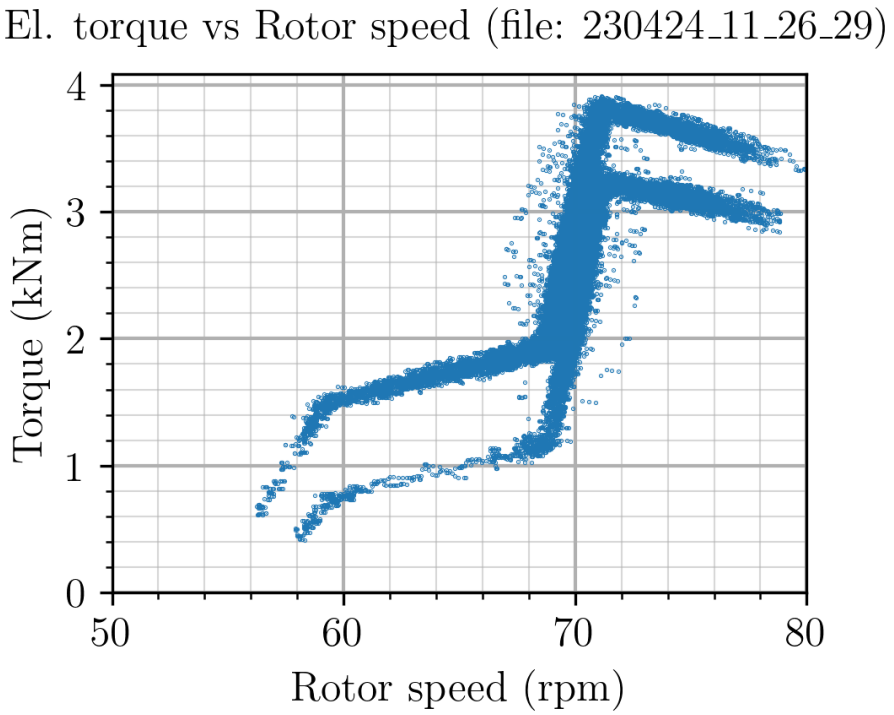
CHALMERS
UNIVERSITY OF TECHNOLOGY



Machine Learning-Based Identification of Control Algorithm for Chalmers Test Wind Turbine

Master's Thesis in Sustainable Energy Systems

KORIN DAMIAN KLEIN

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's Thesis 2025

Machine Learning-Based Identification of Control Algorithm for Chalmers Test Wind Turbine

KORIN KLEIN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Machine Learning-Based Identification of Control Algorithm for Chalmers Test Wind Turbine
KORIN KLEIN

Supervisor: Håkan Johansson, Department of Mechanics and Maritime Sciences
Examiner: Ola Carlson, Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Diagram of electrical torque versus rotor speed, illustrating two distinct clusters of measurement points resulting from the application of different control algorithms.

Typeset in LaTeX
Gothenburg, Sweden 2025

Machine Learning-Based Identification of Control Algorithm for Chalmers Test Wind Turbine
KORIN KLEIN
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Wind turbine control strategies are rarely publicly disclosed, limiting access to detailed information on internal algorithms. This hinders third-party analysis and understanding of turbine operation, which is essential for addressing the challenges and potential of wind energy integration in modern power systems. Therefore, data-driven methods are required to extract insights from available measurement data.

The thesis presents a method to investigate the control behavior of a research wind turbine at Chalmers University of Technology using measured data. The approach is based on graphical analysis of torque versus rotor speed, employing measurements of DC current and voltage at the generator rectifier as well as rotor speed.

To increase the reliability of the method, two data pre-processing strategies are implemented: a binning-based data density filter and a generator torque setpoint reconstruction using supervised machine learning. The effect of various factors on model accuracy is investigated, showing that including the input parameters current, voltage, pitch angle, and rotor speed yields the highest reconstruction precision.

The method also includes features for detecting changes in control strategies and estimating controller-specific configuration parameters. Control change detection is performed using a one-sample hypothesis test, which exhibits a false positive rate of approximately 7.5 % for the cases considered.

Validation results show strong agreement between most identified and actual controller parameters of the Chalmers turbine under both pre-processing approaches indicating high method accuracy. For commercial turbines, reduced accuracy is expected due to simplifying assumptions, such as linearized optimal torque-speed relationships, and the unavailability of torque setpoints for supervised learning. Additional machine learning techniques were explored but did not lead to better performance.

The proposed method offers a data-driven tool to analyze wind turbine control behavior and supports reverse engineering of control strategies where internal system information is unavailable.

Keywords: Wind turbine control, wind turbine characteristics, control algorithm, control parameter, data analysis, data-driven, torque-rotor speed analysis, machine learning, setpoint reconstruction.

Kurzfassung

Die Steuerungsstrategien von Windenergieanlagen sind häufig nicht öffentlich zugänglich, wodurch der Einblick in interne Regelalgorithmen eingeschränkt ist. Dies erschwert eine fundierte externe Analyse des Anlagenverhaltens, die jedoch eine zentrale Voraussetzung dafür ist, die Herausforderungen sowie das Potential der Windenergieintegration in moderne Energiesysteme zu addressieren. Daraus ergibt sich ein Bedarf an datenbasierten Methoden, die aus verfügbaren Messdaten Rückschlüsse auf das Steuerungsverhalten ermöglichen.

Diese Masterarbeit stellt eine Methode zur Untersuchung des Regelungsverhaltens einer Forschungswindenergieanlage an der Technischen Hochschule Chalmers vor. Grundlage der Analyse ist die grafische Auswertung der Beziehung zwischen Drehmoment und Rotordrehzahl, basierend auf Messungen von Gleichstrom, Gleichspannung und Rotordrehzahl.

Zur Erhöhung der Aussagekraft der Methode werden zwei Datenvorverarbeitungsstrategien eingesetzt: ein Filter basierend auf der Punktdichte in Bins sowie eine Rekonstruktion des Sollgeneratordrehmoments mittels überwachtem maschinellen Lernen. Der Einfluss verschiedener Faktoren auf die Modellgenauigkeit wurde untersucht; die Kombination der Eingangsgrößen Strom, Spannung, Pitchwinkel und Rotordrehzahl führte dabei zur höchsten Rekonstruktionspräzision.

Darüber hinaus umfasst die Methode Funktionen zur Erkennung von Änderungen der Steuerungsstrategie sowie zur Abschätzung reglerspezifischer Konfigurationsparameter. Die Erkennung von Regelungsänderungen erfolgt über einen Ein-Stichproben-Hypothesentest mit einer Falsch-Positiv-Rate von etwa 7,5 %.

Validierungsergebnisse zeigen eine hohe Übereinstimmung zwischen den meisten identifizierten und den tatsächlichen Reglerparametern der Chalmers-Turbine, unabhängig von der eingesetzten Vorverarbeitung, was auf eine hohe Genauigkeit der Methode hinweist. Bei kommerziellen Windenergieanlagen ist hingegen mit einer reduzierten Genauigkeit zu rechnen, bedingt durch vereinfachende Annahmen (z.B. linearisierte Drehmoment-Drehzahl-Beziehungen) sowie fehlender Sollgeneratordrehmomentdaten. Weitere getestete Machine-Learning-Ansätze führten nicht zu besseren Ergebnissen.

Die entwickelte Methode stellt ein datenbasiertes Werkzeug zur Analyse des Steuerungsverhaltens von Windturbinen dar. Sie ermöglicht das Reverse Engineering von Steuerungsstrategien bei fehlender Kenntnis der internen Steuerungslogik und trägt damit zur Erhöhung der Transparenz sowie zum besseren Verständnis von Windenergiesystemen bei.

Acknowledgements

Many thanks to my examiner Ola and my supervisor Håkan for giving me the chance to work on this thesis, even though I was new to the topics. I learned a lot of things that will be useful in my future career. I really enjoyed working with you and appreciated your way of being.

I would also like to thank Nowroz, Kristoffer, Arvid, and Philipp for the good atmosphere and enjoyable time working together in the thesis room.

I thank Sarah Barber for hosting the EAWE Data Science Challenge 2024-2025, and Alex Clerc, Osman Cem Yilmaz, Gabriele Calvo, and Peter Matthews for their participation, which also contributed to the development of this work.

I thank Prof. Vogt and the University of Stuttgart for enabling and supporting my participation in the Double Degree program "Energietechnik - Sustainable Energy Systems".

I thank David Frisk, Magnus Gustaver, and Kyriaki Antoniadou-Plytaria for providing this thesis template.

Finally, I thank my family, friends, and all those I met during my studies in Stuttgart, West Lafayette, and Gothenburg for their support and inspiring contribution to this chapter of my life.

Korin Klein, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

a.	angle
Act.	Actual
DC	Direct Current
el., El.	electrical
GTSP	Generator Torque SetPoint
HAWT	Horizontal Axis Wind Turbine
LUT	Look Up Table
met. mast	meteorological mast
Meas.	Measured
ML	Machine Learning
NN	Neural Network
NREL	National Renewable Energy Laboratory
Opt.	Optimal
PI	Proportional-Integral
Pred.	Predicted
sp.	speed
TSR	Tip-Speed Ratio

Nomenclature

Below is the nomenclature of symbols, indices, and non-SI metric units that have been used throughout this thesis.

Symbols

A	m^2	Area
b		Bias term
$b*$		Bias term minimizing the MSE
C	-	Coefficient, Cost function
F	N	Force
H	-	Hypothesis
I	A	DC current
m	-	Number of test examples
MSE	-	Mean squared error
n	rpm	Rotor speed
P	W	Power
R	m	Radius
T	N m	Torque
U	V	DC voltage
v	m/s	Speed
w		Vector of weights
$w*$		Vector of weights minimizing the MSE
x		Example vector
$\hat{x}$		Sample mean
y		True scalar target value
z		Test statistic
$\hat{y}$		Predicted scalar target value

α	-	Significance level
Δs	m	Distance
Δt	s	Time delay
η	-	Efficiency
λ	-	Tip-speed ratio
μ		Value of the population
ρ	kg/m^3	Density
σ		Standard deviation
ω	rad/s	Angular speed

Indices

betz	Betz's law
eff	Effective
el	Electrical
max	Maximum
mec	Mechanical
overall	Overall
opt	Optimum
P	Power
shift	Shifted
tip	Tip of rotor blade
T	Thrust
w	Wind
0	Null, Reference, Standardized
1	Alternative
$1 - \alpha/2$	Quantile of the standard normal distribution

Non-SI metric units

rad	Radians
rpm	Revolutions per minute

Contents

List of Figures

List of Tables

1

Introduction

This first chapter outlines the motivation, objectives, and limitations of the thesis. It begins by presenting the background of the research topic, including its relevance and purpose. Based on this, the second subsection defines the aim of the study. The third subsection delineates the limitations of the work. The fourth subsection highlights the societal, ethical, and ecological implications. Additionally, the participation of components of this thesis in a related challenge is noted. The chapter concludes with a brief overview of the thesis structure.

1.1 Background

Wind energy is among the fastest-growing sources of electrical power and most cost-effective forms of renewable electricity generation. As wind farms are increasingly integrated into the power grid, their influence on grid frequency and stability becomes more substantial. This development presents both challenges and opportunities. One emerging opportunity is the use of wind energy for ancillary services [1, 2].

Currently, the power fed into the grid is compensated by the grid operator based on values provided by the wind farm operator. However, the grid operator lacks transparency regarding how these values are determined. To evaluate feed-in behavior and its influencing factors, more detailed insights are required [1, 2].

The power output of a wind turbine at a given wind speed is governed by an internal control system. Therefore, identifying the underlying control algorithm can yield valuable information about the turbine's behavior and performance. This can be achieved by analyzing measurement data and investigating the relationships between selected input and output parameters [3, 4].

1.2 Aim

The main objective of this thesis is to develop a data-driven method, utilizing machine learning (ML), to identify the control algorithm of a wind turbine and detect transitions between operational states based on measurement data.

1.3 Limitations

The scope of this thesis is subject to several limitations, both technical and methodological.

The measurement system at the Chalmers test wind turbine site includes a substantial but limited number of sensors, which collect data at fixed locations. In some cases, sensor reliability is uncertain. Moreover, the dataset used in this study is restricted in both size and representativeness. Although new data can be acquired, its availability is contingent on wind conditions, resulting in incomplete coverage of certain operational states or wind scenarios. As the turbine is primarily used for research purposes, the dataset includes a wide range of configurations, some of which may not reflect typical operating conditions of commercial turbines. In addition, past experimental interventions have led to variations in measurements under otherwise identical conditions, which reduces the dataset's consistency and generalizability. No controlled experiments were conducted to capture targeted operating conditions, reflecting a realistic scenario more typical of commercial turbine data availability.

Time constraints represent another limiting factor. In particular, while ML provides a wide range of methodological possibilities, this work is limited to a basic subset of approaches. It should also be noted that the author had no prior experience in ML before undertaking this thesis, which further restricts the breadth of exploration in this domain.

These limitations influence the design and implementation of the developed method and may affect its general applicability. The results and code produced should therefore be viewed as an initial step, with potential for further refinement and adaption in future research aimed at broader turbine configurations and more advanced analytical methods.

1.4 Societal, Ethical, and Ecological Aspects

The increasing awareness of climate change has heightened the societal relevance of renewable energy systems, including wind power. As a key contributor to sustainable electricity generation, wind energy plays a central role in the global transition toward low-carbon energy systems.

Researchers in this field bear the responsibility to explore and evaluate the technical and operational potentials of wind energy. The work conducted in this thesis supports these objectives by aiming to increase transparency in turbine behavior and control, thereby contributing to the reliability and integration of wind energy in future power systems.

1.5 Participation in WeDoWind Challenge

The work contributed to the European Academy of Wind Energy (EAWE) and WeDoWind Data Science Challenge 2024-2025, titled "Identifying controller settings from operational data". This challenge was supervised by Sarah Barber (Eastern Switzerland University of Applied Science), with evaluation carried out by Sebastian Mulders (Delft University of Technology), Ola Carlson (Chalmers University of Technology), Håkan Johansson (Chalmers University of Technology), and Dan Zalkind (National Renewable Energy Laboratory, NREL).

The challenge was structured into multiple sub-challenges, some of which aligned the topics addressed in this thesis. Consequently, elements of selected methodological approaches were influenced by insights gained through participation in the challenge. In particular, the ML approach developed in this work draws inspiration from the contributions of challenge participants Alex Clerc, Osman Cem Yilmaz, and Gabriele Calvo from the Renewable Energy Systems (RES) Group.

1.6 Structure of the Report

The report is structured as follows:

Chapter 2 provides the theoretical background by reviewing the state of the art in relevant topics. Furthermore, an overview of wind turbines and their control systems, as well as a presentation of the background of the techniques applied in this work is provided.

Chapter 3 introduces the Chalmers test wind turbine used in this study. The turbine's characteristics and the available measurement data are described to support understanding of its operational behavior.

Chapter 4 outlines the pre-processing strategies necessary to enable reliable application of the main method, including mathematical approaches and a supervised ML technique.

Chapter 5 present the core methods developed in this thesis, including their idea, design, and implementation.

Chapter 6 discusses the results obtained in this study.

Chapter 7 provides the conclusions drawn from the work.

2
Theory

This chapter provides the theoretical foundation necessary to understand the methodology applied in this work. This includes a short review of the state of the art of the topic, an overview of the relevant components, and a description of the key methods and terminology used throughout the thesis.

2.1 State of the Art

The performance and operational states of wind turbines can be assessed through the visualization of measurement data in characteristic diagrams. Depending on the focus of the analysis, power or torque is typically plotted against wind speed or rotor speed. In particular, the relationship between generator torque and rotor speed is a valuable indicator for analyzing the turbine's control algorithm. This thesis develops a method that utilizes this relationship and its graphical representation for identifying control behavior [3, 5].

Data analysis in this context is complicated by noise and inconsistencies in the measurement data. Sensor inaccuracies, processing errors, and the inherent variability of wind conditions can lead to significant deviations from the actual values. To mitigate this, filtering techniques are applied. In addition, both supervised and unsupervised ML methods are employed for data analysis. These approaches offer promising capabilities but also face limitations, particularly in terms of data availability and the complexity of the relationships to be modeled. This thesis applies both a mathematical and a supervised ML approach separately to identify and interpret representative data. Additional ML techniques were explored during the course of this work but were ultimately not implemented due to limitations in data suitability or model performance [5, 6].

One of the primary challenges in characterizing wind turbines is the accurate measurement of wind speed and its association with the corresponding power output. To address this, data-driven models are commonly developed. However, their performance remains limited due to the complexity of the underlying physical processes. In response, ML techniques have been applied to investigate and reproduce nonlinear relationships between input variables such as wind speed and output parameters such as power or torque [5].

This thesis primarily focuses on data analysis and the practical development of a

method; therefore, the literature review comprises only a minor component. Within the scope of the reviewed literature, no prior work was identified that addresses the identification of a control algorithm for a wind turbine.

2.2 Wind Turbine

Wind turbines convert the kinetic energy in the wind into electrical energy. This process involves several stages. Initially, the wind interacts with the turbine blades, generating aerodynamic lift forces. These forces induce rotation in the main shaft, resulting in a net positive torque and the production of mechanical power. This mechanical power is then converted into electrical power by a generator. In standard commercial installations, the generated electricity is fed into the electrical grid [7].

Most modern wind turbines, including the one examined in this work, are designed with a horizontal axis of rotation (HAWT) and typically feature three blades [7].

This section first introduces the fundamental terminology and key parameters associated with wind turbines. It then outlines typical operation types and common nacelle configurations.

2.2.1 Terminology

This subsection lists key terms and parameters used to characterize wind turbines, which are relevant to the methods and analyses conducted in this work.

2.2.1.1 Angular Speed and Rotor Speed

The angular speed ω, expressed in radians per second (rad/s), describes the rotational velocity of the turbine's rotor. Rotor speed n, given in revolutions per minute (rpm), is another commonly used measure in wind turbine analysis. The relationship between these two quantities is defined by Equation 2.1 [8].

$$n = \omega \frac{60}{2\pi} \tag{2.1}$$

2.2.1.2 Power and Torque

Power P and torque T are related through ω by Equation 2.2 [8].

$$P = T\omega \tag{2.2}$$

Unless otherwise stated, references to torque in this work pertain to electrical torque.

Furthermore, electrical power P_{el} can be expressed as the product of direct current (DC) voltage U and DC current I as shown in Equation 2.3 [9].

$$P_{el} = UI \tag{2.3}$$

2.2.1.3 Power Coefficient

The non-dimensional power coefficient C_P is defined as the ratio between the power extracted by the rotor P_mech and the power in the wind P_w, as displayed in Equation 2.4 [10].

$$C_\mathrm{P} = \frac{P_\mathrm{mech}}{P_\mathrm{w}} = \frac{P_\mathrm{mech}}{\frac{1}{2}\rho v_\mathrm{w}^3 A} \tag{2.4}$$

ρ represents the air density, v_w the wind speed, and A the rotor area. Throughout this work, a constant air density of $\rho = 1.225\,\mathrm{kg/m^3}$ is assumed for all calculations.

According to Betz's law, the theoretical maximum power coefficient $C_\mathrm{P,max,betz}$ of an ideal wind turbine is:

$$C_\mathrm{P,max,betz} = 16/27 = 0.5926 \tag{2.5}$$

2.2.1.4 Tip-Speed Ratio

The tip-speed ratio λ, often abbreviated as TSR, represents the ratio of the blade tip speed of the turbine v_tip to the wind speed, as shown in Equation 2.6 [10].

$$\lambda = \frac{v_\mathrm{tip}}{v_\mathrm{w}} = \frac{\omega R}{v_\mathrm{w}} \tag{2.6}$$

The rotor radius of the wind turbine is represented by R. C_P is often plotted against λ to assess turbine performance under varying operation conditions.

