.3.1 Application to a floating offshore wind turbine

To get a better insight into MPC and state a comparable example to the later presented control type, the general MPC set up is applied to a FOWT. MPC has also been applied to floating offshore wind turbines [26]. Like the last subsection revealed, the crux is to translate the real objectives to the quadratic cost. The objectives are considered like presented in 2.6. On the one hand, the minimization of the fatigue loads is quite decent to embed in the cost function, however on the other hand the maximization of the generated power is not that straight forward.

Let's start with the fatigue loads. The aim is to reduce the oscillation of the platform pitch β_p and the tower top displacement x_t, like in the objectives defined. Thus, the quadratic variation, with individual weights, will be penalized with individual weights q_x like

$$\Pi_x(x) = q_{\dot{\beta}_p}\dot{\beta}_p^2 + q_{\dot{x}_t}\dot{x}_t^2. \tag{3.19}$$

And secondly, to alleviate the actuation, thus the input penalties reads as

$$\Pi_u(\dot{u}_c) = q_{\dot{M}_g}\dot{M}_g^2 + q_{\dot{\theta}_c}\dot{\theta}_c^2, \tag{3.20}$$

which can be summarized to the penalty

$$\Pi(x, \dot{u}_c) = \Pi_x + \Pi_u. \tag{3.21}$$

The next step is concerning the optimization of the generated power. Since the generated power is defined as the product of the input M_g and the state Ω_r, it is not possible to embed the product directly into the quadratic cost function. Therefore, another solution has to be found to transform the optimization of the generated power into the cost function. One option is to define reference trajectories for the rotor speed Ω_r and the pitch actuation θ_c. The definition of the trajectory for Ω_r is similar to the baseline control design in region two, where the generated power wants to be maximized as well. This is the case, when the tip speed ratio λ is equal its optimum λ^*. This results in a wind speed v_0 dependent rotational speed Ω_r

$$\Omega_r^* = \frac{\lambda^* v_0}{R}. \tag{3.22}$$

The optimal rotor speed Ω_r^* is tracked, under the presence of the wind disturbance v_0. In addition to this reference trajectory for Ω_r, the blade pitch is hold to zero in region two, which is required to maximize the generated energy. Thus the below rated wind speed part of the second trajectory for the blade pitch $\theta_{c,ref}$ is already defined. In region three θ_c is defined as the wind dependent steady-state pitch angle θ_{ss} to ensure a constant rated power

$$P_g = \frac{1}{2}\rho\pi R^2 c_P(\theta_{ss}, \frac{R\Omega_{rated}}{v_0})v^3 = P_{rated}. \tag{3.23}$$

Everything assembled together looks like

$$\begin{aligned} \min_u \int_t^{t+t_f} & q_{\Omega_r}(\Omega_r - \Omega_{r,ref}(v_0))^2 + q_{\theta_c}(\theta_c - \theta_{c,ref}(v_0))^2 + \Pi(x,\, \dot{u}_c)\, d\tau \\ \text{s.t.} \quad & \dot{x} = f(x, u_c, w), \quad x = \check{x} \\ & h(x, u) \leq 0. \end{aligned} \tag{3.24}$$

The constraints $h(x, u) \leq 0$ contain on the one hand input limitations, like the blade pitch velocity and on the other hand state constraints, for e.g. the rotor speed.

3.3.2 Benefits

The most important advantage is the consideration of the constraints. It is the first presented control design method, which directly includes the state and/or inputs constraints into the online calculation of the control law. So far, the constraints could only be considered after the simulation. Secondly, in this design the disturbance preview can be involved. In every time step a prediction, based on the underlying system dynamic inclusive the disturbance model, takes place to find an optimal control input u^*. Thus, the disturbance preview from the current time step t to the finite horizon T_f is directly used in the prediction, via the disturbance depending model. Consequently, the calculated control input depends on the disturbance preview. Moreover, the known wind disturbance is also used in the cost function via the reference trajectories.

3.3.3 Downsides

The disadvantage, is the high number of parameters to determine. In general, there are $dim(x)+dim(u)$ to find. So, there are many degrees of freedom to determine and accordingly difficult to find. Additionally, in the upper example for a FOWT, the reference trajectories has to be defined. No matter, how smart the trajectories are chosen, it still stays an indirect method to maximize the generated energy. So, in contrast to the high amount of tuning parameters, where there is more freedom than wanted, the indirect description of the power by defining trajectories deprives freedom of the controller. Furthermore, this control technique comes with a high computational effort. The advantage of the optimization in every time step turns into a downside concerning the computation. Related to the high computational effort is consequently the real time applicability. The optimization and the hardware have to be sufficient performative. And finally, the optimization cannot always find a solution.

3.3.4 Types

The above presented MPC set up, is the classic one. The cost function is quadratic, the control law implicit. The system dynamics were described generally. If the underlying dynamic is linear, the design method is called linear MPC, the cost stays quadratic. There are besides these versions also other types of MPC. E.g. explicit MPC, in which explicit control laws are computed offline for different operational regions, thus the computational time decreases. Especially, when the sample time has to be very small, this controller is used. Another example is stochastic or robust MPC. To increase the robustness of the MPC, there are added stochastic uncertainties on e.g. the system dynamic. In case the model is quite unreliable and there are high uncertainties on individual parameters or even on the whole model, stochastic MPC is a good choice to ensure robustness regarding the model.

Finally, the next and for this work most relevant MPC type is economic MPC (EMPC). The focus is on this type, because the main difference to the other versions is the cost function. The objective does not have to be quadratic anymore. So, there is no restriction of the form of the cost function anymore, which brings benefits with it. The next section describes the EMPC in more detail.

3.4 Economic model predictive control

In the last section mentioned, the EMPC [7] is a new developed type of the traditional MPC. The main concept, the optimization of a cost function within a defined finite horizon with subject to the system dynamic and under presence of disturbances and constraints takes place in every time step, stays the same. However, the main drawback of the classical MPC is, concerning the application of FOWT that it is not possible to translate the maximization of the power directly into the cost function. It had to be taken a detour e.g. via the reference trajectories.

The more general cost function of an EMPC looks like

$$J = \int_t^{t+t_f} V(x, u_c, w)\, d\tau + P(x_f), \tag{3.25}$$

where the objective $V(x, u, d)$ is completely free of any forms. It does not have to be quadratic or convex anymore. Everything else stays the same, in comparison the usual MPC. Thus, the set up of an EMPC reads

$$\begin{aligned} \min_u \quad & \int_t^{t+t_f} V(x, u_c, w)\, d\tau + P(x_f) \\ \text{s.t.} \quad & \dot{x} = f(x, u_c, w), \quad x = \check{x} \\ & h(x, u) \leq 0. \end{aligned} \tag{3.26}$$

Also, the same stays the implicit optimal control law $u^* = \kappa(x)$.

The usual MPC, like in the section above described is also called tracking MPC, because there is a steady state defined in advance. The chosen quadratic cost penalizes the deviations of the states and the inputs and forces consequently the system to the chosen steady state. Whereas, the EMPC cost has no specific steady state in advance, it is not a tracking MPC anymore. The cost describes an economic objective of the system. Thus, the computed optimal control inputs depend directly on the economic objective. There is no need of defining fixed steady-states in advance. The steady states vary over time with an economic cost, thus the system will be operated in a time-varying way. In other words, a dynamical optimization takes place. Time-dependent economic objectives are e.g. energy costs or demands.

3.4.1 Benefits

The previous advantages of MPC and the benefits of the special type EMPC are summed up now. The integration of disturbance preview stays the same. The optimal control action u^* is actively dependent on the future disturbances. Secondly, the consideration of the constraints is also treated in the same way, during the optimization in every optimization step. And the third benefit, which differs from tracking MPC is, the already described cost function. With it, the economic system performance is optimized directly and thus also the closed-loop performance is improved in the way it is intentionally wanted. The fourth advantage is the simplification of the control design. It depends on the developed models how much effort it is to tune the controller, however it is theoretically possible to have a tuning free controller. This means, less degrees of freedom in the control design, which is a simplification, besides an increase of the freedom of the controller.

3.4.2 Application to a floating offshore wind turbine

In the last subsections the general construct of an EMPC is presented. This part covers the application of the general framework on the specific case of a FOWT. First of all, the objectives are formulated and translated into the cost function.

3.4.3 Cost function

The overall objectives of a wind turbine are made up of two major criterion. The first one is obviously the maximization of the generated energy that is anyway the only reason why to build wind turbines to transform the kinetic energy of the wind via rotational mechanical energy of the rotor into electrical energy. The generated power is defined by

$$P_g(x, u_c) = M_g \Omega_r \frac{\eta_{el}}{i_{GB}}, \tag{3.27}$$

which is mainly the product of the generator torque M_g as an input and the generator speed Ω_r / i_{GB} as a state. If this would be the only element of the cost function, the controller would try to maximize the generated power. This behaviour is limited by the defined constraints of M_g and Ω_r. In addition, the controller would deprive as much as possible energy from the turbine over the finite cost horizon. This would lead to an unwanted effect, namely that the turbine or rather the rotor itself would have too less kinematic energy. In other words, the controller would deprive the kinetic energy of the rotor till the end of the horizon, because the controller cannot plan further that finite horizon. This known effect in EMPC is called *turnpike* [11]. To prevent this phenomenon, the kinetic energy K, defined by

$$K(x) = \frac{1}{2} J \Omega_r^2, \tag{3.28}$$

has to be maximized at the end of the horizon. Thus, K is added to the cost function as a terminal cost

$$J(t = t_f) = J_f = K(x_f) = \frac{1}{2} J \Omega_r(t_f)^2. \tag{3.29}$$

Hence, the cost function to maximizing the energy with preventing the turnpike effect looks like

$$\min_u \int_t^{t+t_f} -P_g(x, u_c) d\tau - K(x_f). \tag{3.30}$$

Now the physical size of the cost function is energy. This is already a progress to the classical MPC cost function, however there is another option to take. The above mentioned time-dependent economic objectives like price of the energy and demand could be included into the cost function. Then the cost function would have a monetary unit, e.g. in , which would be dependent on the current price on the energy market and also dependent on the energy demand. These kind of models are not covered in this work, still it would be possible with the framework of EMPC.

The second major criterion is the alleviating of structural fatigues. A typical fatigue load is the oscillation of the tower or the input actuation of the blade pitch. In operation, when the FOWT is placed offshore, both sizes are under continuous motion, which leads to fatigue of the components. Moreover to a limited lifetime of the wind turbine. In the progress of the last years, the dimensions of wind turbines have grown steadily, hence the loads have increased in the same way. To enlarge the lifetime of wind turbines, and so to increase the economic efficiency,

the structural fatigues has to be alleviate. The high-level approach is to calculate these fatigue loads into economic parameters. To enable this, economic models have to be developed. That is not an easy and straightforward task and is also out of the scope of this work. If there are models for the generated power and for the structural fatigue loads, to calculate all the physical sizes into economic units, the controller would be completely tuning free. That is the potential of the EMPC.

In this work, the fatigue loads are calculated by the states and inputs itself. The loads to reduce are the oscillation of the tower. The states, which are affected the most of the tower motion, are the platform pitch β_p and the tower top displacement x_t. To prevent high oscillations, the variations of both states have to be penalized like

$$\Pi_x(x) = q_{\dot{\beta}_p}\dot{\beta}_p^2 + q_{\dot{x}_t}\dot{x}_t^2, \tag{3.31}$$

in a usual L_2 or quadratic from. Besides these fatigue loads, there are also other parts of the wind turbine, which are loaded, the two actuators M_g and θ_c. The same method is used as for the tower oscillations. To prevent wear and tear of these actuators, again the variation of the two control inputs gets penalized

$$\Pi_u(\dot{u}_c) = q_{\dot{M}_g}\dot{M}_g^2 + q_{\dot{\theta}_c}\dot{\theta}_c^2. \tag{3.32}$$

Now the main fatigue loads are covered. Of course, there could be developed more sophisticated models of the loads, like above mentioned, however for the investigation of an EMPC on FOWT these quadratic penalties are sufficient. The combined final penalty cost reads

$$\Pi(x, \dot{u}_c) = \Pi_x + \Pi_u. \tag{3.33}$$

Added by the generated energy and the terminal cost the cost function looks like

$$J = \int_t^{t+t_f} -P_g(x, u_c) + \Pi(x, \dot{u}_x)\, d\tau - K(x_f), \tag{3.34}$$

where the two major objectives are combined in one cost function. However, this is not the final version of it.

3.4.3.1 State extension

The penalty term Π, especially the control input penalty Π_u in equation 3.32 contains the derivatives of the generated torque $\dot{M}_g$ and $\dot{\theta}_c$. These are two additional variables, which are not part of the state nor the input vector. However, it is necessary to penalize these two variables. Thus, the state vector has to be extended to apply the above defined penalty Π. The new augmented state vector x_{aug} is build up of the origin state x plus the origin control input u_c

$$x_{\text{aug}} = \begin{pmatrix} x \\ u_c \end{pmatrix}. \tag{3.35}$$

The new augmented state vector x_{aug} is henceforth defined as x, as long as it is not stated differently. This results in a 14 dimensional state vector, which is made up of

$$x = (\underbrace{x_p,\, z_p,\, \beta_p,\, \phi,\, x_t,\, \theta,\, \dot{x}_p,\, \dot{z}_p,\, \dot{\beta}_p,\, \Omega_r,\, \dot{x}_t,\, \dot{\theta}}_{x_{\text{org}}},\, \underbrace{M_g,\, \theta_c}_{u_c})^T, \tag{3.36}$$

where the vector x_{org} denotes the original 12 dimensional state vector like it is defined in equation 2.2. Therefor the state vector has increased to 14 states. Furthermore, the new input v is defined by the derivations of the origin control input u_c as

$$v = \dot{u}_c = \begin{pmatrix} \dot{M}_g \\ \dot{\theta}_c \end{pmatrix}. \tag{3.37}$$

Now the required derivatives of the origin control inputs can be used via the new inputs vector.

3.4.4 Constraints

The big advantage of MPC is the consideration of the constraints, already during the computation of the control actions. Thus, the constraints are defined in advance and are added to the problem set up. There are altogether four constraints. The actuator constraints are simply hard box limitations, defined by the physics of the corresponding hardware. The generator torque is limited from below and above with constant values, time-independently, like

$$M_{g,min} \leq M_g \leq M_{g,max}, \tag{3.38}$$

and the blade pitch and its derivative are limited the same way

$$\begin{aligned} \theta_{c,min} \leq & \theta_c \leq \theta_{c,max} \\ -\dot{\theta}_{c,max} \leq & \dot{\theta}_c \leq \dot{\theta}_{c,max}. \end{aligned} \tag{3.39}$$

The next constraints is a state limitation of the rotor speed Ω_r, it is defined like the inputs constraints

$$\Omega_{r,min} \leq \Omega_r \leq \Omega_{r,max}. \tag{3.40}$$

Like it is mentioned above, the constraints are hardware limits. Thus, the maximal value of the generator torque is the rated value. However, the max torque is indeed higher than the rated value itself, first of all out of security reasons. The usual baseline controller does not have a in advanced constraints handling, thus it is usual, that the torque exceeds its rated value. So, it is also possible in the framework of the MPC to set the boarder higher than the actual rated value. Of course, the generator is not designed to operate above rated over a longer time. Nevertheless, it is possible and to operate above rated in certain situation is not harmful. Consequently, a good choice are soft constraints. They allow the variable to exceed the maximum value in a certain way. That is the reason for the application of soft constraint for the generator torque M_g. Another reason or benefit in case of M_g for soft constraints is feasibility. For variables with uncertainty, e.g. of the model or of the realistic disturbance preview, it is favorable for the feasibility to have soft constraints. They compensate a certain extend of the uncertainty of the variable, the possibility to find a solution within the constraints is higher. Such a variable is also the rotor speed Ω_r. So, there are two candidates for soft constraints.

First of all the soft constraint concept is introduced as an example of an auxiliary optimization. State a usual optimization problem

$$\begin{aligned} \min_{x} \quad & J(x) \\ \text{s.t.} \quad & h(x) \leq 0, \end{aligned} \tag{3.41}$$

where $J(x)$ is the objective to minimize and $h(x)$ the hard constraints. This optimization problem is equivalent to

$$\begin{aligned} \min_{\tilde{x}} \quad & J(x) + q_S^T S q_S \\ \text{s.t.} \quad & h(x) \leq S \\ & 0 \leq S, \end{aligned} \tag{3.42}$$

Table 11: Constraints

Constraint	Unit	Value
$\Omega_{r,min}$	[RPM]	0
$\Omega_{r,rated}$	[RPM]	9.6
$M_{g,min}$	[KNm]	0
$M_{g,rated}$	[kNm]	211.6422
$\theta_{c,min}$	[deg]	0
$\theta_{c,max}$	[deg]	90
$\dot{\theta}_{c,min}$	[deg/s]	-7
$\dot{\theta}_{c,max}$	[deg/s]	7

where $S \in \mathcal{R}^n$, called slack variable [4], lead to the desired soft constraints. The new state $\tilde{x}$ is defined by the enlargement of the origin state x by S to $\tilde{x} = (x, S)^T$. There are now n additional inequality and an additional quadratic cost $q_S^T S q_S$. When the optimization problem is subject to a dynamic system and thus it can only be influenced by the defined control inputs, then the slack variables S have to be defined as new inputs. Otherwise the slack variables will not change at all.

When transforming these concept to the FOWT optimization problem, there are two inequalities more and changed previous inequalities

$$\begin{aligned} \theta_{c,min} \leq & \theta_c \leq \theta_{c,max} \\ -\dot{\theta}_{c,max} \leq & \dot{\theta}_c \leq \dot{\theta}_{c,max} \\ M_{g,min} \leq & M_g \leq M_{g,rated} + S_{M_g} \\ \Omega_{r,min} \leq & \Omega_r \leq \Omega_{r,rated} + S_{\Omega_r} \\ 0 \leq & S \end{aligned} \tag{3.43}$$

with

$$\begin{aligned} M_{g,max} = & M_{g,rated} + S_{M_g} \\ \Omega_{r,max} = & \Omega_{r,rated} + S_{\Omega_r} \\ S = & (S_{M_g}, S_{\Omega_r})^T. \end{aligned} \tag{3.44}$$

Also two terms are added in the cost function

$$\Pi_S(S_{M_g}, S_{\Omega_r}) = q_{S_{M_g}} S_{M_g}^2 + q_{S_{\Omega_r}} S_{\Omega_r}^2. \tag{3.45}$$

The modification affects only $M_{g,max}$ and $\Omega_{r,max}$ and not the corresponding minimum. The reason for this restriction is, that the controller is only tested in region three, speaks above rated, and in the transition region 2-3. The modification has to be extended with two additional slack variables for the lower border, however this is out of the scope of this work.

In table 11 are all the constraints with their corresponding values listed. The minima for Ω_r and M_g are set to zero and not to a cut-in minimum. The reason for this is the same as only just described, the controller is designed to operate in region three and in the transition region 2-3. Like it is explained above, the input has to be extended by the two slack variables $S = (S_{M_g}, S_{\Omega_r})^T$. Hence, the new input $\bar{v}$ is defined by

$$\bar{v} = \begin{pmatrix} v \\ S \end{pmatrix} = \begin{pmatrix} \dot{u}_c \\ S_{M_g} \\ S_{\Omega_r} \end{pmatrix}, \tag{3.46}$$

whereas in the following the new input $\bar{v}$ is denoted by v. The complete input u contains now the following elements

$$u = (\overbrace{\underbrace{\dot{M}_g, \dot{\theta}_c}_{\dot{u}_c}, \underbrace{S_{M_g}, S_{\Omega_r}}_{S}}^{v}, \underbrace{v_0, F_x(\eta), F_z(\eta), M_y(\eta)}_{w})^T \tag{3.47}$$

The input vector has increased to eight inputs, four control inputs and four disturbances. The constraints are now complete.

