Extreme and Fatigue Load Reducing Control for Wind Turbines: A Model Predictive Control Approach using Robust State Constraints

vorgelegt von
Dipl.-Ing.
Arne Körber
geb. in Berlin

von der Fakultät III - Prozesswissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss

Vorsitzender: Prof. Dr. Matthias Kraume
Gutachter: Prof. Dr. Rudibert King
Gutachter: Prof. Dr. Po-Wen Cheng

Tag der wissenschaftlichen Aussprache: 28. April 2014

Berlin 2014

Abstract

The cost of energy for power generated using wind turbines is largely driven by the cost of the wind turbines itself. As many turbine components are dimensioned by the mechanical loads, they need to be able to withstand, reducing those loads through the use of smart control algorithms directly impacts turbine costs. Model Predictive Control (MPC) is an advanced control technique that originated in the process industries but is now used for a large variety of applications. This thesis exemplarily demonstrates the benefits of MPC for wind turbine control and shows how MPC can be employed to reduce both extreme and fatigue loading on wind turbines.

An analytical, nine-state, non-linear model of a wind turbine is developed and its parameters identified via closed-loop system identification. This model is used to design an Extended Kalman Filter for state and wind speed estimation and a Model Predictive Controller. The Model Predictive Controller is continuously linearized, i.e., the plant model is linearized and the optimization problem is formulated as a quadratic program (QP) and solved numerically at each time step. The stability of the linear MPC is ensured through the use of terminal cost and constraint terms in the problem formulation. The controller contains both control and state constraints. Using the example of the rotor overspeed constraint, it is exemplarily demonstrated how state constraints can be robustified against uncertainties in future wind speeds. The developed Model Predictive Controller includes the trajectory of future wind speeds explicitly in the problem formulation. This allows integration of preview signals, for example measured using a turbine mounted Light Detection and Ranging (LiDAR) device, in the controller without increasing complexity and retaining all MPC properties such as optimality and constraint handling.

The performance of the MPC is evaluated in extensive simulations using the aero-elastic simulation tool FAST and the NREL 5 MW reference turbine. Results show that a preview MPC controller significantly outperforms both non-preview MPCs as well as a classically designed baseline controller and significantly reduces extreme and fatigue loads. If no preview information is available, the differences between MPC and the baseline controller are more gradual. However, it is also shown that, due to the use of control and state constraints, the MPC provides benefits especially in non-normal operating conditions such as extreme gusts or turbine fault situations where it reduces loads or helps to keep the turbine online.

It is concluded that MPC is a natural fit for integrating preview signals into the turbine control system. Even if no preview information is available, it offers significant benefits especially in special scenarios. Future research on MPC for wind turbine applications should therefore also consider the case where no LiDAR/ preview is available.

Kurzfassung

Die Energieerzeugungskosten für Strom aus Windenergie werden von den Kosten für die Windkraftanlagen dominiert. Die meisten Anlagenkomponenten werden durch die mechanischen Belastungen, denen sie widerstehen können, müssen dimensioniert Deshalb wirkt sich eine Reduktion dieser Belastungen durch intelligente Anlagenregelungsstrategien direkt auf die Anlagenkosten aus. Modellprädiktive Regelung (engl. Model Predictive Control/ MPC) ist ein fortgeschrittenes Regelungsverfahren, das ursprünglich hauptsächlich in der Prozessindustrie zum Einsatz kam. Inzwischen gibt es allerdings auch Anwendungen in einer Vielzahl anderer Bereiche. Diese Arbeit zeigt exemplarisch, welche Vorteile MPC für die Regelung von Windkraftanlagen bietet und wie sie verwendet werden kann, um sowohl Extrem- als auch Ermüdungslasten an Windkraftanlagen zu reduzieren.

Ein analytisches, nichtlineares Modell einer Windkraftanlage mit neun Zuständen wird entwickelt, wobei die Parameter des Modells anhand des Verhaltens des geschlossenen Regelkreises identifiziert werden. Dieses Modell wird verwendet um einen Erweiterten Kalman-Filter zur Zustands- und Störgrößenschätzung sowie einen Modellprädiktiven Regler zu entwickeln. Dieser Modellprädiktive Regler wird kontinuierlich literarisiert, d.h. in jedem Zeitschritt wird das Streckenmodell neu linearisiert und der Reglerentwurf als Quadratisches Programm (QP) formuliert und numerisch gelöst. Hierbei wird die Stabilität des Reglers durch die Verwendung von finalen Zustandskosten und -beschränkungen sichergestellt. Anhand des Beispiels der Überdrehzahl wird exemplarisch gezeigt, wie Zustandsbeschränkungen gegen Unsicherheiten in der Kenntnis der zukünftigen Windgeschwindigkeiten robustifiziert werden können. Der entwickelte MPC berücksichtigt dabei die zukünftigen Windgeschwindigkeiten explizit in der Reglerproblemstellung. Dadurch können eventuelle vorhandene Vorhersagen der zukünftigen Windgeschwindigkeit zum Beispiel von einem LiDAR (Light Detection and Ranging) System direkt verwendet werden, ohne die Komplexität des Reglers zu erhöhen und unter Beibehaltung aller Vorteile der MPC Regelung wie zum Beispiel der Optimalität und der Berücksichtigung von Beschränkungen.

Die Performance des MPC wird anhand des aero-elastischen Simulationsprograms FAST und der NREL 5 MW Referenzwindkraftanlage untersucht. Die Resultate zeigen, dass ein MPC mit Kenntnis der zukünftigen Windgeschwindigkeit eine deutlich bessere Performance als sowohl ein MPC ohne diese Kenntnisse als auch ein klassisch entworfener Referenzregler hat. Ohne Kenntnisse der zukünftigen Windgeschwindigkeiten sind die Unterschiede zwischen MPC und dem Referenzregler geringer. Allerdings zeigt sich, dass durch die Verwendung der Regel- und Zustandsbeschränkungen, die MPC Regelung gerade in "nicht normalen" Betriebsbedingungen wie zum Beispiel extremen Böen oder Anlagenfehlern Vorteile hat und in diesen Situation die Lasten reduzieren kann bzw. dafür sorgt, dass die Anlage länger am Netz bleibt.

Daraus wird geschlossen, das MPC die natürliche Wahl für den Reglerentwurf ist, falls Informationen über zukünftige Windgeschwindigkeiten vorliegen. Selbst wenn solche Informationen nicht verfügbar sind, bietet MPC immer noch große Vorteile in vielen Betriebsszenarien. Zukünftige Forschung im Bereich der Anwendung von MPC zur Regelung von Windkraftanlagen sollte sich deshalb nicht auf den Fall der Regelung unter der Annahme, dass ein LiDAR oder ähnliches System verfügbar ist, beschränken.

Foreword

This thesis is the result of my time as a doctoral student at the faculty of process sciences of the Technische Universiät Berlin. Foremost, I would like to thank Professor Rudibert King for agreeing to what started out as a vague research proposal several years ago and for all the support I have received since then. Without his guidance, patience, and always helpful feedback this project would not have been possible. Further, I would like to thank Professor Po-Wen Cheng at the University of Stuttgart for agreeing to act as the second reviewer and examiner on short notice. Finally, I would like to thank all colleagues, friends, and collaborators that, through technical discussions, proof-reading or simply encouragement, have contributed and helped me to complete this thesis.

Contents

List of Abbreviations

BEM	Blade Element Momentum
CFD	Computational Fluid Dynamics
CCU	Converter Control Unit
DEL	Damage Equivalent Load
DLL	Dynamic Link Library
DOF	Degree of Freedom
EKF	Extended Kalman Filter
ECG	Extreme Coherent Gust
EOG	Extreme Operating Gust
FEM	Finite Element Method
FRT	Fault Ride Through
HH	Hub Height
IEC	International Electrotechnical Commission
IPC	Individual Pitch Control
LIDAR	Light Detection and Ranging
LPV	Linear Parameter Varying
LTI	Linear Time Invariant
LQR	Linear Quadratic Regulator
LVRT	Low Voltage Ride Through
MIMO	Multi Input Multi Output
MPC	Model Predictive Control(er)
NPP	Normal Power Production
NREL	National Renewable Energy Laboratory
PI	Proportional-Integral(-Controller)
PLC	Programmable Logic Controller
QP	Quadratic Program
RMS	Root Mean Square
SISO	Single-Input-Single-Output
STD	Standard Deviation

List of Symbols

a	scaling constant for effect of blade tip movement on effective wind speed
a_1	blade center of gravity ratio
$\mathbf{A}_c$	inequality constraint matrix
$\mathbf{A}_{tc}$	terminal inequality constraint matrix
C_M	torque coeffcient
C_T	thrust coeffcient
d	vector of disturbance inputs
$\mathbf{D}$	damping matrix
$\mathbf{F}$	force gain matrix
F_A	aerodynamic thrust
F_θ	partial derivative of aerodynamic thrust with respect to θ
F_v	partial derivative of aerodynamic thrust with respect to v
F_ω	partial derivative of aerodynamic thrust with respect to ω
F_T	linearized aerodynamic thrust
J	rotor inertia
J	Cost function
k	quadratic torque control law constant for variable speed operation
K_{fa}	tower excitation parameter in the fore-aft direction
$\mathbf{K}$	unconstrained gain matrix
$\mathbf{K}$	stiffness matrix
L	prediction horizon
l_g	radial position of center of gravity of blade
l_a	radial position of aerodynamic center of blade
$\mathbf{M}$	mass matrix
M_A	aerodynamik torque
m_b	rotor mass
M_G	generator torque
$M_{G,0}$	rated generator torque
$M_{G,C}$	generator torque command

$M_{G,\max}$	maximum generator torque
M_θ	partial derivative of aerodynamic torque with respect to θ
M_v	partial derivative of aerodynamic torque with respect to v
M_ω	partial derivative of aerodynamic torque with respect to ω
n_G	gearbox Ratio
P	electrical power
P	pitch actuator transfer function
P_A	aerodynamic power
P_0	rated power
p_i	additive noise for *i*th model output
$\mathbf{Q}$	output or state weights
$\mathbf{Q}$	model covariance (for EKF)
$q_{i,j}$	entries of **Q**
q_i	additive noise for *i*th model state
R	rotor radius
$\mathbf{R}$	control output weights
$\mathbf{R}$	output covariance (for EKF)
$r_{i,j}$	entries of **R**
s	Laplace variable
t	time
t_s	sampling time
$u_{k\|L}$	control sequence
v	free stream wind speed
$\hat{v}$	estimated wind speed
v_e	effective wind speed
V_f	terminal cost function
v_r	rated wind speed
v_0	linearization wind speed
x	turbine model states
x_a	augmented state vector for EKF
x_b	longitudinal position of blade tip
x_t	longitudinal position of tower top
x_0	states at which system is linearized
ρ	air density
κ	terminal controller
$\omega_{0,b}$	flapwise eigenfrequency of blades
$\omega_{p,0}$	eigenfrequency of pitch actuator system
ω	angular velocity of the rotor

$\omega_{\max}$	maximum rotor speed
ω_0	rated rotor speed
$\omega_{T,0}$	eigenfrequency of tower system
λ	tip speed ratio
θ	pitch angle
θ_0	steady state pitch angle
θ_C	pitch setpoint
ζ_b	relative damping of blade flapwise bending
ζ_p	relative damping of pitch actuator system
ζ_T	relative damping of tower system

List of Figures

List of Tables

1. Introduction

Over the past decade, wind energy has evolved from a niche to a mainstream source of energy in many countries. As part of this trend, there is an ever increasing pressure to reduce the cost of energy generated by wind turbines. Wind energy is increasingly expected to compete with energy from fossil fuel sources on a cost basis and in a subsidiary-free environment. Unlike for most traditional sources of electric energy, the "fuel" of a wind turbine is free. The cost of electricity generated from wind turbines is therefore driven largely by the cost of the wind turbine itself. Since there is little room for improvements in turbine efficiency, wind turbine costs need to be reduced in order to drive their cost of energy down.

A big contributor to the overall turbine cost are the costs for the structural, load bearing components such as the tower, blades, and main shaft. These components are sized by the mechanical load they need to be able to withstand. So any reduction in mechanical loads will directly translate into a reduction in material costs for those components. There is a strong interaction between the turbine control system and the loads the turbine experiences, especially due to the slender structure and associated susceptibility to mechanical oscillations as well as the generally challenging dynamics caused by the strong interaction between the structural dynamics and the aerodynamics. On modern wind turbines, the turbine control system is increasingly used to help mitigate some of these challenges and thereby reduce loads. Generally, the larger the wind turbine, the more pronounced are the dynamic challenges. At the same time, the benefits from any load reduction through use of advanced control algorithms are also higher which is why there is an increasing need for smart controller designs, especially for the very large offshore wind turbines which can have power ratings exceeding 5 MW and rotor diameters exceeding 120 m.

There are two distinct types of mechanical loads: extreme and fatigue loads. Extreme loads are loads that a given component needs to be able to withstand once; while fatigue loads are accumulating over time and threaten to damage the turbine after several years of operation. The design for each turbine component may be driven by either extreme loads, fatigue loads, or a combination of both. Most of the existing research on wind turbine controls focuses on reducing fatigue loads either through the addition of sensors to the turbine or by smartly designing the algorithms for the existing control loops. For example, many modern wind turbines now have some means for sensing rotor imbalance loads and can correct those by individually pitching each of the blades. As a result of the achieved reduction in fatigue loads, the component design is driven more and more by extreme loads.

An important source of extreme loads are rare but extreme gusts occurring while the turbine is in power production operation. One method that has been proposed to reduce those gust loads is to install a forward looking, usually laser-based, sensor on the turbine that is capable of measuring the wind speed several hundred meters ahead of the turbine, thus giving the turbine enough time to adjust itself to a potential incoming gust.

Further extreme loads may occur during so-called fault events. If the turbine spins too fast or if a fault in the electrical network it is connected to is detected, it will generally need to be shut down and extreme loads can occur either prior or during the shut-down process. Any control system that helps to avoid unnecessary faults or that improves the behavior during the shut-down process will therefore also reduce turbine loads or contribute to turbine availability.

Model Predictive Control (MPC) is a modern control method widely employed. While at first it was mainly used in the process industry, it is now exploited in a large variety of control problems. Its basic principle is to use the plant model to predict the output trajectory over a finite horizon at every sampling point given the current state and input information. This information forms the basis to calculate the future control input trajectory as the solution of an optimization problem. As this online optimization requires significant computational power, MPC was initially employed for comparably slow processes where update times in the range of minutes or even hours rather than seconds are required. However, with ever increasing computational power available in real time control systems, MPC is now becoming feasible for much faster systems as well. With the clear trend towards increasing wind turbine dimensions, the costs associated with using advanced controls methods, such as MPC, become smaller in relation to the potential benefits:

- Direct MIMO Formulation: The wind turbine problem is a multi-input-multi-output problem (MIMO). Several actuators, such as torque and pitch, need to react on a multitude of measured variables, such as speed and power, in a coordinated fashion. MPC allows to directly design a single controller for the entire problem instead of having to design individual controllers using decoupling or cascading strategies.

- Trade-offs: There are several conflicting controller objectives in wind turbine control. For example, tight control of the rotor speed usually requires fast pitch actuation. This, however, will lead to increased tower motion and loads. The controller will need to be tuned to provide an acceptable trade-off between tower loads and speed control. Such a tuning trade-off can be performed using most controllers, but MPC allows a direct control over these kinds of trade-off and allows a tuning in terms of the quantities that are relevant for the turbine design.

- Constraint Handling: Wind turbine control is inherently constrained. For example, the pitch rates are usually limited by the maximum power of the pitch motors, while the power electronics limit the maximum allowable generator torque rates. There are also hard limits on some of the controlled variables such as generator speed. As too high rotor speeds can damage the turbine, there are usually safety systems that will shut down the entire turbine if the speed rises above a certain threshold. For any controller, it is therefore crucial to maintain the speed below these trigger values. Using MPC, these control and state constraints can be included in the problem formulation and the controller will not lose optimality when operating on constraints. The inclusion of the constraints is especially beneficial in situations where the turbines routinely hit either the control or state constraints, such as gust or shut-down type events.
- Preview Control: As described, one of the most promising candidates for improving wind turbine controller performance with respect to extreme loads is the use of upstream wind speed measurements. Using MPC, this type of control, also known as preview control, can be easily integrated in the feedback controller without adding complexity, thus optimality and constraint handling are not lost when operating with a combined feedback-feedforward controller.

This thesis explores these potential benefits of applying MPC to the wind turbine control problem. It details how a Model Predictive Controller for the full load operation of a large, variable speed, pitch-controlled wind turbine can be designed. It particularly focuses on two of the MPC benefits: It exemplarily shows how state constraints can be included robustly in the MPC using the case of the overspeed trigger and analyzes the effect on the turbine behavior. It further shows how preview measurements can be included directly in the MPC formulation and its effects.

The thesis is organized as follows: Chapter two provides the necessary background information on wind turbine control, system simulation, and MPC. Chapter three examines the prior research that has been conducted in this area. In chapter four, a simplified turbine model is developed and analyzed which is then used in chapter five as the basis for the actual design of the MPC. Finally, in chapter six, extensive simulation results comparing the MPC with a traditionally designed baseline controller in various scenarios are presented.

2. Background

This chapter introduces the key concepts that are relevant for the controller design and the simulation results that are presented in the later chapters. The first part provides a short overview over the control system design of modern wind turbines and outlines some of the design challenges and commonly used design techniques. The second part introduces the system simulations of wind turbines that serve as the basis for the results presented in chapter 6. Finally, Model Predictive Control and its mathematical formulation in state-space is introduced.

2.1. Control of Wind Turbines

There are several fundamentally different turbine control concepts and in the early days of the wind turbine industry, the main choice to be made when designing a wind turbine controller was the control concept itself. Today however, almost all large turbines that are being built are pitch controlled variable speed turbines. These wind turbines share two distinct features which define their basic control strategy. Their electrical system allows a variable speed operation and the torque demand at the generator can be directly controlled through a frequency converter. Secondly, they can shed power by rotating the blades about their longitudinal axis, also called pitching. While some textbooks (e.g., [26]) still treat the fundamental design choices, the focus of current research on wind turbine controls has shifted to the question of how to best control these variable speed, pitch controlled turbines. With the choice of the basic control concept settled, this section details the control problem for these turbines and introduces the basic strategies that are widely used and serve as a baseline for much of the research on advanced wind turbine control methods.

2.1.1. Problem Description

The primary objective of wind turbine control is to maximize the energy that is extracted from the wind while maintaining the turbine within its operational limits. These operational limits are, for example, the maximum electrical power that can be generated by the generator or the maximum rotor speed the drive train is designed for. As a secondary objective, wind turbine controllers are increasingly expected to also reduce mechanical loading of the turbine.

The main actuators to achieve these goals are the pitch angles at each of the three blades and the generator torque.

The blades can be pitched using either hydraulic or electric motors. While most older turbines could only pitch all three blades collectively, newer turbines are capable of moving them independently. By changing the pitch angle at a blade, the angle of attack of the air flow and thereby also the generated lift and drag forces change. Pitch controlled turbines generally pitch to feather. That means that in order to decrease the generated lift, the pitch angle is increased resulting in a reduction of the angle of attack. The feather position is the angle at which the angle of attack is reduced to zero and very low lift is produced.

The electrical system is designed so that the generator may run at rotational speeds that are independent of the frequency of the grid the turbine is connected to. This is achieved by either converting the entire power output from the generator to direct current (full conversion) and then to the grid frequency or by using a generator setup that allows a variable slip (doubly fed induction generator [87]). With these types of electrical architecture, the generator torque is independent of the rotational speed of the generator as well. The converter control unit (CCU) controls the current and voltage at the generator so that the air gap torque is driven to a value commanded by the main turbine controller. While the converter control strategies for wind turbine applications that are used in the CCU are a widely researched topic as well [35, 12], for the turbine control problem the generator torque is the main actuated variable and the underlying converter control is part of the respective actuator model.

Since the same actuators are used to achieve both the primary and the secondary objective, the two objectives cannot be treated independently and there are significant trade-offs to be made. Nevertheless, there is generally a part of the overall controller that is mainly concerned with the primary objective of controlling speed and power output of the turbine, while performance with respect to the secondary objective is increased by adding additional control loops on top of the main speed and power controller.

2.1.2. Speed and Power Control

The basic speed and power controller is concerned with maximizing the power output under certain constraints on both the stationary and transient rotor speed and generator torque values. These constraints lead to two distinct operating regions that are best described in the torque-speed curve (figure 2.1). For a given turbine configuration, the maximum stationary power output is called rated power P_0, and the corresponding generator speed and torque values are called rated torque $M_{G,0}$ and speed ω_0.

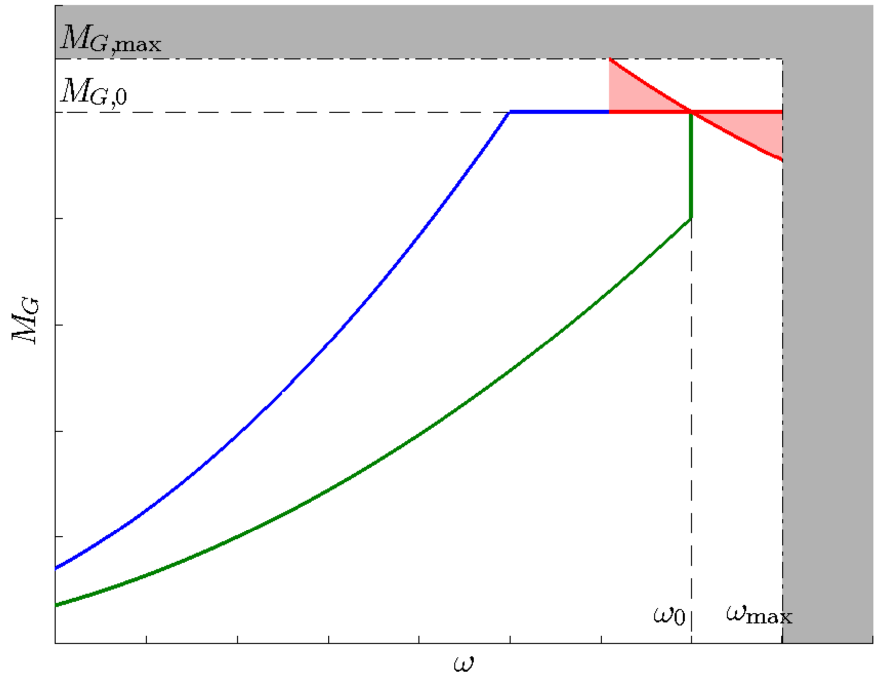


Figure 2.1.: Torque Speed Curves for variable Speed Wind Turbines. Green: Partial load operating region for speed constrained turbine Blue: Partial load operating region for torque constrained turbine. Red: Range of potential full load operating region concepts

In the partial load operating region, the turbine is operating below the rated torque and speed values. Therefore, the speed and torque can be freely varied in order to extract the maximum energy out of the wind. At a certain wind speed, the energy in the wind will be so large that the maximum extraction strategy would cause the turbine to run either above the rated speed or torque value. This region, where the turbine has still not reached rated power but cannot run at the aerodynamic optimum due to either the limit on speed or power, is called the transition region. Whether a turbine is torque or speed constrained depends on a number of factors such as the ratio of rated speed to rated torque and the ratio of rotor area to rated power. The majority of wind turbines is speed constrained and some overview articles do not even consider the torque constrained case at all ([3]). At a certain wind speed the turbine will reach both rated torque and rated speed. This wind speed is called the rated wind speed. The task of the controller for the so-called full load operating region is now to keep the power output at its rated value P_0 at all wind speeds above this rated wind speed.

Rated speed and torque are stationary constraints, i.e., that turbine may not operate above these values for an extended period of time. Due to transient variations in the incoming wind field, the turbine will however operate above either of these limits for a short period of time. There are also constraints on the instantaneous torque $M_{G,\max}$ and speed $\omega_{\max}$. While the maximum generator torque is a control constraint which can easily be honored, the maximum speed is an output constraint that, if violated, would lead to an immediate shut-down of the turbine. While these constraints on the instantaneous values are active in all control regions, they are mainly relevant for control in the full load operating region as here the torque and speed are closest to their respective short-term limits.

Variable Speed Operation The aerodynamic power generated by the rotor is given by:

$$P_A = \frac{1}{2}\rho\pi R^2 v^3 C_P(\lambda, \theta) \tag{2.1}$$

where ρ is the air density, R the rotor radius, v the free-stream wind speed, and C_P the dimensionless power coefficient. The power coefficient itself is a function of the pitch angle θ and the tip speed ratio:

$$\lambda = \frac{\omega R}{v}. \tag{2.2}$$

An example of the power coefficient as a function of pitch and tip speed ratio is shown in figure 2.2. In variable speed operation the goal of the controller is to operate such that C_P is at its maximum $C_{P,\max}(\lambda_{\text{opt}}, \theta_{\text{opt}})$. As the optimum pitch angle θ_{opt} does not depend on the current wind speed, the pitch angle is simply held constant at this optimum value, also called the fine pitch angle, throughout the entire variable speed operating region. The tip speed ratio λ is however a function of the rotor and wind speed. In order to operate at the optimum tip speed ratio, the turbine needs to operate at a rotor speed which depends on the current wind speed:

$$\omega_{\text{opt}} = \frac{\lambda_{\text{opt}} v}{R}. \tag{2.3}$$

With the pitch angles held constant, the generator torque is the only remaining actuator that is used to track this optimum rotor speed. In summary, in variable speed operation the control problem is a SISO tracking problem where the generator torque is used to control the rotor speed so that it tracks a varying reference that depends on the wind speed. This tracking problem is complicated by the fact that on most turbines the current free-stream wind speed v cannot be measured with enough accuracy to use it as an input to the controller. There are two main approaches that are used for designing a controller for this problem.

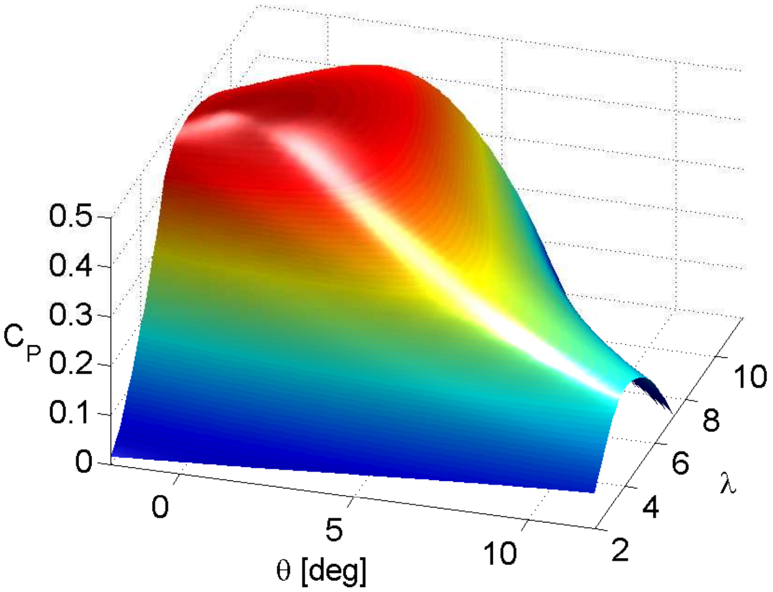


Figure 2.2.: Power coefficient C_P as a function of pitch angle θ and tip speed ratio λ

The first one uses the specific nature of the problem in order to design a tracking controller that does not require knowledge of the current optimum rotor speed: In stationary conditions and neglecting the mechanical and electrical losses the power generated at the generator

$$P_E = M_G \omega n_G, \tag{2.4}$$

where n_G is the gearbox ratio, is equal to the aerodynamic power produced at the rotor

$$M_G \omega n_G = \frac{1}{2} \rho \pi R^2 v^3 C_P(\lambda, \theta) \tag{2.5}$$

which can be solved for the generator torque

$$M_G = \frac{1}{2} \rho \pi \frac{R^5}{n_G} \underbrace{\frac{v^3}{\omega^3 R^3}}_{\lambda^{-3}} C_P(\lambda, \theta) \omega^2. \tag{2.6}$$

Now, assuming the turbine operates at its aerodynamic optimum

$$C_P(\lambda, \theta) = C_{P,\max}, \qquad \lambda = \lambda_{\text{opt}} \tag{2.7}$$

equation (2.6) becomes:

$$M_G = \underbrace{\frac{1}{2}\rho\pi\frac{R^5}{n_G\lambda_{opt}^3}C_{P,max}}_{k}\,\omega^2 \tag{2.8}$$

where all constants are lumped into k and the only remaining variable is the rotor speed ω which is measurable. This nonlinear control law $M_G = k\omega^2$ has been derived here using stationary considerations only. It can, however, be shown that it will also drive the rotor speed towards its optimum if the rotor speed is not at its optimum and therefore will lead to tracking of the rotor speed reference [42]. One disadvantage of this control strategy is that, due to the lack of tuning parameters, there is no control over how tightly the reference signal ω_{opt} is tracked.

The second approach relies on the wind speed estimator that is used on many modern wind turbines. The wind speed estimator uses a turbine model and the available measurements to compute an estimated wind speed $\hat{v}$ that approximates the actual wind speed:

$$\hat{v} \approx v. \tag{2.9}$$

See sections 3.1.2 and 5.1 for more details on wind speed estimators. With $\hat{v}$ available, the reference rotor speed can be explicitly calculated:

$$\omega_{\mathrm{set}} = \frac{\lambda_{\mathrm{opt}}\hat{v}}{R}. \tag{2.10}$$

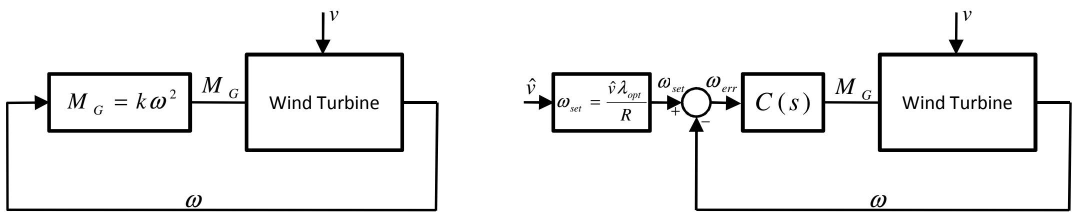
Now the speed can be controlled by designing a controller $-C(s)$[1] that drives the control error to zero:

$$M_G = -C(s)\omega_{err} \qquad \text{with} \quad \omega_{err} = \omega_{\mathrm{set}} - \omega. \tag{2.11}$$

The regulator $C(s)$ is commonly designed as a PI controller, but other designs are also possible. Unlike in the $M_G = k\omega^2$ approach, it is possible to influence the tracking performance through the design of the regulator. Figure 2.3 shows block diagrams of these two commonly used control approaches for variable speed operation.

Transition Region In the transition region, also called upper partial load operation, the control strategy depends mainly on whether the turbine is speed or torque constrained. In both cases the control objective is to maintain operation as close as possible to the optimal

[1] As an increasing generator torque, will generally cause the rotor speed to decrease the controller needs to have a negative gain. Here, and in the pitch controller covered later, the the factor -1 is pulled out so that $C(s)$ has a positive gain.



(a) $M_G = k\omega^2$ control law

(b) wind estimator based control scheme

Figure 2.3.: Block diagrams for the two commonly used control laws for speed control in the variable speed operating region

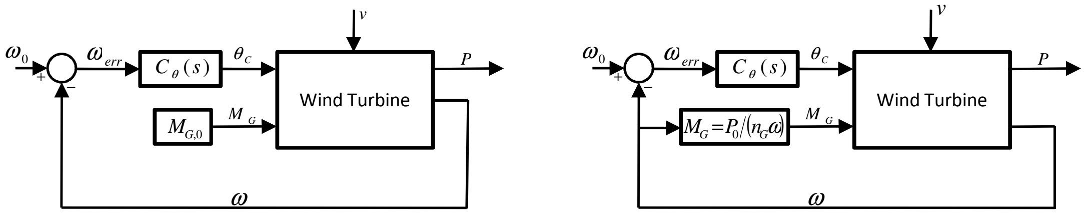
curve that is followed in the variable speed operation but to stay within the speed or torque limits. Therefore, in the speed constrained case, when rated speed is reached, the speed is held constant by controlling the torque accordingly. This can be achieved using a PI-type or similar controller that regulates the difference between the current rotor speed and the rated speed to zero. Due to variations in the wind speed, the torque required to maintain rated speed and thereby also the power output vary in this operating regime. If the power increases above the rated power value the turbine will transition to full load operation. Similarly, if the torque drops below the $k\omega^2$ curve the turbine will transition to the variable speed operation that was previously described. The described approach for transition region control is similar to the estimator approach for variable speed tracking and can also be realized easily by simply using the rated speed as an upper bound for the calculated speed set-point.

In the torque constrained case, the torque is simply held constant at its rated value and the speed is allowed to vary freely. The turbine is essentially uncontrolled in this region. Similar to the speed constrained case, switching to the other operating regions is performed if the $k\omega^2$ curve or respectively the constant power curve in the torque speed plane are hit.

Full Load Operation In full load operation there is more power available in the wind than can be used by the turbine. The turbine has to shed power by pitching the blades. The main goal of the controller is to regulate both power and turbine speed to their respective set-points with the wind acting as a disturbance. As both collective pitch and the generator torque are used to achieve these goals, the control problem becomes a 2x2 MIMO disturbance rejection problem. This 2x2 control problem is often simplified by either maintaining the generator torque constant at the rated value (constant torque control) or by controlling it so that the power output is constant (constant power control):

$$M_G = \begin{cases} M_{G,0} & : \text{constant torque control} \\ \frac{P_0}{n_G \omega} & : \text{constant power control} \end{cases} \tag{2.12}$$

In either case, the power regulation performance is then directly linked to the speed control performance. If the turbine is controlled to run at rated speed, for both these torque control laws the power output will be rated power and regulation of the rotor speed to rated speed translates into regulation of power to rated power. The remaining control problem, using the pitch to control speed, is again a SISO problem. Obviously, the choice of constant power control will give the best performance with respect to the power regulation control objective. Yet, it will also make speed regulation more difficult as for rotor speeds above the set-point the generator torque is decreased which further speeds up the rotor. It has been shown that a hybrid strategy that lies somewhere in between these two extremes might give better overall performance [85].



(a) constant torque

(b) constant power

Figure 2.4.: Two baseline control schemes for the full load speed and power control problem

The aerodynamic torque generated by the rotor is

$$M_A = \frac{1}{2}\rho\pi R^3 v^2 C_M(\lambda, \theta) \tag{2.13}$$

where the C_M is the torque coefficient. Equation (2.13) shows that the controller needs to counter any increase in wind speed v by reducing the torque coefficient through a change in pitch angle θ. In stationary conditions the aerodynamic torque needs to be equal to the rated torque at the generator for all wind speeds above rated wind speed. The pitch controller $C_\theta(s)$ is also typically designed as a PI-based controller that acts on the speed error:

$$\theta = -C_\theta(s)(\omega - \omega_0). \tag{2.14}$$

The partial derivative of the aerodynamic torque with respect to the pitch

$$\frac{\partial M_A}{\partial \theta} = \frac{1}{2}\rho\pi R^3 v^2 \frac{\partial C_M}{\partial \theta} \tag{2.15}$$

is also called the pitch sensitivity. Especially due to the quadratic influence of the wind speed, there is a significant change in this sensitivity over the full load operating region. For typical turbines, the pitch sensitivity near the cut-out wind speed[2] is about five times as large as near rated wind speed as shown in figure 2.5. This nonlinearity needs to be accounted for when designing the pitch controller for the full load operating region. This is typically done by using a gain scheduling approach: The gains are adjusted as a function of the wind speed, for example using the inverse of the pitch sensitivity [66]. If no wind speed estimator is used and the wind speed is not available this scheduling is usually done as a function of the pitch angle.

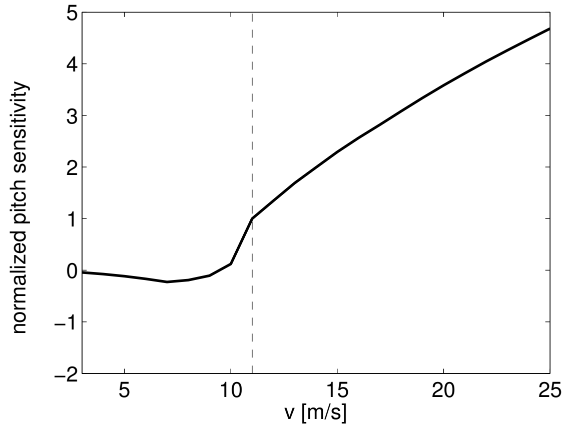


Figure 2.5.: Pitch Sensitivity normalized to sensitivity at rated wind speed for an example turbine with $v_r = 11\,\mathrm{m/s}$

2.1.3. Load Reducing Control

There are several additions that are commonly added on top of the basic speed and power controller which are designed to reduce mechanical loads acting on the turbine.

Drive Train Damper Unlike the flapwise or out-of-plane motion of the blades, the edgewise oscillations modes have very little aerodynamic damping and thus also very little

[2] The cut-out wind speed is defined as the wind speed above which the turbine is not allowed to operate continuously. If the wind speed exceeds this limit for a certain period of time (e.g., 10 minutes), the turbine is powered down by pitching the blades to a completely feathered position.

overall damping. These edgewise motions are also coupled with the drive train dynamics which are arising from the flexible coupling of the generator inertia with the low speed shaft. One of the most problematic modes is the so-called collective edge mode where all three blades oscillate in phase against the generator inertia [26]. To add extra damping to this mode, a drive train damper that uses torque actuation based on the measured generator speed is commonly used. The torque output from the drive train damper is then added to the generator torque command from the main speed and power controller. This drive train damper is often designed to be a bandpass or high pass filter in order to not interfere with the main speed regulation which occurs at lower frequencies than these types of drive train oscillations [3].

Tower Feedback A tower damper or tower feedback controller uses the measured tower top acceleration along the direction of the rotor axis (longitudinal) to generate a pitch offset that is added to the pitch command from the speed controller [4]. Using a simple one degree of freedom model for tower motion

$$\ddot{x}_t + 2\zeta_T\omega_{T,0}\dot{x}_t + \omega_{T,0}^2 x_t = K_{fa}F_A(\omega, v, \theta) \tag{2.16}$$

where x_t is the longitudinal tower top position and F_A is the aerodynamic thrust force acting on the system, and linearizing the thrust at a fixed operating point

$$\ddot{x}_t + 2\zeta_T\omega_{T,0}\dot{x}_t + \omega_{T,0}^2 x_t = K_{fa}\frac{\partial F_A}{\partial v}\delta v + K_{fa}\frac{\partial F_A}{\partial \theta}\delta\theta \tag{2.17}$$

it can be seen that a simple controller consisting only of integral feedback

$$\delta\theta = -K_I\int_0^t \ddot{x}_t \mathrm{d}t \tag{2.18}$$

would suffice to add damping to the tower motion:

$$\ddot{x}_t + \left(2\zeta_T\omega_{T,0} + K_I K_{fa}\frac{\partial F_A}{\partial \theta}\right)\dot{x}_t + \omega_{T,0}^2 x_t = K_{fa}\frac{\partial F_A}{\partial v}\delta v. \tag{2.19}$$

Instead of a pure integral controller the tower damper is often implemented as a PI controller to provide an additional degree of freedom to account for phase loss, for example from the pitch actuator dynamics or necessary filtering on the tower acceleration signal.

The tower damping control has been treated here as independent of the pitch control for speed regulation. However, since they use a common actuator, there is a significant amount of cross-coupling. Any pitch actuation to regulate speed will also excite the tower motion;

while any attempt to dampen the tower motion will also impact the rotor speed. Therefore, proper attention has to be paid during the design of these controllers to limit these interactions [63, 64].

Individual Pitch Control On many modern wind turbines the three blades can be pitched individually. This capability can be used to reduce loads cased by the asymmetric loading of the rotor. For example, due to the wind shear, the wind speed a blade pointing up is subjected to is usually greater than the wind speed for a blade pointing down. This creates a nodding moment that would bend the main shaft upwards as each of the blades generates more thrust while pointing upward than when pointing downwards. This effect can be countered by increasing the pitch angle for each blade a little bit while pointing up and decreasing it while pointing down. Although many different designs for this individual pitch control (IPC) exist, many follow the basic ideas as outlined by Bossanyi [4, 5]:

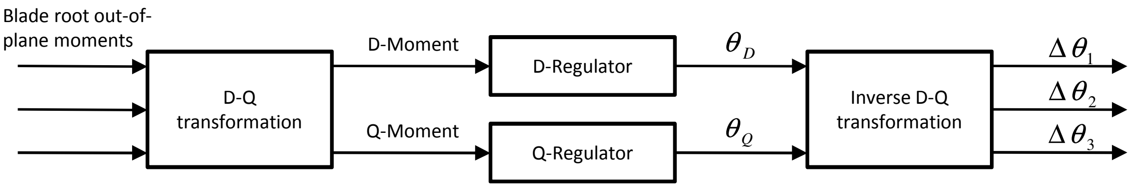


Figure 2.6.: Basic individual pitch control scheme as shown in [5]

The controller is acting and designed in the fixed (non-rotating) coordinate frame. The nodding (D) and yawing (Q) moment are either measured at the main shaft or derived from load measurements at the blade roots. For both the D and the Q axis a controller is designed to regulate the respective moment to zero by generating a fixed frame pitch offset for each axis. The actual pitch offsets to be added to the collective pitch command are then calculated by transforming the fixed frame pitch angles back into the rotating coordinate frame. The design of the D- and Q-regulators is often based on a simple PI controller, but needs to include extensive filtering in order not to react on loads that cannot be countered by once-per-revolution (1P) pitching, such as the load caused by the tower dam effect.

The IPC as described here constrains the pitch offsets to a mean of zero. Thus, there is very limited interaction with the control loops that have the collective pitch angle as an output and the individual control problem can be treated as decoupled [27]. There might however be some benefit in relaxing this constraint and designing the individual and collective pitch controllers in a unified framework [22].

Figure 2.7 shows the controller structure resulting from the combination of the speed and power controller with the load reducing control loops described here.

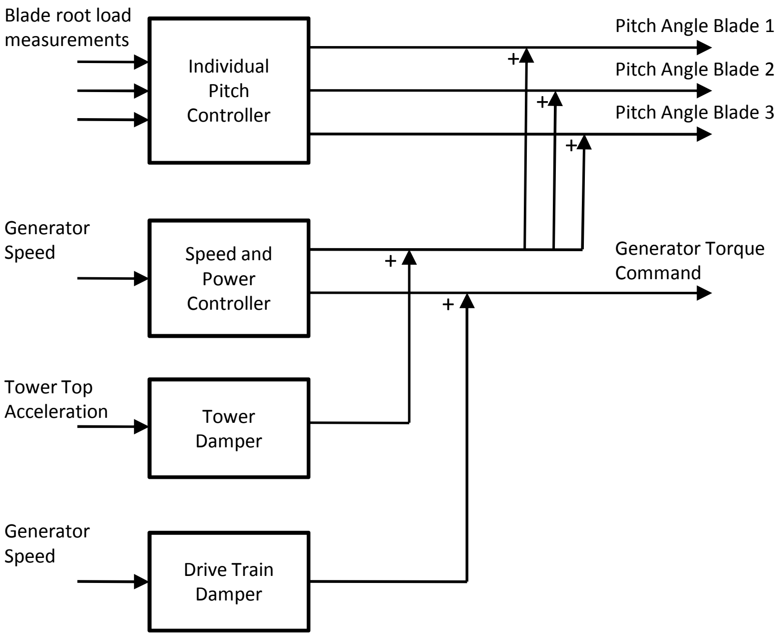


Figure 2.7.: General control structure with the most commonly used load reduction features

2.1.4. Shutdown Control

The control strategy including the load reduction features that are described in the previous sections are only active when the turbine is operating to produce power. Traditionally, most of the research and development has been focused on this controller for normal power production (NPP) operation. However, there are also situations where the turbine needs to be controlled dynamically while it is not producing power, e.g., the start-up and the shutdown process. In the context of load reducing control, the start-up is of relatively low importance since load levels during start-up are low. The loads the turbine encounters during a shut-down, however, can contribute significantly to the overall turbine design and lifetime. A shutdown procedure may be initiated either by an operator request or by a fault that is detected by the turbine controller. Such faults may for example be related to the wind conditions such as the wind speed being above the design wind speed of the turbine, the wind direction exceeding a predefined limit, or to the turbine operation such as the generator speed approaching a safety critical level. Once a fault is triggered, the turbine will be brought to a stand-still using a shutdown controller. Depending on the kind of scenario or error message, different shut-down controllers are used. For example, if a regular stop of the turbine is commanded, the stopping procedure will generally be more gentle than in case of an emergency stop where the objective is to bring the turbine to a halt as quickly as possible. Although more elaborate strategies for such an emergency stop are possible, the most common control scheme, which is also used as a baseline here, simply pitches the blades towards the feathered position at a fixed rate without any feedback of rotor speed or other measured signals. The pitch rate during the shut-down process depends on the capabilities of the pitch system but is in normally somewhere between 5 deg/s and 15 deg/s.