2.2.1.5 Wind Turbine Efficiency

The overall wind turbine efficiency η_overall is expressed as the product of C_P and the mechanical efficiency η_mech, which includes the electrical efficiency, as shown in Equation 2.7 [10].

$$\eta_\mathrm{overall} = \eta_\mathrm{mech} C_\mathrm{P} = \frac{P_\mathrm{el}}{P_\mathrm{w}} \tag{2.7}$$

2.2.1.6 Thrust Coefficient

The non-dimensional thrust coefficient C_T characterizes the aerodynamic thrust force F_T acting on a wind turbine and is defined by Equation 2.8 [10].

$$C_\mathrm{T} = \frac{F_\mathrm{T}}{\frac{1}{2}\rho v_\mathrm{w}^2 A} \tag{2.8}$$

2.2.2 Operation Types

Wind turbines can generally be categorized into two types based on their operational characteristics: fixed-speed and variable-speed operation. variable-speed turbines represent the dominant technology in modern wind energy systems [11].

Variable-speed operation allows n to be adjusted to wind speed. This enables the

turbine to maintain an optimal λ_{opt}, thereby maximizing C_{P}. This results in higher efficiency, better power quality, and less mechanical stress on the wind turbine compared to the fixed rotor speed type [11].

The implementation of variable-speed control is enabled by the use of power electronic converters in combination with the generator system. This setup allows for the regulation of generator torque, and thus the drive train torque, leading to reduced fluctuating loads and lower fatigue stress on mechanical components [4, 7, 12].

2.2.3 Nacelle Types

The nacelle houses the mechanical and electrical components that convert the energy captured by the rotor blades into electrical energy. It is located at the top of the tower and contains key subsystems such as the main shaft, gearbox (if present), generator, and auxiliary equipment [13].

Nacelle designs can be broadly divided into two categories: geared and direct-drive configurations. In geared turbines, a multi-stage gearbox is used to increase the rotational speed from the slow-turning rotor to the higher-speed generator, enabling the use of a smaller and lighter generator. In contrast, direct drive turbines eliminate the gearbox, resulting in a one-to-one speed ratio between the rotor and the generator. This design requires a much larger generator to compensate for the lower rotational speed, which increases the overall weight and size of the nacelle [7, 13].

2.3 Control System

The control system plays a crucial role in optimizing power production, as discussed in Subsection 2.2.2. Additionally, it ensures operational safety by limiting power output when critical wind speeds are exceeded, thereby preventing mechanical overload and potential damage to turbine components [11].

Modern wind turbines employ three main control strategies: stall control, pitch control, and active stall control. For variable-speed turbines, which are the focus of this work, pitch control is typically used [11].

2.3.1 Speed and Torque Control

Pitch control enables the adjustment of blade angles, allowing regulation of power output to maintain operation near the generator's nominal capacity, limiting rotor speed for safety reasons, and facilitating emergency shutdowns [11].

In addition to blade pitch, generator torque is actively regulated to influence turbine performance. The generator torque represents the electrical torque opposing the aerodynamic torque exerted by the wind on the rotor. The characteristics of the generator and its associated power electronic converters play a critical role in

determining the behavior of this torque. Specifically, power electronic interfaces allow fast and flexible adjustment of generator torque by controlling the frequency, phase, and amplitude of the generator current [4].

The aerodynamic torque depends on several factors, including λ, rotor geometry (such as blade pitch), wind speed v_{w}, yaw alignment, and induced rotor drag. Since λ directly influences C_{P}, changes in rotor speed or wind speed affect the aerodynamic torque. In variable-speed, pitch-regulated turbines, independent control of aerodynamic and generator torque is possible, enabling effective regulation of rotor speed [4].

2.3.2 Goals of Control

Wind turbine control strategies are typically designed to achieve four main objectives. First, maximize energy production while respecting component speed and load constraints. Second, minimize fatigue damage and avoid extreme structural loads. Third, ensure high power quality for grid integration. Fourth, guarantee safe operation of the turbine under all conditions [3, 4].

2.3.3 Optimal Turbine Operation

The control objectives listed in the previous subsection translate into distinct operational modes based on the wind speed. The rated wind speed marks the threshold between partial and full load operation.

For wind speeds below the rated value, the control systems aims to maximize the energy capture by maintaining λ_{opt}, and thus $C_{\mathrm{P,opt}}$. This is achieved by adjusting the generator torque to follow an optimal torque curve, which is defined by Equations 2.2 and 2.4 together with parameter values maximizing turbine efficiency.

Above rated speed, the primary objective shifts to limit power production and ensure mechanical safety. Here, the blade pitch angle is adjusted to reduce aerodynamic efficiency by spilling wind, while generator torque is used to smooth power output [4].

2.3.4 Deviation from Optimal Operation

Deviations from optimal operation are introduced for safety and regulatory reasons. The most common cases are described below.

At lower rotor speeds, the turbine operates without power production, i.e., without applying electrical load. This mode allows the rotor to accelerate under wind influence during start-up. It facilitates surpassing the tower's eigenfrequency range more quickly, minimizing resonant vibrations. Once a safety margin relative to the eigenfrequency in terms of rotor speed is reached, the system transitions to optimal power production. Additionally, this mode prevents repeated starting and stopping in low wind conditions [3, 8].

At higher rotor speeds, the operation may deviate from the optimal torque curve before reaching rated power. Although this results in a reduced power coefficient, it allows for lower rated rotor speeds. Such deviations can be motivated by blade speed limitations, the need to reduce overspeed risk, and noise regulation requirements.

2.3.5 Characterization of a Control Algorithm

Control algorithms can be characterized in different ways. The approach used in this study is based on a look up table (LUT). A LUT consists of coordinate pairs containing electrical torque and rotor speed values. These points are assumed to be connected by straight lines enabling the assignment of a torque value to each rotor speed by linear interpolation.

2.4 Considered Methods

In this work, basic mathematical methods, elementary ML techniques, and a simple statistical method are applied. The theoretical background for the ML and statistical components is presented in this section.

2.4.1 Machine Learning

ML is applied to problems whose complexity exceeds the capabilities of conventional, hand-crafted programs. An ML system develops by learning from large amounts of data. Its performance is evaluated by testing its predictions against known outcomes. Once trained, the system is expected to predict values for previously unseen data [14].

According to Goodfellow et al., a typical ML algorithm consists of a dataset, a cost function, an optimization procedure, and a parameterized model. The cost function defines how well the model parameters fit the data. During the optimization process, the model parameters are adjusted to minimize this cost, thereby aligning the model as closely as possible with the provided data [14, 15].

This subsection provides a brief introduction to the aspects of ML that are considered in this work.

2.4.1.1 Input

The input to a ML algorithm is a dataset consisting of a collection of examples. Each example represents a data point containing information about specific parameters, known as features. The data points are typically organized into a design matrix, where each row corresponds to an example and each column corresponds to a feature [14].

The dataset is then split into a training subset and a test subset. The proportions of this split are defined manually [14].

2.4.1.2 Learning Types

Various types of learning exist in ML. This work considers two of them, which are briefly introduced below [14].

The most common learning type in ML is **supervised learning**. In this approach, a labeled dataset is used, which includes both the input features (contained in the design matrix) and a predefined target variable. The target variable represents the output the model is expected to predict. It must not be included among the input features [14, 16].

A supervised learning problem with continuous target variables is defined as a regression problem. The objective is to learn a function that maps input features to continuous outputs. Common approaches to solving regression problems include decision tree-based algorithms, such as random forest or histogram-based gradient boosting, as well as neural networks (NN) [14, 16, 17].

A *decision tree* follows a "divide-and-conquer" strategy. The feature space is partitioned using a tree-like structure that spans from a root node to terminal leaf nodes. The root contains the complete dataset, while the leaf nodes represent final decision outcomes. Intermediate nodes form different tree levels, with each node connected via branches to its successors. Each branch corresponds to a decision rule [18].

*NN*s also employ nodes (or neurons) arranged in layers. These layers include an input layer, one or more hidden layers, and an output layer. The input and output layers contain one node per feature and target variable, respectively. Each node is connected to every node in the subsequent layer through weighted links called synapses. In the hidden layers, each node performs mathematical (logical) operations to generate intermediate predictions [19].

The basic procedure of an ML model can be illustrated using the simple example of linear regression. Linear regression can be interpreted as an NN without any hidden layers. As a regression model it predicts a scalar target value $\hat{y} \in \mathbb{R}$ based on an input vector, example $x \in \mathbb{R}^n$. The linear relationship is defined as shown in Equation 2.9.

$$\hat{y} = w^T x + b = \sum_{j=1}^{n} w_j x_j + b \tag{2.9}$$

Here, w is a vector of weights assigned to each feature in x, and b is a bias term. Both w and b are model parameters learned during training [14].

After training, the model is evaluated using the test dataset. The model is applied to the input features of the test examples without considering the corresponding target values. Predictions are generated and subsequently compared to the actual target values to assess performance [14].

A common performance metric is the mean squared error MSE, as shown in Equa-

tion 2.10.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_\mathrm{i} - \hat{y}_\mathrm{i})^2 \tag{2.10}$$

In this equation, m is the number of test examples and y is the true target value [14].

To optimize the model, the parameters $w*$ and $b*$ are determined such that the MSE is minimized. This defines the optimization objective of the training process. The corresponding cost function $C(w, b)$, including the optimization goal, is expressed in Equation 2.11.

$$\min_{w,b} C(w,b) = \min_{w,b} \frac{1}{m}\sum_{i=1}^{m}(y_\mathrm{i} - (w^T x_\mathrm{i} + b))^2 \tag{2.11}$$

Unsupervised learning is used when no target variable is available to train the model. In this case, the objective is to identify structure in the data, such as the underlying probability distribution. Although this approach typically yields less precise predictions compared to supervised learning, it is effective for detecting patterns and structures in large datasets. Common applications include anomaly detection and clustering. Popular algorithms for these tasks include k-means and autoencoders [14].

2.4.2 Statistics

Statistical methods are used to assess whether observed differences in data result from random variation (errors) or from actual changes in the underlying system. Among these, one-sample hypothesis testing is a standard and comparatively robust approach. This method is employed in the present work [20].

In a one-sample hypothesis test, the goal is to determine whether a sample could plausibly originate from a population with a given distribution, typically assumed to be normal. To this end, a null hypothesis H_0 is formulated, as shown in Equation 2.12 [20].

$$H_0 : \mu = \mu_0 \tag{2.12}$$

Here, μ denotes the expected value of the population, and μ_0 is the specified target or reference value. The alternative hypothesis considered in this work is given by Equation 2.13.

$$H_1 : \mu \neq \mu_0. \tag{2.13}$$

To check the hypothesis, the sample mean $\bar{x}$ is used as the test statistic. This is compared to μ_0 via the standardized test statistic z_0, calculated as follows [20].

$$z_0 = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} \tag{2.14}$$

In this expression, σ denotes the population standard deviation, and n is the sample size. The resulting z_0 value is then compared to the critical value of the standard

normal distribution $z_{1-\alpha/2}$ to determine whether the null hypothesis should be rejected. The rejection criterion at a significance level α is defined in Equation 2.15.

$$|z_0| > z_{1-\alpha/2} \tag{2.15}$$

If this condition is satisfied, the null hypothesis is rejected in favor of the alternative hypothesis, indicating that the observed deviation is statistically significant [20].

3

Chalmers Test Wind Turbine Characteristics

This chapter presents the characteristics of the Chalmers test wind turbine. It begins with general information, including theoretical design parameters considered during the turbine's development. Subsequently, the control system is described, based on the theoretical foundations outlined in the previous chapter. Finally, introducing the available measurement dataset, actual operational values and characteristics are identified and analyzed.

3.1 General Information

The current version of the Chalmers test wind turbine, shown in Figure 3.1, is a HAWT with three blades and variable-speed operation. It was commissioned in 2021 on the island of Björkö, off the coast of Gothenburg. The turbine is positioned approximately 30 m from the sea on its western side, while the remaining surroundings are characterized by hilly terrain. It is equipped with a direct-drive generator, so that the rotor speed directly corresponds to the generator speed [21].

Figure 3.1: Picture of the Chalmers test wind turbine [22]

The figure also shows the meteorological mast (met. mast), located approximately 20 m southwest of the turbine.

Table 3.1 summarizes the key technical parameters of the turbine.

Parameter	Value	Unit	Comment
Diameter	15.9	m	
Blade length	7.5	m	Carbon fiber
Hub height	30	m	Wooden tower
Rated power	$\approx$ 30 (45)	kW	Currently limited by control hardware
Rated rotor speed	$\approx$ 71	rpm	Derived from measurement data
Rated wind speed	$\approx$ 8.6	m/s	Derived from measurement data
Overall efficiency	0.85	-	Determined during turbine development

Table 3.1: Key technical parameters of Chalmers test wind turbine [21, 23]

The rated values shown in Table 3.1 are strongly influenced by the specific control algorithm in use. The values listed are extracted from the datasets primarily used in this work. FOGELSTRÖM et al. report slightly different figures in their analysis [8].

Figure 3.2 shows the "theoretical" C_P-λ curves of the turbine for different pitch angles. These curves, like the efficiency value in Table 3.1, are based on measurements and assumptions made during the planning phase of the turbine.

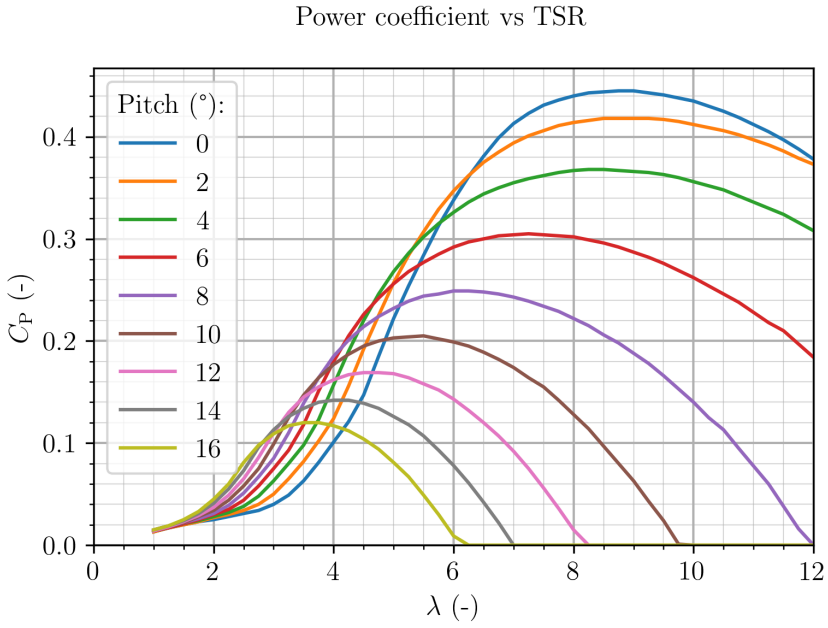


Figure 3.2: Theoretical C_P-λ curve of Chalmers test wind turbine for different pitch angles

From the curves, the characteristic values $\lambda_{\text{opt}} \approx 9$ and $C_{\text{P,max}} \approx 0.42$ can be identified.

The standard pitch angle of the Chalmers test wind turbine during operation below rated speed is 2°. This small deviation from the theoretically optimal pitch angle of 0° serves as a safety measure.

3.2 Control System

The Chalmers test wind turbine is equipped with a pitch-control-based control system. It is implemented using National Instruments Compact Rio hardcore, with the code developed in Labview. The control system is based on the NREL 5 MW offshore reference turbine controller, adapted to the smaller scale and specific characteristics of the Chalmers turbine. Additionally, minor modifications have been incorporated [3, 8, 24].

The section is structured as follows: First, the two primary control loops are introduced. Second, the different control modes implemented in the turbine are described. Finally, the controller parameters used in the current configuration are presented.

3.2.1 Loops

The control system comprises two primary loops: a generator torque control loop and a turbine speed control loop.

The torque control loop determines the generator torque based on turbine rotor speed and the current operating mode. This is implemented in the form of a LUT [8].

The speed controller functions similarly to a traditional proportional-integral (PI) controller but includes additional parameters to enforce a rated speed limit. It adjusts the pitch angle based on the turbine speed. In the Chalmers turbine, all blades are actuated simultaneously, so a single pitch angle is applied to all blades. This loop is only active when the turbine speed exceeds the rated speed [8].

Both control loops use the generator speed, which corresponds to the rotor speed, as their feedback signal. Prior to entering the controller, the signal is exponentially smoothed using a recursive single-pole low-pass filter [3].

3.2.2 Modes

The controller implemented in the Chalmers test wind turbine operates through five distinct modes, also referred to as regions. These are labeled Region 1, 1.5, 2, 2.5, and 3. Each region can be interpreted as a segment of a torque-speed curve, in which a specific generator torque setpoint is assigned to a given rotor speed. This setpoint represents the target value the controller aims to maintain as the operating

conditions change.

The characteristics of each region are described below. Region 1, the "standby mode", corresponds to zero-power production described in Subsection 2.3.4 and is active below a rotor speed of 55 rpm. This threshold is set above the tower resonance speed of approximately 50 rpm. Region 1.5, the "ramp up mode", ensures a smooth transition from Region 1 to Region 2 by gradually increasing the torque for rotor speeds between 55 rpm and 59 rpm. Region 2, the "optimal mode", aims to operate the turbine along the optimal torque curve as described in Subsection 2.3.3. Region 2.5, the "speed limitation mode", introduces an additional increase in torque between the rotor speeds of 68 rpm and 71 rpm. It is required to limit the rotor speed in accordance with considerations presented in Subsection 2.3.4. Finally, Region 3, the "rated power mode", aims to limit the rotor speed while maintaining constant power output through pitch control [3, 24].

Figure 3.3 shows the torque-speed curve of the considered controller for the NREL 5 MW offshore turbine, with the different operating regions indicated. The blue curve represents the actual controller behavior, while the black line illustrates the optimal torque curve.