3.4.5 Cost function adjustment

The current cost function does not change over time, because there are not any dependencies on the time-varying price market or real time data of the energy demand. Thus, the possible shift between maximizing the energy and alleviating the loads are not implemented in the current cost function. So far, there are not such information flows developed. To analyze this possibility nevertheless, a controllable shifting parameter is introduced. Thus, there is now the option to suggest the price market and the energy demand via this shifting parameter. Additionally, a smoothing factor is introduced. From this the following cost function looks like

$$J = \int_t^{t+t_f} -\alpha(T,l)P_g(x) + \beta(T,l)\Pi(x,v) + \gamma(T,l)\Pi_S(v)\, d\tau - K(x_f), \tag{3.48}$$

where T is the shifting or tuning parameter and l the smoothing factor. To better handle the trade off between the two objectives, the cost function is separated into two groups. The first group aims to maximize the generated energy and is made up of

$$O_1 = -P_g, \tag{3.49}$$

whereas the second group aims to minimize the fatigue loads and contains

$$O_2 = \Pi + \Pi_S. \tag{3.50}$$

Where the first summand is obviously part of the group and the second one has the ability of limit the exceedance over the rated values. Or in other words, if Π_S is weighted low, there are high exceedances allowed, if weighted high, Ω_r and M_g do not overshoot the rated values significantly. Consequently, β and γ can be merged. The two remaining functions α and $\beta = \gamma$ are defined as follows

$$\alpha(T,l) = \frac{l}{l+1}(1+\frac{T}{l}) \quad \text{and} \quad \beta(T,l) = \frac{1}{6}\frac{l}{l+1}(1-\frac{T}{l}), \tag{3.51}$$

where the factor $1/6$ is the normalization factor, because the parameter β is multiplied with six individual terms of Π and Π_S. The smoothing parameter k and the shifting parameter are defined by

$$l \geq 1 \quad \text{and} \quad T \in \{-1, ..., 1\}, \tag{3.52}$$

where $l = 1$ leads to the most aggressive behavior of T, no smoothing, and $T = 1$ means the controller aims to maximize the generated energy, whereas $T = -1$ shifts the controller, that the fatigue loads are alleviated. That the energy is maximized for $T = 1$, which is actually a higher weight resulting in lower values of the variable, is because of the maximization of the power and not the minimization. Figure 11 shows the visualization of these correlations. The shifting parameter T is mapped on the x-axis from minus one to one. The y-axis shows the

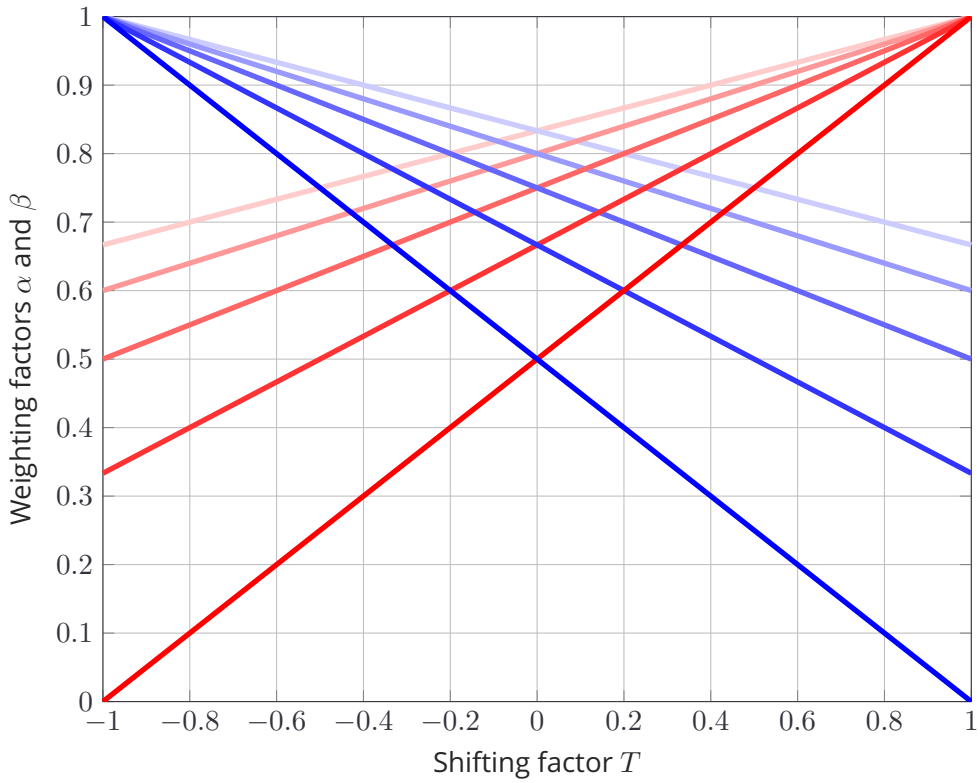


Figure 11: Correlations of the shifting parameter T and the smoothing parameter l in the cost function. A higher $l \in \{1, 2, 3, 4, 5\}$ is visualized by a higher level of fading. α in (—) and β in (—).

weighting factors α and β. A higher $l \in \{1, 2, 3, 4, 5\}$ is visualized by a higher level of fading. E.g. the most intense colors, $l = 1$, map the shifting parameter T to the weighting range from 0, total neglect of this part of the cost function, to 1, highest weighting. If T is in the neutral position 0, both weightings are 0.5. If T is -1, which means alleviating the fatigue loads, then the maximizing energy factor α is 0, whereas β is 1.

All parts for the optimization problem with the nonlinear model are complete. Thus, the final set up reads

$$\begin{aligned}
&\min_{u} \int_{t}^{t+t_f} -\alpha(T,l)P_g(x) + \beta(T,l)\big[\Pi(x,v) + \Pi_S(v)\big]\, d\tau - K(x_f) \\
&\text{s.t.} \quad \dot{x} = f(x,v,w)), \quad x = \check{x} \\
&\qquad \theta_{c,min} \leq \theta_c \leq \theta_{c,max} \\
&\qquad -\dot{\theta}_{c,max} \leq \dot{\theta}_c \leq \dot{\theta}_{c,max} \\
&\qquad M_{g,min} \leq M_g \leq M_{g,rated} + S_{M_g} \\
&\qquad \Omega_{r,min} \leq \Omega_r \leq \Omega_{r,rated} + S_{\Omega_r} \\
&\qquad 0 \leq (S_{M_g},\, S_{\Omega_R})^T,
\end{aligned} \tag{3.53}$$

with the state x as it is defined in 3.36 and the inputs v and w as described in 3.47. Each element of the cost function are defined the the equations 3.27, 3.33, 3.45 and 3.28. The trade off between the two objectives in the cost function is clearly visible and there are now ten inequality constraints.

3.4.6 Optimization problem with linearized model

In the last section, the set up of the optimization problem based on a general nonlinear model is completed. However, like it is described in earlier chapters, for the first analyses of the new control concept EMPC in application to a FOWT, a linear controller is conducted. Thus, the nonlinear set up 3.53 has to be modified to a linear model based optimization problem.

With the relation of the linearization, like it is defined in equation 2.8,

$$x = \Delta x + x_{ss}, \quad u = \Delta u + u_{ss} \tag{3.54}$$

the linearized extended state space vector Δx reads as

$$\Delta x = (\underbrace{\Delta x_p,\ \Delta z_p,\ \Delta\beta_p,\ \Delta\phi,\ \Delta x_t,\ \Delta\theta,\ \Delta\dot{x}_p,\ \Delta\dot{z}_p,\ \Delta\dot{\beta}_p,\ \Delta\Omega_r,\ \Delta\dot{x}_t,\ \Delta\dot{\theta}}_{x_{\text{org}}},\ \underbrace{\Delta M_g,\ \Delta\theta_c}_{u_c})^T, \tag{3.55}$$

the linearized extended input vector Δv as

$$\Delta v = (\underbrace{\Delta\dot{M}_g,\ \Delta\dot{\theta}_c}_{\Delta\dot{u}_c},\ \underbrace{\Delta S_{M_g}\ \ \Delta S_{\Omega_r}}_{\Delta S})^T \tag{3.56}$$

and the linearized extended disturbance vector Δw as

$$\Delta w = (\Delta v_0,\ \Delta F_x(\eta)\ \ \Delta F_z(\eta)\ \ \Delta M_y(\eta))^T. \tag{3.57}$$

The steady state values are listed in table 4 and the remaining four steady states are listed in table 12. The state space, like it is defined in equation 2.9, has to be modified regarding the

Table 12: Additional steady states, required for the linearization.

State/Input	Unit	Value
$\dot{M}_g$	[Nm]	0
$\dot{\theta}_c$	[rad/s]	0
S_{M_g}	[Nm]	0
M_{Ω_r}	[RPM]	0

new extended states and inputs, to

$$\Delta\dot{x} = \begin{pmatrix}\Delta\dot{x}_{\text{org}}\\ \Delta\dot{u}_c\end{pmatrix} = \underbrace{\begin{pmatrix}A & B_u\\ 0_{2x12} & 0_{2x2}\end{pmatrix}}_{A_{\text{aug}}}\underbrace{\begin{pmatrix}\Delta x_{\text{org}}\\ \Delta u_c\end{pmatrix}}_{\Delta x} + \underbrace{\begin{pmatrix}0_{12x2} & 0_{12x2}\\ I_2 & 0_{2x2}\end{pmatrix}}_{B_{u,\text{aug}}}\underbrace{\begin{pmatrix}\Delta\dot{u}_c\\ \Delta S\end{pmatrix}}_{\Delta v} + \underbrace{\begin{pmatrix}B_w\\ 0_{2x4}\end{pmatrix}}_{B_{w,\text{aug}}}\Delta w, \tag{3.58}$$

whereas the identity matrices I and the zero matrices 0 have the corresponding dimensions. The three augmented matrices A_{aug}, $B_{u,\text{aug}}$ and $B_{w,\text{aug}}$ are, from now on, defined as the usual A, B_u and B_w matrices, for the sake of simplicity.

The individual terms of the cost function are dependent on the linearized states and inputs. However, the generated energy needs a further investigation. Because, the linearized power is not unique, regarding whether the power is below or above rated. Because the resulting product depends on the two variables ΔM_g and $\Delta\Omega_r$, which can be positive or negative. Thus, both variables are e.g. below rated, speaks the linearized variables are negative, and the resulting

power would be positive, speaks above rated. All together it is better to keep the cost function as it is defined, this makes physically the most sense, as it is describes above. This is the reason, why the cost function has to be rewritten, depending on the linearized states and inputs to

$$\begin{aligned} P_g(\Delta x) =&(\Delta M_g + M_{g,ss})(\Delta\Omega_r + \Omega_{r,ss})\frac{\eta_{el}}{i_{GB}} \\ \Pi(\Delta x, \Delta v) =&q_{\dot{\beta}_p}(\Delta\dot{\beta}_p + \dot{\beta}_{p,ss})^2 + q_{\dot{x}_t}(\Delta\dot{x}_t + \dot{x}_{t_{ss}})^2 \\ &+q_{\dot{M}_g}(\Delta\dot{M}_g + \dot{M}_{g,ss})^2 + q_{\dot{\theta}_c}(\Delta\dot{\theta}_c + \dot{\theta}_{c,ss})^2 \\ =&q_{\dot{\beta}_p}\Delta\dot{\beta}_p^2 + q_{\dot{x}_t}\Delta\dot{x}_t^2 + q_{\dot{M}_g}\Delta\dot{M}_g^2 + q_{\dot{\theta}_c}\Delta\dot{\theta}_c^2 \\ \Pi_S(\Delta v) =&q_{S_{M_g}}(\Delta S_{M_g} + S_{M_g,ss})^2 + q_{S_{\Omega_r}}(\Delta S_{\Omega_r} + S_{\Omega_r,ss})^2 \\ =&q_{S_{M_g}}\Delta S_{M_g}^2 + q_{S_{\Omega_r}}\Delta S_{\Omega_r}^2 \\ K(\Delta x) =&\frac{1}{2}J(\Delta\Omega_r + \Omega_{r,ss})^, \end{aligned} \tag{3.59}$$

where the steady state values are already omitted, when they are equal zero.

The next step conducts the modification of the constraints in equation 3.53. They also have to be adapted to the linearized states and inputs. So far, they depend on the non linearized variables, thus they are replaced by equation 2.8 to

$$\begin{aligned} & x_{min} \leq x \leq x_{max} \\ \Leftrightarrow \quad & x_{min} \leq \Delta x + x_{ss} \leq x_{max} \\ \Leftrightarrow \quad & \underbrace{x_{min} - x_{ss}}_{\Delta x_{min}} \leq \Delta x \leq \underbrace{x_{max} - x_{ss}}_{\Delta x_{max}}, \end{aligned} \tag{3.60}$$

where x represents the state variable θ_c and the input variable $\dot{\theta}_c$. The individual two inequalities for each slack variable S_{M_g} and S_{Ω_r} have to be separated and transformed for each own. As an example, it is shown for M_g, it is done analogously for Ω_r. The linearization equation 2.8 is inserted into the origin inequality of M_g and then subtracted with $M_{g,ss}$, which looks like

$$\begin{aligned} & M_{g,min} \leq M_g \leq M_{g,rated} + S_{M_g} \\ \Leftrightarrow\ & M_{g,min} \leq \Delta M_g + M_{g,ss} \leq M_{g,rated} + \Delta S_{M_g} + S_{M_{g,ss}} \\ \Leftrightarrow\ & M_{g,min} - M_{g,ss} \leq \Delta M_g \leq M_{g,rated} - M_{g,ss} + \Delta S_{M_g} + S_{M_{g,ss}}. \end{aligned} \tag{3.61}$$

Now this triple is split into two inequalities

$$\overbrace{M_{g,min} - M_{g,ss} \leq \underbrace{\Delta M_g}_{2.} \leq M_{g,rated} - M_{g,ss} + \Delta S_{M_g} + S_{M_{g,ss}}}^{1.}, \tag{3.62}$$

where the first one is already done, the second one has to be converted to

$$\Delta M_g - \Delta S_{M_g} \leq M_{g,rated} - M_{g,ss} + S_{M_{g,ss}}, \tag{3.63}$$

which is not a usual linear state or input inequality, because it contains one state ΔM_g and one input ΔS_{M_g}. This equation has to be treated in a different way when implementing the solver, e.g. like a nonlinear inequality. Like mentioned above, the last three equation have to

be conducted in the same way for Ω_r. Also, the linearization equation 2.8 has to be applied to the two slack variable inequalities

$$\begin{aligned} \begin{pmatrix} 0 \\ 0 \end{pmatrix} &\leq \begin{pmatrix} S_{M_g} \\ S_{\Omega_r} \end{pmatrix} \\ \Leftrightarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix} &\leq \begin{pmatrix} \Delta S_{M_g} + S_{M_{g,ss}} \\ \Delta S_{\Omega_r} + S_{\Omega_{r,ss}} \end{pmatrix} \\ \Leftrightarrow \begin{pmatrix} -S_{M_{g,ss}} \\ -S_{\Omega_{r,ss}} \end{pmatrix} &\leq \begin{pmatrix} S_{M_g} \\ S_{\Omega_r} \end{pmatrix}. \end{aligned} \tag{3.64}$$

The overall adapted constraints, which are now dependent on the linearized variables, follow to

$$\begin{aligned} \theta_{c,min} - \theta_{c,ss} &\leq \Delta\theta_c &&\leq \theta_{c,max} - \theta_{c,ss} \\ -\dot{\theta}_{c,max} - \dot{\theta}_{c,ss} &\leq \Delta\dot{\theta}_c &&\leq \dot{\theta}_{c,max} - \dot{\theta}_{c,ss} \\ M_{g,min} - M_{g,ss} &\leq \Delta M_g & & \\ & \Delta M_g - \Delta S_{M_g} &&\leq M_{g,rated} - M_{g,ss} + S_{M_{g,ss}} \\ \Omega_{r,min} - \Omega_{r,ss} &\leq \Delta\Omega_r & & \\ & \Delta\Omega_r - \Delta S_{\Omega_r} &&\leq \Omega_{r,rated} - \Omega_{r,ss} + S_{\Omega_{r,ss}} \\ (-S_{M_{g,ss}}, -S_{\Omega_{r,ss}})^T &\leq (\Delta S_{M_g}, \Delta S_{\Omega_r})^T. & & \end{aligned} \tag{3.65}$$

The new lower and upper bound are defined in the following equations

$$\begin{aligned} \Delta\theta_{c,min} &= \theta_{c,min} - \theta_{c,ss}, & \Delta\theta_{c,max} &= \theta_{c,max} - \theta_{c,ss} \\ \Delta\dot{\theta}_{c,min} &= -\dot{\theta}_{c,max} - \dot{\theta}_{c,ss}, & \Delta\dot{\theta}_{c,max} &= \dot{\theta}_{c,max} - \dot{\theta}_{c,ss} \\ \Delta M_{g,min} &= M_{g,min} - M_{g,ss}, & \Delta M_{g,max} &= M_{g,rated} - M_{g,ss} + S_{M_{g,ss}} \\ \Delta\Omega_{r,min} &= \Omega_{r,min} - \Omega_{r,ss}, & \Delta\Omega_{r,max} &= \Omega_{r,rated} - \Omega_{r,ss} + S_{\Omega_{r,ss}} \\ (\Delta S_{M_{g,min}}, \Delta S_{\Omega_{r,min}})^T &= (-S_{M_{g,ss}}, -S_{\Omega_{r,ss}})^T. & & \end{aligned} \tag{3.66}$$

All parts for the optimization problem with the linear model are done. Consequently, the final set up is made of the cost function 3.59, the internal model 3.58 and the constraints 3.65 with their defined maximum and minimum 3.66 to

$$\begin{aligned} \min_u \int_t^{t+t_f} & -\alpha(T,l)P_g(\Delta x) + \beta(T,l)\left(\Pi(\Delta x, \Delta v) + \Pi_S(\Delta v)\right) d\tau + K(\Delta x_f) \\ \text{s.t.} \quad & \Delta\dot{x} = A\Delta x + B_u\Delta v + B_w\Delta u_w, \Delta \quad x = \Delta\check{x} \\ & \Delta\theta_{c,min} \leq \Delta\theta_c \leq \Delta\theta_{c,max} \\ & \Delta\dot{\theta}_{c,min} \leq \Delta\dot{\theta}_c \leq \Delta\dot{\theta}_{c,max} \\ & \Delta M_{g,min} \leq \Delta M_g, \quad \Delta M_g - \Delta S_{M_g} \leq \Delta M_{g,max} \\ & \Delta\Omega_{r,min} \leq \Delta\Omega_r, \quad \Delta\Omega_r - \Delta S_{\Omega_r} \leq \Delta\Omega_{r,max} \\ & (\Delta S_{M_{g,min}}, \Delta S_{\Omega_{r,min}})^T \leq (\Delta S_{M_g}, \Delta S_{\Omega_r})^T. \end{aligned} \tag{3.67}$$

3.4.7 Comparison to classical Model Predictive Control

This subsections covers the comparison between the developed EMPC set up out of the last subsections and the classical MPC set up. One mentioned benefit is the simplification of the controller design. Like described below, with the special monetary models and a complete transformation of the cost function into strict economic objectives, the EMPC would be tuning free. However, since this is not the case in this work, some weightings remain, which have

to be tuned. To demonstrate the reduction of the tuning parameters, both cost functions are compared to each other, or rather only the maximizing power part of the corresponding cost function. Since the second objective, reducing the loads, is almost identical in both cases. The power part of the traditional MPC framework is defined by

$$V_{MPC} = \int_{t}^{t+t_f} q_{\Omega_r}(\Omega_r - \Omega_{r,ref}(v_0))^2 + q_{\theta_c}(\theta_c - \theta_{c,ref}(v_0))^2, \tag{3.68}$$

where the to determined parameters are colored in red. Two weightings and two reference trajectories. In comparison to the cost function of the EMPC, the power part looks like

$$V_{EMPC} = -P_g(x), \tag{3.69}$$

where no parameter has to be tuned. The shifting weight α is neglectable, like described above, this parameter only exists to mimic the time-variant economic influences. Thus, the obviously visible difference is the reduced amount of parameters to determine. Or especially in the present case, reduced to even no tuning. Of course, the parameters of the penalty Π and of the slack variable weighting remain.