2.1.5. FRT Control

In the past, whenever there was a disturbance in the electrical grid the wind turbine was connected to, the turbine would simply be shut down. With the increasing penetration of wind energy in many grids, however, this is no longer practical as turbines disconnecting due to grid disturbance would further weaken the grid and could cause even more systems to trip. More specifically, today, in many grids wind turbines are required to stay connected to the grid for a certain amount of time if there is a drop in voltage. This is generally referred to as Fault Ride Through (FRT) capability. The exact type of the requirement is defined by the grid operator and varies between the various grids [40]. Nevertheless, most requirements are similar to the generic curve shown in figure 2.8 for a so-called Low Voltage Ride Through (LVRT) requirement. The grid voltage drops suddenly, potentially stays flat at a low level for a short amount of time, and then increases slowly. Of course, grid disturbances usually do not have this exact shape; a curve like this defines the most severe disturbance the turbine needs to be able to ride through: As long as the grid voltage is above the specified curve the turbine must stay connected to the grid. LVRT is mainly a challenge for the converter control unit and grid interconnection. A significant amount of research has been performed on the design of the control algorithms for the electrical system (e.g., [14, 78, 121]) for these scenarios. However, LVRT and other grid events also challenge the main turbine controller. In many of these grid events, the turbine generators are not able to produce any torque for up to a few seconds because no power can be fed to the grid [76, 20, 21, 74]. Due to the loss of counter torque at the generator, the rotor will speed up threatening to cause an overspeed fault which would in turn lead to an emergency shut-down of the turbine. The loss of counter torque will need to be countered by increasing the pitch angle at the blades. Although this pitch control problem is very similar in structure to pitch control in normal operation, the plant dynamics might be very different since the turbine is operating far off its normal operating curve and therefore requires a specifically tailored pitch controller.

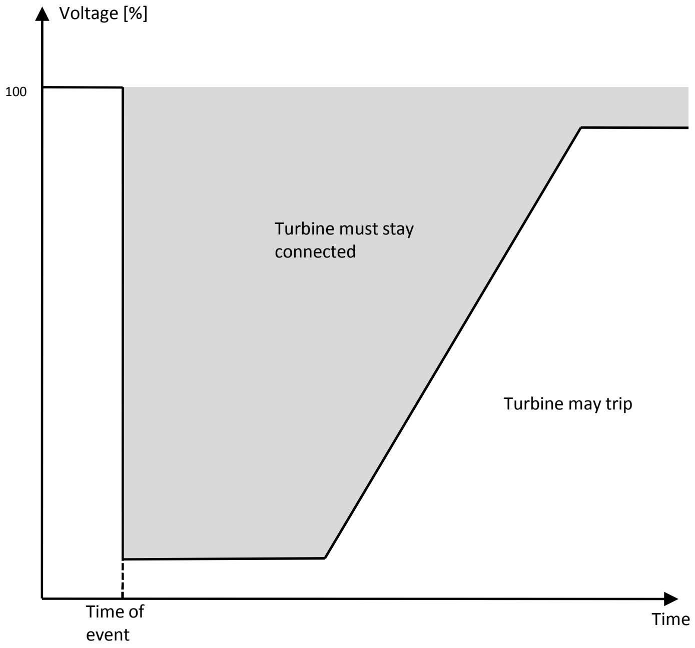


Figure 2.8.: Generic LVRT requirement curve [40]

2.2. System Simulation and Loads Analysis of Wind Turbines

The design of modern wind turbines relies heavily on simulation methods. All mechanical components need to be dimensioned so that they can withstand all loads the turbine is likely to encounter and to last at least a specified lifetime, usually 20 years. The main driver for the mechanical loading of the turbine are the aerodynamic forces acting on the turbine generated by blades. These aerodynamic forces are a function of the local flow conditions on the turbine blades and are thus highly dependent on the incoming wind field. The wind is stochastic in nature and varies both on short (e.g., seconds) and longer time scales (hours to months). The short term fluctuations are mainly caused by local phenomena and the landscape surrounding the turbine. These short time fluctuations in wind speed are superimposed on the slower phenomena caused by more global effects such as diurnal or seasonal variations [10, 26]. The design of a wind turbine needs to consider a set of wind conditions that is representative for both typical and extreme conditions, such as storms and gusts, which the turbine may experience at its site of erection. Next to variations in the wind field, turbine loading is also influenced by the way it is operated. For example, if the turbine is in an idling state, the loading will be very different from when it is producing full power. Finally, a failure in one of the components should not endanger the whole turbine and the various component failure modes also need to be considered. Overall, the combination of wind conditions, operating modes, and potential turbine or component failures, leads to a very large number of scenarios that need to be taken into account. The design assessment of a typical modern wind turbine can easily require more than 1000 single scenarios to be analyzed. Such a large number of scenarios can only effectively be evaluated through the use of automated simulation tools. Hence, the design of wind turbines relies heavily on simulation tools that model the overall aero-elastic system behavior and dynamics. Common tools that are used include for example BLADED [6], FAST [44], or FLEX [84].

For several reasons, the design of wind turbine controllers also relies heavily on the same simulation tools as the loads simulations: With the turbine operating in such a large variety of conditions the controller needs to able to handle all these conditions. Firstly, with no wind tunnels available to test full size, modern wind turbines, the wind as the main driving factor for the turbine dynamics is largely out of control and one has to rely on favorable conditions occurring at the test site which can easily be prohibitive. Secondly, not only can the wind field the turbine is subjected to not be controlled, it is also very difficult to measure the full wind field. With the uncertainty on wind speed, a valid conclusion on the control performance can only be made by measuring the turbine performance over an extended period of time and using statistics to reduce the variability. Finally, if the design of a turbine is to be certified by an external party, the simulations have to follow certain guidelines or standards such as the IEC regulations on the design of wind turbines [39]. Even without an external certification, the conditions and load cases defined in these regulations have become the widely accepted standard for wind turbine design. A significant goal of the design of controllers for wind turbines is the reduction of mechanical loads acting on the turbine. The mechanical loads are assessed through the use of standardized aero-elastic simulation tools and conditions. Therefore, any wind turbine control design focused on load reduction needs to be assessed in terms of its impact on the same loads simulations that define the turbine design and ultimately the turbine cost. Subsequently, the majority of all research on control of wind turbines that has been performed uses simulation studies only. Even in the few existing studies where advanced controllers have been tested on real turbines (e.g., [118, 119, 42, 49]), the actual design was performed in simulation and the field test is only used to confirm that the field performance matches the predictions from simulations for a very limited range of conditions.

The following sections will provide an overview of the key features of the loads simulations and outline their impact on the control design.

2.2.1. Models

In order to model the dynamic behavior of a wind turbine, models for several components need to be combined: An aerodynamic model is used to calculate the forces and moments acting on the rotor blades that are caused by the local flow conditions. A mechanical model then models the response of the main structural components such as main shaft, blades, and tower to the aerodynamic loading. The turbine controller will react on measured turbine states by sending commands to the actuators such as the generator system or the pitch motors, and the behavior of these actuators needs to be modeled as well.

For all of these subsystems, there exist general purpose, high fidelity modeling methods such as CFD or FEM-based tools. However, due to the described large number of simulations that need to be evaluated in the turbine design process, the computational speed of the simulation tools is, even given modern computing power, still of high importance and model fidelity for these system level simulation is kept at a minimum. Further, because of the unique combination of aerodynamic, structural, and electro-mechanical modeling, wind turbine system simulations are generally performed using tools that have been specifically created for wind turbine simulations. Although there is some variance in the exact modeling technique that is used, most of the commonly used codes use similar modeling approaches in principal [86]. Therefore, here, only the main modeling approaches that are implemented in the commonly used tools are introduced.

It should further be noted that the loads evaluation process for wind turbines usually uses two levels of analysis. In the first step, the so-called system simulation, the overall behavior of the entire wind turbine system is simulated. These simulations use the described low fidelity, highly specialized system simulation tools. The results from these simulations are the mechanical loads each component will be subjected to. Additionally, for each subsystem, there are often additional simulation methods and tools to perform a more detailed analysis. As an example, the main frame of a turbine is generally not modeled as a flexible body in the system simulation as the dynamic coupling between the structural modes of the main frame and the overall turbine modes is low. Instead, the main frame is assumed to be a rigid body. The result from the system simulation would then be the loads acting on the mainframe. As a next step, these loads could then be used as inputs, to a more detailed simulation, e.g., using 3D-FEM, of the mainframe dynamics only. For the control design, these higher frequency dynamics generally do not have to be considered since the controller mainly interacts with the low-frequency, turbine level modes. Therefore, only the models typically used in the system simulations are introduced here.

Aerodynamics The aerodynamics are generally modeled using Blade Element Momentum (BEM) Theory [10]. BEM is a quasi-steady approach where it is assumed that the aerodynamic forces acting on each section of the blades can be calculated purely based on the geometry of that section. All 3D effects, i.e., air flow along the span of the blade, are ignored. Under this assumption the lift $F_L$and drag F_D forces acting on each blade section are then calculated based on the local flow velocity c[3] and the lift and drag coefficients c_l and c_d

$$F_L = c_l(\alpha)\frac{\rho}{2}c^2 b_s t_s \qquad F_D = c_d(\alpha)\frac{\rho}{2}c^2 b_s t_s. \tag{2.20}$$

where ρ is the air density and b_s and t_s are the blade section and chord length respectively. The lift and drag coefficients are a function of the local angle of attack α which itself is a function of the free stream wind speed, the rotational speed of the rotor, and the local induction factor[4]. The functional relationships of the lift and drag coefficients, also called profile polars, are generally determined either using wind tunnel tests or in CFD simulations. It should be noted here that because of the quadratic influence of the wind speed and the lift and drag coefficients, the aerodynamic model is highly nonlinear.

This basic BEM model is then augmented with several additional semi-empirical models to improve accuracy: A tip-loss model modifies equation (2.20) so that the blade sections near the tip produce less lift because of the flow along the span of the blade and around the tip. A dynamic stall model [62] modifies the profile polars in case the blade section is rotating about its center axis, for example due to pitch motion or torsional vibrations of the blade. The described BEM model is quasi-stationary, i.e., the assumed local flow conditions and resulting aerodynamic forces only depend on the current conditions and would change immediately with any change in the wind field or blade movement. In reality, the fluid flow needs some time to adjust to the new situation and the change is not instantaneous. Therefore, a dynamic wake model is used that models the fact that the flow conditions do not change instantaneously by adding some dynamics, usually in the form of low-pass behavior, to the calculation of the local induction factor.

The output of the aerodynamic calculations are the distributed forces along each of the blades. These forces then act as the external forcing on the structural dynamics of the turbine.

[3] The local velocity depends largely on the circumferential speed of the particular blade section *and* the free stream wind speed and should not be confused with the latter

[4] The induction factor is the factor by which the free stream wind speed has been slowed down when it reaches the rotor plane due to the rotor extracting kinetic energy.

Structural Dynamics In most tools, the structural model is a combination of rigid and flexible bodies. Usually, only the blades and the tower are modeled as being flexible while all other components are rigid. In order to reduce the computational complexity, most codes provide some means for a modal reduction of the flexible bodies. The blade and tower motion is assumed to be limited to a superposition of the first few eigenmodes. In most cases, only the first two modes are considered, which can be justified by the higher modes being difficult to detect in actual measurements and having a negligible energy content. It should be noted that the modal decomposition and reduction is performed for each axis individually: E.g., each of the blades may have two modal degrees of freedom for each of flapwise and edgewise deflection as well as torsion. Combining these modal degrees of freedom with the rigid body degrees of freedom results in an overall, coupled mechanical system with between 20 and 100 degrees of freedom.

The mechanical system of a modern turbine has a very low inherent damping. Especially for structural motion along the direction of the wind speed, most of the damping is provided by the aerodynamics: Any movement in the wind direction reduces the relative wind speed at the blades and thereby reduces the thrust produced by the rotor while movement towards the wind has the opposite effect. This aerodynamic damping of the fore-aft motion is only one example of an effect arising from the coupling of structural and aerodynamics. These types of aero-elastic phenomena are important for many aspects of wind turbine dynamics and therefore need to be considered specifically not only in the system simulation but also in the control design.

Actuators The three principal actuators yaw, pitch, and generator each consist of several subsystems and underlying control loops and may have complex dynamics. While a detailed modeling of these systems in the overall system level simulation tools can be beneficial for the design of the actuators, it is not necessarily required for the overall system and controller design.

The output of the controller is a commanded generator torque. The converter control unit (CCU) then uses the capabilities of the converter to produce a current in the generator that produces a torque that matches the desired value. The dynamics of this underlying electrical control loop are at least one order of magnitude faster than the mechanical dynamics of the system. Therefore, they are not included explicitly in the simulation of the overall system, but instead are typically represented by a small communication delay and time constant between the torque demanded by the generator and the actual air-gap torque acting on the mechanical system.

The pitch system receives a desired pitch angle for each blade from the controller. It then needs to control the pitch motors to drive the blades in the desired positions. Due to the large inertia of the blades, there is a significant delay between a blade pitch command and the blade actually arriving in the desired position. This delay cannot be ignored in the system simulations. These dynamics are largely dominated by the blade and motor inertia, the capabilities of the pitch system, and the tuning of the underlying control loops while the torque caused by the aerodynamic loading of the blades only has a small impact. Many simulation tools simply assume that all effects that depend on the current state of the turbine can be ignored and lump the entire pitch system dynamics into one SISO model that has the commanded pitch angle as its only input and the actual pitch angle as output. Often the pitch system is modeled as a linear second oder system with low pass characteristics plus additional rate limits on the pitch angle that represent the power limits of the pitch motors.

The yaw system of a typical turbine is significantly slower than the pitch system and the structural dynamics. Since the yaw rates are so slow that there is very little interaction with the rest of the system behavior, it is often not considered at all. Within one typical ten minute time series, the yaw is simply assumed to be fixed. If yawing is to be considered it can be modeled similar to the pitch system but with significantly lower bandwidth and rate limits.

2.2.2. Load Cases

A full loads analysis requires consideration of a number of predefined scenarios, for example representing an operational state or failure mode. Each scenario needs to be evaluated over a large range of external conditions such as mean wind speed, turbulence, and starting values. Such a group of simulations is called a *load case*. The different load cases that are to be considered are for example defined in the IEC guidelines [39]. Not all of the load cases are relevant for the control design, either because in this load case the controller does not affect the loads that occur, or because they are generally not driving the turbine design. For example, during storms most turbines are in an idling mode where the pitch angles are simply kept at a fixed value and the turbine is not actively controlled. Table 2.1 lists the subset of load cases where the controller design has the most impact on the turbine loads.

Normal Power Production NPP is the state in which the turbine will operate for the vast majority of the time. The NPP simulations emulate the case where the turbine is connected to the grid and produces power and need to capture the typical conditions the turbine will be subjected to. These simulations are performed with a turbulent wind field. For a given single simulation, the timeseries of the wind field is a superposition of a constant mean wind speed and a stochastic turbulence with a mean of zero. To capture the low frequency

Load Case	Description	Main Varying Factors	Relevant Controller	Type of Loads Analysis
1.1	Power Production Operation in turbulent conditions without fault	Mean Wind Speed, Turbulence	NPP	U/F
1.3	Extreme Coherent Gust and Direction Change	Wind Speed, Gust Amplitude, Direction Change	NPP, Shutdown	U
1.5	Extreme Operating Gust (1-year) and grid loss	Wind Speed, Gust Amplitude, Timing of Grid Loss	NPP, Shutdown	U
1.6	Extreme Operating Gust (50-year)	Wind Speed, Gust Amplitude	NPP, (Shutdown)	U
1.9	Extreme Coherent Gust	Wind Speed, Gust Amplitude	NPP, (Shutdown)	U
5.1	Emergency Stop	Wind Speed	Shutdown	U/F

Table 2.1.: Main loadcases according to [39] that are relevant for the turbine control design. U:Ultimate(Extreme) Analysis, F: Fatigue Analysis

changes in wind speed, simulations are performed at all wind speeds between the cut-in and the cut-out wind speed, usually in steps of 1 m/s. Generally, the turbulence intensity is a function of the mean wind speed. In the turbine design process generic turbulence levels such as defined by the IEC classes are typically used. See figure 2.9 for the turbulence as a function of mean wind speed for turbulence classes A and B. The turbulent component of the wind speed time series is created using a turbulence model such as the Kaimal or Mann [26] models. Generation of the turbulent wind requires some form of random number generator. In order to reduce the statistical variation, the turbine behavior at each wind speed is simulated with different starting values (”seeds”) for the random number generator used in the creation of the turbulence.

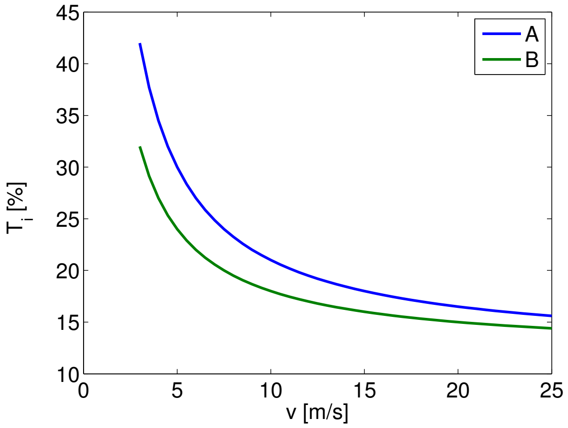


Figure 2.9.: Turbulence intensity as a function of mean wind speed for turbulence classes A and B according to [39]

The turbine controller should be designed so that it can handle all situations that might arise during normal power production. Therefore, in load case 1.1 the turbine is assumed to be operating without fault and only the NPP-controller is of relevance.

Gusts Next to the conditions the turbine would typically encounter, extreme wind conditions that the turbine may only experience in rare situations also need to be considered. For example load case 1.6 describes a gust that is so strong that on average it occurs only once in fifty years while load case 1.3 describes the case where a so-called Extreme Coherent Gust (ECG) gust coincides with an extreme change in wind direction. These simulations are performed using deterministic wind profiles, i.e., the wind is not stochastic but follows a predefined gust shape and only the amplitude and starting wind speed are varied. Figure 2.10 shows the wind speeds for an Extreme Operating Gusts (EOG) which are used in load case 1.5 and 1.6 and the Extreme Coherent Gust (ECG) used in load case 1.3 and 1.9.

The turbine will enter into these events in NPP mode. Therefore, the NPP controller will define the behavior during the gust. However, due to these gust representing extreme conditions, one or more of the safety thresholds might be triggered. For example, in load case 1.6 the gust can easily be so strong that the generator speed rises above the overspeed trigger level even with a functioning NPP controller and the turbine would shut down. In this case, the control strategy used for shutting down the turbine also has an impact on the loads occurring during that load case.

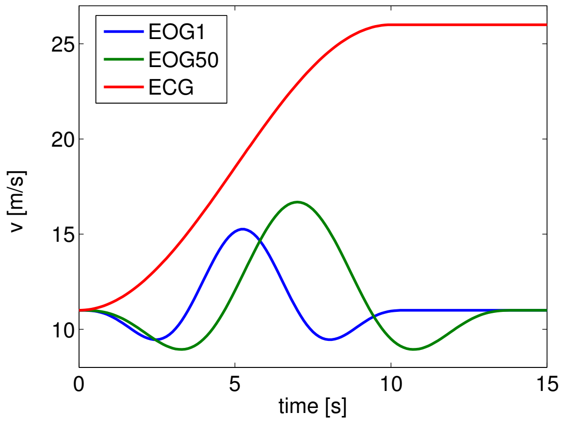


Figure 2.10.: Extreme Operating Gusts with a 1-year (EOG1) and 50-year (EOG50) return period and Extreme Coherent Gust (ECG) according to [39] at a wind speed of 11 m/s for a turbine with a rotor diameter of 126 m

Grid loss Extreme events are not only caused by the wind conditions. They can also be caused by failures in the turbine itself. Most importantly, faults in the electrical system need to be considered. These can be caused by failures of electrical components such as the frequency converters, but can also arise as a result of the connection to the power grid. For example if the grid frequency deviates by more than a specified amount from its base value, most turbines will be disconnected from the grid. These types of events are characterized by an immediate disconnection of the electrical system and resultant loss of counter torque at the generator.

Load case 5.1 simulates an emergency stop. If an emergency shut-down is triggered, the turbine will also disconnect the generator and the behavior is very similar to a grid loss. While these simulations are performed at multiple wind speeds, the wind speed during each of these simulations is constant. Without any gust or turbulence, the NPP controller that is active during the beginning of the simulation has no impact and this load case is only important for the design of the shut-down controller that is used once the fault is triggered.

Load case 1.5 combines a deterministic gust with a grid fault. The gust has the same "Mexican Hat" shape as the one used in load case 1.6, but a smaller amplitude as a return period of only one year needs to be considered. This gust coincides with a grid loss that happens at a point in time during the gust. The relevance to the controller design is similar to load case 1.6 as the NPP controller is used initially and then switches to a shutdown controller. However, unlike in load case 1.6, the turbine will always shut-down independently of how the NPP controller performed during the first part of the gust.

2.2.3. Extreme Analysis

The analysis of extreme or ultimate loads is concerned with dimensioning the turbine components so that they can withstand the maximum load the turbine is likely to incur during its lifetime. Here, the component is required to only withstand the load once and all effects that are a result of multiple applications of a load, such as crack growth, are ignored. Subsequently, the magnitude that is considered in the extreme load analysis is computed by taking the maximum over all considered load cases and time series. This maximum is computed independently for each point and load of interest[5] and the maxima will generally occur in different load cases. For example, the maximum tower base bending moment will usually occur in different conditions than the maximum main shaft torque.

Generally, all load cases including NPP need to be taken into account in the extreme analysis. On the other hand, the maximum load on most sensors usually does not occur during normal power production, but rather in one of the gust, fault, or storm load cases. Therefore, one important aim of the wind turbine controller design is to reduce the load peaks especially under these gust and fault conditions. The reduction of extreme loads that occur

[5] often called a sensor in the loads analysis context

during normal power production loading is less relevant as these loads are often not driving the turbine design: For example, if the maximum load for a given sensor that occurred during load case 1.1 is significantly below the load in load case 1.6, any reduction of this load in load case 1.1 through a modified controls strategy will not bring the extreme load for this sensor down. The sensor is then said to be dimensioned by load case 1.6.

2.2.4. Fatigue Analysis

Unlike the extreme analysis, the fatigue analysis is concerned with the effect of applying a load repeatedly. Most mechanical components on a wind turbine are exposed to a complex sequence of loads. These loads are mainly caused by variations in the wind and also harmonic excitation effects caused by the rotation of the rotor. For example the slight difference in the weight of the three blades that is always present will cause a revolving load on the main shaft with the amplitude and frequency depending on the rotor speed. The entire set of load cycles the turbine will encounter during its lifetime is called the loads collective. For wind turbines, it consists of load cycles covering a large range of magnitudes and number of occurrences.

In order to facilitate the analysis of fatigue loads, the entire loads collective for each sensor that the turbine will encounter during its lifetime is combined into a *damage equivalent load* (DEL). The DEL is the load that, if applied with a specific frequency, would cause the same damage to the component as the loads collective. So a complex load consisting of cycles with a variable amplitude and frequency is transformed into an equivalent load with a fixed amplitude and frequency.

For a given time series, first the loads collective, the number n and amplitudes σ_i of all cycles needs to be determined. This is done using a cycle counting method such as Rain-Flow-Counting [95]. The total damage D caused by this loads collective is computed using the linear damage accumulation hypotheses according to Palmgren-Miner (e.g., [113]): The total damage caused by this loads collective is

$$D = \sum_{i=1}^{n} \sigma_i^m, \tag{2.21}$$

where m is the material-dependent slope of the Wöhler curve. The DEL σ_{DEL} is now calculated such that, if applied for N_{ref} cycles it leads to the same overall damage:

$$N_{ref} \sigma_{del}^m = D. \tag{2.22}$$

The number of reference cycles for a time series with length t_{ts} is chosen so that it corresponds to a reference frequency f_{ref}:

$$N_{ref} = f_{ref} t_{ts}, \tag{2.23}$$

where the reference frequency is a frequency in the same order of magnitude as the frequencies most often seen in the loads collective. For wind turbines, this is usually the frequency of the rotor revolution during operation at rated speed.

The overall DEL for each sensor of a wind turbine is a combination of the loading at different operating conditions. Therefore, for each time series i that is considered, the DELs $\sigma(i)$ are calculated. The overall DEL is then obtained by combining the single DELs according to the occurrence probability $p(i)$ of this time series and using the linear damage accumulation hypotheses:

$$\sigma_{del,tot} = \sqrt[m]{\sum_i p(i)\sigma(i)^m}. \tag{2.24}$$

Although some event load cases such as emergency stops (load case 5.1) are included in the fatigue analysis, their contribution to the overall fatigue is low due to their low share of the overall lifetime. Thus, in order to impact the fatigue loads using controls, the focus needs to be on the normal turbine operation as simulated in load case 1.1 and therefore also on the NPP-controller.

2.2.5. The Aero-elastic Simulation Tool FAST

FAST (*F*atigue, *A*erodynamics, *S*tructures, and *T*urbulence) [44] is a simulation code for the aero-elastic simulation of wind turbines that was developed by the National Renewable Energy Laboratory (NREL) of the United States starting in the mid-nineties. Unlike similar tools such as BLADED or FLEX, FAST is available free of charge and its source code is published. This has lead to FAST probably being the most commonly used tool for aero-elastic wind turbine simulations, especially for academic research.

Using FAST the behavior of two- or three-bladed on- or offshore wind turbines in a up- or down-wind configuration can be studied.

Strictly speaking, the aerodynamic modelling is not a part of FAST itself. Instead, the aerodynamic code AERODYN [57] is used. However, since AERODYN is now fully integrated in FAST, this distinction is generally not made. AERODYN models the aerodynamics of horizontal-axis wind turbines using the Blade Element Momentum Theory (BEM) (e.g., [26, 10]). In addition to the pure BEM, corrections for tip and hub losses, skewed wakes, dynamic stall, and tower shadow are also implemented [75]. Next to deterministic wind speed time series defined at hub height only, AERODYN also has the ability to use fully three-dimensional turbulent wind fields.

The structural model in FAST consists of both flexible and rigid bodies. The tower, drivetrain, and blades are assumed to be flexible while all other components are modeled as being rigid. The flexible bodies are included in the multi-body formulation using a modal formulation. For a three-bladed turbine up two 24 degrees-of-freedom (DOF) are available. Of these 24 DOF, four belong to the tower bending (2 each longitudinal and lateral) and nine to blade bending (2 flap-wise modes and one edge-wise mode for each blade) while the remaining DOF belong to the support platform (6 DOF), generator and rotor speed (2 DOF), yaw flexibility (1 DOF), and rotor tail furl (2 DOF)[6]. Which of these DOF actually needs to be used depends heavily on the specific turbine configuration that is being analyzed, and FAST therefore provides functionality to selectively activate or deactivate individual DOF.

FAST also includes several simple models for the behavior of the generator and simple control algorithm for speed and power control. It is also possible to include custom generator models and controller codes using either a user-written dynamic link library (dll) or by running the program in SIMULINK using the provided interface and modeling the controller and generator there. Further, no model of the pitch actuators is provided and, if required, this also needs to be added as a dll or included in the SIMULINK model.

2.2.6. The NREL 5MW Reference Turbine

The National Renewable Energy Laboratory (NREL) has developed and published a model of a 5MW reference turbine [43]. Although this turbine has not and will not be built, it has been designed to represent typical wind turbines in the class of large offshore turbines and is also heavily based on existing 5MW turbines such as the REPower 5M [108]. This reference model has been developed specifically to act as a baseline for various wind turbine related studies and to give wind turbine researchers access to a full turbine model without restrictions. It has been used in hundreds of studies focused on wind turbine aerodynamics, system dynamics, and controls research.

The NREL 5MW reference turbine is a three-bladed upwind turbine with a variable-speed, active-pitch control system. As such, its configuration matches the majority of modern on- and offshore turbines. It has a rotor with a 126m diameter and a hub height of 90 m. This is a typical size of modern commercial offshore turbines and somewhat larger than turbines

[6] Windmill Furling is a method where the tail of a wind mill is connected to the turbine using a hinge that is angled back from the vertical which causes the turbine to automatically turn out of the wind once a certain wind load threshold is exceeded [109]. Furling is generally not used on large wind turbines.

that are currently deployed at onshore locations. It is however likely that in the future also onshore turbines will reach this size. Table 2.2 summarizes the key properties of this reference turbine configuration.

Rating	5 MW
Rotor Orientation, Configuration	Upwind, 3 Blades
Control	Variable Speed, Collective Pitch
Drivetrain	High Speed, Multiple-Stage Gearbox
Rotor, Hub Diameter	126 m, 3 m
Hub Height	90 m
Cut-In, Rated, Cut-Out Wind Speed	3 m/s, 11.4 m/s, 25 m/s
Cut-In, Rated Rotor Speed	6.9 rpm, 12.1 rpm
Rated Tip Speed	80 m/s
Overhang, Shaft Tilt, Precone	5 m, 5^o, 2.5^o
Rotor Mass	110000 kg
Nacelle Mass	240000 kg
Tower Mass	347460 kg

Table 2.2.: Key Properties of the NREL 5MW Reference Turbine [43]

The NREL 5MW reference turbine has been chosen as the turbine model that is used throughout this study. This ensures maximum comparability to the majority of the wind turbine controls literature that also uses this turbine. It allows a direct comparison and reproduction of the results. Here, the onshore version of the reference turbine is used, as it is expected that all results related to the use of MPC for turbine control are applicable to both onshore and non-floating offshore turbines without modifications so that including the offshore conditions only increases the complexity of the necessary system simulations without providing additional insight.

2.3. Model Predictive Control

Model Predictive Control is a control method that has been used in industrial applications for several decades now [89]. Its basic idea is to use a process model to explicitly predict the outputs of the plant and then use these predictions to calculate a sequence of future control inputs that optimizes the future plant outputs. In general, only the first part of this optimal control sequence is actually applied to the plant and the entire optimization is repeated in the next control interval. MPC is closely linked to many optimal control methods such as LQR control. As in optimal control, instead of a control law, a controller objective based on the resultant performance is specified. In most classical optimal control methods, this control problem is solved analytically and off-line, yielding an optimal control law that only needs to be applied to the process. Contrary to that, using MPC, the optimization is generally performed online using explicit numerical optimization routines. The online solution allows incorporation of control and process constraints, such as actuator limitations or safety critical plant outputs, directly in the control formulation. As these constraints are found in many control applications and are difficult to handle using most other control techniques, it is especially this feature that has contributed to the increasing use of MPC. On the other hand, this online optimization requires significant processing power which is why the use of MPC has initially been limited to comparably ”slow” plants such at those found in the process industries. With the increasing computing power and corresponding drop in prices of modern controls hardware, MPC is now being increasingly used also for faster applications such as walking robots (e.g., [105]) or combustion engines (e.g., [106]).

This section introduces the key concepts of MPC that are relevant for the wind turbine controller design in chapter 5.2. It follows mainly the text books by Maciejowski [70] and Rawlings and Mayne [91]. There exists a large body of literature that deals with both the practical and the more theoretical aspects of MPC. Here, the properties and formulations are only introduced on a level of detail required for the understanding of the specific wind turbine controller design. For a more mathematically rigorous presentation of the material, the respective literature should be consulted.

2.3.1. Models and Problem Formulation

As a model based control method, MPC relies on a process model. While in the early stages impulse response models were often used, now almost the entirety of MPC applications and research is based on discrete state space models

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k, d_k\right) \\ y_k &= h\left(x_k, u_k, d_k\right) \end{aligned} \tag{2.25}$$

where x_k, u_k, d_k, and y_k are the vectors of states, controlled inputs, disturbance inputs, and plant outputs respectively. At any point in time, the goal of the MPC is to determine a sequence of future control moves. The controller only determines the future control inputs for a finite number of time steps, the so-called *control horizon L*. The controller needs to determine all control inputs

$$u_k,\ u_{k+1}, \ldots ,\ u_{k+L}\ . \tag{2.26}$$

The following notation for sequences of vectors is introduced: If the current time step is indicated by k, then $y(k)$ is the value of y at time step k and the vector $y_{m|N}$ is defined as:

$$y_{m|N}(k) = \begin{pmatrix} y_m^T & y_{m+1}^T & \dots & y_{m+N}^T \end{pmatrix}^T . \tag{2.27}$$

To simplify the notation, the explicit dependence on the time step is dropped whenever the vector sequence starts at the current time step which is generally denoted by k:

$$y_{k|N} \equiv y_{k|N}(k). \tag{2.28}$$

At time step k, the controller determines $u_{k|L}$. This sequence of control moves is determined by minimizing a cost function $V\left(x_{k|L}, u_{k|L}\right)(k)$. Only the current and future control inputs $u_{k|L}$ are the decision variables here and the future states $x_{k+1|L}$ depend on the control and disturbance inputs so that the system dynamics (2.25) act as a constraint. Here and in the following, only the regulation problem, i.e., driving the state vector to the origin is considered. If instead of the origin the states were to be driven to a reference trajectory $r_{k|L}$, the objective function would also be a function of this reference trajectory. In addition to the system dynamics constraint, there can also be any number and type of constraint on the control inputs and states so that all values of x and u need to be within the sets $\mathbb{X}$ and $\mathbb{U}$, respectively. The optimization problem that is solved at every time step is thus:

$$\begin{aligned} &\min_{u_{k|L}} V\left(x_{k|L}, u_{k|L}\right) \\ &\text{subject to} \\ &x_{k+1} = f\left(x_k, u_k, d_k\right) \\ &x(i+1) \in \mathbb{X}, \quad u(i) \in \mathbb{U} \qquad \text{for all } i = k \dots L. \end{aligned} \tag{2.29}$$

An MPC controller solves this optimization problem numerically and online. However, the optimization problem in this general form as stated in (2.29) can be extremely challenging to solve as it combines a nonlinear objective function with nonlinear constraints. Not only can it easily lead to computation times that can be prohibitive for an online implementation, the general form of the problem makes it difficult to ensure the chosen algorithm actually finds the solution. Therefore, almost all MPC applications place some conditions on the structure of the objective function, the system dynamics, and the control and state constraints with the most common type being *linear* MPC.

2.3.2. Linear MPC

Linear MPC derives its name from the use of linear state space models

$$\begin{aligned} x_{k+1} &= \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{E}d_k \\ y_k &= \mathbf{C}x_k + \mathbf{D}u_k + \mathbf{F}d_k \end{aligned} \tag{2.30}$$

and constraints

$$\begin{aligned} \mathbf{E_c}u_{k|L} &\leq e_c \\ \mathbf{F_c}x_{k+1|L} &\leq f_c \, . \end{aligned} \tag{2.31}$$

The objective function is chosen to be quadratic in the control and state trajectories:

$$V\left(x_{k|L}, u_{k|L}\right) = u_{k|L}^T \mathbf{R} u_{k|L} + x_{k+1|L}^T \mathbf{Q} x_{k+1|L}. \tag{2.32}$$

Because of the linear system dynamics, the state trajectory depends linearly on the control $u_{k|L}$ and disturbance $d_{k|L}$ trajectories and the current state $x(k)$:

$$x_{k+1|L} = \mathbf{A_p}x(k) + \mathbf{B_p}u_{k|L} + \mathbf{E_p}d_{k|L} \tag{2.33}$$

where prediction matrices $\mathbf{A_p}$, $\mathbf{B_p}$, and $\mathbf{E_p}$ are given by appropriately "stacking" the system matrices from (2.30):

$$\mathbf{A_p} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^L \end{bmatrix} \qquad \mathbf{B_p} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{L-1}\mathbf{B} & \mathbf{A}^{L-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix} \qquad \mathbf{E_p} = \begin{bmatrix} \mathbf{E} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{AE} & \mathbf{E} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{L-1}\mathbf{E} & \mathbf{A}^{L-2}\mathbf{E} & \dots & \mathbf{E} \end{bmatrix} \tag{2.34}$$

Substituting (2.33) in the objective function (2.32) and constraints the optimization problem becomes:

$$\begin{aligned} &\min_{u_{k|L}} \Big(u_{k|L}^T \left(\mathbf{R} + \mathbf{B_p}^T \mathbf{Q} \mathbf{B_p} \right) u_{k|L} + 2u_{k|L}^T \mathbf{B_p}^T \mathbf{Q} \left(\mathbf{A_p}x(k) + \mathbf{E_p}d_{k|L} \right) + \\ &\left(\mathbf{A_p}x(k) + \mathbf{E_p}d_{k|L} \right)^T \mathbf{Q} \left(\mathbf{A_p}x(k) + \mathbf{E_p}d_{k|L} \right) \Big) \\ &\text{subject to} \\ &\mathbf{E_c}u_{k|L} \leq e_c \\ &\mathbf{F_c}\mathbf{B_p}u_{k|L} \leq f_c - \mathbf{F_c}\mathbf{A_p}x(k) - \mathbf{F_c}\mathbf{E_p}d_{k|L}. \end{aligned} \tag{2.35}$$

As $x(k)$ and $d_{k|L}$ are constant, this optimization problem is clearly quadratic in $u_{k|L}$ with linear constraints. This class of optimization problems is called a Quadratic Program (QP) for which there exist numerous efficient and fast numerical solution methods. If further $\mathbf{R}$ and $\mathbf{Q}$ are chosen appropriately, the problem is also convex. This simplifies solving the optimization problem since in convex optimization any local minimum is also the global minimum avoiding the issue of the algorithm not converging to the global minimum. It is precisely because of these benefits that the majority of MPC applications use linear MPC.

2.3.3. Stability

As with any control design technique, designing a Model Predictive Controller should result in a closed loop system that is stable. Even with linear MPC, however, analyzing and ensuring stability is significantly more difficult than for most other controllers.

A system is called input-to-state stable if for any type of bounded input all states will remain bounded [103]. Using this criterion, stability is essentially defined on an infinite time horizon as for a finite horizon usually any signal is bounded. Yet, the MPC formulation introduced in the previous sections, only defines a controller for a finite prediction horizon. So in order to analyze the stability of the controller the system behavior needs to be analyzed on the infinite horizon even if the controller is only defined on a finite horizon.

In essence, according to Mayne et al. [71], using MPC will lead to a stable system if the following criteria are met:

1. The objective function V places a cost on the states for the infinite horizon and not only the prediction horizon, and the contribution of each time step to the overall cost, the so-called stage-wise cost function, is positive definite.
2. This objective function for the infinite horizon is bounded and the additional cost incurred by including an additional time step in the infinite horizon objective function decreases[7].
3. It is ensured that the state and control constrained are not violated on the infinite horizon also.

In practice, this is achieved by making an assumption for how the controller behaves for all time steps past the control horizon:

$$u(i) = \kappa(x(i)) \quad \text{for } i > L. \tag{2.36}$$

[7] This requirement is essentially derived from using the cost associated with the stage-wise infinite horizon as a Lyapunov function.

This assumption has lead to the term *dual-mode* control as now the infinite horizon is split into two parts each using a different controller: The finite horizon where the controlled inputs are determined from solving the constrained optimization problem and the remainder of the infinite horizon where the controlled inputs are given by the control law κ, the so-called terminal controller. It should be noted that due to the moving horizon nature of MPC, the terminal controller is never actually applied to the plant; it is only an assumption that is made to better handle the stability considerations. This terminal controller is assumed to start controlling the system once the final state $x(L)$ of the prediction horizon is reached. If the terminal controller is designed to stabilize the plant, then the cost associated with bringing the system from the terminal state to a rest using this controller is bounded. This cost, which depends only on the terminal state, is termed the *terminal cost* $V_f(x(L))$. If now this terminal cost is added to the objective function, the first and second requirement are already fulfilled. The third requirement is generally fulfilled by placing an additional constraint on the final state $x(L) \in \mathbb{X}$.

How exactly the terminal constraint, controller, and associated cost will need to be set to ensure stability depends heavily on the specific type of plant and the state and control constraints that are used. Rawlings and Mayne [91] list these various requirements with only those relevant for the control design at hand being discussed further here.

To motivate the argumentation, first the simplest possible problem is considered. For a linear plant that is asymptotically stable and has only control constraints but no state constraints, the choice of terminal controller becomes an easy one: It can simply be assumed that no controller is used after the prediction horizon

$$u(i) = 0 \quad \text{for } i > L\,. \tag{2.37}$$

As the plant is asymptotically stable, it will move towards the origin even without any active control and the associated terminal cost is bounded. Further, the third requirement is always met as there are no state constraints and without a terminal controller there is no risk of violating the control constraints.

The linearized plant model that is derived in chapter 4, however, is only marginally stable and further also does have state constraints. Because of the only marginally stable plant, the system will not return to the origin if no terminal controller is used and the terminal cost will not be bounded. Thus, a terminal controller is required. This terminal controller is commonly designed as a Linear Quadratic Regulator (LQR) which will lead to a linear control law $\kappa(x(i)) = -\mathbf{K}_{\text{LQR}} x(i)$. If the cost function for the LQR controller is

$$J = \sum_{i=L+1}^{\infty} x(i)^T \mathbf{Q}_{\text{LQR}} x(i) + u(i)^T \mathbf{R}_{\text{LQR}} u(i) \tag{2.38}$$

then the control matrix $\mathbf{K}_{\mathrm{LQR}}$ can be calculated by solving the discrete algebraic Riccati equation:

$$\mathbf{P} = -\,\mathbf{A}^T\mathbf{P}\mathbf{B}(\mathbf{B}^T\mathbf{P}\mathbf{B} + \mathbf{R}_{\mathrm{LQR}})^{-1}(\mathbf{A}^T\mathbf{P}\mathbf{B})^T + \mathbf{A}^T\mathbf{P}\mathbf{A} + \mathbf{Q}_{\mathrm{LQR}} \tag{2.39}$$

for $\mathbf{P}$ and setting

$$\mathbf{K}_{\mathrm{LQR}} = \left(\mathbf{B}^T\mathbf{P}\mathbf{B} + \mathbf{R}_{\mathrm{LQR}}\right)^{-1}\mathbf{B}^T\mathbf{P}\mathbf{A}^T. \tag{2.40}$$

Designing the terminal controller as a LQR controller has the benefit that it directly provides the terminal cost associated with the controller as

$$V_f(x(L)) = \frac{1}{2}x(L)^T\mathbf{P}x(L). \tag{2.41}$$

Of course, any other choice for a stabilizing linear control law is also possible as long as the terminal cost defined via (2.41) is calculated by solving the associated discrete algebraic Riccati equation. With the control law and terminal cost defined the first and second condition for stability are again fulfilled. Now it only needs to be ensured that by using this terminal controller, the system can be brought to the origin without violating either the control or state constraints. This is generally ensured by choosing a terminal constraint that is a subset of the maximal control admissible set. The initial state of a linear system is control admissible if using this state as the initial condition will not result in violation of either the state or control constraints at any point in time. The maximal control admissible set is the set of all state vectors that meet this condition. The maximal control admissible set for linear systems under linear state and control constraints can for example be calculated using Algorithm 3.1 from Gilbert and Tan [28] which is also reproduced in appendix A.

3. Related Work

As stated in the introductory chapter, one of the key benefits MPC has to offer to the wind turbine control problem is the ability to include feedforward control action in the closed loop controller in an integrated framework. This chapter explores some of the previous research that has been conducted on using feedforward control and MPC for wind turbines. First, wind speed measurements and feedforward control for wind turbines are examined for applications that do not rely on MPC. Then previous studies applying MPC, with and without a preview component, to wind turbines are introduced. Finally, one further promising application of MPC to wind turbine control is control during special events such as shut-down procedures and an overview of existing methods for this control problem is given.

3.1. Feedforward Control

Classical wind turbine speed and power control relies on measurements of the generator speed only [3]. From a control systems point of view, the wind speed fluctuations act as a disturbance which the closed-loop controller needs to reject. However, it is also possible to include control action based on this disturbance directly. This is commonly known as disturbance feedforward control. Feedforward control can be very effective in improving control performance especially if the disturbance can be measured. Further improvements are possible if the disturbance can not only be measured the instant it occurs, but also if it is known some time before it arrives at the plant[1]. Usually, feedforward is implemented as an additional block on top of the feedback control loop. However, it is also possible to incorporate feedforward characteristics directly into the main closed loop controller for example using Model Predictive Control.

Generally, for the use of feedforward two somewhat separate issues have to be solved: The first is the question of how suitable disturbance (wind speed) information can be obtained? The second is how can this information best be used to improve control performance?

[1] e.g, upstream measurements in process applications

All large modern wind turbines are equipped with an anemometer. It is usually located on top of the nacelle behind the rotor. As the wind speed is not uniform over the entire rotor plane, the one point measurement from the anemometer is usually not considered to provide sufficient information about the wind speed driving the turbine dynamics and as such cannot be used directly for turbine control. Therefore, the necessary wind speed information has to be obtained from other sensors.

3.1.1. Wind Speed Measurement

In the last few years, Light Detection and Ranging (LIDAR) [114] has emerged as one of the most commonly used measurement principles for wind speed measurements for wind turbines. Especially the possibility to measure the wind speed at a distance of several hundred meters and being able to measure not just at one point in space but to scan a surface or even volume have been proven to be very valuable for wind energy applications. While at first LIDAR has been mostly considered for an improved accuracy site assessment or power curve validation (e.g., [1, 16]), there are now also several studies that specifically focus on LIDAR measurements as potential controller inputs.

Harris et al. [34] study the potential of turbine mounted LIDAR. Three different setups based on commercially available systems are studied: Two hub mounted system without internal scanning that rely on the turbine rotation for conical scanning and a nacelle mounted system with internal scanning. The systems studied there are capable of measuring at a distance ranging from 40 m to approximately 200 m. While the short ranges are probably not sufficient for preview control, they can still be used for yaw and shear estimation and the longer ranges studied seem more than sufficient to be used for preview control. The study concludes that turbine mounted LIDAR might be economically feasible even if only used for shear regulation via individual pitch feedforward as in this study.

As part of the RAVE project, LIDAR was also tested extensively for variety of applications including turbine mounted LIDAR for control applications [93, 92].