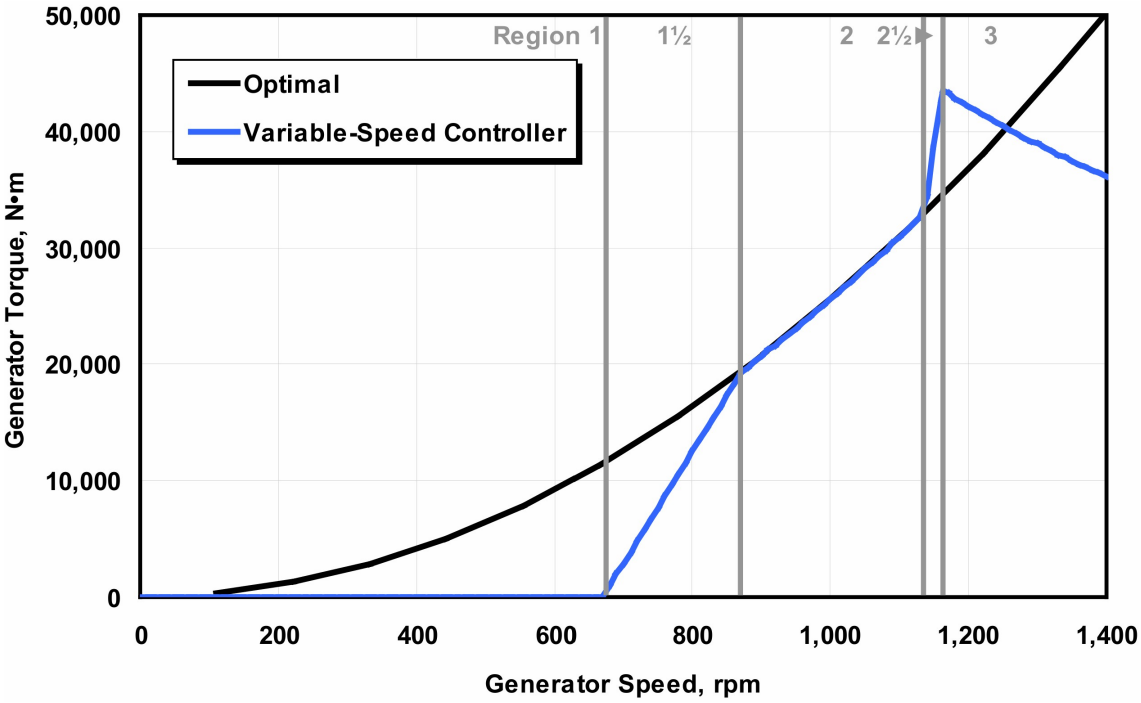


Figure 3.3: Torque-speed response of considered controller for NREL 5 MW offshore turbine [3]

When rotor speeds exceed the operational limits due to strong winds, stop modes are activated to ensure safe shutdown [8].

3.2.3 Parameters

The controller of the Chalmers test wind turbine relies on a set of predefined parameters, listed in Table 3.2. These parameters are entered in the controller settings and determine the operational outcome.

Parameter	Unit	Explanation
Sample_t	s	Sampling time of the controller
CornerFrequency	rad/s	Corner frequency for the generator speed low pass filter
VS_CtInSp	rad/s	Cut-in generator speed
VS_RtPwr	W	Rated generator power
VS_RtGnSp	rad/s	Rated generator speed
VS_Rgn3MP	rad	Pitch angle to stay in Region 3
VS_Slope15	W/(rad/s)	Slope of torque curve in Region 1.5
VS_Rgn2Sp	rad/s	Lower generator speed limit for Region 2
VS_Rgn2K	$W/(rad/s)^2$	Region 2 generator torque constant
VS_TrGnSp	rad/s	Lower generator speed limit for Region 2.5
VS_Slope25	W/(rad/s)	Slope of torque curve in Region 2.5
VS_SySp	rad/s	Generator speed where torque would be zero in Region 2.5
VS_MaxTq	N m	Maximum generator torque
VS_MaxRat	N m/s	Maximum generator torque rate
PC_MaxPit	rad	Maximum allowed pitch angle (Fully feathered position)
PC_MinPit	rad	Minimum allowed pitch angle (Fully fine position)
PC_KP	rad/(rad/s)	Pitch controller Proportional gain
PC_KI	rad/(rad/s)	Pitch controller Integral gain
PC_KK	rad	Pitch angle at which the pitch sensitivity is double the pitch sensitivity at pitch = 0 rad
PC_MaxRat	rad/s	Maximum allowed pitch rate
StoppingPitchSpeed	rad/s	Pitch rate when stopping the turbine
StartingPitchSpeed	rad/s	Pitch rate when starting the turbine
PC_RefSpd	rad/s	Reference generator speed

Table 3.2: Chalmers test wind turbine control parameters [25]

3.3 Analysis of the Measurement Data

This section presents the analysis of measurement data from the Chalmers test wind turbine. The data is recorded over specific periods using sensors installed on the turbine. The analysis serves two main purposes: to characterize the turbine's behavior and to identify parameters most relevant to the subsequent methods. Some of these parameters may need to be derived from others using physical relations.

Visual inspection through plots is used to examine dependencies between variables. This requires evaluating the representativeness of the data, including considerations of system efficiencies and sensor inaccuracies. To reduce measurement noise and clarify variable relationships, filtering and averaging techniques are applied.

Both the data analysis and method development are carried out using the programming language PYTHON. Where necessary, derived parameters are computed using physical formulas from Subsection 2.2.1.

3.3.1 Data Source

The measurement data is stored in tab-separated ".txt" files, with each line corresponding to a discrete time step and each column representing a specific parameter. Data acquisition is conducted at a sampling rate of 20 Hz. Measurements are recorded during defined turbine operating periods and saved in files containing data sequences of up to 30 min.

Table 3.3 lists the parameters and their column positions within the files that are used in this work.

Col no.	Signal short name	Unit	Signal Name
0	Time	s	Time
1	GenTorqSP	N m	Generator Torque SetPoint
9	DCC	A	DC Current from generator rectifier
10	DCV	V	DC Voltage from generator rectifier
11	XTurbSpeed1	rpm	Rotor Speed sensor 1 (low pass filtered)
26	YP	°	Yaw position 0 - (5 * 360) deg
30	PAB1	°	Pitch Angle Blade 1
39	RST2	N m	Rotor Shaft Torque
55	TMBNS	N m	Tower Moment, Base North-South
56	TMBEW	N m	Tower Moment, Base East-West
57	WD30	°	Wind Direction, 30 m height, met. mast
58	WS30	m/s	Wind Speed, 30 m height, met. mast
59	WDN	°	Wind Direction Nacelle
60	WSN	m/s	Wind Speed Nacelle
62	TMBTOR	V/V	Tower Moment Base TORsion
64	WindEst	m/s	Wind Estimation

Table 3.3: Measurement parameters considered in this work [26]

In this work, graphs displaying measurement data over time use a time axis expressed in seconds relative to the start of the respective file. This is achieved by subtracting the first "Time" value in the file from all subsequent "Time" values.

The pitch angle "PAB1" is representative of all blades, as pitch control is applied simultaneously to all three blades on this turbine (collective pitch).

Of the two rotor speed signals recorded, only the low-pass filtered signal "XTurbSpeed1" is used in this work.

The files are structurally labeled using underscores to separate distinct dataset characteristics. When a file is referenced in a graph within this work, the final 15 characters of its filename are included in the graph title (in brackets). These characters indicate the start date and time of the data recording contained in the file, formatted as YYMMDD_hh_mm_ss.

3.3.2 Time Evolution of Fundamental Operating Parameters

This subsection examines the temporal evolution of key operational parameters, such as wind speed, rotor speed, electrical torque, and pitch angle, in order to illustrate the functional behavior of the two control loops described in Section 3.2.

While this joint time-based representation provides an overall understanding of system dynamics, particularly regarding the interaction between control loops, a torque-rotor speed diagram is more appropriate for a detailed assessment of the generator torque control loop.

Figure 3.4 presents the time series of the aforementioned parameters based on measurement data obtained from a selected time interval during which the rated rotor speed is temporarily exceeded. This enables the observation of the dynamic response of both control loops. The wind speed displayed here corresponds to the signal "WindEst".

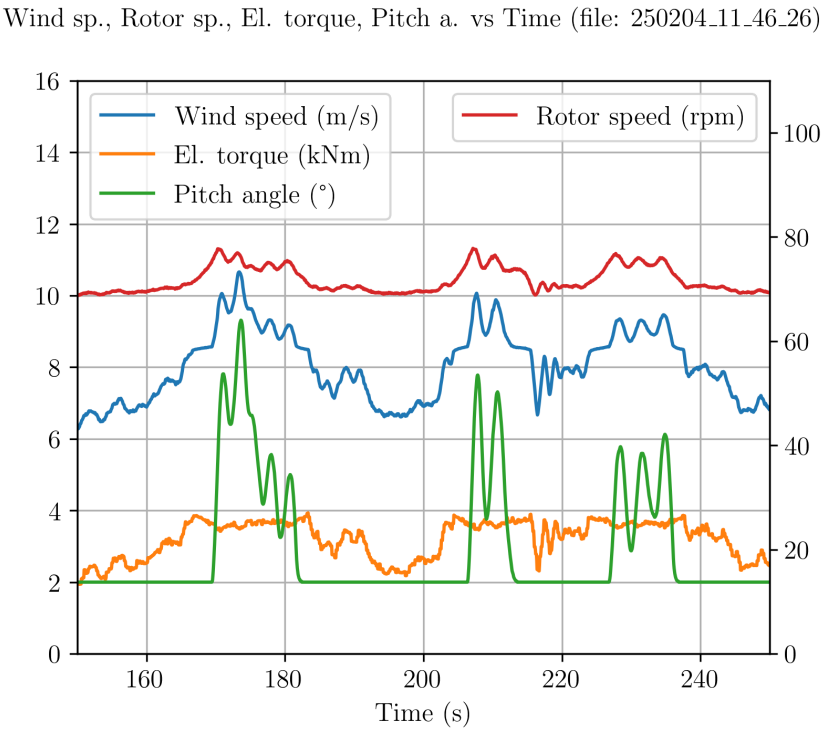


Figure 3.4: Temporal development of wind speed ("WindEst"), rotor speed, electrical torque, and pitch angle

The figure employs a twin y-axis. The left y-axis represents wind speed, electrical torque, and pitch angle. The y-axis on the right hand side shows the values for the rotor speed trajectory.

In the figure, the yellow electrical torque curve follows the reference defined by

the generator torque control loop. As wind speed increases, both rotor speed and electrical torque also increase. This behavior continues until the rated rotor speed is reached, which occurs at approximately 170 s in Figure 3.4. At this point, the pitch control loop is activated, as indicated by the rising pitch angle. This control action limits further increases in rotor speed and electrical torque, despite the continued rise in wind speed.

3.3.3 Wind Speed

This subsection considers the different wind speed signals available in the dataset. At the Chalmers test wind turbine site, two physical 20 Hz wind speed sensors are installed. The signal "WS30" originates from the meteorological mast. The signal "WSN" is measured above the nacelle, i.e., downstream of the rotor (assuming proper alignment of the turbine with the wind), at a height of 32 m.

The "WSN" signal is not consistently reliable in the available data. Several files contain constant values or data that appear implausible. In general, wind speed measurement for wind turbines is challenging due to several factors. For "WS30", one major issue is the high temporal and spatial variability of wind. Wind conditions can change rapidly due to topographical influences and height variations, making it unlikely that the exact same air mass measured at the mast also interacts with the rotor. Moreover, point-based sensors such as "WS30" only capture wind conditions at a single location, whereas the wind acting on the turbine interacts with the entire rotor area, which is subject to spatial gradients. This is especially relevant in the hilly terrain surrounding the test site. Similarly, "WSN" cannot represent the hub wind speed accurately, as the wind reaches the sensor only after being affected by rotor-induced deceleration.

To overcome these limitations, wind speed estimations are derived from turbine operating parameters. The signal "WindEst" represents such an estimated wind speed, averaged over the rotor area. It is calculated using measured values of electrical power output, rotor speed, and pitch angle, along with theoretical and design-based constants such as the mechanical efficiency and $C_{\mathrm{P,max}}$. However, even this estimation is not consistently reliable, as some files also contain constant or otherwise implausible values for "WindEst". Three alternative intuitive approaches to estimate hub-height wind speed based on available measurements are investigated in the following: "WShub_avg", "WShub_wd", and "WShub_cc".

The signal **"WShub_avg"** is calculated by adding the mean difference between "WS30" and "WSN" for the given dataset to the "WSN" signal. This approach assumes that the temporal behavior of "WSN" approximates the wind conditions at the hub and that the wind speed magnitudes at the location of "WS30" and at the hub are approximately equal on average.

"WShub_wd" combines wind direction "WD30" and wind speed "WS30" to account for the wind's propagation delay between the meteorological mast and the

rotor. It is calculated using Equations 3.1, 3.2, and 3.3. First, the effective distance Δs_{eff} between the wind turbine and the mast is determined, based on the actual distance of $\Delta s = 20\,\text{m}$ and the wind direction "WD30". The approach assumes a constant wind speed along a wavefront of at least Δs in length, which also maintains constant between the mast and the hub of the turbine.

$$\Delta s_{\text{eff}} = \Delta s * \cos(\text{"WD30"} - 225^\circ) \tag{3.1}$$

Second, the time delay Δt between the arrival of the wind wave front at the turbine hub and at the mast is calculated using the wind speed measured at the mast "WS30".

$$\Delta t = \Delta s_{\text{eff}} / \text{"WS30"} \tag{3.2}$$

Third, the shifted time is obtained by adjusting the measured "Time" by the calculated Δt.

$$t_{\text{shift}} = \text{"Time"} + \Delta t \tag{3.3}$$

The variable "WShub_wd" is then computed using linear interpolation. The values of "WS30", originally associated with the time points t_{shift}, are interpolated onto the original time axis.

The third approach, **"WShub_cc"**, uses cross correlation between "WS30" and "WSN" to determine a time delay. A sliding window cross-correlation is applied, and a moving average filter is used to reduce high-frequency noise in the signals.

Figure 3.5 illustrates the results of the three approaches for an example file.

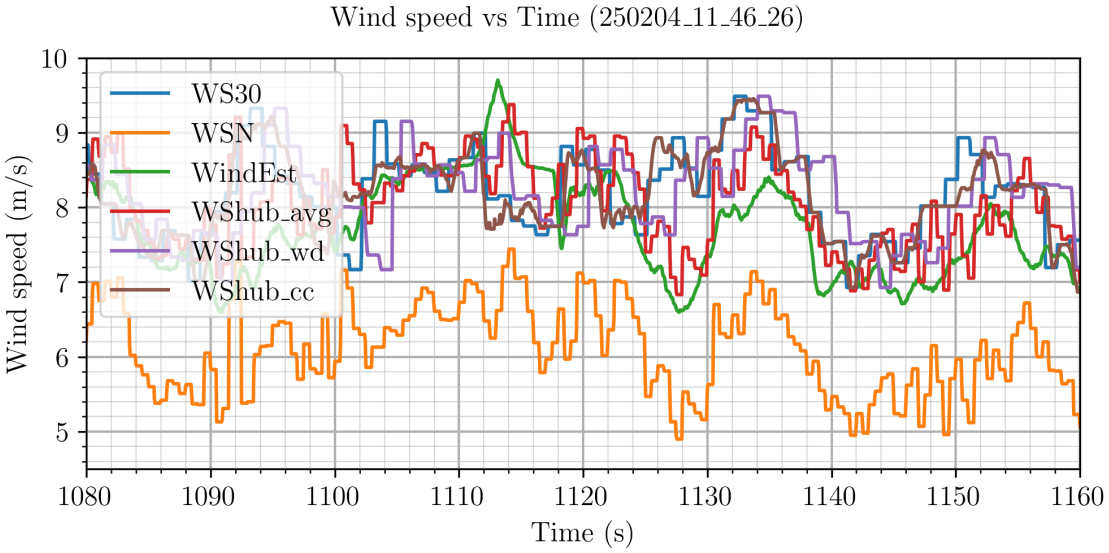


Figure 3.5: Wind speeds over time

The figure shows a short time interval and, consequently, includes only a limited range of wind speeds, wind directions, and associated fluctuations. Therefore, its significance is limited. Since wind speed estimation is not the primary focus in this

work, the purpose of the figure is only to provide a qualitative comparison of the estimation approaches.

In this context, considering the estimated wind speed "WindEst" as the most accurate representation of the hub wind speed and the temporal behavior of "WSN" as a close approximation to that at the hub, "WShub_avg" appears to yield the most plausible results as indicated by the figure.

3.3.4 Power and Thrust Force versus Wind Speed

Figure 3.6 shows the electrical power output and thrust force acting on the turbine as functions of wind speed. The values are represented as mean values obtained through a binned averaging strategy applied to the measurement data.

The electrical power is calculated according to Equation 2.3, using the signals "DCV" and "DCC". The thrust force F_{T} is derived from the signals "TMBNS" and "TMBEW", as well as the tower height $\Delta h_{\mathrm{tower}} = 30\,\mathrm{m}$, using Equation 3.4.

$$F_{\mathrm{T}} = \frac{\sqrt{\text{"TMBNS"}^2 + \text{"TMBEW"}^2}}{\Delta h_{\mathrm{tower}}} \tag{3.4}$$

The estimated wind speed signal "WindEst" is used in this analysis, as it is considered the most representative among the available wind speed signals and estimation approaches for the actual conditions at the hub and exhibits the lowest noise.

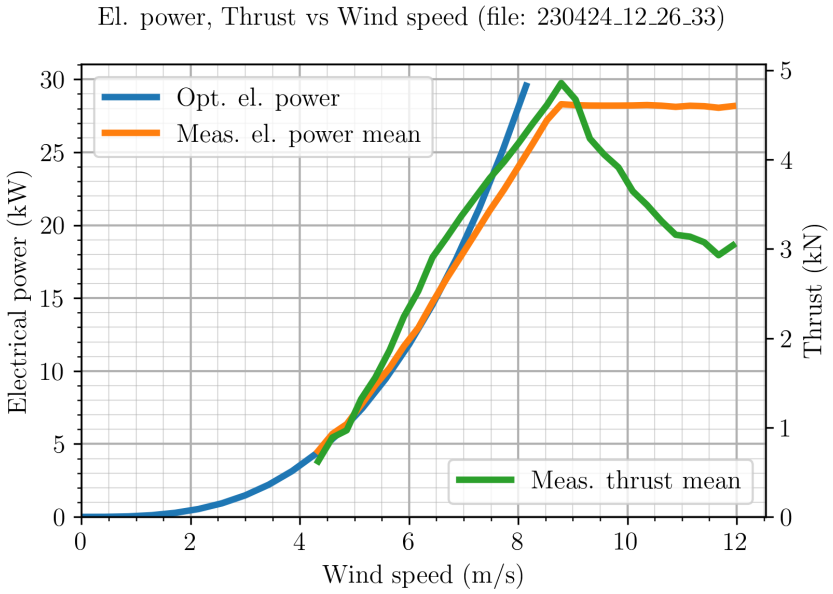


Figure 3.6: Electrical power and thrust vs. wind speed based on "WindEst"

In addition, the curve representing the theoretical optimum electrical power output is shown. It is derived from Equation 2.4 using a maximum power coefficient of $C_{\mathrm{P,max}} = 0.45$, which corresponds to the configuration used in "WindEst".

The figure illustrates the effect of the pitch control system as well as the similarities in the behavior of electrical power and thrust across varying wind speeds. Specifically, it shows that adjustment of the pitch angle above the rated speed significantly reduces the thrust force acting on the turbine, while the electrical power output remains constant.

3.3.5 Mechanical Efficiency

Determining the mechanical efficiency of the turbine components based on the measurement data proves challenging due to the significant noise, which results in widely scattered efficiency values.

The following figures analyze the relationship between mechanical efficiency and rotor speed, electrical power output, and seasonal variation. Figures 3.7 and 3.8 present efficiency data for a winter day in 2025. Figure 3.7 shows efficiency as a function of rotor speed, while Figure 3.8 depicts efficiency as a function of electrical power output.