3.4.8 Overview

A block diagram is shown in figure 12 to visualize the complete control system in a high-level view. In the center of figure is the control system itself, with the plant, the state feedback, the

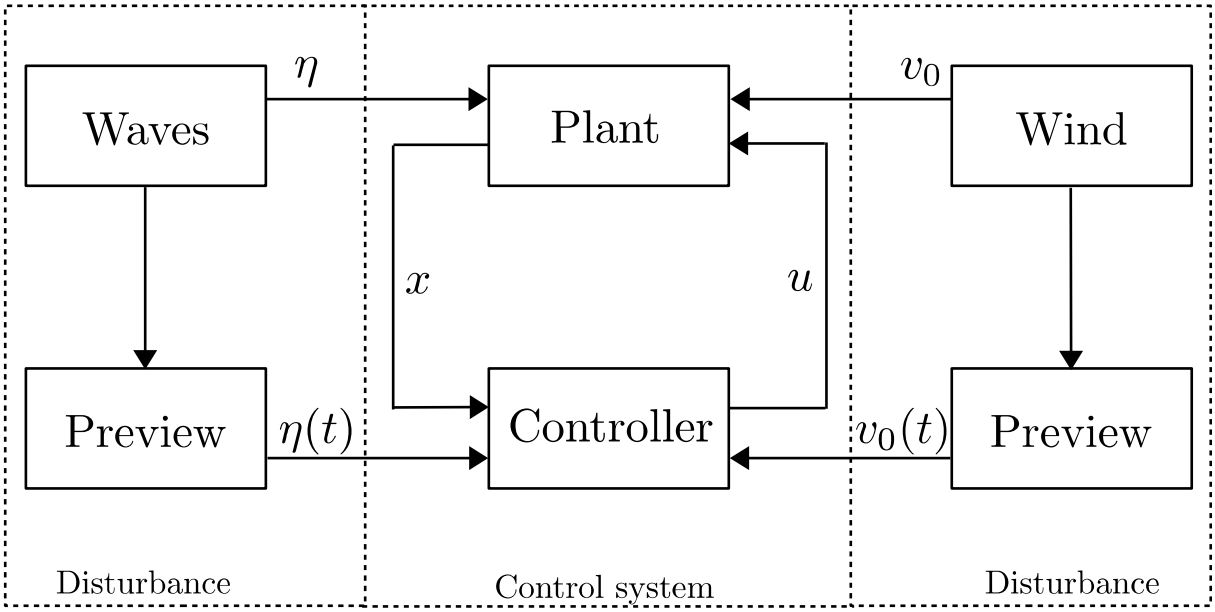


Figure 12: Block diagram of the complete control system.

control action and the controller. This also present the initial situation of the next chapters, first of all, the complete state is measurable and no observer is required. Moreover, on the left and the right outer sides are shown the disturbances. On the left hand side the waves, which act with the current wave height η on the plant. Additional, the preview block provides a

wave prediction $\eta(t) = (\eta(t_0), ..., \eta(t_f))$, beginning with the current wave height η_{t_0} till the wave height of the final prediction horizon $\eta(t_f)$. Also, the same set up is given on the right hand side for the current wind disturbance v_0, inclusive the wind preview $v_0(t)$. Like the measurable feedback, also the preview is initially assumed to be perfect. This perfect framework serves the first analysis of an EMPC for FOWT.

3.4.9 Downsides

Like the classic MPC, the high computational effort required to solve the optimization problems, remains unchanged. It is the biggest drawback of the EMPC. Additional to the already high computational cost, the complexity even increases in the economic case. So far, in the MPC set up the cost function is convex, this changes with the new cost function, which is to be formulated in general terms, not convex anymore. Thus, the biggest advantage of this type, the direct formulation of the objectives, turn in account of the computation into the main downside. The effort to solve such problems increases. This is the reason, why there has to be found a sufficient efficient solver for this optimization problem. With this downside in mind, the next chapter covers the search for a proper solver.

4 Solver

This chapter covers the embedding and implementation of the theoretically developed controller from chapter 3 into a solver environment.

The first section describes the general challenges and compares the different solvers. It is followed by the required discretization of the simulation model and the optimization problem, which is required for the implementation of the set up. The third section describes the necessary scaling process to achieve convergence and real time capability.

4.1 Solvers

In this section three different types of solvers are presented. Each advantage and disadvantage and a short overview of the individual properties. First of all, the challenges for the solver are described in more detail.

4.1.1 Challenges

As already described above, the EMPC requires a high computational effort. Thus, the solver has to provide a high computational efficiency to solve the derived optimization problem in every iteration. This optimization has to done in less than the specified sample time to ensure real time applicability. The controller predicts the states depending on inputs, which are so far unknown, till the prediction horizon. These time series of every state is inserted into the cost function. Finally, this scalar and non-convex cost function has to be optimized with subject to the given constraints. In particular with assumed values of a prediction horizon of 10 s and a sample time of 0.25 s, the resulting number of steps is 40. The system consists of 12 physical states. With the extension of the state vector by the two new inputs, as described in chapter 3, the final state vector contains 14 entries. Additionally, the system consists of two control inputs and two slack variables, as introduced in chapter 3 as well. Altogether, the total number of variables is 18. These 18 variables have to be optimized for every time step within the prediction horizon, which consists of 40 steps. Thus the total number of variables, which has to be determined is $18 \cdot 40 = 720$. Furthermore, the big optimization has to be feasible and has to converge to an optimal solution in every iteration.

4.1.2 Solver overview

In this section, three solvers are presented and compared. In the next three subsections, every solver is described in more detail.

4.1.2.1 ACADO

ACADO is an open source solver developed at the University of Freiburg within the Optimization in Engineering Centre (OPTEC) under supervision of Moritz Diehl [13]. It can be redistributed under the GNU LGPL license. The acronym ACADO means Automatic Control and Dynamic Optimization. The software package provides several algorithms for optimal control, among others for Model Predictive Control. The toolkit can be programmed with C++ or with the MATLAB interface as well. The nonlinear optimization is solved by the application of a sequential quadratic programming (SQP) algorithm .

One benefit of ACADO is the particular support for MPC problems. Also, the open source license and the independence of MATLAB. It could be used as a self-contained C++ software package. Additionally, it is possible to generation code for fast MPC. The toolkit provides an auto generated Gauss-Newton real-time iteration algorithms and a linear algebra QP solver called qpOASES.

The project is not under development anymore, it was closed 2014. However, there is a successor called *acados*, which is under current development. Also by the team of Moritz Diehl at the University of Freiburg and in corporation with the KU Leuven.

4.1.2.2 MATLAB MPC Toolbox

The well-known numerical computing environment MATLAB, developed by the American company Mathworks, is the developer of the solver within the MPC Toolbox [31].

The biggest advantage is the known development environment and all the usual benefits of MATLAB. The documentation is detailed and there is a wide range of examples to help the users.

In first preliminary tests, one disadvantage has been the computational speed. The optimization time per iteration is longer than the sample time and thus not real time applicable. The computational speed could be improved by transfer the MATLAB code to C-code. Another downside is the dependence of MATLAB itself. When MathWorks stops the support for certain features in newer version of MATLAB, one cannot do anything about it, expect to stick to the older version, which has again downsides. This was the case with custom cost feature within the linear MPC framework. Since version 2018b, this feature is not supported anymore, only in the nonlinear MPC framework. However, the same code, in terms of content, only transformed to the new nonlinear set up, is significantly slower than in the linear framework. From the perspective of the nonlinear SLOWf model, several code improvements should be found to increase the computational speed significantly.

4.1.2.3 FORCES PRO

Forces Pro is a real time decision making software of the Swiss company Embotech [6] [35]. Embotech itself is a developer of the latest decision-making software. They have quite big companies as customers. Among those, the firm Vestas, with whom they have developed already a controller for wind turbines. The default solve method is the Primal-Dual Interior-Point Method. A stable and robust method to solve nonlinear convex optimization problems.

One big advantage is the support of this solver. Through the commercial license, there is support included. Embotech provides a very fast solver in comparison to others [6]. As described above, there exists already a wind turbine controller in collaboration with Vestas. Another advantage is the wide range of tuneable settings of the solver.

The major downside is the commercial license and the related costs. Another disadvantage is the dependency on the software itself, as described in the MATLAB section.

4.1.3 Selection of the solver

All three types of solvers could be tested and improved for the described problem. Also benchmarking or a comparison study between these three solvers would be interesting. However, this would be beyond the scope of this work. The focus is to investigate the realizability and performance of a new control concept in application on a FOWT. Thus, one solver is chosen and tested. The selected solver is the FORCES PRO of the company Embotech. The aim of this work is to investigate the performance of this novel and promising solver.

4.2 Discretization

To simulate the closed loop system, either a continuous dynamic model with a discrete integrator within the simulation has to be chosen or a discretization has to be conducted first. In this work the latter way is used. In case of the comparison simulations, the discretization has not been described in more detail. Now it is important to look at it in more detail. Not only the simulation model has to be discretized but also the optimization problem. All the discretization is done within the linearized model and its state space.

The used linear SLOWfLin model is defined in the continuous time domain. In the last chapter it has been extended, see equation 3.58, and can be written as

$$\Delta\dot{x} = A\Delta x + B_u\Delta v + B_w\Delta w. \tag{4.1}$$

With the sample time T_s, which will be determined later, the dynamic system can be discretized to

$$\Delta x_{k+1} = A_d\Delta x_k + B_{u,d}\Delta v_k + B_{w,d}\Delta w_k, \tag{4.2}$$

where k represents the current time step, A_d the corresponding discrete system matrix, $B_{u,d}$ the discrete control input matrix and $B_{w,d}$ the discrete disturbance input matrix. The discretization is done with a zero-order hold, which assumes piecewise constancy over the sample time. The resulting state space vector is defined by

$$\begin{aligned}\Delta x_k =&(\Delta x_{p_k},\, \Delta z_{p_k},\, \Delta\beta_{p_k},\, \Delta\phi_k,\, \Delta x_{t_k},\, \Delta\theta_k,\\ &\Delta\delta x_{p_k},\, \Delta\delta z_{p_k},\, \Delta\delta\beta_{p_k},\, \Delta\Omega_{r_k},\, \Delta\delta x_{t_k},\, \Delta\delta\theta_k,\, \Delta M_{g_k},\, \Delta\theta_{c_k})^T,\end{aligned} \tag{4.3}$$

where δx denotes the numerical derivation with its corresponding linearization

$$\delta x_k = \frac{x_{k+1} - x_k}{T_s}, \quad \Delta\delta x_k = \delta x_k - \dot{x}_{ss}. \tag{4.4}$$

The complete linearized input Δu changes to

$$\Delta u_k = (\underbrace{\Delta\delta M_{g_k},\, \Delta\delta\theta_{c_k},\, \Delta S_{M_{g,k}}, \Delta S_{\Omega_{r,k}}}_{\Delta v_k},\, \underbrace{\Delta v_{0_k},\, \Delta F_{x_k},\, \Delta F_{z_k},\, \Delta M_{y_k}}_{\Delta w_k})^T. \tag{4.5}$$

The determining parameter in the discretization is the sample time T_s. The used sample time in the simulations for the comparison of SLOWfLin and SLOWf is 0.025 s, which corresponds to a sample rate f_s of 40 Hz. This used sample time does not pose any problems for controllers with low computational effort, like the used baseline controller. However, for controller with a high computation time, like MPCs, this sample rate could be too high. The aim is to find on the one hand a small enough sample time to ensure real time capability and on the other hand a sufficient high enough sample time to cover all the phenomenon of the model and do not neglect any dynamics. Thus a convergence study has to be implemented, to find the sample time, which fulfils the described criteria.

Table 13: Convergence study. Comparison of the time series of the rotor speed Ω_r with different sampling times. The differences are listed with the maximum of the absolute error $e_{a,max}$, the maximum of the relative error $e_{r,max}$ and the variance of the absolute error $var(e_a)$.

Sample rate [Hz]	Sample time [s]	$e_{a,max}$	$e_{r,max}$	$var(e_a)$
40	0.025	0	0	0
20	0.05	5.05×10^{-5}	4.55×10^{-5}	5.57×10^{-11}
10	0.1	1.51×10^{-4}	1.37×10^{-4}	5.03×10^{-10}
5	0.2	3.47×10^{-4}	3.14×10^{-4}	2.72×10^{-9}
4	0.25	4.29×10^{-4}	3.89×10^{-4}	4.71×10^{-9}
<4	>0.25	x	x	x

4.2.1 Convergence study

In section 2.5 the comparison between the linear and the nonlinear model has been implemented. The result is, that the linear SLOWfLin model contains all the dynamic and thus is valid to use as a surrogate model, with a sample rate of 40 Hz. This convergence study, conducted in this section, shows the impact of lower sample rates on the simulation error. As a comparing variable, the rotor speed Ω_r is chosen. The time series of the rotor speed of the simulation with 40 Hz is the reference. The sample rate is reduced iteratively. In table 13, the maximum of the absolute, the maximum of the relative error and the variance of the error's time course is listed with the corresponding sample rate and sample time. The two types of error and the variance are defined by

$$e_a = max\left(\Omega_r - \Omega_{r,ref}\right), \quad e_r = max\left(\frac{\Omega_r - \Omega_{r,ref}}{\Omega_{r,ref}}\right), \quad var(e_a) = var(\Omega_r - \Omega_{r,ref}). \tag{4.6}$$

The resulting table shows, that a relatively low sample rate of $f_s = 4$ Hz is sufficient to obtain all the dynamics. For lower frequencies, the simulation diverges. At least for the simulations with the SLOWfLin model, this sample time is sufficient. For other simulation environments like *FAST* the sample rate for the simulation must be adjusted again. Because these models contain dynamics with different frequencies.

Table 5 lists all natural frequencies with the corresponding time constants. The highest natural frequency is the one from the blade pitch with $f_n = 1.6$ Hz. To avoid aliasing, the sample rate has to be twice as high as this frequency

$$f_s \geq 2\max(f_n) = 2 \cdot 1.6Hz = 3.2Hz \tag{4.7}$$

Thus, the sample rate of $f_s = 4$ Hz resulted from the convergence study is also out of this perspective valid. Hence this is the resulting sample time by which the system dynamic is discretized.

4.2.2 Optimization problem

Besides the sample time for the simulation model, the final linearized optimization problem in equation 3.67, has to be discretized as well. Thus, there is also a sample time T_s required. In general, this time constant does not have to be the same as the sample time of the simulation. It also could be higher, or in other words, the controller could also work with a lower frequency than the simulation itself. If this is the case, the control inputs are hold constant for the further

simulation steps till a new control action is computed. The benefit of this method is the reduction of the computational effort. The expensive optimization does not have to be executed as often as the simulation model has to be integrated. However, the lowest allowed sample time is 4 Hz, according to the last section. Hence, the discretization of the optimization problem is done with the same sample time as it is used for the system dynamic.

To transform the problem set up, only small adjustments have to done. The first is the discretization of the internal model, as it is done with the simulation itself. The second one is the change of the continuous integral to a discrete sum. Also the continuous time t of the integral changes to the step number k. The continuous problem is stated in equation 3.67 and the transformed discrete optimization problem looks like

$$\begin{aligned}
\min_u \sum_{k=1}^{N-1} \Big[&- \alpha(T,l) P_{g_k}(\Delta x_k) + \beta(T,l)\Big(\Pi_k(\Delta x_k, \Delta v_k) + \Pi_{S_k}(\Delta v_k)\Big)\Big] - K_k(\Delta x_N) \\
\text{s.t.} \quad & \Delta x_{k+1} = A_d \Delta x_k + B_{u,d}\Delta v_k + B_{w,d}\Delta w_k, \quad \Delta x_k = \Delta \check{x}_k \\
& \Delta\theta_{c,min} \leq \Delta\theta_{c_k} \leq \Delta\theta_{c,max} \\
& \Delta\dot{\theta}_{c,min} \leq \Delta\delta\theta_{c_k} \leq \Delta\dot{\theta}_{c,max} \\
& \Delta M_{g,min} \leq \Delta M_{g_k}, \quad \Delta M_{g_k} - \Delta S_{M_g} \leq \Delta M_{g,max} \\
& \Delta\Omega_{r,min} \leq \Delta\Omega_r, \quad \Delta\Omega_r - \Delta S_{\Omega_r} \leq \Delta\Omega_{r,max} \\
& (\Delta S_{M_{g,min}}, \Delta S_{\Omega_{r,min}})^T \leq (\Delta S_{M_{g,k}}, \Delta S_{\Omega_{r,k}})^T,
\end{aligned} \tag{4.8}$$

where N defines the horizon length. The prediction horizon t_f is set to 10 s, thus with a sample time T_s of 0.25 s the resulting horizon length amounts $N = 40$ stages. The discrete terms in the cost function, follows to

$$\begin{aligned}
P_{g_k}(\Delta x_k) =& (\Delta M_{g_k} + M_{g,ss})(\Delta\Omega_{r_k} + \Omega_{r,ss})\frac{\eta_{el}}{i_{GB}} \\
\Pi_k(\Delta x_k, \Delta v_k) =& q_{\dot{\beta}_p}\Delta\delta\beta_{p_k}^2 + q_{\dot{x}_t}\Delta\delta x_{t_k}^2 + q_{\dot{M}_g}\Delta\delta M_{g_k}^2 + q_{\dot{\theta}_c}\Delta\delta\theta_{c_k}^2 \\
\Pi_{S_k}(\Delta v_k) =& q_{S_{M_g}}\Delta S_{M_{g,k}}^2 + q_{S_{\Omega_r}}\Delta S_{\Omega_{r,k}}^2 \\
K_k(\Delta x_k) =& \frac{1}{2} J(\Delta\Omega_{r_k} + \Omega_{r,ss})^2,
\end{aligned} \tag{4.9}$$

The discrete optimization problem set up is completed.

4.3 Scaling

This section covers the crucial relevance of the scaling of the problem set up. In chapter 3, the theoretic optimization set up has been developed. However, with this unmodified framework, the optimization problem is ill-conditioned. This means, that it is hard to find a solution due to variables, which are in different sizes. As an example, the usual range of the first control input M_g is around 2.11e5 Nm and the usual range of the second control input θ_c is around 0.17 rad. This shows the huge difference between the control inputs, which has to be determined.

The same error using the three solvers is to find an optimal solution. The result of this error can be seen in the termination of the algorithm, which leads to no control inputs at all, and in reaching the maximum amount of iterations. The second appearance of the error, reaching the maximum number of iterations, leads again to two problems. First, to a suboptimal solution and second to a high solving time. Both negative effects have to be eliminated for a fast

and optimal control algorithm.

The crucial point to solve these unwanted errors is scaling. The next subsections describe the solution in detail.

4.3.1 Cost function

The first approach of solving this problem is the scaling of the cost function. The cost function is the most critical part and at the same time the most modifiable part of the complete optimization cost function. Especially in the case of the EMPC, the cost function is fully customizable and not restricted by any norms.