Mikkelsen et al. [73] describes the setup of a test facility in Denmark where a 3D turbulent wind field is measured using several ground based LIDAR systems and already envisions a spinner based LIDAR for control applications. Mikkelsen further reports the first test results from such a spinner mounted forward looking LIDAR [72]. Their system was capable of measuring the oncoming wind field at a distance of 100 m in front of the rotor with a full scan of the measurement cone taking less than one second. Next to the magnitude of the oncoming wind field, the system presented can also measure the instantaneous yaw error which is proposed as an input for both yaw control and individual pitch control.

Simley et al. [102] further analyse the suitability of hub-mounted spinning LIDAR systems as control inputs and quantify the minimum expected measurement errors. If no wind evolution is assumed to occur between wind speed measurement upstream of the turbine and the actual wind field hitting the turbine, the smallest measurement errors for the outboard rotor sections are expected to be at a distance of between 150 m and 200 m in front of the turbine which coincides with the approximate distance required for collective pitch preview control.

3.1.2. Wind Speed Estimation

While the anemometer signal is not sufficiently capturing the effective wind speed, it is possible to derive the effective wind speed from other available signals. This can be seen as using the entire rotor as an anemometer. There are numerous studies that use some form of effective wind speed estimation (see Østergaard [81] for an overview), but most of these techniques rely on measuring the rotor speed or acceleration which together with the commanded generator torque allows an estimation of the aerodynamic torque and subsequently the wind speed if the aerodynamic characteristics of the rotor are known. This type of effective wind speed estimate could also be used to add a feedforward component to the speed control setup and has been suggested in several studies [112, 80, 111, 110]. However, since the wind speed estimate relies on the generator speed, most of the benefits from feedforward, that is the ability to react before the effect of the disturbance is felt at the output of the plant, is lost. It is in fact doubtful whether distinguishing between feedback and feedforward is appropriate at all for situations such as this one where the disturbance is estimated based on the output of the plant that is to be controlled.

Alternatively, it has also been proposed to estimate the local wind speed at each blade and use this as an input for a feedforward component in the individual pitch controller [101]. This type of estimation usually relies on additional load measurements such as main shaft or blade root bending moments sensors.

However, more commonly than for feedforward control, the estimated wind speed is used for improved gain scheduling of linear controllers and improved λ_{opt}-tracking in partial load operation (e.g., [80, 8] and others).

3.1.3. Wind Speed Prediction

As an alternative to actual wind speed measurement for preview applications, it has also been suggested to use wind speed prediction. The estimated effective wind speed or the wind speed measured in the rotor plane is used as a basis for predicting speed ahead over the required interval. In general, the accuracy of such a prediction is expected to be much lower than actual measurements. However, since no additional hardware is required any prediction that beats the consistency assumption of the wind speed remaining fixed at its current value that is either implicitly or explicitly made in almost all turbine controllers, would improve the control performance. Methods that have been used include autoregressive wind models "trained" on recorded wind time series [79] or filtering of the wind data according to its known frequency content in combination with linear [94] or polynomial [15] extrapolation.

3.1.4. Preview Control for Individual Pitch Control

Laks et al. [61] study the use of feedforward from measurement of the tip wind speeds to reduce blade flap loads. They find that if exact measurements of local wind speeds are available, blade loads can be reduced significantly as long as no pitch rate constraints are considered. As the required pitch rates are fairly high and above what most pitch systems are capable of, they conclude that in order to realize these benefits on a realistic system, the individual pitch controller needs to account for rate constraints and also requires knowledge of local wind speeds ahead in time. Finally, in a further study [60] such an individual pitch feedforward controller with preview and explicit bounding of the pitch rates via an LMI formulation is considered. One main result of this study is that the required preview times depend on the allowable pitch rates and that for typical pitch rates around 0.5 seconds of preview time are required.

3.1.5. Preview Control for Speed and Power Regulation

Schlipf and Kühn [98, 96] study the use of preview control[2] for collective pitch control under the assumption of perfect wind speed measurement. They use static system inversion in combination with a predefined time shift as the feedforward component. The mapping of pitch angle to wind speed in stationary conditions is inverted and used as a wind speed dependent feedforward gain. The time shift that gives the best performance was determined to be around 1 second. As such a controller would lead to very aggressive pitching for most types of wind speed fluctuations, the incoming effective wind speed signal is low-pass filtered before being passed to the described feedforward structure. Significant performance improvements in both turbulent time series as well as in gust type events are reported.

Laks et al. [60] also propose a collective pitch preview controller in a simulation study. Their controller is based on the inversion of the gain scheduled transfer function from wind to

[2] called "predictive disturbance compensation" in this study

generator speed with three different realizations of such a controller studied. The wind input to the feedforward controller was an approximate disk average 90 meters ahead of the turbine calculated from 5 measurement points. Similar to the approach by Schlipf, the input to the feedforward controller had to be heavily low pass filtered. They further reported that their approach for collective pitch feedforward has not resulted in a performance improvement possibly due to the low-pass effectively canceling all controller dynamics or the wind speed measurement not being representative of the true effective wind speed. In a follow-up study [19], several methods to improve the system inversion approach are examined with some improvement in the control performance. One of the limiting factors stated in this study is the SISO nature of the inverse transfer function approach and the missing possibility to trade the conflicting controller objectives such as speed regulation, pitch actuator usage, and mechanical loads off against each other.

Wang et. al. [115] also address the design of a collective pitch feedforward controller that is added on top of an existing feedback control algorithm. Two different algorithm are studied: The first is a dynamic system inversion approach that, similar to the earlier results by Laks et. al. [60], does not provide a significant improvement in performance. The second algorithm is an adaptive FIR-filter based algorithm. A recursive least squares algorithm is used to continuously adjust the feedforward path controller based on the speed tracking performance in the feedback path. This approach leads to significant improvement in performance showing that the design of the feedforward controller does need to consider the feedback controller as well.

Finally, in the summer of 2012, Schlipf et. al. performed the first reported field test of a LIDAR enabled feedforward controller [97]. They used the static system inversion approach with a fixed time shift to design a feedforward controller for the CART2 600 kW turbine at NREL and could show a reduction in the rotor speed deviation and tower loads at low frequency for a small measurement period.

3.2. Wind Turbine Control

State-feedback was among the earliest methods used for the wind turbine control problem. While pole placement has been used [41, 117], it is usually hard to specify the pole location required for the wind turbine control objectives. The more common approach is to formulate the problem as an optimal control problem either in LQ [30, 88, 82, 77, 107] or in H_∞ form [27, 30, 13]. The key benefit of using optimal control design techniques was that they provide ”management level tuning knobs” [31, 32] that link the controller design and tuning directly to the performance objectives such as load reduction and speed tracking performance.

Most of these controllers face the same challenge as using linear MPC: They are based on an LTI system description of the plant dynamics and can therefore only be designed for a single operating point. In order to use them, they have to be gain-scheduled or interpolated between the different operating points. Another option is the use of a single sufficiently robust controller for the whole operating region. Although it is possible to find a single stabilizing controller, this will usually lead to sub-optimal performance in all operating points except the nominal operating point. Instead of an LTI formulation, both Lescher et. al. [68, 69, 67] and Østergaard et. al. [80, 83] use LPV-methods to design a robust controller. Through the use of one system description for all operating points, the gain-scheduling is here done implicitly. Both studies showed a significant reduction in loads compared to LQ and PI controllers.

3.2.1. MPC for Wind Turbines

There are several studies investigating the potential application of Model Predictive Control to wind turbines for both non-preview and preview applications.

Henriksen [36] describes the development of a linear MPC for a wind turbine. One linear MPC was developed for each operating region and tested for both floating and land based turbines. Although the nonlinearity of the wind turbine has not been taken into account when the controller was designed, the study shows a single linear non-robust MPC controller to be stable for the entire full load operating range. The study also demonstrates how off-set free tracking of the desired rotor speed can be achieved in MPC even with the wind acting as a persistent disturbance through disturbance estimation and origin shifting. This MPC application is later [37] expanded to include soft state constraints and a re-linearization scheme. Henriksen also states that the implementation of a hard constraint on the rotor speed state is difficult due to potential infeasiblity of the control problem, an issue that is addressed at length in this thesis.

Botasso et al. [7] use an adaptive controller that includes some elements from nonlinear MPC but does not include any constraints in the problem formulation. Using the results obtained with this controller, they conclude that for nominal operating conditions it is hard to beat a well-tuned classical controller.

Kumar and Stol [54] present a scheduled Model Predictive Controller for full load operation. The controller consists of three separate linear MPCs, each similar to the controller presented by Henriksen and based on a turbine model linearized at a different wind speed, that are run in parallel. Using the estimate of the effective wind speed obtained from an EKF, the output from each of these controllers is weighted according to the proximity to the respective linearization wind speeds. The final controller output is then the sum of the three weighted controller outputs. The performance of the scheduled MPC is compared against a single linear MPC and a scheduled PI controller. The results show a slightly higher performance of the scheduled MPC over the two other variants. The authors further argue that an additional benefit of the scheduled MPC is the ability to tune each controller individually so that the response at different wind speeds can be tailored to specific requirements.

Friis et al. [25] propose to use linear MPC to design a repetitive controller for individual pitch control. Unlike the previous MPC applications, the wind speed disturbance states, which are estimated using a Kalman Filter, are included directly in the MPC formulation and thereby adding a feedforward component to the overall control scheme. This study further includes both output and state constraints, although no details are given with respect to how the state constraints are handled in the presence of unknown disturbances or estimation errors. In the proposed scheme, stability is achieved using a sufficiently long prediction horizon. The results show a significant reduction of vibration levels for several components.

Körber and King [50] first show that that preview information can be easily integrated into a linear MPC for the full load operating region. Under the assumption of perfect wind speed measurements five seconds ahead of the turbine, large reductions in both fatigue and extreme loads were observed. If no preview information was available, a small performance improvement of the MPC over a classically-designed baseline controller remained. In a follow-up study [51], the controller was expanded to also cover partial load operation and several schemes to handle the wind turbine nonlinearity were compared with two main results: Firstly, preview MPC can increase the tracking of the optimum tip speed ratio in partial load drastically but, for the examined turbine, this does not lead to any significant increase in power production. Secondly, using a scheduled or continuously linearized MPC did not result in a significant loss of performance compared to a fully nonlinear MPC.

Spencer et al. [104] use a linear MPC to design a combined speed and power controller and individual pitch controller that assumes perfect measurements of the wind field up to five seconds before it arrives at the turbine to be available. It is found that large load reductions are possible and also that the benefit of preview-MPC is mainly seen in the full load operating region.

Laks et al. [59] as well as as Schlipf et al. [99] use MPC to design a preview controller where instead of assuming perfect measurement of the wind speed, a more realistic model for the preview measurements is assumed. Both studies conclude that even in the presence of imperfect wind speed measurement, large reductions in both extreme and fatigue loads are possible.

Finally, many of the concepts introduced and evaluated in this thesis, such as the robust handling of the over-speed constraint and the linearization scheme, were introduced and published by Körber and King [52] in 2013.

To the knowledge of the author no one has tested a Model Predictive Controller on an actual turbine so far.

3.2.2. Extreme Load Control

Given their importance for the overall wind turbine design, very little academic research has been performed on the design of wind turbine controllers that reduce extreme loads. Of the studies that do consider the non-NPP load cases, most deal with handling gust cases (e.g., load cases 1.3, 1.6, 1.9[3]). However, gust cases are mostly considered when designing controllers that make use of additional sensing means, such as the previously discussed LIDAR-based preview controllers or increased actuation capability through the use of smart rotors [55].

Kanev and van Engelen [47] present a method to improve controller performance during extreme gust without additional sensing or actuation for a regular pitch controlled variable speed wind turbine. Their concept relies on an EKF-based wind condition observer that is used to detect extreme events. Once such an event is detected, a special extreme event controller is activated that is designed to prevent the turbine from causing an overspeed fault. Control during these events essentially consists of pitching out the blades at the maximum rate and holding the generator torque constant. One reported down-side of this open-loop scheme is that it can cause unnecessarily high tower loads because the pitch is controlled at the maximum possible rate and not at the rate necessary to prevent the overspeed. Furthermore, because the controller now has two discrete operating modes, toggling between modes can occur and a hysteresis is required. Both of these down-sides are addressed when using the MPC developed in this thesis for the same control problem of overspeed avoidance during gust events.

While the idea of maintaining closed-loop control of the rotor speed during emergency stops is almost as old as the wind turbine industry [53], no academic studies investigating the design of such a shut-down controller are known to the author.

[3] see table 2.1 in section 2.2.2

4. Turbine Model

The Model Predictive Controller relies on the full state and disturbance information being available at every time step. While all modern turbines are equipped with an anemometer, which is usually mounted on the nacelle, the signal from such an anemometer is usually not considered to be suitable for turbine control for several reasons: Firstly, the anemometer only measures at one point of the rotor plane. The wind speed can vary significantly across the area swept by the rotor and measurements taken at a single point in this rotor plane are not necessarily representative of the rotor area effective wind speed. Secondly, the anemometer is usually placed behind the blades so that all disturbance created by the passing blades will be visible in the signal. Finally, by extracting energy from the wind, the turbine causes the wind to slow down significantly. The controller however would require the free stream wind speed which can only be measured at a significant distance ahead of the turbine. While numerous methods for state and disturbance estimation exist, an Extended Kalman Filter is chosen here as it is the most straight-forward and widely employed method used for wind speed estimation on wind turbines (see section 3.1.2).

Thus, as shown in figure 4.1, the full controller consists of the EKF based estimator and the Model Predictive Controller.

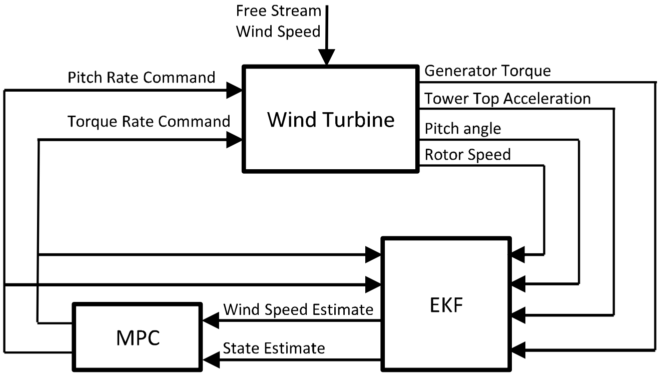


Figure 4.1.: High-Level Controller Structure

Both the EKF and the MPC are model based techniques: They rely on an internal model of the plant for which the states are to be estimated and which is to be controlled. In the first section of this chapter, the dynamic model and its parameters, used by both the estimator and controller, are discussed.

As outlined in section 2.2, normally complex models with highly coupled structural and aerodynamic components, as implemented in specialized software like BLADED, FLEX5, or FAST, are used to evaluate turbine dynamics. These models are too complex to be implemented directly in an MPC scheme. Instead, most control designs for both MPC and non-MPC approaches use models that are based on the models used in the system simulations but are significantly reduced in complexity.

There are two main distinct approaches to generate models that are suitable for the control design which are frequently used:

- Some of the commonly used simulation tools (especially BLADED and FAST) provide routines that allow an automatic generation of linear state space models for a given operating point. The resulting linear models will generally still be fairly large and may consist of several hundred states, yet can be reduced using model order reduction techniques. The main advantages of this approach are the ease at which models can be generated and the fact that all dynamic interactions are included. In the MPC for wind turbines context, this approach has for example been used by Kumar and Stol [54] and Spencer [104].

- The second approach is to use a first principle model of the turbine and identify its parameters from the aero-elastic code using system identification routines. By choosing to ignore certain interactions that are deemed not to be relevant for the control problem at hand, models with just a few states are possible using this approach. Another advantage is that using first principle models makes it is easier to include the nonlinearities of the model explicitly and thus allows more flexibility in handling the nonlinearities in the control design. First principle modeling is the technique that has been used by Henriksen [36, 37] or Friis et al. [25].

In this study, the second approach is chosen because of the smaller size of the resulting model and explicit inclusion of nonlinearities. How to design scheduled MPC based on linear models at different operating points is outlined in [54]. The remainder of this section describes and analyses the simplified first principle model that is used in the control design. One benefit of using a first principle model is that the model can be specifically tailored to the needs of the control problem to be solved. The model presented in this chapter has been chosen to allow a design of an MPC for the speed, power, and tower vibration control problem. It would however require some extensions if other control problems such as individual pitch control or drive train damping are to be considered.

This simplified model consists of linear structural dynamics and actuator models coupled with a nonlinear aerodynamic model as illustrated in figure 4.2.

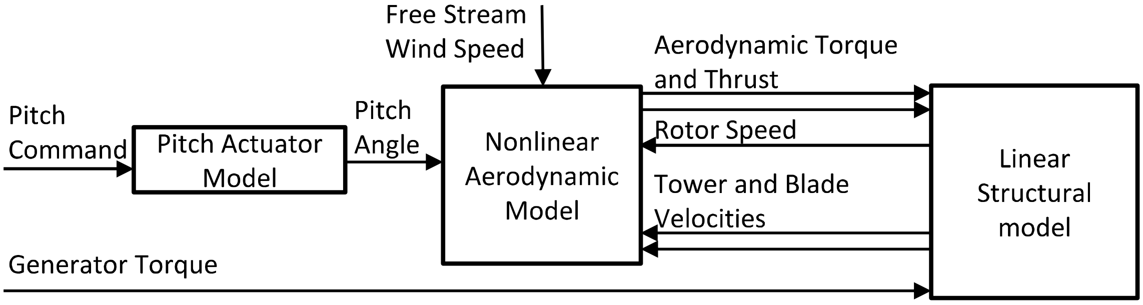


Figure 4.2.: Simplified turbine model used by the controller

The following sections will describe each of the sub models and explain how its parameters are identified from the full aero-elastic simulation tool.

4.1. Aerodynamics

The output of the aerodynamic model are the aerodynamic torque and thrust generated by the rotor. These are mainly dependent on the current local inflow conditions at each section of the blades. Instead of treating each blade section individually, as would be done using Blade Element Momentum (BEM) theory, and calculating the local aerodynamic forces, only the resulting torque and thrust generated by the entire rotor is considered. The aerodynamic torque M_A and thrust F_A are defined in terms of the torque and thrust coefficients C_M and C_T [26]

$$\begin{aligned} M_A &= \frac{1}{2}\rho\pi R^3 v_e^2 C_M(\lambda,\theta) \\ F_A &= \frac{1}{2}\rho\pi R^2 v_e^2 C_T(\lambda,\theta). \end{aligned} \tag{4.1}$$

where v_e is the *effective* wind speed, ρ the air density and R the rotor radius. The effective wind speed v_e is a modification of the free stream wind speed v that was used in section 2.1.2 to define the aerodynamic characteristics of the wind turbine that also includes the effect of the structural turbine motion on the aerodynamics. It will be properly defined in section 4.3. The coefficients themselves are a function of the current collective pitch angle θ

and the tip speed ratio λ:

$$\lambda = \frac{\omega R}{v_e}. \tag{4.2}$$

4.2. Structural Dynamics

Only three degrees of freedom are considered to model the structural dynamics of the turbine. The first describes the rotation of the rotor, consisting of the three blades and all drive train components about its principal axis of rotation. The other two model the fore-aft dynamics of the tower and rotor, respectively.

4.2.1. Rotor

The rotor model lumps all rotating components into a single degree of freedom. On a geared turbine, the generator and so-called high speed shaft runs at a higher rotational speed ω_G than the main shaft, hub, and attached rotor blades. Here, it is assumed that both sides are rigidly coupled through the gearbox and any potential flexibility is ignored.

$$\omega_G = n_G \omega \tag{4.3}$$

By ignoring the drive train flexibility, the generator speed ω_G does not have to be included as a separate degree of freedom since it can always be expressed in terms of the rotor speed ω. Although on most turbine the generator speed and not the rotor speed is measured and used as a controller input, in the following all equations are expressed in terms of the rotor speed and it is understood that rotor and generator speed can be used interchangeably.

It should be noted that some modern wind turbines do not use a gearbox. The drive train model introduced here however remains applicable even for these so-called direct-drive turbines since a turbine without a gearbox is essentially a turbine with a gearbox ratio of $n_G = 1$.

It should further be noted that some studies on wind turbine control design (e.g., [8]) do include flexibility in the drive train and model the rotor as a two-mass system. However, this is necessary only if the drive train damping problem is to be considered. As the drive train damping functionality is not to be included in the MPC, modeling the drive train flexibility is not required and the additional degree of freedom is omitted in order to keep the model as small as possible.

The inertias about the principal axis of rotation for the three blades J_B, all rotating components on the slow speed shaft J_{SS} and high speed shaft, including the generator, J_{HS} are combined into a single inertia J. Since this effective inertia is associated with the rotational speed on the slow speed side of the gearbox, the gearbox ratio needs to be taken into account when including the high speed side inertia:

$$J = 3J_B + J_{SS} + n_G^2 J_{HS}. \tag{4.4}$$

The generator torque M_G and aerodynamic torque M_A are the moments acting on this rotational drive train model:

$$J\dot{\omega} = M_A(\theta, v_e, \omega) - n_G M_G. \tag{4.5}$$

Equation (4.5) represents the simplest model that can be used to design a speed controller for a wind turbine with M_G and θ being the controlled variables, v_e the disturbance, and ω and $P = \omega M_G$ the plant outputs that are to be controlled. It has, however, been shown that especially for large turbine structural dynamics of the tower and blades do have a significant impact on even the pure speed control problem [65]. Hence, the pure drive train model is augmented with tower and blade fore-aft models. The impact of the blade and tower dynamics on the rotor speed dynamics are further analyzed later in this section.

4.2.2. Tower

The aerodynamic thrust causes movement of the wind turbine head mainly in the direction of the wind speed, the so-called fore-aft direction. This fore-aft motion of the tower is modeled as a single mass oscillator as shown in figure 4.3. Using this single degree of freedom model for the tower motion essentially treats the entire tower as a linear spring and ignores the higher modes of the flexible tower. However, due to the low energy content, the impact of the higher modes is comparatively small and can generally be ignored in the control design.

Instead of using the mass m, stiffness k, and damping properties d the resulting equation of motion is directly given in terms of the natural frequency $\omega_{T,0}$, damping ratio ζ_T, and forcing gain K_{fa}

$$\ddot{x}_t + 2\zeta_T \omega_{T,0} \dot{x}_t + \omega_{T,0}^2 x_t = K_{fa} \omega_{T,0}^2 F_A. \tag{4.6}$$

Under certain circumstances, the machine head of a wind turbine may also sway sideways. This so-called side-side tower oscillation is significantly smaller then the fore-aft movement

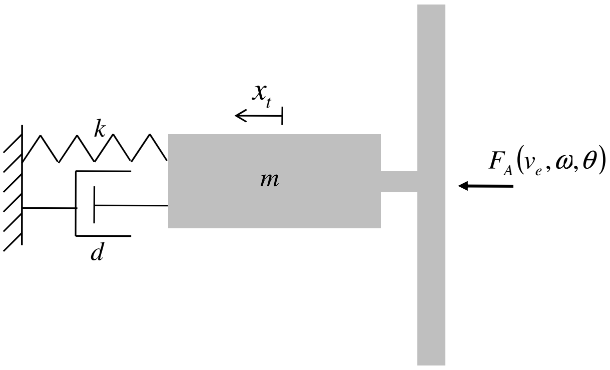


Figure 4.3.: Simplified tower model

and, as it is perpendicular to the wind direction, has very little interaction with the turbine aerodynamics. It is therefore ignored here.

4.2.3. Blade Fore-Aft Motion

The bending motion of a flexible wind turbine blade is usually decomposed into two principal axis of deflection. If the blades have not been pitched, "edgewise" deflection takes place in the plane of rotation while "flapwise" motion describes the motion out of the plane of rotation.

Depending on the control problem at hand, different blade models need to be included in the overall turbine model. While there are some phenomena like controls induced blade flutter that are a result of the dynamic coupling between structural dynamics, aerodynamics and controls that require detailed nonlinear models of the blades like the one developed by Kallesoe [46], most control designs only require rudimentary blade models to be included in the turbine model.

For the design of a drive train damper, the edgewise motion of the three blades should be included as it is heavily coupled with the drive train dynamics, while the design of an individual pitch controller requires a model of the individual motion of each blade in the flapwise direction. Since the tower fore-aft motion is coupled with the flapwise motion of the blades, the flapwise motion should also be included in a structural model for speed control and tower damping [65].

Geyler and Caselitz [27] augment their single degree-of-freedom tower model with a single degree-of-freedom model for the flapwise deflection of each of the blades. They then transform the three individual blade tip deflections in the flapwise direction $x_{b1} - x_{b3}$ into a collective tip deflection x_b, which describes the average tip flapwise deflection, as well as the sine x_{bs} and cosine x_{bc} components of the deviations from the collective deflection.

$$\begin{pmatrix} x_b \\ x_{bs} \\ x_{bc} \end{pmatrix} = \begin{bmatrix} 1 & \sin(\psi) & \cos(\psi) \\ 1 & \sin(\psi + \frac{2}{3}\pi) & \cos(\psi + \frac{2}{3}\pi) \\ 1 & \sin(\psi + \frac{4}{3}\pi) & \cos(\psi + \frac{4}{3}\pi) \end{bmatrix}^{-1} \begin{pmatrix} x_{b1} \\ x_{b2} \\ x_{b3} \end{pmatrix} \tag{4.7}$$

where ψ is the rotor azimuth.

Here, since individual pitch control is not treated, only the collective flap motion x_b is considered and the asymmetric components x_{bs} and x_{bc} are ignored in order to reduce model complexity. The equation of motion for this collective flap deflection is derived using a single blade, but it is implied that all three blades behave identically and are lumped into this one degree-of-freedom.

Figure 4.4 shows the simple model. The blade is modeled as a single stiff body with rotational inertia J and mass m_b connected to the tower via the reactive force R, elasticity k, and damping d. The position of the tower itself is x_t. The motion can be described via the tip deflection of the blade x_b which is measured in the reference frame with its origin at the tower top and is thus not an inertial coordinate system. A deflection angle

$$\varphi = \arctan\left(\frac{x_b}{l}\right) \tag{4.8}$$

is introduced and it is immediately assumed that the deflections angles are small so that the assumption

$$\varphi \approx \frac{x_b}{l} \tag{4.9}$$

can be made. It is important to mention that this angle has no physical meaning since on the real turbine the blade is not stiff and straight but flexible and curved and the bending angle at the hub will always be zero. For the same reason, the scaling factor a_1 has to be introduced which describes the translatory motion of the center of gravity with respect to the tip deflection

$$x_{cog} = a_1 x_b. \tag{4.10}$$

In order to derive the equations of motion D'Alembert's principle of inertial forces is employed where the inertial forces caused by the translatory accelerations $\ddot{x}_t$ and $\ddot{x}_b$ and the angular acceleration $\ddot{\varphi}$ are included in the free body diagram as if they were external forces. The system is then treated as a static system with $\sum F = 0$ and $\sum M = 0$.

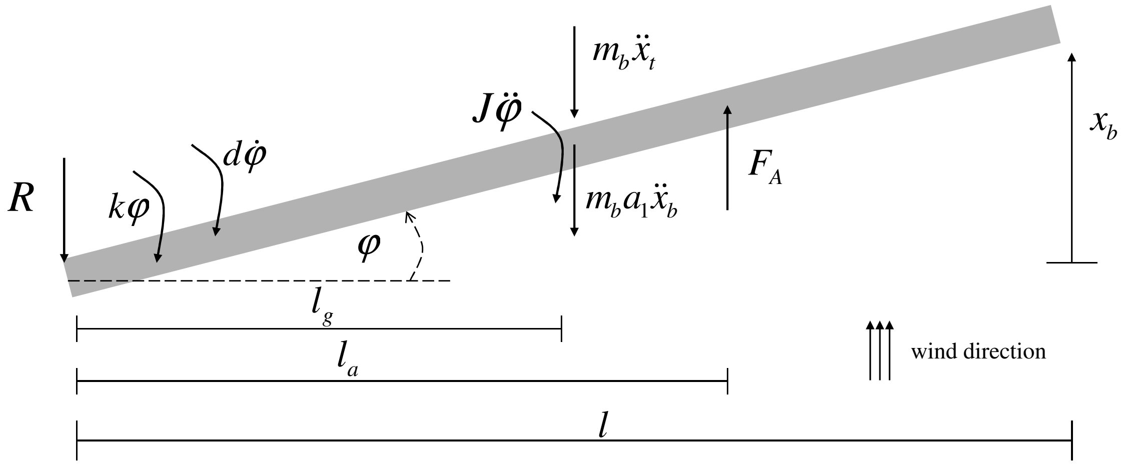


Figure 4.4.: Blade Model

The equation of motion for this system is now derived by taking the balance of moments about the blade root:

$$-J\ddot{\varphi} - k\varphi - d\dot{\varphi} - m_b a_1 l_g \ddot{x}_b + l_a F_A - m_b l_g \ddot{x}_t = 0 \tag{4.11}$$

which can be written as

$$\ddot{x}_b + 2\omega_{0,b}\zeta_b\dot{x}_b + \omega_{0,b}^2 x_b = k_b\omega_{0,b}^2(F_A - m_b\frac{l_g}{l_a}\ddot{x}_t) \tag{4.12}$$

using

$$\begin{aligned}\omega_{0,b}^2 &= \frac{k}{J + m_b l_g a_1 l}\\ \zeta_b &= \frac{d\omega_{0,b}}{2k}\\ k_b &= \frac{l_a l}{k}.\end{aligned} \tag{4.13}$$

The force acting on the tower R becomes:

$$R = F_A - m_b a_1 \ddot{x}_b - \underbrace{m_b\ddot{x}_t}_{\text{already included in tower model}}. \tag{4.14}$$

where the inertial force of the blades $m_b\ddot{x}_t$ with respect to the the tower motion is already included in the mass of the tower system (4.6). Equation (4.12) can be combined with the tower dynamics (4.6) to form the two mass oscillator shown in figure 4.5. The combined

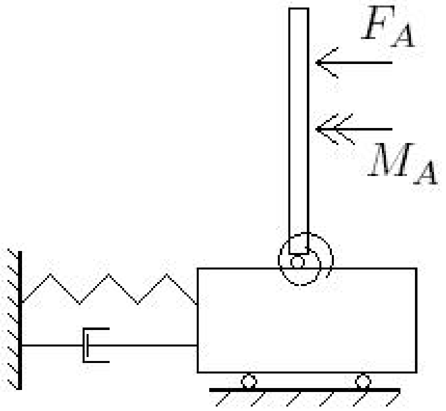


Figure 4.5.: Mechanical Turbine Model

equations in motion can be written in matrix notation:

$$\underbrace{\begin{bmatrix} 1 & K_{fa}\omega_{T,0}^2 m_b a_1 \\ m_b \frac{l_g}{l_a} k_b \omega_{0,b}^2 & 1 \end{bmatrix}}_{\mathbf{M}} \begin{pmatrix} \ddot{x}_t \\ \ddot{x}_b \end{pmatrix} + \underbrace{\begin{bmatrix} 2\omega_{T,0}\zeta_T & 0 \\ 0 & 2\omega_{0,b}\zeta_b \end{bmatrix}}_{\mathbf{D}} \begin{pmatrix} \dot{x}_t \\ \dot{x}_b \end{pmatrix}$$
$$+ \underbrace{\begin{bmatrix} \omega_{T,0}^2 & 0 \\ 0 & \omega_{0,b}^2 \end{bmatrix}}_{\mathbf{K}} \begin{pmatrix} x_t \\ x_b \end{pmatrix} = \underbrace{\begin{bmatrix} K_{fa}\omega_{T,0}^2 \\ k_b \omega_{0,b}^2 \end{bmatrix}}_{\mathbf{F}} F_A \tag{4.15}$$

$$\mathbf{M} \begin{pmatrix} \ddot{x}_t \\ \ddot{x}_b \end{pmatrix} + \mathbf{D} \begin{pmatrix} \dot{x}_t \\ \dot{x}_b \end{pmatrix} + \mathbf{K} \begin{pmatrix} x_t \\ x_b \end{pmatrix} = \mathbf{F} F_A \tag{4.16}$$

4.3. Effective Wind Speed

The wind speed ”seen” by the rotor will change with tower and blade motion. If the turbine moves towards the wind, the relative wind speed at the rotor is increased by the tower top velocity $\dot{x}_t$. Similarly, blade deflection in the flapwise direction will also effect the relative wind speed at the blades. Therefore, the effective wind speed v_e is defined as the rotor plane average of the free stream wind speed v corrected for tower and blade motion

$$v_e = v - \dot{x}_t - a\dot{x}_b. \tag{4.17}$$

If the blade is deflected with $\dot{x}_b$ in the flapwise direction, only the tip will see a change in relative wind velocity of $\dot{x}_b$; all other parts of the blade will move at a lower velocity.

Therefore the scaling constant a is introduced to express the total change in effective wind speed as a fraction of the blade tip out-of-plane velocity. If the blade were actually stiff, a would be equal to the location of the aerodynamic center relative to the blade length $a = \frac{l_a}{l}$.

4.4. Actuator Models

The pitch system is modeled as a linear second order system relating the commanded pitch angle θ_C to the actual pitch angle θ:

$$\ddot{\theta} + 2\omega_{p,0}\zeta_p\dot{\theta} + \omega_{p,0}^2\theta = \omega_{p,0}^2\theta_C. \tag{4.18}$$

The electrical dynamics are considered to be significantly faster than structural dynamics, so it is assumed that the actual generator torque M_G directly follows the torque set-point $M_{G,C}$:

$$M_G = M_{G,C}. \tag{4.19}$$

For the MPC, the constraints on the commanded actuator rates play an important role. In order to allow treating these rate constraints as simple bounds on the controlled variables, equations (4.18) and (4.19) are differentiated with respect to time

$$\begin{aligned} \dddot{\theta} + 2\omega_{p,0}\zeta_p\ddot{\theta} + \omega_{p,0}^2\dot{\theta} &= \omega_{p,0}^2\dot{\theta}_C \\ \dot{M}_G &= \dot{M}_{G,C} \end{aligned} \tag{4.20}$$

and the commanded pitch $\dot{\theta}_C$ and torque $\dot{M}_{G,C}$ rates are seen as the controller inputs to the model.

4.5. State Space Formulation and Linearization

The turbine model, consisting of equations (4.20), (4.17), (4.16), (4.5), and (4.1), is now combined into a single state space model by defining the states shown in table 4.1. Defining the state vector

$$x = \begin{pmatrix}\omega & \theta & \dot{\theta} & \ddot{\theta} & x_t & x_b & \dot{x}_t & \dot{x}_b & M_G\end{pmatrix}^T, \tag{4.21}$$

x_1	Rotor Speed ω
x_2	Collective Pitch Position θ
x_3	Collective Pitch Rate $\dot{\theta}$
x_4	Collective Pitch Acceleration $\ddot{\theta}$
x_5	Long. Tower Position x_t
x_6	Collective Flap Position x_b
x_7	Long. Tower Velocity $\dot{x}_t$
x_8	Collective Flap Velocity $\dot{x}_b$
x_9	Generator Torque M_G

Table 4.1.: Summary of turbine model states

a vector of controlled inputs

$$u = \begin{pmatrix} \dot{M}_{G,C} & \dot{\theta}_C \end{pmatrix}^T, \tag{4.22}$$

and vector of disturbance inputs

$$d = \begin{pmatrix} v \end{pmatrix} \tag{4.23}$$

the turbine model can be represented in the standard form $\frac{dx}{dt} = f(x, u, d)$ with the following state evolution equations

$$\begin{aligned}
\dot{x}_1 &= \frac{1}{J}\left(M_A(x_1, x_2, x_7, x_8, v) - n_G x_9\right) \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -\omega_{p,0}^2 x_3 - 2\omega_{p,0}\zeta_p x_4 + \omega_{p,0}^2 \dot{\theta}_C \\
\dot{x}_5 &= x_7 \\
\dot{x}_6 &= x_8 \\
\begin{pmatrix} \dot{x}_7 \\ \dot{x}_8 \end{pmatrix} &= \mathbf{M}^{-1}\left(\mathbf{F}F_A(x_1, x_2, x_7, x_8, v) - \mathbf{K}\begin{pmatrix} x_5 \\ x_6 \end{pmatrix} - \mathbf{D}\begin{pmatrix} x_7 \\ x_8 \end{pmatrix}\right) \\
\dot{x}_9 &= \dot{M}_{G,C}.
\end{aligned} \tag{4.24}$$

y_1	Rotor Speed ω
y_2	Turbine Power P
y_3	Collective Pitch Position θ
y_4	Generator Torque M_G
y_5	Long. Tower Velocity $\dot{x}_t$
y_6	Collective Flap Velocity $\dot{x}_b$
y_7	Long. Tower Acceleration $\ddot{x}_t$

Table 4.2.: Summary of turbine model outputs

Table 4.2 lists all quantities that are seen as turbine outputs. These are combined in the system output vector y and the output equation is also given in its standard form $y = h(x, u, d)$ when $\dot{x}_7$ is replaced by (4.24):

$$\begin{aligned}
y_1 &= x_1 \\
y_2 &= x_1 n_G x_9 \\
y_3 &= x_2 \\
y_4 &= x_9 \\
y_5 &= x_7 \\
y_6 &= x_8 \\
y_7 &= \dot{x}_7.
\end{aligned} \tag{4.25}$$

Only a subset of these outputs is measurable. The distinction between measurable and unmeasurable outputs will need to be taken into account when designing the state and disturbance estimator (see section 5.1) where only measurable outputs are considered and the designing of the actual controller (see section 5.2) where a full state controller can have its performance specified based on both measurable and unmeasurable outputs. Here, only the quantities which are commonly measured on modern wind turbines are assumed to be measurable: rotor speed, power output, pitch angle, generator torque, and tower top acceleration. This defines the vector of measurable outputs z:

$$z = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 & y_7 \end{pmatrix}^T \tag{4.26}$$

The design for both the controller and the state and disturbance estimator relies on linearizing the turbine model at various operating points. In the described model the entire nonlinearity is contained in the aerodynamics. The structural and actuator models are fully linear. Therefore, in order to linearize the system dynamics only the aerodynamics need to be linearized which can easily be performed analytically.

As the torque and thrust coefficients C_M and C_T in (4.1) depend on the pitch angle and tip speed ratio, which in turn depends on the effective wind speed and rotor speed, the aerodynamic torque and thrust depend nonlinearly on effective wind speed v_e, rotor speed

ω, and pitch angle θ. The partial derivatives with respect to these variables are given by:

$$\begin{aligned}
M_A &= \frac{1}{2}\rho\pi R^3 v_e^2 C_M(\lambda,\theta) \\
\frac{\partial M_A}{\partial \omega} &= \frac{1}{2}\rho\pi R^4 v_e \frac{\partial C_M}{\partial \lambda} \\
\frac{\partial M_A}{\partial \theta} &= \frac{1}{2}\rho\pi R^3 v_e^2 \frac{\partial C_M}{\partial \theta} \\
\frac{\partial M_A}{\partial v_e} &= \rho\pi R^3 v_e C_M - \frac{1}{2}\rho\pi R^4 \frac{\partial C_M}{\partial \lambda}\omega
\end{aligned} \tag{4.27}$$

and

$$\begin{aligned}
F_A &= \frac{1}{2}\rho\pi R^2 v_e^2 C_T(\lambda,\theta) \\
\frac{\partial F_A}{\partial \omega} &= \frac{1}{2}\rho\pi R^3 v_e \frac{\partial C_T}{\partial \lambda} \\
\frac{\partial F_A}{\partial \theta} &= \frac{1}{2}\rho\pi R^2 v_e^2 \frac{\partial C_T}{\partial \theta} \\
\frac{\partial F_A}{\partial v_e} &= \rho\pi R^2 v_e C_M - \frac{1}{2}\rho\pi R^3 \frac{\partial C_T}{\partial \lambda}\omega.
\end{aligned} \tag{4.28}$$

Using these partial derivatives, the entire system can be represented in the standard form of a linear state space system:

$$\begin{aligned}
\dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{E}d(t) \\
y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + \mathbf{F}d(t).
\end{aligned} \tag{4.29}$$

The matrices of this linear state-space model are given by the partial derivatives of the system and output equations with respect to the state, output and disturbance vectors. As the the system is linear with respect to the control inputs u, the linearized system only depends on the chosen state vector x_0 and wind speed v_0 at which the system is linearized.

$$
\begin{aligned}
\mathbf{A} = \left.\frac{\partial f}{\partial x}\right|_{x_0,v_0} = & \\
&\begin{bmatrix}
\frac{1}{J}M_\omega & \frac{1}{J}M_\theta & 0 & 0 & 0 & 0 & -\frac{1}{J}M_v & -\frac{1}{J}aM_v & \frac{-n_G}{J} \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\omega_{p,0}^2 & -2\omega_{p,0}\zeta_p & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
f_{t1}F_\omega & f_{t1}F_\theta & 0 & 0 & -k_{t11} & -k_{t12} & -d_{t11} - f_{t1}F_v & -d_{t12} - f_{t1}F_v & 0 \\
f_{t2}F_\omega & f_{t2}F_\theta & 0 & 0 & -k_{t21} & -k_{t22} & -d_{t21} - af_{t2}F_v & -d_{t22} - af_{t2}F_v & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \\
\mathbf{B} = \left.\frac{\partial f}{\partial u}\right|_{x_0,v_0} = & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \omega_{p,0}^2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\
\mathbf{E} = \left.\frac{\partial f}{\partial d}\right|_{x_0,v_0} = & \begin{bmatrix} \frac{1}{J}M_v & 0 & 0 & 0 & 0 & 0 & f_{t1}F_v & f_{t2}F_v & 0 \end{bmatrix}^T \\
\mathbf{C} = \left.\frac{\partial h}{\partial x}\right|_{x_0,v_0} = & \\
&\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
M_{g,0}n_G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_{r,0}n_G \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
f_{t1}F_\omega & f_{t1}F_\theta & 0 & 0 & -k_{t11} & -k_{t12} & -d_{t11} - f_{t1}F_v & -d_{t12} - f_{t1}F_v & 0
\end{bmatrix} \\
\mathbf{D} = \left.\frac{\partial h}{\partial u}\right|_{x_0,v_0} = & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\
\mathbf{F} = \left.\frac{\partial h}{\partial d}\right|_{x_0,v_0} = & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & f_{t1}F_v & 0 & 0 \end{bmatrix}^T
\end{aligned}
\tag{4.30}
$$

where k_{tij}, d_{tij}, and f_{ti} are the entries of the transformed matrices

$$
\mathbf{K_t} = \mathbf{M}^{-1}\mathbf{K} \qquad \mathbf{D_t} = \mathbf{M}^{-1}\mathbf{D} \qquad \mathbf{F_t} = \mathbf{M}^{-1}\mathbf{F}. \tag{4.31}
$$

and the terms

$$\begin{aligned} M_v &= \frac{\partial M_A}{\partial v_e}, \quad M_\theta = \frac{\partial M_A}{\partial \theta}, \quad M_\omega = \frac{\partial M_A}{\partial \omega} \quad \text{and} \\ F_v &= \frac{\partial F_A}{\partial v_e}, \quad F_\theta = \frac{\partial F_A}{\partial \theta}, \quad F_\omega = \frac{\partial F_A}{\partial \omega} \end{aligned} \tag{4.32}$$

denote the partial derivatives of the aerodynamic torque M_A and thrust F_A as defined in (4.27) and (4.28). These terms are the only terms that will change depending on at which linearization point, defined by x_0 and v_0, the system is linearized. All entries in the system matrices that are a function of the linearization point are marked in red, while the black terms are invariant with respect to the linearization point. If further the linearization points are restricted to lie on the stationary operating curve of the turbine, the linear system matrices only depend on the free stream wind speed as in stationary conditions all state values are functions of this wind speed.

If, instead of the full output vector, only the measurable outputs are to be considered, the linearized output equations becomes

$$z(t) = \mathbf{C}_m x(t) + \mathbf{F}_m d(t) \tag{4.33}$$

with $\mathbf{C}_m$ and $\mathbf{F}_m$ determined by appropriately reducing $\mathbf{C}$ and $\mathbf{F}$.

4.5.1. Discretization

The model that was derived and linearized in this section is in continuous time. For the purpose of implementing the MPC controller, it is preferable to use a discretized version of the model equations. Therefore, the linearized model equations are discretized using the zero order hold method [17]:

$$\begin{aligned} \begin{bmatrix} \mathbf{A} & [\mathbf{B} \quad \mathbf{E}] \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{disc} &= \exp\left(t_s \cdot \begin{bmatrix} \mathbf{A} & [\mathbf{B} \quad \mathbf{E}] \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{cont} \right) \\ \mathbf{C}_{disc} &= \mathbf{C}_{cont} \\ \mathbf{F}_{disc} &= \mathbf{F}_{cont} \end{aligned}$$

where t_s is the sample time. For ease of notation, the distinction between continuous and discrete formulation is omitted in the following, where it is clear from context which matrices are used.

4.6. Parameter Identification

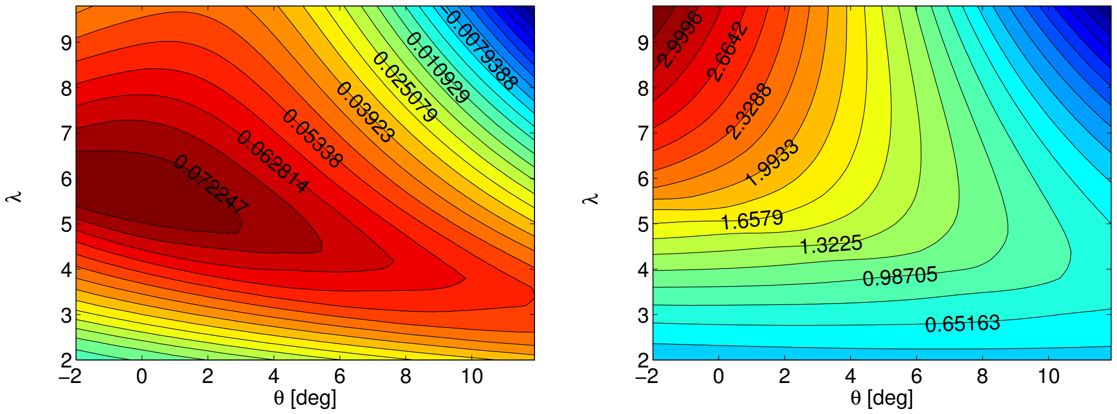
Overall, the model described in this section has 17 parameters. Besides the air density, which is an environmental parameter, they all describe the aerodynamic or mechanical properties of the wind turbine. Some parameters, like the rotor radius or inertia, are known turbine characteristics, while some others are not readily available as the aero-elastic simulation tool models that aspect differently. These parameters are determined by running special system identification simulations and fitting the model parameters appropriately. Table 4.3 lists all model parameters, its values, and the source of these values.

Parameter	Identification	Value
J	Overall Inertia of Rotor	4.05×10^7 kgm^2
R	Turbine Definition	63 m
n_G	Turbine Definition	97
$C_M(\lambda,\theta)$	from WT_PERF	Fig. 4.6
$C_T(\lambda,\theta)$	from WT_PERF	Fig. 4.6
$\omega_{T,0}$	Optimal fit of tower top acceleration to wind step input	0.324 Hz
ζ_T	Optimal fit of tower top acceleration to wind step input	0.0133
K_{fa}	Optimal fit of tower top acceleration to wind step input	1.72×10^{-7}
$\omega_{p,0}$	Parameter of actuator model	0.955 Hz
ζ_p	Parameter of actuator model	0.7
$\omega_{0,b}$	eigenfrequency of first flap mode	0.7 Hz
ζ_b	structural damping of first flap mode	0.0047
m_b	three times mass of individual blade	17740 kg
k_b	fitted to static deflection of blade tip with respect to change in thrust	2.2×10^{-6}
$\frac{l_g}{l_a}$	optimal fit of response to wind speed step	0.558
a_1	optimal fit of response to wind speed step	0.470
a	optimal fit of response to wind speed step	0.399

Table 4.3.: Overview of turbine model parameters

4.6.1. Aerodynamic Coefficients

The aerodynamic coefficients C_M and C_T are generated using the tool WT_PERF [9]. The data is generated on a grid of pitch angles and tip speed ratios with a step size of 0.1 deg and 0.25, respectively. The matrices of partial derivatives with respect to λ and θ are created by differentiating C_M and C_T numerically. Figure 4.6 shows the resulting aerodynamic coefficients. Here, the coefficients are shown only over a relatively small range near the optimal operating point of the rotor. For simulations, however, the coefficients are determined over a range that covers all potential operating points.



(a) Torque Coefficient

(b) Thrust Coefficient

Figure 4.6.: Torque and thrust coefficients C_M and C_T as functions of pitch angle θ and tip speed ratio λ

4.6.2. Structural Model Parameters

The structural model of the blade and tower in the simplified turbine model has only two degrees of freedom. As such, it is much simpler than the model implemented in FAST and most of its parameters cannot be derived directly from the FAST parameters. Instead, these parameters are determined using closed-loop system identification: The system, in closed-loop with a PI pitch controller, is subjected to a step change in wind speed and the responses of the generator speed and tower acceleration are observed. The simulation is performed in parallel with both FAST and the simplified model, and the goal of the system identification is to minimize the difference between the responses of the two models. See figure 4.7 for a schematic of this setup. During these simulations, the generator torque is held constant at the rated torque value and the wind speed is chosen to change from 13 m/s to 14 m/s. A wind speed slightly above rated speed is chosen for the identification, as this operating range is the most critical for the control design and it is desirable to achieve the best accuracy for the simplified model at these wind speeds. In section 4.7, the validity of the model at different wind speeds is further examined.

The actual parameter identification is performed in two steps: First, only those parameters related to the tower are identified and the blades are assumed to be stiff. In a second step, the blade parameters are identified with the tower parameters fixed from the first identification step.

In the first step, only the tower parameter $\omega_{T,0}$, ζ_T, and K_{fa} are determined. The blades are assumed to be rigid by deactivating the respective degrees-of-freedom in FAST and by using $m_b = 0$ in the simplified model. The tower parameters are determined so that the root

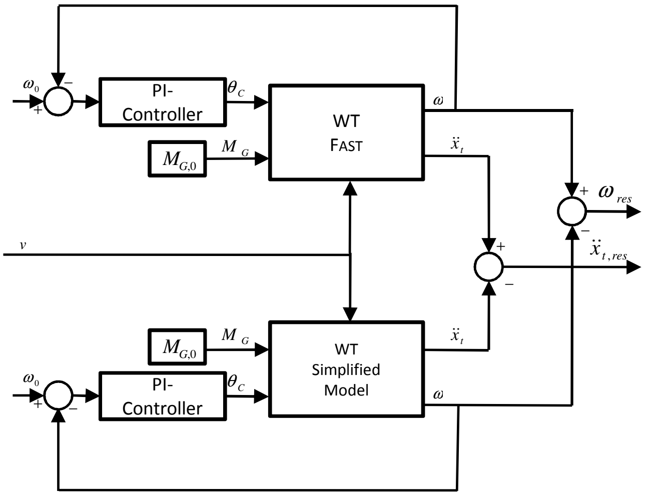
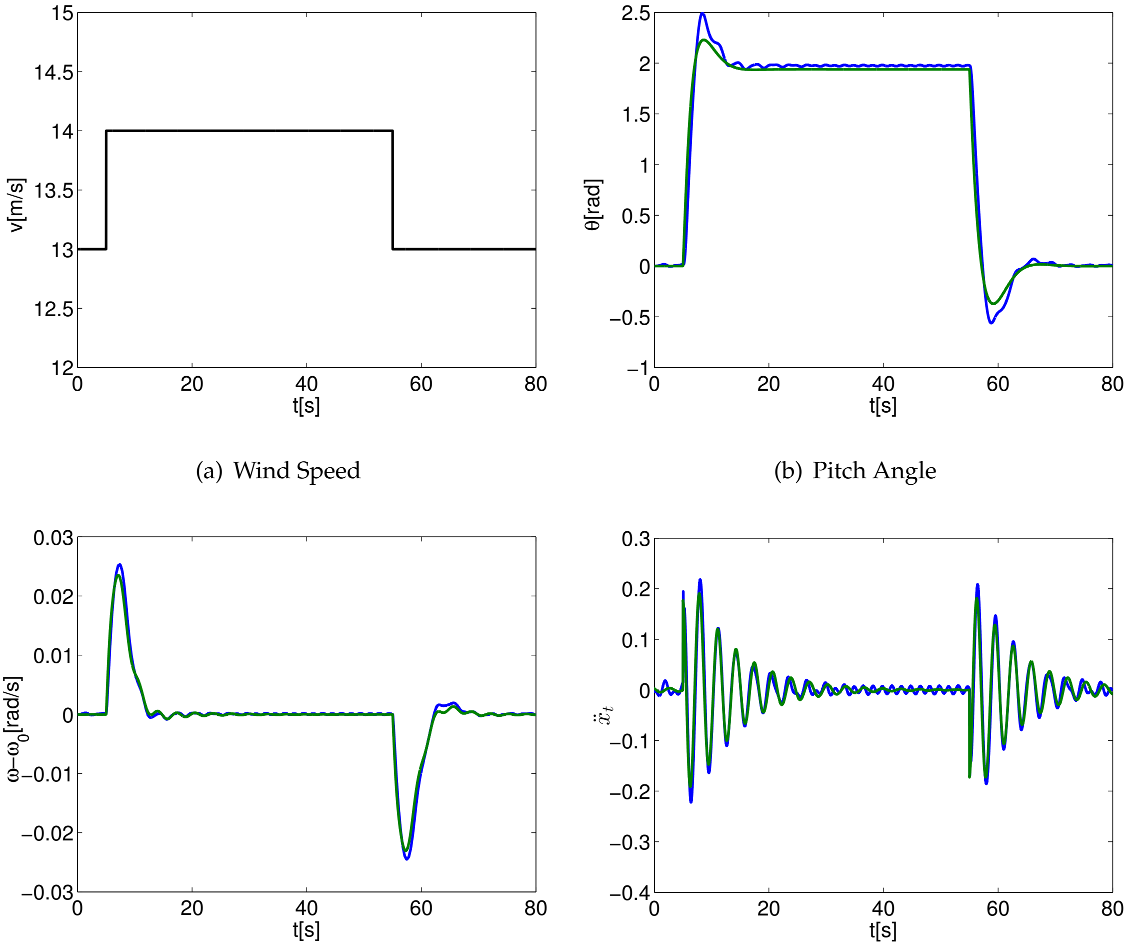


Figure 4.7.: Block diagram of setup for identification of the structural model parameters of the simplified model

mean square value of the residual of the tower top acceleration $\ddot{x}_{t,\text{res}}$ is minimized (least squares fit) where the numerical optimization is performed numerically using the Nelder-Mead Simplex algorithm [56]. See figure 4.8 for a comparison of the FAST response and the response of the fitted tower model. The identified tower eigenfrequency of $\omega_{T,0} = 0.323\,\text{Hz}$ matches the frequency of 0.324 Hz that is reported for the FAST model in [43] almost exactly.

Of the parameters related to the blade dynamics, $\omega_{0,b}$, ζ_b, m_b, and k_b are set directly based on the blade properties reported in [43]. The remaining blade model parameters are identified in a similar fashion as the tower models. While for the tower model identification no blade dynamics were assumed, the blade model identification already assumes the tower model identified in the previous step to be fixed for the simplified model and all degrees-of-freedom to be active in FAST. In order to provide sufficient excitation to the blades, the system is simulated with more aggressive controller gains for the PI-controller so that the blade motion is clearly visible in the generator speed signal. Unlike the tower top acceleration, the blade tip deflection is generally not measured directly. While an accurate tower model is required since one goal of the controller design is to reduce tower vibrations, the blade model is included in the overall model because of the impact of blade motion on the aerodynamics. Further, due to the simplifications described in section 4.2, it is difficult to compare the states of the blade model directly to states of the full order, nonlinear model in FAST. Therefore, the parameters of the blade model are identified not by fitting the response of the tip deflection, but by fitting the generator speed response and the objective



(a) Wind Speed

(b) Pitch Angle

(c) Rotor Speed

(d) Tower Top Acceleration

Figure 4.8.: Response to step changes in wind speed from simulation with FAST (blue) compared against the fitted response (green) of the simplified model for the first step of the identification of the structural model

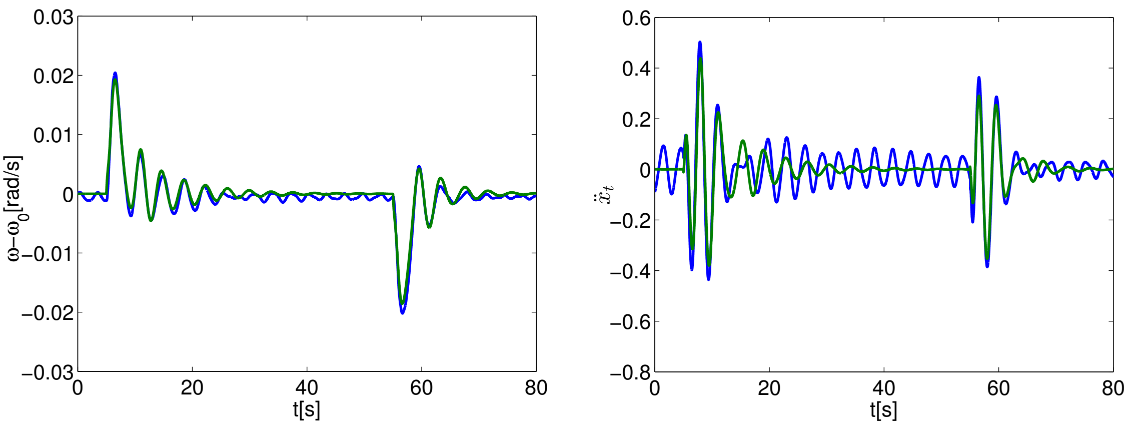
for the numerical optimization problem becomes:

$$\min_{\frac{l_g}{l_a},a_1,a} \left(\mathrm{RMS}(\ddot{x}_{t,\mathrm{res}}) \cdot \mathrm{RMS}(\omega_{\mathrm{res}})\right). \tag{4.34}$$

Figure 4.9 shows the resulting fit.

4.7. Model Validation and Analysis

This section examines the simplified turbine model that is developed in the previous section. The main focus is to show that the simplified model accurately captures the relevant



(a) Rotor Speed

(b) Tower Top Acceleration

Figure 4.9.: Response to step changes in wind speed from simulation with FAST (blue) compared against the fitted response (green) of the simplified model for the second step of the identification of the structural model

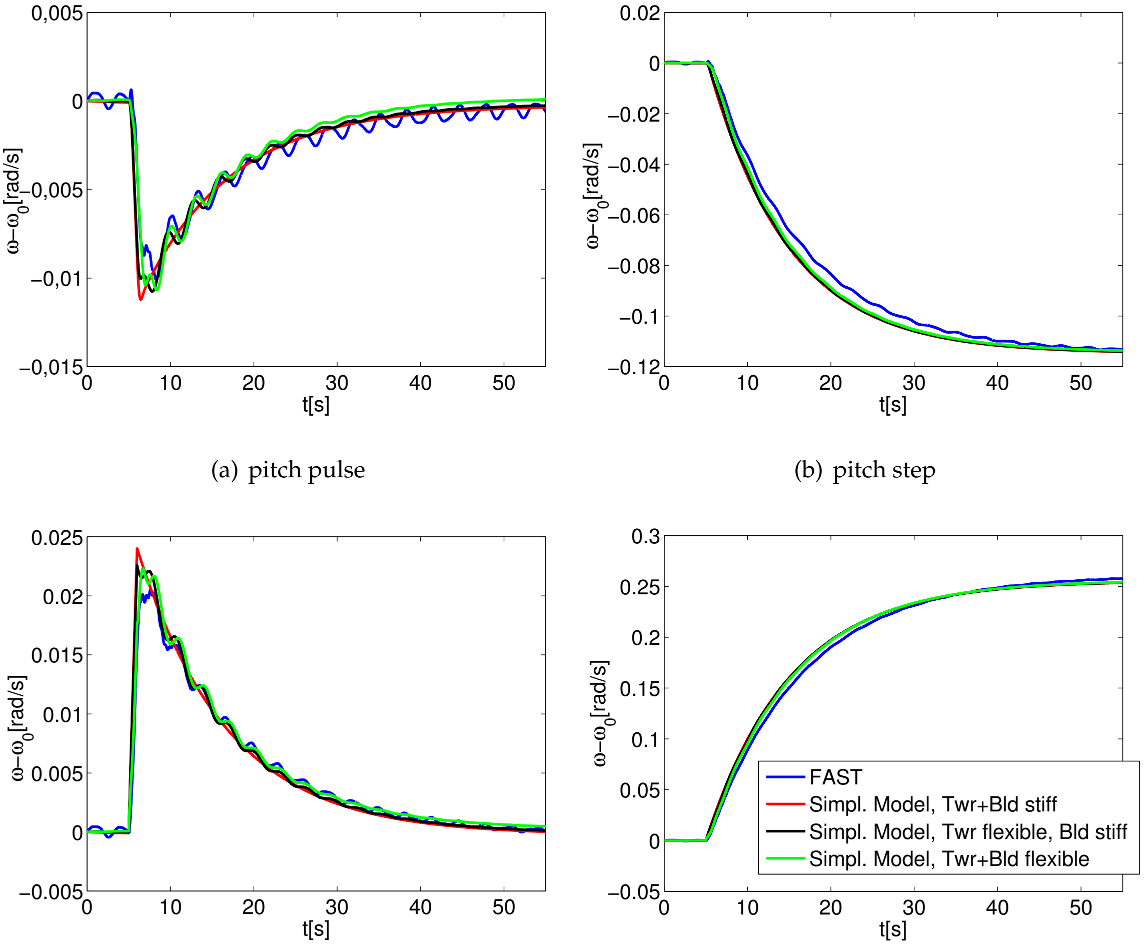
dynamics of the full, nonlinear model and to highlight key characteristics of this model that are relevant for the subsequent control design. The model is examined in both time and frequency domain.

In order to motivate the choice of turbine model, three different stages of the simplified model are considered. The first model is the most basic model that could be used to design a speed controller for a wind turbine. It consists of just the aerodynamic model (4.1) and the simple 1-DoF drive train model (4.5). In the second step, this model is augmented with the flexible tower model (4.6) while the model of the blade flap-wise motion is only added in the third step.

4.7.1. Open-Loop Comparison

At first, the simplified model is compared against the full, nonlinear model in pure open-loop. The system is subjected to both step and pulse inputs in wind speed and pitch angle and the responses of the different models are shown in figure 4.10.

It can be seen that for the steps the response essentially follows a first order behavior and that the pure drive train model captures the dynamics well. There is very little improvement of the fit between full and simplified model by adding the tower and blade models to the drive train model. For the responses to the pulse inputs, the overall dynamics are still well captured by the pure drive train model, but there are significant higher order dynamics on top of the first order behavior.



(a) pitch pulse

(b) pitch step

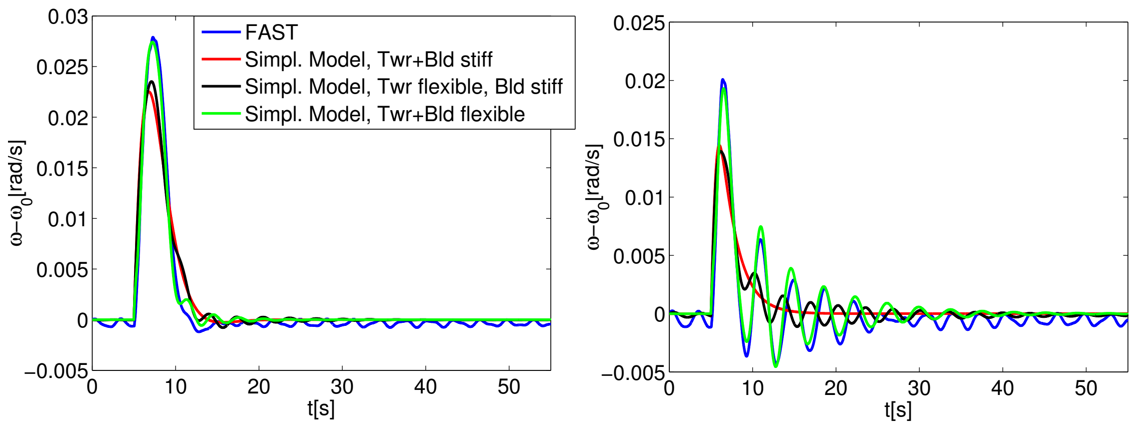
(c) wind pulse

(d) wind step

Figure 4.10.: Comparison of the full non-linear model from FAST with the simple linear model for unit step inputs and unit pulse inputs with a 1s duration.

4.7.2. Closed-Loop Comparison

The open-loop comparison shows that the match between simplified model and the full model depends on the type of excitation the system is subjected to. In order to compare the behavior for situations that are more representative for the actual turbine operation, the systems are compared in a closed-loop configuration. The system is simulated in full load with the generator torque command held constant and the pitch command controlled by a PI controller. The gains of the PI pitch controller have been chosen so that for the simple drive train model the resulting closed-loop transfer function has a unity damping. Figure 4.11 shows the response of the generator speed to a step in wind speed for two levels of controller aggressiveness.



(a) slow controller (b) aggressive controller

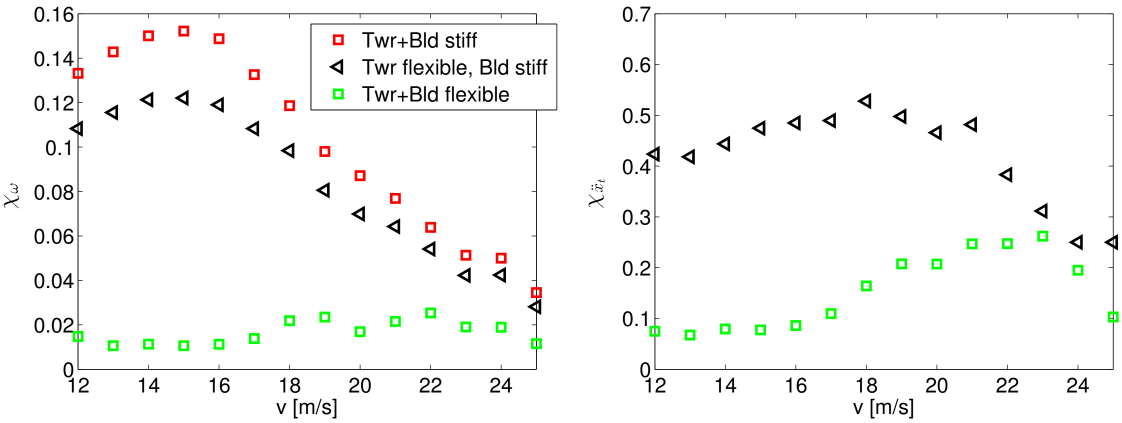
Figure 4.11.: PI control closed-loop comparison of the full non-linear model from FAST with the simple linear model for a wind step from 13 m/s to 14 m/s at different levels of controller actuation

Again, at slow pitch action the pure drive train model already captures the dynamics quite well and the larger models do not provide a large improvement. At higher frequencies, however, it can be seen that the model that includes the flexible rotor captures the dynamics significantly better than the simpler models. As a result, it can be concluded that it is possible to use the simple drive train model only to design a wind turbine controller. However, the range of controller strategies for which such a model provides an adequate match is significantly smaller than if the tower and blade interactions are included. Especially for fast controller actuation, i.e., high closed-loop eigenfrequecies, the interaction of the pitch controller with the blade and tower motion becomes more significant. The more aggressive a controller is designed to regulate the turbine speed, the higher the interaction with the tower and blade motion will be. Therefore, ignoring the tower and blade dynamics in the controller design confines the possible closed loop behavior to a limited frequency range and reduces the potential control performance.

In order to assess whether the identified model also represents the model implemented in FAST accurately at other operating points, the goodness of fit value χ is introduced. It is defined as the root mean square of the residual divided by the root mean square of the FAST-simulation over the step event:

$$\chi_\omega = \frac{\mathrm{RMS}\left(\omega_{\mathrm{res}}\right)}{\mathrm{RMS}\left(\omega_{\mathrm{FAST}} - \omega_0\right)} \qquad \chi_{\ddot{x}_t} = \frac{\mathrm{RMS}\left(\ddot{x}_{t,\mathrm{res}}\right)}{\mathrm{RMS}\left(\ddot{x}_{t,\mathrm{FAST}}\right)}. \tag{4.35}$$

A value of $\chi = 0$ therefore means that the simplified model matches the behavior of FAST perfectly, while a value of $\chi = 1$ would mean that none of the FAST behavior is represented in the simplified model.



(a) Rotor Speed

(b) Tower Top Acceleration

Figure 4.12.: Goodness of fit for responses to wind speed step inputs at different wind speeds

Figure 4.12 shows the values of χ for responses to a 1 m/s step change in wind speed starting at wind speeds between 12 m/s and 25 m/s. Several conclusions can be drawn from this figure:

a. As was already visible in figure 4.11 b), the addition of the structural model to the simple aerodynamic and rotor model improves the fit of the *rotor speed* significantly.

b. The necessity of including the structural model in order to provide a good fit of the rotor speed decreases with wind speed. This is due to the reduced excitation of the structural states at high wind speeds because the higher pitch angles cause the turbine to run at lower thrust levels.

c. If the model that considers the flexible tower and blade is used, the goodness of the fit for the rotor speed is slightly worse at wind speeds above approximately 18 m/s compared to the fit at wind speeds close to rated. This is most likely caused by the higher stiffness of the blades in the out-of-plane orientation at high wind speeds due to running at increased pitch angles. However, the decrease is only small and the overall fit remains very good with only a few percent mismatch even at high wind speeds.

d. For the tower top acceleration, the match of the simplified model is generally lower than for the rotor speed. Overall the majority of the behavior is still captured.

e. Similar to the rotor speed fit, the agreement of the tower top accelerations between FAST and the simplified model that includes flexible blades, decreases with increasing wind speeds which, again, is due to ignoring the changes in the blade dynamics with increasing pitch angles.

f. Including the flexible rotor model leads to better capturing of the system dynamics, also for the tower top acceleration.

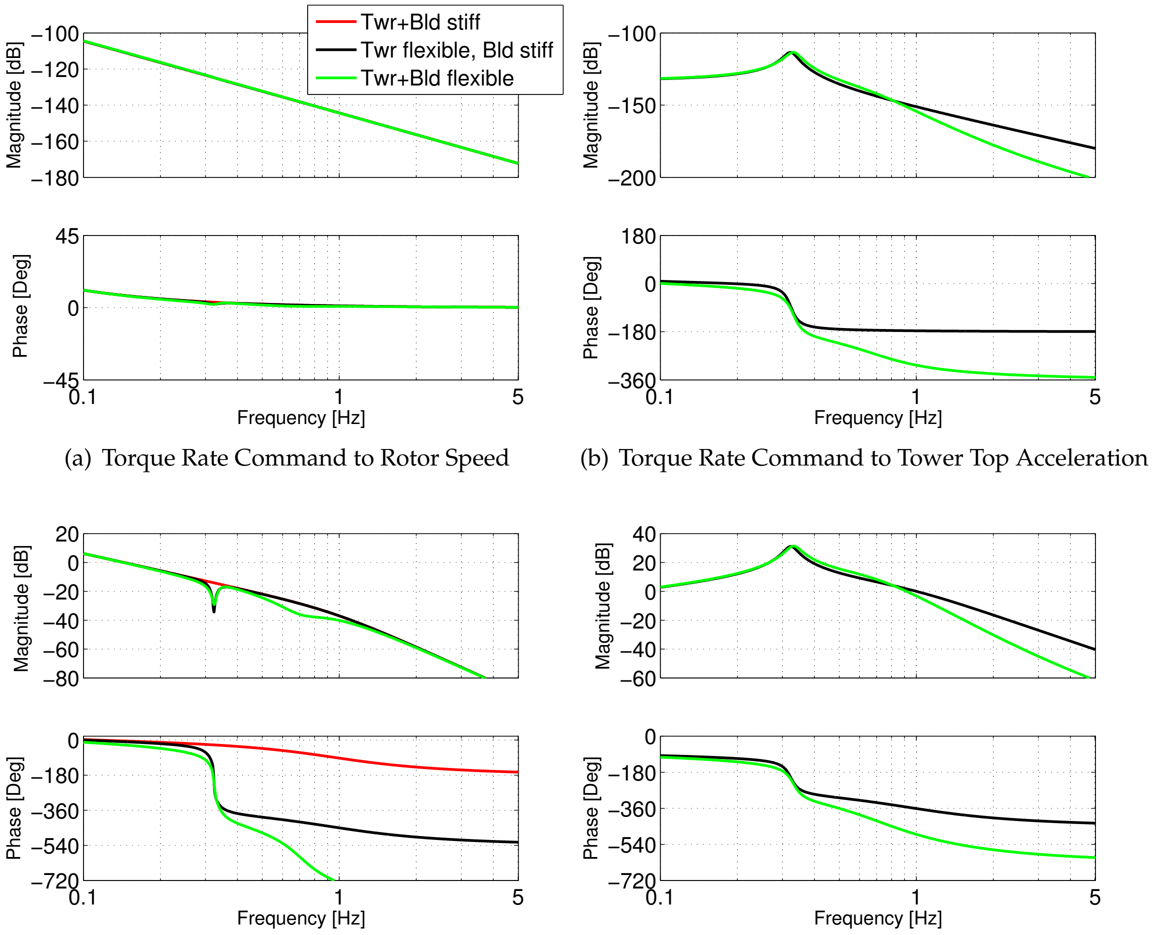
It should be noted that in the simulation of the non-linear model, many effects that lead to harmonic disturbances, like blade misalignment or mass offsets and most importantly the aerodynamic effect of the tower on the passing blade, have been omitted. It remains to be investigated whether including these effects in a model for control design improves the controller performance.

4.7.3. Frequency Domain Characteristics

In order to evaluate the simplified model in the frequency domain, the model is linearized according to section 4.5 at different operating points of the turbine each uniquely defined by the wind speed.

Figure 4.13 compares the resulting bode diagrams from the different models for the key input/output pairs. Again, the analysis is performed for the intermediate stages of the model development as well.

The torque rate command to generator speed transfer function Figure 4.13 a) shows an almost exactly linear magnitude drop with 40 dB per decade, as would be expected from the combination of the integration of the torque command (4.20) and the principal equation of motion for the rotor (4.5). The addition of the tower and blade models has only a minor effect. Further, from Figure 4.13 b) it becomes apparent that only very little excitation to the tower motion is caused by the generator torque commands even though they are coupled. Figure 4.13 c) shows a significant reduction in the magnitude of the pitch to rotor speed transfer function at the tower eigenfrequency. At this frequency, the interaction between the rotor speed and tower motion becomes so strong that any pitch controller designed to regulate rotor speed essentially has a "blind spot" because the impact of pitching the blades on rotor speed is small while at the same time the excitation of the tower motion as seen in Figure 4.13 d) is at its maximum. Corresponding to this "blind spot" is a large drop of phase at the tower eigenfrequency in the pitch to rotor speed transfer function. If no special precautions are taken, it is therefore usually inferred that the control performance is limited to that frequency [64, 63]. Although including the blade model does not change the amplitude plot at frequencies below the tower frequency, it adds an additional phase



(a) Torque Rate Command to Rotor Speed

(b) Torque Rate Command to Tower Top Acceleration

(c) Pitch to Rotor Speed

(d) Pitch to Tower Top Acceleration

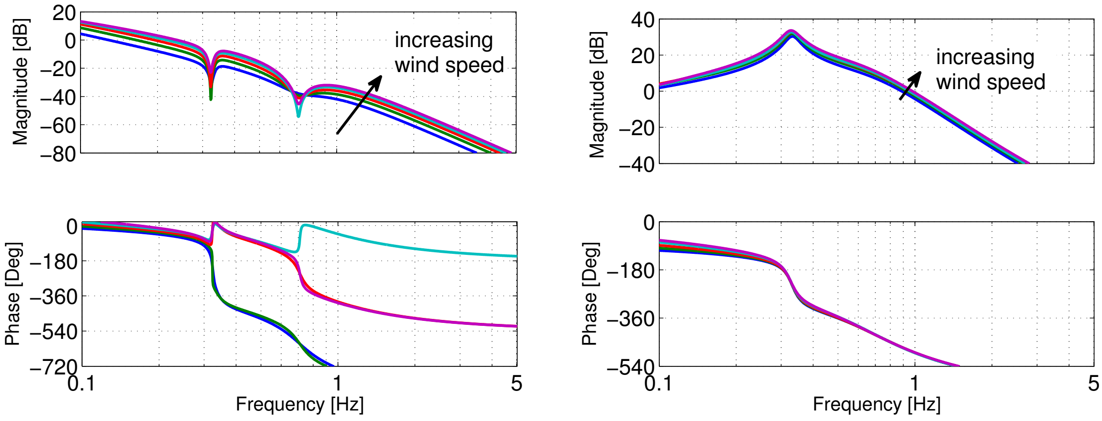
Figure 4.13.: Bode plots for the transfer functions from pitch command to rotor speed for the different models linearized at $v = 14\,\mathrm{m/s}$

loss in the frequency range below 0.2 Hz which is where most of speed control takes place[1]. The analysis of the turbine model in the frequency domain confirms the observation made in the time domain that including the tower/blade model is not only crucial for accurately capturing tower dynamics, but also for the rotor speed.

So far the plant model has only been considered at one wind speed. Since the torque coefficients M_ω, M_v, and M_θ, as well as the thrust coefficients F_ω, F_v, and F_θ, vary depending on the linearization point, the linearization will result in different transfer functions at different wind speeds. Figure 4.14 shows the bode plot of $\frac{\Omega}{s\Theta_C}(s)$ and $\frac{s^2 X_t}{s\Theta_C}(s)$ over the range of wind speeds in which the turbine operates pitch controlled. The low frequency gain in the pitch

[1] As speed control, at least in full load operation, is a disturbance rejection problem, the relevant frequency range is determined by the frequency content of the wind disturbance which is difficult to accurately specify. However, most wind models show a rapidly decreasing power spectral density of the wind above approximately 0.1 Hz [26]

to rotor speed transfer function varies by as much as 10dB. This nonlinearity needs to be accounted for in the controller design and is the main reason why the majority of wind turbine pitch controllers are either gain scheduled linear controllers or nonlinear controllers. It is further important to note that at the lower wind speeds of the full load operating region, the system is a non-minimum phase system, easily identifiable by the large phase drop at the tower and blade eigenfrequencies. With higher wind speeds, this effect becomes less pronounced and at a certain point, the system becomes minimum phase and the phase changes abruptly. Leithead and Dominguez [65] describe this effect and its implications for the control design in detail and also report similar bode plots for the pitch to speed transfer function. The pitch to tower acceleration transfer function in Figure 4.14 b) shows much less variance between the different wind speeds, which explains why most tower damper designs are generally not gain scheduled and instead use one linear controller for the entire full load operating range [4].



(a) Pitch to Rotor Speed

(b) Pitch to Tower Top Acceleration

Figure 4.14.: Bode plots for the transfer functions from pitch command to generator speed and tower top acceleration at wind speeds between 13 m/s and 25 m/s

Figure 4.15 shows the locations of the poles and zeros of the resulting transfer function from pitch command to rotor speed which, as outlined, is the most critical input/output path. At both the component eigenfrequencies of the tower $\omega_{T,0}$ and blades $\omega_{0,b}$, there is a set of zeros which moves with wind speed. At 13 m/s both the zero associated with the tower and the zero associated with the blade is in the right half plane causing the system to be non-minimum phase and leading to the large phase drop at these frequencies observed in Figure 4.13. With increasing wind speed, these zeros move towards the left half plane with first the tower zero crossing the imaginary axis in between 16 m/s and 19 m/s and then the blade zero between 23 m/s and 25 m/s. These crossings of the imaginary axis correspond to the sudden changes in phase already shown in Figure 4.14 a).

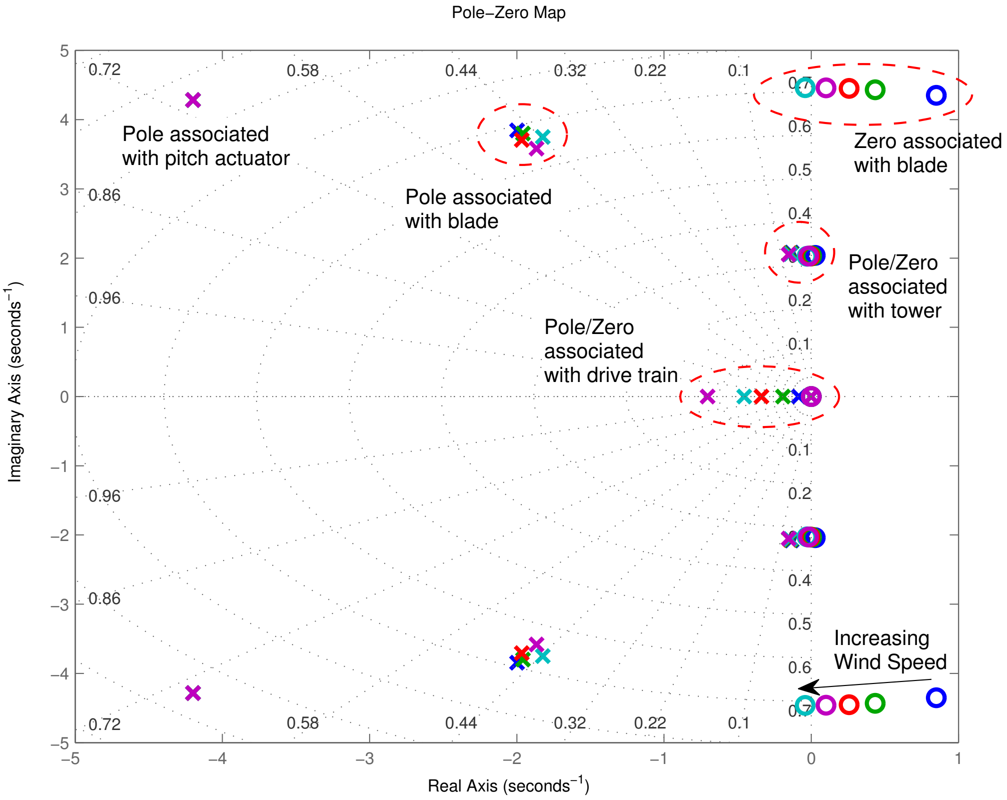


Figure 4.15.: Poles and zeros for the pitch to rotor speed transfer function at wind speeds between 13 m/s and 25 m/s

4.8. Modeling Summary

In this section, a simplified model of the wind turbine has been developed using first principle modeling techniques. The developed model is used as the basis for both the wind speed and state estimator and the Model Predictive Controller that are introduced in the next chapter. The model consists of a nonlinear aerodynamic model coupled with linear structural and actuator models. Using first principle modeling allows tailoring of the model fidelity, expressed chiefly in the chosen number of states, specifically to the control problem at hand: The design of a MIMO controller for speed and power regulation and tower damping. This will generally lead to smaller models than those derived from the automatic linearization routines of the aero-elastic tools which can be essential especially for MPC applications.

The analysis of the model in the time and frequency domain showed that adding a structural model of the blade and tower motion to the simple drive train model, which is the simplest model that can be used for the design of a pitch controller (e.g., [8, 33, 43, 54]), improves the match between the full model as represented by the aero-elastic simulation tool and the simple model for control design significantly. It is exactly this interaction between the aerodynamics and the structural dynamics that makes this control problem challenging by introducing right hand plane zeros and the associated control bandwidth restrictions. Moreover, this behavior varies significantly with wind speed and it is therefore crucial for any controller operating near those limits to accurately know and account for these differences caused by the nonlinearity of the plant model.

5. Controller Design

This chapter describes how first the simplified wind turbine model that was discussed in the previous chapter is used to design a wind speed and turbine state estimator and provides details on its performance. It then details how a continuously linearized Model Predictive Controller that makes use of this information can be designed and introduces a method to robustify the state constraints against unmeasured disturbances.

5.1. Extended Kalman Filter

The MPC is designed as a full state controller, i.e., the controller requires the current value of all model states x to compute the controller output u. Not all states of the turbine model introduced in the previous section are measurable. Therefore, the state vector needs to be estimated based on the available measurements using an estimator. Further, as described previously, on most wind turbines it is not possible to measure the wind speed accurately enough in order to use that signal in the turbine controller and it needs to be estimated as well. In a prior study [50], wind speed estimation and state estimation were performed in two separate estimators. However, since both estimation processes essentially require the same model, sample times, and need to handle the same nonlinearity caused by the turbine aerodynamics, they are combined into a single estimator here. This section first describes the generic Kalman Filter process that is used for the estimator design, then details how the estimator is designed based on the system model derived in the previous section, and finally examines its performance.

The wind speed and state estimator is designed as a standard Extended Kalman Filter [116]. An EKF uses a stochastic system model

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}) + q_{k-1} \\ z_k &= h(x_k) + p_k \end{aligned} \tag{5.1}$$

to reconstruct the state vector x_k based on the measurements z_k where the random variables q_k and p_k represent the noise on the system and outputs respectively. Assuming that q_k and p_k are linear additive disturbances as in equation (5.1), is an assumption that is commonly made but that would not be required. Here, only linear disturbances are considered for the description of the EKF. Its general idea is to linearize the nonlinear system model (5.1) at each time step about the current estimated state vector $\hat{x}_k$. The estimator is then designed to minimize the trace of the covariance of the estimation error

$$\mathbf{P}_k = E(e_k e_k^T) \qquad \text{with: } e_k = \tilde{x}_k - x_k \tag{5.2}$$

under the assumption that q_k and p_k are Gaussian white noise with covariances $\mathbf{Q}$ and $\mathbf{R}$ respectively.

The estimator that results from solving this minimization problem can be described in terms of two major steps that are performed at each time step: A time update where the system behavior is predicted ahead based on the current state estimate and a measurement update where the prediction is corrected by comparing the predicted system outputs with the actual measurements.

Without any further derivation, the resulting equations for both steps are stated here [116]:

$$\begin{aligned}
&\text{Time update:} \\
\hat{x}_k^- &= f(\hat{x}_{k-1}, u_{k-1}) \\
\mathbf{P}_k^- &= \mathbf{A}_{o,k}\mathbf{P}_{k-1}\mathbf{A}_{o,k}^T + \mathbf{Q}_{k-1} \qquad (5.3) \\
&\text{Measurement update:} \\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_{o,k}\mathbf{P}_k^- \mathbf{H}_{o,k}^T + \mathbf{R}_k)^{-1} \\
\hat{x}_k &= \hat{x}_k^- + \mathbf{K}_k(z_k - h(\hat{x}_k^-)) \\
\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_{o,k})\mathbf{P}_k^- \qquad (5.4) \\
&\text{with:} \\
\mathbf{A}_{o,k} &= \frac{\partial f}{\partial x}(\hat{x}_{k-1}, u_{k-1}) \\
\mathbf{H}_{o,k} &= \frac{\partial h}{\partial x}(\hat{x}_k). \qquad (5.5)
\end{aligned}$$

5.1.1. Disturbance Estimation

As stated before, the EKF needs to serve two purposes: It estimates the state and disturbance information that is not measurable. In order to facilitate the disturbance estimation, the equations of the wind turbine model are rewritten to include the wind speed as a state in the model. A state vector for the process to be observed x_a is defined as the state vector of the original model augmented with a disturbance state:

$$x_a = \begin{pmatrix} x \\ v \end{pmatrix} \tag{5.6}$$

The wind speed is now not treated as an input to the model, but is instead assumed to be driven by the system noise process $\dot{q}_{10}$:

$$\dot{x}_{10} = \dot{q}_{10} \tag{5.7}$$

The EKF system model (5.1) and its associated system noise q_k is defined in a discrete formulation while the wind turbine system model (4.24) was derived in a continuous formulation in the previous chapter. Therefore, the augmented system model under the assumption of linear additive system q_k noise is stated here in continuous time but it is understood that the properties of the system noise q_k are specified with respect to the discrete formulation. Using the augmented state vector, assuming only the measurable outputs of (4.26), and under the assumption of linear additive system q and measurement noise p the process to be estimated becomes:

$$\begin{aligned}
\dot{x}_1 &= \frac{1}{J}\left(M_A(x_1, x_2, x_7, x_8, x_{10}) - n_G x_9\right) + \dot{q}_1 \\
\dot{x}_2 &= x_3 + \dot{q}_2 \\
\dot{x}_3 &= x_4 + \dot{q}_3 \\
\dot{x}_4 &= -\omega_{p,0}^2 x_3 - 2\omega_{p,0}\zeta_p x_4 + \omega_{p,0}^2 \dot{\theta}_C + \dot{q}_4 \\
\dot{x}_5 &= x_7 + \dot{q}_5 \\
\dot{x}_6 &= x_8 + \dot{q}_6 \\
\begin{pmatrix} \dot{x}_7 \\ \dot{x}_8 \end{pmatrix} &= \mathbf{M}^{-1}\left(\mathbf{F}F_A(x_1, x_2, x_7, x_8, x_{10}) - \mathbf{K}\begin{pmatrix} x_5 \\ x_6 \end{pmatrix} - \mathbf{D}\begin{pmatrix} x_7 \\ x_8 \end{pmatrix}\right) + \begin{pmatrix} \dot{q}_7 \\ \dot{q}_8 \end{pmatrix} \\
\dot{x}_9 &= \dot{M}_{G,C} + \dot{q}_9 \\
\dot{x}_{10} &= \dot{q}_{10} \\
z_1 &= x_1 + p_1 \\
z_2 &= x_1 n_G x_9 + p_2 \\
z_3 &= x_2 + p_3 \\
z_4 &= x_9 + p_4 \\
z_7 &= \dot{x}_7 + p_5
\end{aligned} \tag{5.8}$$

The EKF estimation process requires linearization of the system and output equations for each time step about the current estimated state vector. The corresponding matrices $\mathbf{A}_{o,k}$

and $\mathbf{H}_{o,k}$ follow directly from the process model (5.8) similar to (4.30):

$$\mathbf{A}_o = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_a} = \begin{bmatrix} \mathbf{A} & \mathbf{E} \\ \mathbf{0} & 0 \end{bmatrix}$$

$$\mathbf{H}_o = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_a} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{C} & \mathbf{F} \\ \mathbf{0} & 0 \end{bmatrix} \tag{5.9}$$

Again, the system model is stated here in the continuous form but and the process needs to be discretized using (4.34) for an algorithmic implementation according to (5.3) and (5.4).