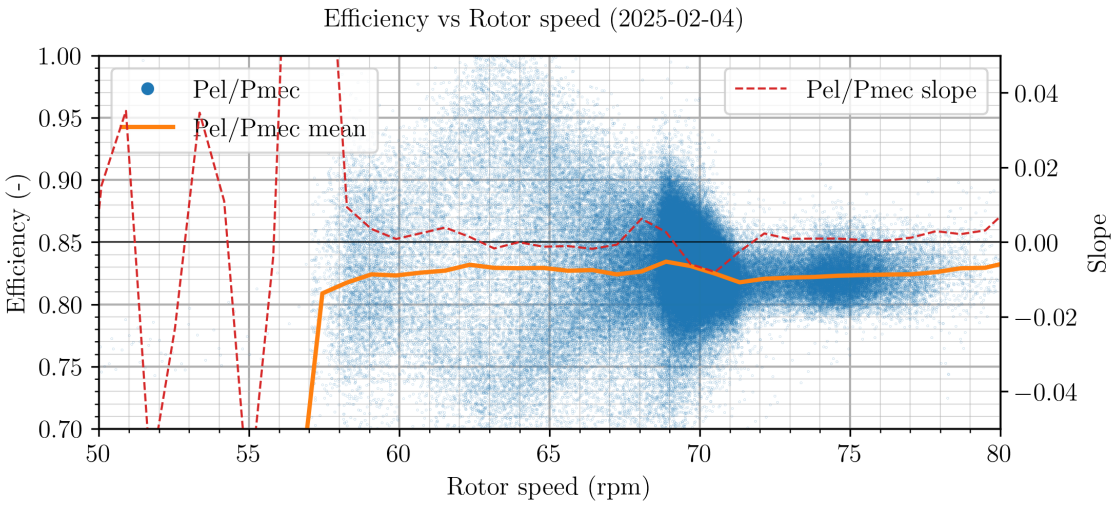


Figure 3.7: Chalmers test wind turbine efficiency vs. rotor speed for a winter day (2025-02-04)

Data points are shown in blue, the average line calculated using one-dimensional bins is shown in orange, and the slope of the average line is indicated by a dashed red line. The gray line represents a zero slope on the right axis and the pre-construction-determined theoretical mechanical efficiency (0.85) on the left axis, as introduced in Section 3.1.

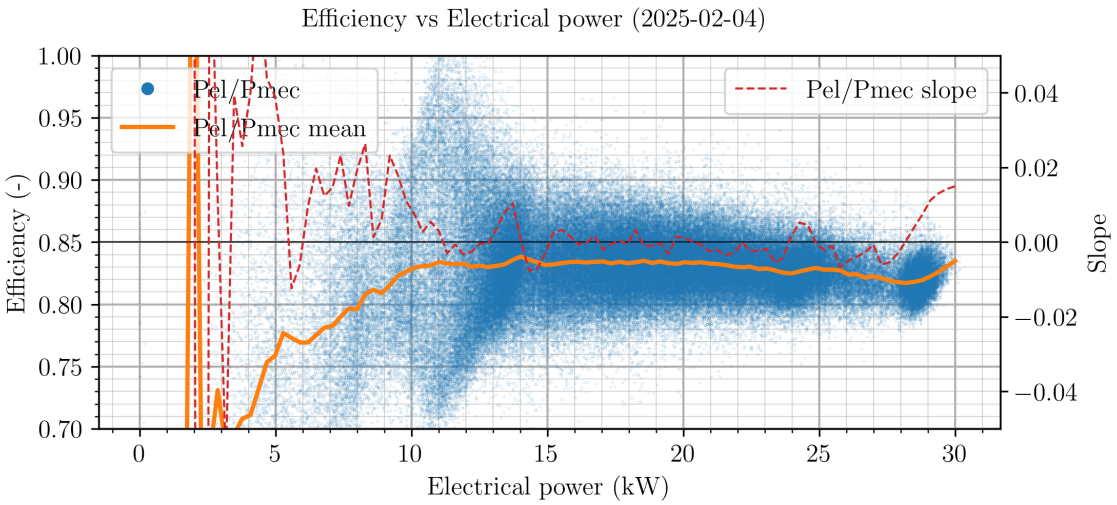


Figure 3.8: Chalmers test wind turbine efficiency vs. electrical power for a winter day (2025-02-04)

Similarly, Figures 3.9 and 3.10 present efficiency data for a summer day in 2024.

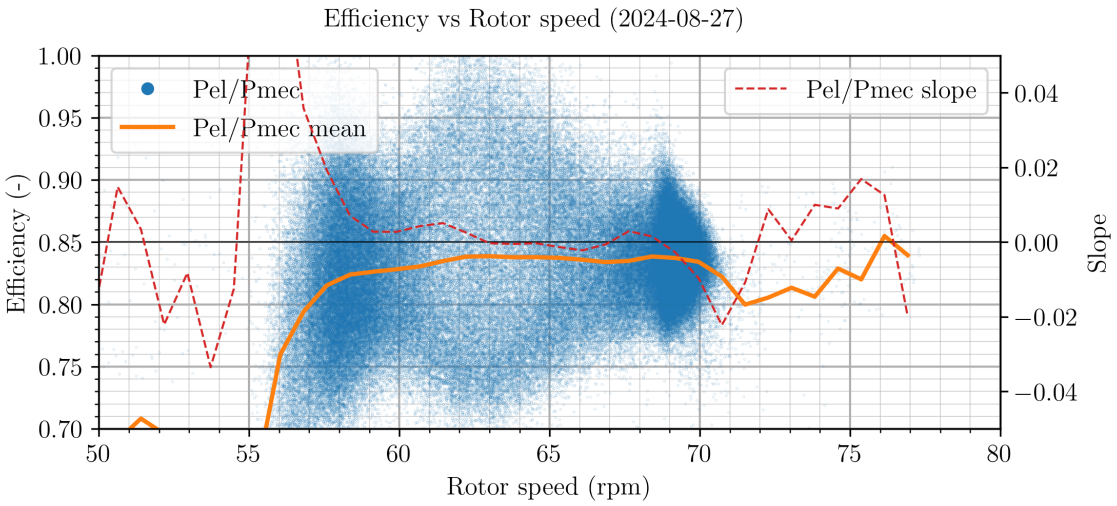


Figure 3.9: Chalmers test wind turbine efficiency vs. rotor speed for a summer day (2024-08-27)

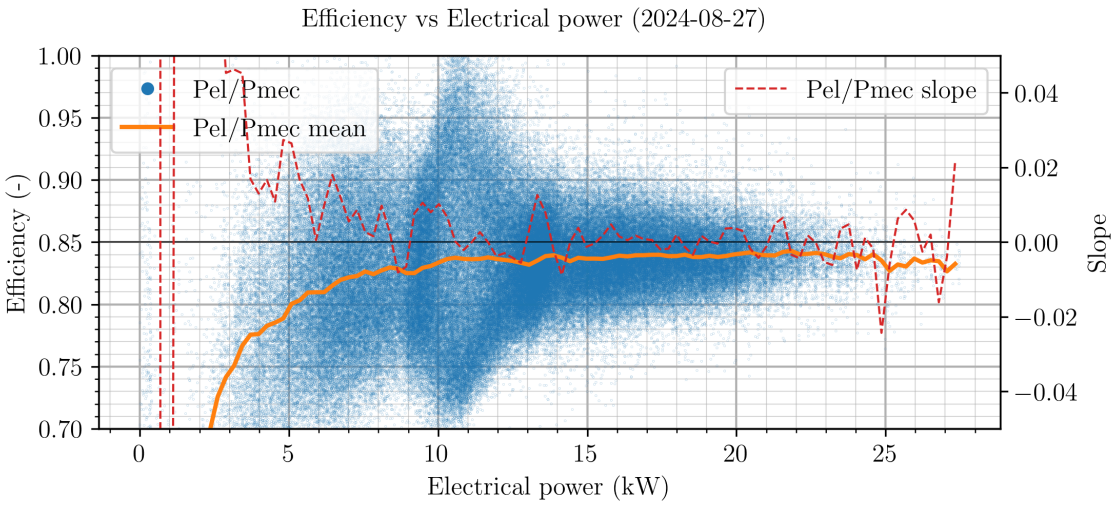


Figure 3.10: Chalmers test wind turbine efficiency vs. electrical power for a summer day (2024-08-27)

The figures indicate that the measured mechanical efficiency deviates slightly from the theoretical pre-construction value of 0.85, with larger deviations observed during winter operation. At lower rotor speeds, the efficiency appears reduced compared to the main operation range. However, this effect might be attributed to the significantly lower data point density in these regions, leading to less representative average values. A clearer trend is observed at lower electrical power levels, where efficiency consistently decreases. Within the main operating range, no clear dependency of mechanical efficiency on rotor speed or electrical power is evident.

3.3.6 Torque and Wind Speed versus Rotor Speed

The relationship between torque and rotor speed provides a clear visualization of the control regions described in Subsection 3.2.2. Consequently, the development of the methods in this work primarily relies on this dependency [3].

The figures presented below consider the electrical torque, calculated from the measured signals "DCC", "DCV", and "XTurbSpeed1" using Equations 2.2 and 2.3. Figure 3.11 displays data from a file assumed to represent a single control setting. In contrast, the dataset in Figure 3.12 is assumed to include at least two control settings. In this figure, the lower-lying cluster of data points results from a control algorithm that spills wind, reducing the power output by approximately 5 kW.

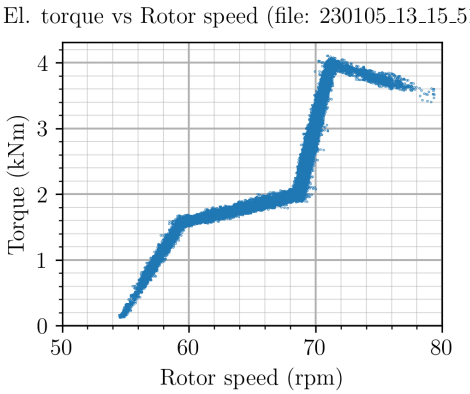


Figure 3.11: Electrical torque vs. rotor speed diagram for one control setting

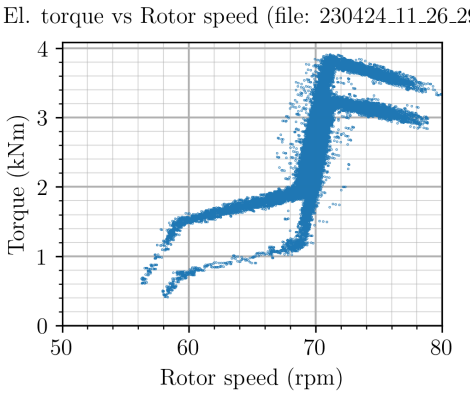


Figure 3.12: Electrical torque vs. rotor speed diagram for two main control settings

Both figures clearly exhibit the control modes described in Subsection 3.2.2. However, since the turbine under consideration is used exclusively for research, many data files contain multiple control settings, as exemplified in Figure 3.12, which may result in increased data noise.

A histogram provides an effective means to display the frequency distribution of measurement points at a specific rotor speed. It provides insight into the statistical distribution of the data and can support the identification of the number of distinct control algorithms active during the measurement period. Figures 3.13 and 3.14 illustrate the distribution of measurement points along the electrical torque axis at a rotor speed of 70.0(1) rpm for the datasets corresponding to Figures 3.11 and 3.12.

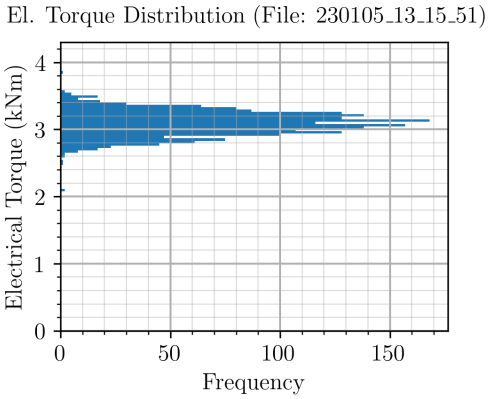


Figure 3.13: Distribution of measurement points at 70.0(1) rpm for one control setting

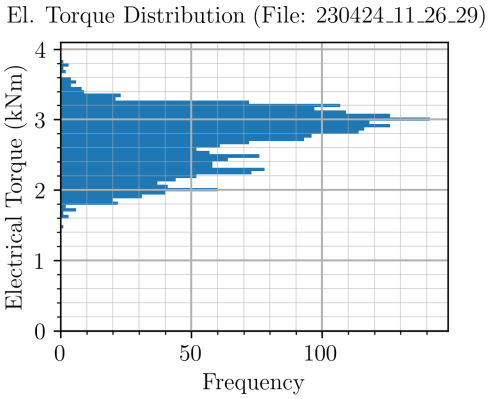


Figure 3.14: Distribution of measurement points at 70.0(1) rpm for two main control settings

The distributions are based on 50 bins. The frequency on the x-axis represents the number of measurement points occurring around a given torque value on the y-axis.

In Figure 3.13, the data points cluster around 3.1 kN m, supporting the assumption of a single dominant control algorithm in operation.

Figure 3.14 shows a primary torque around 3 kN m and a secondary cluster near 2.25 kN m at 70.0(1) rpm. Compared to Figure 3.13, the distribution appears more evenly spread, indicating either the presence of additional control algorithms or a higher level of measurement noise.

Figures 3.15 and 3.16 depict the corresponding relationships between wind speed and rotor speed for the same datasets.

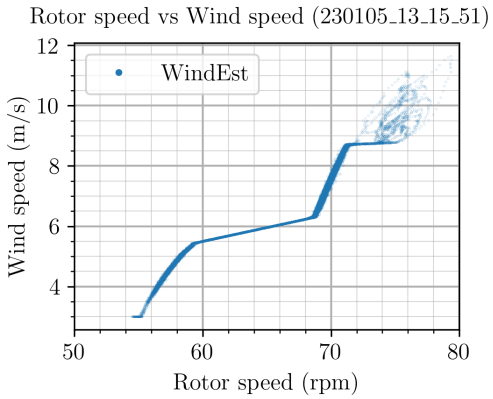


Figure 3.15: Wind speed vs. rotor speed diagram for one control setting

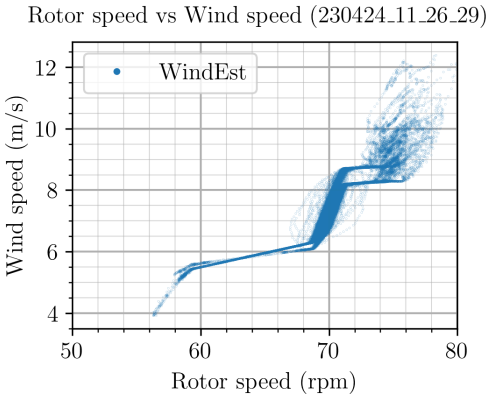


Figure 3.16: Wind speed vs. rotor speed diagram for two main control settings

Similar to the torque versus rotor speed plots, transitions between control regions are evident. Additionally, the scattered data observed at higher rotor speed and torque values is assumed to reflect pitch control intervention.

3.3.7 Power Coefficient versus Tip-Speed Ratio

This section aims to derive the theoretical C_P-λ curve, displayed in Figure 3.2, from measurement data. However, this proves to be challenging.

The main issues are attributed to the lack of accurate wind speed values at the correct time steps, files containing implausible values for "WSN" and "WindEst", and time delays between wind speed changes and corresponding power responses.

Figure 3.17 shows the C_P-λ relation based on multiple measurement files for the wind speeds described in Subsection 3.3.3, at a pitch angle of 2°. Due to the high noise levels, only binned average values are shown. For comparison, the theoretical curve and a model-based curve from the Chalmers test wind turbine simulation in ASHES are also plotted.

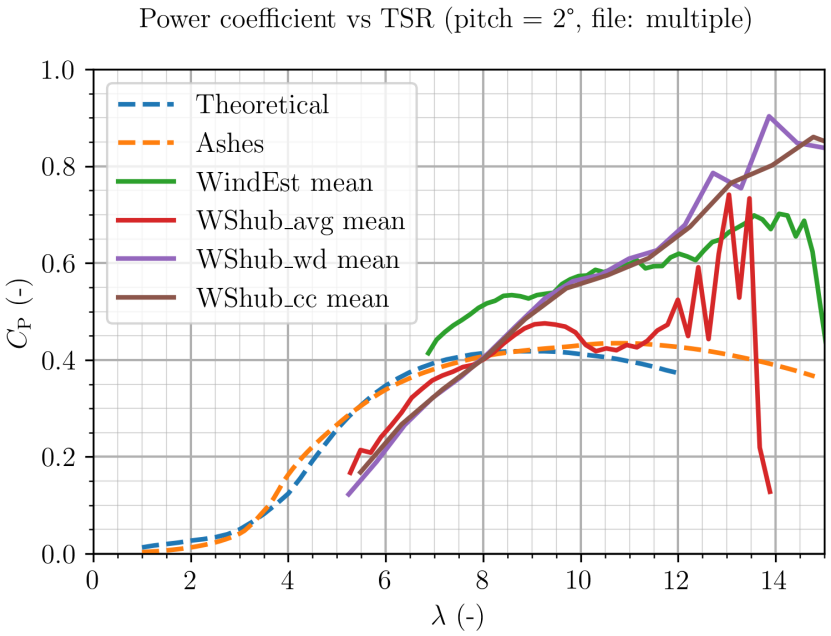


Figure 3.17: C_P vs. λ for the considered wind speed estimations from files with seemingly reasonable wind speed values

Among the evaluated wind speed estimates, the curve based on "WShub_avg" most closely follows the expected theoretical trend.

Figure 3.18 displays the C_P-λ curve derived from a single file, which is the same dataset as in Figure 3.6.

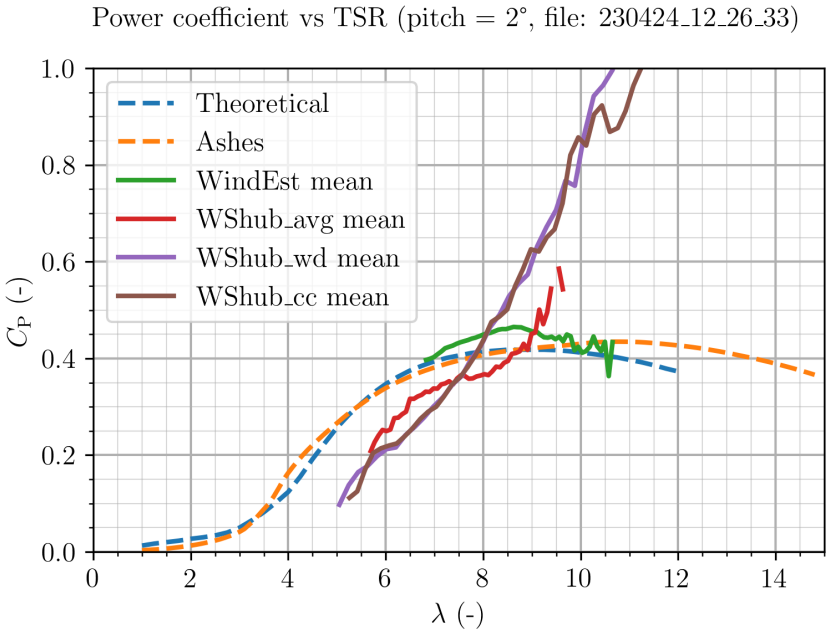


Figure 3.18: C_P vs. λ for the considered wind speed estimations from a single file

In this case, "WindEst" yields the curve that aligns most closely with the theoretical expectations.

Strategies to improve the empirical fit to the theoretical curves were considered. One approach involved filtering the data to include only periods when the wind direction was from the west. This consideration originates from the location of the turbine, described in Section 3.1, and is based on the assumption that wind coming from the sea is less turbulent than wind coming over hilly terrain. This was expected to improve the correspondence between the "WS30" signal and the wind speed at the hub. Another approach restricted the analysis to the rotor speed range associated with the "optimal mode", described in Subsection 3.2.2. However, neither filtering method, whether applied individually or in combination, led to a significant improvement in the datasets considered.