So far the cost function in a reduced form is presented as

$$J = \sum_{k=1}^{N-1} \Big[- P_{g_k}(\Delta x_k) + \Pi_k(\Delta x_k, \Delta v_k) + \Pi_{S_k}(\Delta v_k) \Big] - K_k(\Delta x_N) \tag{4.10}$$

with $\alpha(T, l)$ and $\beta(T, l)$ are set to one. This simplification is possible without restricting the result of further modification, it is just for the sake of clarity. Also, for this reason, every notification of the time step k in the indices is omitted.

In the first step the weighting coefficients of both penalty terms Π and Π_S are set to one. They will be needed later, but first each term of the cost function has to be in the same size range. Only then, the weighting parameters can be used in a proper way. The transformation is done with scaling. Every state and input is divided by a scaling factors like

$$\widetilde{x} = \frac{x}{\widehat{x}} = \frac{\Delta x + x_{ss}}{\widehat{x}} = \underbrace{\frac{\Delta x}{\widehat{x}}}_{\Delta\widetilde{x}} + \frac{x_{ss}}{\hat{x}} \tag{4.11}$$

where x denotes the origin value, $\widehat{x}$ the scaling factor, x_{ss} the steady-state value and $\widetilde{x}$ the scaled variables (see 2.8). Additionally, the scaled and linearized variable is defined by $\Delta\widetilde{x}$.

The next step is to find appropriate scaling factors $\hat{x}$. One option is to scale each variable with the maximum permitted value or the rated value. This would result in a nice meaningful scaling, e.g. the scaled variable is equal one means, that the maximum or the rated value is reached. The magnitude range spans mainly between a one-digit number. This scaling method is easily conducted with e.g. the input M_g or the state Ω_r. Both variables have a rated value and a defined maximum. For this type of variables the method works, however the operation range of other variables are not known. Thus it is not straight forward to determine scaling factors for these variables.

A pragmatic way, to get an approximation of the operation range, is to take the results of the SLOWf simulation with the default baseline controller as it is done in section 2.5. The scaling factors are defined as twice the STD σ of each variable's time series

$$\hat{x} = 2\sigma(x(t)). \tag{4.12}$$

This definition means, that the variables are scaled with the mean of the third largest amplitudes, see [17]. The resulting scaling factors plus additional maximum, minimum and mean

Table 14: Extracted scaling factors from the closed loop simulation with the baseline controller, SLOWf as simulation model, wind speed of $v_0 = 16$m/s and with the corresponding wave disturbances.

State x	Unit	Scaling factor $\hat{x}$	Mean mean(x)	Minimum $min(x)$	Maximum $max(x)$
x_p	[m]	3.8574	12.7455	7.6560	18.7358
z_p	[m]	0.1948	0.0155	-0.4614	0.3667
β_p	[deg]	2.0279	3.2041	0.0000	6.6250
ϕ	[deg]	1	-	-	-
x_t	[m]	0.0699	0.1223	0.0014	0.2597
θ	[deg]	3.6562	12.2386	7.1385	16.4313
δx_p	[m/s]	0.3102	0.0000	-0.5069	0.5324
δz_p	[m/s]	0.1376	0.0000	-0.2212	0.2097
$\delta \beta_p$	[deg/s]	0.3066	0.0000	-0.5195	0.5591
Ω_r	[RPM]	1.3382	9.6044	7.6288	11.4804
δx_t	[m/s]	0.0471	0.0000	-0.1866	0.1580
$\delta \theta$	[deg/s]	0.2122	0.0000	-0.3397	0.3731
M_g	[kNm]	0	211.6422	211.6422	211.6422
θ_c	[deg]	3.6563	12.2386	7.1384	16.4316
Input v	Unit	Scaling factor $\hat{v}$	Mean mean(v)	Minimum $min(v)$	Maximum $max(v)$
δM_g	[kNm/s]	0.0000	0.0000	0.0000	0.0000
$\delta \theta_c$	[deg/s]	0.2124	0.0000	-0.3399	0.3737
P_g	[MW]	1.3939	10.0045	7.9466	11.9588

Table 15: Required scaling factors of M_g and its derivative. Extracted of the simulation with a wind speed of $v_0 = 10$m/s.

State x	Unit	Scaling factor $\hat{x}$	Mean mean(x)	Minimum $min(x)$	Maximum $max(x)$
M_g	[kNm]	55.421	174.1599	82.3124	211.6422
δM_g	[kNm/s]	2.8701	0.0150	-10.813	6.7589

data for each state and input is listed in table 14. The data concerning the new input v is calculated by deriving the time courses of the origin input. The variation of the generator torque δM_g is equal zero and thus also the required scaling factor. The reason for this behavior is the fact, that the simulation is conducted with an average rotor effective wind speed of 16 m/s, which is above the rated wind speed. Consequently, the baseline controller holds the generator torque constant. Hence, $\delta \widehat{M_g}$ cannot be extracted from this simulation results. As an auxiliary simulation, a wind speed of 10 m/s is taken. In this simulation, there are also below rated parts and therefore changes in the generator torque. The computed scaling factor and the other related features of M_g and δM_g are listed in table 15. All the scaling factors are now defined. Thus the scaled variables are introduced like

$$\tilde{x} = \frac{x}{\hat{x}} = \frac{x}{2\sigma(x)} = \frac{\Delta x + x_{ss}}{2\sigma(x)} = \underbrace{\frac{\Delta x}{2\sigma(x)}}_{\Delta \widetilde{x}} + \frac{x_{ss}}{2\sigma(x)}. \tag{4.13}$$

is introduced.

The first term of the cost function, the generated power P_g changes with the introduction of

the scaling to

$$\widetilde{P}_g(\Delta x) = \frac{P_g}{\widehat{P}_g} = \frac{1}{\widehat{P}_g}(\Delta M_g + M_{g,ss})(\Delta\Omega_r + \Omega_{r,ss})\frac{\eta_{el}}{i_{GB}}. \tag{4.14}$$

The first penalty term Π reads as

$$\widetilde{\Pi}(\Delta x, \Delta v) = \delta\widetilde{M}_g^2 + \delta\widetilde{\theta}_c^2 + \delta\widetilde{\beta}_p^2 + \delta\widetilde{x}_t^2 = \left(\frac{\Delta\delta M_g}{\delta\widehat{M}_g}\right)^2 + \left(\frac{\Delta\delta\theta_c}{\delta\widehat{\theta}_c}\right)^2 + \left(\frac{\Delta\delta\beta_p}{\delta\widehat{\beta}_p}\right)^2 + \left(\frac{\Delta\delta x_t}{\delta\widehat{x}_t}\right)^2 \tag{4.15}$$

and the second one Π_S as

$$\widetilde{\Pi}_S(\Delta v) = \Delta\widetilde{S}_{M_g}^2 + \Delta\widetilde{S}_{\Omega_r}^2 = \left(\frac{\Delta S_{M_g}}{\widehat{S}_{M_g}}\right)^2 + \left(\frac{\Delta S_{\Omega_r}}{\widehat{S}_{\Omega_r}}\right)^2. \tag{4.16}$$

And finally the last term of the cost function, the terminal kinetic energy K, rewritten with the scaling factor

$$\widetilde{K}(\Delta x) = \frac{K}{\widehat{K}} = \frac{1}{\widehat{K}}\frac{1}{2}J(\Delta\Omega_r + \Omega_{r,ss})^2. \tag{4.17}$$

All scaling factors are listed in the tables 14 and 15, except $\widehat{K}$, which is defined like

$$\widehat{K} = \frac{1}{2}J(\Delta\widehat{\Omega}_r + \Omega_{r,ss})^2, \tag{4.18}$$

where J denotes the moment of inertia. Altogether the scaled cost function looks like

$$J = \sum_{k=1}^{N-1}\Big[-\alpha\widetilde{P}_{g_k}(\Delta x_k) + \beta\big[\widetilde{\Pi}_k(\Delta x_k, \Delta v_k) + \widetilde{\Pi}_{S_k}(\Delta v_k)\big]\Big] - \widetilde{K}_k(\Delta x_N). \tag{4.19}$$

The main problem, described in the beginning of this section, is the unscheduled termination of the algorithm. With the scaling of the cost function, the algorithm do not abort anymore, however the suboptimal solution, with high solving times due to the reaching of the maximum iterations, remain. Thus, one important step is done, nevertheless the scaling has to be extended to solve the remaining problem of the slow convergence. This is the reason for the extension of the scaling on the complete system and not only on the cost function. It is described in the next subsection.

4.3.2 Complete system

As described in the last subsection, the scaling is extended to the complete system dynamics of the internal model of the optimization problem, to solve the remaining issue of the slow convergence. The two control inputs can be found faster, when the magnitude range of the two control inputs in particular and for the complete variables in general, is not as high as in the unscaled case anymore.

The current linearized internal model, which is defined in 4.8, with the states, control and disturbance inputs defined in 4.3 and 4.5 will be scaled.

The above described scaling in equation 4.11 is applied to the complete state space vector Δx, to the control input vector Δv and to the disturbance input vector Δw. With the relation

Table 16: Scaling factors of the disturbances plus the corresponding additional information.

Dist. w	Unit	Scaling factor $\hat{w}$	Mean mean(w)	Minimum $min(w)$	Maximum $max(w)$
v_0	[m/s]	2.6355	15.7993	12.0478	19.3798
$F_x(\eta)$	[MN]	9.1190	0.0022	-15.769	16.101
$F_z(\eta)$	[MN]	5.3109	0	-8.5060	8.9468
$M_y(\eta)$	[MNm]	135.34	-0.0384	-216.4138	232.2371

$\Delta x = \widehat{x}\,\Delta\widetilde{x}$ for one single state, the relation for the whole state follows to

$$\underbrace{\begin{pmatrix}\Delta x_p\\ \Delta z_p\\ \vdots\\ \Delta\delta\theta\end{pmatrix}}_{\Delta x} = \underbrace{\begin{pmatrix}\widehat{x}_p & 0 & \dots & 0\\ 0 & \widehat{z}_p & & \\ \vdots & & \ddots & \vdots\\ 0 & & \dots & \delta\widehat{\theta}\end{pmatrix}}_{D_x} \underbrace{\begin{pmatrix}\Delta\widetilde{x}_p\\ \Delta\widetilde{z}_p\\ \vdots\\ \Delta\delta\widetilde{\theta}\end{pmatrix}}_{\Delta\widetilde{x}}. \tag{4.20}$$

The same follows for the state transformation of the control input and for the disturbance input

$$\begin{aligned}
\underbrace{\begin{pmatrix}\Delta\delta M_g\\ \Delta\delta\theta_c\\ \vdots\\ \Delta S_{\Omega_r}\end{pmatrix}}_{\Delta v} &= \underbrace{\begin{pmatrix}\delta\widehat{M}_g & 0 & \dots & 0\\ 0 & \delta\widehat{\theta}_c & & \\ \vdots & & \ddots & \vdots\\ 0 & & \dots & \widehat{S}_{\Omega_r}\end{pmatrix}}_{D_u} \underbrace{\begin{pmatrix}\Delta\delta\widetilde{M}_g\\ \Delta\delta\widetilde{\theta}_c\\ \vdots\\ \Delta\widetilde{S}_{\Omega_r}\end{pmatrix}}_{\Delta\widetilde{v}}\\
\underbrace{\begin{pmatrix}\Delta v_0\\ \Delta F_x\\ \vdots\\ \Delta M_y\end{pmatrix}}_{\Delta w} &= \underbrace{\begin{pmatrix}\widehat{v}_0 & 0 & \dots & 0\\ 0 & \widehat{F}_x & & \\ \vdots & & \ddots & \vdots\\ 0 & & \dots & \widehat{M}_y\end{pmatrix}}_{D_w} \underbrace{\begin{pmatrix}\Delta\widetilde{v}_0\\ \Delta\widetilde{F}_x\\ \vdots\\ \Delta\widetilde{M}_y\end{pmatrix}}_{\Delta\widetilde{w}}
\end{aligned} \tag{4.21}$$

All the elements in each of the three matrices are from the tables 14, 15 and the remaining required scaling factors for the disturbances are listed in table 16. With all three state transformations the origin linearized state space model is modified by

$$\begin{aligned}
\Delta x_{k+1} &= A_d\Delta x_k + B_{u,d}\Delta v_k + B_{w,d}\Delta w_k\\
\Leftrightarrow D_x\Delta\widetilde{x}_{k+1} &= A_d D_x\Delta\widetilde{x}_k + B_{u,d}D_u\Delta\widetilde{v}_k + B_{w,d}\Delta\widetilde{w}_k\\
\Leftrightarrow \Delta\widetilde{x}_{k+1} &= \underbrace{D_x^{-1}A_dD_x}_{\widetilde{A}_d}\Delta\widetilde{x}_k + \underbrace{D_x^{-1}B_{u,d}D_u}_{\widetilde{B}_{u,d}}\Delta\widetilde{v}_k + \underbrace{D_x^{-1}B_{w,d}D_w}_{\widetilde{B}_{w,d}}\Delta\widetilde{w}_k
\end{aligned} \tag{4.22}$$

It is like an extended state space transformation, not only with the state, also with the inputs. The transformation of the states and the inputs plus the internal model is visualized in figure 13. The difference of the unscaled and scaled dynamic is pointed out in the condition of the system matrix A and $\widetilde{A}$. The condition cond is defined by the ratio of the largest and the smallest singular value. The relation of the conditions of the two matrices is described like

$$\frac{\text{cond}(A)}{\text{cond}(\widetilde{A})} \approx 12. \tag{4.23}$$

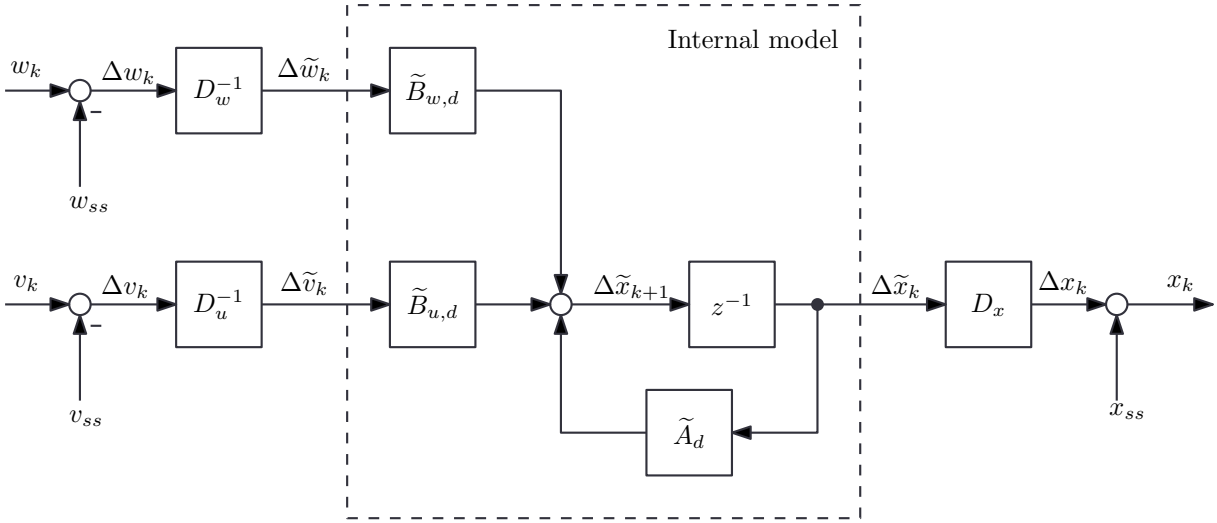


Figure 13: Scaled, linearized and discretized system dynamic of the internal model.

It is a clear improvement visible, the singular values are not widely spread anymore, the ratio is about 12 times lower. This results in a better conditioned numerical problem and thus in a faster convergence of the optimization problem.

The internal model dynamic is scaled now. The next steps are the scaling of the cost function and the scaling of the constraints.

4.3.2.1 Cost function

Already, in subsection 4.3.1 the cost function has been scaled, however without the new scaled states and inputs. This is the reason, why the cost function has to be rewritten again, with the new introduced scaled variables.

The first term of the cost function is the generated power P_g, which changes with the introduction of the scaled variables to

$$\begin{aligned} P_g =& (\widehat{M}_g \, \Delta\widetilde{M}_g + M_{g,ss})(\widehat{\Omega}_r \, \Delta\widetilde{\Omega}_r + \Omega_{r,ss}) \frac{\eta_{el}}{i_{GB}} \\ =& (\Delta\widetilde{M}_g + \frac{M_{g,ss}}{\widehat{M}_g})(\Delta\widetilde{\Omega}_r + \frac{\Omega_{r,ss}}{\widehat{\Omega}_r}) \underbrace{\widehat{M}_g \widehat{\Omega}_r \frac{\eta_{el}}{i_{GB}}}_{\widehat{P}_g} \\ =& (\Delta\widetilde{M}_g + \frac{M_{g,ss}}{\widehat{M}_g})(\Delta\widetilde{\Omega}_r + \frac{\Omega_{r,ss}}{\widehat{\Omega}_r}) \qquad \widehat{P}_g \end{aligned} \tag{4.24}$$

$$\Leftrightarrow$$

$$\widetilde{P}_g(\Delta\widetilde{x}) = \frac{P_g}{\widehat{P}_g} = (\Delta\widetilde{M}_g + \frac{M_{g,ss}}{\widehat{M}_g})(\Delta\widetilde{\Omega}_r + \frac{\Omega_{r,ss}}{\widehat{\Omega}_r}).$$

The two penalties Π and Π_S are almost already defined in the equations 4.15 and 4.16. The occurring fractions are already the actual definition of the scaled and linearized states and inputs, thus the new penalty terms follow to

$$\widetilde{\Pi}(\Delta\widetilde{x}, \Delta\widetilde{v}) = \Delta\delta\widetilde{M}_g^2 + \Delta\delta\widetilde{\theta}_c^2 + \Delta\delta\widetilde{\beta}_p^2 + \Delta\delta\widetilde{x}_t^2, \tag{4.25}$$

and

$$\widetilde{\Pi}_S(\Delta\widetilde{v}) = \Delta\widetilde{S}^2_{M_g} + \Delta\widetilde{S}^2_{\Omega_r}. \tag{4.26}$$

The last term is the terminal kinetic energy K, which looks edited like

$$\begin{aligned} K =& \frac{1}{2} J(\widehat{\Omega}_r\,\Delta\widetilde{\Omega}_r + \Omega_{r,ss})^2 \\ =& \underbrace{\frac{1}{2} J\widehat{\Omega}_r^2}\left(\Delta\widetilde{\Omega}_r + \frac{\Omega_{r,ss}}{\widehat{\Omega}_r}\right)^2 \\ =& \quad \widehat{K} \quad \left(\Delta\widetilde{\Omega}_r + \frac{\Omega_{r,ss}}{\widehat{\Omega}_r}\right)^2 \\ \Leftrightarrow& \\ \widetilde{K}(\Delta\widetilde{x}) =& \frac{K}{\widehat{K}} = \left(\Delta\widetilde{\Omega}_r + \frac{\Omega_{r,ss}}{\widehat{\Omega}_r}\right)^2. \end{aligned} \tag{4.27}$$

Thus all terms of the cost function are now depending on the linearized and scaled variables. The cost function reads as

$$J = \sum_{k=1}^{N-1} \Big[-\alpha\widetilde{P}_{g_k}(\Delta\widetilde{x}_k) + \beta\big[\widetilde{\Pi}_k(\Delta\widetilde{x}_k, \Delta\widetilde{v}_k) + \widetilde{\Pi}_{S_k}(\Delta\widetilde{v}_k)\big]\Big] - \widetilde{K}_k(\Delta\widetilde{x}_N). \tag{4.28}$$

The next step is the modification of the constraints.