Choice of Noise Covariances With the process model for the estimation defined, only the assumed covariances **R** and **Q** remain to be set. The real process noise can practically not be determined. The measurement noise levels could be specified in terms of the accuracy of the used sensors. However, even those do not have the assumed white noise characteristic. In the end, the values of **R** and **Q** are often not corresponding to the real noise levels of the process, but are used as knobs to tune the estimator performance according to specific performance requirements. This tuning generally involves a trade-off between removing the measurement noise from the estimated signals via setting **R** to large values and limiting the time lag in the estimation via using a large **Q**. In the wind turbine context, this trade-off has to be mainly evaluated in terms of the wind speed estimation as the process uncertainty due to the unknown wind speeds is several orders of magnitude larger than the uncertainty in the aerodynamic and structural models and the measurement noise. As such, the main tuning knobs are the entries $q_{i,j}$ and $r_{i,j}$ in **Q** and **R** that correspond to $\dot{q}_{10}$ and p_1: Increasing $r_{1,1}$ leads to the estimator not "believing" the measured rotor speed and thus only slowly adjusting the estimated wind speed if the rotor speed changes. On the other hand, large values of $q_{10,10}$ mean that the estimator will assume that any change in rotor speed is due to a change in wind speed and the estimator will adjust its wind speed estimate quickly if the rotor speed changes. This will, however, cause unmodelled effects, such as the tower shadow, to incorrectly show up in the estimated wind speed.

Here, the covariance matrices are chosen to be diagonal matrices with entries:

$$
\begin{aligned}
&q_{1,1} = 1.0 \times 10^{-1} \quad q_{2,2} = 3.8 \times 10^{2} \quad q_{3,3} = 3.8 \times 10^{2} \quad q_{4,4} = 3.8 \times 10^{2} \quad q_{5,5} = 1.0 \times 10^{2} \\
&q_{6,6} = 1.0 \times 10^{4} \quad q_{7,7} = 1.0 \times 10^{2} \quad q_{8,8} = 1.0 \times 10^{4} \quad q_{9,9} = 0 \quad q_{10,10} = 1.4 \times 10^{4} \\
&r_{1,1} = 3.1 \quad r_{2,2} = 0 \quad r_{3,3} = 1.0 \times 10^{-3} \quad r_{4,4} = 0 \quad r_{5,5} = 3.0 \times 10^{-2}
\end{aligned} \tag{5.10}
$$

5.1.2. Performance

It was discussed in the previous section that the performance of the wind speed estimator is driven by the choice of covariance matrices. To motivate the choice of estimator tuning values, its performance is examined using FAST simulations. The full wind turbine model is simulated in FAST using a simple baseline controller, and the estimator as described in the previous sections is implemented in SIMULINK. The simulation setup and the baseline controller will be discussed in more detail also in the next chapter.

The first simulation is a series of step changes in wind speed ranging from 10 m/s to 25 m/s with a step size of 1 m/s. Figure 5.1 shows this series of steps and the resulting estimated wind speed from the EKF. On a high level, it can be seen that the estimated wind speed clearly tracks the true wind speed. At low wind speeds, in steady-state, the estimated wind speed matches the true wind speed almost exactly. Only at wind speeds of 22 m/s and higher, there is a clearly visible steady state offset with the estimator overestimating the wind speed by close to 0.5 m/s at the highest wind speed. As the steady-state estimator performance is mainly driven by the stored aerodynamic properties, this difference is likely due to slightly different aerodynamic models used for the generation of the C_M tables (in WT_PERF) and for the simulation (AERODYN) at low tip speed ratios or high pitch angles.

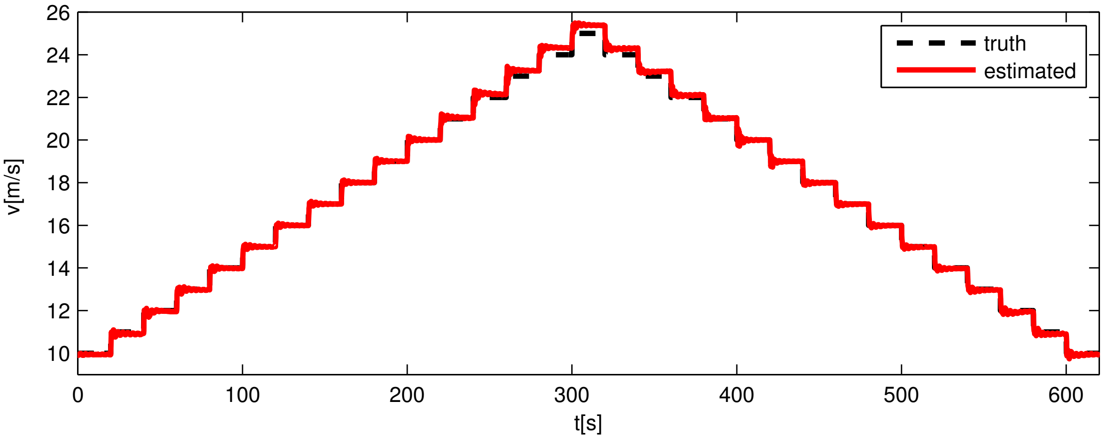


Figure 5.1.: True and estimated wind speed a series of wind speed steps with a step size of 1 m/s

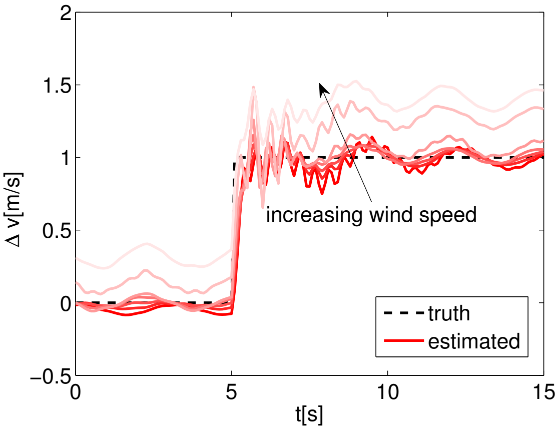


Figure 5.2.: Estimator tracking performance for wind speed steps with a step size of 1 m/s starting between 12 m/s (dark red) and 24 m/s (light red)

In this figure, it can also be observed that there is some time delay and overshooting in the estimated wind. To examine these tracking dynamics in more detail, figure 5.2 shows a zoom at the step responses where several steps are co-plotted by removing the constant component of the wind speed. The estimator tracks the true wind speed with an approximate time constant of 0.5 s and an overshoot of up to 20 %. Generally, this tracking performance can be influenced by the choice of assumed estimator noise covariances. For example, it is easily possible to tune the estimator so that a step change in free stream wind speed is tracked without any overshoot by using a lower value for the assumed model noise $q_{10,10}$ associated with the wind speed state. However, such a tuning would also lead to a significant increase of the tracking time constant. It was found that the MPC presented in the next section can easily handle some overshoot or a slightly oscillatory estimator response but would be heavily affected by a sluggish tracking performance.

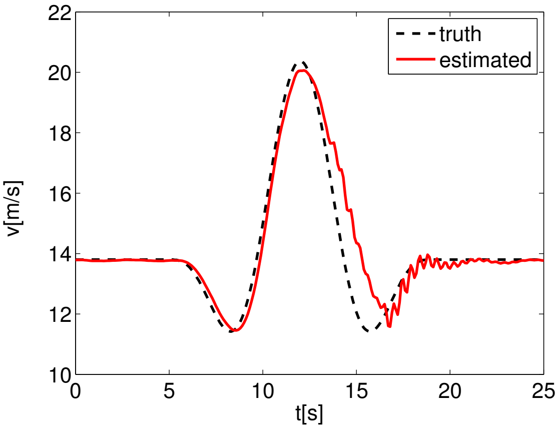


Figure 5.3.: Estimator tracking performance during an EOG50 gust starting at 13.8 m/s

Except for the steady state offsets, the tracking behavior is very similar between wind speeds showing that the estimator correctly accounts for the plant nonlinearity. However, under more realistic conditions, the wind speed never changes in steps. To assess the tracking behavior in conditions that are relevant for the wind turbine design, figure 5.3 shows the tracking performance during an extreme operating gust with a return period of 50 years, which is representative of the most extreme wind speed changes a wind turbine will encounter in power producing operation. The shape of the estimated wind speed is similar to the actual "mexican hat" gust shape, especially during the early part of the gust. Only starting at around $t = 14\,\mathrm{s}$ is there a notably bigger difference and more oscillatory response. This is likely caused by the excitation of unmodelled structural states by the strong gust. Similar to the step responses, for the early portion of the gust, there is a tracking time constant of around 0.5 s and a slightly reduced amplitude. Generally, the difference between the true and estimated wind speed never exceeds 2 m/s and is much smaller for most of the gust. The implications of the estimator mismatch on the ability of the MPC to correctly handle state constraints is discussed at length in section 5.3.

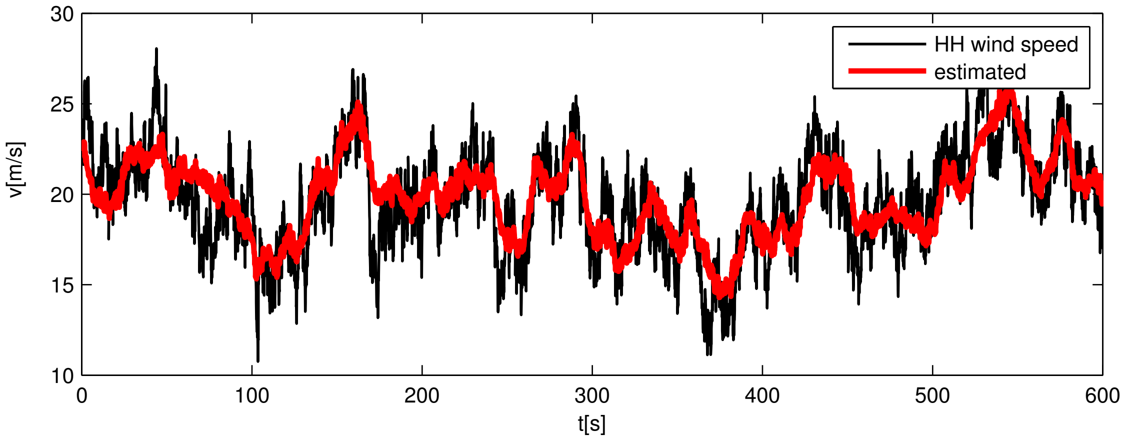


Figure 5.4.: Estimator tracking performance with respect to the hub height (HH) wind speed during turbulent wind conditions with a mean wind speed of 20 m/s and a turbulence intensity of 15%

Finally, figure 5.4 shows the result of the wind speed estimation under turbulent conditions for a 10 minute time series with a mean wind speed of 20 m/s and a turbulence intensity of 15%. Similar to the deterministic simulations, much of the high frequency content is removed by the wind speed estimator. In the turbulent case, not all of this is due to the low-pass type behaviour of the filter itself. In turbulent conditions, the wind speed is not uniform across the rotor plane so there is not one single wind speed to compare the estimator against. The estimator will estimate the rotor area effective wind speed, i.e., the uniform wind speed that would have the same effect on the rotor speed as the current turbulent wind field, and it is clear that due to this averging effect the effective wind speed will be much smoother than the single wind speed the estimator is compared to here. Nevertheless, it can be seen that the estimator correctly tracks the lower frequencxy changes in the wind speed.

5.2. Model Predictive Controller

At every time step, the controller calculates a sequence of future controller outputs $u_{k|L}$ by solving an optimization problem of which only the first controller output is then applied to the plant $u(k) = u_k$. For simplicity, it is assumed that no time delay appears when computing u_k. Otherwise, u_{k+1} would be the first variable determined in the optimization step. The objective function for the minimization problem is the quadratic sum of the controller outputs $u_{k|L}$ and deviations of the plant outputs $y_{k+1|L}$ from a given reference trajectory $r_{k+1|L}$ over the prediction horizon L. The sums are weighted using the matrices $\mathbf{Q}$ and $\mathbf{R}$:

$$\begin{aligned} \min\nolimits_{u_{k|L}} \quad J_k =& (y_{k+1|L} - r_{k+1|L})^T \mathbf{Q} (y_{k+1|L} - r_{k+1|L}) \\ &+ u_{k|L}^T \mathbf{R} u_{k|L} + V_f(x_{L+1}) \end{aligned} \tag{5.11}$$

The term V_f is an additional cost penalizing the final state. It should be noted that, because $u = (\dot{M}_{G,C} \quad \dot{\theta}_C)^T$ contains the commanded actuator *rates*, this choice of objective function will lead to $u = (0 \quad 0)^T$ in steady state even if $r \neq 0$. The optimization problem (5.11) is subject to the system dynamics

$$\begin{aligned} \dot{x} =& f(x, u, v) \\ y =& h(x, u, v), \end{aligned} \tag{5.12}$$

bounds on the controller commands

$$u_{k|L}^{\min} \leq u_{k|L} \leq u_{k|L}^{\max}, \tag{5.13}$$

and state and terminal constraints

$$\begin{aligned} \mathbf{A}_c x_{k+1|L} \leq b_c \\ \mathbf{A}_{tc} x_L \leq b_{tc} \end{aligned} \tag{5.14}$$

5.2.1. Tuning and Choice of Constraints

The MPC is tuned through the choice of weighting matrices $\mathbf{Q}$ and $\mathbf{R}$. For example, large values of $\mathbf{R}$ mean the controller strongly penalizes controller actuation and will only uses the actuators cautiously. On the other hand, if $\mathbf{R}$ is small compared to $\mathbf{Q}$, the controller places a higher importance on regulating the outputs than on careful use of the actuators and will act aggressively. By selecting the values for $\mathbf{Q}$ and $\mathbf{R}$, the wind turbine designer can therefore tune the controller response according to the design needs. It is not only possible to trade-off actuator expenditure against regulation performance. By changing the entries of $\mathbf{Q}$, the tracking performance of the outputs can also be traded-off against each other. For example, if $q_{1,1}$, the entry of $\mathbf{Q}$ corresponding to the rotor speed, is small compared to the tower top acceleration weight $q_{7,7}$, the controller will place more emphasis on reducing tower oscillations than on tightly tracking the generator speed. Here, the following values have been chosen:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{Q}_1 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{R}_1 \end{bmatrix}$$

$$\mathbf{Q}_1 = \begin{bmatrix} q_{1,1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & q_{7,7} \end{bmatrix} \qquad \mathbf{R}_1 = \begin{bmatrix} r_{1,1} & 0 \\ 0 & r_{2,2} \end{bmatrix}$$

with

$$q_{1,1} = 1.5 \times 10^3 \quad q_{2,2} = 7.2 \times 10^{-6} \quad q_{3,3} = 0 \quad q_{4,4} = 0 \quad q_{5,5} = 1.5 \times 10^2 \quad q_{6,6} = 0 \quad q_{7,7} = 0$$
$$r_{1,1} = 1.0 \times 10^{-6} \quad r_{2,2} = 3.0 \times 10^3 \tag{5.15}$$

The entry corresponding corresponding to the pitch rate $\dot{\theta}$, $q_{3,3}$, is set to zero because excessive pitch movement can also be penalized via the penalty on the commanded pitch rate $r_{2,2}$. A generator torque penalty $q_{4,4}$ is not required as only high torque rates but not the torque itself have a negative impact on the system and the torque rate is penalized via $r_{1,1}$. The tower top acceleration is not penalized via $q_{7,7}$ as it was found that using the acceleration penalty does not offer any additional benefit on top of the tower top velocity penalty $q_{5,5}$. Finally, the blade tip speed penalty $q_{6,6}$ is not used due to the limitations of the collective flap blade model. The effect of the choice on the control performance is further discussed in section 6.2.2.

The generic constraint formulation (5.13) and (5.14), allows for a wide range of potential constraints. Here, however, only the following constraints are used. The rates for the both generator torque actuation as well as the pitch actuation are limited:

$$\begin{aligned} |\dot{\theta}_C| &\leq 10\,\text{deg/s} \\ |\dot{M}_{G,C}| &\leq \frac{M_{G,0}}{10}\,\text{1/s}. \end{aligned} \tag{5.16}$$

Two state state constraints are used. The rotor speed is constraint to be below 10% above the rated speed

$$x_1 \leq \frac{11}{10}\omega_0 \tag{5.17}$$

Placing this hard constraint on the rotor speed has severe implications for the feasibility of the control problem that are specifically addressed in section 5.3. Secondly, the pitch angle is limited to not be below 1 deg.

$$x_2 \geq 1\,\mathrm{deg} \tag{5.18}$$

1 deg is roughly the optimal pitch angle for variable speed operation. This constraint is necessary when operating near the rated wind speed. Without it, the turbine might be pitching towards stall in case of low wind speeds.

Although the focus of the designed controller is on the full load operating region, it needs to be able to handle at least upper partial load operation as in turbulent conditions the turbine will have to operate below rated wind speed for short periods of time, even if the average wind speed is above rated. This partial load controller is implemented through simply changing the tuning weights and constraints whenever the turbine is in partial load operation. In partial load operation, the turbine is not supposed to use the pitch to control the turbine. Therefore, the pitch rate is constrained to zero:

$$|\dot{\theta}_C| = 0. \tag{5.19}$$

With the pitch not available, the turbine can no longer track the rotor speed and power output independently. Therefore, the power regulation objective is deactivated by setting the corresponding weight to zero:

$$q_{2,2} = 0. \tag{5.20}$$

It should be noted that this type of partial load controller is capable of handling both lower and upper partial load operation. The only difference between the two is that in upper partial load operation the rotor speed set-point is constant, while in lower partial load the set-point would be a function of the estimated wind speed (see section 2.1.2). However, for the purposes of evaluating the wind turbine performance near and above rated wind speed, the lower partial load controller does not need to be considered any further.

Switching Conditions The turbine switches from full load operation to partial load operation where the modified constraints (5.19) and weights (5.20) are used when the pitch angle is less than or equal to the minimum pitch angle of $\theta = 1\,\text{deg}$ and the power is below rated power. The turbine switches from partial load operation to full load if the power is higher than rated power.

5.2.2. Linearization

The state and output equations in the optimization problem, as given by equations (4.24) and (4.25), are nonlinear functions resulting in the optimization problem being nonlinear. There are several options to handle nonlinear plant models in a model predictive control setup:

- Robustness: If the linear MPC is designed with enough robustness, it is possible to use a single linear MPC for an entire operating range of the plant (e.g., full load operation). This approach was demonstrated by Henriksen for both the partial load operation and the full load operation [36].
- Scheduling: Similar to classical gain scheduling, the plant model used for setting up the MPC-problem can be obtained by linearization not only at one but at several stationary operating points. The most straight forward implementation of this strategy is to run several controllers in parallel with the output being a weighted sum of the outputs from the single controllers depending on the current operating point. This method was employed using three linearization points by Kumar and Stol [54]. If only a few linearization points are used, this method can be seen as accounting for the slow changes due to variations in mean wind speed, but neglects the fast turbulence induced changes in plant dynamics.
- Continuous Linearization: In order to avoid some of the limitations of the scheduling approach, the model and resulting MPC formulation could be updated at every or close to every time step. This would allow for also accounting for the (fast) turbulence induced variations in operating point.
- Nonlinear MPC: In both the scheduling and the continuous linearization approach, a single linear model is used for predicting the output trajectory over the prediction horizon and calculating the optimal control trajectory. In the case of nonlinear MPC, the nonlinear model is used for the output trajectory calculation and the gradient is calculated along this trajectory. As the problem is no longer quadratic and needs numerical integration of the model equations, a much higher computational burden is involved.

While a large body of literature dealing with Nonlinear Model Predictive Control (NMPC) exists, and even application to comparably fast systems seems viable (e.g., [120]), using

NMPC still increases complexity and computational burden over linear MPC significantly and it needs to be evaluated carefully whether using NMPC actually provides the performance improvement that would warrant the additional complexity. The different methods for linearization have been evaluated in a previous study [51]. The main result was that for turbulent simulations with a given mean wind speed and typical turbulence intensities using NMPC provided almost no observable benefit over continuous linearization or even linearization about the stationary operating point corresponding to the mean wind speed. However, as described in section 2.1.2, there is a significant variation in plant behavior if variations in mean wind speed are also considered. Therefore, the approach that is chosen here is a hybrid of the scheduling and the continuous linearization methods: The system is linearized at every time step. The linearization is however not performed at the current state and input vectors. Instead, the system is linearized at the stationary operating point corresponding to the current estimated wind speed. This approach avoids having to run several controllers in parallel as in the scheduling approach while still handling even fast changes in wind speed. Compared to the continuous linearization, where the linearization point is the current state vector instead of the steady state value, the approach will always have the origin in state space as its set-point which significantly simplifies the control formulation and stability assessment.

At each time step, a linear state space model

$$\begin{aligned} \delta\dot{x} &= \mathbf{A}_k\delta x + \mathbf{B}\delta u + \mathbf{E}_k\delta d \\ \delta y &= \mathbf{C}_k\delta x + \mathbf{F}_k\delta d \end{aligned} \tag{5.21}$$

is derived by linearizing the equations (4.24) and (4.25). The index k is used for the matrices **A**, **C**, **E**, and **F** to indicate that they will change with every time step. The linearization is performed about the steady state values x_{ss} corresponding to the current estimate or measurement of the free stream wind speed v:

$$x_0 = x_{ss}(v). \tag{5.22}$$

Offset free tracking is assumed for the rotor speed and electrical power. Therefore, the steady state value for the rotor speed is simply the set-point

$$x_{ss,1} = \omega_{ss} = \omega_0 \tag{5.23}$$

while the steady state generator torque is simply the corresponding rated torque value $M_{G,0}$, which can be calculated based on rated power:

$$x_{ss,9} = M_{G,0} = \frac{P_0}{\omega_0 n_G}. \tag{5.24}$$

With rated rotor speed and generator torque known, there is a pitch angle θ_0 that balances the aerodynamic torque with the generator torque. This pitch angle is obtained using a precomputed mapping Θ of steady state pitch angles as a function of corresponding wind speeds:

$$\theta_0 = \Theta(v). \tag{5.25}$$

The function Θ is calculated by numerically solving $P_0/\omega_0 = M_A(v, \omega_0, \theta)$ for θ at all relevant values of v. Fig. 5.5 shows the resultant pitch angles as a function of wind speed. With $\theta_0(v)$ known, the steady state value of the aerodynamic thrust can be calculated, which in turn is used to calculate steady-state blade and tower positions:

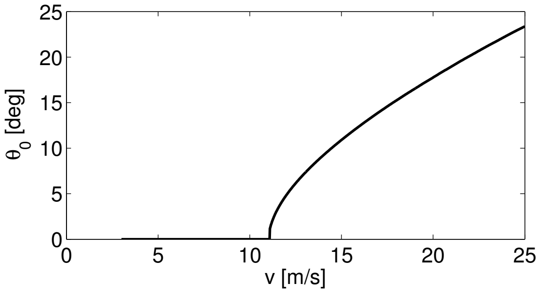


Figure 5.5.: Steady state pitch angles as a function of wind speed

$$\begin{pmatrix} x_{t,ss} \\ x_{b,ss} \end{pmatrix} = \mathbf{K}^{-1}\mathbf{F}F_A(v, \omega_0, \theta_0(v)). \tag{5.26}$$

All remaining states clearly have steady state values of zero. In summary, the steady state vector for a given wind speed is defined as:

$$x_{ss}(v) = \begin{pmatrix} \omega_0 \\ \Theta(v) \\ 0 \\ 0 \\ \mathbf{K}^{-1}\mathbf{F}F_A(v, \omega_0, \theta_0(v)) \\ 0 \\ 0 \\ \frac{P_0}{\omega_0 n_G} \end{pmatrix}. \tag{5.27}$$

For ease of notation, the difference between the current state vector and the linearization point $\delta x(t) = x(t) - x_0(v)$ is simply called x wherever it is clear from context that the deviations are used. Further, while the model has been derived in continuous form, it is implemented in a discrete state space formulation, and the system matrices **A**, **B**, **E**, and **F** need to be discretized accordingly. Therefore, in the following, the system model is simply

$$\begin{aligned} x_{k+1} &= \mathbf{A}_k x_k + \mathbf{B} u_k + \mathbf{E}_k d_k \\ y_k &= \mathbf{C}_k x_k + \mathbf{F}_k d_k. \end{aligned} \tag{5.28}$$

and it is implied that the input, state, and output vectors are the deviations from the linearization point, and that the discrete system matrices were obtained by discretizing the system model derived in this section using the zero order hold method.

5.2.3. Stability

As discussed in section 2.3.3, it is a well-known property of a *linear* MPC that stability can be guaranteed independently of the choice of prediction horizon by choosing an appropriate terminal cost $V_f(x_{L+1})$ and constraint $\mathbf{A}_{tc}$ and b_{tc} [91]. On the other hand, stability can also be achieved using a sufficiently large prediction horizon which also has been used in many applications of MPC for wind turbines. On the other hand, in order to achieve guaranteed stability using large prediction horizons, generally a prediction horizon at least in the order of the settling time of the plant, is required [71]. Especially if the tower dynamics are considered, as in the present study, this is a severe limitation since the natural frequency of the tower is usually below 0.5 Hz and, combined with the low damping, this would lead to prediction horizons that are above the horizons typically considered.

As stated by Mayne and Rawlings [91], stability can be achieved if the terminal cost is chosen to reflect the infinite horizon cost of bringing the system to rest from the terminal state of the prediction horizon using a terminal controller, and the terminal constraint set is chosen appropriately. The terminal constraint set is a control invariant set, meaning that for no state vector in this set the constraints are violated while bringing the system to rest with the terminal controller.

It should be noted that choosing the terminal cost and constraint like this will only lead to stability for the linear system. The system considered in this study, however, is nonlinear, so that generally stability of the closed loop system will only be ensured in a region around the linearization point. In order to ensure stability over a wider range of linearization points, the variations in plant and controller behavior need to be considered explicitly, for example using parametric uncertainties. This, however, is not seen as a severe restriction, as the nonlinearity, while not negligible when considering the full operating envelope, is fairly ”tame” in the proximity of the operating points and thus leads to large regions of attraction.

A linear quadratic regulator (LQR) is chosen as the terminal controller. The infinite horizon cost associated with this controller is $\frac{1}{2}x_{L+1}^T\mathbf{P}_k x_{L+1}$ where $\mathbf{P}_k$ can be calculated by solving the discrete algebraic Riccati equation

$$\begin{aligned}\mathbf{P}_k = &- \mathbf{A}_k^T\mathbf{P}_k\mathbf{B}_k(\mathbf{B}_k^T\mathbf{P}\mathbf{B}_k + \mathbf{R})^{-1}(\mathbf{A}_k^T\mathbf{P}_k\mathbf{B}_k)^T \\ &+ \mathbf{A}_k^T\mathbf{P}_k\mathbf{A}_k + \mathbf{Q}_{k,x} \end{aligned} \tag{5.29}$$

with $\mathbf{Q}_{k,x}$ defined so that the cost placed on the states is equivalent to the cost placed on the system outputs within the prediction horizon:

$$\mathbf{Q}_{k,x} = \mathbf{C}^T\mathbf{Q}_1\mathbf{C}. \tag{5.30}$$

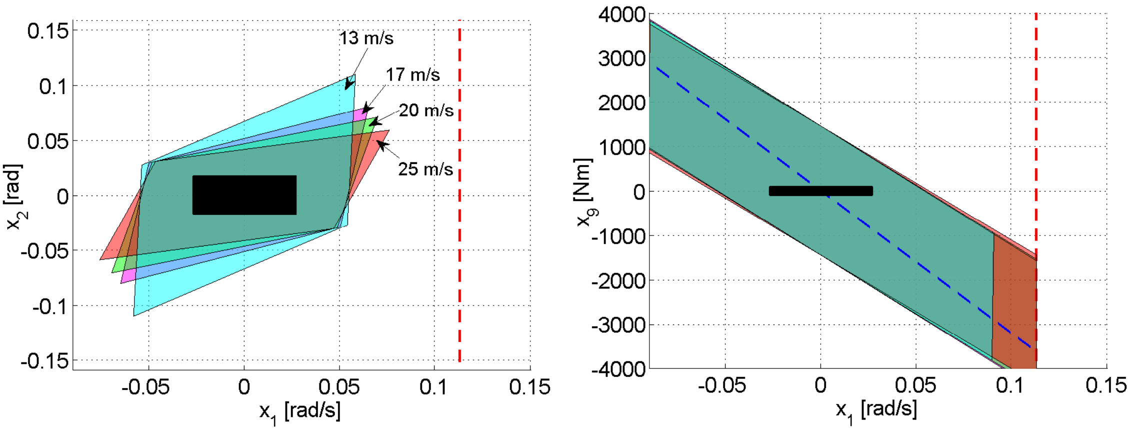
As the system matrices $\mathbf{A}_k$ and $\mathbf{B}_K$ change over time, the Riccati equation (5.29) is solved for $\mathbf{P}_k$ at every time step and the terminal cost is set to:

$$V_f(x_{L+1}) = \frac{1}{2}x_{L+1}^T\mathbf{P}_k x_{L+1}. \tag{5.31}$$

The terminal constraint is chosen to be a subset of the maximum control invariant set. A subset chosen as the maximum set is defined by a large number of inequalities which make the implementation impractical. Also, due to the nonlinearity of the plant model, the maximum admissible set varies with operating point and the terminal constraints would need to be updated at every time step. Instead, the terminal constraint set is chosen to be a hyper cuboid in the 9-dimensional state space which is a subset of the maximum output admissible sets for all wind speeds. As an example, figure 5.6 shows the maximum output admissible sets at four different wind speeds and the chosen terminal constraint set in the x_1-x_2 (speed vs. pitch angle) and the x_1-x_9 (speed vs. generator torque) slices of the state space where the maximum control invariant set is calculated using the algorithm from Gilbert and Tan [28] which is reproduced in appendix A.

For the calculation of the terminal constraint, the constraint placed on the pitch state is ignored. This is done because the constraint is not a true constraint where a violation would lead to some form of system failure. Instead, the pitch angle constraint is used to help the transition between full load and partial load operation. Specifically, violation of the minimum pitch angle constraint is not a problem for the system, but merely one necessary condition for switching from full load to partial load operation. Therefore, only the maximum rotor speed and the bounds on the commanded torque and pitch rates are considered here.

Looking at the x_1-x_2 slice, it can be seen that at low pitch angles lower rotor speed are required. This is intuitively understandable considering that the combination of low pitch



(a) x_1-x_2 slice

(b) x_1-x_9 slice

Figure 5.6.: x_1-x_2 and x_1-x_9 slices of the selected terminal constraint (black) and maximum admissible sets (colored) at four different wind speeds. The dashed blue line marks the rotor speed state constraint while the dashed green line in the x_1-x_9 slice marks the constant power line.

angle and high rotor speed clearly leads to the highest overspeed risk. The maximum admissible set has a fairly similar shape between the different shown wind speeds. However, at high wind speeds, slightly higher rotor speeds can be allowed at non-negative pitch angles. This is due to the higher sensitivity of the aerodynamic torque with respect to pitch changes that allows the controller to bring the speed down faster with the same maximum pitch rate. At the same time, because of this higher sensitivity, lower pitch angles also pose more of a challenge to the system. Subsequently, the maximum admissible set extends further into the negative pitch angles at lower wind speeds.

Another interesting observation can be made when looking at the x_1-x_9 slice: The maximum admissible set seems to be almost centered around the constant power (dashed green) line. This is caused by the very restrictive torque rate bound. If the power is too far above (high rotor speed and high generator torque) or below (low rotor speed and low generator torque), then the terminal controller will aggressively use the torque to maintain rated power and thereby violate the torque rate bound. Unlike in the x_1-x_2 slice, there are only small differences between the different wind speeds showing the power control problem is only remotely affected by the non-linear aerodynamics.

Finally, the selected terminal constraint appears to be much smaller than the shown maximum admissible sets. However, for each of those plots all seven other states are assumed to be zero. The potential variations in the other states will also need to be considered, i.e., the maximum admissible set in the x_1-x_2 slice will be significantly smaller if all potential variations in the other states are covered as well. For the same reason, a terminal constraint needs to be defined for all nine states even if a state constraint is only used for the rotor speed. For example, a very large tower terminal velocity can easily result in controller actuation above the rate limits and high rotor speeds. Therefore, it needs to be ensured that the rotor velocity is within a certain bound at the terminal state even if no explicit state constraint is used on that state. The chosen terminal constraint set is:

$$\begin{aligned} |x_1| &\leq \frac{30\,\mathrm{RPM}}{n_G} & |x_2| &\leq 1\,\mathrm{deg} & |x_3| &\leq 0.2\,\mathrm{deg/s} \\ |x_4| &\leq 0.1\,\mathrm{deg/s^2} & |x_5| &\leq 0.15\,\mathrm{m} & |x_6| &\leq 0.15\,\mathrm{m} \\ |x_7| &\leq 0.1\,\mathrm{m/s} & |x_8| &\leq 0.1\,\mathrm{m/s} & |x_9| &\leq 100\,\mathrm{Nm} \end{aligned} \tag{5.32}$$

It should be noted that, as the linearized system description is used, the terminal constraint set is centered around the origin of the linearized state space. The terminal constraint (5.32) thus defines the maximum deviations at the terminal state from the steady state operating point given by (5.27).

The maximum admissible set would need to be calculated at an infinite number of operating points but here the assumptions is made that, if the chosen set is a subset of the maximum set at a number of operating points (here, all integer valued wind speeds), this holds for all operating points.

5.2.4. Control Equations

In order to solve the controller optimization problem ((5.11)-(5.14)), the system dynamics constraint is eliminated from the problem formulation by expressing the outputs $y_{k+1|L}$ as a function of the controller inputs $u_{k|L}$ which are the decision variables for the optimization. For ease of implementation, a slightly different formulation than that used in section 2.3 is employed here.

The trajectory of future states and outputs can be seen as the sum of the unforced state and output trajectories $x_{f,k|L}$ and $p_{k|L}$ and the effect of the future control inputs

$$\begin{aligned} x_{k+1|L} &= x_{f,k|L} + \mathbf{B_p} u_{k|L} \\ y_{k+1|L} &= p_{k|L} + \mathbf{C_p} \mathbf{B_p} u_{k|L}. \end{aligned} \tag{5.33}$$

The unforced state and output trajectories are a function of the current state and the trajectory of future disturbance inputs

$$\begin{aligned} x_{f,k|L} &= \mathbf{A_p} x(k) + \mathbf{E_p} v_{k|L} \\ p_{k|L} &= \mathbf{C_p} x_{f,k|L} \end{aligned} \tag{5.34}$$

with the prediction matrices for both the unforced and forced responses defined as [18, 70]

$$\mathbf{A_p} = \begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_k^2 \\ \vdots \\ \mathbf{A}_k^L \end{bmatrix} \qquad \mathbf{B_p} = \begin{bmatrix} \mathbf{B}_k & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_k\mathbf{B}_k & \mathbf{B}_k & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_k^{L-1}\mathbf{B}_k & \mathbf{A}_k^{L-2}\mathbf{B}_k & \dots & \mathbf{B}_k \end{bmatrix}$$
$$\mathbf{C_p} = \begin{bmatrix} \mathbf{C}_k \\ \mathbf{C}_k \\ \vdots \\ \mathbf{C}_k \end{bmatrix} \qquad \mathbf{E_p} = \begin{bmatrix} \mathbf{E}_k & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_k\mathbf{E}_k & \mathbf{E}_k & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_k^{L-1}\mathbf{E}_k & \mathbf{A}_k^{L-2}\mathbf{E}_k & \dots & \mathbf{E}_k \end{bmatrix}. \tag{5.35}$$

Finally, the terminal state x_{L+1} also needs to be expressed as a function of the current state and future control and disturbance inputs

$$\begin{aligned} x_{L+1} &= x_{f,L+1} + \mathbf{B_{pL}} u_{k|L} \\ x_{f,L+1} &= \mathbf{A_{pL}} x(k) + \mathbf{E_{pL}} v_{k|L} \end{aligned} \tag{5.36}$$

where the matrices $\mathbf{A_{pL}}$, $\mathbf{B_{pL}}$, and $\mathbf{E_{pL}}$ are defined accordingly

$$\begin{aligned} \mathbf{A_{pL}} &= \begin{bmatrix} \mathbf{A}^{L+1} \end{bmatrix} \\ \mathbf{B_{pL}} &= \begin{bmatrix} \mathbf{A}^L\mathbf{B} & \mathbf{A}^{L-1}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix} \\ \mathbf{E_{pL}} &= \begin{bmatrix} \mathbf{A}^L\mathbf{E} & \mathbf{A}^{L-1}\mathbf{E} & \dots & \mathbf{E} \end{bmatrix}. \end{aligned} \tag{5.37}$$

Substituting the prediction model, (5.33) and (5.34), and the terminal cost (5.31) in the objective function (5.11) gives:

$$\begin{aligned} J_k = u_{k|L}^T & \left(\mathbf{B_p}^T \mathbf{C_p}^T \mathbf{Q} \mathbf{C_p} \mathbf{B_p} + \mathbf{B_{pL}}^T \mathbf{P}_k \mathbf{B_{pL}} + \mathbf{R} \right) u_{k|L} \\ & + 2u_{k|L}^T \left(\mathbf{B_p}^T \mathbf{C_p}^T \mathbf{Q} p_{k|L} + \mathbf{B_{pL}}^T \mathbf{P}_k x_{f,L+1} \right) \\ & + p_{k|L}^T \mathbf{Q} p_{k+1|L} + x_{f,L+1}^T \mathbf{P} x_{f,L+1}. \end{aligned} \tag{5.38}$$

Similar to the objective function, the constraints are also expressed as functions of the unforced response and the control action:

$$\begin{aligned} u_{k|L}^{\min} \leq & u_{k|L} \leq u_{k|L}^{\max} \\ \mathbf{A}_c \mathbf{B_p} u_{k|L} \leq & b_c - \mathbf{A}_c x_{f,k|L} \\ \mathbf{A}_{tc} \mathbf{B_{pL}} u_{k|L} \leq & b_{tc} - \mathbf{A}_{tc} x_{f,L+1}. \end{aligned} \tag{5.39}$$

Equations (5.38) and (5.39) now define a quadratic problem (QP) in $u_{k|L}$.

5.2.5. Feedforward Control

By considering the future disturbance inputs explicitly when calculating the unforced output trajectory (5.34), the described controller includes a disturbance feedforward component in the feedback scheme. Generally, the trajectory of the future disturbance inputs is not known and only an estimate of the current effective wind speed $\hat{v}$ is available. In this case, the assumption is made that the wind speed will remain constant over the entire prediction horizon, effectively causing the terms in the prediction model (5.34) that depend on the disturbance to vanish, as the model has been linearized at the current estimated wind speed.

However, if upwind information, e.g., from LIDAR measurements, is available and used as the trajectory of future disturbance inputs $v_{k|L}$, the controller will consider the effect of future wind speed changes. In the wind turbine controls context, this type of control is often termed *preview control*.

Next to measurements, the controller can also use predictions of the future wind speeds, for example from a wind prediction model as described in section 3.1.3. In this case, this type of controller is not a true disturbance feedforward as the disturbance itself, and potentially also the predicted wind speed, is only derived from the same measurements also used for the feedback control. Nevertheless, this type of controller will improve the performance over a pure feedback controller as the model information of how disturbances act on the controlled outputs is taken into account explicitly.

Most feedforward schemes combine a feedforward controller with a feedback controller, where the feedback controller has to ″handle″ all remaining disturbances that have not been canceled by the feedforward due to modeling or measurement errors. Due to the described explicit output prediction using disturbance information in the control formulation, this MPC scheme, however, combines both feedback and feedforward in a single controller. This has the advantage that optimality and constraint handling can be maintained for the overall controller.

5.3. Robust State Constraints

Most studies investigating the use of MPC for wind turbine control only consider control constraints. These are typically representing hardware limitations on the actuators, such as maximum torque and pitch rates. The main benefit of including those in the problem formulation explicitly is that a controller, which does not account for these limitations and simply saturates the actuated variables, will lose optimality in case it was designed using some form of optimal control design method, and also that stability cannot be guaranteed.

There are, however, also state constraints in the wind turbine control problem. Most notably, the generator may not exceed a certain safety critical threshold running under load. Therefore, most turbines will disconnect the generator and initiate a braking procedure if this threshold is reached. In order to avoid these shut-downs, the controller should ensure that the threshold is never exceeded during operation, which is an inequality constraint on the rotor speed state:

$$x_1 \leq \omega_{\text{max, load}}. \tag{5.40}$$

A constraint like (5.40) could be implemented directly in the described MPC setup similar to the control constraints. Unlike the control constraint, however, it cannot be guaranteed that the constraint is not violated in case of modeling errors or unknown present and future disturbances. For example, if the turbine is subjected to a step like gust that would, if the state constraint was omitted, lead to a rotor speed above the threshold, then including the constraint in the optimization will result in a trajectory that will hit the constraint exactly. Now, if either the current wind speed or future wind speed at any given point in the prediction horizon is even slightly higher than what was assumed by the controller, the generator speed will increase above the threshold value and violate the constraint. If the future wind speed is assumed to remain constant over the prediction horizon, it can clearly be seen that due to the stochastic nature of the wind, this condition will frequently be met and the constraint will be violated. Even if the future wind speeds are measured (see section 5.2.5), there will always be measurement errors causing the controller to transgress the constraint.

In order to make the use of state constraints viable, the controller needs to be robustified so that as long as certain assumptions about the disturbance are met, the state and control constraints will not be violated in the presence of unknown disturbances.

The method that is used to robustify the controller is based on the tube-based robust controller using an additive disturbance model as described by Rawlings and Mayne [91]; although it does not include all of its elements.

Here, only robust handling of the overspeed limit is considered, but the method can be expanded to include any linear state constraint. Furthermore, so far only robustification against unknown disturbances is considered, while modeling and estimation errors are ignored. This can be justified by the fact that the uncertainty on the future wind is several orders of magnitude larger than the modeling uncertainty and completely dominates the overall system behavior. If modeling uncertainty, which usually takes the form of a parametric uncertainty, were to be included, it could however be transferred into an equivalent additive disturbance as demonstrated by Rawlings and Mayne.

The basic concept of the tube-based robust controller relies on the assumption that an unknown but bounded disturbance is acting on the plant. While the actual disturbance is unknown to the controller and cannot be included in the prediction model, it is possible to calculate or approximate the maximum effect this disturbance can have on the predicted output trajectory based on the bounds of the disturbance. Based on this maximum effect, the control problem can then be modified accordingly such that the constraints are honored for all possible disturbance sequences.

Definition: In this context, a vector defined over the prediction horizon being smaller than another vector

$$y_{m|N} < z_{m|N} \tag{5.41}$$

is defined as each entry being smaller than the corresponding entry

$$y_{m+i} < z_{m+i} \qquad \forall \quad i = 0...N. \tag{5.42}$$

If y_{m+i} and z_{m+i} are vectors themselves, this holds for every entry.

5.3.1. Additive Disturbance

An additive disturbance is assumed to be acting on the plant. As described, only the mismatch between assumed and actual wind speed is considered as a disturbance so that the system model becomes

$$\begin{aligned} x_{k+1} &= \mathbf{A}_k x_k + \mathbf{B}_k u_k + \mathbf{E}_k d_k + \mathbf{E}_k v_{d,k} \\ y_k &= \mathbf{C}_k x_k \end{aligned} \tag{5.43}$$

with v_d as the unmeasured wind speed. This unmeasured wind speed encompasses both the measurement error as well as undetected future changes in wind speed at any point within the prediction horizon. Similar to the future wind speeds, a vector of future unknown wind speeds is defined:

$$v_{d,k|L} = \begin{pmatrix} v_{d,k} & v_{d,k+1} & \dots & v_{d,k+L} \end{pmatrix}^T . \tag{5.44}$$

Due to the unmeasured disturbance, the actual state trajectory will differ from the nominal trajectory as assumed by the controller. The real trajectory is the sum of the unforced response of the system $x_{f,k|L}$, which is a function of the current state and known disturbance trajectory, the effect of the controller outputs $u_{k|L}$, and the response to the unknown disturbances $x_{d,k|L}$

$$x_{k+1|L} = \underbrace{x_{f,k|L} + \mathbf{B_p} u_{k|L}}_{x^{\text{nom}}_{k+1|L}} + x_{d,k|L} \tag{5.45}$$

where the state trajectory due to the unknown disturbance is

$$x_{d,k|L} = \mathbf{E_p} v_{d,k|L}. \tag{5.46}$$

Calculating the disturbance trajectory as in (5.46) ignores the effect of the controller as it assumes the controller will not "see" and react on the effect of the disturbance. Due to its receding horizon nature, the Model Predictive Controller will however react on the state deviations caused by the unmeasured disturbance. The effect of closed loop control is included by employing a modified prediction matrix $\mathbf{E_{K,p}}$

$$x_{d,k|L} = \mathbf{E_{K,p}} v_{d,k|L}. \tag{5.47}$$

which is obtained by modifying (5.35) to use a state matrix which includes the effect of feedback

$$\mathbf{A_K} = \mathbf{A} - \mathbf{BK} \tag{5.48}$$

where $\mathbf{K}$ is the feedback matrix corresponding to the unconstrained MPC. As the disturbance trajectory $v_{d,k|L}$ is unknown, it is also not possible to calculate the resulting disturbance state trajectory $x_{d,k|L}$. It is, however, possible to approximate bounds on these disturbance state trajectories. Here, the assumption is made that the disturbance trajectories are bounded by a maximum measurement error:

$$|v_{d,k}| \leq v_d^{\max} \tag{5.49}$$

and that the maximum state trajectory is caused by the maximum measurement error being introduced at any time during the prediction horizon and then remaining constant over the

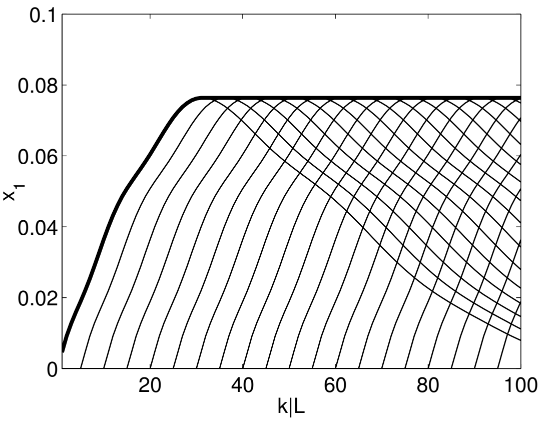


Figure 5.7.: x_1-state trajectories (thin lines) from introduction of a measurement error at different instants and bounding x_1-trajectory (thick line)

rest of the prediction horizon:

$$
\begin{aligned}
x^{\max}_{d,k+1|L} &= \max\left(\mathbf{E}_{\mathbf{K},\mathbf{p}} v^{j}_{d,k|L}\right) \qquad j = 0...L. \\
v^{j}_{d,k+i} &= \begin{cases} 0 & i < j \\ v^{\max}_{d} & i \geq j \end{cases}.
\end{aligned}
\tag{5.50}
$$

The resulting assumed bounding state trajectory for the state x_1 is shown in Fig. 5.7. This assumption can be motivated by the choice of a state constraint on only the rotor speed and knowledge that the wind speed to rotor speed input/output characteristic follows what is essentially a first order low-pass behavior. If different state constraints are considered, approximating the maximum disturbance trajectories becomes more complex. One way to approximate the maximum disturbance trajectories is using Monte-Carlo simulations: The system behavior is simulated for a large number of potential disturbance trajectories and the maximum trajectory is chosen to encompass all or most of the simulated trajectories. This is illustrated in Fig. 5.8, where 200 x_1-trajectories obtained from simulations with 200 randomly chosen disturbance trajectories are plotted, and it can clearly be seen that all state trajectories fall within the assumed bounds. For the generation of the disturbance trajectories, it was only assumed that (5.49) holds. Generally, the more conditions are placed on potential disturbance trajectories, such as rate limits or assumed frequency content, the tighter the bounding state trajectories can be chosen, which will be essential for placing constraints on the more oscillatory states such as tower motion. Under this assumption, it can easily be seen that all possible disturbance trajectories will fall within the assumed lower and upper bounds:

$$
-x^{\max}_{d,k+1|L} \leq x_{d,k+1|L} \leq x^{\max}_{d,k+1|L}. \tag{5.51}
$$

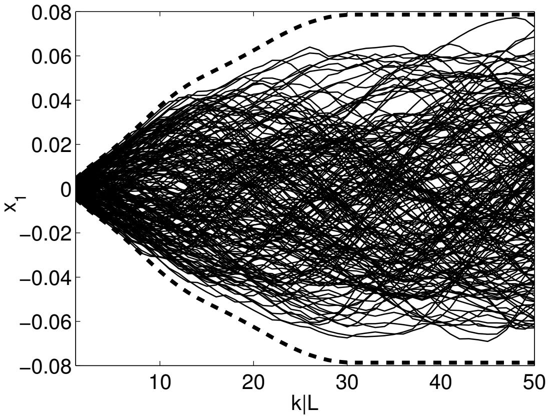


Figure 5.8.: x_1-state trajectories (solid lines) from 200 random disturbance trajectories and bounding trajectory (dashed line)

This maximum disturbance trajectory then also defines the maximum difference between nominal and actual state trajectories:

$$\begin{aligned} x_{k+1|L} &= x_{k+1|L}^{\text{nom}} + x_{d,k|L} \\ x_{k+1|L} &< x_{k+1|L}^{\text{nom}} + x_{d,k+1|L}^{\text{max}} \\ x_{k+1|L} &> x_{k+1|L}^{\text{nom}} - x_{d,k+1|L}^{\text{max}}. \end{aligned} \tag{5.52}$$

The maximum disturbance trajectory can be seen as defining an outer bounding tube on the set of all possible state trajectories. This tube is centered on the nominal trajectory as illustrated in Fig. 5.9.

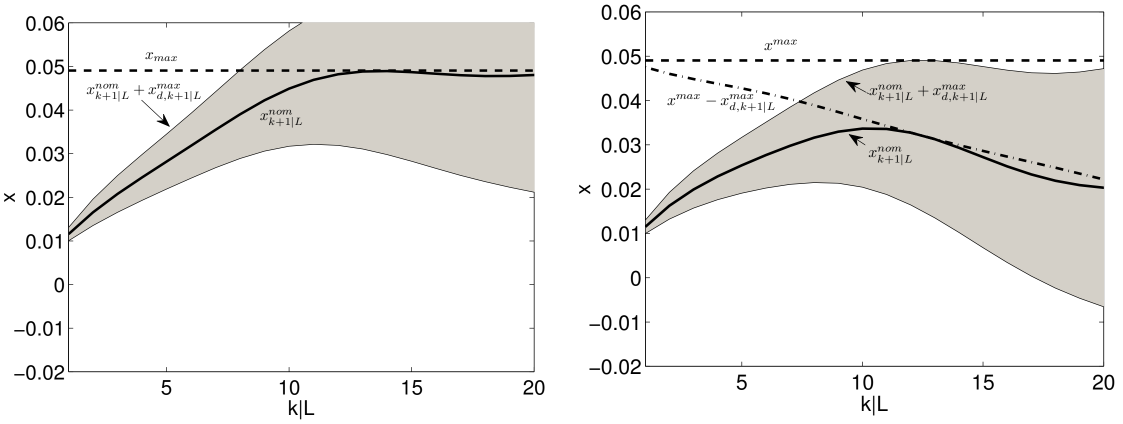
5.3.2. Controller Modification

The main goal of the robust controller is to ensure that the state constraint is not violated in the presence of unknown disturbances. In other words, it needs to be ensured that the actual trajectory $x_{k+1|L}$ does not violate the constraint for any possible sequence of unknown disturbances:

$$x_{k+1|L} = x_{k+1|L}^{\text{nom}} + x_{d,k|L} < x^{\text{max}}. \tag{5.53}$$

However, since in the calculation of the nominal trajectory only the measured disturbances are considered, it will only be ensured that

$$x_{k+1|L}^{\text{nom}} < x^{\text{max}}. \tag{5.54}$$



(a) Without Constraint Tightening

(b) With Constraint Tightening

Figure 5.9.: Illustration of the effect of tube based constraint tightening on the predicted state trajectory.

It can clearly be seen that if the constraints for the nominal control problem are tightened to $x^{\max} - x^{\max}_{d,k+1|L}$ then:

$$\begin{aligned} & x^{\text{nom}}_{k+1|L} < x^{\max} - x^{\max}_{d,k+1|L} \\ \Rightarrow \quad & x_{k+1|L} = x^{\text{nom}}_{k+1|L} + x_{d,k+1|L} < x^{\max} \end{aligned} \tag{5.55}$$

as $x_{d,k+1|L} < x^{\max}_{d,k+1|L}$ for all possible disturbance sequences $v_{d,k|L}$. This is further illustrated in Fig. 5.9. In the left plot no constraint tightening is used. While the nominal trajectory does not cross the bound on the state, parts of the tube of actual state trajectories lie outside of the allowed state values. In the figure on the right the constraints are tightened. This causes the nominal trajectory to have a certain distance to the state bound, such that the entire tube of possible real trajectories lies within the range of allowed values. When calculating the disturbance state trajectories according to (5.47), the assumption was made that the controller acts on the effect of the unmeasured disturbance. This, however, is only possible if the additional control action caused by the unmeasured disturbance does not cause the controller to operate at the control constraints. Therefore, not only the state constraints, but also the control constraints for the nominal control problem need to be tightened so that, even in the case of the maximum disturbance state trajectory occurring, the overall control action will be within the controller limits.

$$u^{\min}_{k+1|L} + \mathbf{K}x^{\max}_{d,k+1|L} \leq u_{k+1|L} \leq u^{\max}_{k+1|L} - \mathbf{K}x^{\max}_{d,k+1|L} \tag{5.56}$$

This can be seen as requiring the controller to keep a certain control reserve, which is to be used to handle unforeseen disturbances. How much control reserve is used depends on the assumed bounds: The better the measurements of current and potentially future wind speed, the more the controller is actually allowed to assume operation near its limits when calculating the optimal control trajectory. It should be noted that the constraint on the first time step u_k is not affected. Since only this first time step will actually be applied to the plant, this type of constraint tightening does not restrict the controllers ability to operate at its limits. It only becomes more cautious with respect to potential disturbances at future points in time.

5.3.3. Unmeasured Disturbance

The choice of bounds on the unmeasured disturbance needs to cover both the measurement/estimation error and potentially the error introduced by assuming the wind speed will remain constant at its current value over the prediction horizon in the non-preview case. Generally, it is not possible to actually determine the upper bounds on the unmeasured disturbance across all operating ranges and wind conditions. Moreover, it is most likely too conservative to actually use this maximum unmeasured disturbance as this would result in a controller which, even under normal operating conditions, will react too cautiously and might not provide the best performance. Instead, the values specifying the assumed unmeasured disturbance should be seen as an additional tuning parameter. They implicitly determine the safety distance the controller maintains to the actual hard limits on the state values. For example, using large values, the probability of an overspeed fault occurring can be reduced, but at the same time due to the large "safety zone", the controller has to keep the rotor speed excursions in a smaller range, which requires more pitch action and increases tower loads.

Here, the unmeasured disturbance is assumed to be $v_d^{\max} = 3.5\,\mathrm{m/s}$ in the non-preview case. This is in the same order of magnitude as the estimation error in the extreme gust cases (e.g., figure 5.3) but less than the worst case change in wind speed over any 5 s prediction horizon during these extreme gust events. Nevertheless, this choice has proven to provide a good performance in the load cases that are discussed in chapter 6. The assumption that the unmeasured disturbance acts as a step input is likely conservative with respect to the actual unmeasured disturbance trajectories explaining why lower than actual assumed disturbances already provide an acceptable performance.

This value is too high only at wind speeds near rated wind speed. Due to the low sensitivity of the aerodynamic moment to pitch movements, choosing a value this high for the unmeasured disturbance leads to the controller constantly operating near or at the robustified state constraint. Therefore, at rated wind speed an unmeasured disturbance of $v_d^{\max} = 1.5\,\mathrm{m/s}$ is used and the value is increased linearly with the wind speed so that at 13 m/s the full value is reached:

$$v_d^{\max} = \begin{cases} 1.5\,\mathrm{m/s} & v < v_r \\ 3.5\,\mathrm{m/s} & v > 13\,\mathrm{m/s} \\ 1.5\,\mathrm{m/s} + 2\,\mathrm{m/s} \cdot \frac{v - v_r}{13\,\mathrm{m/s} - v_r} & \text{else} \end{cases} . \tag{5.57}$$

Whenever preview control is considered, a fixed value of $v_d^{\max} = 0.5\,\mathrm{m/s}$ is used.

5.3.4. Backup Mode

If state constraints are used, there is a possibility that the QP problem is infeasible, i.e., no solution that does not violate any constraint exists. For the presented controller, the most likely cause for infeasibility would be an incorrect assumption on the unmeasured disturbance as discussed in the previous section. If a gust is stronger than the value of the unmeasured disturbance that was assumed, the controller may not be able to contain the turbine states within its limits without violating the controller actuation bounds. Another potential cause for infeasibility is mismatch between the turbine model in the controller and the actual plant leading to incorrect disturbance tubes. Further, one event that routinely needs to be considered in the wind turbine design is a partial or complete loss of counter torque at the generator. As this event is not included in the calculation of the disturbance tube, it can also not be guaranteed that it is possible to stay below the overspeed limit with the available actuator limits. In many cases, the model mismatch or torque loss uncertainty are not a severe challenge to the controller as they are also implicitly covered by the allowance for unmeasured disturbances. Only if they occur together with extreme changes in wind speed, such as in load case 1.5 where a grid loss is assumed to happen during an extreme operating gust with a one year return period, will they actually lead to infeasibility.

Infeasibility means that the controller does not "see" any way of maintaining operation within the given limits and thus, that there is a potential safety hazard. If a classical wind turbine controller were used and a safety hazard was detected, the supervisory control system would switch the controller from a power production mode to shut-down mode, which in many cases means the turbine is brought to a stop by pitching out the blades at a fixed pitch rate. Analogously, if in the MPC case the problem is infeasible the commanded and generator torque pitch rates are set to the maximum:

$$u_k = \begin{pmatrix} \dot{M}_{G,C}^{\max} \\ \dot{\theta}_c^{\max} \end{pmatrix} \tag{5.58}$$

Unlike in the case of a shut-down controller, this backup mode is not used until the turbine has come to a rest; it is only used for one time step. At each time step, there is a new attempt at solving the QP problem and only if it is still not feasible, the backup mode is used. As a result, if the controller is within safe operating limits again, it will immediately resume normal operation. If not, the turbine will shut down completely. It should be noted that the choice of backup mode is clearly driven by the nature of the state constraint that is considered. The backup mode is simply the controller action that reduces the rotor speed as fast as possible and therefore minimizes the severity of the constraint violation. If, e.g., a tower deflection constraint would be considered, there is no such clearly defined backup mode and a more elaborate scheme for handling infeasibility is required.

5.4. Algorithmic Implementation

To summarize the controller design, the steps that need to be performed at each controller cycle are stated here:

1. Perform measurement and time update of EKF according to equations (5.4) and (5.3) using the current turbine output measurements (4.26) and the state estimate of the last time step.
2. Determine the steady state operating point corresponding to the current wind speed estimate according to (5.27) and define the difference between the current state estimate and the linearization point as new state vector.
3. Update the partial derivatives with respect to aerodynamic torque and thrust in the continuous, linear state space model (4.30) according to the current wind speed estimate.
4. Discretize the state space model according to (4.34)
5. Solve the discrete Algebraic Ricatti Equation (2.39) to get the infinite horizon cost $\mathbf{P}_k$ and the terminal controller $\mathbf{K}_{\text{LQR}}$ (2.40).
6. Built up MPC prediction matrices (2.34).
7. If no preview control is considered, set $v_{k|L}$ equal to the current wind speed estimate.
8. Calculate the unforced output trajectory according to (5.34).
9. Calculate disturbance tubes using the current linear system model and terminal controller $\mathbf{K}_{\text{LQR}}$ according to (5.50) and using the chosen values for the unmeasured disturbance $v_d^{\max}$.

10. Calculate the control and state constraints according to the current linearization point and the calculated disturbance tubes (5.55) and (5.56) and update the constraints (5.39) accordingly.

11. Solve QP problem (5.38)+(5.39) for $u_{k|L}$.

12. If no feasible solution exists, set u_k according to backup controller (5.58).

13. Send the first entry u_k in the computed optimal control sequence $u_{k|L}$ to the plant.

6. Results

This chapter examines the performance of the controller design that was presented in the previous chapter. After a quick look at some of the implementation details and introduction of a reference, baseline controller, it discusses results from simulations closely resembling those load cases that are also considered when designing an entire wind turbine.

6.1. Simulation Setup

All simulations are run in SIMULINK. The aero-elastic simulation tool FAST (see section 2.2.5) that is used to simulate the plant, i.e., the wind turbine, is included in the SIMULINK model via the provided S-function interface. For all simulations that require turbulent wind fields, the wind time series are generated using the Kaimal model [45] with TURBSIM [48] according to the IEC regulations [39]. FAST does not include any actuator models so those are added directly in SIMULINK: The pitch actuator model consists of a second order transfer function (4.18) with the same parameters as stated in table 4.3 and rate limitation to $\pm 10\,{}^{\text{deg}}/_{\text{s}}$ for each of the three blades. The generator torque actuator model is a simple one time step delay operator.

The following signals and their respective FAST sensor names are assumed to be measurable and available to the controller:

- The rotor speed $RotSpeed$
- The pitch angle[1] $BldPitch1$
- The tower top acceleration in the longitudinal direction $YawBrTAxp$

Loads are evaluated for the three principal components tower, main shaft, and blades at these three sensor locations:

- Tower base bending moment in the fore-aft direction TM_{yt}
- Main shaft torque $RotTorq$

[1] as no individual pitch control is used the pitch angles for all three blades are always identical and the pitch angle at blade 1 can be used as the collective pitch angle

- Out-of-plane bending moment at the blade root for blade 1 $RootM_{yc1}$

Fatigue loads at these locations are calculated as damage equivalent loads (see section 2.2.4) which are calculated in a postprocessing step after the actual time series simulation using the WAFO toolbox [113] for MATLAB with a reference cycle frequency of 1 Hz. A material slope of $m = 4$ corresponding to steel is used for the tower and main shaft while for the blades, which are made of composite materials, a slope of $m = 10$ is applicable.

6.1.1. MPC Implementation

The MPC that is described in the previous chapter is implemented using an interpreted MATLAB script that is called from the SIMULINK environment. QPOASES [23] is used as the underlying QP-solver which is called once in every time step. The initial condition for the QP-solver is the $u_{k|L} = 0$ at every time step and the option to "hot-start" the solver that is provided in QPOASES is not used.

Both the MPC and the estimator are chosen to run at a sample rate of 10 Hz and the "measurements" taken from FAST are down-sampled accordingly. A prediction horizon of 5 s is used so that $l = 50$ and, as there are two controller outputs, the QP-problem has the dimension 100.

FAST does not provide a convenient sensor for the rotor plane effective wind speed which needs to be assumed measurable if preview control is simulated. Therefore, the estimated wind speed that is calculated via the estimator is used as a surrogate for the wind speed preview measurements. Whenever preview control is considered, the time series of the estimated wind speed, corrected for estimator delay, from a prior run using the same wind field is made available to the controller for the full prediction horizon. Especially the high frequency components of the wind field are attenuated by the wind speed estimation so that this assumption is conservative as it assumes imperfect preview measurements. It has been shown [100, 96, 99] that by using LIDAR measurements, it is not possible to correctly measure the high frequency components of the wind field ahead in time due to the limited spatial resolution of most LIDAR systems and the fact that these frequencies are generally related to smaller structures in the turbulent wind field that tend to disperse before they arrive at the turbine. Hence, the choice of using the estimated wind speed as the preview signal, although made because of lack of suitable alternative, might be a more realistic assumption, than using the true rotor plane effective wind speed.

One important aspect for any MPC application is always the computational load. In the current implementation, it takes about 0.37 s of CPU time to simulate 1 second of real time on a standard commercial PC with a clock frequency of 2.5 GHz. This simulation time includes the computational load that is required to run FAST as well as the MATLAB/SIMULINK overhead. For a real controller implementation, only the controller portion of the simulation setup would need to be implemented on a real-time system such as a Programmable Logic Controller (PLC) and the time required to run the plant model, in this case FAST, is therefore of no concern. Figure 6.1 shows the distribution of the computational load for each of the major steps in the MPC algorithm. It can be seen that a significant amount of time is required for setting up the equations and constraints, while solving the actual QP-problem only requires around 10% of the total computational load. Considering that especially the non-QP steps are in no way optimized for speed[2] and that they could be made significantly faster, an implementation on a real-time system does not seem impossible.

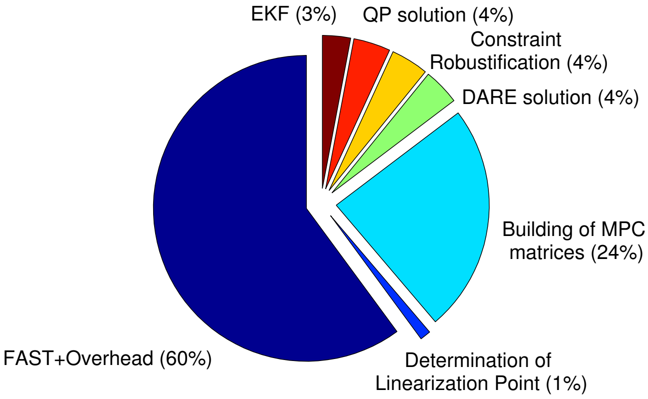


Figure 6.1.: Distribution of the computational load for each controller time step

6.1.2. Baseline Controller

In order to better analyze the behavior of the Model Predictive Controller, its response for the various scenarios that are evaluated is always compared against a baseline controller. This baseline controller is designed to be a representation of a typical, classically designed controller that can be found on modern wind turbines. As the performance evaluation is performed only in the above rate, pitch controlled operating region, only the controller for full load operation is described here. Similarly to the MPC, the baseline controller will also be in partial load operation temporarily, even if the mean wind speed is above rated wind speed. The partial load controller and the switching conditions between partial load and full load are described in appendix B.2.

The torque controller is chosen to follow the constant power strategy with the torque control law accordingly given by (2.12). The pitch controller is designed as a gain-scheduled PI

[2] Unlike the QP-solver they also run in the MATLAB interpreter mode which is slow compared to compiled code

controller which controls the rotor speed. Details on the tuning and gain scheduling of this pitch controller can be found in appendix B.

The EKF, which is used in the MPC setup to estimate the turbine states and wind speed, assumes the tower top acceleration to be measurable. In order to allow a performance comparison that is as fair as possible, the baseline controller also needs to make use of the tower acceleration signal. Therefore, a tower damper is added that uses the measured tower top acceleration to generate a pitch offset that is added to the output from the speed controller. The tower damper is also designed as a PI-controller and details about its tuning can also be found in appendix B.

Since the MPC only considers collective pitch action, an individual pitch controller is not included in the baseline control design. The same applies to a potential Drive Train Damper which was excluded in the MPC problem formulation. Figure 6.2 shows the final baseline controls structure.

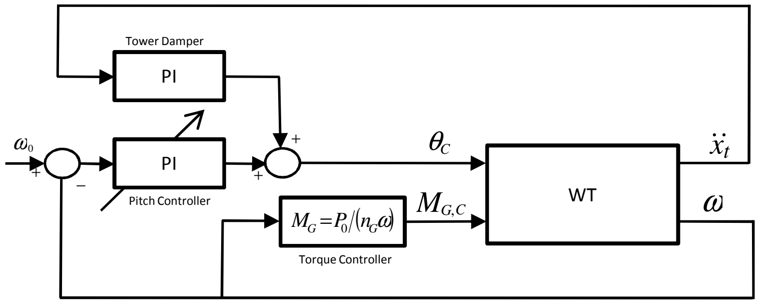


Figure 6.2.: Block diagram of baseline controller in full load operation

One event that is considered in the following simulation results is the so-called grid loss. In case of a grid loss, the turbine is no longer capable of feeding power to the grid so the torque command needs to be set to zero. The baseline controller then stops the turbine by pitching towards a pitch angle of 90 deg with a fixed pitch rate of 10 deg/s. The tower damper is deactivated during this shut-down process.

6.2. Normal Power Production Operation

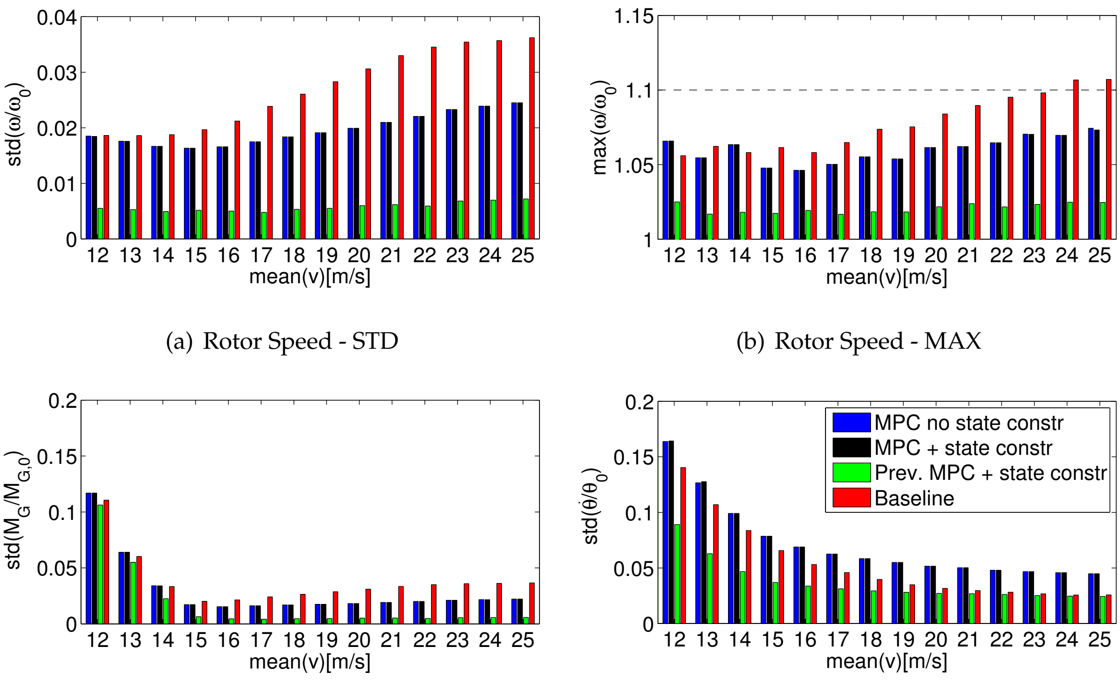
First, the behavior of the MPC is examined under Normal Power Production Operation (NPP) that would typically be run during the design assessment of any wind turbine. The turbine behavior is simulated during multiple time series each with a 10 minute duration and varying wind conditions. The wind speed for each time series consists of a constant mean wind speed and a superimposed turbulent component with zero mean speed where the turbulence intensity is a function of the mean wind speed for the given time series.

Here, mean wind speeds from 12 m/s to 25 m/s in steps of 1 m/s are considered and a turbulence distribution according to the IEC turbulence class "B" (figure 2.9) is assumed. Wind speeds below 12 m/s are not evaluated as, due to the rated wind speed of the given turbine being above 11 m/s, the turbine will spent most of the time in partial load operation during these time series. While the MPC developed in the previous chapter is able to handle partial load operation, the presented MPC has been designed with a focus on full load operation and it has been shown earlier that the benefits of MPC in partial load are negligible [51, 104] so that this operating range is excluded from the analysis. At each mean wind speed, two seeds, i.e., two different realizations of the turbulence generated using a random number generator, are used. Any further variation that is sometimes considered during a loads assessment such as upflow or yaw misalignment is ignored so that in total the NPP set consists of 28 individual time series.

Four different controllers are considered for the analysis: The first is the MPC described in the previous chapter without any preview information available, i.e., it is assumed that the current estimate of the free stream wind speed is valid throughout the prediction horizon. For this first controller, the constraint on the rotor speed state is further deactivated so that only the control constraints and the constraint on the pitch angle are active. The second controller is identical to the first controller except that here the rotor speed constraint, including its robustification, is active. The third controller is also the MPC with the rotor speed constraint, but preview information is assumed to be available and the assumption on unmeasured disturbance is modified according to section 5.3.3. Finally, the fourth controller is the classically designed baseline controller that was introduced in section 6.1.2.

Figures 6.3 and 6.4 show the key operational metrics and resulting loads per wind speed. Whenever a standard deviation (STD) is calculated, it is averaged between the two simulations at the same wind speed. For the maxima (MAX), the higher value of the two simulations is used while for the damage equivalent loads (DEL), the two time series are combined using the Palmgren-Miner Rule (see section 2.2.4). All signals are normalized: The rotor speed ω and the generator torque M_G are normalized to their respective rated values. The tower base bending moment in the fore-aft direction TM_{yt}, the main shaft torque $RotTorq$, and the out-of-plane bending moment at the root of blade 1 $RootM_{yc1}$ are all normalized to the stationary values corresponding to the mean wind speed of the given time series. Similarly, the pitch angle and rate are normalized to pitch angle necessary to achieve steady state at the given mean wind speed and rated torque.

The first observation can be made by looking at the maximum and standard deviation of the rotor speed in figure 6.3 (a) and (b): All four controllers clearly manage to track the



(a) Rotor Speed - STD

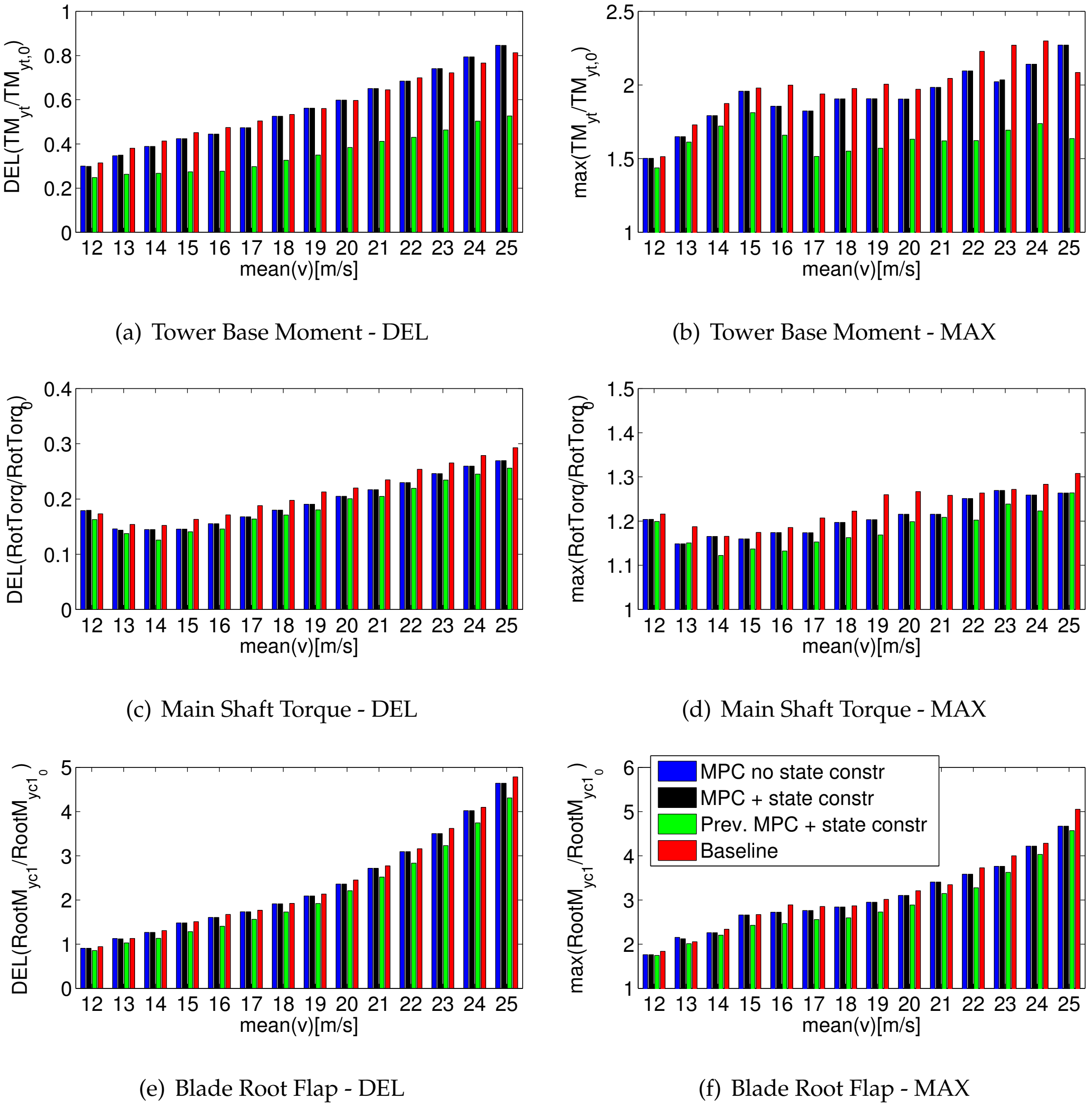
(b) Rotor Speed - MAX

(c) Generator Torque - STD

(d) Pitch Rate - STD

Figure 6.3.: Key operational metrics from NPP simulations with a mean wind speed ranging from 12 m/s to 25 m/s with two seeds per wind speed and using IEC B turbulence

rotor speed set-point tightly with standard deviation of the tracking error only being a few percent of the set-point. Except for the baseline controller at 24 m/s and 25 m/s, all controllers further manage to keep the rotor speed below its limit at all times. Generally, the baseline controller provides a level of speed tracking performance in the same order of magnitude as the two non-preview MPCs. However, with the baseline controller, the speed tracking performance shows greater variability over the different wind speeds with the standard deviation of the rotor speed being significantly higher at the high end of the wind speed range, while with the MPCs, the speed tracking performance is flatter across wind speeds. This is due to the simple gain scheduling approach that is used for the baseline controller which, for example, does not account for the variation in the sensitivity of the aerodynamic moment with respect to the wind speed. Looking at the speed tracking performance, it can further be noted that under the simulated conditions, the rotor speed constraint is not necessary. Even with the first controller, where this constraint is deactivated, the rotor speed is never above the limit. However, as the results with and without this constraint are not numerically identical for some wind speeds, there must be multiple situations where the state-constrained controller "sees" an overspeed threat and therefore reacts differently than the unconstrained controller. The effect of the rotor speed constraint in NPP operation is further analyzed in section 6.2.1.



(a) Tower Base Moment - DEL

(b) Tower Base Moment - MAX

(c) Main Shaft Torque - DEL

(d) Main Shaft Torque - MAX

(e) Blade Root Flap - DEL

(f) Blade Root Flap - MAX

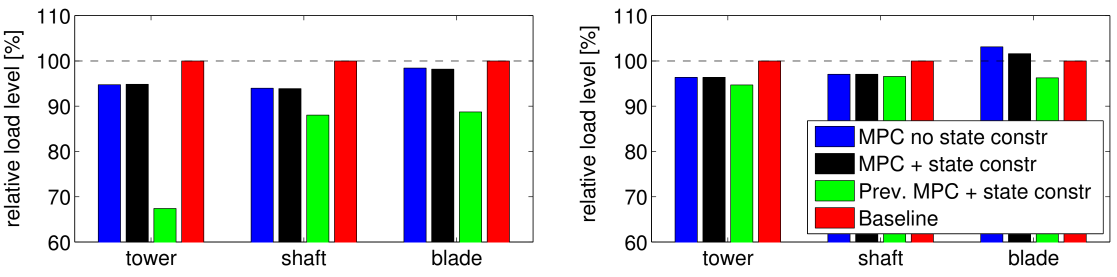
Figure 6.4.: Maximum and Damage Equivalent Loads from NPP simulations with a mean wind speed ranging from 12 m/s to 25 m/s with two seeds per wind speed and using IEC B turbulence

Looking at the distribution of maximum and damage equivalent loads for the range of wind speeds in figure 6.4, it becomes clear that the two non-preview MPCs provide a level of performance that differs only slightly from the baseline controller results; although slight improvement can be seen in the fatigue and extreme loads at most wind speeds. Only the tower fatigue loads at wind speeds above approximately 20 m/s increase through the use of the MPC. However, in this range of wind speeds the difference in speed tracking performance is also the greatest. Within certain limitations, it is generally possible to trade-off speed tracking against tower fatigue loads [50] through the choice of controller tuning, i.e., some speed control performance may be sacrificed to reduce tower loads. Section 6.2.2 will further examine this property of the full load speed control problem. The slight improvement in blade flap and tower DELs at low wind speeds, where the speed control performance is on a similar level, can be explained by the fact that the MPC design does include knowledge of the blade dynamics and blade-tower interaction explicitly through the used state-space model while the baseline controller does not.

Interestingly, the two non-preview MPCs seem to perform worse than the baseline controller at 12 m/s with respect to containing rotor speed excursions. They require more pitch actuation, visible in the increased standard deviation of the pitch rate $\dot{\theta}$, but still lead to a higher maximum rotor speed. Here, the issue is that at 12 m/s the controller will switch frequently between full load operation and partial load operation. The switching behavior is not included in the MPC problem formulation, i.e., for the calculation of the optimal control trajectory the controller will assume to stay in the operational mode in which it is currently operating, even if in reality, there is a mode switch in the next controller time step. This causes the controller to switch sub-optimally during these manoeuvres and diminishes controller performance.

Figure 6.5 shows the extreme and fatigue loads obtained from combining all 28 time series of the NPP set. For the extreme loads, the highest load that occurs at the specific component over all time series determines the overall maximum load, while for the fatigue loads, the DELs for the individual time series are combined into a single DEL under the assumption of an IEC wind class 2 distribution of mean wind speeds. These results show an overall tower and main shaft fatigue loads reduction of 5% and a blade root fatigue reduction of 2% for the MPCs with and without the state constraint compared to the baseline level. Similar to the fatigue loads, the maximum tower and main shaft loads are also decreased by around 4%. Neither of the two non-preview MPC is able to deliver extreme blade root loads below the baseline level though. Nevertheless, the extreme blade root load is reduced by the addition of the state constraint, indicating that the event where the highest blade load occurs is one where the constraint is active. Overall, with fatigue and exteme loads only slightly below the baseline level, the results for the two non-preview MPCs are in line with Botasso's result that it is hard to significantly beat a well tuned classical controller under nominal conditions [7].

Compared to the three other controllers, the preview controller shows a significantly improved performance. It is able to cut the rotor speed deviation at least to a third of the non-preview level with less pitch and generator torque activity. The resulting extreme and fatigue loads are lower at all wind speeds for all three main components. The biggest loads reductions are seen at the tower base where the maximum load is reduced by 6% while damage equivalent load is down by just over 30%. Of course, this loads reduction is only applicable to the full load operating region, but considering that approximatly 70% of all



(a) Fatigue

(b) Extreme

Figure 6.5.: Combined extreme and fatigue loads for the entire full load operating range from 12 m/s to 25 m/s. Results are normalized to the baseline controller level.

tower fatigue is caused in full load and assuming there is no change to partial load operation, the overall reduction in tower DEL is still likely in the range of 20%, which would more than double the lifetime of the tower.

6.2.1. Overspeed Risk in NPP

The results presented in the previous section show that except for the baseline controller at very high wind speeds, there is no real risk of triggering an overspeed fault. This is not very surprising as any controller which is not capable of maintaining operation would be poorly designed. This, however, also means that under these conditions the constraint on the rotor speed state and its robustification would not be necessary. To analyze the behavior and effectiveness of this constraint, first the distribution of rotor speeds within one time series is examined. Figure 6.6 shows the probability of occurrence of a particular rotor speed computed based on one of the time series from the NPP set at a mean wind speed of 20 m/s. The distributions resemble a normal distribution for all four controllers. The distribution obtained with the preview controller is significantly more centered around the mean value. A result which is expected considering the significantly lower standard deviation. The three distributions for the other controllers are all very similar. As no overspeed violations were observed for any controller during this 10 minute time series, the frequency of occurrence is zero for $\omega > 1.1\omega_0$ for all controllers. Nevertheless, it is clear that the shape and variance of the distribution directly drives the probability of triggering an overspeed fault. Especially if the distributions are considered to be normal, the average number of overspeed faults, e.g., trips per month, could be calculated for a given wind speed and turbulence intensity based on the standard deviation of the rotor speed. Further, if the rotor speed tracking is tightened by a more aggressive tuning of the controller, the trip probability will be reduced. Unfortunately, there is always a trade-off involved in tuning the controller. Tighter speed tracking generally also leads to higher tower loads. A relationship which is further examined in the next section.

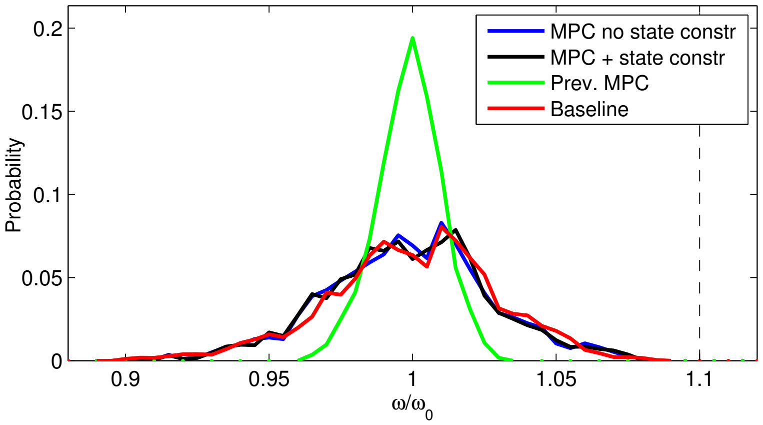


Figure 6.6.: Relative frequency of occurrence of the normalized rotor speed from an NPP simulation at a mean wind speed of 20 m/s using a bin width of 0.005 for $frac\omega\omega_0$

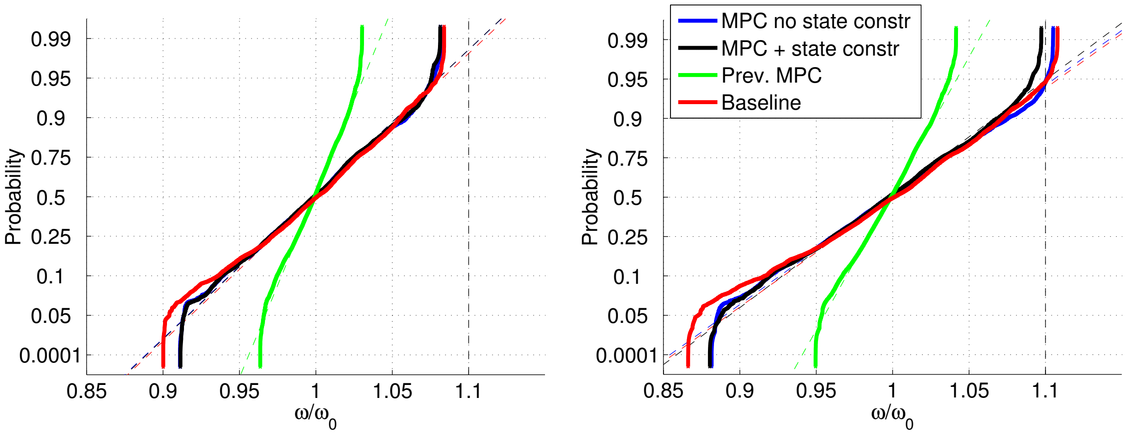
In order to take a deeper look at the influence of the rotor speed constraint on the overspeed risk, the same time series is also simulated with a considerably higher[3] turbulence intensity of $T_i = 21\%$. Figure 6.7 compares the normal probability plots at both turbulence intensities. The first observation is that while the curves are reasonably linear near the center, they depart from the normal curve at both ends for all controllers. The distributions are clearly thin-tailed and normal distribution cannot be assumed in the range near the maximum rotor speed. At the higher turbulence intensity, the slope of the curve is lower, which in the case of the baseline controller and unconstrained MPC causes it to cross the overspeed level. While at $T_i = 15\%$, the upper tails for the three non-preview controllers are almost identical, at $T_i = 21\%$, the constrained MPC differs from the other two at high rotor speeds. Its curve bends upward earlier and thus never crosses the limit.

In summary, the addition of the rotor speed constraint helps to avoid overspeed trips caused by high turbulence by modifying the controller behavior at high rotor speeds, i.e., the high tail of the probability distribution, only. In a certain range, it can avoid having to make the overall controller more aggressive.

6.2.2. Tuning

One of the claimed benefits of MPC is the direct control over the performance trade-offs that need to be considered for the controller tuning. To demonstrate the usefulness of this

[3] Using Turbulence class B, the turbulence intensity at 20 m/s was 15%



(a) $T_i = 15\%$

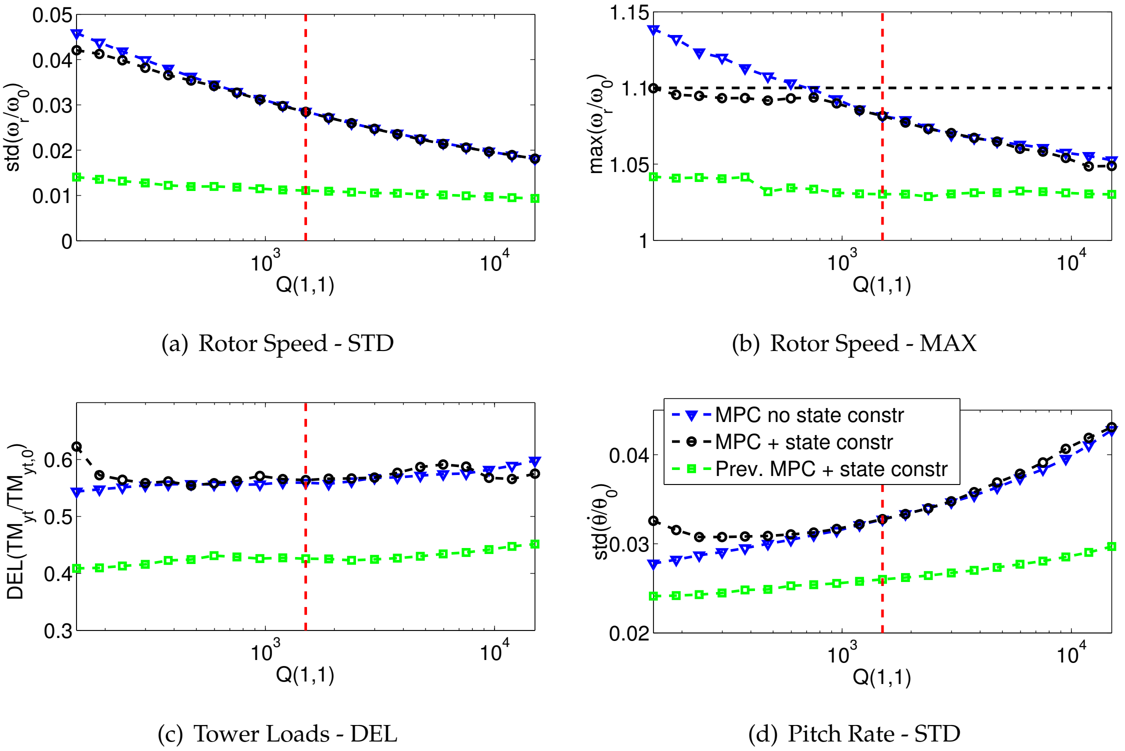
(b) $T_i = 21\%$

Figure 6.7.: Normal probability plot of the rotor speed from a NPP simulation at a mean wind speed of 20 m/s for two different turbulence intensities

MPC feature for wind turbine control, a single NPP time series is simulated for a range of potential tuning settings. Here, the weight placed on the rotor speed state (the first entry of $\mathbf{Q}_1$) is chosen to exemplarily show the effect of the tuning weights and is varied between 1.5×10^2 and 1.5×10^4 using logarithmic spacing. Figure 6.8 shows the effect of these different settings on the controller performance for key metrics.

The first observation is that increasing the weight placed on the rotor speed increases the rotor speed tracking performance as expected. For all three MPC variants, a higher weight leads to a lower standard deviation. This comes at the cost of significantly higher pitch activity and somewhat higher tower fatigue loads confirming the three way trade-off between speed control, tower loads, and pitch activity that was already reported in [50]. The impact on tower fatigue loads seems to be smaller than on pitch rate and rotor speed control, but considering that a material slope of $m = 4$ is used to calculate these DELs, the approximate difference of 10% between the lowest and highest load for each controller still corresponds[4] to a 50% increase in accumulated component damage and corresponding lifetime reduction by one third. Given the relativly small differences in performance that were observed in the previous sections, it is likely that it is also possible to tune the baseline controller to provide a performance similar to the different tuning points. However, since there is no clear link between gains and performance such a tuning is much more cumbersome in the baseline case compared to the "tuning knobs" the MPC, as well as most other optimal control formulations, provide the turbine designer with. For the same reason, having this easy and direct access to the design trade-offs is also a pre-requisite for any kind of site-specific or self-adjusting control design.

[4] $1.1^4 \approx 1.5$



(a) Rotor Speed - STD

(b) Rotor Speed - MAX

(c) Tower Loads - DEL

(d) Pitch Rate - STD

Figure 6.8.: Key operational metrics from NPP simulations with a mean wind speed of 20 m/s and 15% turbulence intensity for different MPC tuning weights placed on the rotor speed state. The dashed red line marks the chosen setting that is considered in all other simulations of this chapter.

Secondly, the MPCs with and without the state constraint behave very similar except at low rotor speed weight settings. Without the state constraint, the overspeed threshold is clearly violated for all settings below 1.0×10^4 while the constrained controller manages to stay within the limit. The effect on the standard deviation of the rotor speed and the tower loads is only seen in the most extreme cases. The addition of the speed constraint can be seen as decoupling the overspeed behavior from the general tuning problem. Only at very low weights placed on the rotor speed, the tracking performance will become so poor that the speed constraint will be active frequently and drive the tower loads and pitch activity up.

Finally, the preview controller provides a significantly improved performance for all considered tuning options and for all four performance metrics. This should be considered when designing any type of preview controller, MPC based or not, as by modifying the tuning settings some of the performance increase in, e.g., overspeed prevention can be sacrificed for further loads reductions. Here, the big benefit of MPC is that with the addition of the feedforward component, the feedforward and feedback components of the controller are still treated and tuned together. Further, some of the proposed non-MPC preview controllers, such as the static system inversion approach [98], do not provide any means for tuning the feedforward path controller as they are designed purely with the objective of canceling the disturbance impact on the rotor speed.

6.3. Gusts

Next to operation in turbulent conditions, which represent normal power production operation, the performance of the controller during deterministic gust load cases is also crucial for the turbine design. Here, the main objective of the controller is to reduce the maximum load that occurs during the event as well as potentially keep the turbine within its operating limit so that it does not need to be shut down. The most important gusts that have to be considered during the design are the Extreme Operating Gust (EOG) and the Extreme Coherent Gust (ECG) as prescribed by the IEC guidelines [39].

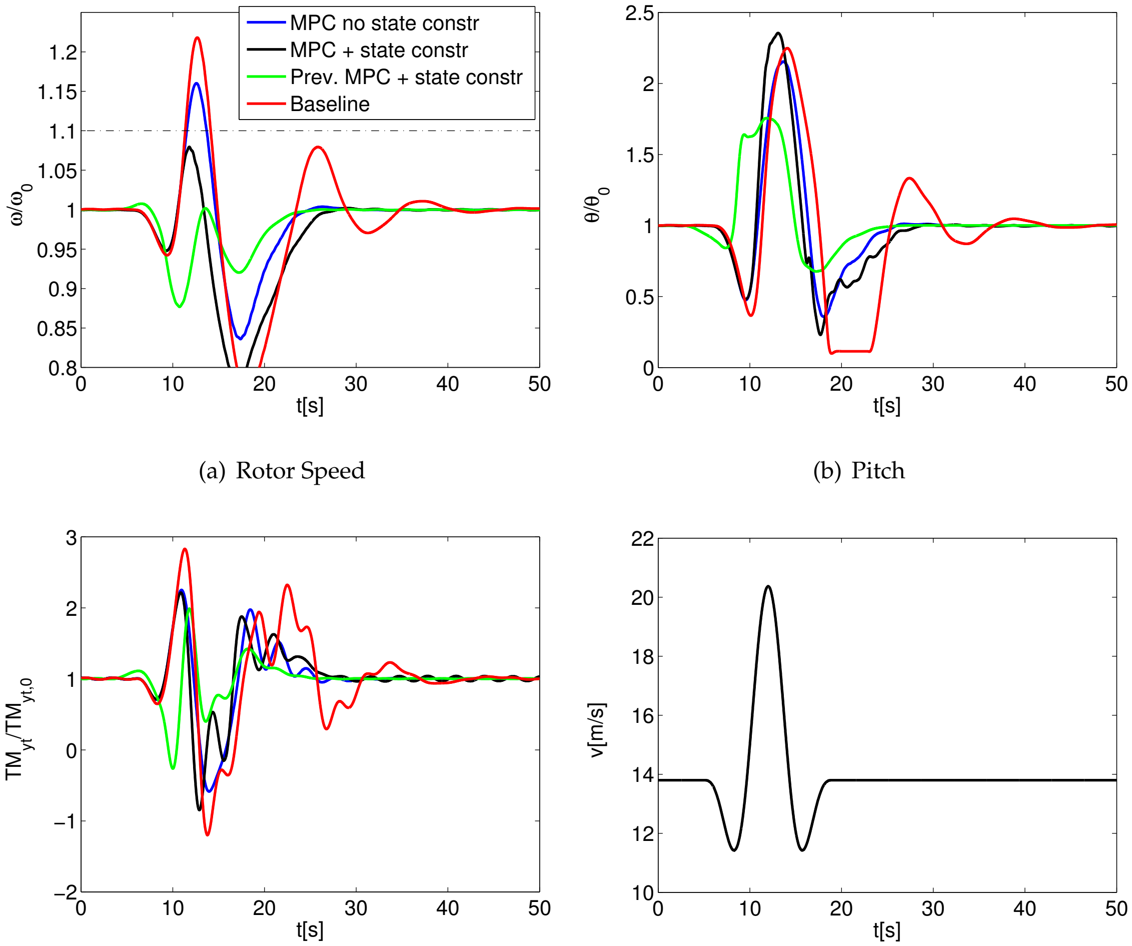
6.3.1. Extreme Operating Gust

Load case 1.6 of the IEC guidelines requires simulation of an EOG with a return period of 50 years. Here the simulation is performed at rated wind speed, 120% of rated wind speed, and the cut-out wind speed or $v = 11.5\,\mathrm{m/s}$, $v = 13.8\,\mathrm{m/s}$, and $v = 25\,\mathrm{m/s}$, respectivly. The magnitude of the gust is based on the turbulence class B assumption that was also used in the NPP simulations. Figure 6.9 shows the resulting time series for the $v = 13.8\,\mathrm{m/s}$ case while table 6.1 compares the resulting maximum values for all three wind speeds.

It can be seen that both the unconstrained MPC and the baseline controller lead to violation of the overspeed constraint, while the constrained MPC manages to keep the rotor speed just below the limit. If preview control is used, the turbine will start pitching much earlier and the speed excursion is greatly reduced so that there is no threat of an overspeed fault. This type of behavior matches what has been found in other studies that applied preview control to the main speed regulation problem (e.g., [50, 51, 98]).

When comparing the maximum resultant tower base bending moments, several observations can be made: As expected, the preview controller also reduces the load on the tower. The two non-preview MPC have almost identical maximum tower loads as the maximum occurs when the controller just starts to "see" the state constraint and the controllers begin to differ in their behavior. So in this case, using a state constraint does not increase the load although the controller reacts more aggressively.

Further observations can also be made from comparing the results for the different wind speeds given in table 6.1:



(a) Rotor Speed

(b) Pitch

(c) Tower Base Bending Moment

(d) Wind Speed

Figure 6.9.: Comparison of MPC and baseline controller during an Extreme Operating Gust at a wind speed of 13.8 m/s with a return period of 50 years

- The preview MPC provides superior performance across all three wind speeds with the resultant rotor speed excursions and tower base bending moments being the lowest of all four controllers at all simulated wind speeds. At $v = 11.5$ m/s and $v = 13.8$ m/s, the resulting tower base bending moment is at least 10% below the three other variants while at $v = 25$ m/s the benefit is even higher. Considering that load case 1.6 is often the load case that drives the design of the tower, this is a significant reduction in loads that in some cases may directly translate into reduced costs for the tower.

- With the addition of the state constraint to the MPC, the maximum rotor speed stays below the assumed overspeed trigger value at all three wind speeds even though in the unconstrained case there is a significant exceedance. That means in these scenar-

	$\max\left(\frac{\omega}{\omega_0}\right)$	$\max\left(\frac{\theta}{\theta_0}\right)$	$\max\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.221	5.68	2.16
unconstrained MPC	1.124	5.11	1.88
constrained MPC	1.071	5.13	1.81
constrained MPC + preview	1.015	8.89	1.55

(a) $v = 11.5\,\mathrm{m/s}$

	$\max\left(\frac{\omega}{\omega_0}\right)$	$\max\left(\frac{\theta}{\theta_0}\right)$	$\max\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.218	2.25	2.83
unconstrained MPC	1.160	2.15	2.25
constrained MPC	1.080	2.36	2.21
constrained MPC + preview	1.008	1.76	1.99

(b) $v = 13.8\,\mathrm{m/s}$

	$\max\left(\frac{\omega}{\omega_0}\right)$	$\max\left(\frac{\theta}{\theta_0}\right)$	$\max\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.319	1.32	3.98
unconstrained MPC	1.185	1.42	3.10
constrained MPC	1.099	1.58	3.80
constrained MPC + preview	1.051	1.48	2.33

(c) $v = 25\,\mathrm{m/s}$

Table 6.1.: Maximum values of rotor speed, pitch angle, and tower base moment during EOG50 gust simulations starting at $v = 11.5\,\mathrm{m/s}$, $v = 13.8\,\mathrm{m/s}$, and $v = 25\,\mathrm{m/s}$

ios, the behavior of the controller is mainly driven by the state constraints and less by trying to minimize the cost function. The controller is operating at its constraints.

- Even at $v = 11.5\,\mathrm{m/s}$, the state constraint is necessary to prevent the overspeed fault. However, it can be seen that with the current choice of tuning parameters and unmeasured disturbance assumption the controller is overly conservative. At $\frac{\omega}{\omega_0} = 1.071$, the maximum rotor speed is still significantly below the limit of 1.1, but the constraint was clearly active as the unconstrained case shows a different result. This behavior can be explained with the low sensitivity of the aerodynamic torque with respect to changes in pitch angle near rated wind speed (compare figure 2.5). Due to the low sensitivity, the controller would need to pitch the blades out more in case of an unmeasured disturbance and the controller therefore keeps more control reserve via the tube formulation.

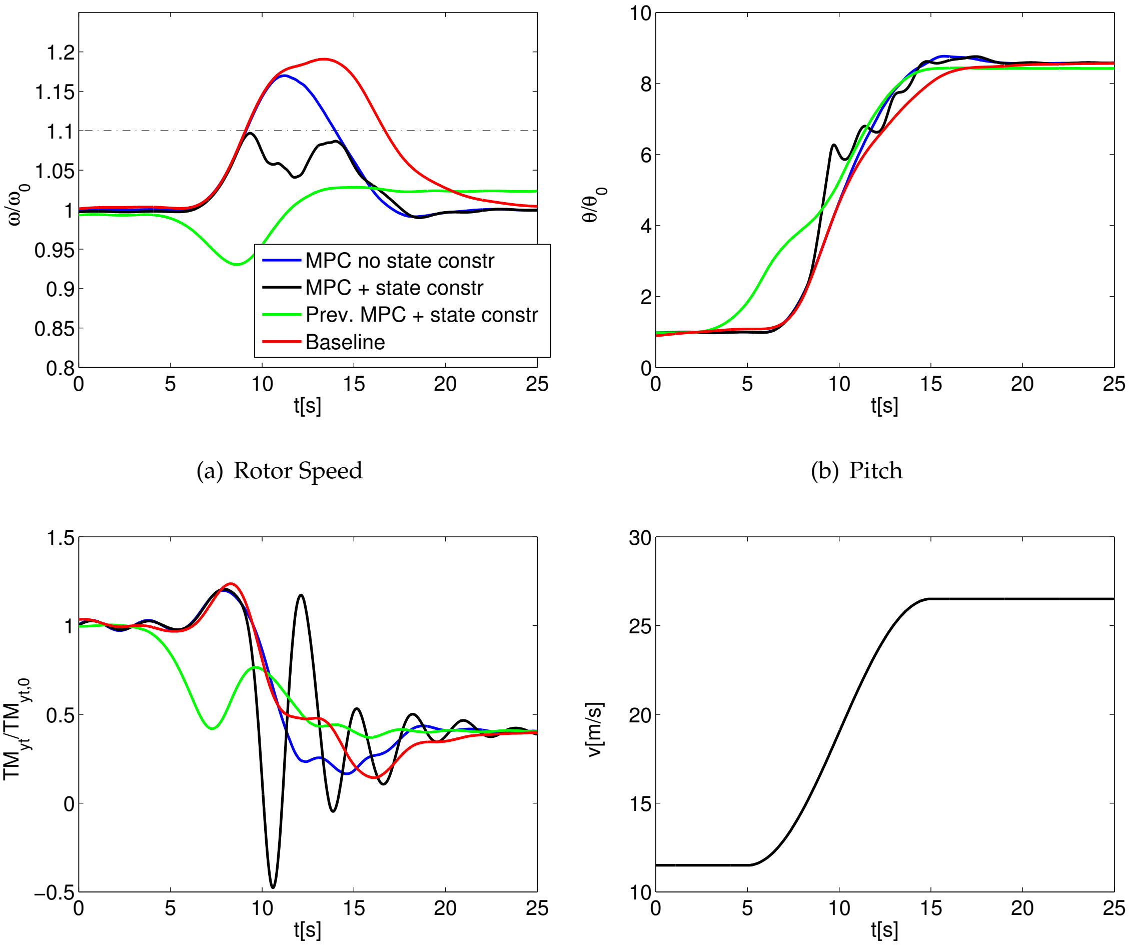
- At $v = 11.5\,\mathrm{m/s}$, using the preview controller results in the highest pitch excursion. This is due to the turbine briefly going into partial load operation during the early part of the gust. In the current design, the MPC is not allowed to use the pitch, i.e., switch to full load operation, for the entire duration of the prediction horizon even if the previewed wind speed is above rated wind speed. Therefore, once the turbine is back to full load operation it needs to pitch especially fast. This is a unique situation for the preview controller near rated wind speed. The behavior during these events could be improved by an improved set-point scheduling in the transition region [58].

- At $v = 25\,\mathrm{m/s}$, it can be observed that adding the state constrained increases the maximum resultant tower base bending moment significantly over the non-constrained case. This is due to the very fast pitch motion and corresponding change in thrust that the constrained controller requires to maintain rotor speed below the limit. In case the tower load at this wind speed is of greater concern than the gust-ride-through feature of the MPC, it might be necessary to relax the constraint and include the maximum tower load as a further state constraint.
- At $v = 25\,\mathrm{m/s}$, even using the preview controller leads to a very significant speed excursion. This is mainly due to the assumption that the preview controller has the wind speed that the estimator would observe available ahead in time and not the true wind speed. As shown in section 5.1.2, the estimator will attenuate the gust amplitude somewhat and this attenuation will cause imperfect gust ride-through, especially at high wind speeds where the assumed gust amplitudes are also the highest. For the controller, this means that some unmeasured disturbance needs to be assumed also in the preview case. How much exactly will be needed depends directly on the performance of the measurement device that is used to generate the preview measurements and can only be evaluated using a detailed sensor model [100].

6.3.2. Extreme Coherent Gust

Figure 6.10 shows the resulting time series from an Extreme Coherent Gust simulation starting at rated wind speed corresponding to load case 1.9 of the IEC guidelines and table 6.2 lists the resulting maximum values. These results echo what was also observed for the Extreme Operating Gust: If preview information is available, the turbine is able to ride through the gust without much trouble. For this type of gust, neither the rotor speed nor the tower base bending moment increase above the respective stationary values at all. If no preview information is available, then the MPC without the state constraint behaves similar to the baseline controller: The rotor speed rises well above the assumed maximum rotor speed and the turbine would have triggered an overspeed fault. If the speed constraint is activated, the turbine starts to pitch aggressively once it ”sees” the overspeed threat at approximately $t = 8\,\mathrm{s}$ and manages to contain the speed below the limit. This, however, comes at the cost of a strong excitation of the tower which the controller then needs to dampen out through rapid pitch motions which are clearly visible between $t = 10\,\mathrm{s}$ and $t = 20\,\mathrm{s}$ in figure 6.10 (b).

It can be concluded that for gust events, as has been shown earlier, preview MPC provides a significant performance increase. Even without preview measurements, MPC has signif-



(a) Rotor Speed

(b) Pitch

(c) Tower Base Bending Moment

(d) Wind Speed

Figure 6.10.: Comparison of MPC and baseline controller during an extreme coherent gust at a wind speed of 13.8 m/s with a return period of 50 years

	max $\left(\frac{\omega}{\omega_0}\right)$	max $\left(\frac{\theta}{\theta_0}\right)$	max $\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.191	8.57	1.24
unconstrained MPC	1.170	8.76	1.20
constrained MPC	1.097	8.76	1.20
constrained MPC + preview	1.028	8.44	1.00

Table 6.2.: Maximum values of rotor speed, pitch angle, and tower base moment during an Extreme Coherent Gust (ECG) starting at $v = 11.5\,\mathrm{m/s}$

icant benefits as with the use of state constraints unnecessary shut-downs can be avoided. Due to its multi-variable nature and optimal control formulation, it also provides improved performance when compared to a simple controller, even if state constraints can be ignored.

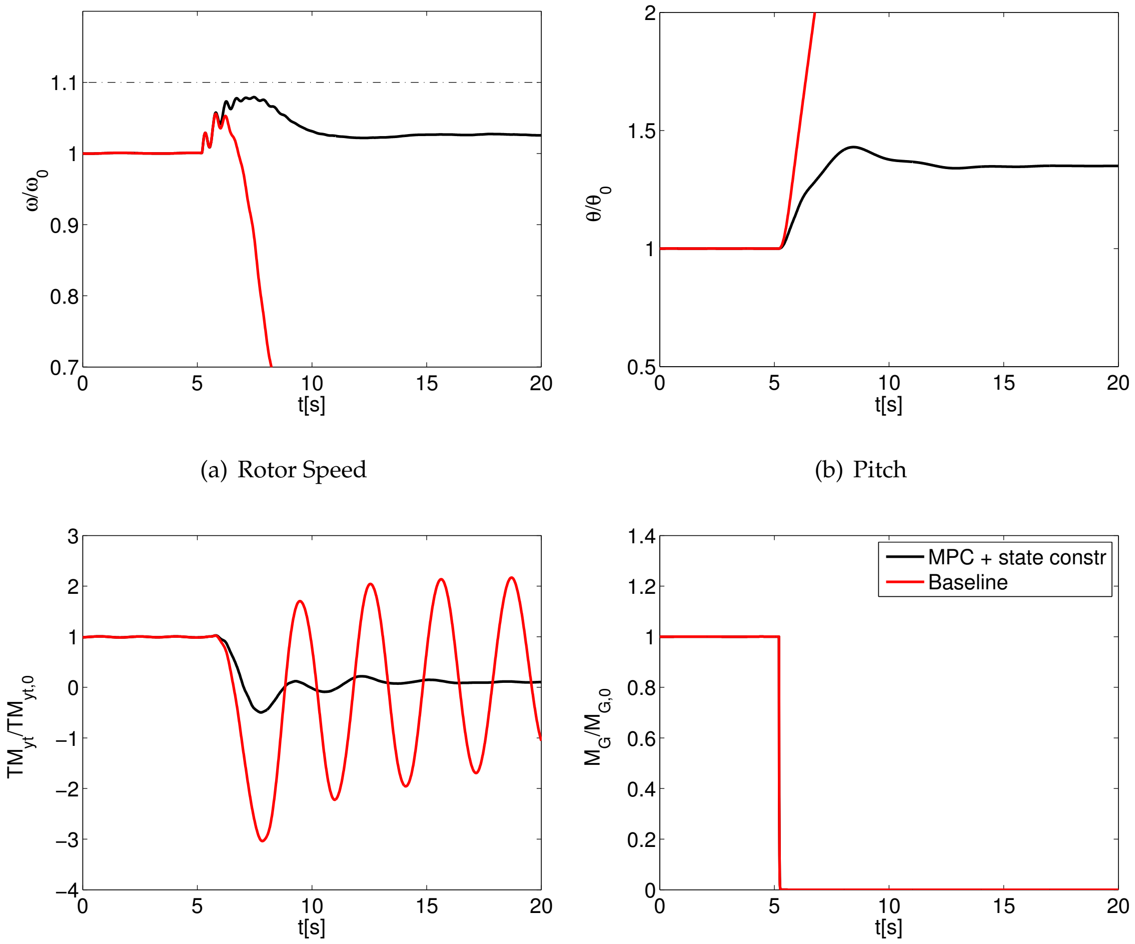
It should be noted that using state constrained MPC cannot guarantee avoiding overspeed faults for all gust situations. A gust can be so strong that no feasible solution exists; i.e., even running at the control constraints cannot prevent the rotor from transgressing the constraint. Additionally, as stated in section 5.3.3, especially during gust situations, it is not possible to include the entire future wind speed in the assumed unmeasured disturbance for the tube calculation. Even if perfect state constraint handling is not possible with the presented tube-based MPC, it will still greatly reduce the overspeed probability and increase overall controller performance, as is also evident from the performance evaluation under turbulent conditions. The values that were chosen here for the unmeasured disturbance resulted in the state-constrained MPC being able to not trigger an overspeed fault even in the event of a gust that has a return period of 50 years so that the residual overspeed risk can be considered as small. For a real turbine, the risk of an overspeed due to fault modes like temporary loss of communication between the turbine controller and the pitch system will be several orders of magnitude larger than the risk linked to the controller not being able to handle a certain gust scenario. Nevertheless, it remains an interesting probabilistic design challenge to calculate the real life overspeed trigger probabilities with the controller design and tuning.

6.4. Fault Conditions

6.4.1. Emergency Stop

As outlined in section 2.2.2, one of the fault scenarios that is commonly considered in the design of wind turbines is load case 5.1 according to the IEC 61400 [39] guidelines: An emergency stop of the turbine with immediate and complete disconnection of the generator. In case of such an event, most turbines will pitch the blades out at a fixed rate to bring the turbine to a stop using some kind of backup power source, e.g., batteries. With the backup power available, it is however also possible to maintain closed-loop control of the turbine.

In order to demonstrate how state-constraint MPC can be used in a shut-down event, a grid loss is simulated once with the MPC and once with a baseline strategy. This baseline controller simply increases the pitch angles of all three blades using a fixed rate of 10 deg/s. In the MPC case, once the grid-loss is detected, the generator torque state, the commanded torque rate bounds, and the weights placed on the power output are set to zero. The controller will now try to maintain rated speed without the torque from the generator and will therefore start increasing the pitch angle immediately. As unconstrained MPC is not able to handle this scenario, it is excluded from the comparison. Further, since the wind speed, at least under IEC design conditions, does not change during this event, preview control does



(a) Rotor Speed

(b) Pitch

(c) Tower Base Bending Moment

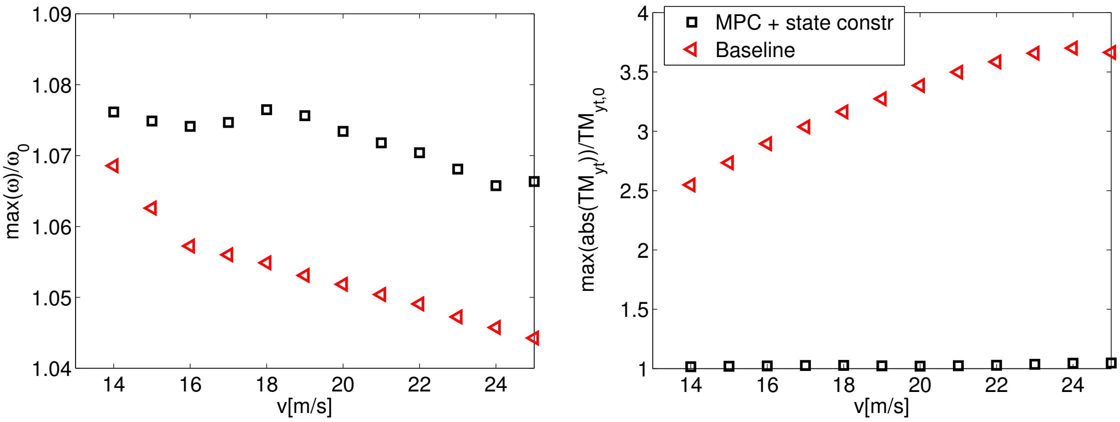
(d) Generator Torque

Figure 6.11.: Comparison of MPC and simple shutdown procedure during grid loss event at a wind speed of 17 m/s

not have an effect, so that preview and non-preview MPC are identical and only a single state-constrained MPC is considered here. The set-point for the rotor speed is held constant during the entire event. Unlike in the simple open-loop scheme, the rotor speed is maintained near its rated value and the turbine is not coming to a stop. Stopping the turbine can however easily be achieved by ramping down the speed set-point after the initial transient has died out.

As can be seen in figure 6.11, using the MPC, the maximum rotor speed during the event increases slightly above the level obtained with the simple shut-down, but remains below the chosen constraint. On the other hand, the fast increase in pitch angle in the simple shutdown case causes strong tower vibration with little damping, while when the MPC is used, the tower base bending moment does not increase above its stationary value at all.

Figure 6.12 shows the maximum rotor speed and tower base bending moment from grid loss simulations over the entire range of full load wind speeds. It can be seen that what was observed in Figure 6.11 is also true at other wind speeds: Using MPC during the shut-down increases the maximum rotor speed slightly, but still manages to hold it below the rotor speed constraint of $\omega_{\max} = 1.1\omega_0$ while the maximum tower load is reduced significantly.



(a) Rotor Speed

(b) Tower Base Bending Moment

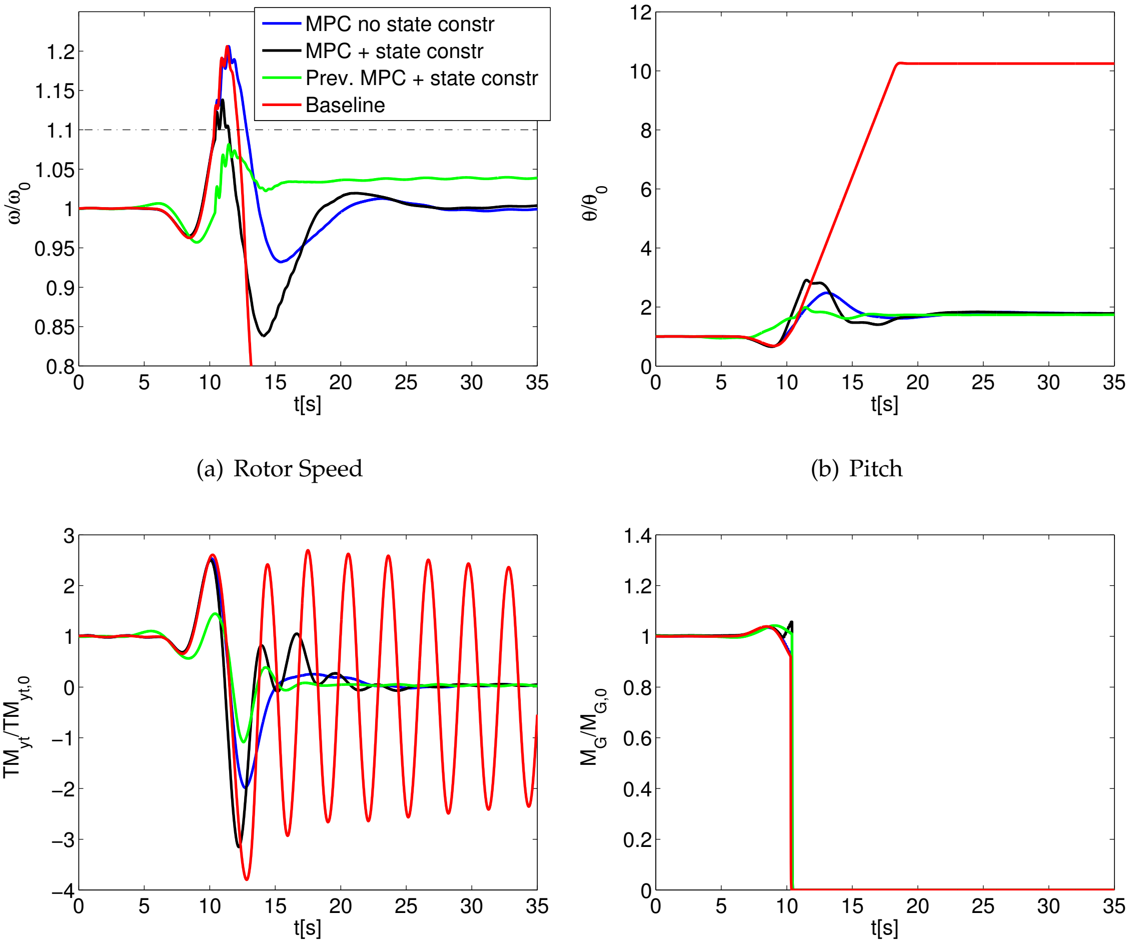
Figure 6.12.: Maximum rotor speed and tower base bending moment for MPC and open-loop pitch ramp shut-down control for wind speeds from 14 m/s to 25 m/s

Not all of this performance gain can be attributed to the use of MPC alone. Most of it is probably due to the use of closed-loop control instead of the simple open-loop scheme. However, the main aim of such a closed-loop shutdown controller is to maintain the rotor speed and possibly additional states, such as the tower deflection, below safety critical levels for a large range of operating conditions and fault scenarios (e.g., different levels of available torque), which is difficult to achieve using non-MPC controller designs.

6.4.2. Loss of Grid Connection during an Extreme Operating Gust

Load case 1.5 simulates the loss of the grid connection and thus counter torque at the generator during an extreme operating gust. To account for the reduced probability of these two events occurring at the same time, a return period of only year and corresponding lower amplitude has to be considered for the EOG instead of the 50 years that were considered in load case 1.6 (see figure 2.10). As part of a full turbine loads assessment, different times of occurrence of the grid loss with respect to the gust would need to be considered. Here, the intent is only to highlight the controller behavior in this scenario and not to compute detailed loads so that the analysis is limited to the case where the grid loss occurs exactly in the instant where the gust reaches its maximum wind speed. The controller reaction to the grid loss is the same as described in the previous section: The torque command is set to zero for all controllers. The baseline controller pitches the blades to the feathered position at a fixed rate while the MPC tries to maintain the rotor speed at its set-point, but is constrained to use only pitch actuation instead of pitch and torque.

Similar to the load case 1.6, the analysis is performed at rated wind speed, 120% of rated wind speed, and the cut-out wind speed. Figure 6.13 shows the resulting time series for the $v = 13.8\,\mathrm{m/s}$ case, while table 6.3 shows the relevant maximum values for all three wind speeds.



(a) Rotor Speed

(b) Pitch

(c) Tower Base Bending Moment

(d) Generator Torque

Figure 6.13.: Comparison of MPC and baseline controller during an Extreme Operating Gust at a wind speed of $13.8\,\mathrm{m/s}$ with a return period of 1 year and occurrence of a grid loss at the wind speed maximum

The first important observation is that at all three wind speeds only the preview controller is able to maintain operation below the overspeed level. Even the constrained MPC fails to

	max $\left(\frac{\omega}{\omega_0}\right)$	max $\left(\frac{\theta}{\theta_0}\right)$	max $\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.207	30.48	2.00
unconstrained MPC	1.183	6.47	1.98
constrained MPC	1.139	8.27	1.93
constrained MPC + preview	1.041	7.24	2.17

(a) $v = 11.5\,\mathrm{m/s}$

	max $\left(\frac{\omega}{\omega_0}\right)$	max $\left(\frac{\theta}{\theta_0}\right)$	max $\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.206	10.26	2.70
unconstrained MPC	1.206	2.48	2.53
constrained MPC	1.138	2.91	2.50
constrained MPC + preview	1.082	1.99	1.45

(b) $v = 13.8\,\mathrm{m/s}$

	max $\left(\frac{\omega}{\omega_0}\right)$	max $\left(\frac{\theta}{\theta_0}\right)$	max $\left(\frac{TM_{yt}}{TM_{yt,0}}\right)$
Baseline	1.223	3.79	4.54
unconstrained MPC	1.320	1.51	3.68
constrained MPC	1.218	1.78	3.98
constrained MPC + preview	1.083	1.57	1.70

(c) $v = 25\,\mathrm{m/s}$

Table 6.3.: Maximum values of rotor speed, pitch angle, and tower base moment during an EOG1 gust with a grid loss event occurring at the highest wind speed

limit the rotor speed as the scenario of a sudden loss of counter torque is not included in the robustification of the state constraint and most of the control margin the controller had has been "used up" by the gust. In figure 6.13 b) the constrained MPC can be clearly seen as being in *Backup Mode* between approximately $t = 10\,\mathrm{s}$ and $t = 12\,\mathrm{s}$ where the rotor speed is above the limit and the pitch is increased at a fixed rate similar to the baseline controller. Unlike the baseline controller, however, as soon as the rotor speed has been reduced below the constraint, closed loop operation is resumed.

Similar to the pure grid loss simulations, it can be seen that maintaining closed loop control and thus also tower damping, significantly helps in bringing the tower to a rest. In all of the MPC cases, the tower base bending moment only has to go through two full cycles before being at rest, while in the baseline case with its open-loop braking procedure, the tower oscillation only has the low damping that the fore-aft mode naturally has at high pitch angles.

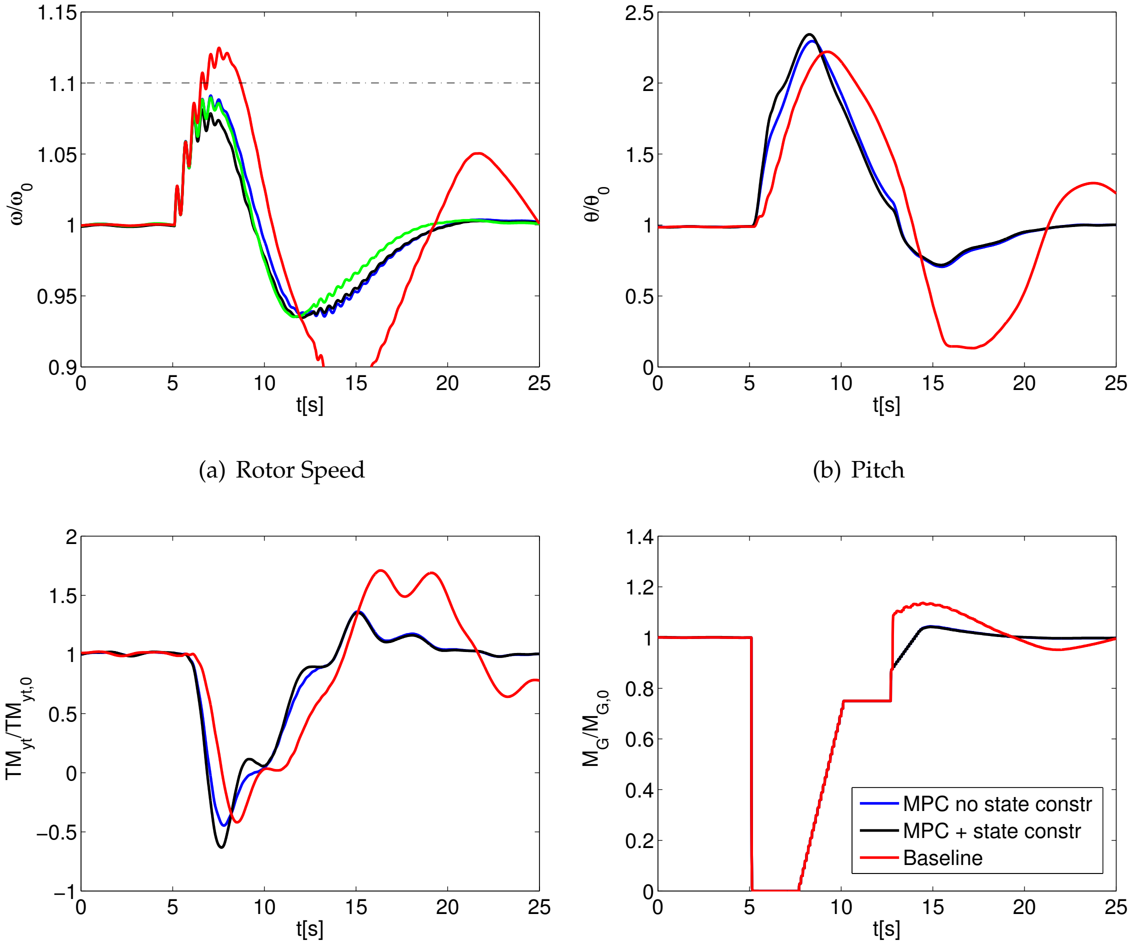
In the preview case, most of the speed excursion coming from the gust can be canceled by the controller by pitching ahead in time so that there is enough controller reserve to handle the loss of counter-torque at $t = 10.3\,\mathrm{s}$ and a potential rotor speed fault is avoided. While at first, prevention of an overspeed might not seem necessary in this event as the turbine is already in fault mode and does not produce any power, there are certain scenarios where it is beneficial to keep the turbine running even though it is not producing power at least for a short period of time. This behavior is analyzed in more detail in the next section. Finally, the sub-optimal behavior of the preview controller for gusts near rated wind speed that was observed in load case 1.6 can be observed here as well: The preview controller reacts very aggressively during the gust at 11.5 m/s causing a significant pitch excursion and tower load.

6.4.3. Fault Ride Through

The loadcases 5.1 and 1.5 that were discussed in the previous sections simulate an emergency stop with disconnection of the generator. This emergency stop can be caused by a number of reasons with one of the most important being a so-called grid loss: Due to a disturbance in the electrical network the turbine is connected to, the turbine needs to be disconnected from the grid. There are, however, also grid disturbances that especially new, modern wind turbines are expected to run through without triggering a fault. A feature which is generally called Fault Ride Through (FRT). The types of grid disturbance a turbine needs to run through vary heavily depending on the specific grid conditions. Running through these faults is mainly a challenge that needs to be dealt with when choosing and designing the system architecture of the electrical system and the control algorithms for the CCU.

Nevertheless, for many potential grid faults and electrical system designs, the turbine might need to run at a reduced level of generator torque or even no generator torque for a short period of time and due to the sudden loss of counter torque, the entire rotor will speed up. At this point, it is no longer only a CCU control issue but a turbine-level control issue as well. As outlined by Ramtharan et al. [90], the sudden loss of counter torque can easily cause the rotor speed to increase above its trip limit, trigger an overspeed fault, and lead to a complete shut-down of the turbine which would be a violation of the FRT requirement. The main goal of the turbine controller during these types of events is therefore to maintain safe operation without triggering a fault even if the generator torque is not or only partially available as an actuator to the turbine control system.

In order to demonstrate the usefulness of MPC during these types of events, the turbine behavior is simulated during a fictitious grid event based on the generic Low Voltage Ride Through (LVRT) curve introduced in figure 2.8: The grid voltage drops suddenly to 0% of nominal. After 2.5 seconds, it starts to increase linearly up to about 75%. At that level, it remains constant for another 2 seconds until it finally jumps back to 100%. For simplicity, this voltage curve is directly translated into available generator torque. It should be noted that the actual electrical behaviour is much more complicated and that in most cases, the generator torque is not proportional to the voltage at the grid side converter. Nevertheless, no grid voltage generally means that no power can be transmitted into the grid and that, if the turbine would continue to generate power, it would, for example, overload the DC link on a full conversion turbine [90]. Therefore, using the voltage curve as a torque limitations is an, although not fully realistic, reasonable choice to model this behaviour.



(a) Rotor Speed

(b) Pitch

(c) Tower Base Bending Moment

(d) Generator Torque

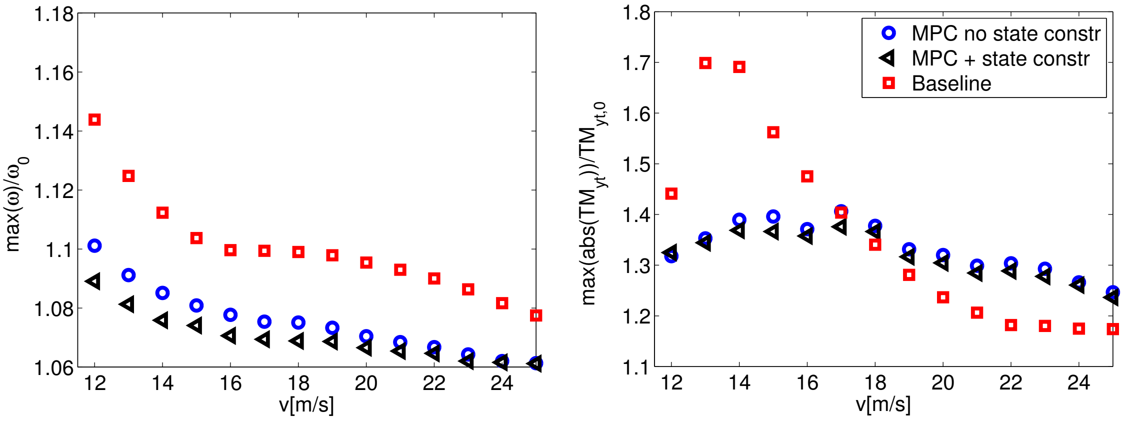
Figure 6.14.: Comparison of MPC and baseline control procedure during a grid fault event at a wind speed of 13 m/s

In the MPC case, as soon as the available torque drops below 100%, the torque rate in the controller is constrained to be zero and the torque command from the MPC is replaced with the maximum available torque. The turbine controller now only uses the pitch angle as an actuator. Torque control authority is essentially ceded to the CCU. The turbine level controller however is "aware" of the current generator torque[5] and includes it in its predictions.

[5] e.g., it is fed that information from the CCU

Similar to the MPC case, in the baseline case, the torque controller is replaced directly by the maximum torque level according to the grid fault definition described above.

Figure 6.14 shows the behavior for the MPC variants and the baseline controller during such an event with a constant wind speed of 13 m/s. Although all three controllers behave fairly similarly, it can be seen that using the baseline controller, the speed increases above the overspeed trigger level and, had the respective supervision been modeled, would cause an overspeed shut-down. There is also a slight difference between the MPC with the rotor speed state constraint and the one without as due to the robustification, the constrained MPC sees an overspeed threat and reacts more aggressively. The performance of the MPCs is better than the baseline controller in this scenario because right at the time of the drop in available torque, they "see" the upcoming rise in generator speed and start to increase the pitch angle while the baseline feedback controller needs to wait for the rotor speed to actually increase. In a way, the MPC in the given formulation includes a feedforward component from the available torque.



(a) Rotor Speed

(b) Tower Base Bending Moment

Figure 6.15.: Maximum rotor speed and tower base bending moment during an FRT event for state constrained and unconstrained MPC variants and baseline control for wind speeds from 13 m/s to 25 m/s

In order to more systematically examine the performance, the same event is simulated at all wind speeds from 12 m/s to 25 m/s and the resulting maximum rotor speed levels are shown in figure 6.15 a). Using the baseline controller, the rotor speed increases above the limit only at the lower wind speeds up to 15 m/s. At higher wind speeds, the sensitivity of the aerodynamic moment to changes in pitch is so high that even the baseline controller manages to reduce the aerodynamic moment fast enough. At high wind speeds, due to the faster pitch actuation, the tower bottom loads actually increase with the use of the MPCs. Compared to other load cases, however, they are still small.

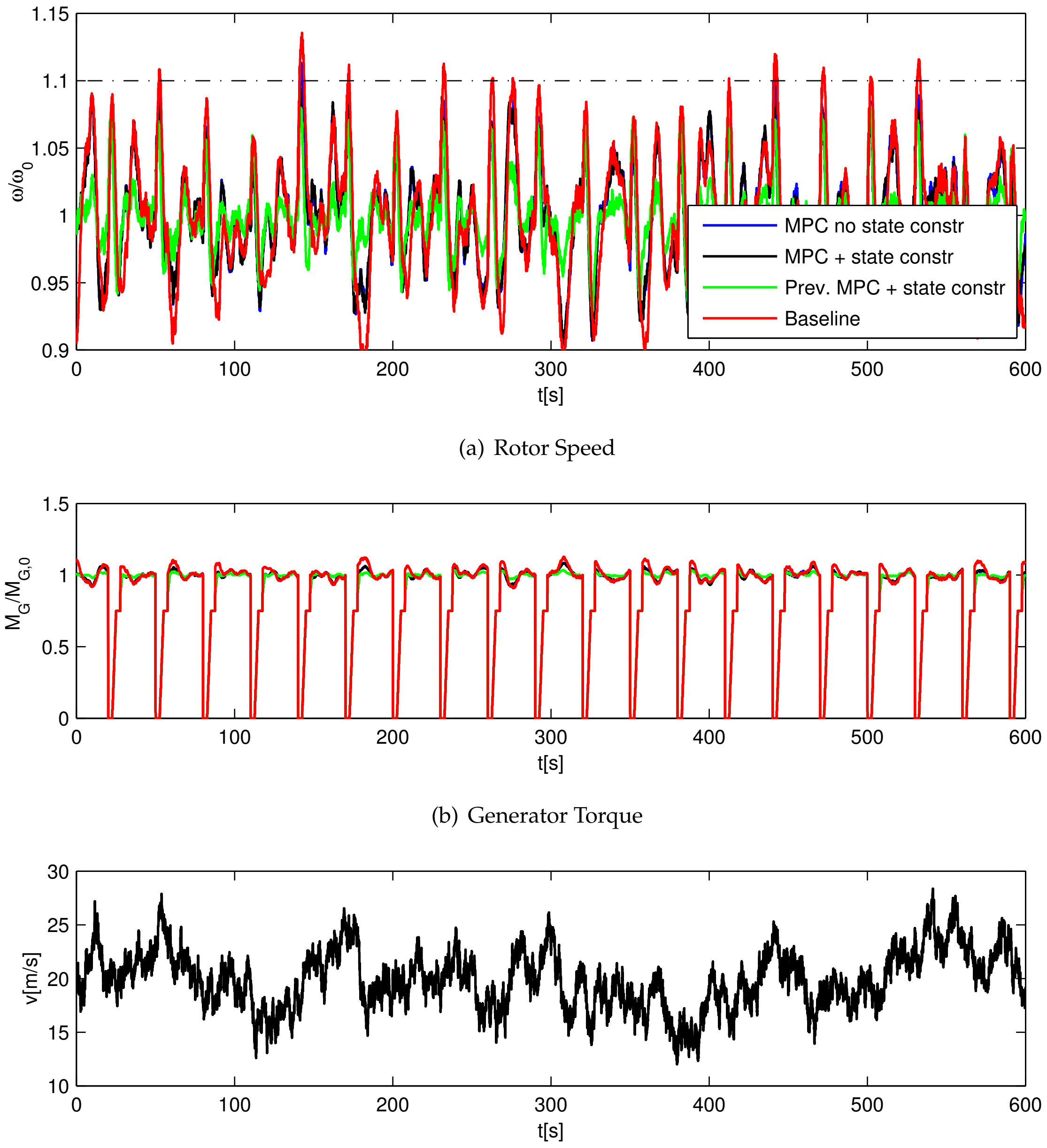
Controller	No. of max rotor speed violations (out of 20)	average maximum rotor speed per event	maximum rotor speed for all events
MPC no state constraint	2	1.071	1.114
MPC + state constraint	0	1.067	1.096
Preview MPC + state constraint	0	1.069	1.082
Baseline	10	1.094	1.136

Table 6.4.: FRT performance of the different control methods in turbulent conditions: Results from a 600 second simulation with a FRT event triggered every 30 seconds (20 events total) and an average wind speed of 20 m/s and a turbulence intensity of 15%.

In reality, the wind speed is almost never constant. So even if, as shown in figure 6.15, there is no threat of causing an overspeed fault during an FRT using the baseline controller at high wind speeds, the FRT could be triggered while a gust is hitting the turbine and the combination of turbulence and FRT could bring the speed above the limit. To show the FRT performance of the MPC while the wind speed is varying, the turbine is simulated in turbulent conditions and the FRT is triggered repeatably. Figure 6.16 shows the resulting time series from a 10 minute simulation at an average wind speed of 20 m/s and a turbulence intensity of 15% where the same grid event that was used previously is triggered every 30 seconds for a total of 20 FRT manoeuvres. Because the wind speed is now varying, MPC with preview is also considered here.

As shown in table 6.4, even though in static conditions the baseline controller did not cause rotor speeds above the overspeed threshold, in turbulent conditions an overspeed fault would have been triggered in 10 out of 20 events. The unconstrained MPC is also no longer capable of maintaining operation below the limit in all cases. Yet, due to the faster reaction it only causes two transgressions. Finally, both the preview and non-preview constrained MPC end up at a maximum rotor speed on a similar level as in the case with static wind conditions showing that adding the constraint helps avoiding some of the variation due to turbulence.

In summary, state-constrained MPC can provide improved Fault Ride Through performance when compared to the baseline controller. Similar to the discussion on the closed-loop braking procedure, it is most likely possible to achieve similar performance also by modifying the classically designed baseline controller. For example, the pitch controller could be made more aggressive whenever a grid fault is detected. However, especially for real-life faults, a large number of potential fault modes and operating conditions need to be considered and treating all these cases using specifically designed controllers can easily become prohibitive. On the other hand, using MPC one common framework for both NPP *and* fault conditions can be used as, via the explicit prediction and control and rate constraints, many of these scenarios can be directly included in the problem formulation. In the end, almost all fault scenarios require the controller to run through an event without



(a) Rotor Speed

(b) Generator Torque

(c) Hub Height Wind Speed

Figure 6.16.: Comparison of MPC and baseline control procedures under turbulent conditions with an average wind speed of 20 m/s and a turbulence intensity of 15% and an occurence of a grid fault every 30 seconds.

leaving the operational limits of the turbine by using limited available control capability in a somehow optimal manner.

7. Conclusion and Outlook

The application of the Model Predictive Control technique to the wind turbine control problem has been studied through one exemplary control design for a full load torque and collective pitch controller. The controller has been evaluated through extensive aero-elastic system simulations for various scenarios similar to those that are typically considered during the design of a wind turbine.

The presented controller, although not formulated and implemented in a speed optimized manner, is already capable of running in real time so that an implementation on an actual turbine does not seem unfeasible. The comparably low computational burden was achieved through the use of a plant model that is specifically tailored to the control problem at hand and which has, especially compared to the models that are generated by some of the aero-elastic simulation tools, only few states. Computational complexity is reduced further through the use of the presented linearization scheme that avoids any type of true nonlinear MPC formulation. One major step that would be required for full implementation is the extension of the MPC to also cover the entire partial load region. However, as has also been shown for the upper partial load operating region, extension of the presented concept to the entire wind turbine operation is straightforward and has only been excluded here as earlier studies [51] indicated that there is little benefit.

It has been shown that MPC is a natural fit for integrating knowledge about the future wind speed, for example from LIDAR measurements, in the control formulation. MPC is especially suitable for preview control, as it allows combining the feedback and the feedforward controller in a single controller so that all the other features of MPC, such as constraint handling and optimal control formulation, are applicable to the combined controller. While the vast majority of all current MPC research for wind turbines focuses on using MPC to design a LIDAR enabled preview controller, here both preview and non-preview control are considered.

Much of the previous research on the application of MPC to wind turbines only included constraints on the control variables, but not on the states. In this study, it was detailed that including state constraints requires robustification of the controller due to the future wind speed acting as a large unknown disturbance on the system. One method for robustification based on the tube-based robust control approach by Rawlings and Mayne [91] has been applied here and was used to include the overspeed limit of the turbine explicitly in the control formulation.

The results showed only small benefits of the non-preview MPC when compared to a baseline controller under normal, turbulent operating conditions. Bigger performance improvements were found especially in the special scenarios such as the gust and fault load cases where very significant extreme load reductions were observed. Especially the use of MPC to control the turbine during shut-down and grid failure events has proven to be an application with significant potential for loads reduction.

The inclusion of an inequality constraint on the rotor speed state using the presented robustification scheme results in a significantly lower risk of the turbine creating an overspeed fault. In fact, across all considered simulations, only the combination of a full grid loss, which is not considered in the robustification scheme, and an extreme gust caused a violation of the overspeed limit if the state constraint was used. It was further shown that the addition of the overspeed constraint helps in separating the overspeed avoidance objective from the general tuning problem of the controller to a certain extent.

If the upwind wind speed information is available, it can be included without any further controller modification. Even with the non-perfect preview wind speed information that was considered in the simulations, the control performance improved drastically compared to the non-preview controllers. Overall, tower extreme and fatigue loads were reduced by around 5% and 30%, respectively, with at the same time reduced pitch activity and almost completly eliminated overspeed risk. Even main shaft and blade loads were reduced, even though they were not explicitly considered. It would however be foolish to attribute all these gains to the use of MPC alone. Simply having the preview information available drives most of the benefit that can be obtained by using the preview MPC. The performance difference between a non-MPC preview controller and a MPC preview controller is likely to be only of the same order of magnitude as the difference observed between the non-preview MPC and the baseline controller.

Many of the examined benefits of MPC do not necessarily require MPC. For example, preview control can be implemented using system inversion approaches, better overspeed handling can be achieved by switching to a dedicated overspeed controller mode as suggested by Kanev and van Engelen [47], and a closed-loop shut-down controller can likely also be designed using classical methods. On the other hand, all of this adds complexity to the controller design. Every new controller module or independent controller for a very specific task will need to be specifically designed and likely have its own tuning parameters. Given the high number of different scenarios, operating points, system constraints this approach of ″add-on″ features can easily become prohibitive with numerous hard-to-evaluate interactions between the various functions.

By using state constraints and appropriately modifying the controller objective, MPC is able to handle not only normal turbine operation but also special scenarios such as extreme gusts or grid-loss cases. This is a first step towards fault tolerant control of wind turbines where a single MPC is used to handle the large variety of conditions and scenarios a wind turbine needs to be able to handle. This type of controller would not only focus on reducing the mechanical loading of the turbine in normal operation, but also on minimizing the loads during special events. By including the probability of certain faults occurring directly in the control design via the selection of state constraints, it can further contribute to the overall turbine reliability by keeping the turbine online longer and reducing unnecessary faults.

The presented approach for robustification of the state constraint essentially treats the unmeasured disturbance as a tuning parameter that is not connected to the true changes in future wind speed. While this has proven to be a powerful method to improve the overspeed performance, it also means the parameter needs to be set by "trial and error" and there is no way of automatically adjusting it based on, e.g., site turbulence or estimator tuning. Particularly the estimator performance has proven to have a big impact on the performance of the presented robustification scheme and careful tuning was required to minimize the wind speed estimation error. For further research in this area, rethinking how the disturbance tubes are generated and developing an approach that is less conservative than the currently used step input approach will likely also have a big impact on the accuracy and usefulness of the presented method.

One of the biggest benefits of MPC is the ability to also address extreme and not only the fatigue loads that most controller design for wind turbines try to reduce. Using the presented methods, extreme loads reductions were achieved especially in the gust and fault load cases. The next logical step would be to include the loads directly in the MPC formulation in the form of another state constraint. The most obvious candidate to be included next is the maximum tower base bending moment. As this load is closely related to the tower top deflection, which is already a state in the current model, this constraint could be included without any update to the plant model. Further, while the collective pitch and torque controller can generally be treated as decoupled from a potential individual pitch controller, integrating them in the MPC framework would allow implementation of a state constraint on the blade tip deflection whenever a blade passes in front of the tower which is a key blade design driving criterion that is relevant especially in some of the gust and fault load cases.

Overall, it was shown that MPC offers numerous benefits to the wind turbine control problem. Some of which were examined in detail in this thesis. Future studies on MPC for wind turbines should therefore not be limited to preview control, but also explore the benefits MPC has to offer even if no preview control is used.

Bibliography

[1] I. Antoniou, H.E. Jørgensen, T. Mikkelsen, T.F. Pedersen, G. Warmbier, and D. Smith. Comparison of wind speed and power curve measurements using a cup anemometer, a LIDAR and a SODAR. In *European Wind Energy Conference and Exhibition*, pages 47–51, 2004.

[2] N. Blet, D. Megias, J. Serrano, and C. De Prada. Nonlinear MPC versus MPC using on-line linearisation - a comparative study. In *IFAC 15th Triennial World Congress*, 2002.

[3] E. A. Bossanyi. The design of closed loop controllers for wind turbines. *Wind Energy*, 3:149–163, July 2000.

[4] E. A. Bossanyi. Wind Turbine Control for Load Reduction. *Wind Energy*, 6:229–244, July 2003.

[5] E. A. Bossanyi. Further load reductions with individual pitch control. *Wind Energy*, 8:481 – 485, July 2005.

[6] E.A. Bossanyi. *GH Bladed user manual*. GH & Partners Ltd, 2003.

[7] C. L. Bottasso, A. Croce, and B. Savini. Performance comparison of control schemes for variable-speed wind turbines. *Journal of Physics: Conference Series*, 75:012079 (9pp), 2007.

[8] B. Boukhezzar and H. Siguerdidjane. Nonlinear control of variable speed wind turbines without wind speed measurement. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 3456–3461, Dec. 2005.

[9] M. Buhl. NWTC design codes (WT_Perf by marshall buhl). `http://wind.nrel.gov/designcodes/simulators/wtperf/`. visited 04-May-2012.

[10] T. Burton, D. Sharpe, and N. Jenkins. *Wind Energy Handbook*. Wiley & Sons, 2001.

[11] D. Carrasco and G. Goodwin. Preview and feedforward in Model Predictive Control: A preliminary robustness analysis. In *18th IFAC World Congress. Milan, Italy*, 2011.

[12] M. Chinchilla, S. Arnaltes, and J.C. Burgos. Control of permanent-magnet generators applied to variable-speed wind-energy systems connected to the grid. *Energy Conversion, IEEE Transactions on*, 21(1):130–135, 2006.

[13] B. Connor, S.N. Iyer, W.E. Leithead, and M.J. Grimble. Control of a horizontal axis wind turbine using h_∞ control. In *Control Applications, 1992., First IEEE Conference on*, 1992.

[14] J.F. Conroy and R. Watson. Low-voltage ride-through of a full converter wind turbine with permanent magnet generator. *Renewable Power Generation, IET*, 1(3):182–189, 2007.

[15] P. Costa, A. Carvalho, and A. Martins. Increasing power wind generation through optimization of the dynamics of control system based on accurate forecasting of the very short-term wind. In *Proc. DEWEK 2008 Bremen*, 2008.

[16] M. Courtney, R. Wagner, and P. Lindelöw. Testing and comparison of LIDARS for profile and turbulence measurements in wind energy. In *IOP Conference Series: Earth and Environmental Science*, volume 1, page 012021. IOP Publishing, 2008.

[17] R. A. DeCarlo. *Linear systems: A state variable approach with numerical implementation*. Prentice-Hall, 1989.

[18] D. Di Ruscio and B. Foss. On state space model based predictive control. In *Proc. of the 5th IFAC Symposium on Dynamics and Control of Process Systems, June 8-10, Corfu, Greece*, pages 304–309, 1998.

[19] F. Dunne, L.Y. Pao, A.D. Wright, B Jonkman, N Kelley, and E Simley. Adding feedforward blade pitch control for load mitigation in wind turbines: Non-causal series expansion, preview control, and optimized FIR filter methods. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[20] I. Erlich, W. Winter, and A. Dittrich. Advanced grid requirements for the integration of wind turbines into the german transmission system. In *Power Engineering Society General Meeting, 2006. IEEE*. IEEE, 2006.

[21] I. Erlich, H. Wrede, and C. Feltes. Dynamic behavior of DFIG-based wind turbines during grid faults. In *Power Conversion Conference-Nagoya, 2007. PCC'07*, pages 1195–1200. IEEE, 2007.

[22] F. Heß and R. King. Holistic control of wind turbines. In *EWEA Annual event 2012*, 2012.

[23] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[24] J. Forrest, D. de la Nuez, and R. Lougee-Heimer. *CLP User Guide*, 2004.

[25] J. Friis, E. Nielsen, J. Bonding, F.D. Adegas, J. Stoustrup, and P.F. Odgaard. Repetitive model predictive approach to individual pitch control of wind turbines. In *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, 2011.

[26] R. Gasch and J. Twele, editors. *Windkraftanlagen: Grundlagen, Entwurf, Planung und Betrieb*. Teubner, 5th edition, 2007.

[27] M. Geyler and P. Caselitz. Individual blade pitch control design for load reduction on large wind turbines. In *Proc. EWEC, Milan*, 2007.

[28] E.G. Gilbert and K.T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *Automatic Control, IEEE Transactions on*, 36(9):1008–1020, 1991.

[29] G. Goodwin, D. Carrasco, D. Mayne, M. Salgado, and M. Seron. Preview and feedforward in Model Predictive Control: Conceptual and design issues. In *18th IFAC World Congress. Milan, Italy*, 2011.

[30] M. J. Grimble. Horizontal axis wind turbine control: Comparison of classical, LQG and h_∞ designs. *Dynamics and Control*, 6(2):143–161, 1996.

[31] K. Hammerum. A fatigue approach to wind turbine control. Master's thesis, Technical University of Denmark, DTU, 2006.

[32] K. Hammerum, P. Brath, and N. K. Poulsen. A fatigue approach to wind turbine control. *Journal of Physics: Conference Series*, 75:012081 (11pp), 2007.

[33] M.M. Hand. Variable-speed wind turbine controller systematic design methodology: A comparison of non-linear and linear model-based designs. Technical Report NREL/TP-500-25540, National Renewable Energy Laboratory, Golden, Colorado, 1999.

[34] M. Harris, M. Hand, and A. Wright. LIDAR for turbine control: March 1, 2005 - november 30, 2005. Technical Report NREL/TP-500-39154, National Renewable Energy Laboratory, Golden, Colorado, 2005.

[35] S. Heier and R. Waddington. *Grid integration of wind energy conversion systems*. Wiley Hoboken, 1998.

[36] L. C. Henriksen. Model predictive control of a wind turbine. Master's thesis, Technical University of Denmark, DTU, 2007.

[37] L. C. Henriksen. *Model Predictive Control of Wind Turbines*. PhD thesis, Technical University of Denmark, 2011.

[38] L.C. Henriksen and N.K. Poulsen. An online re-linearization scheme suited for model predictive and linear quadratic control. Technical Report IMM - 2010-13 - REV. 2, DTU Informatics, Building 321, Kgs. Lyngby, 2010.

[39] IEC. Wind turbines - part 1: Design requirements. Technical Report IEC 61400-1 Ed. 3, IEC, 8 2005.

[40] F. Iov, A.D. Hansen, P.E. Sørensen, and N.A. Cutululis. Mapping of grid faults and grid codes. Technical report, Risø National Laboratory, 2007.

[41] M. Jelavić, N. Perić, I. Petrović, S. Car, and M. Mađerčić. Design of a wind turbine pitch controller for loads and fatigue reduction. In *Proc. EWEC, Milan*, 2007.

[42] K.E. Johnson, L.J. Fingersh, M.J. Balas, and L.Y. Pao. Methods for increasing region 2 power capture on a variable-speed wind turbine. *Journal of Solar Energy Engineering*, 126(4):1092–1100, 2004.

[43] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden Colorado, 2009.

[44] J.M. Jonkman and M.L. Buhl. FAST user's guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, Colorado, 2005.

[45] J.C. Kaimal, J.C. Wyngaard, Y. Izumi, and O.R. Cote. Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, 98(417):563–589, 1972.

[46] B. S. Kallesøe. A low-order model for analysing effects of blade fatigue load control. *Wind Energy*, 9(5):421–436, 2006.

[47] S. Kanev and T. van Engelen. Wind turbine extreme gust control. *Wind energy*, 13(1):18–35, 2009.

[48] N. Kelley and B. Jonkman. NWTC computer-aided engineering tools (TurbSim by Neil Kelley, Bonnie Jonkman). `http://wind.nrel.gov/designcodes/preprocessors/turbsim/`. visited 20-November-2012.

[49] T. Knudsen, P. Andersen, and S. Tøffner-Clausen. Comparing PI and robust pitch controllers on a 400kW wind turbine by full scale tests. In *Proc. Europaen Wind Energy Conference*, 1997.

[50] A. Körber and R. King. Model predictive control for wind turbines. In *EWEC 2010 Warsaw*, 2010.

[51] A. Körber and R. King. Nonlinear model predictive control of wind turbines. In *EWEA Annual event 2011 Brussles*, 2011.

[52] A. Körber and R. King. Combined feedback-feedforward control of wind turbines using state-constrained model predictive control. *Control Systems Technology, IEEE Transactions on*, 21(4):1117–1128, 2013.

[53] J.M. Kos, J.P. Patrick, and K.I. Harner. Multi-mode control system for wind turbines, 1980. US Patent 4,193,005.

[54] A.A. Kumar and K.A. Stol. Scheduled model predictive control of a wind turbine. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, number AIAA-2009-0480, 2009.

[55] M.A. Lackner and G.A.M. van Kuik. The performance of wind turbine smart rotor control approaches during extreme loads. *Journal of Solar Energy Engineering*, 132(1), 2010.

[56] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[57] D.J. Laino. NWTC computer-aided engineering tools (AeroDyn by David J. Laino). `http://wind.nrel.gov/designcodes/simulators/aerodyn/`, 2012. accessed 09-November-2012.

[58] J. Laks and L.Y. Pao. Preview-enabled set-point scheduling for Model Predictive Control of wind turbines. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2012.

[59] J. Laks, L.Y. Pao, E. Simley, A. Wright, N. Kelley, and B. Jonkman. Model predictive control using preview measurements from LIDAR. In *Proc. 49th AIAA Aerospace Sciences Meeting, Orlando, FL*, 2011.

[60] J.H. Laks, F. Dunne, and L.Y. Pao. Feasibility Studies on Disturbance Feedforward Techniques to Improve Load Mitigation Performance. Technical report, National Renewable Energy Laboratory, 2010.

[61] J.H. Laks, L.Y. Pao, and A. Wright. Combined Feed-forward/Feedback Control of Wind Turbines to Reduce Blade Flap Bending Moments. In *Proc. AIAA/ASME Wind Energy Symp*, 2009.

[62] J.G. Leishman and T.S. Beddoes. A semi-empirical model for dynamic stall. *Journal of the American Helicopter Society*, 34:3, 1989.

[63] W.E. Leithead and S. Dominguez. Controller design for the cancellation of the tower fore-aft mode in a wind turbine. In *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC*, 2005.

[64] W.E. Leithead and S. Dominguez. Coordinated control design for wind turbine control systems. In *Proc. EWEC, Athens*, 2006.

[65] W.E. Leithead and S. Dominguez. Size related performance limitations on wind turbine control systems. In *Proc. EWEC, Athens*, 2006.

[66] W.E. Leithead, D.J. Leith, F. Hardan, and H. Markou. Global gain-scheduling control for variable speed wind turbines. In *Proc. EWEC, Nice*, 1999.

[67] F. Lescher, J.-Y. Zhao, and P. Borne. Switching LPV controllers for a variable speed pitch regulated wind turbine. *International Journal of Computers, Communications & Control*, 1(4):73–84, 2006.

[68] F. Lescher, J.Y. Zhao, and P. Borne. Robust gain scheduling controller for pitch regulated variable speed wind turbine. *Studies in Informatics and Control*, 14(4):299–315, 2005.

[69] F. Lescher, J.Y. Zhao, and A. Martinez. Multiobjective h_2/h_∞ control of a pitch regulated wind turbine for mechanical load reduction. In *Proc. EWEC, Athens*, 2006.

[70] J.M. Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[71] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O. Scokaert. Constrained model predictive control: Stability and optimality. *AUTOMATICA*, 36:789–814, 2000.

[72] T. Mikkelsen, K.H. Hansen, N. Angelou, M. Sjöholm, M. Harris, P. Hadley, R. Scullion, G. Ellis, and G. Vives. LIDAR wind speed measurements from a rotating spinner. In *2010 European Wind Energy Conference and Exhibition*, 2010.

[73] T. Mikkelsen, J. Mann, M. Courtney, and M. Sjöholm. Windscanner: 3-D wind and turbulence measurements from three steerable doppler LIDARs. In *IOP Conference Series: Earth and Environmental Science*, volume 1, page 012018. IOP Publishing, 2008.

[74] R. Mittal, K.S. Sandhu, and D.K. Jain. Low voltage ride-through (LVRT) of grid interfaced wind driven PMSG. *ARPN Journal of Engineering and Applied Sciences*, 4(5):73–83, 2009.

[75] P.J. Moriarty and A.C. Hansen. AeroDyn theory manual. Technical Report NREL/EL-500-36881, National Renewable Energy Laboratory, 2005.

[76] J. Morren and S.W.H. De Haan. Ridethrough of wind turbines with doubly-fed induction generator during a voltage dip. *Energy Conversion, IEEE Transactions on*, 20(2):435–441, 2005.

[77] I. Munteanu, A. Bratcu, N.A. Cutululis, and E. Ceanga. A two loop optimal control of flexible drive train variable speed wind power systems. In *16th IFAC World Congress*, 2005.

[78] S.M. Muyeen, R. Takahashi, T. Murata, and J. Tamura. A variable speed wind turbine control strategy to meet wind farm grid code requirements. *Power Systems, IEEE Transactions on*, 25(1):331–340, 2010.

[79] N. Nanayakkara, M. Nakamura, and H. Hatazaki. Predictive control of wind turbines in small power systems at high turbulent wind speeds. *Control Engineering Practice*, 5:1063–1069(7), August 1997.

[80] K.Z. Østergaard. *Robust, Gain-Scheduled Control of Wind Turbines*. PhD thesis, Aalborg University, 2008.

[81] K.Z. Østergaard, P. Brath, and J. Stoustrup. Estimation of effective wind speed. *Journal of Physics: Conference Series*, 75:012082 (9pp), 2007.

[82] K.Z. Østergaard, P. Brath, and J. Stoustrup. Gain-scheduled linear quadratic control of wind turbines operating at high wind speed. In *Control Applications, 2007. CCA 2007. IEEE International Conference on*, 2007.

[83] K.Z. Østergaard, J. Stoustrup, and P. Brath. Rate bounded linear parameter varying control of a wind turbine in full load operation. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 2008.

[84] S. Øye. Flex 4 - simulation of wind turbine dynamics. In *Proc. of the 28th IEA Meeting of Experts "State of the Art of Aeroelastic Codes for Wind Turbine Calculations"*, pages 71–76, 1996.

[85] T.G. van Engelen P. Schaak. Torque control for variable speed wind turbines. In *European Wind Energy Conference*, 2004.

[86] P. Passon, M. Kühn, S. Butterfield, J. Jonkman, T. Camp, and T.J. Larsen. OC3-benchmark exercise of aero-elastic offshore wind turbine codes. In *Journal of Physics: Conference Series*, volume 75, page 012071. IOP Publishing, 2007.

[87] R. Pena, J.C. Clare, and G.M. Asher. Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation. In *IEE Proceedings - Electric Power Applications,*, volume 143, pages 231–241. IET, 1996.

[88] N. K. Poulsen, T. J. Larsen, and M. H. Hansen. Comparison between a PI and LQ-regulation for a 2 MW wind turbine. Technical Report 2320, Risø National Laboratory, Roskilde, Denmark, Feb. 2005.

[89] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[90] G. Ramtharan, C. Newton, and E. Bossanyi. Importance of advanced simulations of electrical transients in wind turbines. *Proc. of the European Wind Energy Energy Confererence (EWEC) Warsaw, Poland*, pages 1–10, 2010.

[91] J.B. Rawlings and D.Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pulications, 2009.

[92] A. Rettenmeier, M. Hofsäß, M. Wächter, M. Kühn, J. Peinke, T. Neumann, H. Mellinghoff, Y. Käsler, S. Rahm, B. Siegmeier, et al. Final results of the joint project "Development of LIDAR wind sensing for the german offshore test site". In *DEWEK 2010*, 2010.

[93] A. Rettenmeier, M. Kühn, M. Wächter, S. Rahm, H. Mellinghoff, B. Siegmeier, and L. Reeder. Development of LIDAR measurements for the german offshore test site. *IOP Conference Series: Earth and Environmental Science*, 1:012063 (6pp), 2008.

[94] G.H. Riahy and M. Abedi. Short term wind speed forecasting for wind turbine applications using linear prediction method. *Renewable energy*, 33(1):35–41, 2008.

[95] I. Rychlik. A new definition of the rainflow cycle counting method. *Int. J. Fatigue*, 9:119–121, 1987.

[96] D. Schlipf, T. Fischer, C.E. Carcangiu, M. Rossetti, and E. Bossanyi. Load analysis of look-ahead collective pitch control using LIDAR. In *Proceedings of the Deutsche Windenergie Konferenz (DEWEK)*, 2010.

[97] D. Schlipf, P. Fleming, F. Haizmann, A. K. Scholbrock, M. Hofsäß, A. Wright, and P. W. Cheng. Field testing of feedforward collective pitch control on the cart2 using a nacelle-based lidar scanner. In *Science of Making Torque from Wind, Oldenburg, Germany*, 2012.

[98] D. Schlipf and M. Kühn. Prospects of a collective pitch control by means of predictive disturbance compensation assisted by wind speed measurements. In *Proc. DEWEK 2008 Bremen*, 2008.

[99] D. Schlipf and M. Kühn. Nonlinear model predictive control of wind turbines using LIDAR. *Wind Energy*, 2012.

[100] D. Schlipf, J.J. Trujillo, V. Basterra, and M. Kühn. Development of a wind turbine LIDAR simulator. In *Proc. EWEC 2009 Marseille*, 2009.

[101] K. Selvam, S. Kanev, J. W. van Wingerden, T. van Engelen, and M. Verhaegen. Feedback-feedforward individual pitch control for wind turbine load reduction. *International Journal of Robust and Nonlinear Control*, 19:72–91, 2009.

[102] E. Simley, L.Y. Pao, R. Frehlich, B. Jonkman, and N. Kelley. Analysis of Wind Speed Measurements using Continuous Wave LIDAR for Wind Turbine Control. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[103] E.D. Sontag and Y. Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

[104] M.D. Spencer, K.A. Stol, C.P. Unsworth, J.E. Cater, and S.E. Norris. Model predictive control of a wind turbine using short-term wind field predictions. *Wind Energy*, 2012.

[105] B.J. Stephens and C.G. Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 52–59. IEEE, 2010.

[106] G. Stewart and F. Borrelli. A model predictive control framework for industrial turbodiesel engine control. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 5704–5711. IEEE, 2008.

[107] K.A. Stol. Disturbance tracking and blade load control of wind turbines in variable-speed operation. In *AIAA/ASME Wind Symposium*, 2003.

[108] REpower Systems. Repower 5M product information. `http://www.repower.de/en/wind-power-solutions/wind-turbines/5m/`, 2012. visited 02-November-2012.

[109] thebackshed.com. Windmill furling. `http://www.thebackshed.com/windmill/Docs/Furling.asp`. visited 01-April-2013.

[110] E.L. Van der Hooft and T.G. Van Engelen. Estimated wind speed feed forward control for wind turbine operation optimisation. In *Proceedings of European Wind Energy Conference in London, UK*, 2004.

[111] T.G. van Engelen and E.L. van der Hooft. Feed forward control of estimated wind speed. Technical Report ECN-C–03-137, ECN Wind Energy, 2003.

[112] H. Vihriälä, R. Perälä, P. Mäkilä, and L. Söderlund. A gearless wind power drive: Part 2: Performance of control system. In *European Wind Energy Conference*, volume 1, pages 1090–1093, 2001.

[113] WAFO-group. *WAFO - A Matlab Toolbox for Analysis of Random Waves and Loads - A Tutorial*. Math. Stat., Center for Math. Sci., Lund Univ., Lund, Sweden, 2000.

[114] U. Wandinger. *Lidar*, chapter Introduction to LIDAR, pages 1–18. Springer, 2005.

[115] N. Wang, K. E. Johnson, and A. D. Wright. FX-RLS-based feedforward control for LIDAR-enabled wind turbine load mitigation. *Control Systems Technology, IEEE Transactions on*, PP(99):1 –11, 2011.

[116] G. Welch and G. Bishop. An introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[117] A.D. Wright and M.J. Balas. Design of state-space-based control algorithms for wind turbine speed regulation. *Journal of Solar Energy Engineering*, 125(4):386–395, 2003.

[118] A.D. Wright, L.J. Fingersh, and K.A. Stol. Progress in implementing and testing state-space controls for the controls advanced research turbine: Preprint. In *24th ASME Wind Energy Symposium, Reno*, 2004.

[119] A.D. Wright and K.A. Stol. Designing and testing contols to mitigate dynamic loads in the controls advanced research turbine: Preprint. In *2008 ASME Wind Energy Symposium, 7-10 January 2008, Reno*. National Renewable Energy Laboratory (NREL), Golden, CO., 2008.

[120] V.M. Zavala, C.D. Laird, and L.T. Biegler. Fast implementations and rigorous models: Can both be accommodated in NMPC? *International Journal of Robust and Nonlinear Control*, 18(8):800–815, 2008.

[121] X. Zhang, L. Zhang, S. Yang, Y. Yu, and R. Cao. Low voltage ride-through technologies in wind turbine generation. *Proceedings of the Chinese Society of Universities for Electric Power System and its Automation*, 2:000, 2008.

Appendix A.

Determination of the Maximal Control Invariant Set

In order to design a Model Predictive Controller that is stable, it is often necessary to constrain the final state so that it lies within an output admissible set. A set of state vectors is admissible if using any vector within the set as an initial condition for a constrained dynamic system will result in all future future states lying within the constraint set. Therefore, if the system is unconstrained every state vector is admissible. The term output admissible is used for the more general case where the constraint set is not only defined on the system states but also on further outputs. The maximal output admissible set is the set of all state vectors that are output admissible. Gilbert and Tan [28] provide a method for the determination of the maximal output admissible set. For a discrete linear state space system without inputs

$$\begin{aligned} x_{k+1} &= \mathbf{A}x_k \\ y_k &= \mathbf{C}x_k \end{aligned} \tag{A.1}$$

which is subjected to a set of s scalar linear output constraints

$$E_i^T y \leq e_i \qquad i = 1...s, \tag{A.2}$$

where E_i is a vector and e_i a scalar, the maximum output admissible set can be calculated using the following algorithm:

1. Set $k = 0$.

2. Solve the following optimization problem:

$$
\begin{aligned}
&J_i^* = \max_x \left(E_i \mathbf{C}\mathbf{A}^{k+1} x \right) \\
&\text{s.t.} \\
&\begin{pmatrix} E_j \mathbf{C} x \\ E_j \mathbf{C}\mathbf{A} x \\ E_j \mathbf{C}\mathbf{A}^2 x \\ \vdots \\ E_j \mathbf{C}\mathbf{A}^k x \end{pmatrix} \leq \begin{pmatrix} e_j \\ e_j \\ e_j \\ \vdots \\ e_j \end{pmatrix} \quad \text{for all } j = 1...s
\end{aligned} \tag{A.3}
$$

   for all values $i = 1...s$. If $J_i^* < e_i$ for all $i = 1...s$ then set $k^* = k$ and exit algorithm.

3. Set $k = k + 1$ and go to step 2.

With k^* determined, the maximum output admissible set $\mathbb{O}_\infty$ is described by s times k^* scalar inequalities:

$$\mathbb{O}_\infty = \left\{ x \in \mathbb{R}^n : E_i \mathbf{C}\mathbf{A}^k x < e_i,\ i = 1...s,\ k = 0...k^* \right\} \tag{A.4}$$

The optimization problems defined in (A.3) are linear optimization problems which can be solved very efficiently even for the large number of inequality constraints that are likely to occur here using simplex based linear program solvers such as the COIN Linear Program code [24].

In many MPC applications, it cannot be assumed that the plant is operated uncontrolled, i.e., without any inputs, and an output admissible set for controlled operation is required. In this case, the system model is:

$$
\begin{aligned}
x_{k+1} &= \mathbf{A} x_k + \mathbf{B} u_k \\
y_k &= \mathbf{C} x_k + \mathbf{D} u_k
\end{aligned} \tag{A.5}
$$

Further, constraints are usually not only placed on the outputs but also on the controlled variables:

$$F_i u \leq f_i \qquad i = 1...s_c \tag{A.6}$$

If a linear controller $u = -\mathbf{K}x$ is used then by defining new system matrices

$$
\begin{aligned}
\mathbf{A}^* &= \mathbf{A} - \mathbf{B}\mathbf{K} \\
\mathbf{C}^* &= \begin{bmatrix} \mathbf{C} - \mathbf{D}\mathbf{K} \\ -\mathbf{K} \end{bmatrix}
\end{aligned} \tag{A.7}
$$

and output vector

$$y^* = \begin{pmatrix} y \\ u \end{pmatrix} \tag{A.8}$$

the system can be brought into the format of (A.1) again

$$\begin{aligned} x_{k+1} &= \mathbf{A}^* x_k \\ y_k^* &= \mathbf{C}^* x_k. \end{aligned} \tag{A.9}$$

As the control inputs u are now part of the modified output vector y^*, the control constraints (A.6) can be treated as output constraints. Using these definitions, it can be seen that the problem of calculating a maximum admissible set for controlled operation is equivalent to calculating the set for the uncontrolled case that was treated previously.

Appendix B.

Design of a Baseline Controller

B.1. Pitch Controller

The pitch controller is designed as a PI controller that sets the pitch command θ_c based on the difference of the rotor speed ω from a reference rotor speed ω_{ref}

$$\theta_c = \left(K_p + \frac{1}{s} K_i \right) n_G (\omega - \omega_{\mathrm{ref}}) \tag{B.1}$$

where K_p and K_i are the gains of the controller. The gearbox ratio n_G is included here as instead of the rotor speed the generator speed is the measured turbine output. During the full load operation of a wind turbine the reference speed will simply be the rated rotor speed $\omega_{\mathrm{ref}} = \omega_0$. However, for the purpose of this analysis, the reference is assumed to be variable.

The gains of the PI controller are selected based on the fundamental drive train equation (4.5):

$$J\dot{\omega} = M_A(\theta, v_e, \omega) - n_G M_G. \tag{B.2}$$

The generator torque, pitch angle, and wind speed can all be seen as inputs to this dynamic system. Now, linearizing the aerodynamic moment gives the following transfer function from pitch angle to rotor speed:

$$\frac{\omega}{\theta}(s) = \frac{\frac{\partial M_A}{\partial \theta}}{Js - \frac{\partial M_A}{\partial \omega}}. \tag{B.3}$$

By combining equations (B.1) and (B.3) a closed-loop transfer function from the speed set-point ω_{ref} to the actual rotor speed is calculated:

$$\frac{\omega}{\omega_{\text{ref}}}(s) = \frac{\frac{\frac{\partial M_A}{\partial \theta}}{J}\left(K_p s + K_i\right) n_G}{s^2 + \underbrace{\frac{-\frac{\partial M_A}{\partial \theta}\left(K_p n_G - \frac{\partial M_A}{\partial \omega}\right)}{J}}_{2\zeta_{cl}\omega_{cl}} s + \underbrace{\frac{-\frac{\partial M_A}{\partial \theta} K_i n_G}{J}}_{\omega_{cl}^2}}. \tag{B.4}$$

Using (B.4) at a given operating point with known $\frac{\partial M_A}{\partial \theta}$ and $\frac{\partial M_A}{\partial \omega}$, the gains for the PI-controller K_p and K_i will follow directly from the choice of closed-loop poles in terms of ω_{cl} and ζ_{cl}.

In the present case, the PI controller is designed at a wind speed of 20 m/s and ω_{cl} and ζ_{cl} are chosen so that the speed control level is roughly similar to the MPC controller resulting in poles characterized by

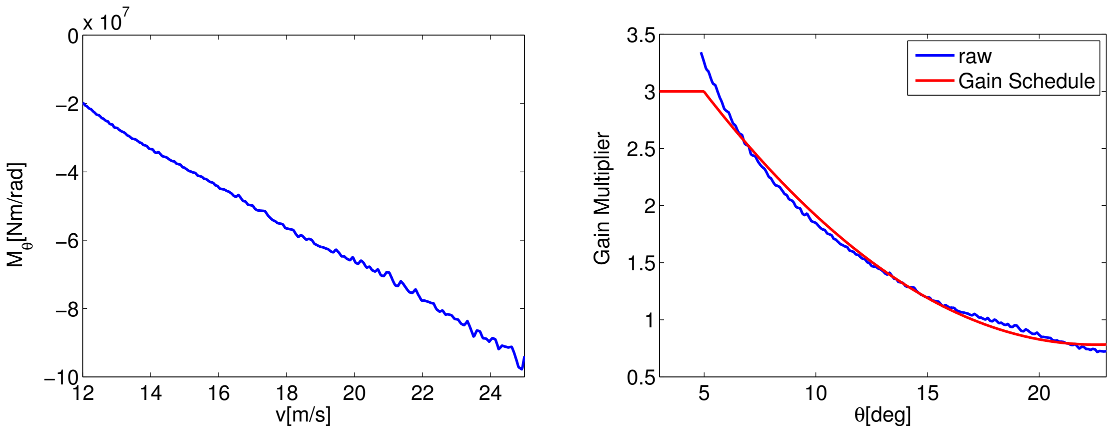
$$\omega_{cl} = 0.48\,\text{rad/s} \qquad \zeta_{cl} = 0.85 \tag{B.5}$$

and corresponding PI gains

$$K_p = 0.003\,\text{rad}/\tfrac{\text{rad}}{\text{s}} \qquad K_i = 0.0015\,\text{rad/rad}. \tag{B.6}$$

The pitch controller further needs to be gain-scheduled to account for the differences in the system dynamics due to the nonlinear aerodynamics. Here, the gains are adjusted so that the terms $K_i \cdot \frac{\partial M_A}{\partial \theta}$ and $K_p \cdot \frac{\partial M_A}{\partial \theta}$ are constant. It can be seen in equation (B.4) that this choice of gain scheduling corresponds to a constant value of ω_{cl}. Since $\frac{\partial M_A}{\partial \omega}$ is small compared to K_p, the damping value ζ_{cl} is also almost constant. Figure B.1 (a) shows the pitch sensitivity $\frac{\partial M_A}{\partial \theta}$ as a function of wind speed for the steady operating curve. As, on a wind turbine without a wind speed estimator, the wind speed is difficult to measure reliably, the gain schedule is however not implemented as a function of wind speed but as a function of the pitch angle instead. Figure B.1 (b) shows the resulting gain multiplier, based on nominal PI gains at a wind speed of 20 m/s, directly calculated from the aerodynamic properties as well as the smoothed curve which is implemented in the controller.

The controller is further implemented with a rate limitation $\max(|\dot{\theta}_c|) \leq 9\,\text{deg/s}$ and a conditional integrator to prevent integrator wind-up.



(a) Pitch Sensitivity

(b) Gain Schedule

Figure B.1.: Stationary torque-to-pitch sensitivity as a function of wind speed and resulting gain schedule for baseline PI pitch controller

B.2. Torque Controller and Switching Conditions

In partial load operation, the turbine speed is controlled via torque actuation, while the pitch angle is fixed at $\theta = 1\,\text{deg}$. Only a torque controller for the upper partial load region (the vertical line in the torque-speed-curve of figure 2.1) is implemented and just like the pitch controller, the torque controller tries to maintain the rated rotor speed. A controller for variable speed control in lower partial load operation is not implemented. Similar to the pitch controller, the torque controller is designed as a PI-controller:

$$M_{G,c} = \left(K_{p,\mathrm{trq}} + \frac{1}{s}K_{i,\mathrm{trq}}\right)(\omega - \omega_{\mathrm{ref}}) \tag{B.7}$$

Here, no gain scheduling is used and the gains are chosen to be

$$K_{p,trq} = 1200\,{}^{\mathrm{Nm}}/_{(\mathrm{rad/s})} \qquad K_{i,trq} = 600\,{}^{\mathrm{Nm}}/_{\mathrm{rad}} \tag{B.8}$$

corresponding roughly to closed-loop poles characterized by

$$\omega_{cl} = 0.4\,{}^{\mathrm{rad}}/_{\mathrm{s}} \qquad \zeta_{cl} = 0.45 \tag{B.9}$$

using similar considerations as for the pitch controller.

Switching Conditions The turbine switches from full load operation, where the pitch is controlling the speed and torque is used to maintain constant power, to partial load operation, where the pitch is held constant and the torque is controlling the speed when the pitch angle is less than or equal to the fine pitch angle of $\theta = 1\,\text{deg}$ and the power is below rated power. The turbine switches from partial load operation to full load if the power is higher than rated power. In both cases, the integrator for the pitch and torque PI-controllers are reset to the current pitch and torque values to allow for a smooth transition between the two operating regions.

B.3. Tower Damper

The tower damper is designed as a PI controller

$$\theta_{c,TFA} = \left(K_{p,TFA} + \frac{1}{s} K_{i,TFA} \right) \ddot{x}_t \tag{B.10}$$

acting directly on the measured tower top acceleration in the longitudinal (along the wind) direction. The chosen gains are

$$K_{p,\text{TFA}} = 0.2\,{}^{\text{degs}^2}/_{\text{m}} \qquad K_{i,\text{TFA}} = 1\,{}^{\text{degs}}/_{\text{m}} \tag{B.11}$$

The choice of an integral gain significantly larger than the proportional gain is motivated by the considerations presented in section 2.1.3, where it was shown that integral control can be seen as adding additional damping to the tower. The small proportional feedback is added in addition to account for the phase delay caused by the pitch actuator. To prevent the tower damper from interacting too much with the regulation of the rotor speed, the output of the tower damper is limited to $|\theta_{c,TFA}| \leq 1\,\text{deg}$ and implemented using conditional integration to prevent integrator wind-up as was also done for the main pitch controller. The tower damper is deactivated whenever the turbine is not in full load operation.