3.3.8 Thrust Coefficient versus Tip-Speed Ratio

The same limitations encountered in deriving the C_P-λ curve apply to the C_T-λ relation. Figure 3.19 presents C_T as a function of λ for the same datasets used in Figure 3.17.

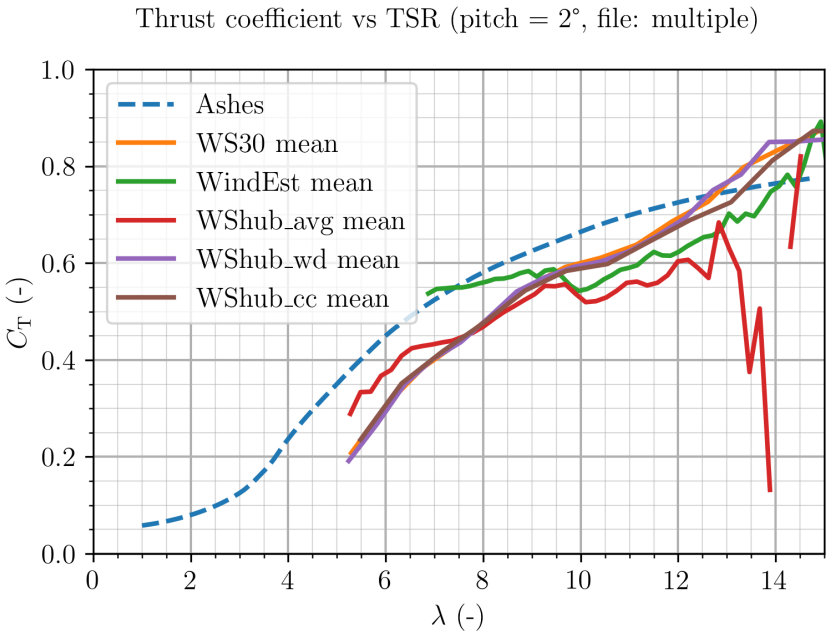


Figure 3.19: C_T vs. λ for the considered wind speed estimations from files with seemingly reasonable wind speed values

Compared to the C_P-λ curve, the C_T-λ curves show less variations across different wind speeds. However, all curves lie consistently below the theoretical reference obtained from the Ashes model.

Analogously, Figure 3.20 shows the C_T-λ curve based on a single dataset, which is the same file used in Figure 3.18.

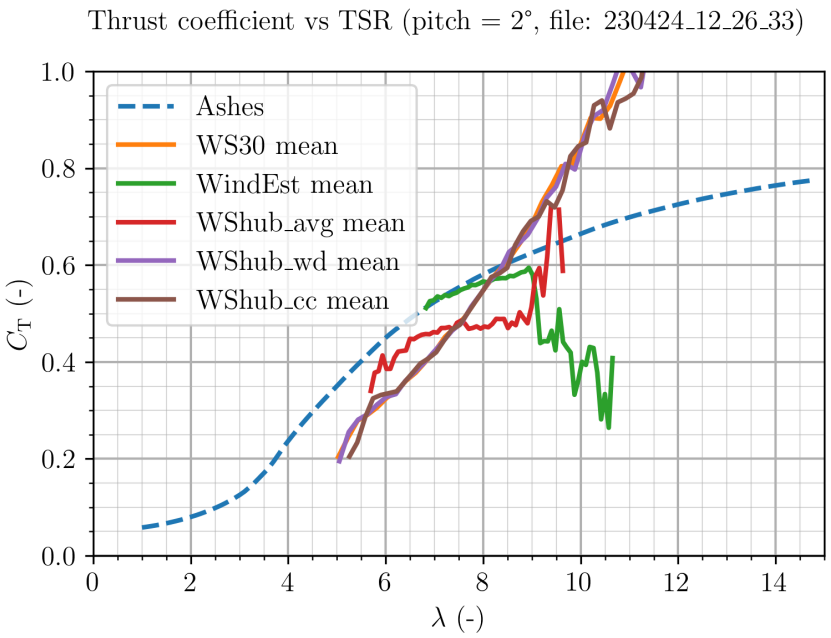


Figure 3.20: C_T vs. λ for the considered wind speed estimations from a single file

In this case, the curves derived from different wind speed estimates exhibit significant deviations from the theoretical one. "WindEst" aligns well with the theoretical reference in certain regions. However, in higher λ regions, significant deviations are also observed for this estimate. In contrast, Figure 3.6 shows much smaller deviations from the theoretical values when using "WindEst". The reason for these differing results remain unclear.

4
Data Pre-Processing

Data pre-processing is a crucial step to improve the reliability of subsequent analyses. It is applied before the main methods to reduce the influence of noise and eliminate unrepresentative data. In this work, two types of data pre-processing are used: conventional mathematical filtering techniques and a ML-based approach designed to reconstruct the generator torque setpoint. These approaches are described in this chapter.

4.1 Mathematical Data Preparation

Mathematical data preparation involves the application of basic filtering techniques, as outlined below, as well as the use of physically motivated mathematical relationships.

4.1.1 Calculation of Parameters

The determination of electrical torque in this work is based on available measurement data and physical equations. The approach follows established mathematical formulations of physical laws outlined in Section 2.2.1. However, applying these formulations requires compliance with mathematical constraints, such as avoiding division by zero. Although programming environments such as PYTHON include built-in safeguards, a more robust solution involves the implementation of specific filters, as described below.

4.1.2 Filters

Filters are required prior to applying the method in order to remove outliers and isolate representative data. This is necessary due to the presence of corrupted data points arising from sensor inaccuracies, processing errors, special operating conditions such as emergency shutdowns, or normal turbine operational noise of the turbine. The two mathematically defined filters employed in this work are described below.

First, a **values filter** is applied, which either excludes or retains all the points for which the value of a specific parameter falls within a defined range. For example, in a torque versus rotor speed diagram, the filter can be used to ignore all data points below specific thresholds in rotor speed and torque that are not relevant for

the method.

Additionally, this filter can be configured to retain only those data points where a given parameter takes on a specific value. An example is a pitch filter, which isolates data based on the pitch angle.

Second, a **density filter** is used to remove data points located in sparsely populated regions of the data space. This prevents outliers from distorting statistical measures such as averages. The density filter implemented in this work is two-dimensional bin-based. A threshold is defined in advance to represent the percentile below which low-density areas are excluded from the analysis.

4.1.3 Settings

Since the methods applied in this work do not rely on data points from Region 1, values within this region are excluded. For data exhibiting typical noise characteristics, removing all data points with torque values below approximately 300 N m is effective. In cases with increased noise levels, additional filtering based on a minimum rotor speed threshold may be applied.

For the density filter, a configuration of 50 bins and a percentile threshold of 2 is used. These settings were found to yield consistent and reasonable results.

4.2 ML-Based Reconstruction of Generator Setpoint

In a torque-rotor speed diagram, the generator setpoint "GenTorqSP" exhibits significantly less noise compared to the torque calculated from measured values. Consequently, reconstructing "GenTorqSP" from measurement data offers a promising approach to obtain a smoother and more reliable representation.

4.2.1 Approach

The following applies the procedure outlined in Subsection 2.4.1. The target variable is the generator torque setpoint "GenTorqSP", which the model is trained to predict based on available measurement data.

Instead of using the full set of available parameters, selected subsets of features are used. This decision is based on two main considerations. First, many of the recorded parameters are not relevant for the prediction of this target variable. Second, as shown in Chapter 3, not all measured values exhibit direct or consistent correlations. An example is the variation among different wind speed measurements and estimates. To evaluate the impact of the feature selection, different feature sets are considered. The features used are "DCC", "DCV", "XTurbSpeed1", "PAB1", "WindEst", "WSN", "WS30", "WD30", "YP", "RST2", "TMBNS", "TMBEW", and

"TMBTOR".

The data used for model training is taken from a selected subset of available files. This choice is based on the following considerations. First, using all files would lead to significantly increased computation time. Second, as described in Subsection 3.3.3, only certain files seem to contain reasonable values for the required parameters. Third, limiting the training data allows for clearer evaluation of model performance and generalization by retaining more data for testing.

For model training, both strategies described in Subsubsection 2.4.1.2 are applied: a NN and a decision tree. Specifically, the multi-layer perceptron regressor and the histogram-based gradient boosting regression tree are used, both implemented with the SKLEARN library in PYTHON. Each method is applied with a single, representative configuration, for which no extensive hyperparameter optimization was executed.

4.2.2 Model Development

The accuracy of an ML model is influenced by multiple factors, including the quality and quantity of training and test data, the selection of relevant features, and the choice of ML technique. These factors and their relative impacts are examined in this subsection.

All models are trained on a selected set of files that primarily contains the control algorithms depicted in Figure 3.12. The trained models are then evaluated on a separate file not included in the training set.

Model performance is assessed using the coefficient MSE defined in Subsubsection 2.4.1.2.

4.2.2.1 Choice of Training and Test Data

The influence of training and test data selection on model accuracy is analyzed through two sub-factors: the **assignment of multiple torque values to a single rotor speed value** and the **considered rotor speed range**.

To assess the first sub-factor, the model is tested on two datasets. The first contains only data resulting from the application of a single, known control algorithm present in training. The second includes multiple, partially unknown control algorithms. Table 4.1 shows the corresponding MSEs.

Type	MSE
Known algorithm	3065
Partially unknown algorithms	8484

Table 4.1: MSE values for model tested on data with known and partially unknown control algorithms

Lower MSE corresponds to testing on a known algorithm, indicating better prediction accuracy. The presence of unknown algorithms significantly degrades performance.

Figures 4.1 and 4.2 show the model output for the known algorithm case. The actual and predicted "GenTorq" values (abbreviated as "GTSP") are plotted against each other and over rotor speed, respectively. The graphs are included solely for the factor considered here to provide context for the MSE values, which serve as the primary evaluation metric in this section.

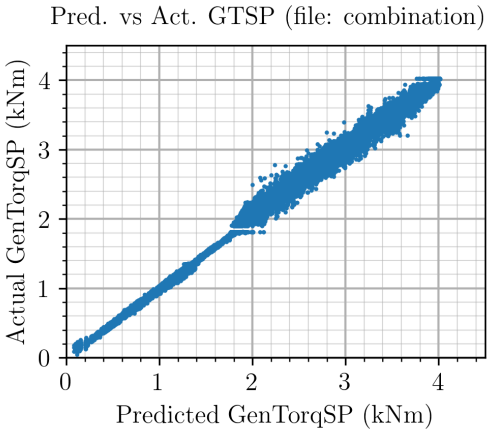


Figure 4.1: Actual "GenTorqSP" (GTSP) over the predicted value for model applied to dataset including one algorithm

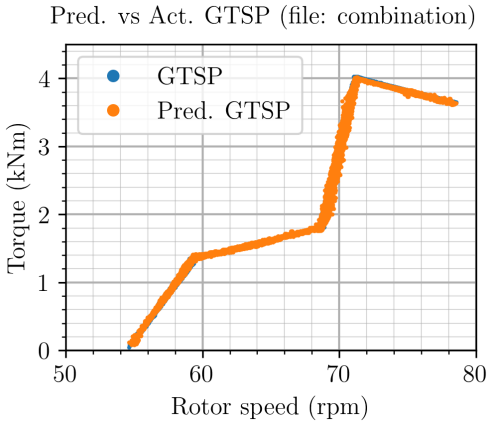


Figure 4.2: Actual and predicted "GenTorqSP" (GTSP) over rotor speed for model applied to dataset including one algorithm

In Figure 4.2, the actual "GenTorqSP" (blue) is nearly entirely overlapped by the predicted one (orange), indicating strong model performance. The change in noise behavior observed in Figure 4.1 near 2 kN m is likely caused by the merging of two data files used in the analysis.

Figures 4.3 and 4.4 present results for data containing multiple (partially unknown) algorithms.

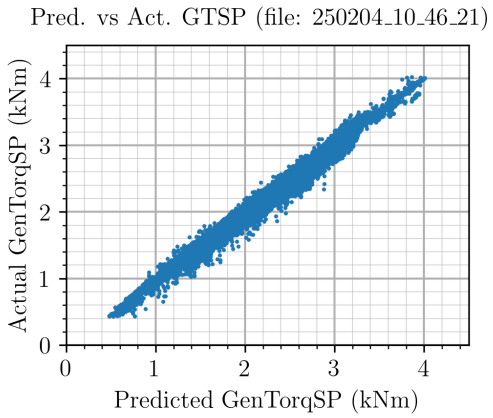


Figure 4.3: Actual "GenTorqSP" (GTSP) over the predicted value for model applied to dataset including multiple (partially unknown) algorithms

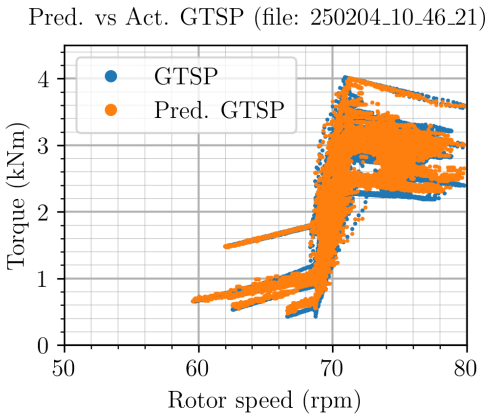


Figure 4.4: Actual and predicted "GenTorqSP" (GTSP) over rotor speed for model applied to dataset including multiple (partially unknown) algorithms

Compared to Figure 4.2, more blue dots are visible in Figure 4.4, implying a more significant deviation between the predicted and the actual "GenTorqSP". This observation corresponds with the higher MSE reported in Table 4.1.

The second sub-factor addresses the rotor speed range represented in the training data. To evaluate its impact, two models are compared: one trained using data covering a limited rotor speed range and another trained on a broader range. Both training datasets contain similar control algorithms, and both models are tested on the same dataset. The corresponding MSE values are shown in Table 4.2.

Type	MSE
Smaller rotor speed range	5459
Larger rotor speed range	3065

Table 4.2: MSE values for models trained using data covering different breadths of rotor speed range

It can be observed that excluding specific rotor speed ranges from the training data can lead to a notable loss in model accuracy. However, this loss is confined to the rotor speed intervals not represented in the training data, as illustrated in Figures 4.5 and 4.6.

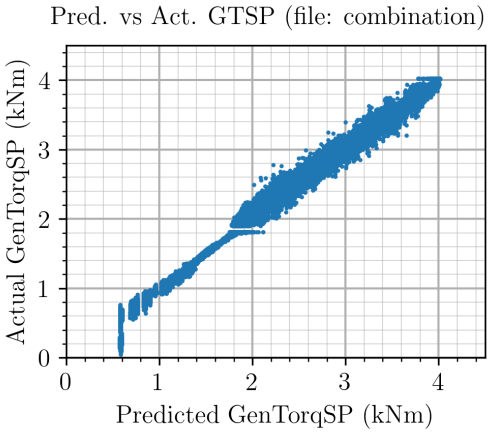


Figure 4.5: Actual "GenTorqSP" (GTSP) over the predicted value for model trained using data covering a smaller rotor speed range

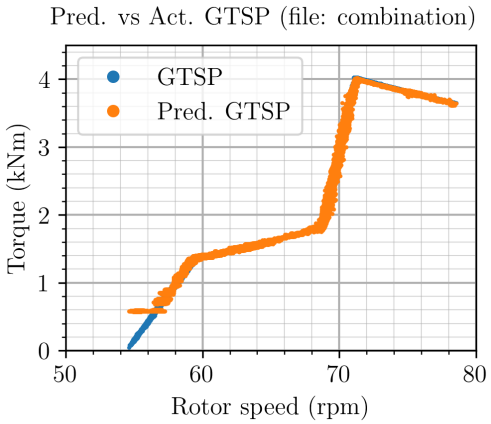


Figure 4.6: Actual and predicted "GenTorqSP" (GTSP) over rotor speed for model trained using data covering a broader rotor speed range

For rotor speeds below approximately 57 rpm the predicted target value remains constant, resulting in a significant deviation from the actual value. This behavior arises because the training dataset lacked samples in this rotor speed range.

These findings indicate that the model accuracy is highly dependent on the selection of training and test data. Accurate predictions require both the inclusion of known control algorithms and comprehensive coverage of the rotor speed range in the training data.

4.2.2.2 Wind Speed

The second factor examined is the impact of different wind speed determination approaches on model accuracy. Specifically, the influence of using either measured wind speed, represented by "WSN", "WS30", or a combination of both, or the estimated wind speed "WindEst" is evaluated. A histogram-based gradient boosting regression tree is applied for this comparison. Table 4.3 lists the resulting MSE values for the respective wind speed input, with all other features kept constant.

Type	MSE
WSN	8687
WS30	8726
WSN, WS30	8972
WindEst	43 835

Table 4.3: Model performance using "WSN", "WS30", both, or "WindEst" as wind speed feature

Among the tested inputs, "WSN" yields the lowest MSE, suggesting that it most

closely reflects the actual wind speed relevant for the model. The marginally higher MSE for "WS30" indicates slightly lower accuracy. The combination of "WSN" and "WS30" results in increased error, implying that the two measurements are not easily reconcilable in the feature space. The use of the estimated wind speed "WindEst" leads to a substantially higher MSE, indicating poor alignment with the actual system dynamics and low suitability as an input feature. However, this result may be due to the unintended inclusion of data files containing manipulated "WindEst" values, as noted in Subsection 3.3.3.

4.2.2.3 Combinations of Features

The third factor analyzed is the influence of specific feature combinations on model performance. This analysis also reveals correlations between the measured parameters. All models are evaluated using the same noisy test dataset, which includes multiple control algorithms and corresponds to the file shown in Figure 4.4. Identical training and test datasets are used consistently across all model configurations.

For evaluation, the parameters listed in Subsection 4.2.1 are divided into groups as displayed in Table 4.4.

Group	Features
Electrical power (EP)	DCC, DCV
Rotor speed (RS)	XTurbSpeed1
Pitch angle (PA)	PAB1
Mechanical torque (MT)	RST2
Tower moment (TM)	YP, TMBNS, TMBEW, TMBTOR
Wind speed (WS)	WSN, WS30

Table 4.4: Considered groups for the evaluation and their included features

Although the previous subsubsection identified "WSN" alone as optimal for wind speed estimation, both "WSN" and "WS30" are included due to their expected complementary sensor characteristics described in Subsection 3.3.3. "WSN" captures relative trends accurately, while "WS30" better reflects actual magnitudes.

In the following, the impact of considering or excluding individual groups in combination with others is investigated. A histogram-based gradient boosting regression tree model is used. Table 4.5 summarizes the group combinations tested and the resulting MSE values. It is assumed that rotor speed measurements are always available.

Model	Groups						MSE
	EP	RS	PA	MT	TM	WS	
1	X	X					9876
2	X	X	X				8484
3	X	X	X		X		8915
4	X	X	X			X	8972
5	X	X			X		12 009
6	X	X				X	11 567
7		X	X	X	X		65 672
8		X	X	X		X	65 959
9		X		X	X		72 091
10		X		X		X	110 291
11		X	X		X	X	86 188
12	X	X	X	X	X	X	31 308

Table 4.5: Considered groups and resulting MSE

Groups "EP", "RS", and "MT" are assumed to represent torque-related measurements. "TM" and "WS" are considered to reflect wind speed. The latter classification is based on the following considerations.