4.3.2.2 Constraints

The constraints of the linearized states are defined above in equation 3.60. Now they must be changed, so that the constraints are depending on the linearized and scaled states $\Delta\widetilde{x}$. The modification follows by

$$\begin{aligned} \Delta x_{min} \leq& \Delta x \leq \Delta x_{max} \\ \Leftrightarrow \Delta x_{min} \leq& \widehat{x}\,\Delta\widetilde{x} \leq \Delta x_{max} \\ \Leftrightarrow \frac{\Delta x_{min}}{\widehat{x}} \leq& \Delta\widetilde{x} \leq \frac{\Delta x_{max}}{\widehat{x}}, \end{aligned} \tag{4.29}$$

where the new upper and lower limits are defined by

$$\Delta\widetilde{x}_{min} = \frac{\Delta x_{min}}{\widehat{x}}, \quad \Delta\widetilde{x}_{max} = \frac{\Delta x_{max}}{\widehat{x}}. \tag{4.30}$$

This scaling of the constraints can be applied to the first two constraints. The constraints with slack variables must be treated separately. As an example, it is shown with the two constraints of M_g, the modifications can be transformed one by one to the constraints of Ω_r. Firstly, the lower bound is transformed to

$$\begin{aligned} \Delta M_{g,min} \leq& \Delta M_g \\ \Leftrightarrow \Delta M_{g,min} \leq& \widehat{M}_g\,\Delta\widetilde{M}_g \\ \Leftrightarrow \frac{\Delta M_{g,min}}{\widehat{M}_g} \leq& \Delta\widetilde{M}_g \\ \Leftrightarrow \Delta\widetilde{M}_{g,min} \leq& \Delta\widetilde{M}_g, \end{aligned} \tag{4.31}$$

secondly, the upper bound to

$$\begin{aligned} \Delta M_g - \Delta S_{M_g} \leq& \Delta M_{g,max} \\ \Leftrightarrow \widehat{M}_g\,\Delta\widetilde{M}_g - \widehat{S}_{M_g}\,\Delta\widetilde{S}_{M_g} \leq& \Delta M_{g,max}. \end{aligned} \tag{4.32}$$

The last constraints, which have to be edited, are regarding the slack variables itself, and are modified to

$$\begin{aligned}
&(\Delta S_{M_{g,min}},\, \Delta S_{\Omega_{r,min}})^T \leq (\Delta S_{M_g},\, \Delta S_{\Omega_r})^T \\
\Leftrightarrow\; &(\Delta S_{M_{g,min}},\, \Delta S_{\Omega_{r,min}})^T \leq (\widehat{S}_{M_g}\, \Delta\widetilde{S}_{M_g},\, \widehat{S}_{\Omega_r}\, \Delta\widetilde{S}_{\Omega_r})^T \\
\Leftrightarrow\; &(\frac{\Delta S_{M_{g,min}}}{\widehat{S}_{M_g}},\, \frac{\Delta S_{\Omega_{r,min}}}{\widehat{S}_{\Omega_r}})^T \leq (\Delta\widetilde{S}_{M_g},\, \Delta\widetilde{S}_{\Omega_r})^T \\
\Leftrightarrow\; &(\Delta\widetilde{S}_{M_{g,min}},\, \Delta\widetilde{S}_{\Omega_{r,min}})^T \leq (\Delta\widetilde{S}_{M_g},\, \Delta\widetilde{S}_{\Omega_r})^T.
\end{aligned} \tag{4.33}$$

Altogether the complete constraints read as

$$\begin{aligned}
&\Delta\widetilde{\theta}_{c,min} \leq \Delta\widetilde{\theta}_c \leq \Delta\widetilde{\theta}_{c,max} \\
&\Delta\delta\widetilde{\theta}_{c,min} \leq \Delta\delta\widetilde{\theta}_c \leq \Delta\delta\widetilde{\theta}_{c,max} \\
&\Delta\widetilde{M}_{g,min} \leq \Delta\widetilde{M}_g \\
&\widehat{M}_g\, \Delta\widetilde{M}_g - \widehat{S}_{M_g}\, \Delta\widetilde{S}_{M_g} \leq \Delta M_{g,max} \\
&\Delta\widetilde{\Omega}_{r,min} \leq \Delta\widetilde{\Omega}_r \\
&\widehat{\Omega}_r\, \Delta\widetilde{\Omega}_r - \widehat{S}_{\Omega_r}\, \Delta\widetilde{S}_{\Omega_r} \leq \Delta\Omega_{r,max} \\
&(\Delta\widetilde{S}_{M_{g,min}},\, \Delta\widetilde{S}_{\Omega_{r,min}})^T \leq (\Delta\widetilde{S}_{M_g},\, \Delta\widetilde{S}_{\Omega_r})^T.
\end{aligned} \tag{4.34}$$

4.3.2.3 Final set up

The final set up of the discrete optimization problem, which is depending on the scaled and linearized states and inputs, is defined by

$$\begin{aligned}
&\min_u \sum_{k=1}^{N-1} \Big[-\alpha\widetilde{P}_{g_k}(\Delta\widetilde{x}_k) + \beta\big[\widetilde{\Pi}_k(\Delta\widetilde{x}_k, \Delta\widetilde{v}_k) + \widetilde{\Pi}_{S_k}(\Delta\widetilde{v}_k)\big]\Big] - \widetilde{K}_k(\Delta\widetilde{x}_N) \\
&\text{s.t.} \quad \Delta\widetilde{x}_{k+1} = \widetilde{A}_d\Delta\widetilde{x}_k + \widetilde{B}_{u,d}\Delta\widetilde{v}_k + \widetilde{B}_{w,d}\Delta\widetilde{w}_k, \quad \Delta\widetilde{x}_k = \Delta\widetilde{\check{x}}_k \\
&\quad \text{Constraints:} \quad 4.34.
\end{aligned} \tag{4.35}$$

All the following simulations, within this chapter and the next one, are run with a prediction time of 10 s, which leads to 40 prediction steps, because the determined step time is 0.25 s. The chosen prediction time is usual selected time for LIDAR based preview.

The results of this optimization problem, which are the results of the controller, have to be transformed back in order to be able to apply them to the plant. Also, the current state of the plant plus the disturbances have to be transformed first, so that the controller is able to compute the optimization. This structure is visualized in figure 14.

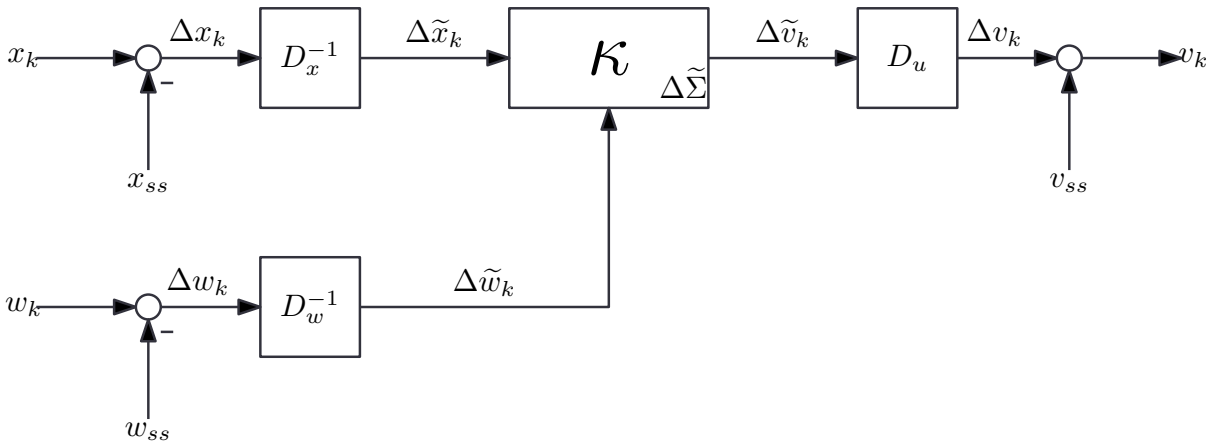


Figure 14: Block diagram of the complete scaled and shifted controller. With two inputs x and w and the output v.

4.4 Optimization

The scaling is finished, thus the remaining parameter to determine are the weights of the different terms in the cost function. The variables in the cost function are scaled, because the whole system is scaled. However, the scaling factors are fix and not meant to be adjustable. Thus, the weights are the possibility to change the cost function for individual purposes. The cost function plus the corresponding advertised form looks like

$$\begin{aligned} J &= \sum_{k=1}^{N-1} \Big[-\alpha(T,l) q_{P_g} \widetilde{P}_{g_k}(\Delta\widetilde{x}_k) + \beta(T,l)\big(\widetilde{\Pi}_k(\Delta\widetilde{x}_k, \Delta\widetilde{v}_k) + \widetilde{\Pi}_{S_k}(\Delta\widetilde{v}_k)\big)\Big] - \widetilde{K}_k(\Delta\widetilde{x}_N) \\ &= \sum_{k=1}^{N-1} \Big[\alpha(T,l) q_{P_g} \widetilde{P}_{g_k} + \beta(T,l)\big(q_{M_g}\Delta\delta\widetilde{M}_g^2 + q_{\theta_c}\Delta\delta\widetilde{\theta}_c^2 + q_{\beta_p}\Delta\delta\widetilde{\beta}_p^2 \\ &\qquad + q_{x_t}\Delta\delta\widetilde{x}_t^2 + q_{S_{\Omega_r}}\Delta\delta\widetilde{S}_{\Omega_r}^2 + q_{S_{M_g}}\Delta\delta\widetilde{S}_{M_g}^2\big)\Big] - \widetilde{K}_k, \end{aligned} \tag{4.36}$$

with the seven weighting parameters, denoted with q plus corresponding index. As described above, the transformation of the first overall objective, maximizing the generated power, is straight forward. With the new presented EMPC control design, the generated power P_g can be adopted directly. However, the second overall aim, the reduction of the structural loads, is not that easy without complex, not yet developed models. With these models, the structural loads could be calculated as economic objectives and directly compared to the generated power and its economic efficiency. However, these models are not developed yet and consequently, the above weights of the states and inputs, have to be determined. It is harder to estimate the importance of each penalty in taking account of the structural damage, which comes with the oscillation of the corresponding state or input. Hence, the following optimizations are estimations and the trade-offs of the different structural fatigues.

Because of the highly complex dynamics of a FOWT, and the resulting potential of an unpredictable behavior by changing weighting parameters, a brute force approach is used. A benefit of this approach is the possibility to analyze the system dynamic with respect to the chosen weights. It makes it possible to find out certain connections between the objectives. The shifting parameter T is set to zero and the smoothing factor l equal to one for the whole optimization process.

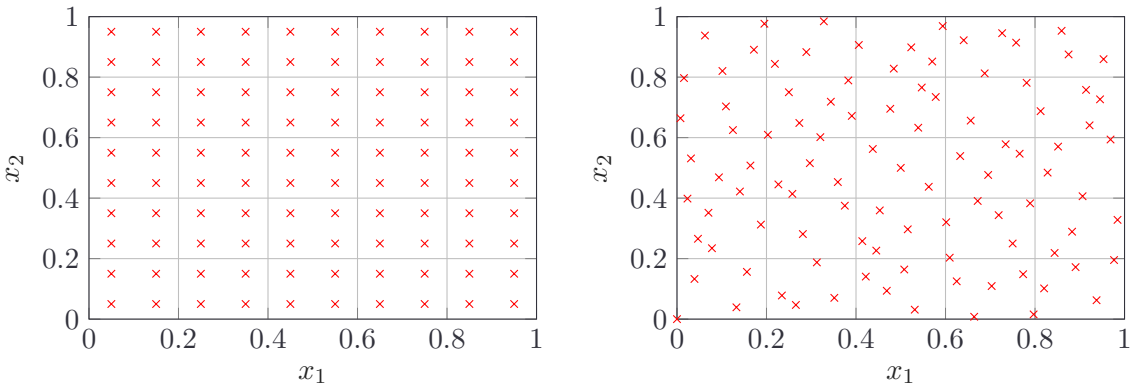


Figure 15: On the left hand side, the full factorial set is plotted. On the right hand side, the Sobol sample design. Both designs have 100 samples.

4.4.1 Multi-objective optimization

For a multi-objective optimization, the first thing to do is, to find a appropriate sample design. There are different space-filling designs available, e.g. a full factorial design or the latin hypercube sampling. The downside of a full factorial design is, that the values of each design parameter is used several times, or in other words, less different values are tested. In this work a Sobol sequence is used to fill the sampling space. This sequence is a quasi-random number generator, which has good space-filling properties and is also invariant in account on the samples. The figure 15 shows the different space-filling of the full factorial and the Sobol sample design in case of two design dimension x_1 and x_2. Both designs have 100 samples. It is not possible to run so many simulations, thus all combinations of the seven weighting parameters are included. If all six parameters should be tested with ten different values, which is not that much, this would lead to 1×10^7 simulations, when all combinations are taken into account. This is the reason for splitting this complex multi-objective optimization into several smaller optimizations.

If all parameter are set to one, the simulation runs successfully, however also, the rotor speed Ω_r operates in its maximum constraint and the same applies to the generator torque M_g. Because of the defined constraints and the compliance by the EMPC, these two variables do not increase further. The reason for this excess is the term, which maximizes the generated power in the cost function. To prevent this behavior, the two slack variables S_{Ω_r} and S_{M_g} have been introduced. By increasing the weight, the excess of Ω_r and M_g is penalized. Hence, this is a good starting point to start the optimization process. Only the described weights of each optimization are changed, every other weight is set to one, as long as not differently stated. Additionally, with two variables it is still possible to use graphical tools, to analyze the simulation results afterwards.

4.4.1.1 Solver option

All the following simulations within the optimization process are run with 150 s simulation time. This length is a trade-off between an acceptable overall simulation time and revealing the interesting objectives to optimize the weights. Other solver options are described now. An optimization flag is set, which means, the compilation time takes longer, but the deployment of the solver is faster. The datatype is set to 64 bit double floating point. The code is compiled for the

host machine, FORCES PRO offers also target specific compilation. The barrier strategy is set the logarithmic option. The hessian approximation is done via BFGS. As linear system solver the symmetric indefinite form is used, to solve the KKT system. This solver is the most robust one, the fast version of it, have not achieved to converge and it was faster than the default one. As automatic differentiation tool, FORCES uses the external tool CasAdi. Another option would be the Symbolic Toolbox of MATLAB. As solve method the default Primal-Dual Interior-Point Method is used. The solver initialization of this method is set to a centered start instead of a cold start. This change enables a more robust and faster convergence than the default one.

4.4.1.2 Optimization of the slack variables

To apply the Sobol sequence to the two slack variables, the range of the design space has to be determined first. As a start, a large space is applied. The benefit is to get a good overview and discover knowledge to apply a second sample design in a smaller and more decisive space. The first set of weights include values from 1×10^{-3} to 1×10^{4}. Because of the wide range, the Sobol sequence is distributed exponentially, therefore the axes are plotted in a logarithmic way. Similar to the example of the sample design above, 100 samples are used to investigate the results. To find out appropriate weighting parameters, the relative mean of the rotor speed Ω_r and the mean of the generator torque M_g are used as indicator to evaluate the samples. The relative mean values of the two variables

$$\text{mean}(\widetilde{\Omega}_r) = \frac{\text{mean}(\Omega_r)}{\Omega_{r,ss}}, \quad \text{mean}(\widetilde{M_g}) = \frac{\text{mean}(M_g)}{M_{g,ss}} \tag{4.37}$$

are taken to evaluate the weights. Everything above 5% exceed, which is represented by 1.05 of the relative mean, is defined as damaging for the generator and its related parts. This is a very simplified assumption, however, in this work are not used any further models for the acceptable damage of the generator. In [12] an approach for the limitation of the cumulated constraint violation is given. It provides a model with corresponding limits, which defines how much over-rated power can be tolerated, while the heat dissipation capabilities of the generator does not exceed. In figure 16 the two relative mean values of the 100 simulations with respect to the two slack variable weights are plotted. On the left hand side, it is visible, that

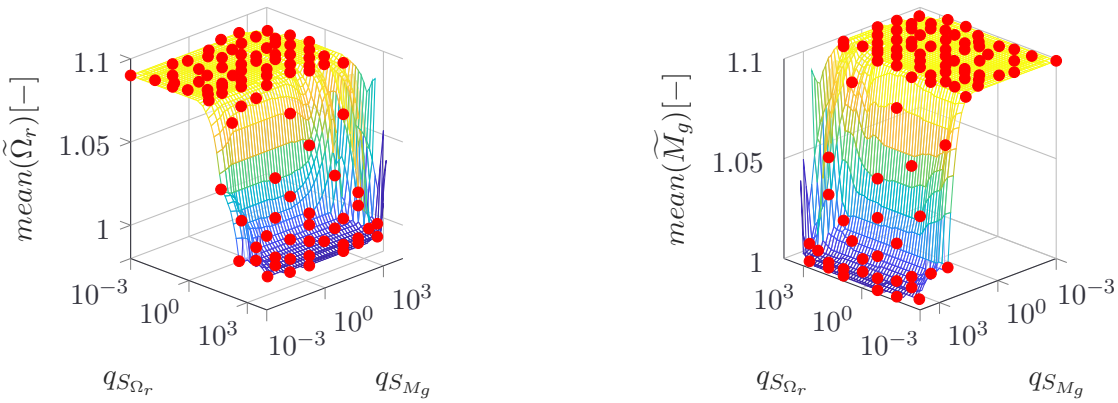


Figure 16: On the left hand side, the mean of the rotor speed Ω_r with respect to the two weights of the slack variable is plotted. On the right hand side, the mean of the generator torque M_g.

the rotor speed weight $q_{S_{\Omega_r}}$ has to be high enough, to fall below the 5% limit of the mean. The same applies for the right hand side plot for $q_{S_{M_g}}$. Small weights do not have any effect. This

knowledge is taken to define a new range of the parameters for a second set of simulations. All the values smaller 1×10^1, are neglected. And the upper bound for the weights is set to 1×10^3. The resulting contour plot is shown in figure 17. Out of this defined area, all the re-

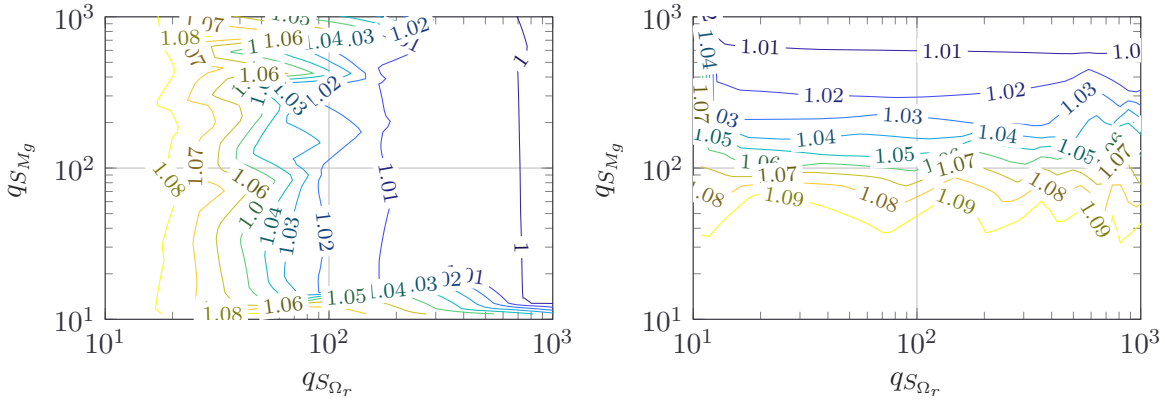


Figure 17: On the left hand side, the relative mean of the rotor speed Ω_r with respect to the two weights of the slack variable is plotted. On the right hand side, the relative mean of the generator torque M_g.

maining samples, which have a smaller relative mean than 1.05 for both variables, are taken as the final set. Now all the defined objectives have to be compared. Because, there are five objectives to compare, it is not that easy to visualize it in a proper way. One possibility to visualize a multi-objective optimization is the parallel plot. Figure 18 shows the remaining samples in such a parallel plot. All objectives are drawn parallel to each other and every line represents

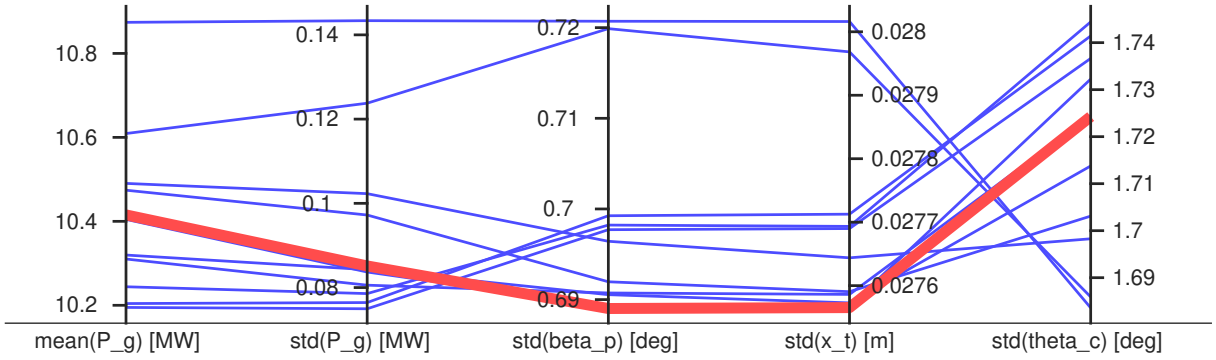


Figure 18: Parallel plot of five objectives with respect to samples of the slack variables. Every line represents one sample out of the defined set of $(q_{S_{\Omega_r}}, q_{S_{M_g}})$. The chosen sample $(q_{S_{\Omega_r}}, q_{S_{M_g}}) = (205.3525, 205.3525)$ is marked with a thicker red line.

one sample. A parallel plot is a good option to visualize trade-offs between objectives, if this is the case the sample lines cross each other or at least have a high slope between two objective axes. There are still different possible solutions left. One sample, marked with a thicker red line, is chosen. This sample combines low structural loads, low variation of the generated power and still a quite high mean of the generated power. The weighting pair $(q_{S_{\Omega_r}}, q_{S_{M_g}}) = (205.3525, 205.3525)$ is used for the following optimization process. There are other possibilities as well, e.g. a higher mean of P_g, however this comes along with higher structural loads. The chosen weights dependent on different use cases. It is not straight forward to choose an overall good sample, because the estimation of the impact of e.g. the different structural loads it is quite hard. The platform pitch STD is between 0.69 deg and 0.72 deg and the STD of the

Table 17: Comparison of different structural load weights.

	(a)	(b)
q_{β_p}	1	67.7188
q_{x_t}	1	430.6875
mean(P_g)	10.3419	10.1196
$\sigma(P_g)$	0.1132	0.4630
$\sigma(x_t)$	0.0274	0.0257
$\sigma(\beta_p)$	0.6750	0.7153
$\sigma(\theta_c)$	1.6860	1.5031

tower top displacement between 0.0276 m and 0.028 m. The range of $\sigma(x_t)$ is much smaller, but to estimate the resulting fatigue loads in comparison to $\sigma(\beta_p)$ is difficult. It is also important to check the resulting solving times of the chosen weights. The individual solving time can vary a lot. But in general, the lines are mostly parallel, which means, there are no significant trade-offs.

4.4.1.3 Optimization of the structural load weights

Again, a large exponential range is used, from 1×10^{-3} to 1×10^{4}. By comparing the contour plots of all objectives with respect to the two weights q_{β_p} and q_{x_t}, it is visible, that small values do not have any effect on the objectives. The relations between different weighting parameters and the resulting objective is complex. After several simulation sets, the final area is defined by $1 \leq q_{\beta_p} \leq 70$ and $1 \leq q_{x_t} \leq 500$. The resulting parallel plot is shown in figure 19. In

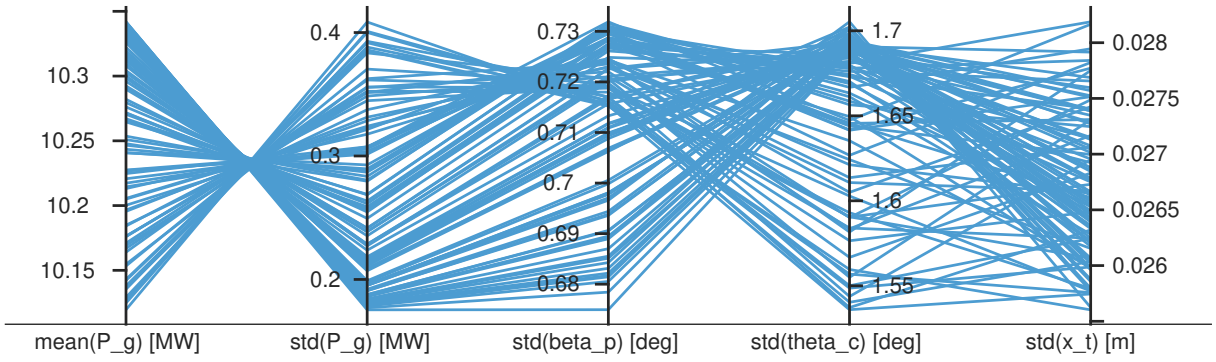


Figure 19: Parallel plot of five objectives with respect to different structural load samples. Every line represents one simulation with a sample out of the defined set of (q_{x_t}, q_{β_p}).

comparison to the last parallel plot, there is a high number of crossings between two objectives. E.g. between the STD of β_p and the STD of θ_c. High values of $\sigma(\beta_p)$ lead to low values of $\sigma(\theta_c)$. These crossings mean a high amount of trade-offs. Also, the first two objectives are in contrast to each other. It is hard to find a proper sample to satisfy all objectives. There is no weighting pair, which has low structural loads, low variation of the generated power and simultaneously low control actuation. If further knowledge of the impact of the different fatigue loads or preferences for one individual objective is available, it is possible to choose certain weighting parameter to fulfill them. In table 17 are two different cases listed. Case (a) uses the default parameters and achieves low variation of the generated power and of the platform pitch. In contrast to case (b), which has to focus on lower variation of the blade actuation and of the tower top displacement.

All weighting pairs have to be checked for realistic time courses. The solving time for the optimization is not changing much in dependence on the two weights. For the further discovery, the default samples are used.

4.4.1.4 Optimization of the generated power weight

The next parameter to discover is the weight of the generated power q_{P_g}. Again, a wide range is taken for the first set of simulations to get an overview of the influence of q_{P_g} on the objectives. The same as for the weights of the slack variables applies also for the generated power parameter. Higher averages of the generated power than 10.5 MW, which is equal to 105%, are not allowed. This is the case for $q_{P_g} \geq 1.5$. Because there is only one parameter in this optimization step, 2D plots are sufficient to analyze the impact of the weight. Figure 20 shows the result of the simulations on the objective. In the range of q_{P_g} between 1×10^{-1} and 1.5 is almost

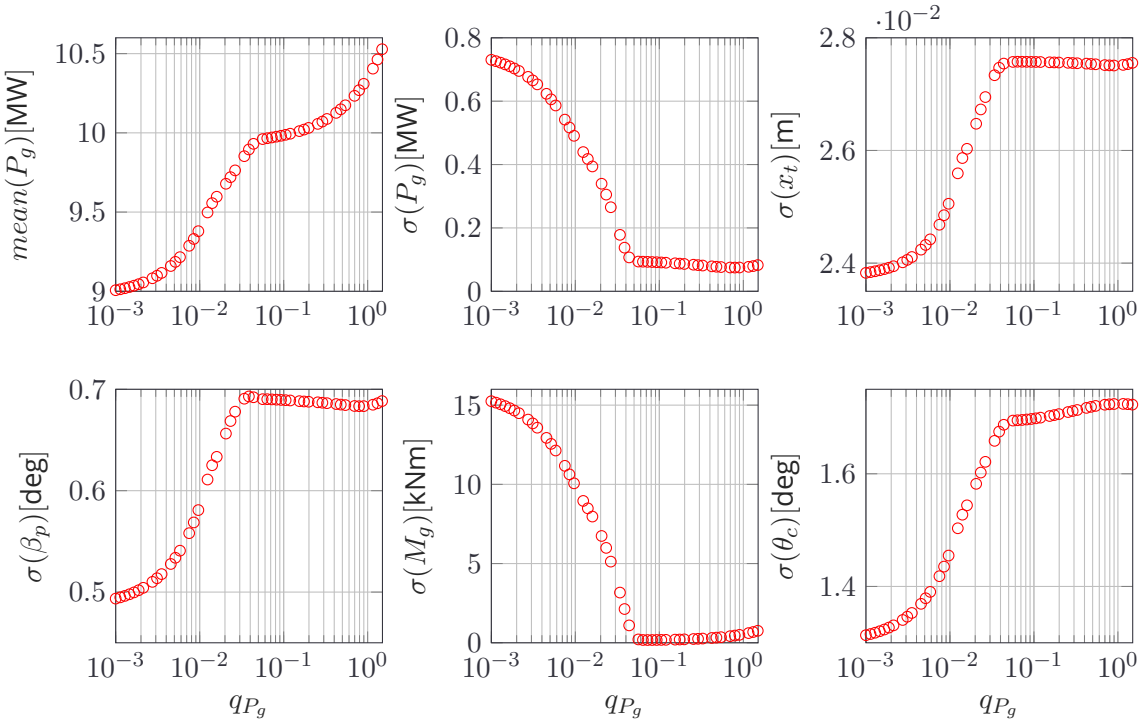


Figure 20: Comparison of different objectives with respect to the generated power weight q_{P_g}.

no change in all objectives expect mean(P_g). The weights below 5×10^{-2} are not reasonable anymore in account on the resulting time courses. The generated energy sinks too much over longer simulation times. The several times occurred trade-off between the two structural loads $\sigma(x_t)$ and $\sigma(\beta_p)$ and the STD of the power generation $\sigma(P_g)$ is clearly visible. Both structural loads decrease and simultaneously the STD of P_g increases. Also interesting is the observation, that mean(P_g), the structural objectives and the actuation load correlate.

4.4.1.5 Optimization of the control input weights

The first simulations are run in the sample design range between 5×10^{-1} and 5×10^{2}. Smaller values are not possible, because the optimizer cannot solve the problem anymore. By discovering the 3D plots of the simulation results, the most significant observation is the invariance

of the weight of the generator torque q_{M_g}, it has almost no influence on the objectives. The second observation is the significant deterioration of the structural and actuation objectives, when the blade pitch weight is higher than 1×10^1. The same is observed with the STD of the generated power. Therefore the second simulation set is run with the parameter range of q_{θ_c} between 1.5 and 10. The resulting two dimensional plots of all objectives are shown in figure 21. With a higher weight of the blade pitch actuation than the default value of one, a lower struc-

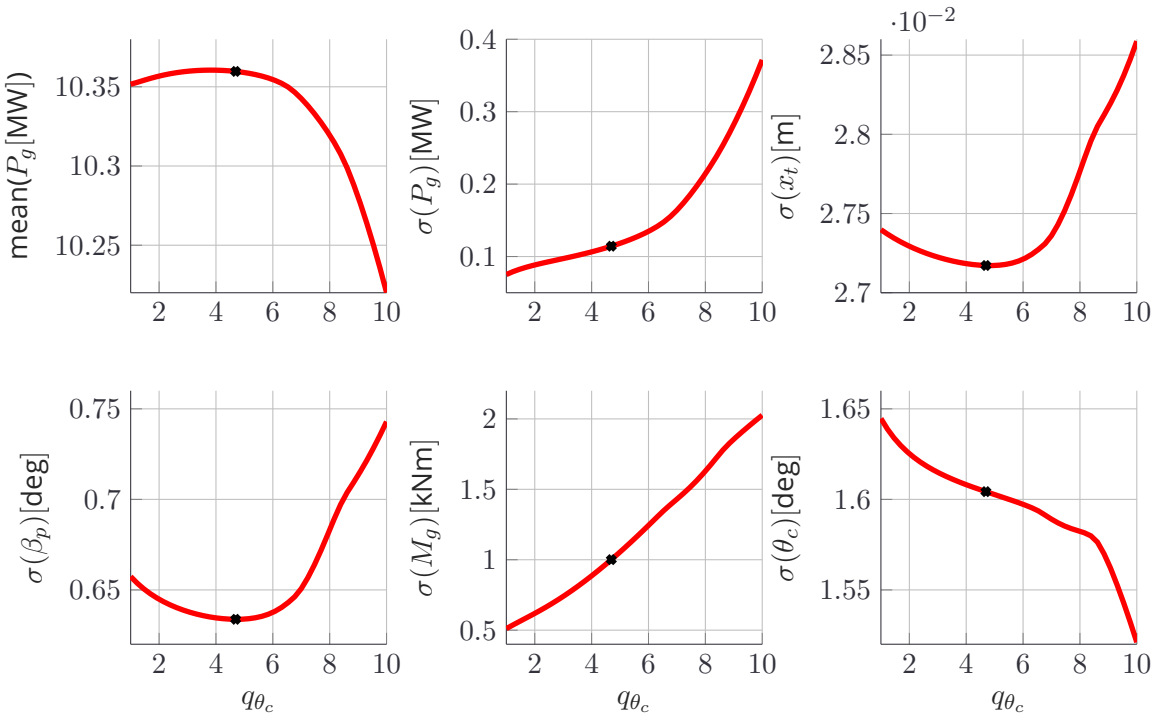


Figure 21: Comparison of different objectives with respect to the blade pitch weight q_{θ_c}. Plus the minimum of the norm of the STD of the two structural fatigues $\sigma(x_t)$ and $\sigma(\beta_p)$.

tural load can be achieved. The minimum of the norm of the two structural load objectives is minimized

$$arg\min_{q_{\theta_c}} \| \begin{pmatrix} \sigma(x_t) \\ \sigma(\beta_p) \end{pmatrix} \| = 4.6923. \tag{4.38}$$

This chosen weight is marked with the black dot in the diagrams. Again, the trade-off between the STD of P_g and the structural loads is visible. Between 1.5 and approximately 5, the fatigue loads of the tower and the pitch actuation decrease, whereas $\sigma(P_g)$ increases.

Also interesting is the dependence of the control input weights on the solving time of each optimization step. Or rather, the influence of the blade pitch weight q_{θ_c} on the solving time. Again the generator torque parameter has almost no influence. Figure 22 shows the mean of the solving time with respect to different weighting parameters. So far, the solving time amounts 70 ms for one optimization. The figure shows a significant decrease between the default weight 1 and 4. The above described parameter change reduces the average solving time of an optimization about 50% to 36 ms. This points out the impact of the weights on the solving time and the importance of the analysis of the time behavior.

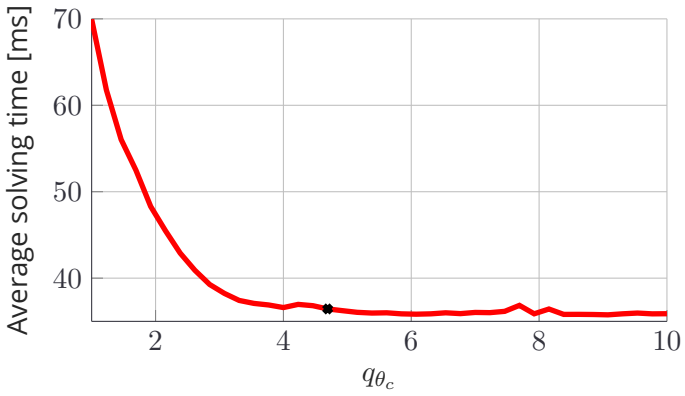


Figure 22: Comparison of different objectives with respect to the blade pitch weight q_{θ_c}. Plus the minimum of the norm of the STD of the two structural fatigues $\sigma(x_t)$ and $\sigma(\beta_p)$.

4.4.2 Resulting weights and simulation

The last section 4.4.1 have given an overview of the impact of all weighting parameters. In specific use cases the decision maker has to choose different weights, according to the desired objectives. A knowledge discovery of the influence of the different weights is given and also a restriction in which areas a optimization process is reasonable. The resulting weights are listed in table 18. As described in the beginning of the optimization process in section 4.4, the shifting

Table 18: Optimized weighting parameters.

q_{P_g}	q_{M_g}	q_{θ_c}	q_{β_p}	q_{x_t}	$q_{S_{\Omega_r}}$	$q_{S_{M_g}}$
1	1	4.6923	1	1	205.3525	205.3525

parameter T is set to zero, neutral position, and the smoothing factor to one. The initial idea of splitting the cost function into a maximizing generated power part and into an alleviating structural fatigue part has to be investigated further. On the one hand, the optimization process reveals, that the impact of different weighting parameters is very complex and on the other hand, that it cannot be analyzed in a proper way, if only above rated simulations are conducted. Concerning the last argument, region two would be more suitable to investigate the trade-off between maximization of the power and alleviating the loads, because only a limited amount of power is available. In region three, the difficulty is not to generate more power, rather to keep it constant. Because of these two reasons, the deployment of the continuous factors α and β, has to be developed further.

5 Results

This chapter presents the simulation results of corresponding evaluations and comparisons to the baseline controller.

5.1 General results and comparison

The control set-up from the last chapter, i.e. the values in table 18 and the described solver option, is utilized for the following analysis. To demonstrate the long-term stability and performance, a simulation of one hour has been conducted and is presented.

5.1.1 Simulation result

In figure 23 the comparison of the baseline controller and the EMPC is shown. Due to the limited space, only a simulation length of 500 s is extracted out of the longer simulation. The set-up of the individual plots are the same as in the comparison of the LQR and the baseline controller. The first two subplots show the identical disturbances. In the third subplot the platform surge is presented. The max/min values are equally high, however the phase is shifted, which is a result of the different control inputs of the two controllers. The next subplot plots the platform heave, where no visible difference is seen. In the fifth subplot, a smaller max/min value of the platform pitch of the EMPC case is observed. Roughly speaking, both time series behave similarly. The same applies for the the tower top displacement in the next subplot. Sometimes, the position differs slightly, but altogether the performance is comparable. In addition to the visual observation, further detailed analysis is presented later. In the last three subplots, the two different time series differ more. In comparison to the baseline controller, which keeps the generator torque constant, the control actuation of the EMPC varies about its rated value slightly at above rated operation range. The pitch actuation of the EMPC behaves in the same way as the wind disturbance, but in a smoothed manner. Because of the perfect wind preview assumption, there is no time delay between the actuation of the blade pitch and the wind disturbance. Therefore the adjustments to the changing wind can perfectly be applied, whereas the baseline controller has a delayed reaction, because no preview information is available. This difference can be seen clearly between the time step 1100 s and 1250 s. A result of this different control inputs is visible in the rotational speed and generated power plot. While the generated power using EMPC stays relatively constant, the power with the baseline controller falls on 8 MW at 1200 s. Here the superior behavior of the EMPC is clearly visible.

Table 19 lists the resulting performance of the baseline controller and that of the EMPC. In the last column the relative improvement, defined as the quotient of them, is presented. The EMPC achieves a 3.28% higher average of the generated power, corresponding to 328.4 kW. The most significant improvement of the EMPC in comparison to the baseline controller is the reduction of $\sigma(P_g)$. The STD of the generated power is decreased by 70% to 0.21 MW. Besides, the higher power and its better quality, both structural fatigue loads are reduced by approximately 15%. Another difference is the generator torque. Regarding the control input actuation, the EMPC requires a 7% higher STD of the blade pitch actuation. Altogether, the EMPC achieves better resulting performance than the baseline controller. On the one hand, the improved generated power and improved structural loads and on the other hand the deterioration of the blade pitch actuation.

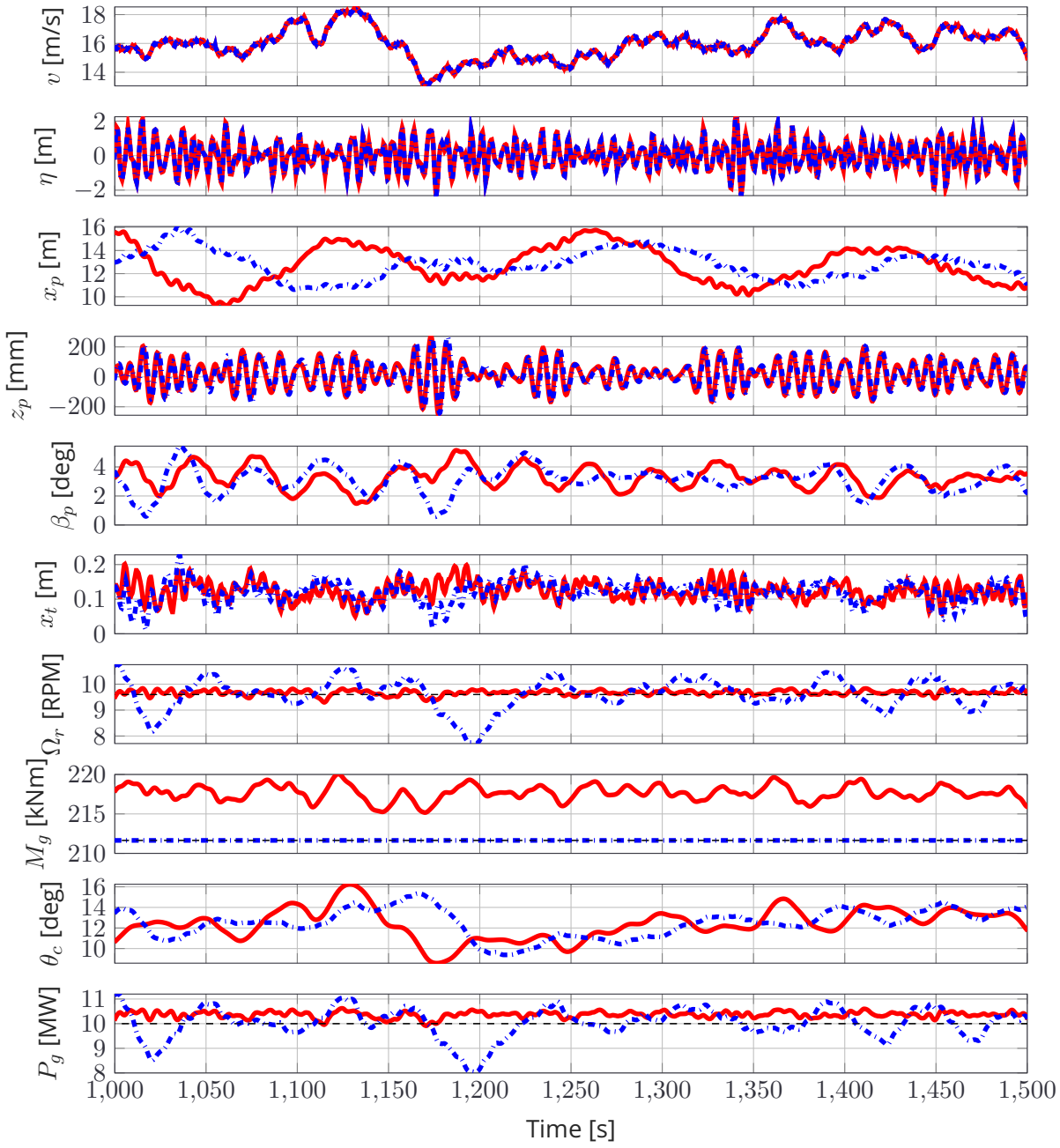


Figure 23: Comparison of the baseline controller and the EMPC. Time section from 1000 to 1500 s. EMPC is marked in (—) and the baseline controller in (- -).

Table 19: Comparison of the baseline controller and the EMPC results of an one hour long time simulation.

Objective	Unit	BC	EMPC	EMPC / BC
$mean(P_g)$	[MW]	10.0045	10.3329	103.28%
$\sigma(P_g)$	[MW]	0.6970	0.2098	30.1%
$\sigma(x_t)$	[m]	0.0349	0.0306	87.68%
$\sigma(\beta_p)$	[deg]	1.0140	0.8685	85.65%
$\sigma(M_g)$	[kNm]	0	1.1918	-
$\sigma(\theta_c)$	[deg]	1.8282	1.9573	107.06%

Another interesting result is the comparison of the maximum values of certain variables. The rotor speed and the generator torque of the wind turbine have to be treated carefully, otherwise there could be some damage to the generator. As above described, the maximum mean values of both variables are limited to 5% above the rated value. In case this upper border is defined to high, with the EMPC concept, it would be no problem to decrease the mean or the overshoot of the variables. Nevertheless, with this simplified assumption in mind, the maximum overshoot of the rotor speed is 4%. Whereas the maximum overshoot of the baseline controller of Ω_r is around 20%. The maximum power overshoot of the EMPC is about 8%, because the variation of the generator torque is a part of the product. All the values are summarized in table 20. In general, the generator torque and the rotor speed could be regulated to

Table 20: Comparison of the baseline controller and the EMPC results regarding the overshoots.

Overshoot	Unit	BC / Rated	EMPC / Rated
$max(\Omega_r)/\Omega_{r,rated}$	[RPM]	119.59%	103.94%
$max(M_g)/M_{g,rated}$	[kNm]	100.00%	104.59%
$max(P_g)/P_{g,rated}$	[MW]	119.59%	108.34%

be closer to their rated value. Consequently, the generated power would be closer to the rated value as well. However, the wind turbine has to bear these higher loads, because of the unsuppressible and high overshoots of the baseline controller. This is the reason for the intentional overrated rotor speed and generator torque of the more accurate EMPC. The concept enables a wide range of adjustments.

For control techniques like MPC, the computational time is especially important. The mean solving time for one optimization step amounts 35.9 ms. In comparison, the step time of the controller is 250 ms. Therefore, the optimization itself requires only 14% of the available time. In figure 24 on the left hand side, the histogram of the amount of iterations is plotted. The optimization problem requires these amounts of iterations to converge, or rather to fall below defined accuracy borders. The right hand side shows the histogram of the solving times.

5.2 Influence of maximizing the generated energy

In the complete optimization process of the chapter 4 and the presented results in the chapter 4, where the focus was on staying below defined limits for $mean(\Omega_r)$ and $mean(M_g)$. While these requirements are satisfied, the structural loads are minimized as much as possible. Because, the chosen operating point of $v = 16\,\mathrm{m/s}$ is above rated, there is no problem in maximiz-

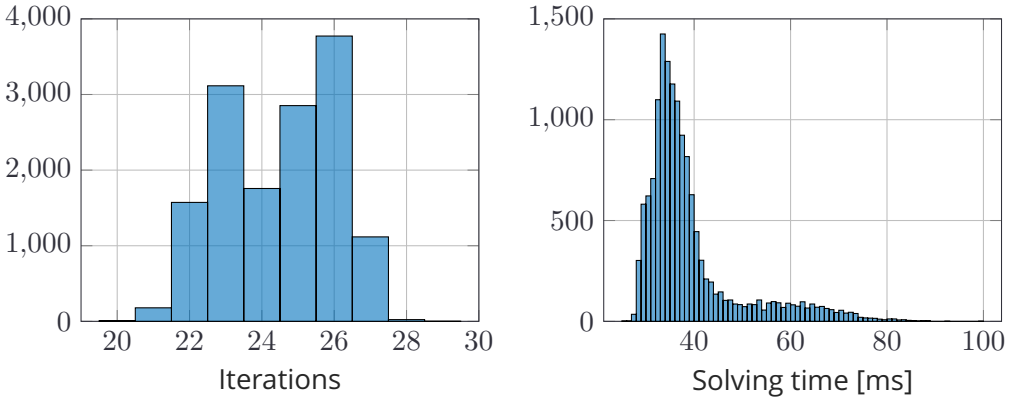


Figure 24: On the left hand side, the histogram of the number of iterations is plotted. On the right hand side, the histogram of the solving time is visualized.

ing the generated power. However, as explained above, there are other limitations, such as the overheating of the generator. To analyze the effects and the ability of different cost functions, which comes along the EMPC method, to change between the use cases of maximizing power and mitigating structural loads, region two is more suitable. In the below rated operational region, only a limited power is available. Therefore, the cost function can be defined in such a way, that as much power as possible is generated. Nevertheless, in order to have two different comparable cost functions in the above rated region, another use case is presented, within the defined ranges for the rotor speed and generator torque.

This set-up, allows a higher $mean(P_g)$ through settling the penalties of the slack variables to

$$q_{S_{\Omega_r}} = 45.3158, \quad q_{S_{M_g}} = 124.0938. \tag{5.1}$$

A time section of the results is shown in figure 25. The overall simulation length is one hour, however the plot shows the time section from second 1000 to 1500. The reduced weight of the two slack variables, which reduces the penalties of the overshoots of the rotor speed and the generator torque, is directly visible. In all three diagrams, of Ω_r, M_g and P_g, the lower penalties are visible. The time series with lower penalties are higher than the time series with the higher penalties. Another observation is, that not only the mean increases, also the variation is higher. In table 21, the individual objectives are listed and compared. The column of EMPC (a)

Table 21: Comparison of the above defined EMPC (a) and the EMPC with focus on the maximization of the generated power (b).

Objective	Unit	EMPC (a)	EMPC (b)	(b) / (a)
$mean(P_g)$	[MW]	10.3329	10.9465	105.94%
$\sigma(P_g)$	[MW]	0.2098	0.3310	157.77%
$\sigma(x_t)$	[m]	0.0306	0.0321	104.90%
$\sigma(\beta_p)$	[deg]	0.8685	0.9360	107.77%
$\sigma(M_g)$	[kNm]	1.1918	1.9211	161.19%
$\sigma(\theta_c)$	[deg]	1.9573	1.9378	99.00%

set-up provides the objectives of the general, above defined controller and set-up (b) the new controller with focus on the maximization of the generated power.The power is increased as desired, and the mean value of which is approximately 6% higher. However, as expected, not

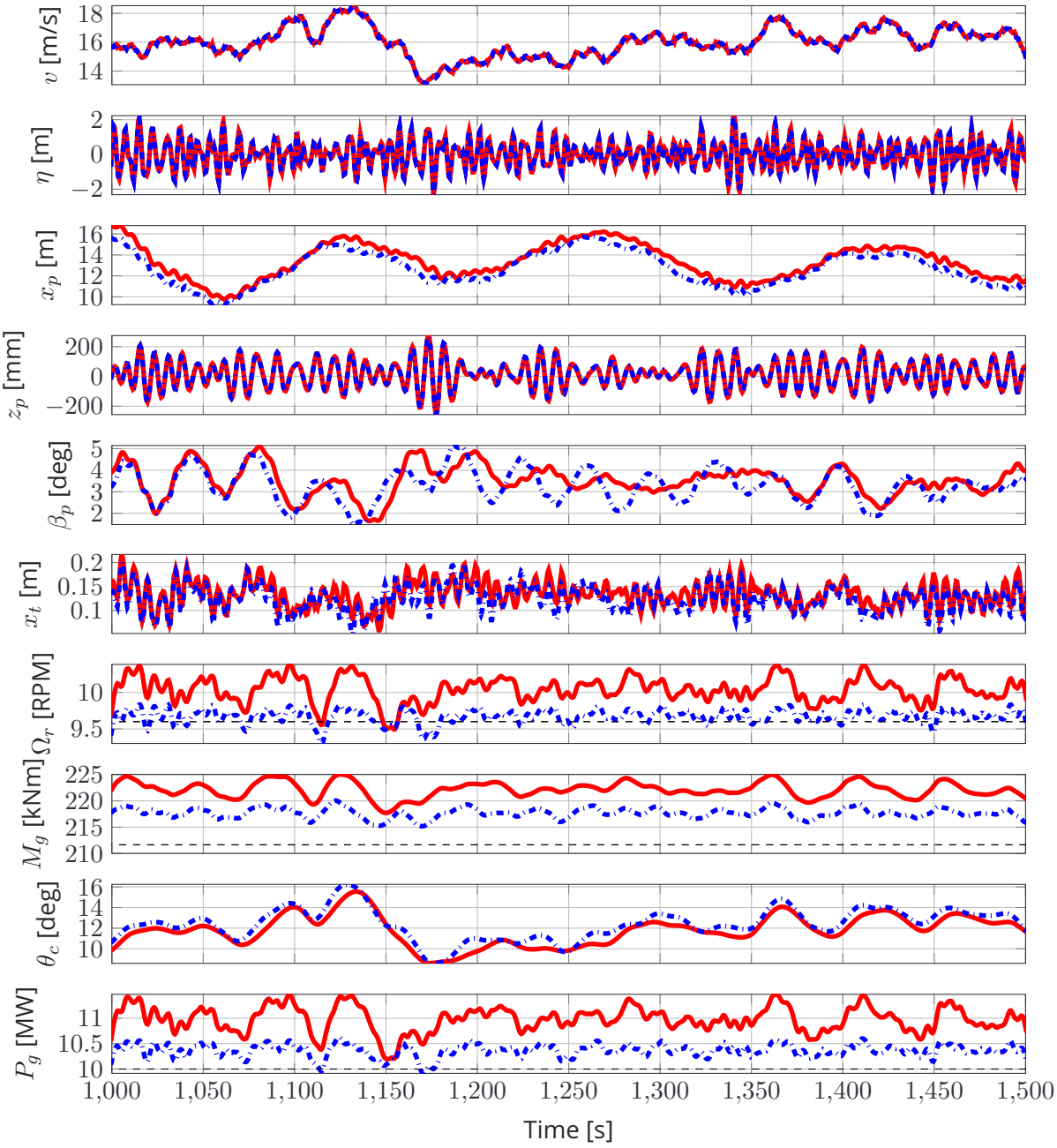


Figure 25: Comparison of two different EMPC set-ups. Time section from 1000 to 1500 s. The in figure 19 presented set-up, is colored in (- -) and the set-up with a higher average of the generated power in (—).

all main goals can be achieved simultaneously, contrarily the structural loads $\sigma(x_t)$ and $\sigma(\beta_p)$ have increased with 5% and 8%. The most significant increase is about 60% of the STD of the generated power. In comparison to the default baseline controller is still only 50%. The STD of the blade pitch actuation stays the same.

In corresponding overshoots are listed in table 22. The last column provides the normalized

Table 22: Comparison of the baseline controller and the two different EMPC set-ups regarding the overshoots. set-up (a) is the general EMPC and set-up (b), the controller with focus on the maximization of the generated power.

Overshoot	Unit	BC / Rated	EMPC (a) / Rated	EMPC (b) / Rated
$max(\Omega_r)/\Omega_{r,rated}$	[RPM]	119.59%	103.94%	104.33%
$max(M_g)/M_{g,rated}$	[kNm]	100.00%	104.59%	104.91%
$max(P_g)/P_{g,rated}$	[MW]	119.59%	108.34%	116.63%

percentage of overshoot of the new controller set-up, the maximum relative Ω_r and M_g are approximately equal to the general EMPC set-up (a). The overshoot of the generated power is risen sharply to 17% above the rated value. The reason for this increase, is firstly the general higher mean of P_g and secondly the higher STD of the power signal. Still, the maximum overshoot of the generated power is lower than the baseline one.

5.3 Imperfect disturbance preview

In this section, two different scenarios with imperfect disturbance preview are simulated and analyzed. In the first scenario, the controller has perfect wind preview, however no wave preview. In the second one, the controller has preview knowledge of both disturbances, however these previews are jammed with noise.

5.3.1 Perfect wind and no wave preview

In this section, the wind preview is assumed to be perfect, whereas the wave preview is completely omitted. This is the case, when only a LIDAR system for the wind preview is mounted on the FOWT. The simulation has a length of one hour again. Table 23 shows the comparison between the EMPC with perfect preview properties (a) and the EMPC with only perfect wind preview and without any wave preview (b). The resulting relation of the average generated power

Table 23: Comparison of the EMPC with complete perfect preview (a) and the EMPC without wave preview (b).

Objective	Unit	EMPC (a)	EMPC (b)	(b) / (a)
$mean(P_g)$	[MW]	10.3329	10.3277	99.95%
$\sigma(P_g)$	[MW]	0.2098	0.2246	107.05%
$\sigma(x_t)$	[m]	0.0306	0.0312	101.96%
$\sigma(\beta_p)$	[deg]	0.8685	0.9716	111.87%
$\sigma(M_g)$	[kNm]	1.1918	1.3273	111.37%
$\sigma(\theta_c)$	[deg]	1.9573	1.9605	100.16%

is almost identical, only 0.05% difference. Also, the actuation loads of the blade pitch and the STD of the tower top position are similar. The most significant change is the 12% deterioration

of $\sigma(\beta_p)$. The maximum overshoots, listed in table 24, are approximately one percent worse, than the complete perfect EMPC. Summarizing, the generated energy stays the same, however

Table 24: Comparison of the EMPC with complete perfect preview (a) and the EMPC without wave preview (b) regarding the overshoots.

Overshoot	Unit	BC/Rated	EMPC (a) / Rated	EMPC (b) / Rated
$max(\Omega_r)/\Omega_{r,rated}$	[RPM]	119.59%	103.94%	104.04%
$max(M_g)/M_{g,rated}$	[kNm]	100.00%	104.59%	105.48%
$max(P_g)/P_{g,rated}$	[MW]	119.59%	108.34%	109.37%

there are negative impacts of the omitted wave preview on the power quality and the structural loads. Especially on the platform pitch variation.

5.3.2 Noisy disturbance preview

The perfect preview assumptions defined so far, is not realistic. There are uncertainties in every model and measurements, which distort the original signal [24]. The disturbance preview via e.g. LIDAR has uncertainties as well. In [3], the uncertainty is determined as $\sigma(v_0)_{noise} = 0.16\,\mathrm{m/s}$. The STD of the original rotor effective wind speed amounts to $\sigma(v_0) = 1.3177\,\mathrm{m/s}$. Thus, the STD of the added noise is approximately 0.12% of $\sigma(v_0)$. This relation is transferred to the wave disturbance in the same way. The original STD is $\sigma(\eta) = 0.8147\,\mathrm{m}$ and the additional computed STD of the wave noise is $\sigma(\eta)_{noise} = 0.12 \cdot \sigma(\eta) = 0.0978\,\mathrm{m}$. The noise level of the wave preview is a simplified assumption and transferred of the measured noise level of the wind. The wave noise level is dependent on the measurement technique as well.

Both individual noises are added with a normal distribution and the corresponding STD. A time section of the both disturbance signals is shown in figure 26. The plots visualize the original signal plus the noisy disturbance.

The controller is tested with these more realistic distorted disturbances as preview signals. The same framework is used as in the previous simulations. The original disturbances are applied to the simulation model, whereas the disturbances with extra noise are used as input for the preview. In table 25 are the results and the comparison to the perfect preview EMPC listed. The

Table 25: Comparison of the EMPC with complete perfect preview (a) and the EMPC noisy preview (b).