Wind speed can be quantified through various methods. The most direct method uses wind sensors. However, obtaining representative values from this strategy is challenging, as described in Subsection 3.3.3. An alternative approach is to estimate wind speed indirectly via tower bending moments. The thrust force of the wind induces bending of the tower, whose mechanical response can be measured and related to the wind load acting on the turbine. Table 4.5 investigates both approaches.

Table 4.5 shows that Model 2, which considers electrical power and pitch angle along with rotor speed, achieves the lowest MSE and thus the best performance. The poorest performance occurs in Model 10, which includes only rotor speed, mechanical torque, and wind speed. This indicates that the selection of input parameters is critical for model accuracy.

4.2.2.4 Applied Strategy

The fourth factor examined is the impact of the applied ML strategy on model accuracy. A histogram-based gradient boosting regressor is used as the decision tree approach, while a multi-layer perceptron regressor with one hidden layer represents the NN approach. The training files are specified in the introduction to this subsection. Testing includes the single-algorithm files from Figure 4.2 and the multiple-algorithm file from Figure 4.4. Table 4.6 lists the resulting MSE values.

Strategy	MSE
Decision tree, one algorithm	3065
Decision tree; multiple algorithms	8484
NN; one algorithm	2560
NN; multiple algorithms	12 423

Table 4.6: MSE values of models, created based on decision tree and NN, applied on different data

For the considered files, the NN achieves lower error on data closely matching training conditions (single, known control algorithm). The decision tree model, however, generalizes better to data containing multiple, previously unseen control algorithms.

4.2.3 Limitations

Since training requires knowledge of the actual value to be predicted, this strategy cannot be applied to an unknown turbine and controller for which values of this parameter are not available.

Furthermore, the results in the previous subsection indicate that a model trained on data from a specific turbine and its controller does not achieve the same prediction accuracy when applied to measurement data from a turbine featuring different characteristics.

Depending on the amount of training data, the ML technique applied, its configuration, and the optimization methods used, the resulting model may have a very large file size, and generating it can involve lengthy compilation times.

4.2.4 Settings

The configuration considered for the **histogram-based gradient boosting regression tree** is summarized in Table 4.7.

Parameter	Value
Maximum number of iterations	500
Learning rate	0.05
Maximum depth of each tree	6
L2 regulation	0.1
Verbosity level	1

Table 4.7: Settings of histogram-based gradient boosting regression tree models [27]

The settings for the **multi-layer perceptron regressor** are shown in Table 4.8.

Parameter	Value
Hidden layer sizes	(128, 64, 32)
Activation function	Rectified linear unit function
Solver for weight optimization	'adam'
Strength of L2 regularization term	0.1
Maximum number of iterations	3000
Early stopping	True

Table 4.8: Settings of multi-layer perceptron regressor models [28]

The model development settings used for subsequent ML applications in this work are based on the findings of this section. The selected configuration corresponds to the most complete, accurate, and generalizable model for the cases considered. This includes the features of Model 2, training over a broad rotor speed range, and the use of the histogram-based gradient boosting regression tree.

4.3 Results

This section presents the results of the two pre-processing strategies applied in this work. Both figures display the value-filtered electrical torque data points in blue, calculated from the measurements of "DCC", "DCV", and "XTurbSpeed1" using Equations 2.2 and 2.3. In addition, Figure 4.7 displays the remaining data points after applying the density filter described in Section 4.1 in orange, while Figure 4.8 presents the results of the ML-based data preparation method introduced in Section 4.2, using the reconstructed generator torque setpoint ("GTSP_pred").

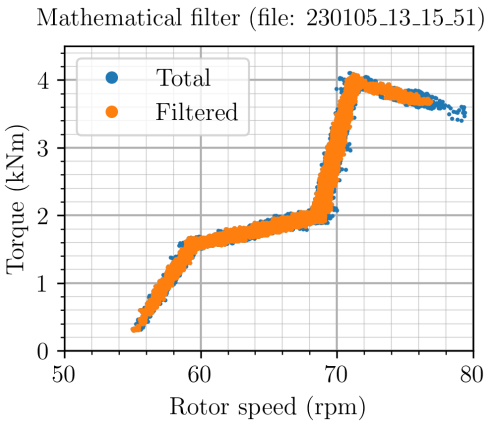


Figure 4.7: Application of mathematical (density-based) filtering

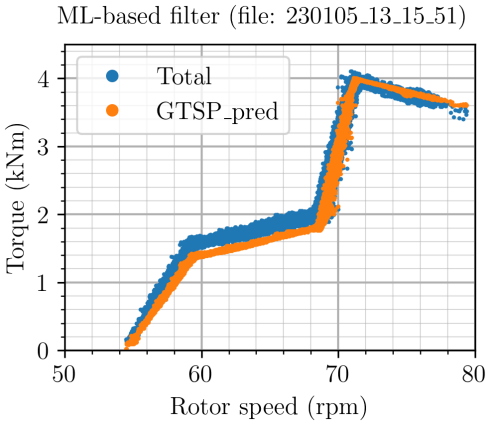


Figure 4.8: Application of ML-based filtering

The importance of the density filter applied in Figure 4.7 is demonstrated by its exclusion of rotor speed areas with low data density, such as the range between 77 rpm

and 80 rpm. Without this filtering, sparse data would reduce representativeness and distort results of subsequent analyses in these regions.

Figure 4.8 presents the positive effect of the ML-based pre-processing strategy, notably a significant noise reduction. This enhances the reliability and accuracy of further data analysis. The deviations between the "GTSP_pred" and measured torque curves, primarily in the rotor speed range corresponding to Region 2, are likely caused by discrepancies between the values used for controller calibration and the turbine's actual operational behavior, as discussed in Chapter 3.

5
Main Methods

This chapter presents the main methods developed in this work. It covers the tools utilized, the underlying working principles, the results obtained from applying the methods, the evaluated configurations, and the identified limitations.

The development of each method strives for an application without knowing the data in advance. The methods are created using the programming language PYTHON and its libraries.

5.1 Tools

Following the pre-processing of the data, several tools are applied to structure the dataset. These tools include methods for binning, averaging, and segmenting the data.

5.1.1 Binning

The averaging procedure employed in this method is based on the binning of the data points. This means that the considered data point area is divided into a predefined amount of intervals (bins) of equal length.

In the one-dimensional case, the binning only takes place along the x-axis. This implies that for each bin all values along the y-axis are considered. This approach is suitable when the dataset contains only one control algorithm as illustrated in Figure 3.11.

If two or more control algorithms are applied, as in Figure 3.12, a one-dimensional approach is insufficient. In such cases, a two-dimensional binning is required, where the data range is partitioned along both the x- and y-axes. This enables differentiation between the control strategies within the dataset.

The dimensionality of the binning also affects subsequent processing. One- and two-dimensional binnings necessitate different procedures for handling and interpreting the resulting bins and the data contained within them.

Several challenges are associated with the selection of bin sizes or counts. A small number of bins can lead to oversmoothing and a reduction in the representative-

ness of trends when the values in each bin are averaged later. On the other hand, too many bins can result in too little measurement points per bin which leads to a higher weight on outliers. However, this latter issue is primarily relevant to one-dimensional binning.

In this work, only one-dimensional binning is applied.

5.1.2 Averaging

For each bin, the average of the y-values from the data points contained within the bin is calculated. As a representative x-value, the center of the bin is taken. These two values define a representative coordinate (average point) for each bin. Based on these coordinates, the slope between two consecutive bins can be determined. This procedure is illustrated in Figure 5.1.

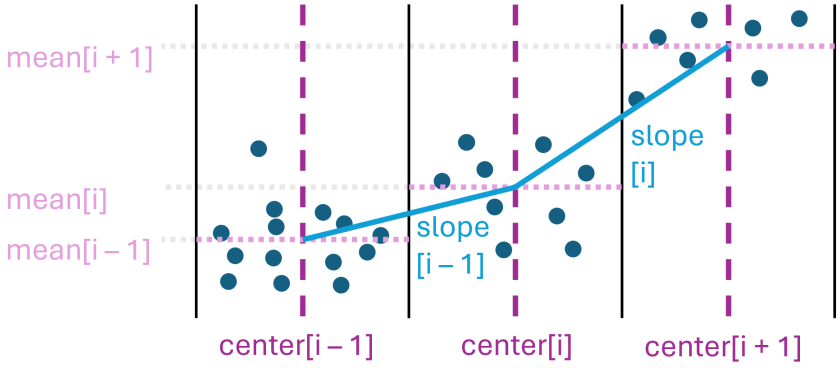


Figure 5.1: Binning and averaging strategy

In the figure, bin boundaries are shown as black lines. The dashed lines indicate the calculated means and centers, while the blue continuous line represent the slopes between bins.

To represent each region with a single straight line, the average of all slopes within that region is computed. Additionally, one average x-value and one average y-value are calculated from the bin centers and means within the region. These three values combined define the region's average line.

5.1.3 Segmenting

As previously discussed, some datasets may contain multiple control algorithms. In such cases, the averaging strategy described above becomes unsuitable for one-dimensional binning. To avoid this, the data can be segmented into so-called chunks.

Chunks are a collection of consecutive data points over a predefined time span, such as 1 min or 10 min. This temporal segmentation exploits the assumption that control algorithms typically remain constant over short time periods. As a result, the likelihood of including multiple algorithms within a single chunk is reduced.

5.2 Identify Control Algorithm

The primary objective of this work is to develop a method for identifying the control algorithm used in a given dataset. A control algorithm is defined by its control regions, as outlined in Subsection 3.2.2. Consequently, identifying the control regions from measurement data is equivalent to identifying the control algorithm.

The electrical torque-rotor speed diagram is well suited for this purpose. This representation offers a clear visualization of the system's operational characteristics, including the control regions described in Subsection 3.2.2.

As shown in Figure 3.11, when a single control algorithm is active, the data points follow a distinct trajectory across the considered rotor speed range. Although the points do not align perfectly along a single curve, the deviations, such as those visible in Figure 3.11, are mainly attributed to functional noise caused by normal turbine and control behavior in response to wind speed variations. Nevertheless, to model and quantify the underlying torque-rotor speed relationship, a single representative curve is required. In this context, an average line is deemed appropriate. Such a line is assumed to approximate the actual torque-rotor speed relation in this region set by the controller.

Since the method is intended to operate without prior knowledge of the dataset, standard linear fit tools provided by Python packages are unsuitable, as they require the number of lines (i. e., regions) to be specified in advance. Instead, the developed method determines the number of regions autonomously based on the input data.

This section describes the process of generating the regions' average lines. The resulting line segments are then stored in a LUT described in Subsection 2.3.5, which characterizes the control algorithm and allows the reconstruction of the region-specific torque-speed relationships.

5.2.1 Strategy

The method is based on the binning and averaging strategy outlined in the previous section. The slopes between consecutive bins form the core of the analysis. The underlying assumption is that a region transition in a torque-rotor speed diagram is indicated by a significant change in slope behavior.

Initially, the strategy considered involved detecting such transitions by directly comparing two consecutive slopes. However, this approach proved unreliable. Due to data noise and the smoothing effect introduced by binning, the differences between consecutive slopes are too small or inconsistent for some cases to allow for robust detection of transition points.

Instead, an alternative strategy was adopted: each slope is compared to a running

mean slope. This mean is calculated from the preceding slopes that are assumed to belong to the same region. The affiliation of the current slope to this region is evaluated based on its deviation from the mean. This approach is considered more robust, as a slope from a different region is expected to deviate more clearly from the regional mean than from its immediate predecessor.

If the deviation exceeds a predefined tolerance, a transition between regions is assumed. In that case, the region is considered complete, and a region average line is calculated according to the procedure described in Subsection 5.1.2. The intersection points of these lines then define the boundaries between regions.

To apply the method to data containing more than one control algorithm, segmenting into chunks in accordance with Subsection 5.1.3 can be considered. However, it must be ensured that each chunk contains a sufficient number of data points to allow reliable averaging and slope calculation. Inadequate data density within a chunk may lead to unstable or unrepresentative results.

5.2.2 Method Development

The method development is structured into three stages. First, the straight region-specific lines are created from the pre-processed measurement data. Second, correction strategies are introduced to improve the robustness and general applicability of the method by compensating for irregularities. Third, the identified regions are represented in a LUT.

Appendix A.1 presents the core components of the method described in this subsection in the form of PYTHON code.

5.2.2.1 Create Straight Control Region Lines

The torque-rotor speed relations in the relevant control regions, except the one corresponding to the "optimal mode", follow straight lines. For the turbine considered in this work, the "optimal mode" line does not significantly deviate from a linear shape, as shown in Figures 3.11 and 3.12. Therefore, a linear approximation is also applied here to simplify the implementation.

At the beginning of the procedure, **comparison parameters** are defined. As introduced in the previous subsection, the basis of the method is the comparison of the current slope to the mean of the preceding slopes within the currently considered region. Since no preceding slopes exist at the start, the initial reference ("mean slope") is set to the very first slope. Accordingly, the algorithm begins its evaluation with the second slope.

A tolerance value is defined to determine whether the difference between the current slope and the average of the preceding slope in the currently considered region is significant enough to indicate a transition between regions. This threshold is empirically determined.

Additional internal objects are initialized to support the method's execution in Python. These include the orientation variable making the start of a region, which is initially set to zero, and several lists to store values computed in the main part of the method.

Entering the **main part** of the procedure, the algorithm first checks whether the currently considered slope is the last in the sequence. If this is the case, the region currently being processed is closed, and the averages of the slopes, centers, and means in this region are calculated in accordance with Subsection 5.1.2. These calculated parameters are then added to a list for subsequent use.

The core of the method functions as follows. If the currently considered slope is not the last one and the tolerance is satisfied, the slope is mathematically included in the mean of the region "average of the preceding slopes" used for comparison. This scenario is illustrated on the left side of Figure 5.2. Here, the difference between slope[3] and the average of the preceding slopes in the considered region, which are slope[1] and slope[2], is smaller than the tolerance. Consequently, the method classifies slope[3] as part of the same region.

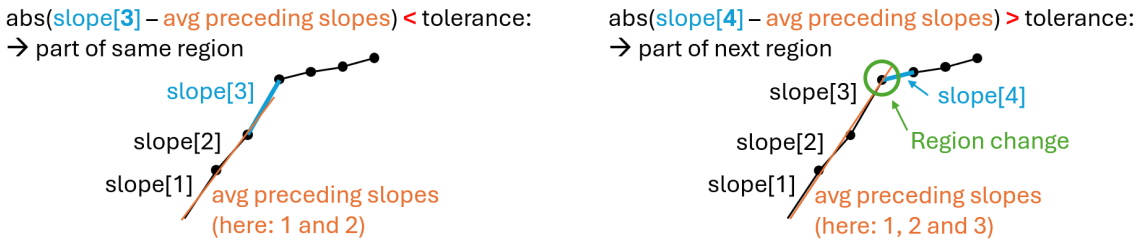


Figure 5.2: Main principle of the method to identify control algorithm

If the tolerance is not met, the current region is closed. The same parameters as in the case of the final slope are computed and added to the list. In addition, a new region is initialized: a new region start is defined, and a new preliminary mean slope is set based on the slope at this point. This case is depicted on the right side of Figure 5.2, where the difference between slope[4] and the average of the preceding slopes in the current region (now including slope[3]), exceeds the tolerance, indicating a transition to a new region.

5.2.2.2 Correction Strategies

The prevailing wind conditions determine the rotor speeds that are reached in the period under consideration. As a result, the number of data points per bin can vary significantly. Bins with too few data points can lead to non-representative mean values, potentially distorting slope calculations and subsequent region classification.

Correction strategies address this issue by improving the handling of regions with fewer available measurement data. They reduce the influence of bin averages that deviate from the actual trend within a region. Consequently, the method becomes

less sensitive to the specific choice of the tolerance value.

The first correction strategy **permits one slope to exceed the defined tolerance** if the deviation is likely the result of unfavorable bin placement. Specifically, a slope that individually violates the tolerance is still accepted if the average of this slope and the immediately following slope does not deviate from the mean slope beyond the tolerance. This is supposed to compensate for fluctuations caused by insufficient data density or local anomalies.

This idea is illustrated in Figure 5.3.

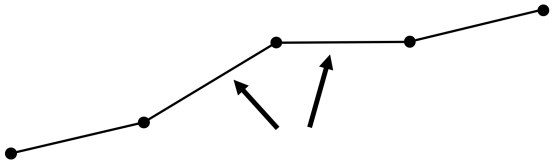


Figure 5.3: Correction strategy to tolerate exceptions

In this example, all slopes are assumed to belong to the same control region. However, without the correction strategy, the method may misinterpret the increased slope of the second line as a region transition. This misclassification can occur when bin borders fall at positions with low data density, leading to atypical slope values. To mitigate this, the strategy calculates the mean of the current and the subsequent slope. If this average slope remains within the tolerance when compared to the average of the preceding slopes, both slopes are accepted as part of the current region. This allows the method to handle short-term fluctuations without falsely detecting a control region change.

The second correction strategy **only allows regions that include more than one bin**. This is necessary because transitions between control regions may span multiple bins and exhibit several consecutive slopes that exceed the tolerance. Without correction, such transitions could result in the false identification of short, non-representative regions.

To avoid this, the strategy allows for a single intermediate slope to be ignored if it is likely part of a broader transition zone. This is sufficient for the analyzed data and prevents over-fragmentation of regions. Figure 5.4 illustrates such a case.

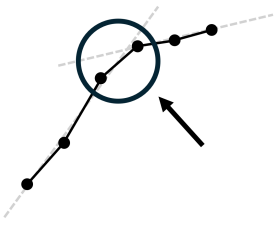


Figure 5.4: Correction strategy to handle stepwise transitions

In this example, the black circle marks the intermediate step, while the dashed gray lines indicate the actual region average lines.

The third correction strategy is applied after the regions have been created and functions as a **backup** in case the preceding technique is insufficient. Here, the mean slopes of two consecutive regions are re-evaluated against the tolerance criterion. If the criterion is not satisfied, it is assumed that both regions actually form a single region, and their corresponding average lines are merged. This merging is performed by calculating weighted averages of the means, centers, and slopes, based on the number of elements in each region.

The fourth correction strategy **prevents the creation of a final region that lacks representativeness** due to insufficient measurement data in the last bin. Such situations arise when the number of data points in the final interval is too low to produce meaningful statistical values. To avoid this, the method only saves the final region if it covers more than one bin.

5.2.2.3 Extraction of Look Up Table

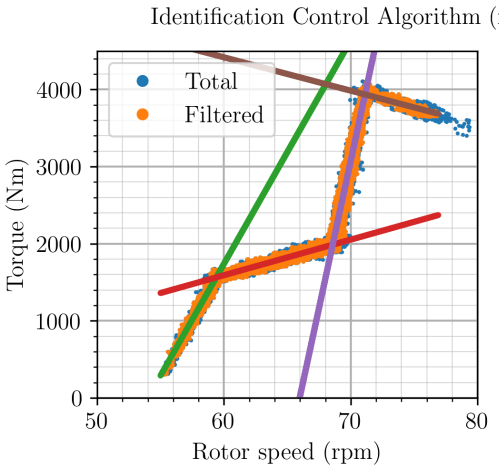
The LUT extraction strategy depends on the regions detected in the data. If Region 1.5 is the first region, the initial point listed in the LUT corresponds to the intercept of the Region 1.5 average line with the x-axis. If another region is first, this region's average coordinate from Subsubsection 5.2.2.1 is used. The same procedure applies for the last region. Between these, the LUT points correspond to the intersections of adjacent region average lines.

5.2.3 Results

This subsection presents the results obtained by applying the described method to the data file shown in Figure 3.11. Both pre-processing strategies introduced in Chapter 4 are considered for this analysis.

5.2.3.1 Mathematical Data Preparation

Figure 5.5 shows the results of the method to identify the control algorithm, applied to data pre-processed using the density filter described in Section 4.1.



Rotor Speed	El. Torque
53.99	0.00
59.39	1563.28
68.57	1987.04
71.13	3935.47
73.98	3811.87

Figure 5.5: Result of method to identify control algorithm

Similar to Figure 4.7, the diagram on the left shows the measurement data from the considered file after excluding value ranges irrelevant to the method (dark blue dots) and the data remaining after applying the density filter (orange dots). The colored lines represent the average lines corresponding to the identified regions.

On the right side, the LUT is displayed. Based on these values, the average lines of the regions, and thus the control curve, can be reconstructed. Additionally, the method exports the LUT as a ".csv" file, whose structure is shown in Figure 5.6.

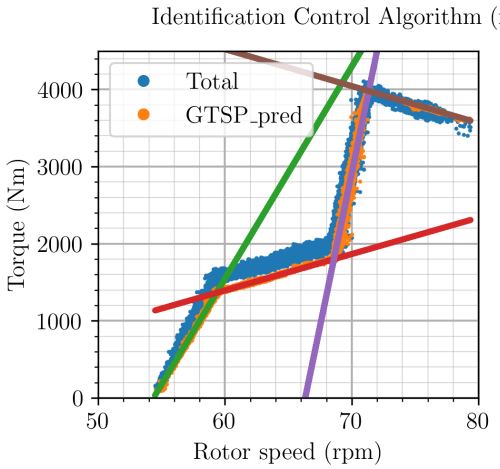
```
1   region  rs  Tel sigma
2   1.5 53.99   0.00    134.38
3   2   59.39   1563.28 134.38
4   2.5 68.57   1987.04 254.03
5   3   71.13   3935.47 140.22
6   3   73.98   3811.87 140.22
```

Figure 5.6: Structure of output ".csv" file generated by method

In addition to rotor speed and torque values, the region numbers and the corresponding σ for the considered file and region are listed. The numbering enables a clearer structure for further analyses using the file. The extraction and application of the σ values are explained later in conjunction with another method.

5.2.3.2 ML-Based Setpoint Reconstruction

The method is now applied to data pre-processed using the ML-based approach introduced in Section 4.2 and shown in Figure 4.8. The corresponding results are presented in Figure 5.7, which follows the same structure as Figure 5.5.



Rotor Speed	El. Torque
54.36	0.00
59.32	1362.98
68.60	1799.93
71.34	3979.91
74.91	3809.63

Figure 5.7: Result of method to identify control algorithm using a ML-based generator torque setpoint prediction model

As with the previous version, the LUT can be exported as a ".csv" file. However, due to the significant deviations between the "GenTorqSP" prediction (shown in orange) and the measured values (shown in blue), explained in Section 4.3, the extracted σ values are not representative in this case.

5.2.4 Settings

The tolerance value is set based on experiences with the method. Here, a value of 150 N m/rpm is used. Other datasets may require adjustment of this tolerance to yield meaningful results. The correction strategies presented earlier reduce the necessary tolerance range.

The number of bins is likewise selected based on experience. For this work, 30 bins were found appropriate.

5.2.5 Limitations

As stated in the beginning of this section, the method approximates the torque-rotor speed relation in Region 2 by a straight line. For short Region 2 intervals, as in the case of the Chalmers test wind turbine, the approximation is sufficiently accurate, as demonstrated in the following section. However, turbine controllers designed for optimal operation over wider rotor speed ranges may introduce deviations due to this assumption. These deviations arise from the averaging strategy based on the mean coordinates within a region. For larger, curved Region 2 intervals, the resulting straight line may be mispositioned, leading to inaccuracies in determining region transition points and thus in the LUT values. The method has so far been tested only on data from the Chalmers test wind turbine; its applicability to other turbines under this assumption remains unverified.

The method assumes that only a single control algorithm is active within the dataset.

If multiple control algorithms are present, the data must be divided into chunks as described in Subsection 5.1.3 to isolate data corresponding to a single algorithm. The appropriate chunk size may vary and typically requires empirical determination.

Sparse data can reduce the representativeness of bin averages. Despite the implemented correction strategies, this may lead to slopes between bins that are unreliable for interpretation by the method.

5.3 Identify Controller Parameters for a Given Architecture

As part of the WeDoWind challenge introduced in Section 1.5, a feature is developed to identify parameters of the NREL controller used in the Chalmers test wind turbine. This graphical approach builds on the method described in the preceding section. In addition, it enables an assessment of the core method's accuracy by comparing the identified control parameters to the actual ones.

5.3.1 Idea

The control parameters used to configure the considered NREL controller correspond to characteristics of the torque-rotor speed relationship within the individual control regions defined in Subsection 3.2.2. These characteristics include the transition points between control regions and the slopes of the lines representing the torque-rotor speed mapping. The parameters are therefore derived from the intersections and slopes of the average region lines identified in Section 5.2.

To assign the parameters to their respective control characteristics, the method must first allocate each average line to its corresponding control region.

5.3.2 Region Identification

To enable unambiguous classification, the control regions defined in Subsection 3.2.2 are specified here in terms of their relevant characteristics. Unique identifiers are formulated for each region.

The torque-rotor speed relationship in Region 1 is represented by a straight line with a slope of $slope1 = 0$ and torque values of $torque1 = 0$. Since this region does not provide any information about the control algorithm, it is excluded from further consideration.

Region 1.5 is linear and characterized by a positive slope $slope15 > 0$.

The slope of the approximated straight line in Region 2 satisfies the condition $0 < slope2 < slope15$.

The line in Region 2.5 has a steeper slope, with $slope25 > slope15$.

The line in Region 3 shows a slight curvature but is also approximated as linear here. It is the only line characterized by a negative slope, i.e., $slope3 < 0$ [24].

5.3.3 Parameter Assignment

Based on the region classification outlined in the previous subsection, each control parameter can be unambiguously assigned to a specific intersection point or slope of the corresponding average line. The extraction logic is summarized in Table 5.1.

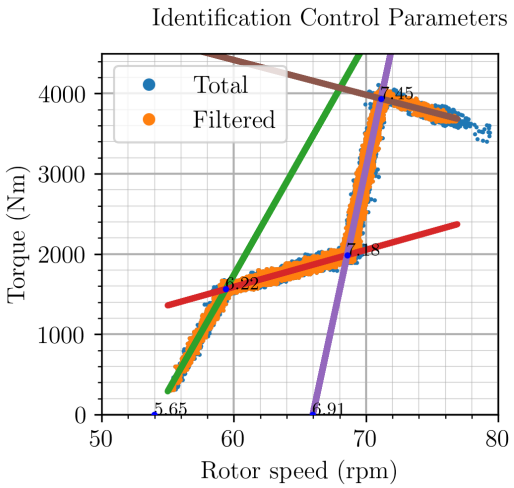
Parameter	Extraction
VS_CtInSp	Intersection Region 1.5 average line & $y = 0$
VS_RtPwr	Intersection Region 2.5 average line & Region 3 average line
VS_RtGnSp	Intersection Region 2.5 average line & Region 3 average line
VS_Slope15	Slope Region 1.5 average line
VS_Rgn2Sp	Intersection Region 1.5 average line & Region 2 average line
VS_Rgn2K	Slope Region 2 average line
VS_TrGnSp	Intersection Region 2 average line & Region 2.5 average line
VS_Slope25	Slope Region 2.5 average line
VS_SySp	Intersection Region 2.5 average line & $y = 0$

Table 5.1: Extraction of Chalmers test wind turbine control parameters [3]

The parameter assignments are derived from the NREL controller specification by Jonkman et al. [3].

5.3.4 Results

Analogous to the results of the previous method, the outcomes are presented in a torque-rotor speed diagram and a table listing the extracted parameters, as shown in Figure 5.8. The analysis first considers the mathematical pre-processing approach.

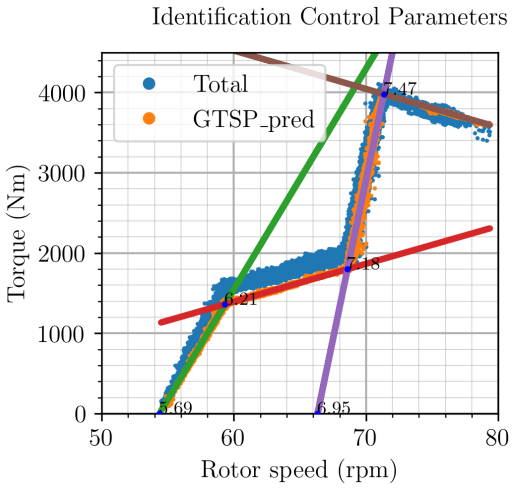


Parameters	Value
VS_CtInSp	5.65
VS_Rgn2Sp	6.22
VS_Rgn2K	30.35
VS_Slope15	2765.69
VS_SySp	6.91
VS_TrGnSp	7.18
VS_Slope25	7288.50
VS_RtGnSp	7.45
VS_RtPwr	29312.89

Figure 5.8: Results of method to identify control parameters applied on density filtered data

Here, the table on the right lists the identified control parameters using the units defined in Table 3.2. In addition, the corresponding rotor speed values are annotated directly in the plot at their respective extraction points.

Figure 5.9 shows the results of the parameter identification method based on the ML-based generator torque setpoint prediction.



Parameters	Value
VS_CtInSp	5.69
VS_Rgn2Sp	6.21
VS_Rgn2K	31.00
VS_Slope15	2623.02
VS_SySp	6.95
VS_TrGnSp	7.18
VS_Slope25	7580.98
VS_RtGnSp	7.47
VS_RtPwr	29733.96

Figure 5.9: Results of method to identify control parameters applied on ML-based prediction of generator torque setpoint

To assess the accuracy of the core method, Table 5.2 summarizes the previously presented results and compares them with control parameter values assumed to have been applied in practice.

Parameter	Actual		Identified	
	Sep.June 2023	Sep. 2022	Math. filt.	ML-based
VS_CtInSp	5.75	5.7	5.65	5.69
VS_Rgn2Sp	6.2	6.2	6.22	6.21
VS_Rgn2K	38	35	30.35	31.00
VS_Slope15	3250	2600	2766	2623
VS_SySp	6.96	6.97	6.91	6.95
VS_TrGnSp	7.2	7.2	7.18	7.18
VS_Slope25	8250	8250	7289	7581
VS_RtGnSp	7.45	7.45	7.45	7.47
VS_RtPwr	30000	30000	29313	29734

Table 5.2: Comparison between the actual Chalmers test wind turbine control parameters and the identified ones [25]

The actual values are taken from an Excel file listing control parameter settings for different operational strategies of the turbine. The set of values most likely in effect during the period covered by the dataset analyzed in Figures 5.8 and 5.9 is included in the table. However, it cannot be stated with absolute certainty whether the values in the "Sep.June 2023" or "Sep. 2022" column were actually used, or whether either of these sets fully corresponds to the configuration applied during the measurement period. The column names are taken directly from the file; their exact meanings remain unclear.

5.3.5 Settings

As described in Subsection 3.2.3, rotor speed parameters in the controller are given in rad/s. Since the diagrams used in this method display rotor speed in rpm, a conversion according to Equation 2.1 is required after parameter extraction.

For clarity in presenting the results, parameters are rounded to two decimal places.

5.3.6 Limitations

This method is specific to the parameter structure of the NREL controller implemented in the Chalmers test wind turbine.

Additionally, this method relies on the application of the method developed in the previous section.

5.4 Identify Changes in Control Algorithm

In a torque-rotor speed diagram, different control algorithms produce distinct characteristic curves, as shown in Figure 3.12. Consequently, variations in the curve indicate changes in the control algorithm. This section describes a method to identify these changes.

5.4.1 Method Development

Two approaches are considered for identifying changes in the control algorithm. The first relies on pitch angle sensor data. The second, more computationally demanding, operates independently of pitch angle measurements.

5.4.1.1 Considering Pitch Angle Measurement Data

A straightforward method utilizes the characteristics of the wind turbine's control system, as described in Section 3.2, in combination with the measured pitch angle data. The method relies on the structure of the two control loops and their respective operating regions. Specifically, pitch control is active only above the rated rotor speed. Consequently, under an unchanged control algorithm, the pitch angle remains constant below the rated speed. Therefore, any change in the pitch angle observed below the rated speed indicates a modification of the control algorithm at that time.

Although this thesis is on the Chalmers test wind turbine and its associated measurement database, the aim is to develop a method that does not rely on inside knowledge of the control system or on the availability of pitch angle measurement data.

5.4.1.2 Not Considering Pitch Angle Measurement Data

A more computationally intensive but pitch angle sensor-independent approach involves comparing measured data points with a reference LUT corresponding to the initial control algorithm. This is done by calculating the residuals between measured values and the LUT values along the torque axis.

To distinguish between normal measurement noise and deviations caused by a change in the control algorithm, a suitable decision criterion is required. Two options were considered: the use of a fixed residual tolerance based on empirical experience, and a statistical evaluation of the residuals. The statistical approach was selected, as it provides a more robust and justifiable basis for decision-making.

This approach applies a one-sample hypothesis test, as described in Subsection 2.4.2. The test requires the standard deviation σ and the expected torque values at each rotor speed from the reference control algorithm. These reference values are available from the LUT obtained in Subsection 5.2.3.

The procedure for extracting the σ values and applying them in the one-sample

hypothesis test are described in the following.

The **extraction of** σ is integrated into the method described in Section 5.2, utilizing the previously generated LUT. As outlined in Subsubsection 5.2.3.2, the LUT must be derived using the density filter-based pre-processing approach to ensure sufficient data quality.

To determine σ, the measurement data is segmented according to the control regions, whose boundary values are defined by the LUT. Within each region, these boundaries are used to construct a continuous line that provides a reference torque value for each rotor speed under consideration.

Preliminary investigations have shown that σ varies significantly across the control regions. This variation is likely influenced by the strategy used to calculate the residuals. In this approach, residuals are computed along the torque axis, i.e., as the difference between the actual measurement and the corresponding LUT value at a given rotor speed. This method is assumed to allow for a more distinct representation and classification of deviations than calculating residuals orthogonally to the LUT curves.

Applying either a global σ across all regions or region-specific σ values directly in the one-sample hypothesis test resulted in a high rate (around 50 %) of false positives, i.e., control algorithm changes were detected in cases where none had occurred.

To address this issue, a refined strategy was developed to reduce sensitivity while preserving the data's σ characteristics. This approach involves calculating three σ values for each region: the total σ across the entire dataset, the region-specific σ, and the maximum σ observed in data chunks of predefined size within each region. The maximum of these three values is stored in a new column in the LUT for each region. This helps to minimize z_0 (as defined in Equation 2.14), thereby improving the likelihood that the null hypothesis is retained, in accordance with the rejection criterion defined in Equation 2.15.

By combining this strategy with a minimum required number of data points per sample, the rate of falsely identified control algorithm changes was reduced to approximately 7.5 % for the considered files.

To **apply the one-sample hypothesis test**, the reference LUT and a separate dataset for which a control algorithm change is to be assessed is loaded.

To allow the method to be applied to datasets containing multiple control algorithms, the data is first divided into chunks, as described in Subsection 5.1.3. This segmentation increases the likelihood that each chunk represents data from only a single control algorithm. The chunk size is selected manually based on the expected number of control algorithms within the dataset. For instance, if a file contains data collected over a 30-minute period and is assumed to include several control

algorithms, a chunk size of 1 min may yield better results than a 10-minute chunk.

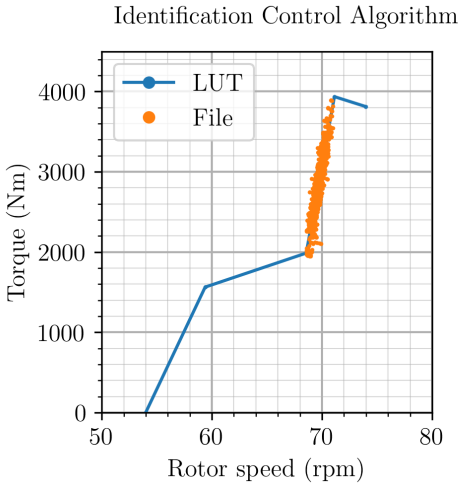
Each chunk is then subdivided into rotor speed intervals, corresponding to the control regions defined by the rotor speed boundaries in the LUT. This facilitates a region-specific comparison between the measured data and the reconstructed torque-rotor speed relation from the LUT.

The core of the method is applied individually to each region within each chunk. For each case, a continuous reference curve is generated from the LUT values, representing the nominal torque as a function of rotor speed. Residuals are then computed along the torque axis between each measured data point and the corresponding LUT value. The mean of these residuals is determined and interpreted as the term $\bar{x} - \mu_0$ in Equation 2.14.

Subsequently, the rejection criterion defined in Equation 2.15 is evaluated. Based on this test, the method determines, separately for each region and chunk, whether the data points deviate significantly from the reference algorithm, thereby indicating the presence of a different control algorithm.

5.4.2 Results

Figure 5.10 presents the output of the proposed method applied on a dataset in which the same control algorithm is used as represented by the reference LUT.



Region	2.5	
Points region	864	
Current chunk	23	
Chunks	30	
Start time	1322.2	s
End time	1382.3	s
Chunk size	1.0	min
Points chunk	1200	
Different algorithm	No	
z_0	0.06	
$z_{1-\alpha/2}$	2.81	
$\bar{x} - \mu_0$	0.52	Nm
σ	254.03	Nm

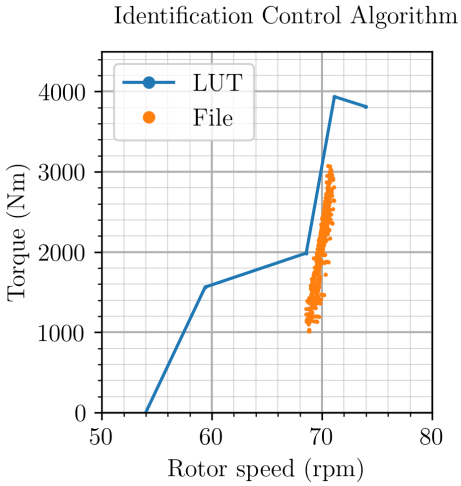
Figure 5.10: Results of method to identify changes in control algorithm indicating no different algorithm

The left panel of the figure shows the torque-rotor speed diagram, where the LUT is plotted as a blue line and the corresponding measurement points for the selected region and chunk are shown as orange dots. The panel on the right hand side contains a summary table with relevant metadata and the results of the hypothesis test.

The metadata include the control region, the number of data points within this

region of the chunk, the chunk index, the total number of chunks in the file, the start and end timestamps of the chunk, its duration, and the number of points per chunk. The test results report whether a deviation from the LUT is detected, along with the statistical parameters used for the decision introduced in Subsection 2.4.2: z_0, $z_{1-\alpha/2}$, $\bar{x} - \mu_0$, and σ.

Figure 5.11 illustrates the outcome of the method when a change in the control algorithm is detected.



Region	2.5	
Points region	1124	
Current chunk	25	
Chunks	30	
Start time	1442.19	s
End time	1502.22	s
Chunk size	1.0	min
Points chunk	1200	
Different algorithm	Yes	
z_0	-122.57	
$z_{1-\alpha/2}$	2.81	
$\bar{x} - \mu_0$	-928.7	Nm
σ	254.03	Nm

Figure 5.11: Results of method to identify changes in control algorithm indicating a different algorithm

5.4.3 Settings

To increase the statistical reliability of the results, the method is configured to evaluate only those samples containing more than 100 data points.