Objective	Unit	EMPC (a)	EMPC (b)	(b)/(a)
$mean(P_g)$	[MW]	10.3329	10.3299	99.97%
$\sigma(P_g)$	[MW]	0.2098	0.2128	101.43%
$\sigma(x_t)$	[m]	0.0306	0.0307	100.33%
$\sigma(\beta_p)$	[deg]	0.8685	0.8707	100.25%
$\sigma(M_g)$	[kNm]	1.1918	1.2046	101.07%
$\sigma(\theta_c)$	[deg]	1.9573	1.9604	100.16%

largest difference to the perfect EMPC is, that the STD of the generated power is 1.43% higher. All remaining differences are smaller than this percentage. The overshoots of the EMPC with noisy preview is shown in table 26. The overshoots are almost identical to the controller with perfect preview. In summary, preview signals with the defined amount of noise do not have significant effect on the performance of the controller.

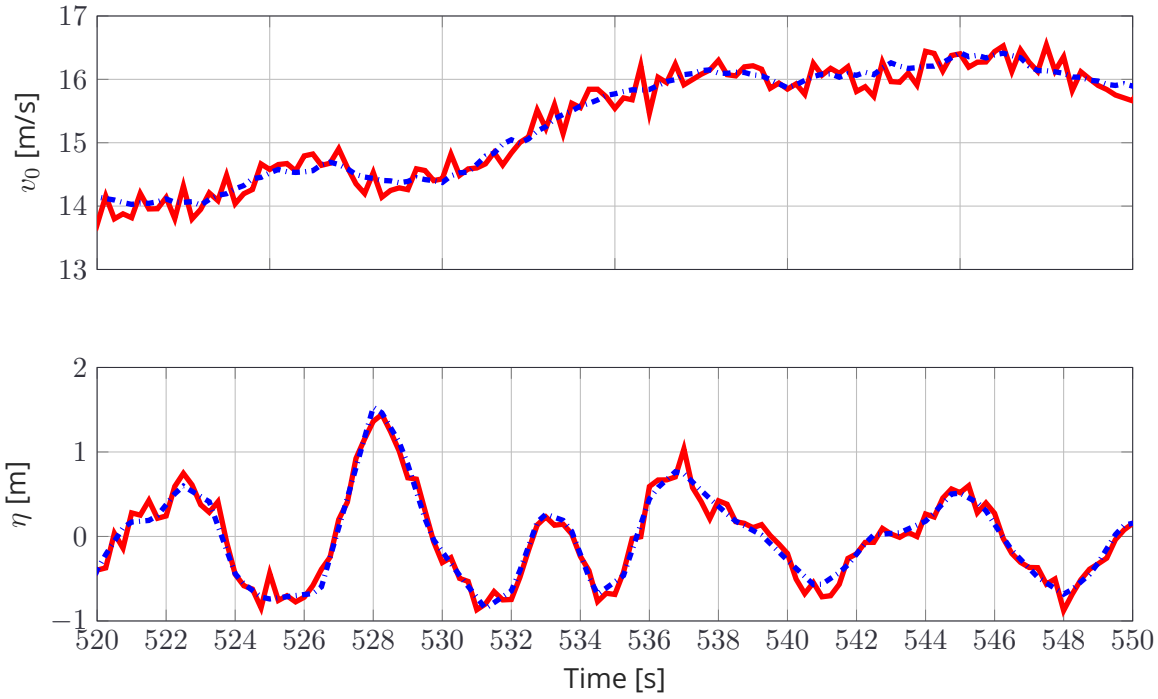


Figure 26: The upper subplot visualizes the original wind disturbance (- - -) and the signal plus the overlaid noise (—). The lower subplot shows the same set-up for the wave disturbance.

Table 26: Comparison of the EMPC with complete perfect preview (a) and the EMPC with noisy preview (c) regarding the overshoots.

Overshoot	Unit	BC/Rated	EMPC (a) / Rated	EMPC (c) / Rated
$max(\Omega_r)/\Omega_{r,rated}$	[RPM]	119.59%	103.94%	103.79%
$max(M_g)/M_{g,rated}$	[kNm]	100.00%	104.59%	104.56%
$max(P_g)/P_{g,rated}$	[MW]	119.59%	108.34%	108.04%

6 Summary

6.1 Conclusion

In this work the reference wind turbine, a DTU 10 MW offshore wind turbine, with a Olav Olsen OO-Star Wind Floater as platform is presented. The corresponding simplified low order model of a floating offshore wind turbine is shown. The linear model derived out of this is used as the controller design model, or rather as internal controller model. This derived linear model is compared to the original nonlinear model to verify it. It is done with a deterministic wind gust and with stochastic wind and wave disturbances.

After this modeling part, the state-of-the art baseline controllers are presented, with its benefits and downsides. The next covered controller introduced the class of optimal controllers, namely the LQR. Here as well, the benefits and downsides are explained. Also, the current state of the LQ controller is shown by a closed loop simulation. This result is compared to the standard baseline controller, which points out the potential of optimal controllers. The functionality of the consecutive MPC and the application of this control methodology to floating offshore wind turbines is explained. As well, the advantages and disadvantages of the Model Predictive Control technique and the different types explained. One specific type, the Economic MPC, which is the core of this work, is introduced. Also, the application of the controller on FOWT. This is done by defining the proper cost function and the constraints of the turbine. The general set-up is adapted to the linearized model and compared to the classical MPC.

With this theoretically complete optimization problem, the next step was the adjustment to a solver, thus it can be implemented. But first different types of solvers are presented and compared. A discretization of the problem has to be conducted first, hence that the set-up can be implemented with the chosen solver FORCES PRO. To run the optimization algorithm successfully and with an acceptable performance, the cost function and the complete system has to be scaled. The remaining weights in the cost function have to be determined in such a way, that the resulting objectives are as good as possible. Several trade-offs had to be decided. Some trade-offs could not be decided in an optimal way, because additional knowledge of the applied use case or more exactly defined requirements for certain use cases have to be known. In general, knowledge of the behavior of the complex FOWT system is discovered, with respect to the different penalty parameters and their weighting. Parameters are determined, which minimize the structural loads, while holding the generated power a certain amount above the rated value. To achieve a high amount of generated power comes easy, because the turbines operates above rated wind speed. Therefore, rather the aims are to hold the power as constant as possible, around a slightly higher rated value. The maximizing of the power can be analyzed and developed in a better way in region two, below rated.

In the case of perfect wind and wave preview plus perfect state estimation, the EMPC achieves to 3% higher mean of the generated power, while reducing the standard deviation up to 70%. The last objective is the most significant one. Additionally, to these two improved objectives, the structural load could be decreased to about 15%, both the tower top variation and the movement of the platform pitch. The only deterioration is the higher blade pitch actuation. Also, the maximum overshoots of the rotor speed could be improved up to 16%. These results are computed in real time, or in other words, the mean solving time of one optimization

requires 36 ms, while the step time of the optimization problem is 250 ms. Therefore, the solving time requires only 14% of the available time. To have a comparable second cost function, higher power generation was allowed. Thus, the objectives can be compared to each other. The results are higher structural loads, which are understandable, however also a significant deterioration of the STD of the power generation.

To make the assumption more realistic, imperfect disturbance preview is applied and simulated. The first test has perfect wind preview and no wave preview at all. This results in a higher standard deviation of the generated power and in higher structural load of the platform pitch. The second tests covers realistic noisy disturbance preview. Both disturbance signals corrupted with noise. The resulting objectives have declined, however only up to a slightly amount.

Altogether, this new control technique enables the online involvement of constraints and the integration of disturbance preview signals to an optimal control actions. Also, the design of the cost function is simpler in comparison to the classical MPC. It enables a more straight forward controller design, it is possible to formulate the maximization of the generated power in a more direct way. This results in a better closed loop performance, because the objectives are directly in the cost function. These benefits and the results based on this are shown in this work successfully.

In summary, the new control methodology Economic Model Predictive Control has been adapted to a floating offshore wind turbine model. This optimization model has been adjusted and implemented with a highly efficient solver successfully. The real time capable controller has achieved to provide significant better objectives than the standard baseline controller. The EMPC decreases the structural loads, while minimizing the variation of the generated power with a higher average power. Even with noisy disturbance preview.

6.2 Outlook

The derivation and implementation of an EMPC for floating offshore wind turbines is described and presented in this work. The results outperform the objectives of the standard baseline controller, also with noisy disturbance preview. Additionally, it is shown, that the EMPC has the potential to be adaptable for different use cases, by which the high computational effort is manageable and real time capable.

However, the EMPC is design on the basis of a linear FOWT model. This mean, a linearization of the nonlinear original model has been necessary. Thus, this controller is only valid around a specific operation point, in this case the above rated wind speed of 16 m/s. Out of this, two problems occur. Firstly, the developed EMPC is for most of the operation range (from 4 m/s to 25 m/s) not valid. And secondly, the controller is only tested in the third region, in the above rated region. About the behavior in the below rated region or in transition between two regions is nothing known. Therefore, different solutions are possible. Firstly, the development of a gain-scheduling or adaptive EMPC. For these cases, the design model is still linear, e.g for the gain-scheduling option, there are several, offline computed, linear models used to cover the complete range of operation. Whereas the adaptive EMPC would compute a variant linear model online, as soon as the environmental conditions change. Both options are only approximations of the original nonlinear model. Thus, a fundamentally different solving approach is to use the original nonlinear model as design model. Then the ENMPC operates as good as the nonlinear model is, over the whole range of operation, without any approximation. There are already developed on the one hand classical NMPC for FOWT [26] or on the other hand ENMPC for onshore wind turbines [11], however not yet an ENMPC for floating offshore wind turbines. These options would represent three approaches to develop a controller, which operates over the whole range.

In this work, the objective of the standard deviation of the generated power is taken into account. However, there has no equivalent penalty in the cost function been defined. The derivation of the generated power leads to the requirements, that the derivation of the generator torque and the derivation of the rotor speed is available. The derivation of the torque is already available and also already penalized. However, the derivation of the rotor speed would have to be introduced, thus it can be penalized. The result would be less variation in the rotor speed, which would lead to less even less variation in the generated power signal. There is potential to further improve the power signal, or in other words, the provided power quality of the wind turbine.

So far, perfect state estimation is assumed, which is not realistic. Therefore, an observer has to be developed. There are simpler observers, like a standard Luenberger observer, or optimal linear observer, like it is used in the LQG. Or e.g. in the field of LIDAR research nonlinear observer comes into play. However, the equivalent pendant to the receding horizon Model Predictive Control technique is the moving horizon estimator (MHE). The functionality is similar to the MPC, there is also a finite cost function with subject to the system model, which is steadily moved with every iteration. In every iteration this optimization problem is solved, however the result is no control action, but rather the optimal estimated state. The development of an observer enables the application of the EMPC in field tests.

A critical part of this controller is probably the robustness. The related controller type LQR has robustness problems, as well. A first step to investigate the robustness of the controller, is to

introduce model errors in the internal model of the EMPC. Especially, in hard to identify parameters like the damping coefficients. To analyze the impact of model errors regarding stability and performance is an important step. Which uncertainties of model parameters are allowed to maintain stability? If there is less robustness or if more robustness is required, robust or stochastic MPC can be implemented. Additional, in account on the robustness are further simulations with not simplified models as simulation models, like FAST. This more realistic, higher dimensional simulation environments, spawns new phenomenons, possible instabilities and lack of robustness. Also, different wind and wave disturbances, in all possible ways, should be tested, to gain more knowledge of the behavior of the controller.

There are MPC for FOWT developed, however not yet for this combination of the wind turbine and floater, which are used in this work. Thus, it is not possible to compare the developed EMPC to a classical MPC, instead the standard baseline controller is used for comparing. It would be interesting to compare the objectives to the classical MPC, also the solving times and in general the behavior. How is the effect of the more direct way of formulating the cost function, like it is possible in the EMPC case, in comparison to the indirect way of the normal MPC?

In this work, some decision making tools are started to be applied. The space-filling methods for the weighting parameter investigation are a good basis for further multi-objective optimization research. One possibility is the use of Pareto front based algorithm. A more structured approach is possible with such methods. With more knowledge of the requirements of the FOWT, such, e.g. evolutionary algorithms, can be used effectively.

In general the new control technique of EMPC offers new possibilities in controlling FOWTs. The big potential is the possible tuning free and self-adjusting cost function. Under the conditions of available models, which map the individual structural loads and the generated energy into terms of monetary and comparable units. Then it is possible to take the time varying energy market, the weather and the required power demand into account. And all is adapted automatically and weigh up by the controller and its time varying cost function.

References

[1] D. O. O. AS. *OO-Star Wind Floater Semi 10MW concept*. [Online; accessed January 20, 2020]. URL: https://www.olavolsen.no/en/business-areas/renewable-energy.

[2] C. Bak, F. Zahle, R. Bitsche, T. Kim, A. Yde, L. Henriksen, M. Hansen, J. Blasques, M. Gaunaa, and A. Natarajan. *The DTU 10-MW Reference Wind Turbine*. English. 2013.

[3] H. F. Borraccino A Schlipf D and W. R. In: *Wind Energy Science 2* (2017), 269–283.

[4] S. P. V. Boyd and L. V. Vandenberghe, eds. *Convex optimization*. Englisch. 6th pr. with corr. Cambridge [u.a.]: Cambridge Univ. Press, 2008, XIII, 716 Seiten. ISBN: 0-521-83378-7.

[5] S. Christiansen, S. M. Tabatabaeipour, T. Bak, and T. Knudsen. "Wave disturbance reduction of a floating wind turbine using a reference model-based predictive control." In: *2013 American Control Conference*. 2013, pp. 2214–2219.

[6] A. Domahidi and J. Jerez. *FORCES Professional*. Embotech AG. 2014–2019. URL: https://embotech.com/FORCES-Pro".

[7] M. Ellis, J. Liu, and P. D. Christofides, eds. *Economic Model Predictive Control : Theory, Formulations and Chemical Process Applications*. Englisch. Springer International Publishing AG, 2017, XXIV, 292 p. ISBN: 3319411071.

[8] P. A. Fleming, I. Pineda, M. Rossetti, A. D. Wright, and D. Arora. "Evaluating Methods for Control of an Offshore Floating Turbine." In: (2014).

[9] S. Gros. "An economic NMPC formulation for Wind Turbine Control." In: Dec. 2013, pp. 1001–1006. ISBN: 978-1-4673-5717-3.

[10] S. Gros, R. Quirynen, and M. Diehl. "An Improved Real-time Economic NMPC Scheme for Wind Turbine Control Using Spline-Interpolated Aerodynamic Coefficients." In: vol. 2015. Dec. 2014.

[11] S. Gros and A. Schild. "Real-time economic nonlinear model predictive control for wind turbine control." In: *International Journal of Control* 90.12 (2016), pp. 2799–2812. ISSN: 0020-7179.

[12] S. Gros, M. Vukov, and M. Diehl. "A real-time MHE and NMPC scheme for wind turbine control." In: Dec. 2013, pp. 1007–1012. ISBN: 978-1-4673-5717-3.

[13] B. Houska, H. Ferreau, and M. Diehl. "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization." In: *Optimal Control Applications and Methods* 32.3 (2011), pp. 298–312.

[14] T. Hovgaard, S. Boyd, and J. Jorgensen. "Model predictive control for wind power gradients: MPC for wind power gradients." In: *Wind Energy* 18 (Apr. 2014).

[15] ICE. "IEC 61400-3 Wind turbines - Part 3: Design requirements for offshore wind turbines." In:

[16] K. E. Johnson, L. Y. Pao, M. J. Balas, and L. J. Fingersh. "Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture." In: *IEEE Control Systems Magazine* 26.3 (2006), pp. 70–81.

[17] F. Lemmer. "Low-order modeling, controller design and optimization of floating offshore wind turbines." In: (2018). URL: http://dx.doi.org/10.18419/opus-10526.

[18] F. Lemmer, D. Schlipf, and P. W. Cheng. *Control design methods for floating wind turbines for optimal disturbance rejection*. 2016.

[19] F. Lemmer, W. Yu, D. Schlipf, and P. W. Cheng. "Robust gain scheduling baseline controller for floating offshore wind turbines." In: *Wind Energy* 23 (Sept. 2019).

[20] *LIFES50plus*. https://lifes50plus.eu/. [Online; accessed March-2020]. 2020.

[21] D. Matha and R. Ebenhoch. *Comparative Levelized Cost of Energy Analysis*. EERA DeepWind 2015. Accessed April-2020. 2015. URL: https://www.sintef.no/globalassets/project/eera-deepwind-2015/presentations/f/f_matha_univ-stuttgart.pdf.

[22] A. Pegalajar-Jurado, H. Bredmose, M. Borg, J. Straume, T. Landbø, H. Andersen, W. Yu, K. Müller, and F. Lemmer. "State-of-the-art model for the LIFES50+ OO-Star Wind Floater Semi 10MW floating wind turbine." In: *Journal of Physics: Conference Series* 1104 (Oct. 2018), p. 012024.

[23] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, eds. *Model Predictive Control: Theory, Computation, and Design*. Englisch. Nob Hill Publishing, LLC, 2018. ISBN: 9780975937730.

[24] D. Schlipf, M. Koch, and S. Raach. "Modeling Uncertainties of Wind Field Reconstruction Using Lidar." In: *Journal of Physics: Conference Series* 1452 (2020), p. 012088. URL: https://doi.org/10.1088%2F1742-6596%2F1452%2F1%2F012088.

[25] D. Schlipf, D. Schlipf, and M. Kühn. "Nonlinear Model Predictive Control of Wind Turbines Using LIDAR." In: *Wind Energy* 16 (Oct. 2013).

[26] D. Schlipf, F. Sandner, S. Raach, D. Matha, and P. W. Cheng. *Nonlinear model predictive control of floating wind turbines*. 2013.

[27] M. Shaltout, Z. Ma, and D. Chen. "An Economic Model Predictive Control Approach using Convex Optimization for Wind Turbines." In: July 2016.

[28] M. L. Shaltout, Z. Ma, and D. Chen. "An Adaptive Economic Model Predictive Control Approach for Wind Turbines." In: *Journal of Dynamic Systems, Measurement, and Control* 140.5 (2018), p. 2096. ISSN: 0022-0434.

[29] Y. Si and H. R. Karimi. "Gain Scheduling H2 / H-inf Structural Controlof a Floating Wind Turbine." In: *The International Federation of Automatic Control, Cape Town* (2014).

[30] S. V. Skogestad and I. V. Postlethwaite, eds. *Multivariable feedback control: analysis and design*. Englisch. 2. ed., repr. with corr., repr. June 2009. Chichester: Wiley, 2009, XIV, 574 Seiten. ISBN: 0-470-01168-8.

[31] I. The MathWorks. *Model Predictive Control Toolbox*. Natick, Massachusetts, United State, 2019. URL: https://de.mathworks.com/products/mpc.html.

[32] Umweltbundesamt. *Erneuerbare Energien in Zahlen*. 2020, accessed April-2020. URL: https://www.umweltbundesamt.de/themen/klima-energie/erneuerbare-energien/erneuerbare-energien-in-zahlen.

[33] G. Veen, I. Couchman, and R. Bowyer. "Control of floating wind turbines." In: June 2012, pp. 3148–3153. ISBN: 978-1-4577-1095-7.

[34] M. K. Yu W and L. F. *LIFES50+ D4.2 Public Definition of the Two LIFES50+ 10MW Floater Concepts*. 2018.

[35] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari. "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs." In: *International Journal of Control* (2017), pp. 1–1.