For the result presented above, a chunk size of 1 min was selected.

The significance level for the hypothesis test was set to $\alpha = 0.005$.

5.4.4 Limitations

As previously noted, the method may incorrectly indicate a change in the control algorithm in certain cases. The underlying cause of these false positives remains unidentified. To support the interpretation of such results, the torque-rotor speed diagram on the left side of the output window serves as a visual validation tool. If the method detects a change, but the measurement points seem to align with the LUT reference curve, it can be reasonably assumed that no actual change in the control algorithm has occurred.

6
Discussion

This chapter discusses the key results of this thesis, which are structured into three main topics: the characterization of the Chalmers test wind turbine, the development and subsequent evaluation of the proposed methods, and the analysis of the application of machine learning techniques within these methods. In addition, the relevance of this work for the research in this field is assessed, and recommendations for future research and implementation are provided.

6.1 Characteristics of Chalmers Test Wind Turbine

In this work, analyses of the characteristics of the investigated wind turbine were conducted, focusing on aspects assumed to be relevant for understanding its operational behavior. These characteristics are considered essential for the development of the methods proposed in this thesis. The most significant findings are discussed below.

First, the challenges associated with determining the wind speed acting on the turbine are addressed. Various calculation approaches based on the two available wind speed sensors, "WSN" (located on the nacelle) and "WS30" (positioned on a nearby meteorological mast), are considered and compared. The accuracy of these approaches is evaluated using a C_P-λ plot. Significant deviations are observed for most approaches, including the estimated wind speed "WindEst" provided by the database. For the "WindEst" and "WSN" signal, the choice of measurement data files proves to be a critical factor, likely due to the varying types of research conducted on the turbine. However, if a suitable file is selected, "WindEst" appears to deliver reliable results. Among the other approaches for determining the hub-height wind speed, the method that adds the average of the time-resolved differences between "WS30" and "WSN" to the "WSN" signal seems to yield the most accurate results for the considered datasets.

Second, the impact of the pitch angle adjustment above rated speed on the output power and the thrust force acting on the turbine is examined. In the considered case, the output power remains constant, while the thrust force is significantly reduced, leading to a substantial decrease in mechanical stress on the turbine components.

Third, the mechanical efficiency of the turbine and its influencing parameters are

analyzed. The parameters considered are season, electrical power output, and rotor speed. The efficiency appears to be higher during the summer months, while no significant influence of power output and rotor speed is observed within the main operational range. However, at lower electrical power output levels, a decrease in efficiency is evident. Additionally, slight discrepancies are observed between the efficiency value determined during the planning phase and the values obtained from recent measurements. As this and other pre-construction parameters are used in the calculation of the estimated wind speed "WindEst", the accuracy of this estimation is consequently affected.

Fourth, the manifestation of the applied control algorithm is analyzed using diagrams that depict the relationships between torque and rotor speed, as well as wind speed and rotor speed. This analysis also considers the distribution of measurement values.

6.2 Development and Examination of Method

The presented method shows that information about the control algorithm can be derived from measurement data of current, voltage, and rotor speed, and represented in a LUT. This is achieved through basic graphical and mathematical analysis of data points within an electrical torque-rotor speed plot, utilizing binning and averaging of the data. Based on this approach, control parameters of the investigated controller can be determined by evaluating slopes and transition positions of the control regions within the control algorithm. In addition, the method enables the identification of modifications or changes in the control strategy.

The application of the method requires pre-processing to enhance reliability and functionality. Two pre-processing strategies are employed in this work: a mathematical density filter and an ML-based model for predicting the generator torque setpoint. The density filter effectively removes data points in sparsely populated areas in the torque-rotor speed diagram where the torque is calculated from measurements of DC current and DC voltage of the generator rectifier, as well as rotor speed, using physical equations. In contrast, the generator torque setpoint prediction, based on the same input parameters, significantly reduces noise.

The method can be evaluated by comparing the extracted control parameters to the actually applied values. Results show that the method delivers accurate estimates for most parameters using both pre-processing strategies.

The statistical interpretation of changes in the control algorithm based on deviations of measurement points proves challenging. One-sample hypothesis testing is generally considered too insensitive; however, in this case, it exhibits excessive sensitivity. Consequently, it incorrectly detects changes in the control algorithm in some instances where control algorithm remains unchanged.

The applicability of the method to other wind turbines is limited and depends on

several factors. A primary constraint is the availability of measurement data for the specific parameters required by the method. The ML model used in this work is trained on the generator torque setpoint from the controller of the Chalmers test wind turbine; its predictive performance on other controllers and turbines is unknown. Training a new model for a different turbine requires access to measurement data of the generator torque setpoint, which is typically not available.

Another limiting factor is the size of the rotor speed range in Region 2. The method approximates the torque-rotor speed relationship in this region as linear. As the rotor speed range in Region 2, the optimal operation mode, increases, this linear approximation becomes less accurate, which in turn reduces the overall reliability of the method.

6.3 Assessment of Machine Learning Techniques Considered

During the development of this work, various implementations of basic ML techniques were created. The initial objectives for applying ML included improving the filtering representative data points, identifying patterns in control algorithms, and detecting changes in these patterns. The following approaches were pursued to address these objectives.

The only approach that proved effective was the supervised learning-based reconstruction of the generator torque setpoint, as described in the previous section. As part of the model development, the influence of various factors on prediction accuracy was examined. This included the quality and quantity of training and test data, different combinations of input parameters, and the choice of ML technique. These investigations also provided insight into the strength of correlation and dependency among the measured parameters. Among all tested configurations, the model using DC current and DC voltage of the generator rectifier, rotor speed, and pitch angle delivered the best results. Furthermore, for the detection of multiple control algorithms, the considered decision tree-based approach showed higher accuracy than the tested NN-based approach.

A supervised learning approach was applied to recreate the LUT from measurement points. Both the histogram-based gradient boosting regression tree technique and the NN multi-layer perceptron regressor were trained using measurement points and the corresponding LUT generated with the method developed in this work. However, both models ultimately reproduced the LUT they were trained on, regardless of the measurement points provided, limiting their applicability.

An unsupervised learning approach was intended to identify changes in the control algorithm. For this purpose, an autoencoder was developed and applied to detect anomalies in torque-rotor speed plots. The model was trained on measurement data corresponding to the initial control algorithm, with the goal of learning the charac-

teristic structure and noise of the data. It was then expected to identify deviations from this baseline when applied to other measurement datasets, thereby indicating whether a different control algorithm was in use. However, the autoencoder failed to detect the intended anomalies and instead identified unrelated regions as anomalous, rendering the approach ineffective for this specific task.

The unsupervised k-means clustering technique was applied to identify patterns in the torque-rotor speed diagram. The aim was to detect the control regions as distinct clusters. However, as with the other approaches, no configuration of the algorithm yielded results that matched the intended objective.

6.4 Comment on Meaning for Research

As noted in Chapter 2, no prior work directly addressing the thesis topic was identified during the literature review. The findings and results presented here were developed using basic tools and knowledge, with a focus on fundamental characteristics of wind turbines and their measurement data. These initial strategies provide a foundational basis for further research in multiple directions, enabling more advance and targeted investigations in the field.

6.5 Recommendations for Future Implementation and Research

To increase the generality and applicability of the method to other turbines and controllers, several further steps can be taken.

One potential improvement is the implementation of a fit that reflects the actual form of the torque-rotor speed relationship in Region 2. Instead of a linear approximation, a second-degree polynomial function that passes through the origin $(0, 0)$ should be considered to better capture the underlying physical behavior.

In addition, based on the experiences gained in this work, further efforts should be directed towards leveraging the strengths of machine learning in a more targeted and problem-specific manner.

The current method relies on a one-dimensional binning approach to compute averages, which limits its applicability to datasets containing a single control algorithm. To extend the method to datasets that reflect multiple algorithms simultaneously, a two-dimensional binning method could be introduced. However, this would require the development of a new procedure, as identifying and associating coherent bin groups corresponding to a single control algorithm is significantly more complex in two dimensions.

7
Conclusion

This thesis presented a method to identify the control algorithm of the Chalmers test wind turbine using measured operational data and to represent it in a look up table. The approach utilizes ".txt" files containing current, voltage, and rotor speed measurements, and applies physical equations to calculate torque. It relies on a graphical analysis and is designed to function without prior knowledge of the dataset structure. Data pre-processing is performed using mathematical filtering and supervised machine learning. For both strategies, the method proved effective in the evaluated cases. Other machine learning approaches were explored but did not yield usable results.

Additional features were implemented including the extraction of control parameters and detection of control algorithm changes. Parameter identification is based on the recognition of transition points between control regions and the slopes of torque-rotor speed relations. Change detection compares a reference control state, derived from one dataset, to segments of another dataset using a statistical method to identify deviations. The reference control state is represented by a look up table extracted using the main method.

The study provided insights into the behavior and control characteristics of the Chalmers test wind turbine and underlined the challenge of determining representative wind speed values. It also illustrates discrepancies between pre-construction specifications and post-installation measurements, contributing to the empirical understanding of turbine performance.

Since the investigated turbine shares basic features with commercial turbines, such as three-bladed design, variable speed, and pitch control, this study is expected to contribute to a broader understanding of wind turbine operation. It provides insights into the relationships and correlations between measured parameters. However, differences in scale and operational requirements limit the direct applicability of the proposed method. Thus, the generalizability of the developed approach to other turbines remains uncertain and requires further study.

One limitation of the control algorithm identification method lies in the approximation of the optimal turbine operation region by a linear fit. Replacing this linear approximation with a second degree polynomial, analogous to the optimal torque curve, could improve the method's generalizability. Additionally, the change detection feature exhibits a false positive rate of approximately 7.5 %, highlighting

potential for further refinement in future work.

Bibliography

[1] C. Feng and J. Zhang. "Wind Power and Ramp Forecasting for Grid Integration". In: *Advanced Wind Turbine Technology*. Cham: Springer International Publishing, 2018, pp. 299–315. DOI: 10.1007/978-3-319-78166-2_11.

[2] N. Wang. "Advanced Wind Turbine Control". In: *Advanced Wind Turbine Technology*. Cham: Springer International Publishing, 2018, pp. 281–297. DOI: 10.1007/978-3-319-78166-2_10.

[3] J. Jonkman et al. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. Tech. rep. NREL/TP-500-38060. National Renewable Energy Laboratory (NREL), 2009.

[4] J. Manwell, J. McGowan, and A. Rogers. "Wind Turbine Control". In: *Wind Energy Explained*. John Wiley & Sons, Ltd, 2009. Chap. 8, pp. 359–405. DOI: https://doi.org/10.1002/9781119994367.ch8.

[5] Y. Wang et al. "A novel data-driven deep learning approach for wind turbine power curve modeling". In: *Energy* 270 (2023), p. 126908. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2023.126908.

[6] C. Moss, R. Maulik, and G. V. Iungo. "Augmenting insights from wind turbine data through data-driven approaches". In: *Applied Energy* 376 (2024), p. 124116. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2024.124116.

[7] J. Manwell, J. McGowan, and A. Rogers. "Introduction: Modern Wind Energy and its Origins". In: *Wind Energy Explained*. John Wiley & Sons, Ltd, 2009. Chap. 1, pp. 1–22. ISBN: 9781119994367. DOI: https://doi.org/10.1002/9781119994367.ch1.

[8] S. Fogelström, M. Ellsén, and O. Carlson. *Wind Turbine Lab at Björkö*. Lab manual, course ENM097: Future Renewable Based Power Systems. July 2024.

[9] H. J. P. Keighley. "Circuits, Series and Parallel, Electrical Units, Household Electricity and Electronics". In: *Work Out Physics 'O' Level and GCSE*. London: Macmillan Education UK, 1986, pp. 151–182. DOI: 10.1007/978-1-349-07213-2_13.

[10] J. Manwell, J. McGowan, and A. Rogers. "Aerodynamics of Wind Turbines". In: *Wind Energy Explained*. John Wiley & Sons, Ltd, 2009. Chap. 3, pp. 91–155. ISBN: 9781119994367. DOI: https://doi.org/10.1002/9781119994367.ch3.

[11] K. Bruhn, S. Lorensson, and J. Svensson. *Wind Power - A Renewable Energy Source in Time*. Comprehensive report. 2009.

[12] J. Manwell, J. McGowan, and A. Rogers. "Electrical Aspects of Wind Turbines". In: *Wind Energy Explained*. John Wiley & Sons, Ltd, 2009. Chap. 5,

pp. 205–256. ISBN: 9781119994367. DOI: https://doi.org/10.1002/9781119994367.ch5.

[13] S. Fogelström. *Wind Power*. Presentation. Presented on September 9, 2024. Sept. 2024.

[14] S. Kollmannsberger et al. "Fundamental Concepts of Machine Learning". In: *Deep Learning in Computational Mechanics: An Introductory Course*. Cham: Springer International Publishing, 2021, pp. 5–18. DOI: 10.1007/978-3-030-76587-3_2.

[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[16] F. Chollet. *Deep Learning with Python*. OCLC: ocn982650571. Shelter Island, New York: Manning Publications Co, 2018, p. 361. ISBN: 978-1-61729-443-3.

[17] S. Kollmannsberger et al. "Neural Networks". In: *Deep Learning in Computational Mechanics: An Introductory Course*. Cham: Springer International Publishing, 2021, pp. 19–45. DOI: 10.1007/978-3-030-76587-3_3.

[18] T. Zhang. "Decision Tree and K-Nearest-Neighbors (KNN)". In: *An Introduction to Materials Informatics : The Elements of Machine Learning*. Singapore: Springer Nature Singapore, 2025, pp. 117–140. DOI: 10.1007/978-981-99-7992-9_5.

[19] T. Zhang. "Neural Networks". In: *An Introduction to Materials Informatics : The Elements of Machine Learning*. Singapore: Springer Nature Singapore, 2025, pp. 245–323. DOI: 10.1007/978-981-99-7992-9_9.

[20] L. Fahrmeir et al. "Statistik: Der Weg zur Datenanalyse". In: 9th ed. Springer, 2023. Chap. 10, pp. 111–178. ISBN: 978-3-662-67526-7.

[21] S. Fogelström. *Chalmers Wind Turbine at Björkö: Description, Control and Research Facilities*. Tech. rep. Chalmers University of Technology, 2023.

[22] S. Fogelström. *Chalmers Wind Turbine*. https://www.chalmers.se/en/departments/e2/resources-and-collaboration/chalmers-wind-turbine/. Accessed on June 14, 2025; last updated on August 15, 2024. Feb. 2024.

[23] S. Fogelström et al. *ENM097 Future renewable based power system: Wind energy assignment*. 2023.

[24] A. Wickstrom et al. *SWE-0068-C Hub design for Hönö IV, three bladed individual pitch rotor*. Tech. rep. Chalmers tekniska högskola, Aug. 2018.

[25] S. Fogelström and O. Carlson. *Tabell 16 NREL controller 2022*. Excel file. Last modified on 2023-06-05. 2021.

[26] M. Ellsén. *MeasFileStructure_B1_CL6_Rev0*. Word file. 2024.

[27] Scikit-learn developers. *sklearn.ensemble.HistGradientBoostingRegressor*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html#sklearn.ensemble.HistGradientBoostingRegres
Accessed: 2025-06-02.

[28] Scikit-learn developers. *sklearn.neural_network.MLPRegressor*. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html. Accessed: 2025-06-02.

A

Appendix 1

A.1 Main Principle of Method

In the following, the core of the main method is shown as PYTHON code.

```
# create arrays for control parameters and look up table
control_parameters = {}
lut = []
# initialize variables
region_start = 0
regions_preliminary = []
mean_slope_regionsofar = bin_slopes[0]
tolerance = 150   # empirically seemingly reasonable
   value
# loop over all bin_slopes
for i in range(1, len(bin_centers)):
    # check whether slope in point i is similar to mean
       slope of previous slopes
    if i == len(bin_slopes):   # length of bin_slopes
       is one less than length of bin_centers and
       bin_means
        # calculate mean slope of all slopes in region
        mean_slope = np.mean(bin_slopes[region_start :
           i])
        number_slopes = i - region_start
        # calculate bin_centers value of point in
           middle of region
        rotorspeed_center =
           np.mean(bin_centers[region_start : i + 1])
           # to get end point of slope / last center
           point
        Tel_center = np.mean(bin_means[region_start : i
           + 1])   # to get end point of slope / last
           mean point
        number_centers = i - region_start + 1
        # save region
        if abs(i - region_start) > 1:
```

```
            regions_preliminary.append((rotorspeed_center,
                Tel_center, mean_slope, number_slopes,
                number_centers))
        elif abs(bin_slopes[i] - mean_slope_regionsofar) <
            tolerance:
            mean_slope_regionsofar =
                (mean_slope_regionsofar * (i - 1 -
                region_start) + bin_slopes[i]) / (i -
                region_start)
        else:
            if i + 1 < len(bin_slopes) and
                abs((bin_slopes[i] + bin_slopes[i + 1]) / 2
                - mean_slope_regionsofar) < tolerance:  #
                accept exeptions
                mean_slope_regionsofar =
                    (mean_slope_regionsofar * (i - 1 -
                    region_start) + (bin_slopes[i] +
                    bin_slopes[i + 1]) / 2) / (i -
                    region_start + 1)
                i += 1
            elif abs(i - region_start) < 2:   # at least
                two points with similar slope per region
                region_start = i   # if not, interval is
                    skipped
                mean_slope_regionsofar = bin_slopes[i]
            else:
                # calculate mean slope of all slopes in
                    region
                mean_slope =
                    np.mean(bin_slopes[region_start : i -
                    1])   # i-1 because slope at i should
                    not be included in the region
                number_slopes = i - region_start
                # calculate bin_centers value of point in
                    middle of region
                rotorspeed_center =
                    np.mean(bin_centers[region_start : i])
                Tel_center = np.mean(bin_means[region_start
                    : i])
                number_centers = i - region_start
                # save region
                regions_preliminary.append((rotorspeed_center,
                    Tel_center, mean_slope, number_slopes,
                    number_centers))
                # set start of next region
                region_start = i
```

```
            mean_slope_regionsofar = bin_slopes[i]
    # merge regions if the difference between
       mean_slopes is too small
    regions = []
    j = 0
    while j < len(regions_preliminary):
        rotorspeed_center, Tel_center, mean_slope,
           number_slopes, number_centers =
           regions_preliminary[j]
        k = j + 1
        while k < len(regions_preliminary):
            next_rotorspeed_center, next_Tel_center,
               next_mean_slope, next_number_slopes,
               next_number_centers =
               regions_preliminary[k]
            if abs(mean_slope - next_mean_slope) <
               tolerance:
                total_slopes = number_slopes +
                   next_number_slopes
                total_centers = number_centers +
                   next_number_centers
                mean_slope = (mean_slope *
                   number_slopes + next_mean_slope *
                   next_number_slopes) / total_slopes
                rotorspeed_center = (rotorspeed_center
                   * number_centers +
                   next_rotorspeed_center *
                   next_number_centers) / total_centers
                Tel_center = (Tel_center *
                   number_centers + next_Tel_center *
                   next_number_centers) / total_centers
                number_slopes = total_slopes
                number_centers = total_centers
                k += 1      # skip next line because it
                   is merged
            else:
                break
        regions.append((rotorspeed_center, Tel_center,
           mean_slope, number_slopes, number_centers))
        j = k
```

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY