



Universitetet  
i Stavanger

Faculty of Science and Technology

## MASTER'S THESIS


Study program/Specialization:  
Mechanical and Structural Engineering and Materials  
Science

Spring semester, 2019

Open / ~~Restricted access~~

Author:

Eirik Thunold Halvorsen

  
(Signature of author)

Faculty supervisor: *Hirpa G. Lemu*, UiS

Co-supervisor: *Knut Erik Teigen Giljarhus*, UiS

External advisor(s): .....

Thesis title:

CFD Analysis of Topographically Induced Turbulence and its Effect on a Fatigue Life Analysis of a Wind Turbine Yaw Gear

Credits (ECTS): 30

Keywords:

Wind turbines, Topography, Turbulence, CFD,  
OpenFOAM, Fatigue life, NREL FAST v8, Rainflow  
counting, Yaw drive

Number of pages: 62

+ Supplemental material/other: 36

Stavanger, 13<sup>th</sup> June 2019

date / year

(This page is intentionally left blank)

## Preface

This thesis is my final project for the degree of Master of Science in Engineering structures and Materials Science, with a specialisation in mechanical systems. The thesis covers 30 ECTs, and is the closure of a 5 year study at the University of Stavanger.

I would like to use this opportunity to express my gratitude to everyone who has helped me complete this thesis.

To my supervisor Prof. Hirpa G. Lemu. Thank you for helping me find an interesting and relevant topic, and thank you for your guidance and support. Thank you Prof. Knut Erik Giljarhus for your help with the CFD simulations and for guiding me when the simulations got stuck, and for answering all my questions. Thank you Prof. Charlotte Obhrai for the counsel sessions and for guiding me on the right path.

I would also like to thank Siemens Gamesa Renewable Energy for this interesting task and I hope this thesis will prove useful in the future.

Thank you to the people at IEA for providing the data from their reference wind turbine. Without their work this thesis would be a lot less complete.

And finally, I would like to thank my family and Kari Elise. Thank you for your support during this thesis and during my five years of study.

Eirik Halvorsen

June 13, 2019

## **Abstract**

This master thesis contains a case study of a failure of a yaw pinion spur gear for a wind turbine in Northern-Europe. The yaw gear has experienced unsuspected loads and vibrations which has caused a fatigue failure. Turbulence, due to either terrain or other turbines is the suspected culprit.

Two computer softwares have been used to simulate the conditions of the wind farm. CFD software OpenFOAM along with a turbulence model have been used to find where in the terrain the turbulent kinetic energy is most severe, and FAST v8 aeroelastic simulator to calculate the aero-elastic force response of a 3.35 MW IEA virtual reference wind turbine. A similar turbine to the Siemens 3.0DD turbine platform installed at the site.

The simulation results indicate that the positions of the wind turbines are located in areas possibly prone to turbulent wind conditions, and that exceed the technical specifications they were designed for.

The fatigue calculated for the different turbulent wind conditions show a significant amount of damage accumulation for higher turbulence intensities and is potentially the cause for the pinion gear failure.

This thesis present tools that can be used to analyse the turbulence based on the profile of the terrain, along with tools to evaluate the fatigue life of a component based on material properties and load history. This thesis show the importance of proper turbine placement by calculating and visualising the effect of the turbulence.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Drive-train systems . . . . .	4
1.3	Design Standards . . . . .	4
1.4	Yaw system and turbine control . . . . .	6
1.5	Scope . . . . .	6
1.6	Report structure . . . . .	8
<b>2</b>	<b>Theory</b>	<b>9</b>
2.1	Fatigue loading and structural stresses . . . . .	9
2.2	Wind Turbine Loads . . . . .	12
2.2.1	Steady loads . . . . .	12
2.2.2	Cyclic loads . . . . .	13
2.2.3	Transient loads . . . . .	13
2.2.4	Stochastic Loads . . . . .	13
2.2.5	Resonance-Induced Loads . . . . .	13
2.3	Turbulence . . . . .	13
2.4	Fluid Dynamics . . . . .	15
2.5	Actuator Disk Model . . . . .	17
2.5.1	Terrain roughness parameter . . . . .	17
2.5.2	Atmospheric boundary conditions . . . . .	18
2.6	Aerodynamic forces . . . . .	19
2.6.1	Wind power calculations . . . . .	19
2.6.2	Airfoils and general concept . . . . .	24
2.7	Blade Element Momentum Theory . . . . .	24
2.8	Geometrically exact beam theory . . . . .	26
2.9	Rainflow Counting . . . . .	27
2.10	Other methods of analysing random time history . . . . .	29
2.10.1	Peak count method . . . . .	31
2.10.2	Peak between mean crossing count method . . . . .	31
2.10.3	Range count method . . . . .	31
<b>3</b>	<b>OpenFOAM</b>	<b>32</b>

<b>4</b>	<b>FAST v8</b>	<b>33</b>
4.1	Subsystems . . . . .	33
4.1.1	InflowWind . . . . .	33
4.1.2	TurbSim . . . . .	33
4.1.3	AeroDyn . . . . .	35
4.1.4	BeamDyn . . . . .	35
4.1.5	ElastoDyn . . . . .	35
4.2	Output file . . . . .	35
<b>5</b>	<b>Method</b>	<b>37</b>
5.1	CFD Simulation . . . . .	37
5.1.1	Environmental loads . . . . .	37
5.1.2	Case study . . . . .	38
5.1.3	Mesh generation . . . . .	38
5.1.4	SimpleFoam . . . . .	40
5.1.5	Post processing . . . . .	41
5.2	FAST v8 simulation . . . . .	41
5.3	Rainflow Counting analysis . . . . .	42
<b>6</b>	<b>Results and Discussion</b>	<b>47</b>
6.1	Wind measurements . . . . .	47
6.2	CFD Simulations . . . . .	48
6.3	FAST v8 Simulations . . . . .	53
6.4	Rainflow analysis . . . . .	56
<b>7</b>	<b>Conclusion and future work</b>	<b>59</b>
7.1	Conclusion . . . . .	59
7.2	Future work . . . . .	59
	<b>References</b>	<b>60</b>
	<b>Appendix A OpenFOAM input files</b>	<b>1</b>
	<b>Appendix B FAST v8 input files</b>	<b>16</b>
	<b>Appendix C Technical Yaw drive data</b>	<b>30</b>
	<b>Appendix D MATLAB script for Rainflow counting algorithm</b>	<b>34</b>
	<b>Appendix E MATLAB script for plotting simulated FAST data</b>	<b>34</b>

## List of Figures

1.1	Share on total down time per system of onshore wind turbines as published by different initiatives. (Pfaffel et al. 2017)	1
1.2	Constructed wind-farms in Norway, the size of the markers differentiate between the power output of the wind-farms, ranging from 10, 10-100 and >100 MW	3
1.3	This figure show upcoming wind-farms projects in Norway, along with some declined projects. Figure is intended to give and overview of the wind-farm activity in Norway.	3
1.4	Frequencies of failures of the sub-assemblies and typical down-time per failure for different power classes.(Faulstich et al. 2008)	4
1.5	Modular drive-train configuration (Oyague 2009).	5
1.6	Model of the vibrational behaviour of the yaw system (Hau 2006)	7
1.7	A figure of the Yaw gear assembly provided by Siemens Gamesa of a 3.0DD wind turbine.	8
2.1	S-N diagram. Where N is cycles, and $\sigma$ is the stress. Figure from Lalanne 2014	10
2.2	An example of a yaw drive system configuration. Picture from M.-G. Kim and Dalhoff 2014	10
2.3	A representation of how the steady wind profile is influenced by a random turbulence to produce an instantaneous wind field.	12
2.4	Types of free turbulent flows. The mixing of different speeds in a fluid cause turbulence (Versteeg et al. 1995)	14
2.5	A typical jet flow example with regions of differing fluid velocities causing turbulence. The vortices will increase in size from small to large eddies covering a significant portion of the flow (Versteeg et al. 1995)	14
2.6	Circular tube of air flowing through an ideal wind turbine. Top curve representing the speed of the wind and bottom curve representing the pressure in the air parcel (Muyeen et al. 2009)	20
2.7	A change in wind velocity and volume, before and after passing the wind turbine blades. (William Shepherd 2011)	21
2.8	A graphical representation of the total power extractable, compared to the reduction of wind-speed before and after passing the windmill.	22
2.9	Theoretical maximum power coefficient as a function of tip speed ratio for an ideal horizontal axis wind turbine, with and without wake rotation (Manwell et al. 2009).	23
2.10	Aerodynamic forces acting on a section of the air-foil. Drawing from Hau 2006.	24
2.11	Representation of the blade elements used by the Beam Element Momentum Theory (Manwell et al. 2009).	25
2.12	Reference coordinate system used by BeamDyn.	27
2.13	A typical example of the Rainflow counting method. Imagine a raindrop falling on a “pagoda” roof (Lalanne 2002).	28
2.14	The description of the gear tooth model. Figure from Lemu 2016	30
2.15	Visualisation of the moments and forces acting on the gear teeth. Figure from Lemu 2016	30

3.1	Structure of the case folder in OpenFOAM, containing the necessary folders. Each folder contain text files describing boundary conditions, solvers, terrain files (in this case), etc. . . . . .	32
4.1	Example of TurbSim grids as implemented in AeroDyn . . . . .	34
4.2	The degrees of freedom for a two or three bladed wind turbine in ElastoDyn . . .	36
5.1	Ishikawa or Fish-bone diagram, commonly used to identify the case and effects of a failure. Figure based on <i>OnyxInsight, Root cause analysis</i> 2016. . . . .	37
5.2	Paraview representation of the DEM. file using the built in DEMreader function. A 2D plane containing information about the height of each point. Blue to red are low to high respectively. The height is meters above sea-level. . . . .	39
5.3	The filter “Tetrahedralize” changes the grid to contain triangles instead of squares. The triangles are used by the “Extract Surface” filter to create the .STL file. . .	40
5.4	The grid generated by snappyHexMesh. The grid gets more refined closer to the ground and to the wind turbines (represented as squares). . . . .	41
5.5	.STL file containing topographical data, used by OpenFOAM to simulate the wind conditions. . . . .	42
5.6	A chart of how the different software’s and the different modules that are used to produce the output files of FAST v8. . . . .	45
6.1	The turbulence intensity calculated for every 10-minute measurement. 16% Turbulence intensity is shown as a black line, and is the IEC design standard for IIA turbines. . . . .	47
6.2	A windrose visualising the wind directions. Generally wind is blowing from the north, north-west and east, south-east. MATLAB code borrowed from <i>Wind Rose</i> 2019 . . . . .	48
6.3	Terrain and location of Turbine 1 . . . . .	49
6.4	The figure show the turbulent kinetic energy, $k$ , as the wind progress over the terrain in Figure 6.3. The turbine in the figure is not implemented in the simulation but is added to show scale. . . . .	49
6.5	This figure shows the location of wind turbine 2 and 3. Its is located at a top of a hill, behind an incline. However, the incline is more shallow than compared to Turbine 1. The wind travel predominantly from left to right. Wind turbine 2 is situated as the second in a row of turbines. There will be wake effects when the wind is blowing from the east due to the positioning of Turbine 3. Note that the scale is for elevation. . . . .	50
6.6	The figure shows the turbulent kinetic energy, $k$ , as the wind progress over the terrain in Figure 6.5. The wind is moving from left to right at 10 m/s. The figure shows an increase of turbulent kinetic energy as the wind approach peaks in the terrain. . . . .	50
6.7	Top down view of the wind turbines and terrain, showing the turbulent wake behind the wind turbine. The turbulent wake behind the wind turbine is just a representation of how a wake looks. It does not necessarily have the same turbulent magnitude or shape. . . . .	51
6.8	The results from FAST v8 with 5% turbulence intensity. . . . .	53
6.9	The results from FAST v8 with 16% turbulence intensity. . . . .	54
6.10	The results from FAST v8 with 20% turbulence intensity. . . . .	55



6.11	Turbulence intensity plotted against the mean (top figure) and amplitude (bottom figure) of the moment on the yaw bearing. . . . .	56
6.12	Accumulation of damage for 25 years for the 24 different simulations, each with increasing turbulence intensity. . . . .	58

## List of Tables

1.1	Basic parameters for wind turbine classes. See NEK-IEC-61400-1 2019 . . . . .	5
2.1	Values for parameters in the $k - \epsilon$ solver. Default value for $\sigma_\epsilon$ 1.3 (Versteeg et al. 1995). Modified according to Hargreaves et al. 2007 . . . . .	16
2.2	The roughness parameter for some terrain types according to DNVGL-RP-C205 2017. . . . .	17
2.3	Boundary conditions for the different fields. ABL (atmBoundaryLayer) . . . . .	18
2.4	The maximum theoretical Power (kW) extractable by a wind-turbine, with different wind-speed and cross-sectional areas. . . . .	23
4.1	Valid Turbsim Spectral Models (Jonkman et al. 2012). . . . .	34
5.1	The mean and maximum wind velocities for each month in 2017 and 2018. The wind measurements are taken from the wind mast located at the site. . . . .	38
5.2	Degrees of freedom set in FAST v8 simulations in the ElastoDyn module. . . . .	42
5.3	Damage fraction for one impact in each simulation case. Ref S.-W. Kim et al. 2017	44
6.1	Wind measurements taken from a wind mast at the wind farm. $I$ is calculated using Eq. (1.1) . . . . .	47
6.2	The turbulent kinetic energy compared to the wind speed for each of the two wind turbines. The wind turbines experience no wake effect as the surrounding wind turbines have been removed. The turbulence intensity $I$ is calculated using Eq. (2.16) . . . . .	51
6.3	The damage accumulated for 1 hour simulation, not taking into account impact damage. Wind speed at 10 m/s, simulated in FAST v8. . . . .	57

# Nomenclature

The next list describes several abbreviations and symbols that will be later used within the thesis.

## Abbreviations

<i>ASTM</i>	American Society for Testing and Materials
<i>BEM</i>	Blade Element Momentum theory
<i>CFD</i>	Computational Fluid Dynamics
<i>DNVGL</i>	Det Norske Veritas og Germanischer Lloyd
<i>DOE</i>	United States Department Of Energy
<i>DOF</i>	Degrees Of Freedom
<i>FAST</i>	Fatigue, Aerodynamics, Structures, and Turbulence
<i>FFT</i>	Fast Fourier Transform
<i>FOAM</i>	Field Operation And Manipulation
<i>GEBT</i>	Geometrically Exact Beam Theory
<i>HAWT</i>	Horizontally Aligned Wind Turbine
<i>IEA</i>	International Energy Agency
<i>IEC</i>	International Electrotechnical Commission
<i>NEK</i>	Norsk Elektroteknisk Komite
<i>NREL</i>	Natinal Renewable Energy Laboratory
<i>NWTC</i>	National Wind Technology Centre
<i>RANS</i>	Reynolds Averaged Navier Stokes
<i>SGRE</i>	Siemens Gamesa Renewable Energy
<i>SIMPLE</i>	Semi-Implicit Method for Pressure Linked Equations
<i>WMEP</i>	Scientific Measurement and Evaluation Program

## Symbols

$\alpha$	Power-law exponent for the wind speed profile
$\epsilon$	Turbulence dissipation rate
$\kappa$	Von Karman's constant
$\nabla$	Gradient operator or a vector field operator for the Navier Stokes equations
$\rho$	Density
$\sigma_{10}$	Standard deviation of the 10 minute wind speed

$a$	Induction factor
$C_p$	Power coefficient of turbine
$C_t$	Torque coefficient of the turbine
$F_t$	Tangential force acting between the interactions of spur gears
$h$	Planck constant
$H_{op}$	Hours of operation for a turbine per year
$I$	Turbulence intensity
$k$	Turbulent kinetic energy
$m$	Gear tooth modulus
$n_L$	Number of load cycles for the wind turbine lifetime
$n_{rotor}$	The average rotational speed of the turbine rotor
$P$	Power, rate of work
$Q$	Torque
$Re$	The Reynolds number
$T$	Thrust
$t$	Time
$U$	The instantaneous wind speed
$u, v$ and $w$	Instantaneous wind speeds for x, y and z respectively
$u_*$	Friction velocity
$U_\infty$	The wind speed of the ambient or surrounding air, not affected by the turbine
$U_d$	Wind speed at the rotor
$U_{10}$	The mean wind speed for a 10 minute recording
$u_{ref}$	A reference wind speed used in OpenFOAM boundary condition
$Y$	Lewis shape form factor for spur gears
$Y$	Years of operation
$z_0$	Roughness parameter for the topography

# 1. Introduction

Installation of wind turbines in Norway has increased significantly in the last years with new wind-farms all over Norway. Figure 1.2 and 1.3 show the current overview of the constructed wind farms, to the left, and to the right, wind farms that are under-construction, have been granted approval, denied or under review as of 12.02.2019.

## 1.1. Background

Looking at past studies of wind turbine downtime, a clear portion of the downtime can be attributed to the transmission system and drive-train, along with the rotor systems and electrical components (Faulstich et al. 2008 and 2011, Pfaffel et al. 2017), shown in Figure 1.4 and Figure 1.1.

The difference in the definition of a failure varies from initiative to initiative, and downtime are not exact and not necessarily comparable. However, the trend show in Figure 1.1, indicates that a significant amount of failures occur in the mechanical systems of the wind turbine, where the downtime per failure is also most significant.

This is also backed up by an older study conducted from 1989 to 2006, on onshore wind turbines located in Europe, performed by the Scientific Measurement and Evaluation Program (WMEP). A large amount of failures occur in the electrical system for higher power capacity wind turbines compared to lower capacity, while the failure rate of the drive-train system is barely affected as shown in Figure 1.4. Close to a third of the annual failure rate is accounted for by the electrical system alone. In the report of WMEP Faulstich et al. (2008) it is also briefly mentioned that even though the failure rates of the electrical system is high, the cost and downtime is mainly attributed by the failures of the drive-train, yaw-drive and the structural system.

Several studies have been conducted evaluating the wind turbine performance under turbulent conditions. These studies have used digital models, as well as experimental models in wind

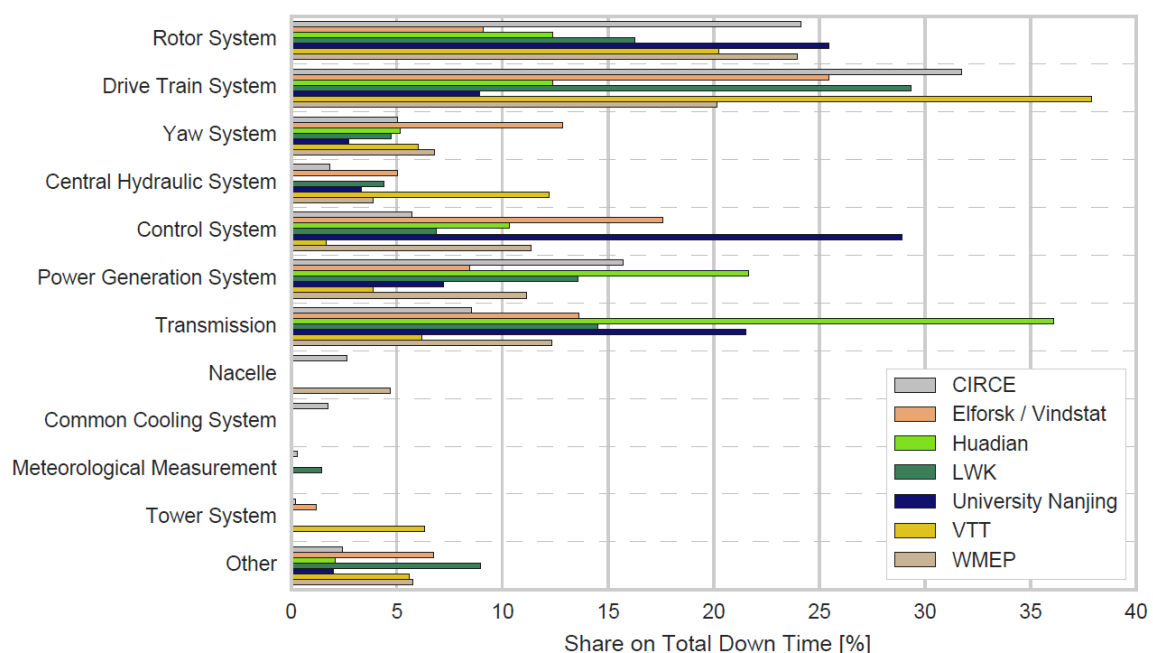


Figure 1.1: Share on total down time per system of onshore wind turbines as published by different initiatives. (Pfaffel et al. 2017)

tunnels using grids to generate turbulent wind behaviour (Al-Abadi et al. 2016). The wind turbines installed and maintained by Siemens Gamesa Renewable Energy (AS) (SGRE), are designed and tested in Esbjerg, Denmark. A recent failure of the yaw gear, controlling the orientation in a newly installed wind turbine (less than two years), has led to an investigation as to why it failed. The hypothesis is that the turbulence created by the surrounding hills and valleys and other turbines lead to unexpected movements and fatigue which have caused the yaw gear to break.

For an offshore wind turbine farm to be reliable and cost-effective, the need for maintenance need to be reduced. Wind turbines that need a visit from a technician introduce a significantly higher cost and downtime than an on-shore wind turbine. The need for larger weather-windows, travel time, ships, equipment etc. decrease the availability for maintenance. Although electrical components is the cause for the largest amount of failures per year, the repair time is short. The structural system like the drive-train, yaw drive and turbine blades require a significant amount of time and effort to repair and understanding the causes of these failures is important to be able to prevent them.

Siemens Gamesa is one of the leading suppliers of wind power solutions, and have installed their products in over 90 countries, and have a total energy capacity of over 89 GW world wide. The purpose of this report is to give an understanding of the challenges related to wind turbine placement in a wind farm with respect to the turbulence induced from the terrain, to understand the effect of the wind turbine placement, according to both the topographical surroundings, and the effect of other wind turbines. The case study of this thesis specifically look at two turbines that are placed in a wind farm in Northern-Europe where the terrain can typically produce turbulence. There is a strong interest in understanding the effect of the terrain and how the fatigue life of the wind turbines are effected. This thesis will focus especially on the yaw drive motors and the pinion gears.



Figure 1.2: Constructed wind-farms in Norway, the size of the markers differentiate between the power output of the wind-farms, ranging from 10, 10-100 and >100 MW

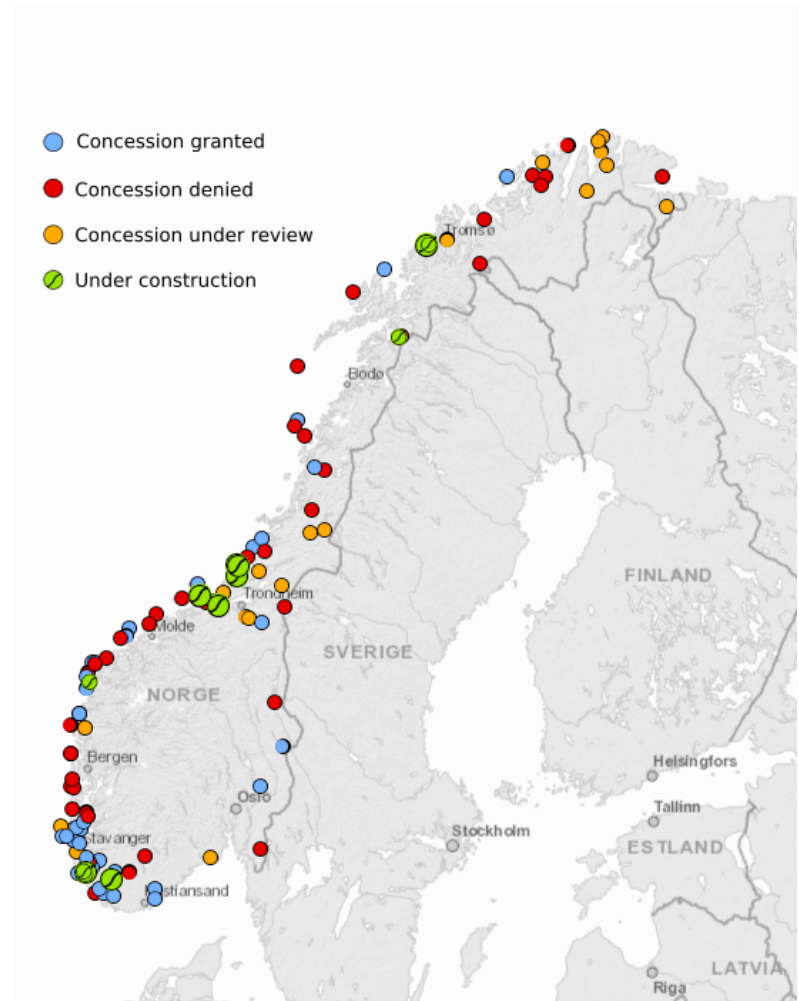


Figure 1.3: This figure show upcoming wind-farms projects in Norway, along with some declined projects. Figure is intended to give and overview of the wind-farm activity in Norway.

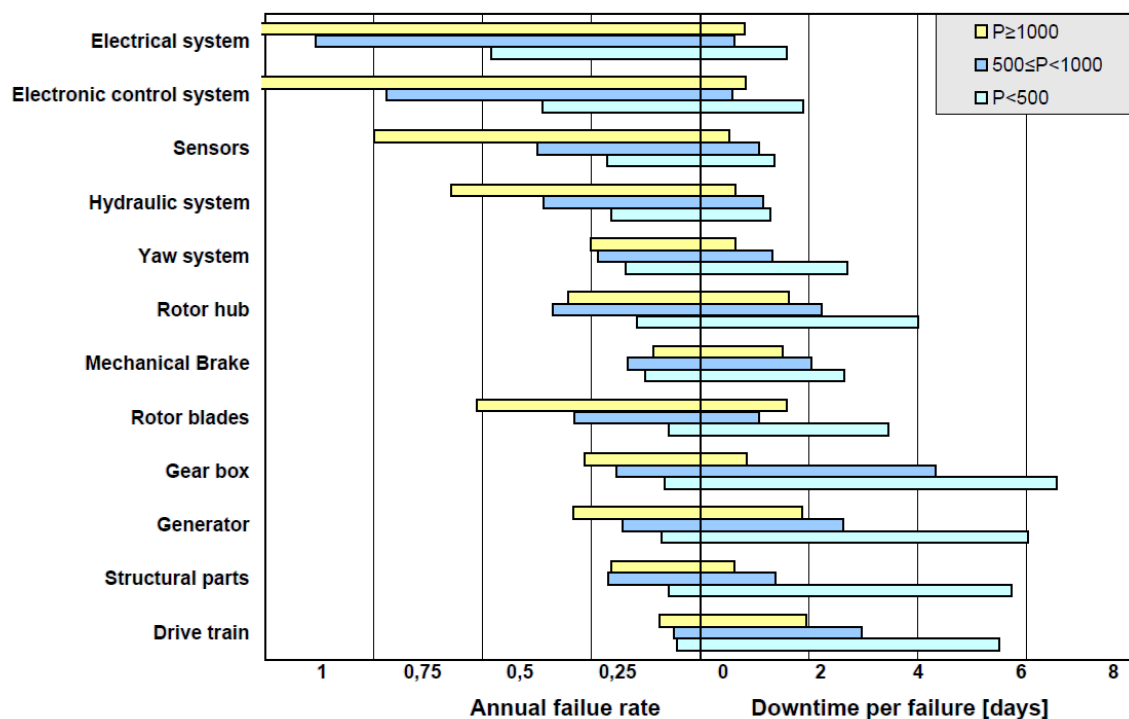


Figure 1.4: Frequencies of failures of the sub-assemblies and typical down-time per failure for different power classes. (Faulstich et al. 2008)

### 1.2. Drive-train systems

The drive-train system in a wind turbine is designed to convert the rotational energy stored in the blades, and turn it into electricity. The typical wind turbine drive-train system consist of 7 parts, shown in Figure 1.5. The hub, holding the three wind turbine blades, the main bearing, allowing the transfer of power from the hub and into the main Shaft. The input of the main shaft goes into the gearbox, where it is converted to a more usable output, into the high-speed shaft. Located on the high-speed shaft is the Break and at the end, the Generator. The generator used in a wind turbine cannot receive the low rotation, high torque input generated by the rotation of the wind turbine blades so a gearbox is used to transform the input through a system of gears to produce a low torque, high speed output that the generator can use.

The gearbox is the most mechanically complex part in the drive-train system and also responsible for the most costly repairs and longest operational down-time. Knowledge about the operational conditions of wind turbine gearboxes and drive-trains in general is a much studied subject, and the requirements for long lasting and efficient wind turbines is only increasing. Wind turbines are usually rated by the power they can produce. For example, the Siemens Gamesa turbines in this thesis is a 3.0DD MW wind turbine, which can be tuned to produce between 3.0 and 4.2 MW.

### 1.3. Design Standards

The International Electrotechnical Commission (IEC) develop standards for which wind turbines experiencing certain wind speeds and turbulence should be designed.

The Siemens Gamesa wind turbines at the wind farm have a classification according to Table 1.1 as IIA.



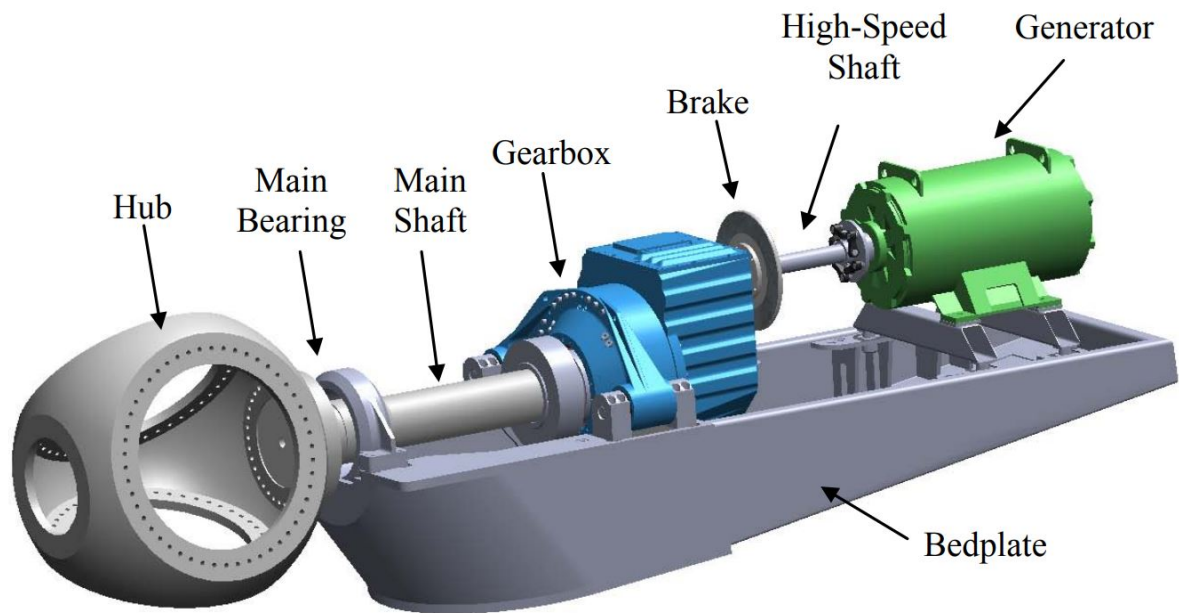


Figure 1.5: Modular drive-train configuration (Oyague 2009).

Table 1.1: Basic parameters for wind turbine classes. See NEK-IEC-61400-1 2019

Wind turbine class		I	II	III	S
$V_{ave}$	(m/s)	10	8.5	7.5	Values specified by the designer
$V_{ref}$	(m/s)	50	42.5	37.5	
	Tropical (m/s) $V_{ref,T}$	57	57	57	
$A^+$	$I_{ref}(-)$		0.18		
A	$I_{ref}(-)$		0.16		
B	$I_{ref}(-)$		0.14		
C	$I_{ref}(-)$		0.12		
The parameter values apply at hub height and $V_{ave}$ is the annual average wind speed; $V_{ref}$ is the reference wind speed average over 10 min; $V_{ref,T}$ is the reference wind speed average over 10 min applicable for areas subjected to tropical cyclones; $A^+$ designates the category for very high turbulence characteristics; A designates the category for higher turbulence characteristics; B designates the category for medium turbulence characteristics; C designates the category for lower turbulence characteristics; $I_{ref}$ is a reference value of the turbulence intensity.					

Turbulence intensity  $I$  according to the DNV GL standards (DNVGL-RP-C205 2017) is

$$I = \frac{\sigma_{10}}{U_{10}} \quad (1.1)$$

where  $\sigma_{10}$  is the standard deviation, and  $U_{10}$  is the mean wind speed of a 10 minute wind recording.

## 1.4. Yaw system and turbine control

The orientation of wind turbine rotors and nacelle are generally done in three different ways.

1. aerodynamic yawing, using wind vanes or fan-tail wheels
2. free yawing, locating the rotors down-wind
3. active yawing, using a motorised yaw drive

The yaw-drive is built in the interface between the nacelle and the tower-head, and is used to control the orientation of the nacelle, hub and rotor blades. The Yaw-control system, along with the control system for the entire wind turbine is advanced and automated. There are usually sensors (rain, wind speed, wind direction, vibration etc.) located at the top of the turbine on the nacelle. These sensors provide the information the wind turbine needs to control the cut-in, cut-off, pitch angle and yaw-orientation.

Yaw-mechanisms installed in turbines varies, but generally for large wind turbines, the active yawing, with a motorised drive is most used. The yaw drive pinion motors are common to all types of motorised yaw drives. The pinion motors are usually placed on the inside of the yaw gear, and are mounted within the housing of the nacelle. The torque that these pinion motors can produce are up to 300 kNm. With a gear ratio of up to 1:3000. The individual weight of each motor range from 350 to 800 kg. The Siemens Gamesa 3.0DD turbine yaw drive have installed eight or ten electric pinion drive motors, see Figure 1.7. The specifications of yaw drives installed in the turbine can be found in Appendix C.

The vibrational behaviour of a complex system like a wind turbine with several degrees of freedom need to be treated as a whole system. Complex vibrational coupling modes exist in the system that cannot be explained looking at the natural frequencies of the individual components alone. However, it can be useful to develop a mathematical model of the system capable of describing the vibrational behaviour of the system. The first step is to look at the basic vibrational character of the turbine, so that critical vibrational frequency modes can be recognised. Calculating first, and some higher order natural frequencies and vibration modes can be calculated in a stand-still condition.

A mathematical model can be developed for such a system, using Figure 1.6, as a guide. The springs, moments, forces etc. can be modelled analytically, using theory from mechanical vibrations.

## 1.5. Scope

The scope for this thesis is to investigate the cause for a yaw-gear failure of a wind turbine located in a wind farm in Northern-Europe. The hypothesis is that due to terrain and wind conditions, the wind turbine experience an increase in turbulence, which lead to an increase in vibrations and stresses on the yaw gears.

The first objective is to use a model of the terrain to simulate the wind conditions and to compare to the results of the CFD simulations to the data gathered from a wind mast located at the site. This thesis will outline the relevant theory that have been used to simulate the turbulence intensity using CFD through OpenFOAM, and give a step by step process of the simulation.

Second, using FAST v8 developed by NREL and a virtual reference wind turbine produced by IEA, calculate the forces acting on the wind turbine.

The third and final step is to use the forces calculated with FAST v8, to calculate the fatigue life of the wind turbine yaw gears.

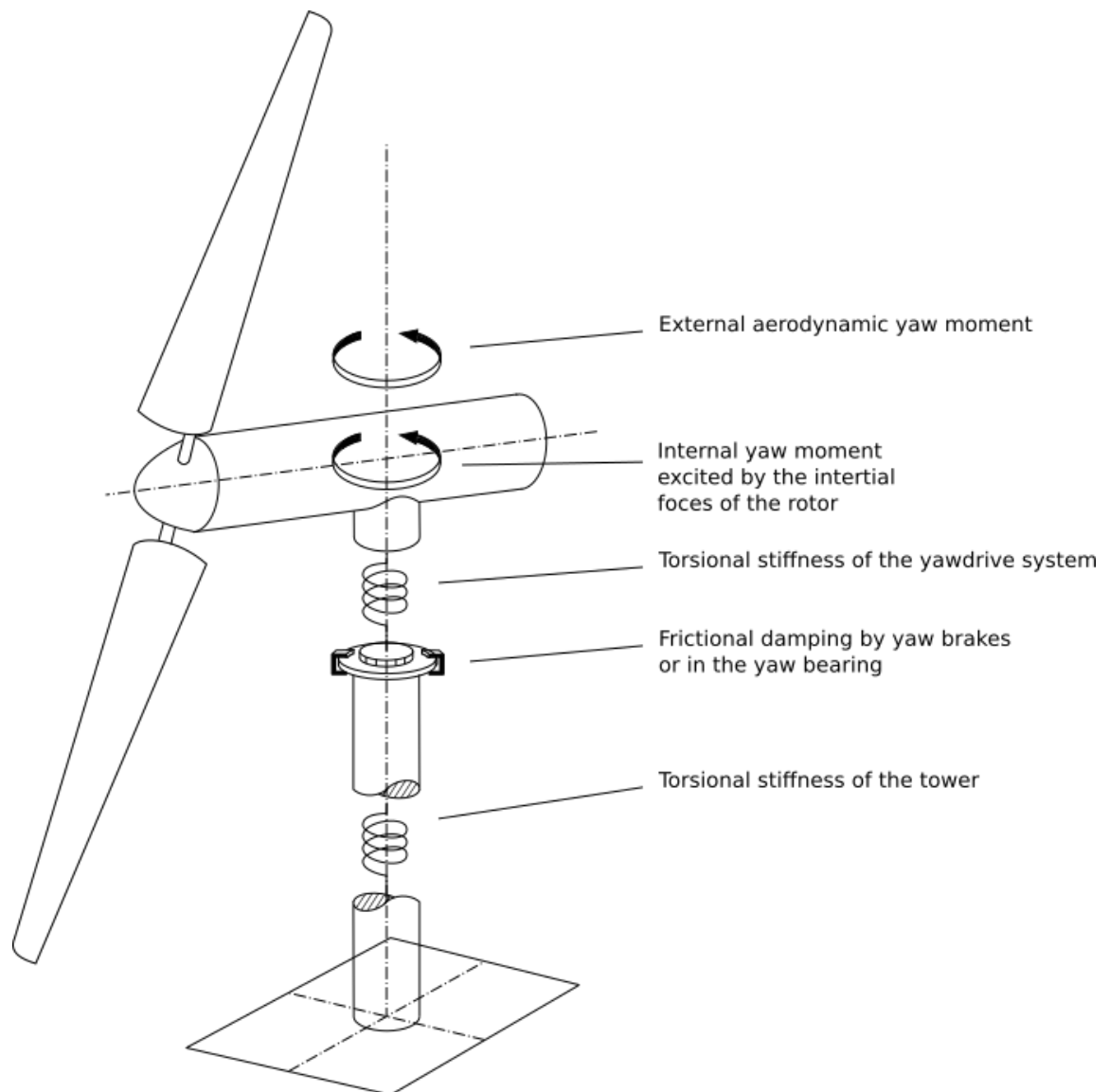


Figure 1.6: Model of the vibrational behaviour of the yaw system (Hau 2006)

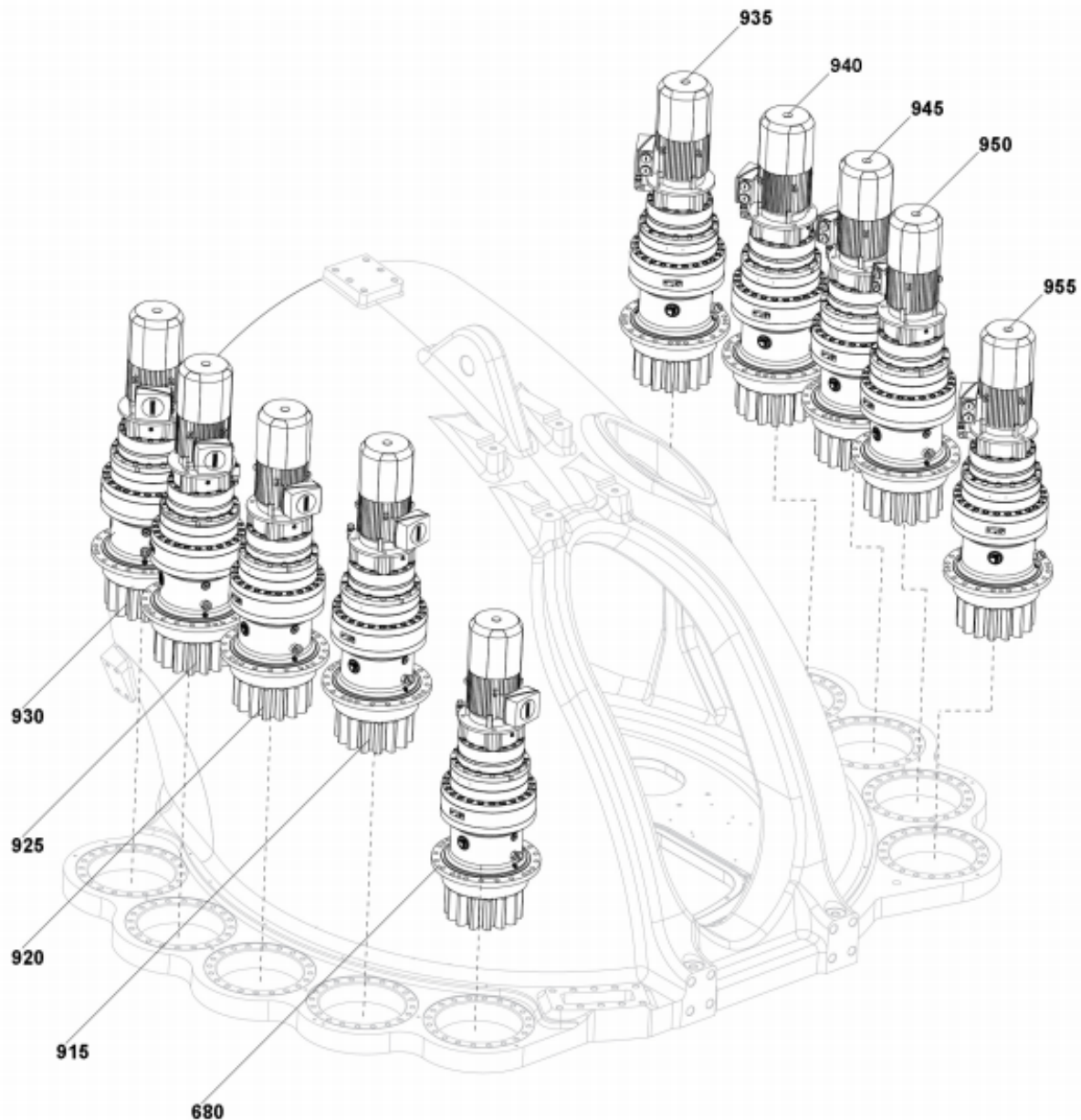


Figure 1.7: A figure of the Yaw gear assembly provided by Siemens Gamesa of a 3.0DD wind turbine.

## 1.6. Report structure

Including this introduction section, the report is divided into six parts. The theory section present an discuss the most relevant topics that is needed to understand the topics of the thesis. The OpenFoam and FAST v8 sections are introductions to the software's and what they are used for in this thesis. Method section is essentially a step by step process of how the results of this thesis were achieved, along with explanations of figures and tables. The results and discussion part take each of the results and discuss in detail as much as possible. Finally, a summarised conclusion and future work is discussed.

There are three main topics of this thesis, which are also divided into these chapters. The CFD analysis of the terrain, the simulation of a reference wind turbine under different turbulence intensities, and finally a fatigue model using the results of the previous simulations. The goal is to connect them all into a single conclusion.

## 2. Theory

The theory section discuss the relevant theory used in this thesis to give an understanding of how the theory was used, and to justify the use of the different models. Starting first with the root cause of the problem, fatigue damage. Then continuing along the process which was used to find that fatigue.

### 2.1. Fatigue loading and structural stresses

In situations where the loading is simple and static, it is sufficient to calculate the structural strength of each individual load case. However, for a situation with varying loads, and where a safe fatigue life is required, consideration of the fluctuations of the loads need to be taken into account. Following elementary fatigue strength theory, if the stress amplitudes are below the fatigue strength/endurance limit of the material, a material can sustain any number of fluctuations and if the stress amplitudes are higher than the fatigue strength, the material can only sustain a certain amount of cycles before failure occur, meaning the material is “fatigue-limited”. This process can be represented by the S-N diagram (see Figure 2.1), which is useful for “normal” engineering problems. In the case of a wind-turbine, where the loads are high, dynamic and complex, and the load spectrum consists of stochastic and periodic stress fluctuations, with varying mean values, elementary theory is not adequate in designing for fatigue strength. The load situation has to be assessed in its totality, as a load spectrum. A damage accumulation model can be used to summarise the endurance strength (Hau 2006). The Rainflow method is more thoroughly discussed in Subsection 2.9.

An approximation of the amount of cycles a wind turbine experience can be done using;

$$n_L = 60 \cdot n_p \cdot n_{rotor} \cdot H_{op} \cdot Y \quad (2.1)$$

Where  $n_p$  is the number of cyclic events per revolution,  $n_{rotor}$  the average rotational speed of the rotor.  $H_{op}$  is the operational hours per year and  $Y$  years of operation. The shifting of weight due to the rotation of the blades means  $n_p$  equal 1 for the blade root stress, while the tower would experience  $n_p$  equal to the number of blades.

For instance, a wind turbine with three blades operating at an average of 13 rpm, 5000 hours a year, for 25 years would experience at a minimum, close to  $10^8$  number of cycles.

There are several failure modes for a yaw pinion motor. Since the yaw gear are most of the time stationary and only correcting slight changes in the angle of the positioning of the rotor blades, they no not experience the same kind of fatigue damage that a spinning gear would. The backlash that exists in every gear configuration can produce impact loads that these motors are especially prone to encounter, because the gears change direction as the wind change and due to the vibrations caused by turbulence.

The yaw mechanism in turbines consist of an elector motor, a reduction gear, and the shaft pinion for the yaw drive actuator and on the tower, a ring gear, a yaw bearing, and a break ring, seen in Figure 2.2. The break ring is designed to hold the nacelle in place when the turbine is in its correct position. However, if the loads and moments that the wind turbine experience is too high, the break system is not strong enough and a load is transferred to the gear ring and pinion motors (M.-G. Kim et al. 2014).

Fatigue of a material is a complex phenomena, and almost impossible to understand completely. The processes are ranging from the atomic scale, where strain hardening/work hardening cause defects in the lattice structure of the atoms. The process is causing the material to harden, and brittle, which in it self is a study of its own, and very complex and different for any composition

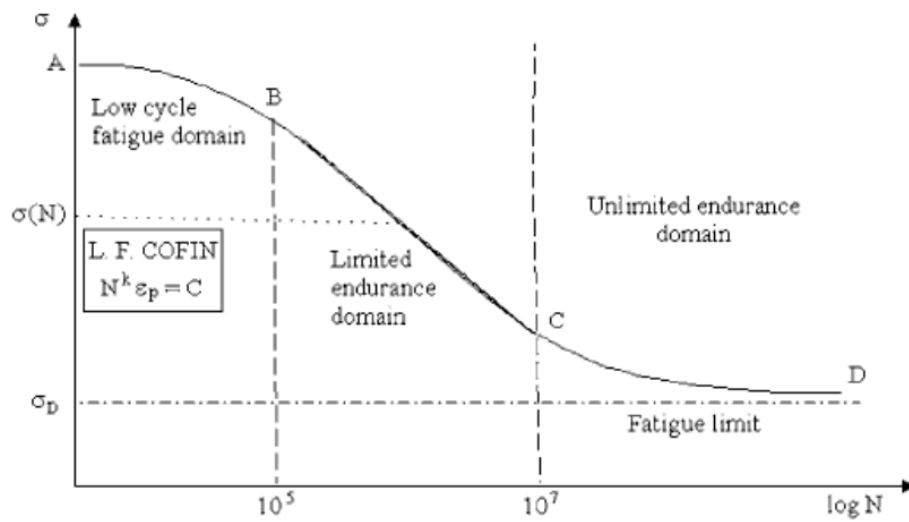


Figure 2.1: S-N diagram. Where N is cycles, and  $\sigma$  is the stress. Figure from Lalanne 2014

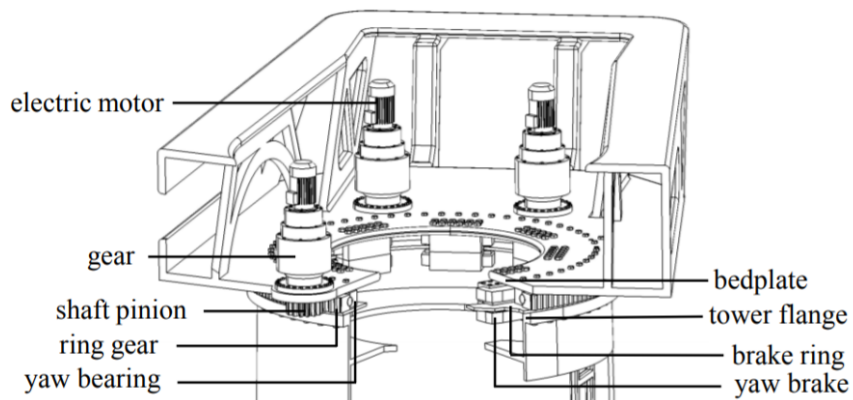


Figure 2.2: An example of a yaw drive system configuration. Picture from M.-G. Kim et al. 2014

of material. The different alloys behaves different under the same stress and strain cycles. The microscopic to macroscopic scale involves, among others, the effect of crack growth, where unavoidable imperfections in the material cause an initiation of a crack. The crack will start to grow as the material is exposed to cyclic strains under the right conditions, eventually leading to complete failure of the structure. Corrosion and erosion will cause material imperfections to occur and can attribute to the britteling of material in localised zones and increase the speed at which the crack will grow (Lalanne 2002). All these processes are all linked together, and borderline impossible to explain under a single theory. This is why approximations and assumptions are made to simplify the process for engineers, and this is where the S-N curves, Rainflow counting, Peak counting etc, becomes useful.

Fatigue damage is defined as “*the modification of the characteristics of a material, primarily due to the formation of cracks and resulting form the repeated application of stress cycles.*” (Lalanne 2014).

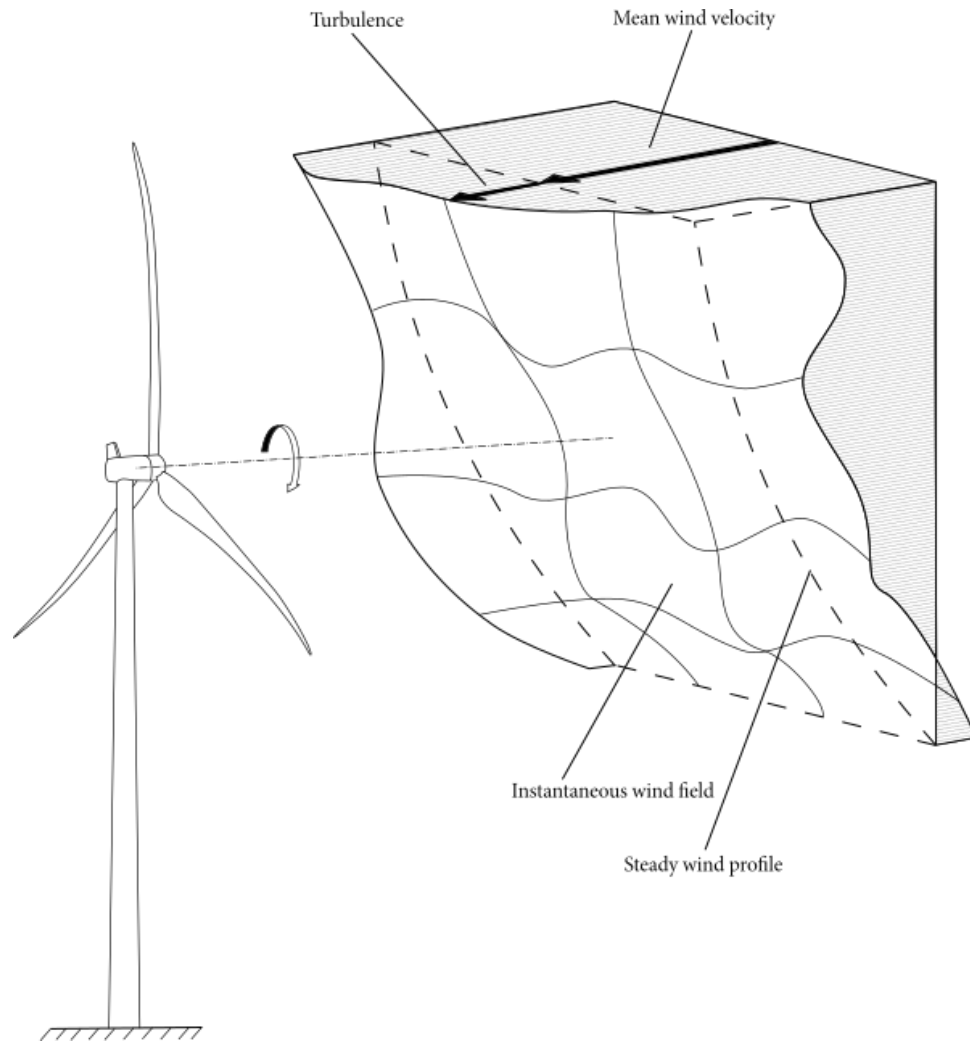


Figure 2.3: A representation of how the steady wind profile is influenced by a random turbulence to produce an instantaneous wind field.

## 2.2. Wind Turbine Loads

The following chapter explain the different types of loading a wind turbine is subjected to. According to Manwell and colleagues (Manwell et al. 2009) and NEK-IEC-61400-1 2019, wind turbines are generally subjected to 5 types of loading:

1. Steady (static and rotating)
2. Cyclic
3. Transient
4. Stochastic
5. Resonance-induced loads

### 2.2.1. Steady loads

The steady load include the forces that do not change over a period of time. Examples of a static load on a wind turbine would be the weight of the tower, the forces the tower experience



because of a steady wind, or the centrifugal force experienced by the blades as the rotor blades are steadily rotating.

### 2.2.2. Cyclic loads

The cyclic loading that a wind turbine experience are related particularly to the rotation of the rotor blades. Cyclic forces arise from the yaw motion, weight of the blades shifting. The cyclic loads of a wind turbine have a specific notation “Per revolution”, where P1 describes a cyclic load that happens once per revolution, and P3, three times per revolution. As an example a wind turbine blade will experience a P1 cyclic force as the weight shift from one direction to the other due to gravity during one rotation. The main shaft of the wind turbine will experience a P3 cyclic load for a wind turbine with three blades. The tower will experience a P3 cyclic load as the wind turbine blades interfere with the steady wind speed approaching the tower. If the structures natural frequency falls within the range of P3, large vibrations can be produced affecting the performance and structural integrity of the turbine, even complete structural collapse (Hau 2006).

### 2.2.3. Transient loads

The transient loads relate to the time varying loads. The transient loads originate from specific events, and cause a force/vibration that eventually decay. An example of this would be the activation of the yaw gear which produce a moment due to the activation of the yaw gears in the nacelle. Another important transient load to consider is the force associated with the “tower shadow” when a downwind rotor passes behind the tower (into the wake of the tower) and the blades experience a change in load.

### 2.2.4. Stochastic Loads

The stochastic loads that a wind turbine experience can generally be attributed to the wind turbulence. Seemingly random variation or fluctuation of loads caused by deviations of the mean value of the wind speed and wind direction. Figure 2.3 show how the turbulence can be “added” to a steady wind profile, to represent the full instantaneous wind profile. This is also how the TurbSim (see 4.1.2) module of NREL is programmed, and used later in this thesis.

### 2.2.5. Resonance-Induced Loads

The resonance induced loads are the result of the dynamic response from the wind turbine structure if the wind turbine experience cyclic, transient or stochastic loads in the frequency range of the structures natural frequency, and the damping response of the structure is not adequate, large vibrations can be occur.

## 2.3. Turbulence

Turbulence is described as the chaotic motions of a fluid through changes in pressure and velocities. Turbulent flow is characterised by the following important features.

- **Irregularity** The turbulent behaviour is highly unsteady. Making deterministic modelling difficult. Statistical approach is more convenient.

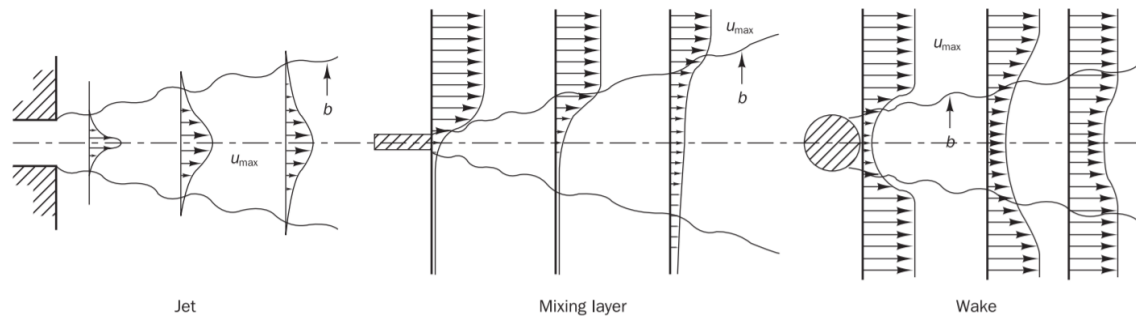


Figure 2.4: Types of free turbulent flows. The mixing of different speeds in a fluid cause turbulence (Versteeg et al. 1995)

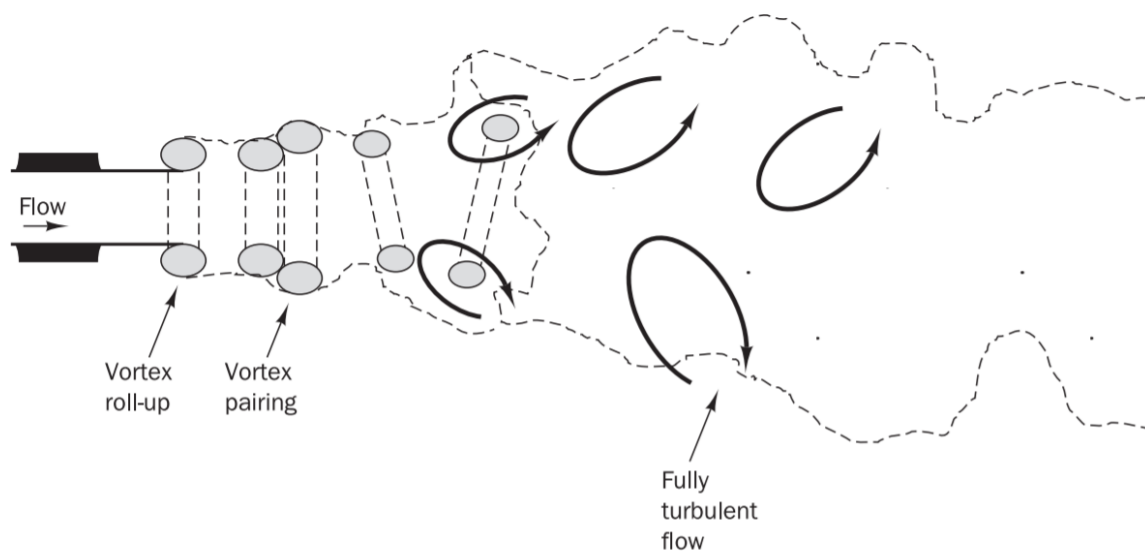


Figure 2.5: A typical jet flow example with regions of differing fluid velocities causing turbulence. The vortices will increase in size from small to large eddies covering a significant portion of the flow (Versteeg et al. 1995)

- **Rationality** The turbulence is three dimensional and the turbulence creates vorticity, i.e rotational flow.
- **Dissipation** The turbulence will vary in time and length scales. The turbulent behaviour will bring regions of differing momentum and energy into contact. Kinetic energy dissipate due to viscous shear stress, a constant supply of energy is required to sustain the turbulent flow.

The Reynolds number ( $Re$ ), which is the ratio of internal forces versus viscous forces, can predict when a laminar flow will transition into a turbulent flow.

One of the more important topics of CFD in general is the free turbulent flows. i.e mixing layers of differing speeds, jets and wakes, see Figure 2.4 and Figure 2.5. Turbulence will occur at location with one fast and one slow moving layer of fluid in contact with each other. This scenario is typical for terrain profiles, see Figure 6.3 and Figure 6.5. The changing topography forces flows of different speeds together causing turbulence Versteeg et al. 1995.

## 2.4. Fluid Dynamics

Fluid mechanics use the concepts of conservation of momentum and mass as the governing equations to compute the behaviour of fluids. These equations are represented by the Navier-Stokes equations (Eq. 2.2, 2.3 and 2.4) and the continuity equation (Eq. 2.5). The following equations contain a transient term, while this thesis focus on the steady-state solution. The transient term can be removed from the equations but is left in, for the following explanations.

Navier-Stokes equations:

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \nabla \cdot (\nabla u) \quad (2.2)$$

$$\frac{\partial v}{\partial t} + \nabla \cdot (v\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \nabla \cdot (\nabla v) \quad (2.3)$$

$$\frac{\partial w}{\partial t} + \nabla \cdot (w\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \nabla \cdot (\nabla w) \quad (2.4)$$

Continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0 \quad (2.5)$$

Since the Navier-Stokes equations require an unrealistic amount of computing power to solve, the equations are simplified. Different models for the turbulence has been developed and each have their advantages and disadvantages. The models are able to predict the solutions of the Navier-stokes equations.

One of these methods is the Reynolds-averaged Navier-Stokes equations (RANS). The idea behind the RANS theory, is to decompose the instantaneous wind speed into two segments, the mean values, and the turbulent fluctuating value (Versteeg et al. 1995).

$$\frac{\partial U}{\partial t} + \nabla \cdot (U\mathbf{U}) + \nabla \cdot (\overline{u'\mathbf{u}'}) = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \nabla \cdot (\nabla U) + S_\phi \quad (2.6)$$

which can be repeated for y and z direction.

The term  $(\overline{u'\mathbf{u}'})$  is a result of time averaging, and involve the products of the fluctuating velocities. These are connected to the convective momentum transfer from the turbulent eddies, and  $S_\phi$  corresponds with a source term for a field within the domain.

The turbulent kinetic energy ( $k$ ) in computational fluid dynamics is associated with the eddies and currents in the turbulent flow. The kinetic turbulent energy is described as the root mean square of the velocity changes in Versteeg et al. 1995. The dissipation rate of the turbulence is represented by  $\epsilon$ .

The following equations for the CFD simulations are derived by Versteeg et al. 1995, and presented in *OpenFOAM: User Guide, k-epsilon* 2019.

$$k = \frac{1}{2} \left( \overline{(u')^2} + \overline{(v')^2} + \overline{(w')^2} \right) \quad (2.7)$$

where  $u$ ,  $v$  and  $w$  are the instantaneous turbulent wind velocities for each coordinate direction,  $x$ ,  $y$  and  $z$  respectively.

The main feature of the  $k - \epsilon$  model, is the assumption that the turbulent viscosity is isotropic, which means that the Reynolds stresses, which are the components of the stress tensor in a fluid, and the mean rate of deformation of the fluid is the same in all directions.

In the OpenFOAM model of the  $k - \epsilon$  model, the following equations 2.8 and 2.9 are used to solve for turbulence intensity and dissipation.

$$\frac{D}{Dt}(\rho k) = \nabla \cdot (\rho D_k \nabla k) + G_k + \frac{2}{3} \rho (\nabla \cdot \mathbf{u}) k - \rho \epsilon + S_k \quad (2.8)$$

$$\frac{D}{Dt}(\rho \epsilon) = \nabla \cdot (\rho D_\epsilon \nabla \epsilon) + \frac{C_1 G_k \epsilon}{k} - \left( \frac{2}{3} C_1 - C_{3,\text{RDT}} \right) \rho (\nabla \cdot \mathbf{u}) \epsilon - C_2 \rho \frac{\epsilon^2}{k} + S_\epsilon \quad (2.9)$$

Where  $\frac{D}{Dt}(\rho k)$  is the mean-flow material derivative of the turbulent kinetic energy,  $\rho$  is the fluid density, and  $\nabla$  is a gradient operator.

$$\nabla = \left[ \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right] = \vec{i} \frac{\partial}{\partial x} + \vec{j} \frac{\partial}{\partial y} + \vec{k} \frac{\partial}{\partial z} \quad (2.10)$$

and  $\nabla \cdot$  is a divergence operator and produce a scalar:

$$\nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \quad (2.11)$$

where  $G_k$  is the production term for turbulent kinetic energy.

$$G_k = 2\mu_t S_{ij} \cdot S_{ij} \quad (2.12)$$

The eddy viscosity,  $\mu_t$  in Equation 2.12 is defined, by the kinematic viscosity  $\nu_t$  and the fluid density as as:

$$\mu_t = \rho \nu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad (2.13)$$

where  $C_\mu$  is a dimensionless constant and  $S_{ij}$  is the rate of deformation in tensor matrix given as:

$$S_{ij} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \quad (2.14)$$

The parameters of the equations are given in Table 2.1.

Table 2.1: Values for parameters in the  $k - \epsilon$  solver. Default value for  $\sigma_\epsilon$  1.3 (Versteeg et al. 1995). Modified according to Hargreaves et al. 2007

$C_\mu$	$C_1$	$C_2$	$C_{3,\text{RDT}}$	$\sigma_k$	$\sigma_\epsilon$
0.09	1.44	1.92	0	1	1.11

The turbulent kinetic energy can be described using the formula:

$$k = \frac{3}{2} (I \cdot |\mathbf{u}_{\text{ref}}|)^2 \quad (2.15)$$

Table 2.2: The roughness parameter for some terrain types according to DNVGL-RP-C205 2017.

Terrain type	Roughness parameter $z_0$ (m)	Power-law exponent $\alpha$
Plane ice	0.00001 to 0.0001	
Open sea without waves	0.0001	
Open sea with waves	0.0001 to 0.01	0.12
Long grass, rocky ground	0.05	
Pasture land	0.2	
Forests and suburbs	0.3	0.3
City centres	1 to 10	0.4

which then give the turbulence intensity  $I$ , as:

$$I = \frac{\sqrt{\frac{2}{3}k}}{|\mathbf{u}_{\text{ref}}|} \quad (2.16)$$

## 2.5. Actuator Disk Model

The model of the wind turbines in the CFD simulations use an Actuating disk model. As the wind stream pass through the cell sett defined by “topoSet”, see Section 5.1.3, the cell sett act as actuators and will impose a momentum to the stream passing through the turbine model. Instead of modelling the rotor blades of the turbine, and computing the momentum through the interaction of an actual revolving rotor, the actuator disk acts as a placeholder for the turbine, and simply add the expected momentum to the flow.

The moment is added through a source term, see Equation: 2.6:

$$S_d = 2\rho A_d \cdot U_\infty^2 \cdot a(1 - a) \quad (2.17)$$

$A_d$  is the area of the disk and where  $a$  is the induction factor:

$$a = 1 - \frac{C_p}{C_T} \text{ or } 1 - \frac{U_d}{U_\infty} \quad (2.18)$$

$C_p$  is the power coefficient of the turbine, and  $C_t$  is the torque coefficient.  $U_d$  is the wind speed at the disk, and  $U_\infty$  is the wind speed of the ambient air.

### 2.5.1. Terrain roughness parameter

The roughness of the terrain has its own parameter,  $z_0$ . The roughness parameter is a measure of length and depends on the topography, and affect the profile of the wind. DNV GL standards (DNVGL-RP-C205 2017) contain a table of different terrain types and their roughness parameter, along with the power-law exponent for wind profile. Table 2.2 show the description of some different terrains along with a value for the roughness parameter.

There are several ways to calculate the  $z_0$  parameter, using quite fundamentally different techniques. The study of the wind profile using a wind mast with sensor at several different heights, as in the study by Tharmaraj et al. 2016, or by using satellite images as in the study by Ramli et al. 2009. For this thesis an assumption is made for  $z_0$  to be equal to 0.3.

To accommodate for the roughness in the terrain the boundary condition of the terrain is according to Hargreaves et al. 2007.

Table 2.3: Boundary conditions for the different fields. ABL (atmBoundaryLayer)

Boundary	Type	Field			
		k	epsilon	p	U
inlet	patch	ABLInletK	ABLInletEpsilon	zeroGradient	ABLInletVelocity
outlet	patch	inletOutlet	inletOutlet	uniformFixedValue	inletOutlet
terrain	wall	kqRWallFunction	epsilonWallFunction	zeroGradient	uniformFixedValue

### 2.5.2. Atmospheric boundary conditions

The boundary conditions for the CFD simulations is an essential part and is not only connected to the interactions of the walls but inherently change the entire result of the simulations. The boundary conditions enter the discretised equations by suppression of of the link to the boundary side and the modification of the source term (Versteeg et al. 1995). There are several different types of boundaries used, but for this thesis we only need to discuss a few. *patch* contains no geometric or topological information and is used for inlet and outlet conditions. The *wall* correspond to a solid object, i.e a wall that nothing will pass through, but can change the flow across it. For this thesis, the *wall* function is used for the terrain profile, while *patch* is used for all other boundaries i.e top, inlet, outlet and side patches (*OpenFOAM v6 User Guide* 2019).

*zeroGradient* boundary condition sets the boundary value to the closest near-wall cell. The flow can go both in and out of the patch.

*fixedValue* set a fixed predefined value for the field for all locations in the patch, the flow across the patch will not change.

*inletOutlet* condition changes between *zeroGradient* as the flow is going out of the domain, and *fixedValue* as the flow is returning to the domain.

*atmBoundaryLayerInletK* is the inlet condition for the turbulent kinetic energy and defines the starting turbulent starting condition as the flow enters the domain. The turbulent kinetic energy at the inlet boundary vary in the  $k-\epsilon$  model as shown in Equation 2.19 (Hargreaves et al. 2007). The values for the atmospheric boundary conditions are found in the *include/ABLConditions*. Examples of the input files are presented in Appendix A.

$$k = \frac{u_*^2}{\sqrt{C_\mu}} \quad (2.19)$$

*atmBoundaryLayerInletVelocity* defines the inlet conditions of the velocity. Defined by the equation:

$$U = \frac{u_*}{\kappa} \ln \left( \frac{z + z_0}{z_0} \right) \quad (2.20)$$

where  $u_*$  is the friction velocity and defined as:

$$u_* = \frac{u_{\text{ref}} \cdot \kappa}{\ln \left( \frac{z_{\text{ref}} + z_0}{z_0} \right)} \quad (2.21)$$

*atmBoundaryLayerInletVelocity* is the inlet boundary condition for the turbulent dissipation.

$$\epsilon = \frac{u_*^3}{\kappa(z + z_0)} \quad (2.22)$$

where  $z$  is the height above the boundary condition and  $\kappa$  is the von Karman constant.

## 2.6. Aerodynamic forces

The power generated by the wind turbine is dependent on the interaction of the rotor blades and the wind. The wind can be composed into a mean wind speed along with a turbulent wind speed. The modelling and design of a wind turbine with respect to performance and mean loads are attributed to the mean wind speed. While the fatigue loads and peak values are strongly dependant on the turbulent activity. These subsections discuss relevant theory related to the modelling of wind turbines, touching on the topics of performance and aerodynamic drag and wake.

### 2.6.1. Wind power calculations

Generally, there are two ways of expressing energy of an object, kinetic or potential. Kinetic energy is related to the objects relative moment, while potential energy is stored energy, which can be released or converted into kinetic energy. If we consider a pocket of air, moving uniformly and steadily relative to the ground, its kinetic energy is described by the mass of the air pocket and its velocity. Potential energy can also be described as a pressure variations, chemical energy (gasoline) etc.

The following equations, derived by Manwell (William Shepherd 2011) give the Betz limit, which is a limit to how much energy can be extracted from the airflow.

Power, as a function of time, describe the work done over a period. Where  $W$  is the work or kinetic energy, and  $t$  is the time.

$$P = \frac{dW}{dt} \quad (2.23)$$

The mass of the air contained in an element of volume is given by the density of the air  $\rho$ , its cross sectional area  $A$ , and its thickness  $x$ .

$$m = \rho Ax \quad (2.24)$$

The velocity  $v_w$  of the air element can be given by the time derivative of the incremental distance travelled:

$$\frac{dx}{dt} = v_w \quad (2.25)$$

The kinetic energy in a parcel of air, of mass,  $m$ , flowing at a speed of  $v_w$ , in a direction  $x$ , can be given by:

$$W = \frac{1}{2}mv_w^2 = \frac{1}{2}\rho Axv_w^2 \quad (2.26)$$

The kinetic energy reduction of the air passing the wind-turbine is then the kinetic energy before minus the kinetic energy after the passing of the wind turbine.

$$\begin{aligned} W &= \frac{1}{2}mv_{w1}^2 - \frac{1}{2}mv_{w3}^2 \\ &= \frac{1}{2}m(v_{w1}^2 - v_{w3}^2) \end{aligned} \quad (2.27)$$

To find the total amount of power stored in the wind we take the time derivative of the kinetic energy:

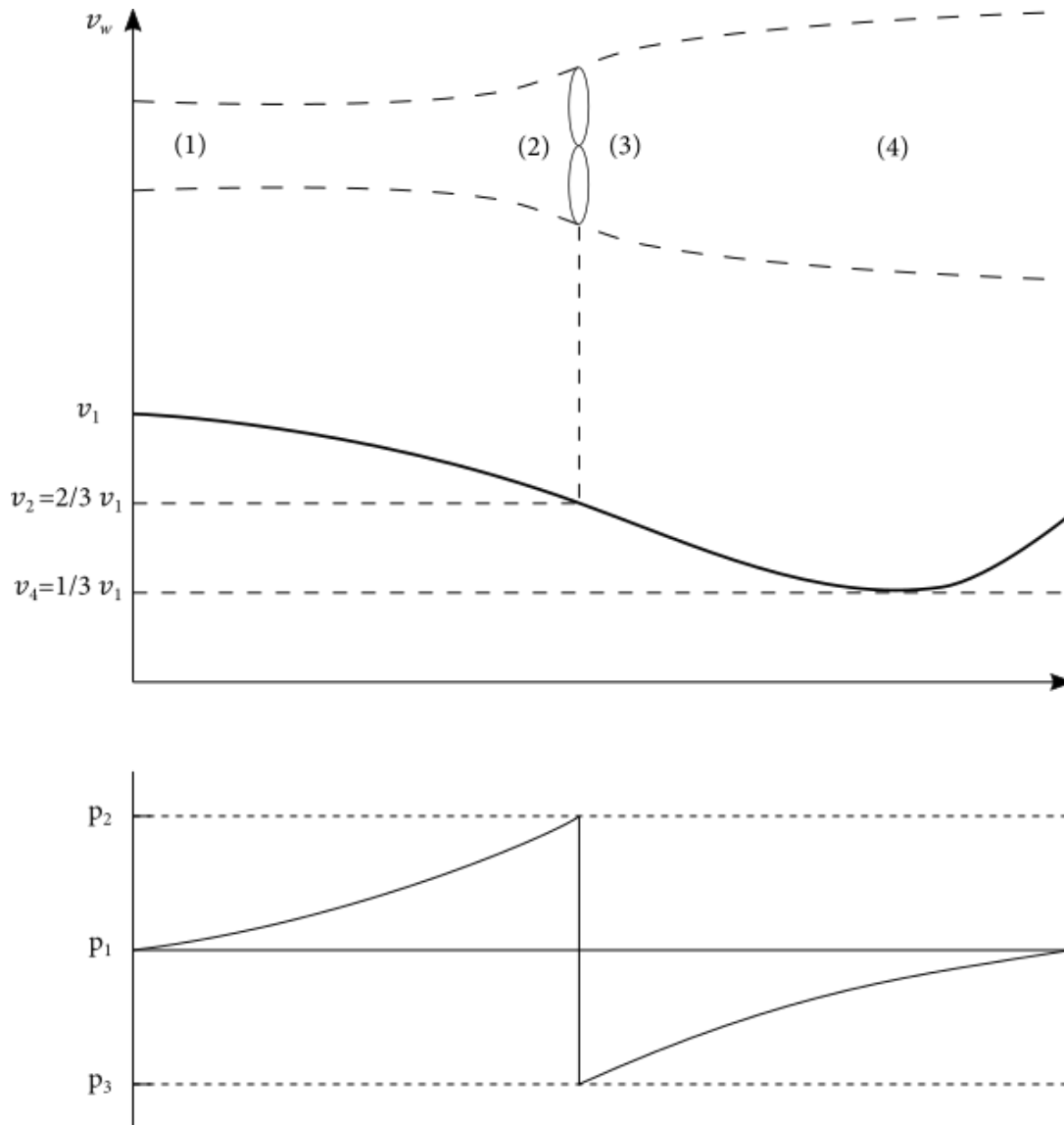


Figure 2.6: Circular tube of air flowing through an ideal wind turbine. Top curve representing the speed of the wind and bottom curve representing the pressure in the air parcel (Muyeen et al. 2009)

$$P = \frac{dW}{dt} = \frac{1}{2} \rho A x v^2 \frac{dx}{dt} \quad (2.28)$$

where  $A$ , is the cross-sectional area in  $m^2$ ,  $\rho$  is the density of the air flowing through the parcel in  $kg/m^3$ , and  $x$  is the thickness of the parcel in meters.

The average power in the wind  $P_w$ , is given by the time derivative of the kinetic energy.

$$P_w = \frac{dW}{dt} = \frac{1}{2} \rho A v_w^2 \frac{dx}{dt} = \frac{1}{2} \rho A v_w^3 \quad (2.29)$$

A wind turbine will extract power from the wind, where  $P_w$ , is representing the total amount of power available at the cross-sectional area. Consider a tube of moving air, with initial or undisturbed diameter,  $d_1$ , speed  $v_{w1}$ , and pressure  $p_1$  approaching the wind-turbine, shown in Figure 2.6. As the wind get closer to the wind turbine, the pressure builds, the speed decrease and the volume or the air pocket increase. The pressure will increase to a maximum just in front of the wind-turbine blades and drop to a minimum (below atmospheric pressure) just



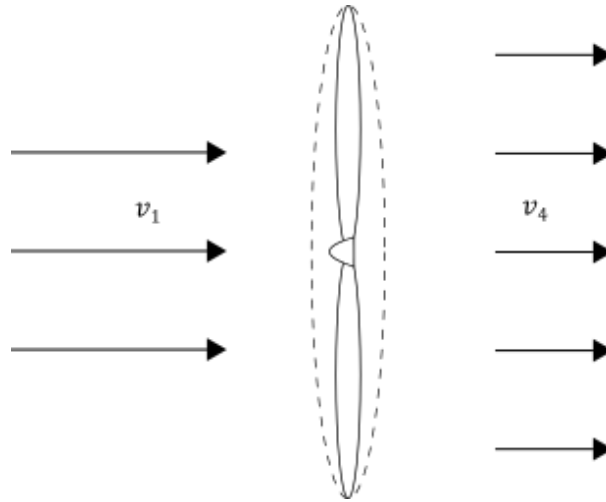


Figure 2.7: A change in wind velocity and volume, before and after passing the wind turbine blades. (William Shepherd 2011)

behind the blades. The kinetic energy of the wind is converted into potential energy in the form of a pressure variation, both in front of and behind the wind-turbine blades which causes the wind-speed to decrease in front and behind the wind-turbine blades. The surrounding air, will after some time, increase the wind of the air pocket to match the speed of the surrounding air.

In an ideal system where there is no mechanical energy loss due to friction, and the wind-turbine blades perfectly capture the kinetic energy of the wind and convert it into a rotational kinetic energy. The power received by the wind, is then the difference between the input and output kinetic energy of the wind. To derive the equation for the maximum amount of power we can extract, we derive an equation of power from the incremental work done, and through conservation of energy.

To find the rate of air mass transferred at the rotor blades, we take time derivative of the equation for air density Eq. 2.24 combined with the average air velocity Eq. 2.25.

$$\frac{dm}{dt} = \rho A \frac{dx}{dt} = \rho A v_r \quad (2.30)$$

The wind power at the rotor is then the time rate of kinetic energy transferred.

$$P_r = \frac{dW_r}{dt} \quad (2.31)$$

Substitution of eqs. 2.27 and 2.30 into Eq. 2.31, gives:

$$P_r = \frac{1}{2} \frac{dm}{dt} (v_1^2 - v_4^2) = \frac{1}{2} \rho A v_r (v_1^2 - v_4^2) \quad (2.32)$$

which gives us a representation of the power extracted from the wind through the perspective of energy conservation.

The mass passing through the wind-turbine blades undergoes a change in kinetic energy, and a change in linear momentum  $m(v_1 - v_4)$ .

The time rate of the change of momentum reduction is a force, given as:

$$\frac{d}{dt}(m(v_1 - v_4)) = \frac{dm}{dt}(v_1 - v_4) = \rho A v_r (v_1 - v_4) \quad (2.33)$$

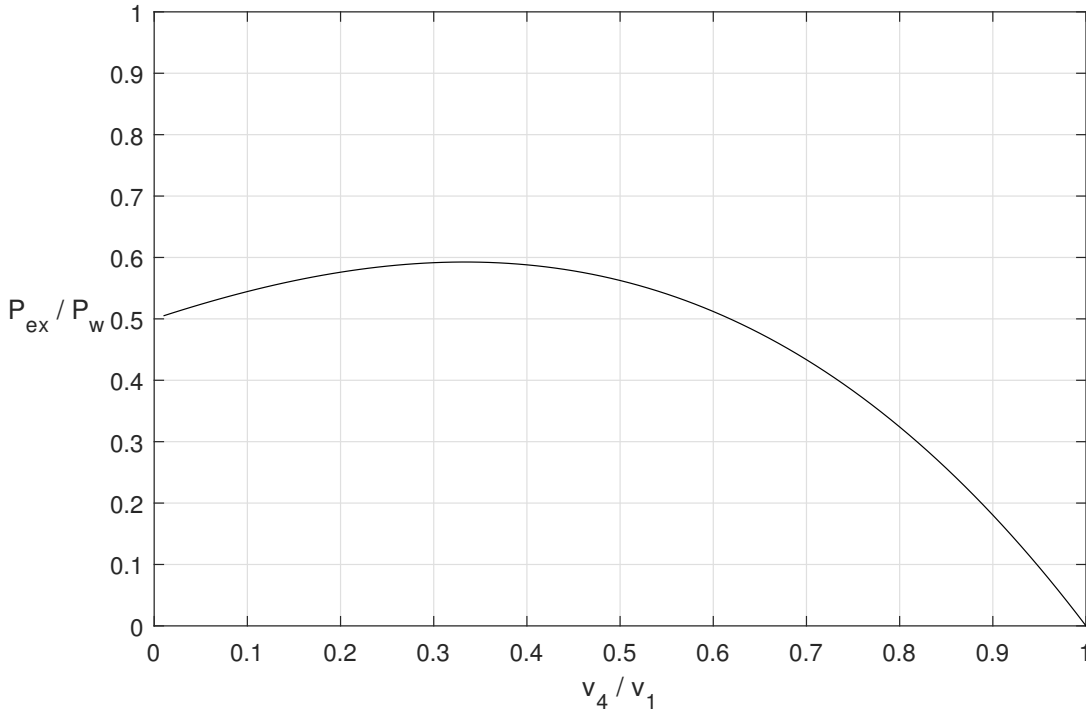


Figure 2.8: A graphical representation of the total power extractable, compared to the reduction of wind-speed before and after passing the windmill.

The power related to the rate of change of momentum at the wind-turbine blades is given by multiplying Eq. 2.33, with the relative average velocity at the wind-turbine blades, Eq. 2.25.

$$P_r = \rho A v_r^2 (v_1 - v_4) \quad (2.34)$$

By equating eqs. 2.31 and 2.34, we can find the velocity at the wind-turbine blades,  $v_r$ .

$$v_r = \frac{v_1 + v_4}{2} \quad (2.35)$$

With the velocity of the air passing the rotors known, the extractable power  $P_{ex}$  can be obtained by substitution of  $v_r$ , Eq. (2.35), into Eq. (2.31)

$$P_{ex} = \frac{1}{4} \rho A v_1^3 \left( 1 + \frac{v_4}{v_1} - \frac{v_4^2}{v_1^2} - \frac{v_4^3}{v_1^3} \right) \quad (2.36)$$

Derivation of Eq.(2.36), with respect to  $v_4/v_1$ , gives the value when  $P_{ex}$  reaches its maximum value.

Plotting the value of  $P_{ex}$  divided by the maximum theoretical power stored in the wind, calculated in Eq. (2.29), we obtain a graphical representation of the fraction of power theoretically extractable (Figure 2.8).

The maximum value occurs at the point where  $v_4/v_1 = 1/3$  in Figure 2.7. Substitution of this, into Eq. (2.36) yields:

$$P_{ex}(\max) = \frac{1}{2} \rho A v_1^3 \left( \frac{16}{27} \right) \quad (2.37)$$

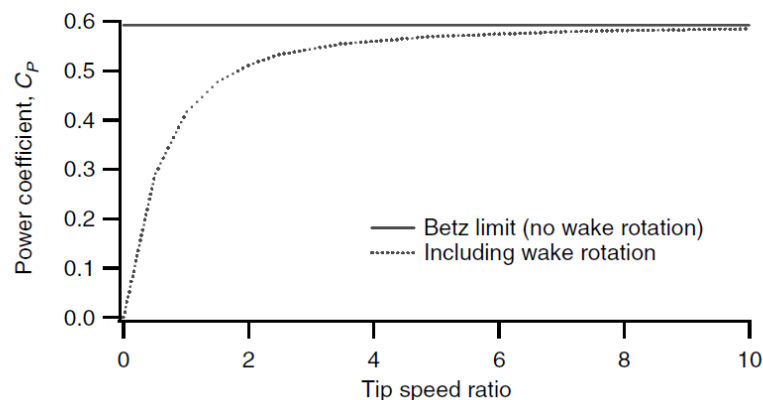


Figure 2.9: Theoretical maximum power coefficient as a function of tip speed ratio for an ideal horizontal axis wind turbine, with and without wake rotation (Manwell et al. 2009).

The fraction  $(16/27)$ , called Betz Law after Albert Betz (Betz, Grissmer, and Morse 1966), states that even under ideal wind conditions and perfect power transfer from the wind to the turbine blades, only 59.3% can be extracted. The fraction is also called the “ideal power coefficient”, usually denoted as  $C_p$ . Worth mentioning is the fact that these calculations are based on calculations assuming an ideal, frictionless flow and thus the Betz limit cannot be achieved due to:

- rotation of the wake behind the rotor
- finite number of blades
- non-zero aerodynamic drag

Presented in Table 2.4 are examples of theoretical wind-turbines and their ideal Betz limit energy production. Taking into account the the wake rotation behind the wind turbine, where some of the energy conversion is lost, gives the graph presented below in Figure 2.9. Showing both Betz limit without wake rotation and including the wake rotation (William Shepherd 2011).

Table 2.4: The maximum theoretical Power (kW) extractable by a wind-turbine, with different wind-speed and cross-sectional areas.

Wind speed (m/s)	Circular area of different diameter (m)				
	10	20	40	80	160
5	3.56	14.25	57.0	228	912
10	28.51	114.02	456.1	1824	7297
15	96.21	384	1539	6157	24630
20	445.42	1781	7126	28507	114028
25	769.69	3078	12315	49260	197040

The graph visualise the effect of differing tip speed ratio. A high-speed, low torque wind turbine will experience less wake rotation loss compared to a low-speed, high torque wind turbine. It is common for a wind turbine to operate at a tip-speed ratio of around 8-10. Meaning the tip of the wind turbine blade is moving 8 to 10 times faster than the mean wind speed approaching the wind turbine.

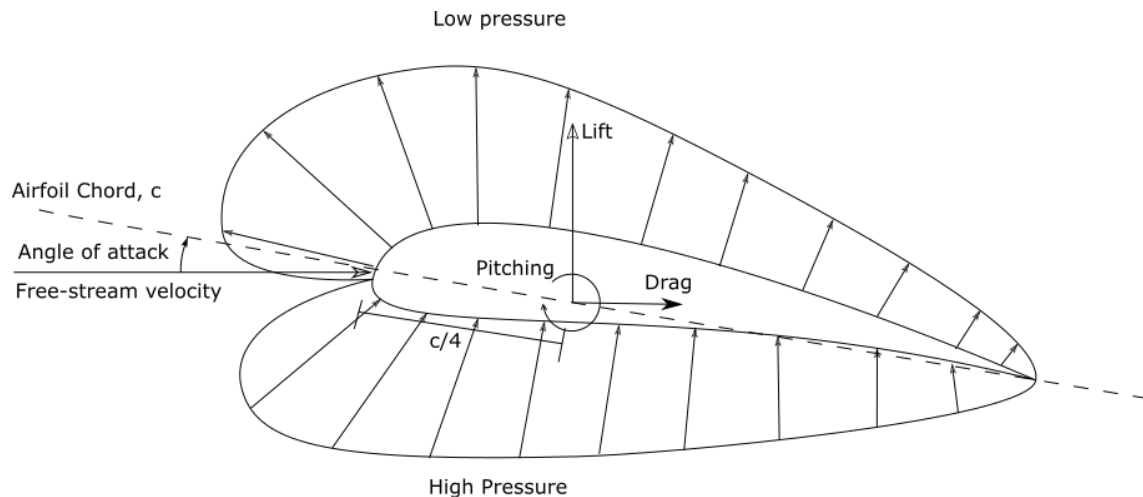


Figure 2.10: Aerodynamic forces acting on a section of the air-foil. Drawing from Hau 2006.

### 2.6.2. Airfoils and general concept

All the effects of aerodynamics on an airfoil are generally resolved into two forces and a moment.

- **Lift force** is perpendicular to the direction of the oncoming airflow, and is the force responsible for the rotation of the rotor blades. The force is a result of differing pressures on either side of the rotor blade, see Figure 2.10.
- **Drag force** is the force that acts parallel to the airflow and is caused by both differing pressure on either side of the blade, and from viscous friction forces at the surface of the airfoil.
- **Pitching moment** is the moment about the centre of the airfoil cross-section ( $c/4$ ).

## 2.7. Blade Element Momentum Theory

Building upon Betz law derived in Section 2.6.1, and now taking into account a more detailed description on how the wind turbine blades affect wind stream along with wake rotation, the blade element momentum theory (BEM) is developed. These equations are derived in most wind turbine design books and will only discussed in short. See Burton et al. 2011

Based on the principle of momentum conservation, the BEM theory breaks the rotor blade into several (N) smaller parts, and determine the forces acting on each separate element, see Figure 2.11

With the following assumptions (Manwell et al. 2009)

- There is no aerodynamic interaction between elements (thus, no radial flow)
- The forces on the blades are determined solely by the lift and drag characteristics of the airfoil shape of the blades.

The thrust and torque generated by the wind speed can be given by the following equations. Which give the distributed thrust produced by each element with a width of  $dr$

$$dT = \rho U^2 4a(1-a)\pi r dr \quad (2.38)$$

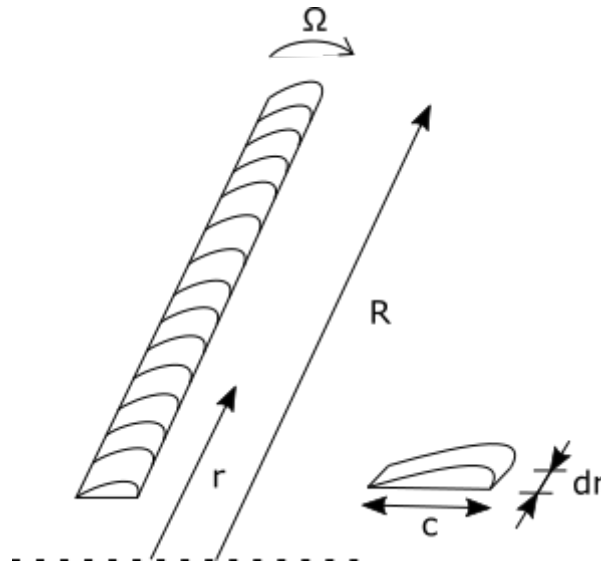


Figure 2.11: Representation of the blade elements used by the Beam Element Momentum Theory (Manwell et al. 2009).

and the torque produced by that element, at a distance  $r$

$$dQ = 4a'(1 - a)\rho U \pi r^3 \Omega dr \quad (2.39)$$

where  $a$  is the axial induction factor,  $a'$  is the rotational induction factor,  $\phi$  is the local flow angle,  $U$  is the mean wind speed and  $\Omega$  is the rotor rotational speed.

AeroDyn in Section 4.1.3 use several modifications to the BEM method to correct for some effects that are not covered by the simple BEM theory. Tip- and hub-loss models to account for vortices shed and a skewed wake correction model to account for wind not perpendicular to the rotor plane (Moriarty et al. 2005).

The corrections are done using a factor  $F$  in equations 2.38 and 2.39.

$$F = \frac{2}{\pi} \cos^{-1} e^{-f} \quad (2.40)$$

where

$$f = \frac{B R - r}{2 r \sin \phi} \quad (2.41)$$

for the tip loss, and

$$f = \frac{B r - R_{hub}}{2 r \sin \phi} \quad (2.42)$$

for the hub loss model. Giving us the modified equations

$$dT = \rho U^2 4a(1 - a)\pi r F dr \quad (2.43)$$

$$dQ = 4a'(1 - a)\rho U \pi r^3 \Omega F dr \quad (2.44)$$

## 2.8. Geometrically exact beam theory

FAST v8, used later in this thesis use GEBT to calculate the deformations of the rotor blades. “Geometrically exact” means that there are no approximations on the geometries, including initial and deformed geometries.

The following derivation and description of formulas are from Bauchau 2011. The governing equations of motions for the geometrically exact beam theory presented below used by the BeamDyn module is referenced from Wang et al. 2016, in the BeamDyn theory manual and Bauchau 2011.

$$\dot{\underline{h}} - \underline{F}' = \underline{f} \quad (2.45)$$

$$\dot{\underline{g}} + \tilde{\underline{u}}\dot{\underline{h}} - \underline{M}' + (x'_0 + \tilde{\underline{u}}')^T \underline{F} = \underline{m} \quad (2.46)$$

$\dot{\underline{h}}$  and  $\dot{\underline{g}}$  are the derivatives of the linear and angular momentum, respectively, resolved in the inertial coordinate system.  $\underline{F}$  and  $\underline{M}$  are the beam’s sectional force and moments resultants.  $\underline{u}$  is the one-dimensional(1D) displacement of a point on the reference line.  $\underline{x}_0$  is the position vector of a point along the beam’s reference line, and  $\underline{f}$  and  $\underline{m}$  are the distributed force and moment applied to the beam structure.

- the underline operator  $\underline{f}$  is used to denote a vector
- the dot notation  $\dot{\underline{h}}$  represent a time derivative
- while the apostrophe  $\underline{F}'$ , represent a derivative with respect to the beam axis  $x_1$
- the tilde operator  $\tilde{\underline{u}}'$  is a skew-symmetric tensor corresponding to the given vector, also called a “cross-product matrix”.

$$\begin{Bmatrix} \underline{h} \\ \underline{g} \end{Bmatrix} = \underline{\underline{\mathcal{M}}} \begin{Bmatrix} \dot{\underline{u}} \\ \underline{\omega} \end{Bmatrix} \quad (2.47)$$

$$\begin{Bmatrix} \underline{F} \\ \underline{M} \end{Bmatrix} = \underline{\underline{\mathcal{S}}} \begin{Bmatrix} \underline{\epsilon} \\ \underline{\kappa} \end{Bmatrix} \quad (2.48)$$

where  $\underline{\underline{\mathcal{M}}}$  and  $\underline{\underline{\mathcal{S}}}$  are  $6 \times 6$  sectional mass and stiffness matrices.  $\underline{\epsilon}$  and  $\underline{\kappa}$  are 1D strains and curvatures.  $\underline{\omega}$  is the angular velocity vector defined by the rotation tensor  $\underline{R}$  as  $\underline{\omega} = \text{axial}(\underline{\dot{R}} \underline{R}^T)$ . The axial vector  $\underline{a}$  associated with a second-order tensor  $\underline{\underline{A}}$  is denoted  $\underline{a} = \text{axial}(\underline{\underline{A}})$  and its components are defined as

$$\underline{a} = \text{axial}(\underline{\underline{A}}) = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} A_{32} - A_{23} \\ A_{13} - A_{31} \\ A_{21} - A_{12} \end{Bmatrix} \quad (2.49)$$

the 1D strain measures are defined as

$$\begin{Bmatrix} \underline{\epsilon} \\ \underline{\kappa} \end{Bmatrix} = \begin{Bmatrix} x'_0 + u' - (\underline{R} \underline{R}_0) \bar{\mathbf{i}}_1 \\ \underline{k} \end{Bmatrix} \quad (2.50)$$

Where  $\underline{k} = \text{axial}[(\underline{R}\underline{R}_0)'(\underline{R}\underline{R}_0)^T]$  is the sectional curvature vector resolved in the inertial basis;  $\underline{R}_0$  is the initial rotation tensor; and  $\bar{\mathbf{i}}_1$  is the unit vector along  $x_1$  direction in the inertial basis.

The equations presented above, 2.47, 2.48, 2.49 and 2.50, are the full mathematical description of the beam elasticity problems, and is the mathematical tool used to describe motions and kinetics in the BeamDyn module in the FAST v8 software (Bauchau 2011 and Wang et al. 2016).

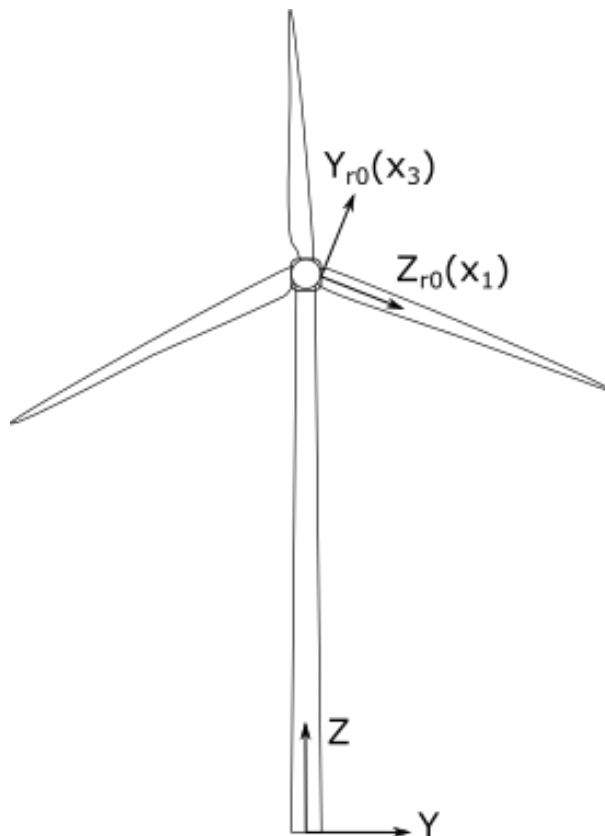


Figure 2.12: Reference coordinate system used by BeamDyn.

The coordinate system used by BeamDyn (see Section 4.1.4) can be seen in Figure 2.12. The global coordinate system is denoted as  $X$ ,  $Y$  and  $Z$  and is located at the bottom of the tower. The initial starting location BD coordinate system and acts as an inertial frame used internally by BeamDyn. In the the BeamDyn coordinate system, each blade is denoted as  $x_i$ ,  $y_i$  and  $z_i$  for each blade (1, 2 and 3). The reference coordinate system does not rotate with the rotor. There is also a rotating coordinate system, rotating along with the rotor blades seen in Figure 2.12. The local coordinate system is used for the cross-sectional mass and stiffness matrices from Equations 2.47 and 2.48, along with the sectional force and moment resultants.

## 2.9. Rainflow Counting

The Rainflow counting algorithm or *Pagoda roof method* is commonly used to evaluate the fatigue life of a structure with varying loads and amplitudes. The Rainflow counting algorithm is a modification of the standard S-N curve method and Minors rule where the numbers of cycles and amplitude along with other material properties give the fatigue life of a structure.

Figure 2.13 show how the counting mechanism in Rainflow counting works. The counting mechanism is analogous to a raindrop falling off a roof tile. There are six rules which the "raindrop" has to follow, the rules are taken from (Lalanne 2002).

1. The drop will stop if it meets an opposing peak larger than that of departure

2. it will also stop if it meets the path traversed by another drop, previously determined
3. the drop can fall on another roof and continue to slip according to rules 1 and 2
4. the fall will stop if the drop meets a valley deeper than that of departure
5. the fall will stop if it crosses the path of a drop coming from a preceding valley
6. the drop can fall on another roof and continue according to rules 4 and 5.

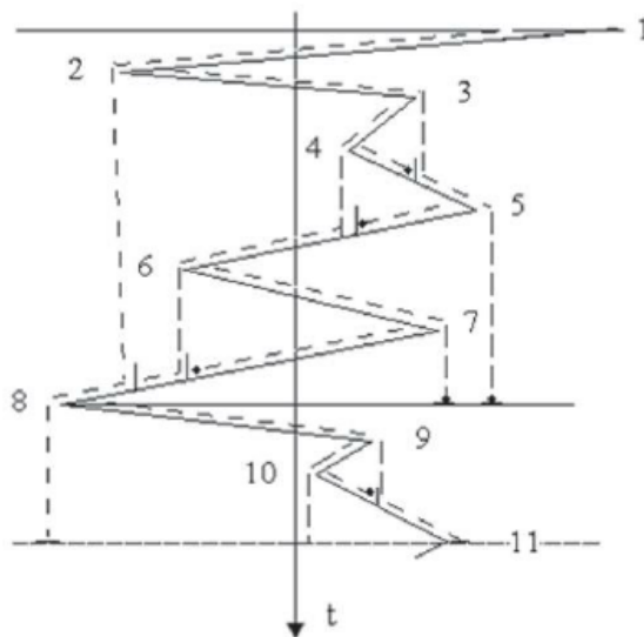


Figure 2.13: A typical example of the Rainflow counting method. Imagine a raindrop falling on a “pagoda” roof (Lalanne 2002).

The Rainflow algorithm is the concept of damage accumulation (Minors rule) to evaluate the integrity of the structure. By segmenting different amplitudes and loads into several different load “blocks”, and then estimating the damage accumulation of each “block”, the damage can be estimated over time, and an estimate of the fatigue life can be made.

$B$  represent the number of “blocks”,  $k$ , that the stress range is divided into,  $N_k$  is the number of cycles of loading condition  $k$  and  $N_{fk}$  is the number of cycles of the loading condition until failure (ASTM-E1049:85 2017).

$$B = \left( \frac{N_1}{N_{f1}} + \frac{N_2}{N_{f2}} + \dots + \frac{N_k}{N_{fk}} \right) = 1 \quad (2.51)$$

The formula is based on the Basquin’s equation:

$$S_r = AN_f^b \quad (2.52)$$

A tolerable reversing stress  $S_r$  can be found from Equation 2.52. Where  $A$  and  $b$  are material properties, and  $N_f$  is the number of cycles until failure.

An approximation of  $b$  can be found from the equation:



$$b = \frac{\log(S_e) - \log(0.9S_u)}{3} \quad (2.53)$$

and  $A$ , from:

$$A = \frac{S_e}{10^{6b}} \quad (2.54)$$

where  $S_e$  is the endurance limit of the material. The loading at which the material can endure unlimited amount of cycles.

To find the number of cycles until failure under each loading condition, Goodman-Basquin relation is used as;

$$N_f = \left( \frac{S_r}{A} \right)^{\frac{1}{b}} \quad (2.55)$$

where, for every “block”  $k$ ,  $\sigma_a$  is the stress amplitude,  $\sigma_m$  is the mean stress amplitude. The material properties  $\sigma_f$ ,  $\sigma_u$  and  $b$  are the fatigue strength coefficients, ultimate strength and fatigue strength exponent respectively.

The rainflow-counting method is used for structural components, and in wind turbine design, specifically on the wind turbine blades and tower (Lee et al. 2012). Moving and rotating parts like the gears where the forces are acting in cycles usually will not be suitable for such a method. However, since the gears are at a stand still when the turbine is yawed into position and the gears are locked in place, it is a fair assumption to model the gears in such a way (S.-W. Kim et al. 2017).

The model used for the teeth profile is depicted in Figure 2.14 and Figure 2.15. The force acting on the pinion gear is equal to:

$$F_t = \frac{M_{yaw}}{r_b} \quad (2.56)$$

and the maximum stress acting on the gear tooth

$$\sigma_f = \frac{F_t}{m \cdot b \cdot Y} \quad (2.57)$$

where  $m$  is the modulus of the gear teeth profiles,  $b$  is the width of the gear tooth, and  $Y$  is the Lewis form factor, dependent on the amount of gear teeth on the spur gear (Lemu 2016).

The yaw system is often underdimensioned to reduce costs (Mann et al. 2014), which means that the yaw breaking system is not capable of withstanding the extreme loads that the weather conditions might induce. This means that under these conditions, slippage in the breaks of the yaw system will cause forces to act on the gears of the yaw drive. The model used in this thesis does not account for the fact that the yaw turbine breaks hold some of the moment induced by the wind on the rotor blades (M.-G. Kim and Dalhoff 2014), but instead, only count the actual operational time of the yaw system.

## 2.10. Other methods of analysing random time history

A few similar counting methods that has been developed are presented in the following sub-chapters. The Rainflow method is chosen for its accuracy and ability to produce similar results to actual experiments (Lalanne 2002), and its repeated use in turbine design.

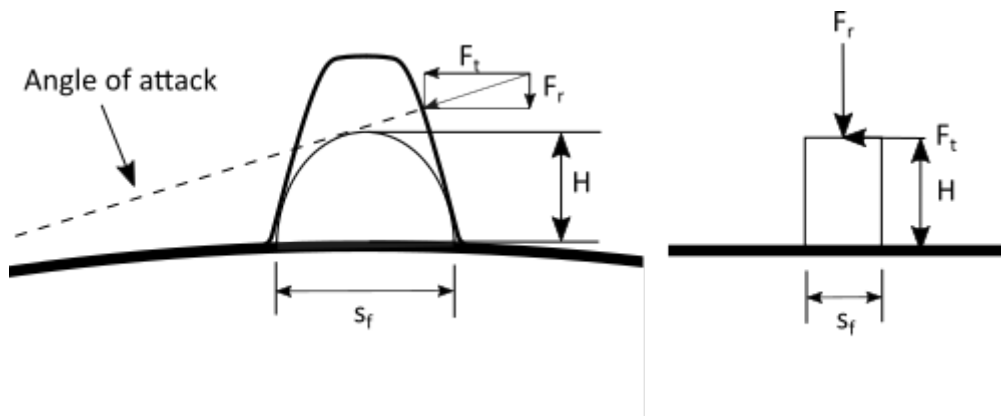


Figure 2.14: The description of the gear tooth model. Figure from Lemu 2016

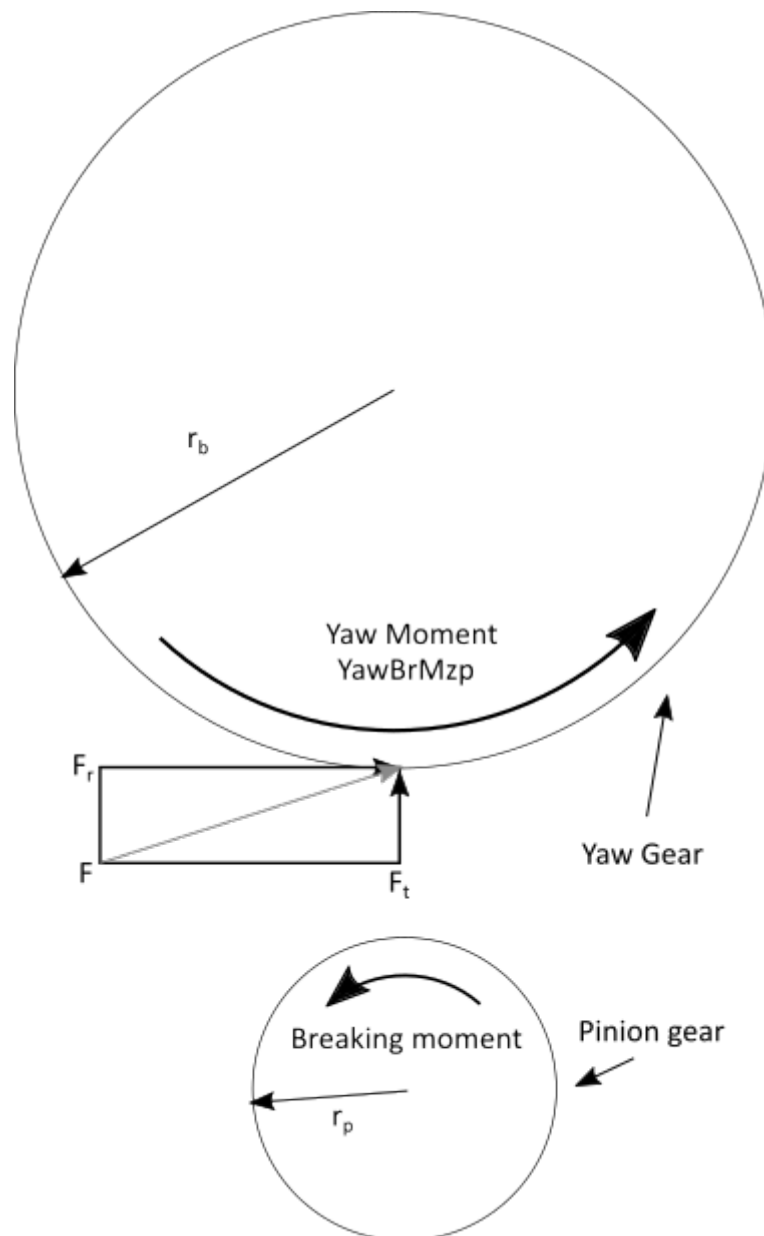


Figure 2.15: Visualisation of the moments and forces acting on the gear teeth. Figure from Lemu 2016

### 2.10.1. Peak count method

The peak count method is the simplest and oldest version for these types of analysis. Simply counting the peak values of the signal, counting values above the mean stress as positive and below mean as negative. With a counting mechanism like this, a lot of information is lost, where the signal originated from, or order the the peak loads and sequence. The method is simple and fast. This method have been derived into more refined methods.

### 2.10.2. Peak between mean crossing count method

This method count the maximum peak between the mean values of the signal. For every cycle, only the maximum peak is counted as a load cycle, which means that, for any load cycle all the fluctuations above the mean value is lost.

### 2.10.3. Range count method

This method count the difference between the maximum and minimum values of all fluctuations, recording only the difference for each cycle. With this method the absolute peak values are lost. The damage is calculated using S-N curves and the accumulation model in Miners rule. There are also modifications of this, for example where the smallest fluctuations (noise), are removed from the counting.

*“The range counting method detect variations which actually takes place, but neglect the variations of the mean load” (Lalanne 2002).*

### 3. OpenFOAM

OpenFOAM is a free open-source software for CFD simulations, and is developed by OpenCFD Ltd. The software was first released in version 1.0 in 2004, and since then has been developed by the OpenFOAM team and its community partners, as a freely available software for academic and industrial use, and now exist in two official versions with the most recent version from OpenFOAM foundation as OF v6.0 used in this thesis, and a version from ESI OpenCFD as OF v1812. The code is written in C++ making it possible for users with coding knowledge to modify and change the functionality of OpenFOAM to suit the needs of the users. The community has a large user base and an active forum where users can ask questions and find solutions to problems. There is also an library of tutorials that come with OpenFOAM which can be used for teaching and learning, or as a starting point for a study.

OpenFOAM, much like FAST v8 and OpenFAST, is run through unix commands, and use text files to modify the case conditions. The code is stored in large libraries, where each solver use code from different parts of the library.

Every OpenFOAM case use the same “standard” setup with the same types of folders, presented in Figure 3.1.

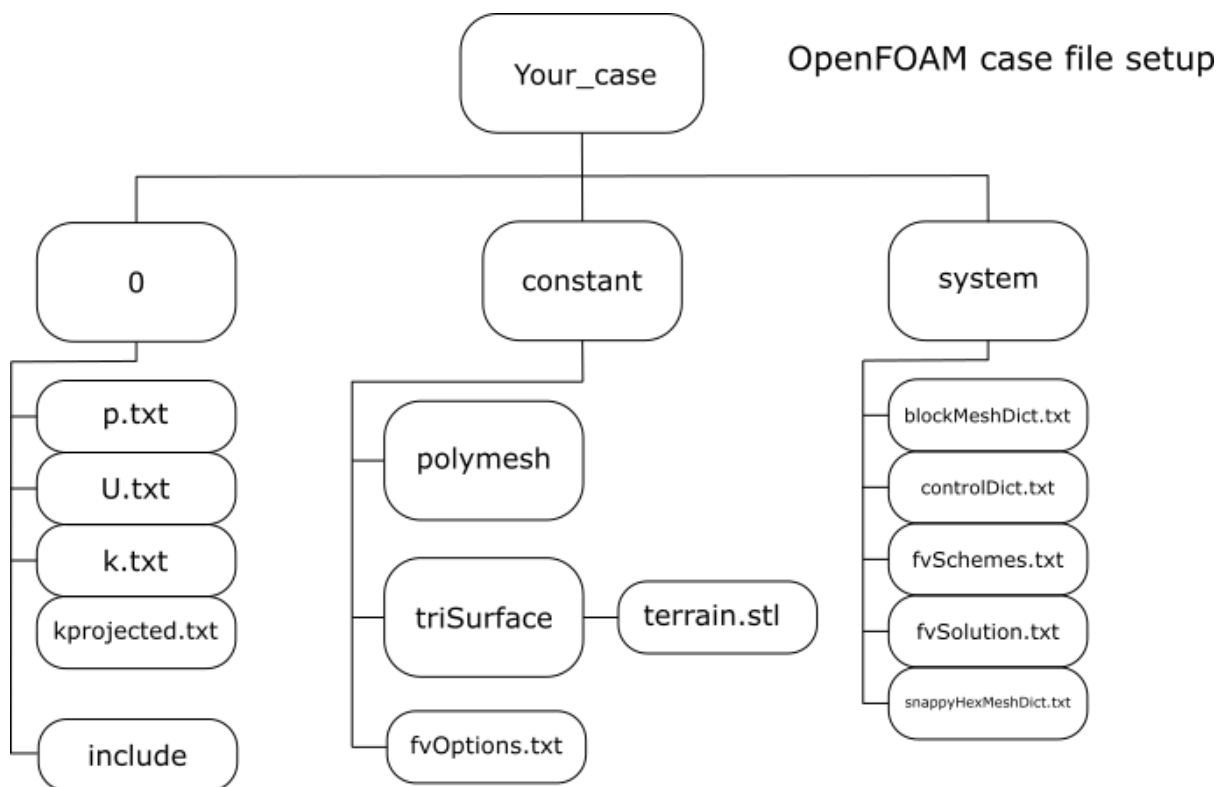


Figure 3.1: Structure of the case folder in OpenFOAM, containing the necessary folders. Each folder contain text files describing boundary conditions, solvers, terrain files (in this case), etc.

## 4. FAST v8

FAST v8 is a software developed by the United States *Department of Energy's* (DOE's) National Renewable Energy Laboratory (NREL). FAST is a time domain *computer-aided engineering* software for simulating the dynamic responses of horizontal-axis wind turbines. The software contains modules for both hydrodynamic and aerodynamic loads to simulate offshore and onshore wind turbines and support for modelling of rotor blades, nacelle, tower, support structure and the control and electrical systems (*National Renewable Energy Laboratory* 2019). The software can be obtained from the National Wind Technology Centre's (NWTC) information portal (*NWTC Information Portal* 2019).

### 4.1. Subsystems

FAST v8 consists of several different subsystems, the input files from each of the subsystems are used to generate the output file in FAST v8. This subchapter will go over the different subsystems, some of the important input parameters and their role in the simulation.

#### 4.1.1. InflowWind

InflowWind contain all the wind data necessary to develop a model of the wind profile and solve equations in a time domain. For each time step InflowWind receives coordinate position for various points and output the undistributed wind-inflow for these coordinates. Each wind velocity component is calculated as a function of the input and time-varying parameters internally in the code. Input files from TurbSim, using binary TurbSim full-field (FF) format the wind profile is simulated.

An example of an input file for InflowWind, along with the other relevant FAST v8 is presented in Appendix B.

**Echo**, when **Echo** is *True*, InflowWind will output an .ech file which simply is a read-back of the input file and is used for debugging the input file.

**WindType** parameter choose the type of wind data that InflowWind will read. Steady, Uniform, binary Turbsim FF, binary Bladed-style FF etc.

**PropagationDir**, is the direction of wind propagation in degrees. For each simulation a specific wind direction need to be input and will not change during the simulation.

#### 4.1.2. TurbSim

**TurbSim** is a full-field (FF) numerical simulation tool used to model stochastic inflow wind. TurbSim generate randomised coherent turbulent structures that are then superimposed onto a random background turbulent field produced by non-neutral spectral models. See Table 4.1.

**WrADFF** set *True*, makes TurbSim output a FF time-series in Turbsim or Aerodyn format (RootName.bts)

**NumGrid\_Z** and **NumGrid\_Y** define the grid point at where the wind will be generated, and are the number of grid points in vertical and horizontal direction respectively. See Figure 4.1.

**HubHt** set the hub height for the wind turbine.

**TurbModel** define the turbulence model that TurbSim will use to generate the FF wind file. Several different types of turbulence models have been implemented in TurbSim.

Table 4.1: Valid Turbsim Spectral Models (Jonkman et al. 2012).

TurbModel Input Value	Description
GP_LLJ	NREL Great Plains low-level jet
IECKAI	IEC Kaimal
IECVKM	IEC von Karman
NWTCUP	NREL National Wind Technology Center
SMOOTH	Risø smooth terrain
WF_07D	NREL wind farm: 7 rotor-diameters downwind
WF_14D	NREL wind farm: 14 rotor-diameters downwind
WF_UPW	NREL wind farm: upwind
TIDAL	Tidal channel turbulence model (water)

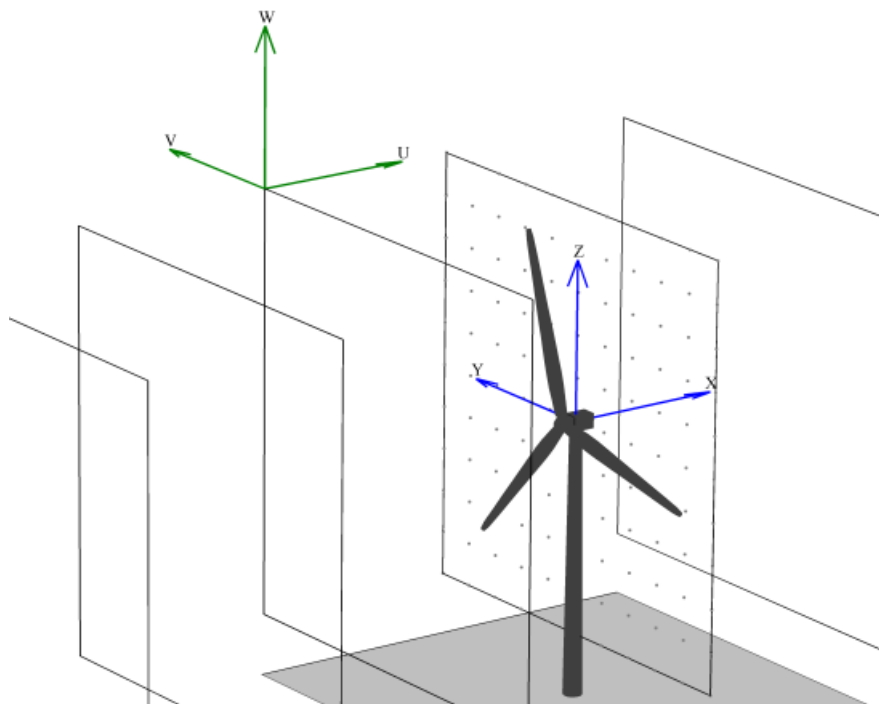


Figure 4.1: Example of TurbSim grids as implemented in AeroDyn

**IECturbc** tells Turbsim which turbulence intensity to use for Kaimal or von Karman spectral model. “A”, “B” and “C” select turbulence intensity according to IEC categories, A being the strongest turbulence and C the lowest. You can also input a desired turbulence intensity percentage, calculated from the following formula.

$$\text{IECturbc} = \frac{\sigma_1}{u_{hub}} \cdot 100\% \quad (4.1)$$

Where  $\sigma_1$  is the standard deviation and  $u_{hub}$  is the mean value of the wind velocity at hub height.

**IEC\_WindType** select the desired IEC turbulence model, i.e normal turbulence, extreme turbulence, extreme wind speed with 1 or 50 year recurrence, with class 1, 2 or 3.

The wind profile **WindProfileType** select between the different wind models. Where the “default” profile type is the IEC wind profile.

### 4.1.3. AeroDyn

AeroDyn module is responsible for calculating the forces caused by the wind that are acting on the wind turbine (i.e. aero-elastic calculations). The module uses Blade Element Momentum theory (see Section 2.7) in a set of routines to carry out the calculations. Blade Element Theory is the conventional method to calculate these forces.

### 4.1.4. BeamDyn

BeamDyn is the module within FAST that calculates the deflections and dynamic behaviour of the wind turbine blades using Geometrically exact beam theory (see Section 2.8).

For the input file of BeamDyn, the user can select the blade model and its parameters. The module needs an input file containing a geometry description using points and interpolation to define the shape along with which order of interpolation, material parameters, control parameters and time step per iteration.

### 4.1.5. ElastoDyn

ElastoDyn is the module that contain the structural models of the rotor, drivetrain, nacelle, tower and the platform. It's inputs are the aero-dynamic loads from AeroDyn and controller commands, but also hydrodynamic loads and substructure reactions which are not relevant for this thesis. ElastoDyn calculate the deflections of the structure, its velocities and accelerations along with the reactions loads. The input file for ElastoDyn require the degrees of freedom to be accounted for. For example, bending of the tower structure 1.st and 2.nd Fore-Aft Mode, 1.st and 2.nd flap mode, drivetrain, yaw direction and generator etc. There are a total of 24 available degrees of freedom for a 3-bladed wind turbine seen in Figure 4.2. ElastoDyn also require initial conditions like rotor speed, initial blade pitch angle, and initial conditions of deformations of structure. ElastoDyn also contain turbine configuration parameters like number of blades, distance from the rotor apex, to the rotor blades, centre of mass of different parts of the structure. The degrees of freedom used for this case study is presented in table 5.2

ElastoDyn provide the main output parameter that this thesis is investigating, which are the forces acting on the bearing between the nacelle and tower.

## 4.2. Output file

FAST v8 generates an output file "your\_case".out. The output file will contain the values of each parameter which have been specified in the input files for each time step. This subsection discuss the relevant output parameters.

**YawBrMzp** is the moment about the z-axis of the bearing in the nacelle or the yaw moment. The value is based on the forces that the wind turbine blades are subjected to, transferred through the blades, to the hub and nacelle.

**YawBrMyp** is the moment about the y-axis, or the pitch moment. We expect to see some variation in load from P1, P2 and P3 vibrations in an FFT.

**YawBrFzn** represent the Tower-top yaw bearing axial force.

To see an example of how the different modules are connected to fast see Figure 5.6. Examples plotted output files are presented in Figure 6.8.

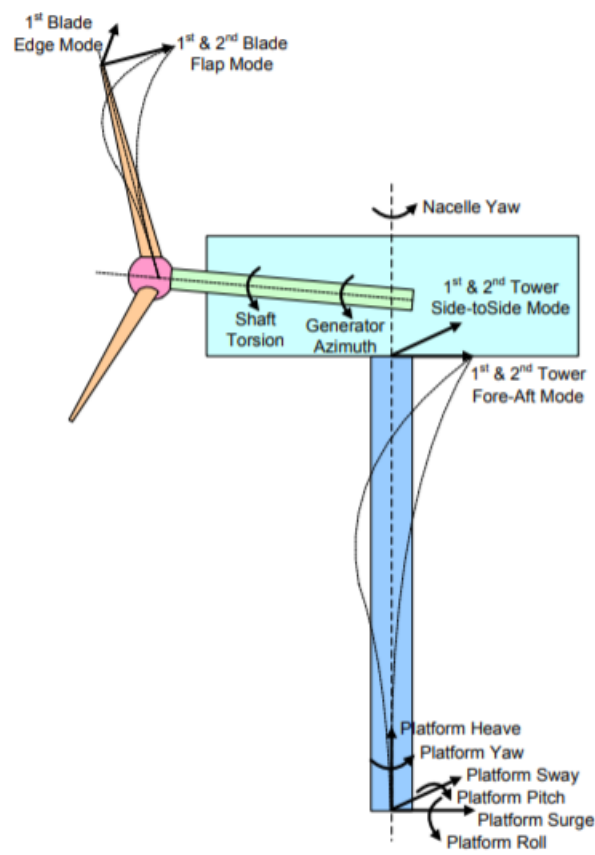


Figure 4.2: The degrees of freedom for a two or three bladed wind turbine in ElastoDyn



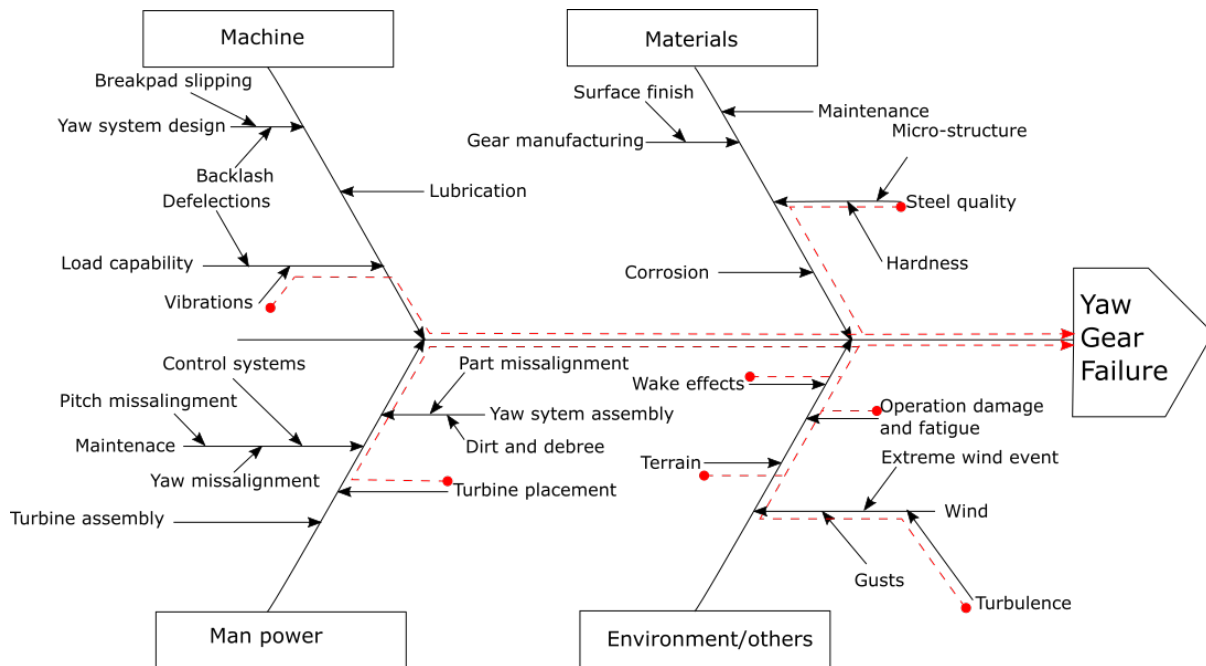


Figure 5.1: Ishikawa or Fish-bone diagram, commonly used to identify the cause and effects of a failure. Figure based on *OnyxInsight, Root cause analysis* 2016.

## 5. Method

The method section describes the processes which this thesis has used to evaluate the full case study. The CFD simulations, the FAST v8 simulations, the Matlab Rainflow analysis based on the measured values from the wind mast. At the end of the method section there is a software flowchart showing the process which is used to find the results of this thesis, see Figure 5.6.

To analyse the cause and effects and to identify the possible causes for the yaw failure an analysis tool like the Fish-bone diagram is useful. The Fish-bone diagram functions as a visualisation tool and is used to find the failure causes and to show how the different failures are connected. The tool can be used to specify which factors are likely to be the reason for a failure, and can be used later prevent the failures by identifying the root cause of the problem. Indicated by the red lines, the figure show what this thesis has used as focus points, turbulence, terrain, fatigue damage and wake effects from environmental loads, the placement of turbines, which is connected to the turbulence, shown in Section 6.2. Vibrations and loading forces due the the environmental conditions and fatigue damage based on the material properties.

The Fish-bone diagram in Figure 5.1 have sampled some of the failure causes that could lead to a failure in the yaw gears. This study only have information about the wind conditions from a specific wind mast, which makes most of these failure causes difficult to analyse and is a limitation for this thesis. The wind data provided does give us information about the turbulence which is why this study focus on the effect of vibrations and material fatigue.

### 5.1. CFD Simulation

#### 5.1.1. Environmental loads

The wind speed affecting the wind turbine is highly variable, changes in mean velocities, turbulence produced by topology or even other turbines and wind gusts all have an impact on the turbine and generally, when designing a wind turbine a deterministic gusts, steady inflow

winds and oblique wind flow have been used to determine the parameters. However, the growing trends of fatigue failure require a look at the turbulence and the vibrations induced by the turbulence, which then require a wind model that can represent that turbulence (R. Gasch 2012 and M. O. L. Hansen 2008).

The wind data from the wind turbine farm gives an annual average wind speed of 7.56 m/s, and a maximum wind speed at 25.11 m/s. During the months from September to February and March the site experience higher average wind speeds, reaching a maximum average wind speed in October at 9.14 m/s. See Table 5.1.

Table 5.1: The mean and maximum wind velocities for each month in 2017 and 2018. The wind measurements are taken from the wind mast located at the site.

Month	Wind speed (m/s)	
	Mean	Maximum
January	7.42	22.27
February	8.38	21.97
March	8.13	24.65
April	7.15	22.41
May	6.20	12.78
June	No available data	No available data
July	5.34	16.96
August	5.31	18.67
September	7.66	24.81
October	9.14	22.03
November	7.81	21.09
December	7.86	25.11

### 5.1.2. Case study

The wind park has wind turbines located at different parts of the terrain and spread over different areas or zones. OpenFOAM can be used to study the effect of the terrain by looking at the turbulence being generated by the flow over it. As mentioned before, OpenFOAM contains several tutorials, among which is “turbineSiting”. “turbineSiting” is a predefined case where two turbines have been installed on a terrain for the purpose of looking at the turbulent wake that are being generated from the turbines.

### 5.1.3. Mesh generation

The terrain profile can be found from an open access web-page “hoydedata.no”, provided by “Statens Kartverk”. Where it is possible to download up to 0.25m resolution topography files. The data can be extracted using two types of files. LAS files, which are essentially point clouds containing x, y, and z location for each measurement, where each measurement is a point on the surface, generally these LAS files can be several giga bytes big and rely on compression tools, most commonly compressed to LAZ files. These files are very detailed and contain a lot of information. DEM files contain information in a evenly spaced grid with elevation data for each point, much like the LAS files, but less detailed. For the purpose of this thesis, DEM files are sufficient.

Paraview contains all the filters needed to convert the .DEM file into an .STL file usable by OpenFOAM. The process is described below. In Figure 5.5 is the .STL file of the terrain profile as viewed in paraView.

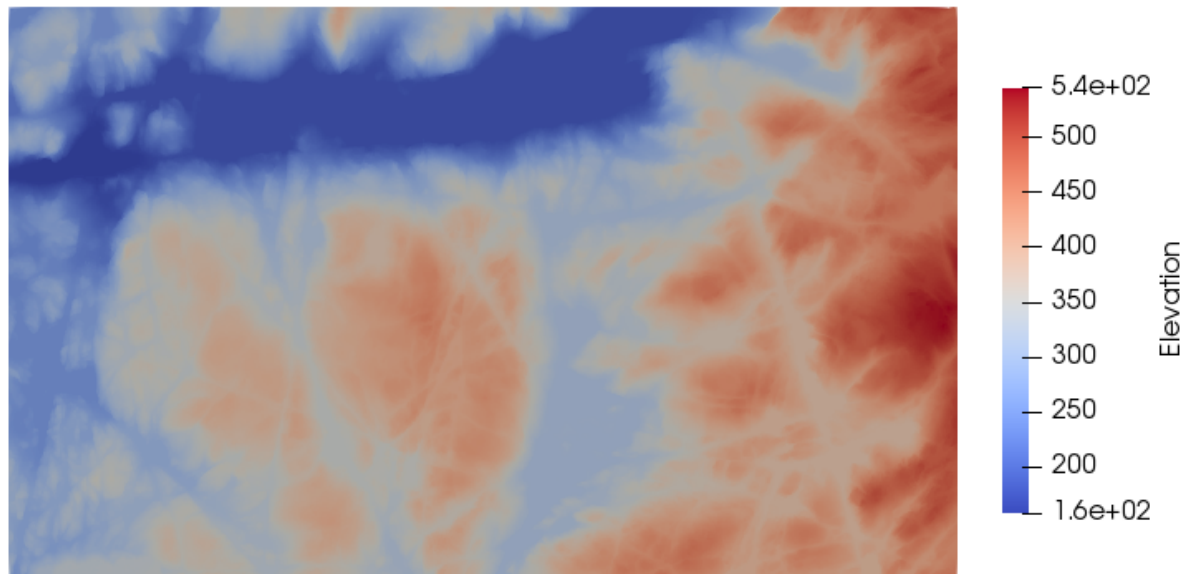


Figure 5.2: Paraview representation of the DEM file using the built in DEMreader function. A 2D plane containing information about the height of each point. Blue to red are low to high respectively. The height is meters above sea-level.

1. Download the .DEM file from [hoydedata.no](http://hoydedata.no)
2. Upload the .DEM file in Paraview using the DEMreader function. Figure 5.2 show how the DEMreader represent the 2D .DEM file.
3. Use “Threshold” filter to remove unwanted information, set elevation data from 0 to the maximum height (544.1 m)
4. “WarpByScalar” translate the points on the 2D surface into a 3D surface using the elevation data as input.
5. “Tetrahedralize” seen in the Figure 5.3, changes the squares to triangles
6. “Extract surface” filter creates an .STL file from the tetrahedral surface.
7. “Decimate” filter to compress the .STL file.

With the .STL file created, and imported into the case folder in “Your\_case/constant/triSurface” under the name “terrain.STL”. The files under system need to be manipulated to suit. This include changing the wind directions in “ABLConditions” found in “your\_case/0/include” Changing the inlet and outlet patch to suit the wind direction in blockMeshDict in “system” folder, along with the corners for the terrain. “snappyHexMesh” need to contain the coordinates of the turbines, along with the name of the .STL file. Check refinement level to be suitable for the case, and select a coordinate in “locationInMesh” over the terrain (selecting over or under the terrain file to be the domain where the case is run). “topoSetDict” in “system” folder need to contain the coordinates of the wind turbines (a plate or box) where the turbines blades rotate.

To simulate the effect of a wind turbine, the boxes act as porous plates allowing some air to pass through. Wake rotation is not included in this model but is sufficient to visualise the wake. “fvOptions” in the “constant” folder, describe the area and the porosity of the plates representing the rotor blades. The “upstreamPoint” should be placed upstream of the wind turbines at a location where the wind turbines has not yet affected the air flow. “upStreamPoint”

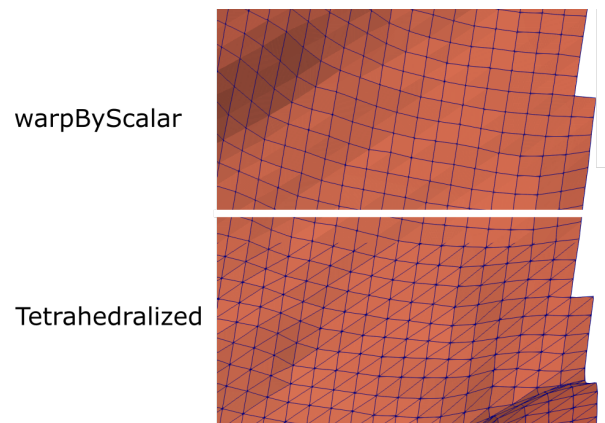


Figure 5.3: The filter “Tetrahedralize” changes the grid to contain triangles instead of squares. The triangles are used by the “Extract Surface” filter to create the .STL file.

select the location for the reference wind conditions used to calculate the effect of the porous plates. The “discDir” vector should be facing towards the air flow.

To generate the mesh in the Ubuntu operating system, while in the case directory, run the commands:

1. blockMesh
2. snappyHexMesh -overwrite
3. topoSet

“blockMesh” create the outer bound of the domain, generating side patches, ground, top and inlet and outlet patches. “snappyHexMesh” generate a 3-dimensional mesh containing hexahedra and split-hexahedras, from the triangulated .STL file. The mesh conforms to the .STL file approximately through iterations, until it reaches a predefined mesh quality. “topoSet” create a cellSet at the location at which the turbines have been placed in the topoSetDict file. An example of this grid can be seen in Figure 5.4

#### 5.1.4. SimpleFoam

SimpleFoam in OpenFOAM, is a steady-state solver for the incompressible turbulent flow using a technique called “Semi-implicit Method for Pressure linked Equations” (SIMPLE). SIMPLE solves the Navier-Stokes equations in a numerical step by step process looking for a converged solution. (*OpenFOAM: SimpleFoam* 2019)

1. Initial pressure guess,  $p^*$
2. Solve momentum equations for  $u^*$
3. Solve pressure correction equations for  $p'$
4. Correct pressure and velocity
5. Iterate 2-4 until convergence

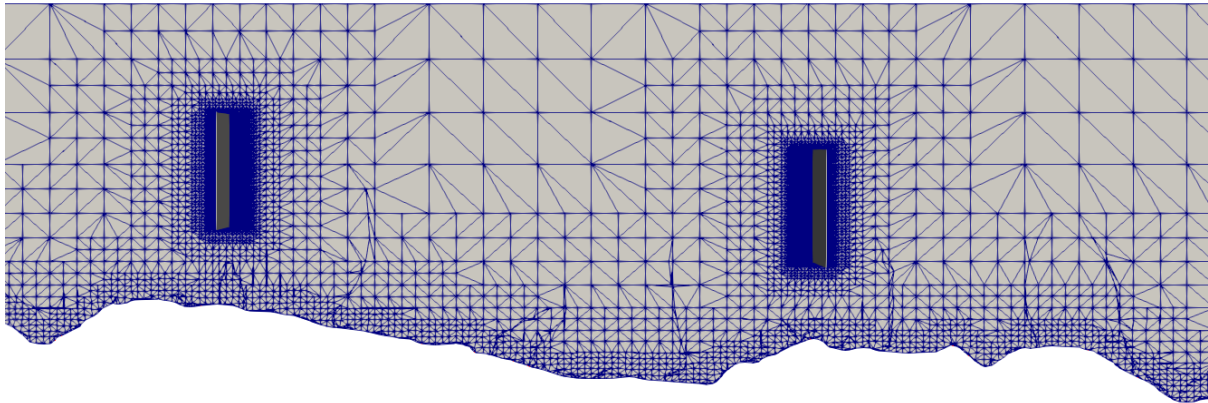


Figure 5.4: The grid generated by snappyHexMesh. The grid gets more refined closer to the ground and to the wind turbines (represented as squares).

The simulation will run until it reaches a maximum iteration, or until the solution reaches a minimum residual, i.e the solution has converged.

To tune simulations, to give the most accurate results possible it is important to use actual measurements to compare the CFD results to measured values. The wind park has a wind mast installed at a location far from the wind turbines this thesis is interested in investigating. The best option would be to simulate the entire wind farm, along with all the different wind turbines and wind mast, but the area is too large and can't be done without stitching maps and having larger computers. The second option is to make three different terrain files, each covering the areas and wind turbines of interest, and then calibrate the three simulations according to the wind mast measurements.

The atmospheric boundary conditions used for the simulation parameters is listed in Table 2.3, and examples can be found in Appendix A.

### 5.1.5. Post processing

ParaView is used to visualise the solution from simpleFoam.

The turbulent kinetic energy is solved for every volume element in the mesh, and to visualise the turbulence created by the terrain or by the body of the wind turbine at a specific height, a script is used to translate the value of the kinetic energy at that point, onto the surface of the terrain.

## 5.2. FAST v8 simulation

The simulation of the response of the wind turbine using the turbulence intensity that was calculated from the wind measurements made at the site and modelled in the CFD simulation is performed using FAST v8. The reference wind turbine that is used as a replacement for the Siemens Gamesa 3.0DD wind turbine is delivered by IEA. The IEA turbine is a 3.35 MW wind turbine designed for IEA Wind Task 37 (*IEA Task 37* 2019). IEA Wind task 37 is an international wind research project to coordinate research on wind farms as a system, looking

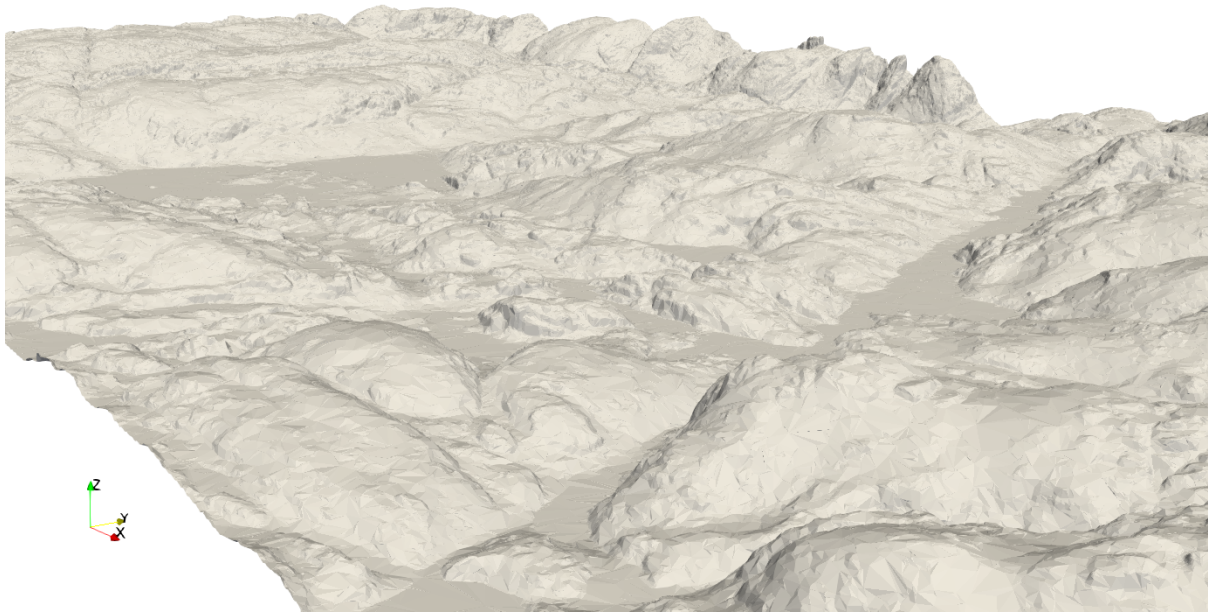


Figure 5.5: .STL file containing topographical data, used by OpenFOAM to simulate the wind conditions.

Table 5.2: Degrees of freedom set in FAST v8 simulations in the ElastoDyn module.

DOF	Description	True/False
FlapDOF1	First flapwise blade mode DOF	True
FlapDOF2	Second flapwise blade mode DOF	True
EdgeDOF	First edgewise blade mode DOF	True
TeetDOF	Rotor-teeter DOF [unused for 3 blades]	False
DrTrDOF	Drivetrain rotational-flexibility DOF	True
GenDOF	Generator DOF	True
YawDOF	Yaw DOF	True
TwrFADOF1	First fore-aft tower bending-mode DOF	True
TwrFADOF2	Second fore-aft tower bending-mode DOF	True
TwrSSDOF1	First side-to-side tower bending-mode DOF	True
TwrSSDOF2	Second side-to-side tower bending-mode DOF	True

at how the effects of different uncertainties in design factors of wind turbine farms change the systems as whole, to reduce cost and increase performance of wind turbine farms.

The turbulent wind conditions are modelled in turbSim, using the turbulent wind conditions at 5, 16 and 20 % turbulence intensity.

Using FAST v8, the dynamic response of the wind turbine is simulated for 1 hour each, plotting the results of the moments and forces induced on the yaw bearing. The results are plotted against time for visualisation. Several more simulations using increasing turbulence are also plotted for damage.

### 5.3. Rainflow Counting analysis

A Rainflow analysis (see Section 2.9) for each turbulence intensity is performed. Due to the lack of information about technical specifications of yaw gear dimensions and material properties, a standard material, see Section 6.4, for material properties and yaw gear dimensions are

used. The dimensions are selected from a standard series produced by Bonfiglioli. The damage accumulation over 25 years is shown in Figure 6.12, and is simply the multiplication for the 1 hour simulation over the course of 25 years, also taking into account the fact that the yaw drive system is only active for about 10% (M.-G. Kim et al. 2014) of the service life of the turbine.

The MATLAB code used to calculate the Rainflow counting algorithm is presented in Appendix D

The results of this thesis are compared to the analysis of S.-W. Kim et al. 2017. The article study the effect of impact load on the gear of a yaw drive pinion motor. The study used a finite element approach to simulate the effects of an impact in yaw pinion spur gear, using a surface to surface contact condition in ABAQUS. The 10-minute simulations use a mean wind speed of 14 m/s up to 24 m/s, its corresponding torque, and a turbulence pattern (turbulence intensity is not given). The damage fraction from one impact is shown in Table 5.3. In the report they also show a 25 year lifetime damage accumulation, and for the higher wind speeds and torque, they find a damage accumulation above 100%, meaning a structural failure of the spur gear..

The calculated damage fraction for an impact load is estimated to be around  $5.3 \times 10^5$  to  $1.5 \times 10^7$  times larger than the damage fraction where the impact of the backlash is not taken into account. The previous study found that when the wind speed operated at around 24 m/s and when taking into account the impact damage, the damage accumulation exceeded the life of the spur gear. The study did not evaluate the changing turbulence intensity, but the results from this thesis indicate a clear increase in stress amplitude as the turbulence increase.

Table 5.3: Damage fraction for one impact in each simulation case. Ref S.-W. Kim et al. 2017

Torque (applied by motor)	Whether and impact load occurs or not	Damage fraction
3300	Yes	$6.90 \times 10^{-14}$
	No	$4.48 \times 10^{-21}$
3700	Yes	$3.64 \times 10^{-13}$
	No	$3.32 \times 10^{-20}$
4000	Yes	$1.05 \times 10^{-12}$
	No	$1.38 \times 10^{-19}$
4600	Yes	$2.61 \times 10^{-12}$
	No	$1.56 \times 10^{-18}$
5300	Yes	$2.16 \times 10^{-11}$
	No	$2.14 \times 10^{-17}$
6000	Yes	$1.01 \times 10^{-10}$
	No	$8.35 \times 10^{-17}$
6600	Yes	$4.91 \times 10^{-10}$
	No	$1.34 \times 10^{-15}$
7200	Yes	$2.26 \times 10^{-9}$
	No	$4.24 \times 10^{-15}$



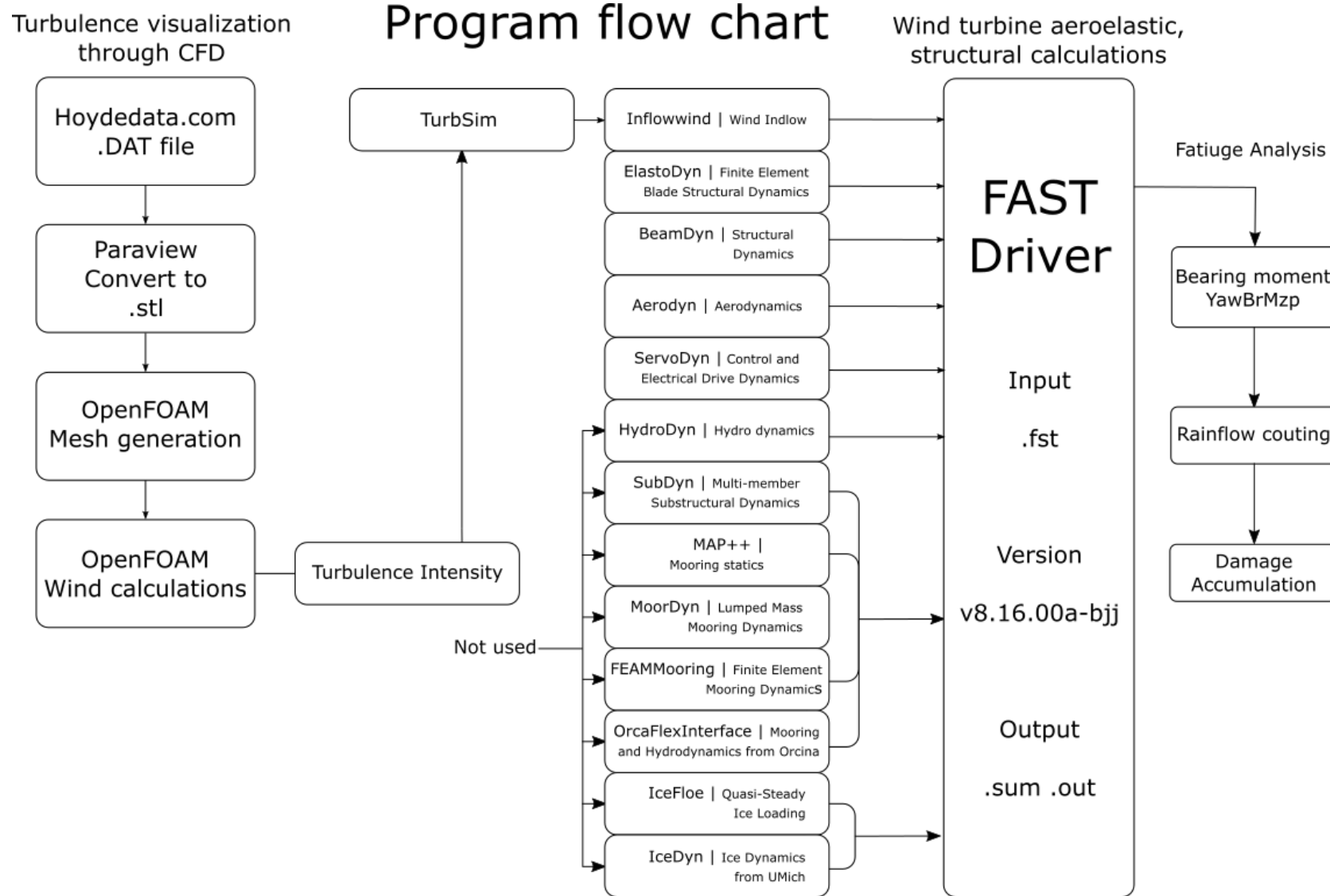


Figure 5.6: A chart of how the different software's and the different modules that are used to produce the output files of FAST v8.



## 6. Results and Discussion

The following subchapters cover the results of the thesis. Divided into the three main subchapters, CFD simulations, FAST v8 simulations and a fatigue analysis. The section present and discuss the results of this thesis, and compare the results with other similar studies.

### 6.1. Wind measurements

The average wind measurements from this case study is presented in Table 6.1. The results are condensed from almost two years of measurements and indicate an average turbulence intensity above the design standard of a class IIA wind turbine according to IEC.

Table 6.1: Wind measurements taken from a wind mast at the wind farm.  
 $I$  is calculated using Eq. (1.1)

(m)	$U$ (m/s)	$\sigma$ (m/s)	$I$ (%)
39	6.67	1.46	21.94
58	6.29	1.26	20.05
85	7.68	1.40	18.21
89	7.50	1.35	17.97
92	7.56	1.34	17.77

The turbulence intensity is calculated using the DNV GL standard from Equation 1.1 and is visually presented in Figure 6.1 for the entire measurement period, along with the gaps in measurements. The measurements are taken in proximity of another turbine and could be affected by its wake, but then again, also indicate how the wake effect of other turbines could increase the turbulence to exceed the design standards.

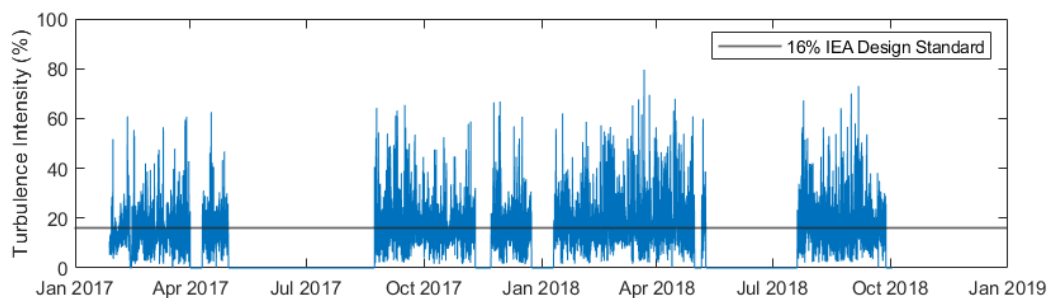


Figure 6.1: The turbulence intensity calculated for every 10-minute measurement. 16% Turbulence intensity is shown as a black line, and is the IEC design standard for IIA turbines.

### 6.2. CFD Simulations

The windrose in Figure 6.2 show the predominant wind direction as approaching from north, north-east and from east, south-east with the strongest winds from the east. The simulations performed in the CFD analysis use the eastern wind as a baseline. Taking the 10 m/s as parameters for the simulation. There is a significant amount of higher wind speeds between 15 to 25 m/s. The CFD simulations run through OpenFOAM give the results presented in Table 6.2.

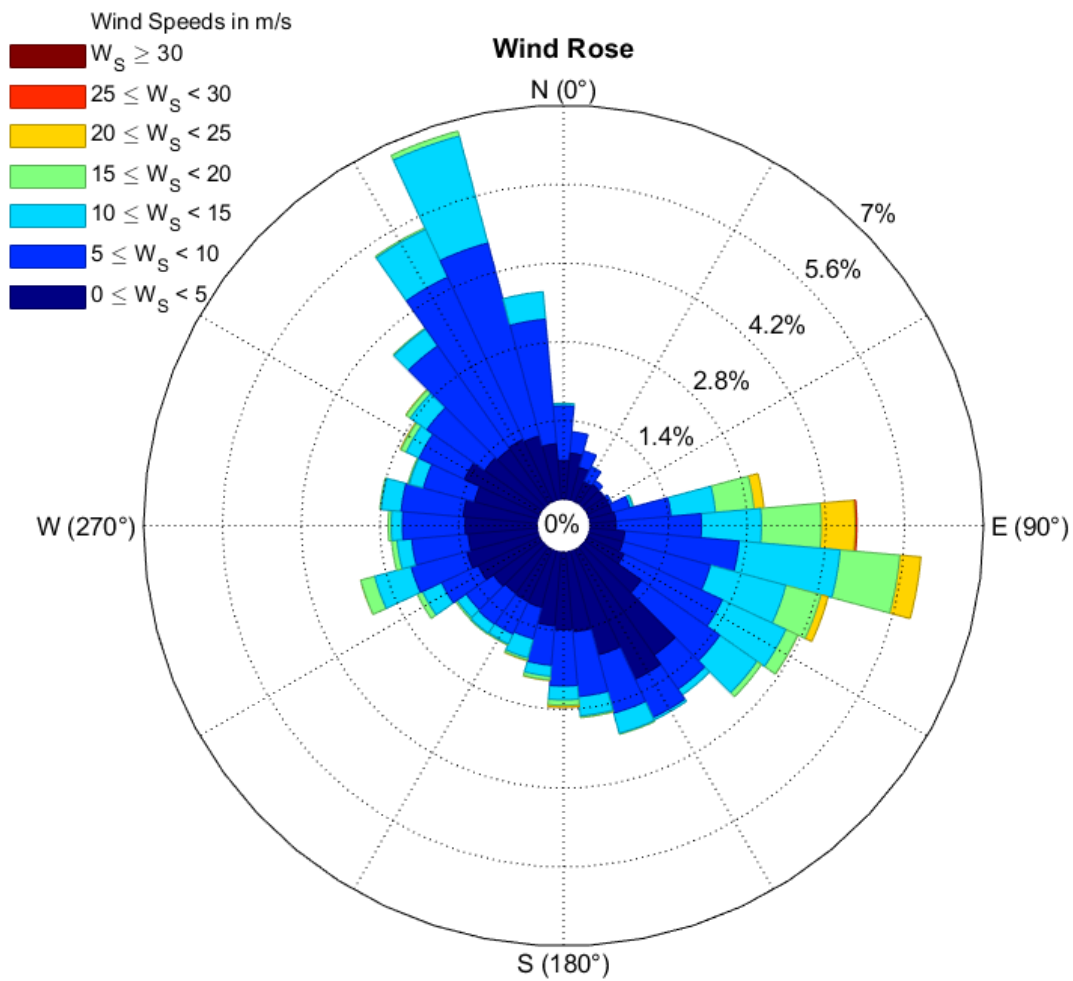


Figure 6.2: A windrose visualising the wind directions. Generally wind is blowing from the north, north-west and east, south-east. MATLAB code borrowed from *Wind Rose* 2019

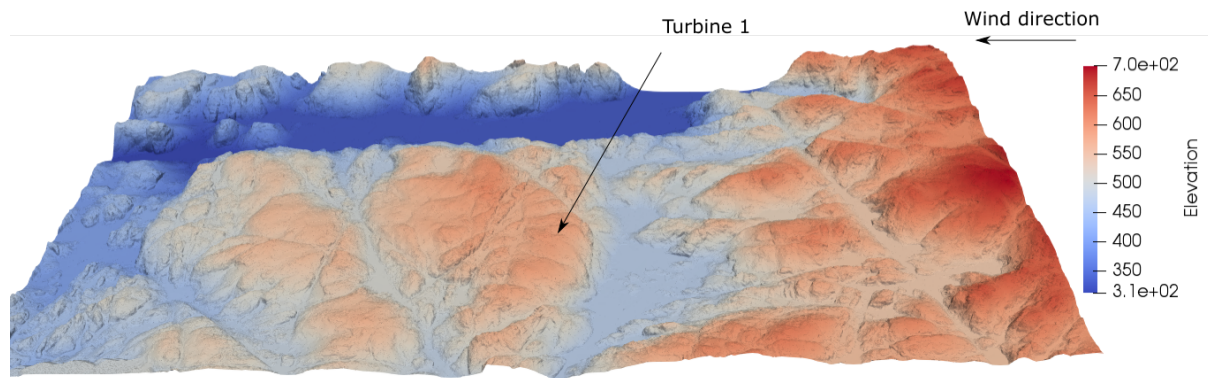


Figure 6.3: Terrain and location of Turbine 1

Figure 6.3 and Figure 6.5 show the locations of the turbines 1, 2 and 3 that this thesis has focused on. The surrounding terrain is modelled from an .STL file in Paraview. The wind park has a total of 50 turbines placed in different location in the terrain, but most of these turbines are far away or placed in such a way that their wake does not interact with the three studied turbines. The figures show hilly terrain, with waters and mountain tops and each span over 3 to 5 kilometres. The size of the terrain files need to be long enough for the flow in the CFD simulation to develop.

The position of Turbine 1 is located at a top of a hill behind a steep incline, see Figure 6.3. The wind travel predominantly from left to right. The wind turbine is situated as the first in a row of turbines and with a wind direction approaching from the east, the turbine will experience no wake effects from other surrounding wind turbines.

The wind in Figure 6.4 is moving from left to right at 10 m/s. The figure show an increase of turbulent kinetic energy as the wind approach peaks in the terrain. The steep declination to the east of the turbine has an impact on how the wind will move as it is approaching the turbine. The kinetic energy from the simulations at the site of Turbine 1 in Table 6.2, give a turbulence intensity ranging between 15% and 20%. This turbulence intensity is just at or above the design specifications of a IIA wind turbine (see table 2.1).

Figure 6.4 is showing the turbulent kinetic energy of the wind is it flow over the terrain. The figure is a 2D view of of the terrain, passing right through the location of Turbine 1 in Figure 6.3. The simulation show an increase in turbulent kinetic energy right in front of and above peak values in the terrain, indicating turbulent wind condition at these locations.

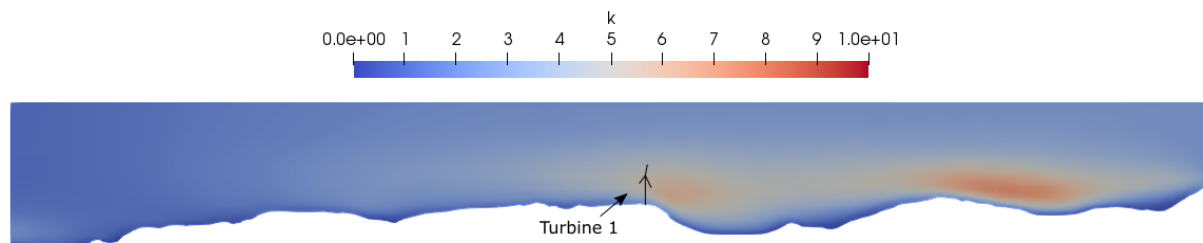


Figure 6.4: The figure show the turbulent kinetic energy,  $k$ , as the wind progress over the terrain in Figure 6.3. The turbine in the figure is not implemented in the simulation but is added to show scale.

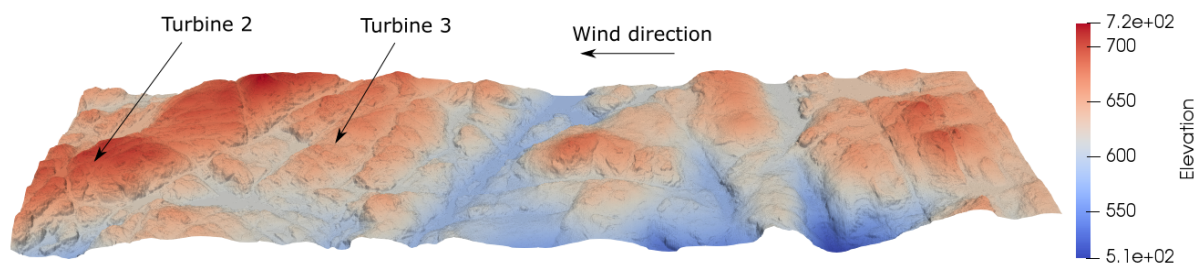


Figure 6.5: This figure shows the location of wind turbine 2 and 3. Its is located at a top of a hill, behind an incline. However, the incline is more shallow than compared to Turbine 1. The wind travel predominantly from left to right. Wind turbine 2 is situated as the second in a row of turbines. There will be wake effects when the wind is blowing from the east due to the positioning of Turbine 3. Note that the scale is for elevation.

In Figure 6.5 the position of Turbine 2 and 3 is marked by arrows along with the wind direction. Turbine 2 is located behind turbine 3 and will experience a wake effect as the wind approach from the east. The wake effect will increase the turbulent kinetic energy of the air, and cause turbulence, which again will induce vibrations and higher loads on the wind turbine. Figure 6.7 is a top down view of the same terrain. Showing how the positioning of wind turbine 3 will case a wake effect on wind turbine 2. Because the model of the turbines in this figure are only represented as non-rotating porous plates they do not model any rotating wake effect. The wake effect in 6.7 is only a visual representation of how the wake effect looks like. The magnitude and shape is not necessarily accurate. The wake effect is a much discussed subject, and studies have increased in recent years as the building of wind farms has increased world wide. There are several studies (Thomsen et al. 1999)(S.-H. Kim et al. 2015)(Karlina-Barber et al. 2016) of the wake effect, and how the fatigue life of a wind turbine increase as the distance between the turbines increase. Usually, when discussing the distance between turbines, the distance is relative to the diameter of the wind turbine rotor blades. The study by Thomsen and colleagues show a load increase of up to 15% for wind turbines located inside the wake of nearby wind turbines.

The same kind of pattern in the turbulence can be seen from Figure 6.6, as seen in Figure 6.4. There is an increase in the turbulent kinetic energy at the peaks in the terrain, owing from the fact the the wind speed increase as the parallel flows are forced together. The figure show how the turbulence decrease with the distance from the ground.

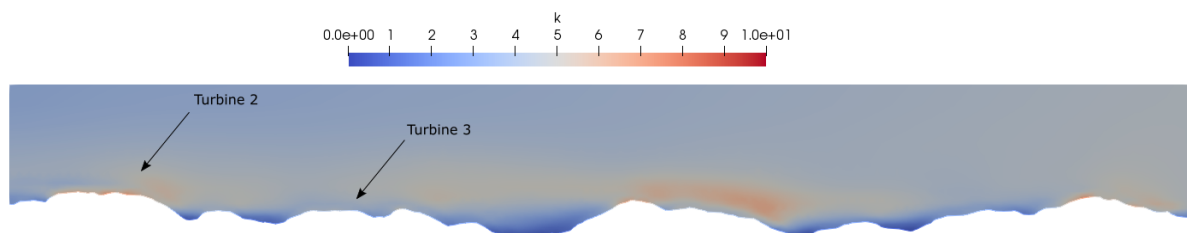


Figure 6.6: The figure shows the turbulent kinetic energy,  $k$ , as the wind progress over the terrain in Figure 6.5. The wind is moving from left to right at 10 m/s. The figure shows an increase of turbulent kinetic energy as the wind approach peaks in the terrain.

Table 6.2: The turbulent kinetic energy compared to the wind speed for each of the two wind turbines. The wind turbines experience no wake effect as the surrounding wind turbines have been removed. The turbulence intensity  $I$  is calculated using Eq. (2.16)

U (m/s)	Turbine 1		Turbine 2		Wind Mast	
	k	I	k	I	k	$I_{sim}$
5	1.05	15.3 %	0.48	15.1 %	0.91	15.6 %
10	5.31	16.4 %	3.42	18.7 %	5.19	18.6 %
15	10.28	18.8 %	7.7	15.1 %	8.52	15.9 %
20	23.9	20.0 %	15.1	15.9 %	15.74	16.20 %

The results of this thesis give clear indication that the positioning of a wind turbine require more than a general wind measurements in surveys. The results gathered from the CFD simulation of the flow conditions, suggest that there are locations in the terrain that are significantly prone to turbulent conditions.

These locations are typically the first locations that comes to mind while placing a wind turbine. Exposed directly to the wind on a hill top. Contrary to this, the best solution is perhaps to move it further back, away from the steep decline, and away from the area with the turbulent energy. A well performed CFD analysis have shown that CFD could be able to determine areas with lower amount of turbulent kinetic energy, which will reduce the vibrations and increase the fatigue life of the wind turbines.

A top down view of the turbulent kinetic energy at hub height projected down onto the terrain gives a good visual representation of the turbulence that the wind turbine experience. This visual representation can be used to identify locations more suitable for positioning regarding the turbulence.

The simulations of the CFD terrain files are very time consuming. Using the openFOAM software, which in it self is complicated, is not ideal regarding the modification of the simulation parameters. Every case have multiple parameters, many of which are uncertain or variable. Parameters related to turbulence and topography, wind speed and wind direction all have a direct impact on the simulated turbulence intensity. The simulations done in this thesis have only focused on very specific wind speeds, and one single direction, using set values for turbulence parameters and topography. It is necessary while performing a survey to look at multiple directions to discover all possible locations where turbulence might be a problem. Using

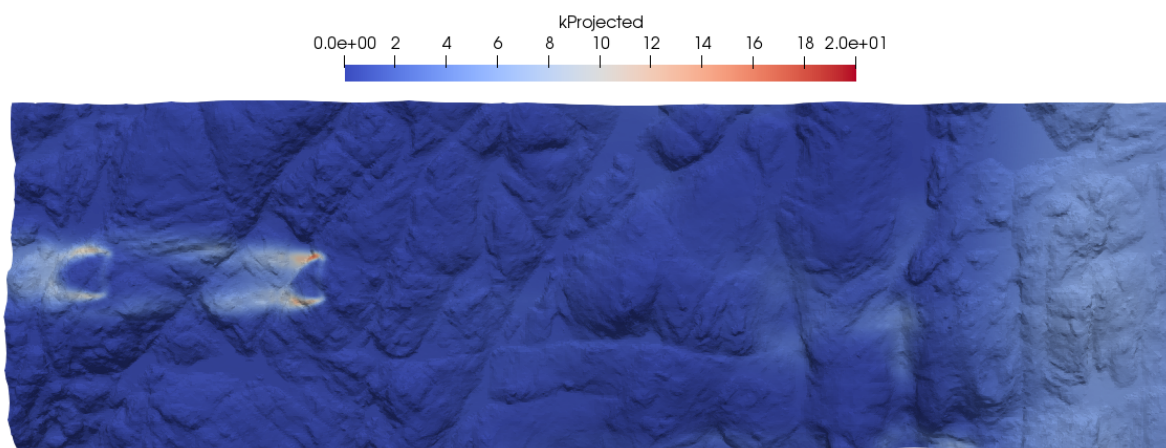


Figure 6.7: Top down view of the wind turbines and terrain, showing the turbulent wake behind the wind turbine. The turbulent wake behind the wind turbine is just a representation of how a wake looks. It does not necessarily have the same turbulent magnitude or shape.

a windrose, it is possible to identify the most prone wind directions and use these as simulation guidelines.

There could exist areas within the wind farm that experience significantly different turbulent wind conditions that have been measured in the survey.



### 6.3. FAST v8 Simulations

This subsection contain the results of the FAST v8 simulations. Using the IEA 3.35 MW wind turbine as a reference turbine.

The simulations used changing turbulence intensities (simulated from TurbSim) to calculate the bending moments that the yaw bearing was subjected to. The graphs of these simulations are presented in figures 6.8, 6.9 and 6.10 and show an increasing amount of noise and vibrations with the increasing turbulence.

Figure 6.8 shows a 5% turbulence intensity. The simulations of the effect of wind turbulence is plotted against time. The wind profile created by TurbSim is shown from the wind speed as the wind in x-direction has a fluctuating value about a mean value of 10 m/s, with a fluctuating y and z value according to the turbulence intensity. The bottom graph in Figure 6.8 shows a Fast Fourier Transform (FFT) showing the peak values at the frequencies P1, P2 and P3 for the pitch movement of the yaw turbine. These peak values can be explained by the shifting weights as the rotor of the turbine rotate about the y-axis. See Section 2.2.2.

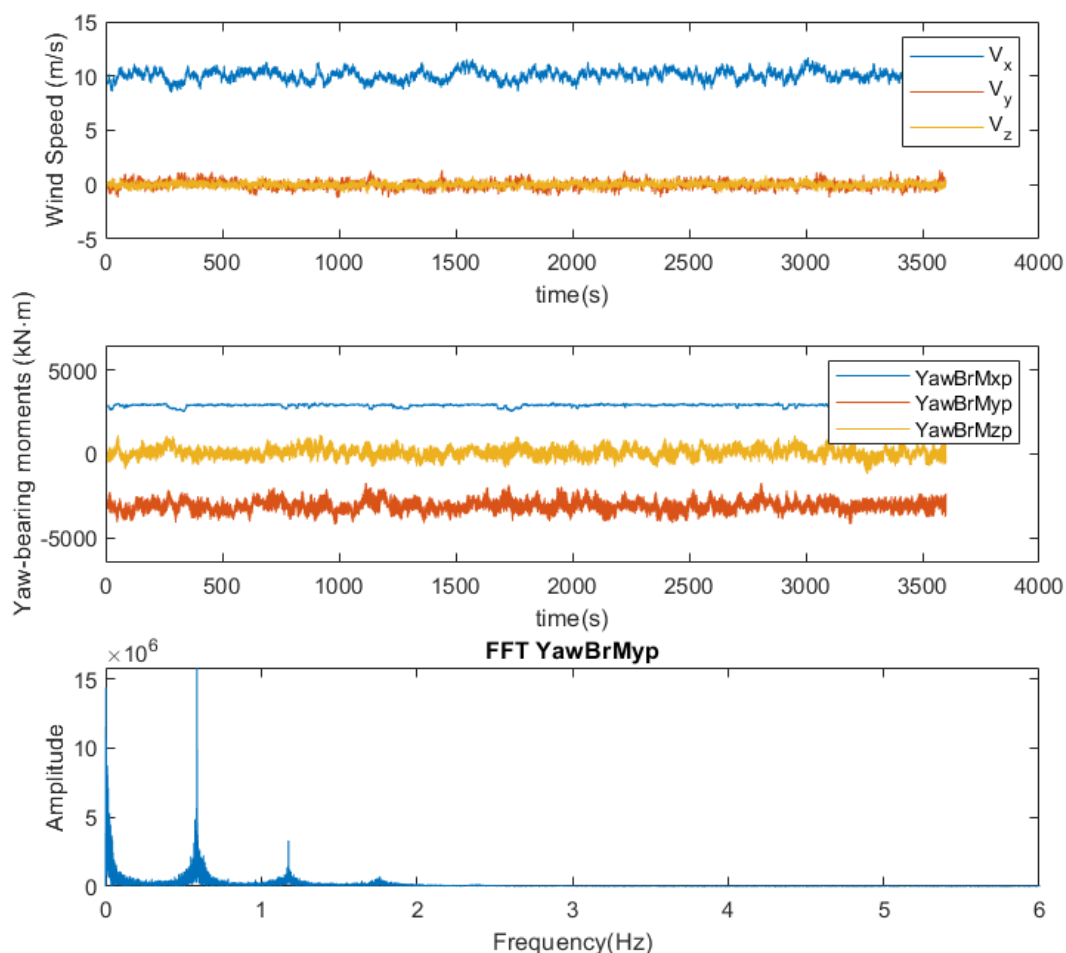


Figure 6.8: The results from FAST v8 with 5% turbulence intensity.

Figure 6.9 is the result of a 16% turbulence intensity. The simulations of the effect of wind turbulence is plotted against time. The wind profile created by TurbSim is shown from the wind speed as the wind in x-direction has a fluctuating value about a mean value of 10 m/s, with a fluctuating y and z value according to the turbulence intensity. The FFT show the peak values at the frequencies P1, P2 and P3 for the pitch movement of the yaw turbine. There is a clear difference between 5% and 16% turbulence intensity, in fact, the difference in amplitude of the yaw moment signal between 5% and 16% turbulence intensity is 3.03 times.

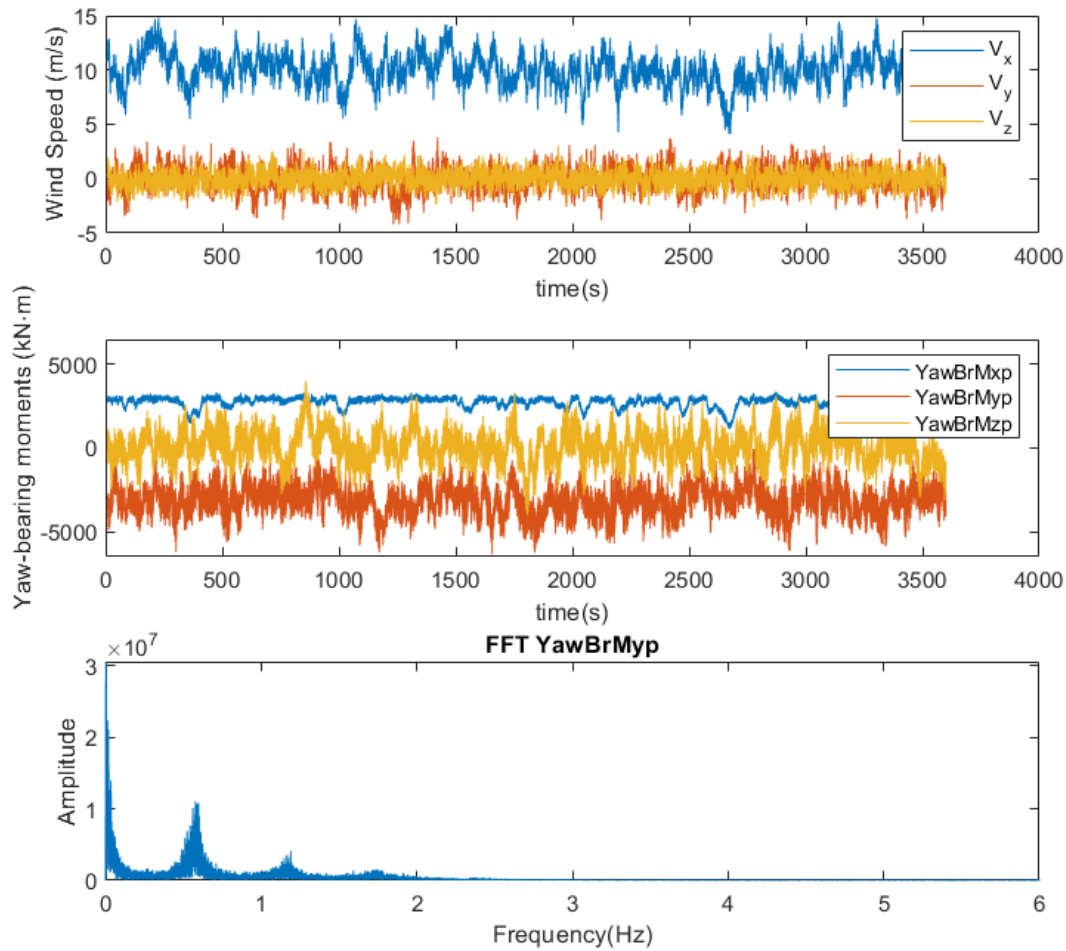


Figure 6.9: The results from FAST v8 with 16% turbulence intensity.

The result of the 20% turbulence intensity simulation is plotted in Figure 6.10. The wind profile created by TurbSim is shown from the wind speed as the wind in x-direction has a fluctuating value about a mean value of 10 m/s, with a fluctuating y and z value according to the turbulence intensity. The FFT show the peak values at the frequencies P1, P2 and P3 for the pitch movement of the yaw turbine.

From the graphs in the figures above it is hard to distinguish any similarities between the moments produced by the wind and the wind speed at any time step. The only trend is that the turbulence increase the amplitudes of the moments, and that the increased turbulence diffuse the FFT signal. The clearly defined peaks in Figure 6.8, become wider and smaller in Figure 6.10. This means that there are more vibrations in the turbine, not only caused by the P1, P2 and P3 events, described in Section 2.2.2.

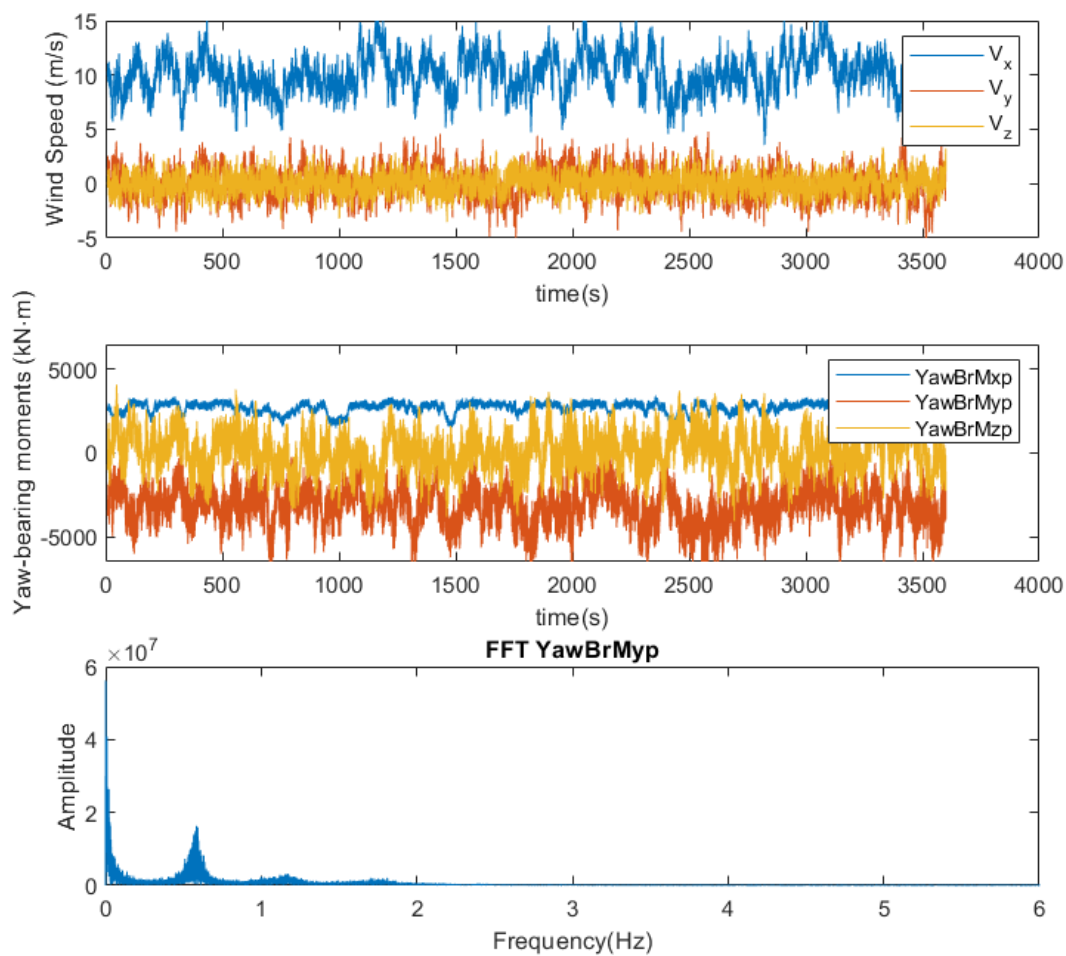


Figure 6.10: The results from FAST v8 with 20% turbulence intensity.

#### 6.4. Rainflow analysis

Figure 6.11 shows how the the amplitude and the mean value of the yaw bearing moment signal behaves with an increasing value of turbulence intensity. The amplitude of the signal has a linear increase as the turbulence increases. Larger amplitudes means highly varying loads that reduce the fatigue life of the structure. The mean value of the bearing moment fluctuate between 80 to 100 kNm, and is relatively low compared to the amplitude.

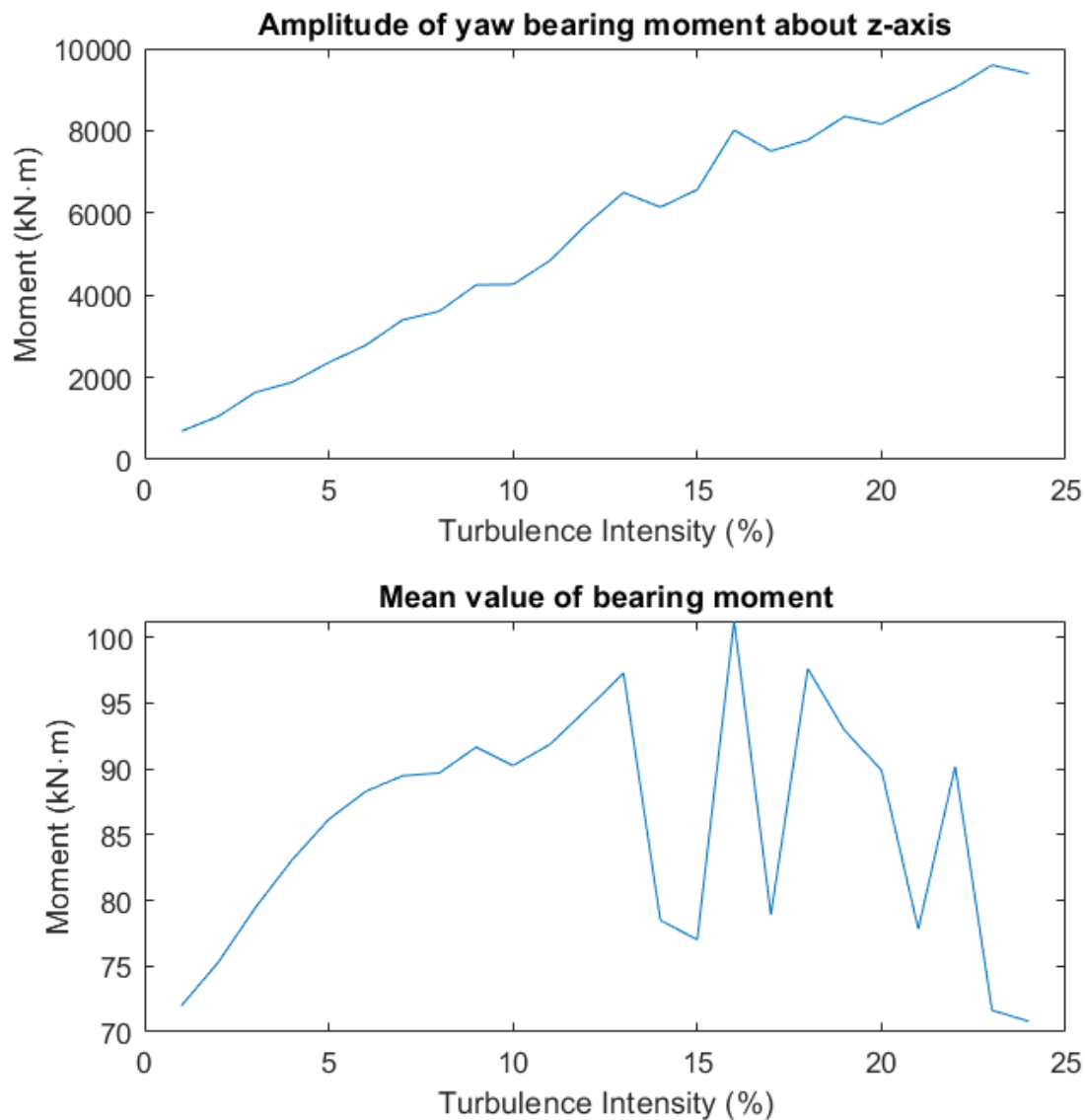


Figure 6.11: Turbulence intensity plotted against the mean (top figure) and amplitude (bottom figure) of the moment on the yaw bearing.

The material chosen for this analysis is taken from S.-W. Kim et al. 2017. The material is a carburized SCM420H steel CrMo alloy. Similar steels to this are ASTM A29 4118, 18CrMo4. Material properties that are needed to calculate the fatigue life is the ultimate limit strength  $S_u$ , and an approximation to the endurance life of the material (Lemu 2016):

$$S_e \approx 0.5S_u \quad (6.1)$$

Table 6.3: The damage accumulated for 1 hour simulation, not taking into account impact damage. Wind speed at 10 m/s, simulated in FAST v8.

Intensity, I (%)	Bearing moment ( $kN \cdot m$ )	Damage accumulation
1	$6.90 \times 10^{02}$	$1.15 \times 10^{-12}$
2	$1.05 \times 10^{03}$	$3.32 \times 10^{-11}$
3	$1.64 \times 10^{03}$	$5.56 \times 10^{-11}$
4	$1.88 \times 10^{03}$	$6.78 \times 10^{-11}$
5	$2.38 \times 10^{03}$	$4.53 \times 10^{-10}$
6	$2.77 \times 10^{03}$	$5.41 \times 10^{-10}$
7	$3.38 \times 10^{03}$	$6.64 \times 10^{-10}$
8	$3.61 \times 10^{03}$	$7.06 \times 10^{-10}$
9	$4.24 \times 10^{03}$	$6.77 \times 10^{-09}$
10	$4.25 \times 10^{03}$	$6.79 \times 10^{-09}$
11	$4.83 \times 10^{03}$	$7.68 \times 10^{-09}$
12	$5.75 \times 10^{03}$	$9.07 \times 10^{-09}$
13	$6.47 \times 10^{03}$	$1.01 \times 10^{-08}$
14	$6.16 \times 10^{03}$	$9.42 \times 10^{-09}$
15	$6.54 \times 10^{03}$	$1.02 \times 10^{-08}$
16	$7.99 \times 10^{03}$	$1.25 \times 10^{-08}$
17	$7.49 \times 10^{03}$	$1.94 \times 10^{-07}$
18	$7.76 \times 10^{03}$	$2.00 \times 10^{-07}$
19	$8.35 \times 10^{03}$	$2.14 \times 10^{-07}$
20	$8.13 \times 10^{03}$	$2.07 \times 10^{-07}$
21	$8.60 \times 10^{03}$	$2.19 \times 10^{-07}$
22	$9.05 \times 10^{03}$	$2.26 \times 10^{-07}$
23	$9.53 \times 10^{03}$	$2.40 \times 10^{-07}$
24	$9.35 \times 10^{03}$	$2.31 \times 10^{-07}$

For simplification, and since the mean stress is relatively low compared to the amplitude we assume the stress range is equal to  $R = -1$ . The stress amplitude fluctuate about a mean value of 0.

In Table 6.3 the results of this thesis Rainflow analysis is presented as turbulence intensity, and its respective damage accumulation for 1 hour of simulation. There is a good correlation between no impact load calculated in this thesis and the results from Table 5.3.

The damage accumulated for 25 years of operation, in Figure 6.12, show how the the expected damage of the yaw bearing increase. The exponential increase in damage as turbulence increase show how important it is to have turbulence in mind while designing at wind turbine.

Simulating a mean wind speed of 20 m/s, using 20% turbulence intensity in FAST v8 yields a damage accumulation over 25 years at 14.7%. The frequencies of the signal produced by the YawBrMzp matches the signal shown by the S.-W. Kim et al. 2017. However, the amplitudes of the yaw moment produced by the wind and vibrations are different. It is reasonable to assume a higher damage due to higher turbulence, shown by the increased amplitudes and highers yaw moments in Figure 6.11.

The difference between the damage accumulation with 1% turbulence intensity and 17% turbulence intensity is close to a million times. The increase of damage accumulation is exponential with the increase of turbulence intensity.

The results from the 1 hour simulations, and now taking into account the lifetime of the wind turbines operating for 25 years, the yaw pinion gears will soon start to approach their fatigue

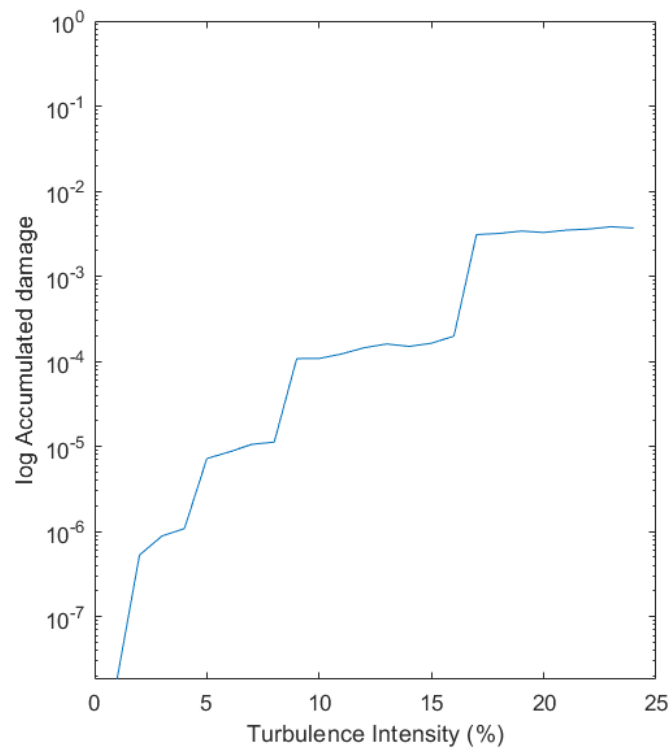


Figure 6.12: Accumulation of damage for 25 years for the 24 different simulations, each with increasing turbulence intensity.

life they were designed for. Assuming the results in Table 6.3 does not show a conservative damage due to the fact the backlash and impact loads have not been taken into account, there is a real possibility that the fatigue life of these turbines experiencing a strong turbulence have been exceeded.

## 7. Conclusion and future work

### 7.1. Conclusion

The wind measurements taken from the wind mast show a significantly high average turbulence intensity 2% above the classification of the turbine. The measurements were taken in close vicinity of another turbine, which could affect the measurement results.

The results of the CFD calculation show areas of enhanced turbulence intensity which may have a significant impact on the design life of the wind turbine. These areas are usually locations which seem fitting regarding the energy production with high wind speeds but are not optimum from a design point of view.

Wind turbine 2, which had a failure in the yaw pinion gear is located close to, or inside the wake of wind turbine 3, see Figure 6.5. The wake of other wind turbines increase the fatigue loads and can cause failures of structural and mechanical parts, and could be the cause of the failure.

The Rainflow analysis show a significant increase in accumulated damage with increasing turbulence intensity. A similar study by S.-W. Kim et al. 2017 found significant damage on the pinion gears while taking into account the backlash and impact loads associated with the activation of the yaw gears, with an increase of  $10^5 - 10^7$  times.

From the results gathered in this thesis it is shown how the fatigue life of a wind turbine is dependant on the positioning within the complex terrain of a wind farm, typical to onshore projects developed in Norway. The study show how the terrain can increase the turbulence significantly enough to exceed the design standards and cause fatigue damage which eventually lead to downtime and costly repairs. The wake effect and its impact on fatigue is a much studied topic, and the study of the topographically induced turbulence need more attention. This is especially important for turbine placement in rough terrain.

### 7.2. Future work

The wind data used in this thesis are 10 minute mean values with a standard deviation. This data does not give any information about where the frequencies of the turbulence are, and where the energy in the turbulence is most critical. With better resolution data, around 20 Hz a spectral energy analysis can be implemented in a survey (Belu and Koracin 2013). With the data gathered in this thesis, there is no use for a vibrational frequency analysis of the towers.

The model of the pinion motors in the fatigue analysis does not include an impact model. To simulate the damage accurately under each of the conditions, it should be included. Material properties of the yaw gears and exact dimensions of the turbine are usually disclosed information which the property owner does not wish to share, thus information and numbers are based on estimations. A finite element model of the yaw gears taking into account the backlash due to direction change and impact loads will provide a more accurate result.

The simulations do not take into account the effect of the braking and the load reduction due the brakes. Removing the peaks where the load is below the force of the braking, and lowering the amplitude of the peak values equal to the braking force should give more accurate numbers. However, the results presented in this thesis remove about 90 % of damage accumulation, to counteract this. The yaw gear is active for about 10% of the service life of the wind turbine.

The model does not consider the the actual yawing of the wind turbine which produce stresses on the yaw drive due to the large inertia of the rotor blades and nacelle.

## References

- Al-Abadi, A, Y J Kim, Ö Ertuğ, and A Delgado (2016). “Turbulence Impact on Wind Turbines: Experimental Investigations on a Wind Turbine Model”. In: *Journal of Physics: Conference Series* 753. URL: <https://doi.org/10.1088/1742-6596/753/3/032046>.
- ASTM-E1049:85 (2017). *Standard Practices for Cycle Counting in Fatigue Analysis*. Standard. West Conshohocken PA: ASTM International.
- Bauchau, O. A. (2011). *Flexible Multibody Dynamics*. 1st ed. Solid Mechanics and Its Applications 176. Springer Netherlands.
- Belu, Radian and Darko Koracin (2013). “Statistical and Spectral Analysis of Wind Characteristics Relevant to Wind Energy Assessment Using Tower Measurements in Complex Terrain”. In: vol. 2013.
- Betz, Albert, David W. Grissmer, and Jane S Morse (1966). *Introduction to the theory of flow machines*. English. 1st English ed. An edited translation of the original German Einführung in die Theorie der Stromüings-maschinen. Oxford : Pergamon Press.
- Burton, Tony, Nick Jenkins, David Sharpe, and Ervin Bossanyi (2011). *Wind Energy Handbook*. 2nd ed. John Wiley Sons, Ltd.
- DNVGL-RP-C205 (2017). *Environmental conditions and environmental loads*. Standard. DNV GL.
- Faulstich, S, M. Durstewitz, B. Hahn, K. Knorr, and K. Rohrig (2008). “Wind Energy Report Germany 2008”. In: *Institut für Solare Energieversorgungstechnik (ISET)*.
- Hansen, Martin O. L. (2008). *Aerodynamics of Wind Turbines*. 2nd ed. Earthscan.
- Hargreaves, D.M. and N.G. Wright (2007). “On the use of the k- model in commercial CFD software to model the neutral atmospheric boundary layer”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 95.5, pp. 355–369. ISSN: 0167-6105. URL: <http://www.sciencedirect.com/science/article/pii/S016761050600136X>.
- Hau, E. (2006). *Wind Turbines*. 2nd ed. Springer-Verlag Berlin Heidelberg.
- Helsen, J., F. Vanhollebeke, D. Vandepitte, and W. Desmet (2012). “Some trends and challenges in wind turbine upscaling”. In: *Proceedings of ISMA2012-USD2012: International Conference on Noise and Vibration Engineering, Leuven, Belgium, 2012;4345-4360*.
- IEA Task 37 (2019). <https://community.ieawind.org/task37/home>. Accessed: 25/03/2019.
- Jonkman, B.J. and L. Kilcher (2012). *TurbSim user guide version 1.06.00*. Tech. rep. National Renewable Energy Laboratory(NREL).
- Karlina-Barber, Sarah, Sebastian Mechler, and Mario Nitschke (2016). “The effect of wakes on the fatigue damage of wind turbine components over their entire lifetime using short-term load measurements”. In: *Journal of Physics: Conference Series* 753, p. 072022. DOI: 10.1088/1742-6596/753/7/072022. URL: <https://doi.org/10.1088/1742-6596/753/7/072022>.
- Kim, M-G and P H Dalhoff (2014). “Yaw Systems for wind turbines – Overview of concepts, current challenges and design methods”. In: *Journal of Physics: Conference Series* 524, p. 012086. DOI: 10.1088/1742-6596/524/1/012086. URL: <https://doi.org/10.1088/1742-6596/524/1/012086>.
- Kim, Soo-Hyun, Hyung-Ki Shin, Young-Chul Joo, and Keon-Hoon Kim (2015). “A study of the wake effects on the wind characteristics and fatigue loads for the turbines in a wind farm”. In: *Renewable Energy* 74, pp. 536–543. ISSN: 0960-1481. DOI: <https://doi.org/10.1016/j.renene.2014.08.054>. URL: <http://www.sciencedirect.com/science/article/pii/S0960148114005242>.
- Kim, Si-Won, Yong-Seok Lee, Dong-Won Jang, Jungchul Choi, and Soon-Bok Lee (2017). “Fatigue life evaluation of pinion gears for the reliability of pitch systems in wind turbines”. In: *Journal of mechanical science and technology*. 31.2, pp. 753–758. ISSN: 1738-494X.



- Lalanne, Christian (2002). *Mechanical Vibration and Shock: Fatigue Damage*. 2nd. Vol. 4. Hermes Penton Ltd.
- (2014). *Mechanical Vibration and Shock Analysis Volume 4: Fatigue Damage*. 3rd. Vol. 4. John Wiley Sons, Inc.
- Lee, Sang, Matthew Churchfield, Patrick Moriarty, J Jonkman, and J Michalakes (2012). “Atmospheric and wake turbulence impacts on wind turbine fatigue loadings”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 540.
- Lemu, Hirpa G. (2016). *Kompendium: Dimensjonering av maskinelementer*. 1.st. Universitetet i Stavanger.
- Mann, Jakob, Christian Bak, Andreas Bechmann, Ferhat Bingöl, Ebba Dellwik, Nikolay Dimitrov, Gregor Giebel, Martin O L Hansen, Dorte Juul Jensen, et al. (2014). “The Science of Making Torque from Wind 2014 (TORQUE 2014)”. In: *Journal of Physics: Conference Series* 524, p. 011001. DOI: 10.1088/1742-6596/524/1/011001. URL: <https://doi.org/10.1088/1742-6596/524/1/011001>.
- Manwell, J.F, J.G McGowan, and A.L Rogers (2009). *Wind Theory explained: Theory, Design and Application*. 2nd ed. Wiley.
- Moriarty, P.J. and A.C. Hansen (2005). *AeroDyn Theory Manual*. Tech. rep. National Renewable Energy Laboratory(NREL).
- Muyeen, S.M, Junji Tamura, and Toshiaki Murata (2009). *Stability Augmentation of a Grid-connected Wind Farm*. 1st ed. Springer-Verlag London Limited.
- National Renewable Energy Laboratory (2019). <https://www.nrel.gov/about/>. Accessed: 23/05/2019.
- NEK-IEC-61400-1 (2019). *Wind Energy Generation Systems Part 1: Design Requirements*. Standard. Geneva, Switzerland: International Electrotechnical Commission.
- NWTC Information Portal (2019). <https://nwtc.nrel.gov/>. Accessed: 23/05/2019.
- OnyxInsight, Root cause analysis (2016). <https://onyxinsight.com/2016/10/18/developing-solutions-failing-pitch-bearings-wind-turbines/>. Accessed: 29/04/2019.
- OpenFOAM: SimpleFoam (2019). Accessed: 29/04/2019. URL: <http://openfoamwiki.net/index.php/SimpleFoam>.
- OpenFOAM: User Guide, k-epsilon (2019). Accessed: 29/04/2019. URL: <https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence-ras-k-epsilon.html>.
- OpenFOAM v6 User Guide (2019). Accessed: 29/04/2019. URL: <https://cfd.direct/openfoam/user-guide>.
- Ostachowicz, Wiesław, Malcolm McGugan, Jens-Uwe Schröder-Hinrichs, and Marcin Luczak (2008). *MARE-WINT, New Materials and Reliability in Offshore Wind Turbine Technology*. 2nd ed. Springer Open.
- Oyague, F. (2009). “Gearbox Modeling and Load Simulation of a Baseline 750kW Wind Turbine Using State-of-the-Art Simulation Codes”. In: DOI: 10.2172/947884.
- Pffafel, S., S. Faulstich, and K. Rohrig (2017). “Performance and Reliability of Wind Turbines: A Review”. In: *Energies*.
- R. Gasch, Jochen Tewe (auth.) (2012). *Wind Power Plants: Fundamentals, Design, Construction and Operation*. 2nd ed. Springer-Verlag Berlin Heidelberg.
- Ramli, Noram Irwan, M Idris Ali, M Syamsyul, H Saad, and Taksiah Majid (2009). “Estimation of the Roughness Length ( $z_0$ ) in Malaysia using Satellite Image”. In: *APCWE-VII The Seventh Asia-Pacific Conference on Wind Engineering, November 8-12, 2009, Taipei, Taiwan*.
- Tharmaraj, Jesan, C Manonmani, JThulasi Brindha, S Rajaram, PM Ravi, and RM Tripathi (2016). “Estimation of roughness length  $Z_0$  for Kalpakkam site”. In: *Radiation Protection and Environment* 39, p. 44. DOI: 10.4103/0972-0464.185182.

- Thomsen, Kenneth and Poul Sørensen (1999). “Fatigue loads for wind turbines operating in wakes”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 80.1, pp. 121–136. ISSN: 0167-6105. URL: <http://www.sciencedirect.com/science/article/pii/S0167610598001949>.
- Versteeg, Henk Kaarle and Weeratunge Malalasekera (1995). *An introduction to computational fluid dynamics - the finite volume method*. Addison-Wesley-Longman, pp. I–X, 1–257. ISBN: 978-0-582-21884-0.
- Wang, Q., J. Jonkman, M. Sprague, and B. Jonkman (2016). *BeamDyn User’s Guide and Theory Manual*. Tech. rep. National Renewable Energy Laboratory(NREL).
- William Shepherd, Li Zhang (2011). *Electricity Generation Using Wind Power*. World Scientific Publishing Company.
- Wind Rose* (2019). Accessed: 06/02/2019. URL: <https://se.mathworks.com/matlabcentral/fileexchange/47248-wind-rose>.

## A. OpenFOAM input files

The following Appendix contain text input files from OpenFOAM CFD calculations. The following text files are examples for one simulation.

### ABLConditions

```
/*-----* C++ -*-----*\
=====
\ \      F i e l d      |   OpenFOAM: The Open Source CFD Toolbox
 \ \      O p e r a t i o n |   Website:  https://openfoam.org
  \ \      A n d           |   Version:   dev
   \ \      M a n i p u l a t i o n |
=====
\*-----*/

Uref          10; // Wind speed at hub height
Zref          109; // Hub height
zDir          (0 0 1);
flowDir       (-1 0 0);
z0            uniform 0.3;
zGround       uniform 650;
// ***** //
```

### blockMeshDict

```
/*-----* C++ -*-----*\
=====
\ \      F i e l d      |   OpenFOAM: The Open Source CFD Toolbox
 \ \      O p e r a t i o n |   Website:  https://openfoam.org
  \ \      A n d           |   Version:   dev
   \ \      M a n i p u l a t i o n |
=====
\*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
// ***** //

convertToMeters 1;

vertices
(
    (0      0      200) // 0
    (3250   0      200) // 1
    (3250   980    200) // 2
    (0      980    200) // 3
    (0      0      1000) // 4
    (3250   0      1000) // 5
    (3250   980    1000) // 6
    (0      980    1000) // 7
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (60 20 20) simpleGrading (1 1 2)
);

edges
(
```

```

);

boundary
(
    outlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }
    sides
    {
        type patch;
        faces
        (
            (3 2 7 6)
            (0 1 4 5)
        );
    }
    inlet
    {
        type patch;
        faces
        (
            (1 2 5 6)
        );
    }
    ground
    {
        type wall;
        faces
        (
            (0 3 2 1)
        );
    }
    top
    {
        type patch;
        faces
        (
            (4 5 6 7)
        );
    }
}
);

mergePatchPairs
(
);

// ***** //

controlDict

/*----- C++ -----*/
=====
\\ \\      F ield      | OpenFOAM: The Open Source CFD Toolbox
\\ \\      O peration   | Website:  https://openfoam.org
\\ \\      A nd         | Version:   dev
\\ \\      M anipulation |
=====

```



```

    default          Euler;
}

gradSchemes
{
    default          Gauss linear;
    grad(p)          Gauss linear;
}

divSchemes
{
    default          none;

    div(phi,U)       bounded Gauss upwind;
    //div(phi,U)     bounded Gauss linearUpwindV grad(U);
    div(phi,epsilon) bounded Gauss upwind;
    div(phi,k)        bounded Gauss upwind;

    div((nuEff*dev2(T(grad(U)))) Gauss linear;
}

laplacianSchemes
{
    default          Gauss linear limited corrected 0.33;
    //default        Gauss linear corrected;
}

interpolationSchemes
{
    default          linear;
}

snGradSchemes
{
    default          limited corrected 0.33;
}

```

// \*\*\*\*\* //

**fvSolution**

```

/*-----* C++ -*-----*\
=====
\\      /   F ield      |   OpenFOAM: The Open Source CFD Toolbox
\\    /    O peration   |   Website:  https://openfoam.org
\\  /      A nd         |   Version:   dev
\\ /       M anipulation |
\*-----*

```

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       fvSolution;
}
// *****

```

```

solvers
{
    p
    {
        solver      GAMG;
        tolerance   1e-7;
    }
}

```

```

        relTol          0.1;
        smoother        GaussSeidel;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance        1e-8;
        relTol          0.1;
        nSweeps          1;
    }

    k
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance        1e-8;
        relTol          0.1;
        nSweeps          1;
    }

    epsilon
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance        1e-8;
        relTol          0.1;
        nSweeps          1;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 0;

    residualControl
    {
        p                1e-2;
        U                1e-3;
        "(k|epsilon)"    1e-3;
    }
}

relaxationFactors
{
    fields
    {
        p                0.3;
    }
    equations
    {
        U                0.7;
        k                0.7;
        epsilon          0.7;
    }
}

cache
{
    grad(U);
}

```

```
// ***** //
```

## initialConditions

```
/*----- C++ -----*\n=====\n\\ \\      F ield      | OpenFOAM: The Open Source CFD Toolbox\n\\ \\      O peration   | Website:  https://openfoam.org\n\\ \\      A nd         | Version:   dev\n\\ \\      M anipulation | \n*\n-----*/
```

```
flowVelocity      (0 0 0);\npressure          0;\nturbulentKE       1.3;\nturbulentEpsilon 0.01;
```

```
// ***** //
```

## k

```
/*----- C++ -----*\n=====\n\\ \\      F ield      | OpenFOAM: The Open Source CFD Toolbox\n\\ \\      O peration   | Website:  https://openfoam.org\n\\ \\      A nd         | Version:   dev\n\\ \\      M anipulation | \n*\n-----*/
```

```
FoamFile\n{\n  version      2.0;\n  format       ascii;\n  class        volScalarField;\n  object       k;\n}
```

```
// * * * * * *
```

```
#include "include/initialConditions"
```

```
dimensions [0 2 -2 0 0 0 0];
```

```
internalField uniform $turbulentKE;
```

```
boundaryField\n{\n  outlet\n  {\n    type          inletOutlet;\n    inletValue    uniform $turbulentKE;\n    value         $internalField;\n  }\n\n  inlet\n  {\n    type          atmBoundaryLayerInletK;\n    #include      "include/ABLConditions"\n  }\n\n  terrain\n  {\n    type          kqRWallFunction;\n    value         uniform 0.0;\n  }\n}
```



```

    ground
    {
        type          zeroGradient;
    }

#include "include/sideAndTopPatches"
}

// ***** //

snappyHexMeshDict

/*----- C++ -----*\
=====
\\ \\      F ield           | OpenFOAM: The Open Source CFD Toolbox
\\ \\      O peration       | Website:  https://openfoam.org
\\ \\      A nd              | Version:   dev
\\ \\      M anipulation     |
=====
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       snappyHexMeshDict;
}

// * * * * * //

// Which of the steps to run
castellatedMesh true;
snap             true;
addLayers        false;

// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting it
// - to specify refinement for any mesh cell inside/outside/near
// - to 'snap' the mesh boundary to the surface

geometry
{
/*   windTurbine1
    {
        type searchableBox;
        min (267 446.5 777);
        max (307 559.5 890);
    }

    windTurbine2
    {
        type searchableBox;
        min (851 434.5 741.5);
        max (891 547.5 854.5);
    }
*/
    terrain
    {
        type triSurfaceMesh;
        file "dtm1_8_1.stl";
    }
}

```

```

};

// Settings for the castellatedMesh generation.
castellatedMeshControls
{
    // Refinement parameters
    // ~~~~~

    // If local number of cells is >= maxLocalCells on any processor
    // switches from from refinement followed by balancing
    // (current method) to (weighted) balancing before refinement.
    maxLocalCells 200000;

    // Overall cell limit (approximately). Refinement will stop immediately
    // upon reaching this number so a refinement level might not complete.
    // Note that this is the number of cells before removing the part which
    // is not 'visible' from the keepPoint. The final number of cells might
    // actually be a lot less.
    maxGlobalCells 3000000;

    // The surface refinement loop might spend lots of iterations
    // refining just a few cells. This setting will cause refinement
    // to stop if <= minimumRefine are selected for refinement. Note:
    // it will at least do one iteration (unless the number of cells
    // to refine is 0)
    minRefinementCells 0;

    // Allow a certain level of imbalance during refining
    // (since balancing is quite expensive)
    // Expressed as fraction of perfect balance (= overall number of cells /
    // nProcs). 0=balance always.
    maxLoadUnbalance 0.10;

    // Number of buffer layers between different levels.
    // 1 means normal 2:1 refinement restriction , larger means slower
    // refinement.
    nCellsBetweenLevels 6;

    // Explicit feature edge refinement
    // ~~~~~

    // Specifies a level for any cell intersected by its edges.
    // This is a featureEdgeMesh, read from constant/triSurface for now.
    features
    (
        //{
        //    file "someLine.eMesh";
        //    level 2;
        //}
    );

    // Surface based refinement
    // ~~~~~

    // Specifies two levels for every surface. The first is the minimum level,

```

```

// every cell intersecting a surface gets refined up to the minimum level.
// The second level is the maximum level. Cells that 'see' multiple
// intersections where the intersections make an
// angle > resolveFeatureAngle get refined up to the maximum level.

refinementSurfaces
{
    terrain
    {
        // Surface-wise min and max refinement level
        level (2 2);

        // Optional region-wise level specification
/*
        regions
        {
            windTurbine
            {
                level (3 3);
            }
        }
*/
        //-- Optional angle to detect small-large cell situation
        // perpendicular to the surface. Is the angle of face w.r.t.
        // the local surface normal. Use on flat(ish) surfaces only.
        // Otherwise leave out or set to negative number.
        // perpendicularAngle 10;

        //-- Optional faceZone and (for closed surface) cellZone with
        // how to select the cells that are in the cellZone
        // (inside / outside / specified insidePoint)
        // faceZone sphere;
        // cellZone sphere;
        // cellZoneInside inside; // outside/insidePoint
    }
}

resolveFeatureAngle 30;

// Region-wise refinement
// ~~~~~

// Specifies refinement level for cells in relation to a surface. One of
// three modes
// - distance. 'levels' specifies per distance to the surface the
// wanted refinement level. The distances need to be specified in
// descending order.
// - inside. 'levels' is only one entry and only the level is used. All
// cells inside the surface get refined up to the level. The surface
// needs to be closed for this to be possible.
// - outside. Same but cells outside.

refinementRegions
{

    windTurbine1
    {
        mode inside;
        levels ((6 6));
    }
}

```

```

    windTurbine2
    {
        mode inside;
        levels ((6 6));
    }

    // sphere.stl
    //{
    //    mode distance;
    //    levels ((1.0 5) (2.0 3));
    //}
}

// Mesh selection
// ~~~~~

// After refinement patches get added for all refinementSurfaces and
// all cells intersecting the surfaces get put into these patches. The
// section reachable from the locationInMesh is kept.
// NOTE: This point should never be on a face, always inside a cell, even
// after refinement.
locationInMesh (1000 500 700);

// Whether any faceZones (as specified in the refinementSurfaces)
// are only on the boundary of corresponding cellZones or also allow
// free-standing zone faces. Not used if there are no faceZones.
allowFreeStandingZoneFaces true;
}

// Settings for the snapping.
snapControls
{
    //- Number of patch smoothing iterations before finding correspondence
    // to surface
    nSmoothPatch 3;

    //- Relative distance for points to be attracted by surface feature point
    // or edge. True distance is this factor times local
    // maximum edge length.
    tolerance 4.0;

    //- Number of mesh displacement relaxation iterations.
    nSolveIter 50;

    //- Maximum number of snapping relaxation iterations. Should stop
    // before upon reaching a correct mesh.
    nRelaxIter 5;
}

// Settings for the layer addition.
addLayersControls
{
    // Are the thickness parameters below relative to the undistorted
    // size of the refined cell outside layer (true) or absolute sizes (false).
    relativeSizes true;
}

```

```

// Per final patch (so not geometry!) the layer information
layers
{
}

// Expansion factor for layer mesh
expansionRatio 1.0;

// Wanted thickness of final added cell layer. If multiple layers
// is the thickness of the layer furthest away from the wall.
// Relative to undistorted size of cell outside layer.
// See relativeSizes parameter.
finalLayerThickness 0.3;

// Minimum thickness of cell layer. If for any reason layer
// cannot be above minThickness do not add layer.
// See relativeSizes parameter.
minThickness 0.25;

// If points get not extruded do nGrow layers of connected faces that are
// also not grown. This helps convergence of the layer addition process
// close to features.
// Note: changed(corrected) w.r.t 17x! (didn't do anything in 17x)
nGrow 0;

// Advanced settings

// When not to extrude surface. 0 is flat surface, 90 is when two faces
// are perpendicular
featureAngle 60;

// Maximum number of snapping relaxation iterations. Should stop
// before upon reaching a correct mesh.
nRelaxIter 5;

// Number of smoothing iterations of surface normals
nSmoothSurfaceNormals 1;

// Number of smoothing iterations of interior mesh movement direction
nSmoothNormals 3;

// Smooth layer thickness over surface patches
nSmoothThickness 10;

// Stop layer growth on highly warped cells
maxFaceThicknessRatio 0.5;

// Reduce layer growth where ratio thickness to medial
// distance is large
maxThicknessToMedialRatio 0.3;

// Angle used to pick up medial axis points
// Note: changed(corrected) w.r.t 16x! 90 degrees corresponds to 130 in 16x.
minMedianAxisAngle 90;

// Create buffer region for new layer terminations
nBufferCellsNoExtrude 0;

// Overall max number of layer addition iterations. The mesher will exit
// if it reaches this number of iterations; possibly with an illegal

```

```

// mesh.
nLayerIter 50;

// Max number of iterations after which relaxed meshQuality controls
// get used. Up to nRelaxIter it uses the settings in meshQualityControls,
// after nRelaxIter it uses the values in meshQualityControls::relaxed.
nRelaxedIter 20;
}

// Generic mesh quality settings. At any undoable phase these determine
// where to undo.
meshQualityControls
{
    //-- Maximum non-orthogonality allowed. Set to 180 to disable.
    maxNonOrtho 65;

    //-- Max skewness allowed. Set to <0 to disable.
    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    //-- Max concaveness allowed. Is angle (in degrees) below which concavity
    // is allowed. 0 is straight face, <0 would be convex face.
    // Set to 180 to disable.
    maxConcave 80;

    //-- Minimum pyramid volume. Is absolute volume of cell pyramid.
    // Set to a sensible fraction of the smallest cell volume expected.
    // Set to very negative number (e.g. -1E30) to disable.
    minVol 1e-13;

    //-- Minimum quality of the tet formed by the face-centre
    // and variable base point minimum decomposition triangles and
    // the cell centre. Set to very negative number (e.g. -1E30) to
    // disable.
    // <0 = inside out tet,
    // 0 = flat tet
    // 1 = regular tet
    minTetQuality 1e-30;

    //-- Minimum face area. Set to <0 to disable.
    minArea -1;

    //-- Minimum face twist. Set to <-1 to disable. dot product of face normal
    // and face centre triangles normal
    minTwist 0.05;

    //-- Minimum normalised cell determinant
    // 1 = hex, <= 0 = folded or flattened illegal cell
    minDeterminant 0.001;

    //-- minFaceWeight (0 -> 0.5)
    minFaceWeight 0.05;

    //-- minVolRatio (0 -> 1)
    minVolRatio 0.01;

    // must be >0 for Fluent compatibility
    minTriangleTwist -1;

    //-- If >0 : preserve single cells with all points on the surface if the
    // resulting volume after snapping (by approximation) is larger than

```

```

// minVolCollapseRatio times old volume (i.e. not collapsed to flat cell).
// If <0 : delete always.
// minVolCollapseRatio 0.5;

// Advanced

//-- Number of error distribution iterations
nSmoothScale 4;
//-- Amount to scale back displacement at error points
errorReduction 0.75;

// Optional : some meshing phases allow usage of relaxed rules.
// See e.g. addLayersControls::nRelaxedIter.
relaxed
{
    //-- Maximum non-orthogonality allowed. Set to 180 to disable.
    maxNonOrtho 75;
}

// Advanced

// Merge tolerance. Is fraction of overall bounding box of initial mesh.
// Note: the write tolerance needs to be higher than this.
mergeTolerance 1e-6;

// ***** //

topoSetDict
/*----- C++ -----*\
  \\\  F ield      | OpenFOAM: The Open Source CFD Toolbox
  \\\  O peration  | Website: https://openfoam.org
  \\\  A nd        | Version: dev
  \\\  M anipulation |
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       topoSetDict;
}
// ***** //
actions
(
    // actuationDisk1
    {
        name      actuationDisk1CellSet;
        type      cellSet;
        action    new;
        source    boxToCell;
        sourceInfo
        {
            box (287 446.5 777) (288 559.5 890);
        }
    }
)
{
}

```





```
{
  Cmu          0.09;
  C1           1.44;
  C2           1.92;
  sigmaEps     1.11; // Original value:1.44
  // See:
  // D.M. Hargreaves and N.G. Wright
  // "On the use of the k-Epsilon model in commercial CFD software
  // to model the neutral atmospheric boundary layer",
  // J. of wind engineering and industrial aerodynamics,
  // 95(2007) 355-269
}
}

// ***** //
```

## B. FAST v8 input files

The following Appendix contain text input files from FAST v8 calculations. The following text files are examples for one simulation.

### Turbsim inputfile

```
1 TurbSim Input File. Valid for TurbSim v1.06.00, 21-Sep-2012
2
3 -----Runtime Options-----
4 5446546      RandSeed1      - First random seed (-2147483648 to 2147483647)
5 RANLUX      RandSeed2      - Second random seed (-2147483648 to 2147483647) for intrinsic pRNG, or an alternative
   pRNG: "RanLux" or "RNSNLW"
6 False      WrBHHTP      - Output hub-height turbulence parameters in binary form? (Generates RootName.bin)
7 False      WrFHHTP      - Output hub-height turbulence parameters in formatted form? (Generates RootName.dat)
8 False      WrADHH      - Output hub-height time-series data in AeroDyn form? (Generates RootName.hh)
9 True       WrADFF      - Output full-field time-series data in TurbSim/AeroDyn form? (Generates Rootname.bts)
10 False     WrBLFF      - Output full-field time-series data in BLADED/AeroDyn form? (Generates RootName.wnd)
11 True      WrADIWR      - Output tower time-series data? (Generates RootName.twr)
12 False     WrFMTFF      - Output full-field time-series data in formatted (readable) form? (Generates RootName.u
   , RootName.v, RootName.w)
13 False     WrACT      - Output coherent turbulence time steps in AeroDyn form? (Generates RootName.cts)
14 True      Clockwise    - Clockwise rotation looking downwind? (used only for full-field binary files - not
   necessary for AeroDyn)
15 0         ScaleIEC     - Scale IEC turbulence models to exact target standard deviation? [0=no additional
   scaling; 1=use hub scale uniformly; 2=use individual scales]
16
17 -----Turbine/Model Specifications-----
18 13        NumGrid_Z     - Vertical grid-point matrix dimension
19 13        NumGrid_Y     - Horizontal grid-point matrix dimension
20 0.05     TimeStep      - Time step [seconds]
21 3600     AnalysisTime  - Length of analysis time series [seconds] (program will add time if necessary:
   AnalysisTime = MAX(AnalysisTime, UsableTime+GridWidth/MeanHHWS) )
22 3600     UsableTime    - Usable length of output time series [seconds] (program will add GridWidth/MeanHHWS
   seconds)
23 108      HubHt         - Hub height [m] (should be > 0.5*GridHeight)
24 160      GridHeight    - Grid height [m]
25 160      GridWidth     - Grid width [m] (should be >= 2*(RotorRadius+ShaftLength))
26 0        VFlowAng     - Vertical mean flow (uptilt) angle [degrees]
27 0        HFlowAng     - Horizontal mean flow (skew) angle [degrees]
28
29 -----Meteorological Boundary Conditions-----
```

30 "IECKAI" TurbModel - Turbulence model ("IECKAI"=Kaimal, "IECVKM"=von Karman, "GP\_LLJ", "NWTICUP", "SMOOTH", "WFUPW", "WF\_07D", "WF\_14D", "TIDAL", or "NONE")

31 "1-ED3" IECstandard - Number of IEC 61400-x standard (x=1,2, or 3 with optional 61400-1 edition number (i.e. "1-Ed2") )

32 "3" IECturbc - IEC turbulence characteristic ("A", "B", "C" or the turbulence intensity in percent) ("KHTEST" option with NWTICUP model, not used for other models)

33 "NIM" IEC\_WindType - IEC turbulence type ("NIM"=normal, "xETM"=extreme turbulence, "xEWMI"=extreme 1-year wind, "xEWM50"=extreme 50-year wind, where x=wind turbine class 1, 2, or 3)

34 default ETMc - IEC Extreme Turbulence Model "c" parameter [m/s]

35 default WindProfileType - Wind profile type ("JET";"LOG"=logarithmic;"PL"=power law;"H2L"=Log law for TIDAL spectral model;"IEC"=PL on rotor disk, LOG elsewhere; or "default")

36 108 RefHt - Height of the reference wind speed [m]

37 10 URef - Mean (total) wind speed at the reference height [m/s] (or "default" for JET wind profile)

38 default ZJetMax - Jet height [m] (used only for JET wind profile, valid 70-490 m)

39 default PLExp - Power law exponent [-] (or "default")

40 0.3 Z0 - Surface roughness length [m] (or "default")

41

42 -----Non-IEC Meteorological Boundary Conditions-----

43 default Latitude - Site latitude [degrees] (or "default")

44 0.05 RICH\_NO - Gradient Richardson number

45 default UStar - Friction or shear velocity [m/s] (or "default")

46 default ZI - Mixing layer depth [m] (or "default")

47 default PC\_UW - Hub mean u'w' Reynolds stress (or "default")

48 default PC\_UV - Hub mean u'v' Reynolds stress (or "default")

49 default PC\_VW - Hub mean v'w' Reynolds stress (or "default")

50 default IncDec1 - u-component coherence parameters (e.g. "10.0 0.3e-3" in quotes) (or "default")

51 default IncDec2 - v-component coherence parameters (e.g. "10.0 0.3e-3" in quotes) (or "default")

52 default IncDec3 - w-component coherence parameters (e.g. "10.0 0.3e-3" in quotes) (or "default")

53 default CohExp - Coherence exponent (or "default")

54

55 -----Coherent Turbulence Scaling Parameters-----

56 "C:\Aerosim\Software\turbSim\Eventpath\eventdata" CTEventPath - Name of the path where event data files are located

57 "Random" CTEventFile - Type of event files ("LES", "DNS", or "RANDOM")

58 true Randomize - Randomize the disturbance scale and locations? (true/false)

59 1.0 DistScl - Disturbance scale (ratio of wave height to rotor disk). (Ignored when Randomize = true)

60 0.5 CTLy - Fractional location of tower centerline from right (looking downwind) to left side of the dataset. (Ignored when Randomize = true.)

61 0.5 CTLz - Fractional location of hub height from the bottom of the dataset. (Ignored when Randomize = true.)

```
62 30.0          CTStartTime    - Minimum start time for coherent structures in RootName.cts [seconds]
63
64 =====
65 NOTE: Do not add or remove any lines in this file!
66 =====
```

## InflowWind inputfile

```
1  _____ InflowWind v3.01.* INPUT FILE _____
2  12 m/s turbulent winds on 31x31 FF grid and tower for FAST CertTests #18, #19, #21, #22, #23, and #24
3  _____
4  False      Echo          - Echo input data to <RootName>.ech (flag)
5           3  WindType      - switch for wind file type (1=steady; 2=uniform; 3=binary TurbSim FF; 4=binary Bladed-style FF;
6           0  PropagationDir - Direction of wind propagation (meteorological rotation from aligned with X (positive rotates
           1  NWindVel      - Number of points to output the wind velocity      (0 to 9)
           0  WindVxiList   - List of coordinates in the inertial X direction (m)
           0  WindVyiList   - List of coordinates in the inertial Y direction (m)
10          110.0000 WindVzList - List of coordinates in the inertial Z direction (m)
11  ===== Parameters for Steady Wind Conditions [used only for WindType = 1] =====
12          9  HWindSpeed    - Horizontal windspeed                          (m/s)
13          110.0000 RefHt    - Reference height for horizontal wind speed      (m)
14          0.    PLexp      - Power law exponent                          (-)
15  ===== Parameters for Uniform wind file [used only for WindType = 2] =====
16  "Wind/90m_12mps_twr.bts"  Filename      - Filename of time series data for uniform wind field.      (-)
17          110.0000 RefHt    - Reference height for horizontal wind speed      (m)
18          110.0000 RefLength - Reference length for linear horizontal and vertical sheer (-)
19  ===== Parameters for Binary TurbSim Full-Field files [used only for WindType = 3] =====
20  "Wind/turbSim.bts"       Filename      - Name of the Full field wind file to use (.bts)
21  ===== Parameters for Binary Bladed-style Full-Field files [used only for WindType = 4] =====
22  "Wind/90m_12mps_twr"     FilenameRoot  - Rootname of the full-field wind file to use (.wnd, .sum)
23  False      TowerFile    - Have tower file (.twr) (flag)
24  ===== Parameters for HAWC-format binary files [Only used with WindType = 5] =====
25  "wasp\Output\basic_5u.bin"  FileName_u   - name of the file containing the u-component fluctuating wind (.bin)
26  "wasp\Output\basic_5v.bin"  FileName_v   - name of the file containing the v-component fluctuating wind (.bin)
27  "wasp\Output\basic_5w.bin"  FileName_w   - name of the file containing the w-component fluctuating wind (.bin)
28          64  nx          - number of grids in the x direction (in the 3 files above) (-)
29          32  ny          - number of grids in the y direction (in the 3 files above) (-)
30          32  nz          - number of grids in the z direction (in the 3 files above) (-)
31          16  dx          - distance (in meters) between points in the x direction      (m)
32          3   dy          - distance (in meters) between points in the y direction      (m)
33          3   dz          - distance (in meters) between points in the z direction      (m)
34          110.0000 RefHt    - reference height; the height (in meters) of the vertical center of the grid (m)
35  _____ Scaling parameters for turbulence _____
36          1   ScaleMethod  - Turbulence scaling method [0 = none, 1 = direct scaling, 2 = calculate scaling factor based
           on a desired standard deviation]
```

```

37      1   SFx      - Turbulence scaling factor for the x direction (-) [ScaleMethod=1]
38      1   SFy      - Turbulence scaling factor for the y direction (-) [ScaleMethod=1]
39      1   SFz      - Turbulence scaling factor for the z direction (-) [ScaleMethod=1]
40      12  SigmaFx   - Turbulence standard deviation to calculate scaling from in x direction (m/s) [ScaleMethod
      =2]
41      8   SigmaFy   - Turbulence standard deviation to calculate scaling from in y direction (m/s) [ScaleMethod
      =2]
42      2   SigmaFz   - Turbulence standard deviation to calculate scaling from in z direction (m/s) [ScaleMethod
      =2]
43  ----- Mean wind profile parameters (added to HAWC-format files) -----
44      5   URef      - Mean u-component wind speed at the reference height (m/s)
45      2   WindProfile - Wind profile type (0=constant;1=logarithmic,2=power law)
46      0.2 PLExp     - Power law exponent (-) (used for PL wind profile type only)
47      0.03 Z0      - Surface roughness length (m) (used for LG wind profile type only)
48  ===== OUTPUT =====
49  False      SumPrint - Print summary data to <RootName>.IfW.sum (flag)
50              OutList - The next line(s) contains a list of output parameters. See OutListParameters.xlsx for a listing
                    of available output channels, (-)
51  "Wind1VelX" X-direction wind velocity at point WindList(1)
52  "Wind1VelY" Y-direction wind velocity at point WindList(1)
53  "Wind1VelZ" Z-direction wind velocity at point WindList(1)
54  END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
55  -----

```

## ElastoDyn inputfile

```

1  _____ ELASTODYN v1.03.* INPUT FILE _____
2  Elasto dyn main input file
3  _____ SIMULATION CONTROL _____
4  False          Echo          - Echo input data to "<RootName>.ech" (flag)
5              3  Method        - Integration method: {1: RK4, 2: AB4, or 3: ABM4} (-)
6  "DEFAULT"      DT            - Integration time step (s)
7  _____ ENVIRONMENTAL CONDITION _____
8              9.80665  Gravity    - Gravitational acceleration (m/s^2)
9  _____ DEGREES OF FREEDOM _____
10 True           FlapDOF1      - First flapwise blade mode DOF (flag)
11 True           FlapDOF2      - Second flapwise blade mode DOF (flag)
12 True           EdgeDOF       - First edgewise blade mode DOF (flag)
13 False          TeetDOF       - Rotor-teeter DOF (flag) [unused for 3 blades]
14 True           DrTrDOF       - Drivetrain rotational-flexibility DOF (flag)
15 True           GenDOF        - Generator DOF (flag)
16 True           YawDOF        - Yaw DOF (flag)
17 True           TwFADOF1      - First fore-aft tower bending-mode DOF (flag)
18 True           TwFADOF2      - Second fore-aft tower bending-mode DOF (flag)
19 True           TwSSDOF1      - First side-to-side tower bending-mode DOF (flag)
20 True           TwSSDOF2      - Second side-to-side tower bending-mode DOF (flag)
21 False          PtfmSgDOF     - Platform horizontal surge translation DOF (flag)
22 False          PtfmSwDOF     - Platform horizontal sway translation DOF (flag)
23 False          PtfmHvDOF     - Platform vertical heave translation DOF (flag)
24 False          PtfmRDOF      - Platform roll tilt rotation DOF (flag)
25 False          PtfmPDOF      - Platform pitch tilt rotation DOF (flag)
26 False          PtfmYDOF      - Platform yaw rotation DOF (flag)
27  _____ INITIAL CONDITIONS _____
28              0  OoPDefl      - Initial out-of-plane blade-tip displacement (meters)
29              0  IPDefl       - Initial in-plane blade-tip deflection (meters)
30              0.8696  BLPitch(1) - Blade 1 initial pitch (degrees)
31              0.8696  BLPitch(2) - Blade 2 initial pitch (degrees)
32              0.8696  BLPitch(3) - Blade 3 initial pitch (degrees) [unused for 2 blades]
33              0  TeetDefl     - Initial or fixed teeter angle (degrees) [unused for 3 blades]
34              0  Azimuth      - Initial azimuth angle for blade 1 (degrees)
35              12.3  RotSpeed   - Initial or fixed rotor speed (rpm)
36              0  NacYaw       - Initial or fixed nacelle-yaw angle (degrees)
37              0  TTDspFA     - Initial fore-aft tower-top displacement (meters)
38              0  TTDspSS     - Initial side-to-side tower-top displacement (meters)
39              0  PtfmSurge    - Initial or fixed horizontal surge translational displacement of platform (meters)

```

22

40	0	PtfmSway	- Initial or fixed horizontal sway translational displacement of platform (meters)
41	0	PtfmHeave	- Initial or fixed vertical heave translational displacement of platform (meters)
42	0	PtfmRoll	- Initial or fixed roll tilt rotational displacement of platform (degrees)
43	0	PtfmPitch	- Initial or fixed pitch tilt rotational displacement of platform (degrees)
44	0	PtfmYaw	- Initial or fixed yaw rotational displacement of platform (degrees)
45	<hr/> TURBINE CONFIGURATION <hr/>		
46	3	NumBl	- Number of blades (-)
47	65.0000	TipRad	- The distance from the rotor apex to the blade tip (meters)
48	2.0000	HubRad	- The distance from the rotor apex to the blade root (meters)
49	-3.0000	PreCone(1)	- Blade 1 cone angle (degrees)
50	-3.0000	PreCone(2)	- Blade 2 cone angle (degrees)
51	-3.0000	PreCone(3)	- Blade 3 cone angle (degrees) [unused for 2 blades]
52	0	HubCM	- Distance from rotor apex to hub mass [positive downwind] (meters)
53	0	UndSling	- Undersling length [distance from teeter pin to the rotor apex] (meters) [unused for 3 blades]
54	0	Delta3	- Delta-3 angle for teetering rotors (degrees) [unused for 3 blades]
55	0	AzimB1Up	- Azimuth value to use for I/O when blade 1 points up (degrees)
56	-5.0000	OverHang	- Distance from yaw axis to rotor apex [3 blades] or teeter pin [2 blades] (meters)
57	2.5000	ShftGagL	- Distance from rotor apex [3 blades] or teeter pin [2 blades] to shaft strain gages [positive for upwind rotors] (meters)
58	-5.0000	ShftTilt	- Rotor shaft tilt angle (degrees)
59	0.0995	NacCMxn	- Downwind distance from the tower-top to the nacelle CM (meters)
60	0	NacCMyn	- Lateral distance from the tower-top to the nacelle CM (meters)
61	0.5843	NacCMzn	- Vertical distance from the tower-top to the nacelle CM (meters)
62	0.0995	NcIMUxn	- Downwind distance from the tower-top to the nacelle IMU (meters)
63	0	NcIMUyn	- Lateral distance from the tower-top to the nacelle IMU (meters)
64	0.5843	NcIMUzn	- Vertical distance from the tower-top to the nacelle IMU (meters)
65	0.0	Twr2Shft	- Vertical distance from the tower-top to the rotor shaft (meters)
66	108.0000	TowerHt	- Height of tower above ground level [onshore] or MSL [offshore] (meters)
67	0	TowerBsHt	- Height of tower base above ground level [onshore] or MSL [offshore] (meters)
68	0	PtfmCMxt	- Downwind distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
69	0	PtfmCMyt	- Lateral distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
70	0	PtfmCMzt	- Vertical distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
71	0	PtfmRefzt	- Vertical distance from the ground level [onshore] or MSL [offshore] to the platform reference point (meters)
72	<hr/> MASS AND INERTIA <hr/>		
73	0	TipMass(1)	- Tip-brake mass, blade 1 (kg)
74	0	TipMass(2)	- Tip-brake mass, blade 2 (kg)
75	0	TipMass(3)	- Tip-brake mass, blade 3 (kg) [unused for 2 blades]
76	27975.00	HubMass	- Hub mass (kg)
77	60131.00	HubIner	- Hub inertia about rotor axis [3 blades] or teeter axis [2 blades] (kg m <sup>2</sup> )
78	500.0	GenIner	- Generator inertia about HSS (kg m <sup>2</sup> )



79 190612.00 NacMass - Nacelle mass (kg)  
80 1210200.00 NacYIner - Nacelle inertia about yaw axis (kg m<sup>2</sup>)  
81 0 YawBrMass - Yaw bearing mass (kg)  
82 0 PtfmMass - Platform mass (kg)  
83 0 PtfmRIner - Platform inertia for roll tilt rotation about the platform CM (kg m<sup>2</sup>)  
84 0 PtfmPIner - Platform inertia for pitch tilt rotation about the platform CM (kg m<sup>2</sup>)  
85 0 PtfmYIner - Platform inertia for yaw rotation about the platform CM (kg m<sup>2</sup>)

---

86 BLADE

87 20 BldNodes - Number of blade nodes (per blade) used for analysis (-)  
88 "ElastoDyn\_Blade.dat" BldFile(1) - Name of file containing properties for blade 1 (quoted string)  
89 "ElastoDyn\_Blade.dat" BldFile(2) - Name of file containing properties for blade 2 (quoted string)  
90 "ElastoDyn\_Blade.dat" BldFile(3) - Name of file containing properties for blade 3 (quoted string) [unused for 2 blades]

---

91 ROTOR-TEETER

92 0 TeetMod - Rotor-teeter spring/damper model {0: none, 1: standard, 2: user-defined from routine UserTeet} (switch) [unused for 3 blades]  
93 0 TeetDmpP - Rotor-teeter damper position (degrees) [used only for 2 blades and when TeetMod=1]  
94 0 TeetDmp - Rotor-teeter damping constant (N-m/(rad/s)) [used only for 2 blades and when TeetMod=1]  
95 0 TeetCDmp - Rotor-teeter rate-independent Coulomb-damping moment (N-m) [used only for 2 blades and when TeetMod=1]  
96 0 TeetSStP - Rotor-teeter soft-stop position (degrees) [used only for 2 blades and when TeetMod=1]  
97 0 TeetHStP - Rotor-teeter hard-stop position (degrees) [used only for 2 blades and when TeetMod=1]  
98 0 TeetSSSp - Rotor-teeter soft-stop linear-spring constant (N-m/rad) [used only for 2 blades and when TeetMod=1]  
99 0 TeetHSSp - Rotor-teeter hard-stop linear-spring constant (N-m/rad) [used only for 2 blades and when TeetMod=1]

---

100 DRIVETRAIN

101 95.5000 GBoxEff - Gearbox efficiency (%)  
102 97.0000 GBRatio - Gearbox ratio (-)  
103 8.67637E+08 DTTorSpr - Drivetrain torsional spring (N-m/rad)  
104 6.215E+06 DTTorDmp - Drivetrain torsional damper (N-m/(rad/s))

---

105 FURLING

106 False Furling - Read in additional model properties for furling turbine (flag) [must currently be FALSE]  
107 "unused" FurlFile - Name of file containing furling properties (quoted string) [unused when Furling=False]

---

108 TOWER

109 20 TwrNodes - Number of tower nodes used for analysis (-)  
110 "ElastoDyn\_Tower.dat" TwrFile - Name of file containing tower properties (quoted string)

---

111 OUTPUT

112 True SumPrint - Print summary data to "<RootName>.sum" (flag)  
113 1 OutFile - Switch to determine where output will be placed: {1: in module output file only; 2: in glue code output file only; 3: both} (currently unused)  
114 True TabDelim - Use tab delimiters in text tabular output file? (flag) (currently unused)

115 "ES10.3E2" OutFmt - Format used for text tabular output (except time). Resulting field should be 10 characters. (quoted string) (currently unused)

116 0 TStart - Time to begin tabular output (s) (currently unused)

117 1 DecFact - Decimation factor for tabular output {1: output every time step} (-) (currently unused)

118 0 NTwGages - Number of tower nodes that have strain gages for output [0 to 9] (-)

119 10, 19, 28 TwrGagNd - List of tower nodes that have strain gages [1 to TwrNodes] (-) [unused  
if NTwGages=0]

120 3 NBIGages - Number of blade nodes that have strain gages for output [0 to 9] (-)

121 5, 9, 13 BldGagNd - List of blade nodes that have strain gages [1 to BldNodes] (-) [unused  
if NBIGages=0]

122 OutList - The next line(s) contains a list of output parameters. See OutListParameters.xlsx for a listing of available output channels, (-)

123 "BldPitch1" - Blade 1 pitch angle

124 "Azimuth" - Blade 1 azimuth angle

125 "RotSpeed" - Low-speed shaft and high-speed shaft speeds

126 "RotTorq" - Rotor torque and low-speed shaft 0- and 90-bending moments at the main bearing

127 "RotThrust"

128 "OoPDefl1" - Blade 1 out-of-plane and in-plane deflections and tip twist

129 "IPDefl1" - Blade 1 out-of-plane and in-plane deflections and tip twist

130 "TwstDefl1" - Blade 1 out-of-plane and in-plane deflections and tip twist

131 "TipClrc1"

132 "TipDxb1"

133 "TipDyb1"

134 "TipDzb1"

135 "NcIMUTAxS" - Nacelle IMU translational accelerations (absolute) in the nonrotating, shaft coordinate system

136 "NcIMUTAyS" - Nacelle IMU translational accelerations (absolute) in the nonrotating, shaft coordinate system

137 "NcIMUTAzS" - Nacelle IMU translational accelerations (absolute) in the nonrotating, shaft coordinate system

138 "PtfmSurge" - Platform translational surge, sway, and heave displacements

139 "PtfmSway" - Platform translational surge, sway, and heave displacements

140 "PtfmHeave" - Platform translational surge, sway, and heave displacements

141 "PtfmRoll" - Platform rotational roll, pitch and yaw displacements

142 "PtfmPitch" - Platform rotational roll, pitch and yaw displacements

143 "PtfmYaw" - Platform rotational roll, pitch and yaw displacements

144 "NacYaw" - Nacelle yaw angle and nacelle yaw error estimate

145 "RootMxb1" - In-plane bending, out-of-plane bending, and pitching moments at the root of blade 1

146 "RootMyb1" - In-plane bending, out-of-plane bending, and pitching moments at the root of blade 1

147 "RootMzb1" - In-plane bending, out-of-plane bending, and pitching moments at the root of blade 1

148 "YawBrMxp"

149 "YawBrMyp"

150 "YawBrMzp"

151 "YawBrFxp"

152 "YawBrFyp"  
153 "YawBrFzp"  
154 END of input file (the word "END" must appear in the first 3 columns of this last OutList line)  
155 \_\_\_\_\_

## Servodyn Inputfile

```
1  _____ SERVODYN v1.05.* INPUT FILE _____
2  Servodyn main input file
3  _____ SIMULATION CONTROL _____
4  False      Echo      - Echo input data to <RootName>.ech (flag)
5  "default"  DT        - Communication interval for controllers (s) (or "default")
6  _____ PITCH CONTROL _____
7          5  PCMode    - Pitch control mode {0: none, 3: user-defined from routine PitchCntrl, 4: user-defined from
          Simulink/Labview, 5: user-defined from Bladed-style DLL} (switch)
8          0  TPCOn     - Time to enable active pitch control (s) [unused when PCMode=0]
9          9999.9 TPitManS(1) - Time to start override pitch maneuver for blade 1 and end standard pitch control (s)
10         9999.9 TPitManS(2) - Time to start override pitch maneuver for blade 2 and end standard pitch control (s)
11         9999.9 TPitManS(3) - Time to start override pitch maneuver for blade 3 and end standard pitch control (s) [unused for
          2 blades]
12         7.0000 PitManRat(1) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 1 (deg/s)
13         7.0000 PitManRat(2) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 2 (deg/s)
14         7.0000 PitManRat(3) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 3 (deg/s)
          [unused for 2 blades]
15         0  BIPitchF(1) - Blade 1 final pitch for pitch maneuvers (degrees)
16         0  BIPitchF(2) - Blade 2 final pitch for pitch maneuvers (degrees)
17         0  BIPitchF(3) - Blade 3 final pitch for pitch maneuvers (degrees) [unused for 2 blades]
18  _____ GENERATOR AND TORQUE CONTROL _____
19         5  VSContrl  - Variable-speed control mode {0: none, 1: simple VS, 3: user-defined from routine UserVSCont, 4:
          user-defined from Simulink/Labview, 5: user-defined from Bladed-style DLL} (switch)
20         2  GenModel  - Generator model {1: simple, 2: Thevenin, 3: user-defined from routine UserGen} (switch) [used
          only when VSContrl=0]
21         98.0000 GenEff  - Generator efficiency [ignored by the Thevenin and user-defined generator models] (%)
22 True      GenTiStr  - Method to start the generator {T: timed using TimGenOn, F: generator speed using SpdGenOn} (flag
          )
23 True      GenTiStp  - Method to stop the generator {T: timed using TimGenOf, F: when generator power = 0} (flag)
24         9999.9 SpdGenOn - Generator speed to turn on the generator for a startup (HSS speed) (rpm) [used only when
          GenTiStr=False]
25         0  TimGenOn  - Time to turn on the generator for a startup (s) [used only when GenTiStr=True]
26         9999.9 TimGenOf - Time to turn off the generator (s) [used only when GenTiStp=True]
27  _____ SIMPLE VARIABLE-SPEED TORQUE CONTROL _____
28         9999.9 VS_RtGnSp - Rated generator speed for simple variable-speed generator control (HSS side) (rpm) [used only
          when VSContrl=1]
29         9999.9 VS_RtTq  - Rated generator torque/constant generator torque in Region 3 for simple variable-speed generator
          control (HSS side) (N-m) [used only when VSContrl=1]
30         9999.9 VS_Rgn2K  - Generator torque constant in Region 2 for simple variable-speed generator control (HSS side) (N-
```

m/rpm<sup>2</sup>) [used only when VSContrl=1]  
 31 9999.9 VS\_SlPc - Rated generator slip percentage in Region 2 1/2 for simple variable-speed generator control (%)  
 [used only when VSContrl=1]

---

32 SIMPLE INDUCTION GENERATOR

33 9999.9 SIG\_SlPc - Rated generator slip percentage (%) [used only when VSContrl=0 and GenModel=1]  
 34 9999.9 SIG\_SySp - Synchronous (zero-torque) generator speed (rpm) [used only when VSContrl=0 and GenModel=1]  
 35 9999.9 SIG\_RtTq - Rated torque (N-m) [used only when VSContrl=0 and GenModel=1]  
 36 9999.9 SIG\_PORT - Pull-out ratio (Tpullout/Trated) (-) [used only when VSContrl=0 and GenModel=1]

---

37 THEVENIN-EQUIVALENT INDUCTION GENERATOR

38 9999.9 TEC\_Freq - Line frequency [50 or 60] (Hz) [used only when VSContrl=0 and GenModel=2]  
 39 9998 TEC\_NPol - Number of poles [even integer > 0] (-) [used only when VSContrl=0 and GenModel=2]  
 40 9999.9 TEC\_SRes - Stator resistance (ohms) [used only when VSContrl=0 and GenModel=2]  
 41 9999.9 TEC\_RRes - Rotor resistance (ohms) [used only when VSContrl=0 and GenModel=2]  
 42 9999.9 TEC\_VLL - Line-to-line RMS voltage (volts) [used only when VSContrl=0 and GenModel=2]  
 43 9999.9 TEC\_SLR - Stator leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]  
 44 9999.9 TEC\_RLR - Rotor leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]  
 45 9999.9 TEC\_MR - Magnetizing reactance (ohms) [used only when VSContrl=0 and GenModel=2]

---

46 HIGH-SPEED SHAFT BRAKE

47 0 HSSBrMode - HSS brake model {0: none, 1: simple, 3: user-defined from routine UserHSSBr, 4: user-defined  
 from Simulink/Labview, 5: user-defined from Bladed-style DLL} (switch)

48 9999.9 THSSBrDp - Time to initiate deployment of the HSS brake (s)  
 49 0.6 HSSBrDT - Time for HSS-brake to reach full deployment once initiated (sec) [used only when HSSBrMode=1]  
 50 52253.5 HSSBrTqF - Fully deployed HSS-brake torque (N-m)

---

51 NACELLE-YAW CONTROL

52 0 YCMode - Yaw control mode {0: none, 3: user-defined from routine UserYawCont, 4: user-defined from  
 Simulink/Labview, 5: user-defined from Bladed-style DLL} (switch)

53 0 TYCon - Time to enable active yaw control (s) [unused when YCMode=0]  
 54 0 YawNeut - Neutral yaw position—yaw spring force is zero at this yaw (degrees)  
 55 9.02832E+09 YawSpr - Nacelle-yaw spring constant (N-m/rad)  
 56 1.916E+07 YawDamp - Nacelle-yaw damping constant (N-m/(rad/s))  
 57 9999.9 TYawManS - Time to start override yaw maneuver and end standard yaw control (s)  
 58 0.2500 YawManRat - Yaw maneuver rate (in absolute value) (deg/s)  
 59 0 NacYawF - Final yaw angle for override yaw maneuvers (degrees)

---

60 TUNED MASS DAMPER

61 False CompNTMD - Compute nacelle tuned mass damper {true/false} (flag)  
 62 "" NTMDfile - Name of the file for nacelle tuned mass damper (quoted string) [unused when CompNTMD is false]  
 63 False CompTTMD - Compute tower tuned mass damper {true/false} (flag)  
 64 "" TTMDfile - Name of the file for tower tuned mass damper (quoted string) [unused when CompTTMD is false]

---

65 BLADED INTERFACE [used only with Bladed Interface]

66 "FBv1\_discon.dll" - DLL\_FileName - Name/location of the dynamic library {.dll [Windows] or .so [Linux]} in the Bladed-DLL  
 format (-) [used only with Bladed Interface]

```

67 "FBv1_discon.in"  DLL_InFile  - Name of input file sent to the DLL (-) [used only with Bladed Interface]
68 "DISCON"         DLL_ProcName - Name of procedure in DLL to be called (-) [case sensitive; used only with DLL Interface]
69 "default"        DLL_DT       - Communication interval for dynamic library (s) (or "default") [used only with Bladed Interface]
70 false           DLL_Ramp      - Whether a linear ramp should be used between DLLDT time steps [introduces time shift when true]
      (flag) [used only with Bladed Interface]
71 9999.9          BPCutoff      - Cutoff frequency for low-pass filter on blade pitch from DLL (Hz) [used only with Bladed
      Interface]
72 0              NacYaw_North - Reference yaw angle of the nacelle when the upwind end points due North (deg) [used only with
      Bladed Interface]
73 0              Ptch_Cntrl  - Record 28: Use individual pitch control {0: collective pitch; 1: individual pitch control} (
      switch) [used only with Bladed Interface]
74 0              Ptch_SetPnt - Record 5: Below-rated pitch angle set-point (deg) [used only with Bladed Interface]
75 0              Ptch_Min    - Record 6: Minimum pitch angle (deg) [used only with Bladed Interface]
76 0              Ptch_Max    - Record 7: Maximum pitch angle (deg) [used only with Bladed Interface]
77 0              PtchRate_Min - Record 8: Minimum pitch rate (most negative value allowed) (deg/s) [used only with Bladed
      Interface]
78 0              PtchRate_Max - Record 9: Maximum pitch rate (deg/s) [used only with Bladed Interface]
79 0              Gain_OM     - Record 16: Optimal mode gain (Nm/(rad/s)^2) [used only with Bladed Interface]
80 0              GenSpd_MinOM - Record 17: Minimum generator speed (rpm) [used only with Bladed Interface]
81 0              GenSpd_MaxOM - Record 18: Optimal mode maximum speed (rpm) [used only with Bladed Interface]
82 0              GenSpd_Dem   - Record 19: Demanded generator speed above rated (rpm) [used only with Bladed Interface]
83 0              GenTrq_Dem   - Record 22: Demanded generator torque above rated (Nm) [used only with Bladed Interface]
84 0              GenPwr_Dem   - Record 13: Demanded power (W) [used only with Bladed Interface]
85 ----- BLADED INTERFACE TORQUE-SPEED LOOK-UP TABLE -----
86 0              DLL_NumTrq   - Record 26: No. of points in torque-speed look-up table {0 = none and use the optimal mode
      parameters; nonzero = ignore the optimal mode PARAMETERS by setting Record 16 to 0.0} (-) [used only with
      Bladed Interface]
87 GenSpd_TLU     GenTrq_TLU
88 (rpm)          (Nm)
89 ----- OUTPUT -----
90 True          SumPrint     - Print summary data to <RootName>.sum (flag) (currently unused)
91 1             OutFile      - Switch to determine where output will be placed: {1: in module output file only; 2: in glue code
      output file only; 3: both} (currently unused)
92 True          TabDelim     - Use tab delimiters in text tabular output file? (flag) (currently unused)
93 "ES10.3E2"    OutFmt       - Format used for text tabular output (except time). Resulting field should be 10 characters. (
      quoted string) (currently unused)
94 0             TStart       - Time to begin tabular output (s) (currently unused)
95              OutList      - The next line(s) contains a list of output parameters. See OutListParameters.xlsx for a listing
      of available output channels, (-)
96 "GenPwr"      - Electrical generator power and torque
97 "GenTq"

```

98 "YawMom" - Electrical generator power and torque  
99 "NTMD\_XQ" - Nacelle X TMD position (displacement)  
100 "NTMD\_XQD" - Nacelle X TMD velocity  
101 "NTMD\_YQ" - Nacelle Y TMD position (displacement)  
102 "NTMD\_YQD" - Nacelle Y TMD velocity  
103 END of input file (the word "END" must appear in the first 3 columns of this last OutList line)  
104 \_\_\_\_\_

## C. Technical Yaw drive data



# TECHNICAL DATA

## YAW DRIVE



**711T4N Series**  
Gearboxes for 1.0 to 2.0 MW wind turbines



**714T4F+BN132 Series**  
Gearboxes for 2.0 to 4.0 MW wind turbines



**716T4U Series**  
Gearboxes for 5.0 to 8.0 MW wind turbines

WIND TURBINE SIZE [MW]	YAW DRIVE NUMBER	GEARBOX TYPE & MAX STATIC TORQUE	ELECTRIC MOTOR TYPE & POWER	INVERTER TYPE & POWER
up to 1.0	2 ÷ 4	706 T ÷ 709 T 15 ÷ 55 [kNm]	BN 80 ÷ 100 0.75 ÷ 2.2 [kW]	AGL / ACU size 1÷2 0.75 ÷ 2.2 [kW]
1.0 ÷ 1.5	2 ÷ 4	709 T ÷ 712 T 55 ÷ 110 [kNm]	BN 90 ÷ 112 1.1 ÷ 4.0 [kW]	AGL / ACU size 1÷2 1.1 ÷ 4.0 [kW]
1.5 ÷ 2.0	4 ÷ 6	709 T ÷ 714 T 55 ÷ 150 [kNm]	BN 100 ÷ 132 2.2 ÷ 5.5 [kW]	AGL / ACU size 2÷3 2.2 ÷ 5.5 [kW]
3.0 ÷ 4.0	4 ÷ 8	710 T ÷ 714 T 60 ÷ 150 [kNm]	BN 100 ÷ 132 3.0 ÷ 9.2 [kW]	AGL / ACU size 2÷3 3.0 ÷ 9.2 [kW]
5.0 ÷ 6.0	6 ÷ 8	714 T ÷ 716 T 150 ÷ 250 [kNm]	BN 132 ÷ 160 7.5 ÷ 15.0 [kW]	ACU size 3÷4 7.5 ÷ 15.0 [kW]
7.0 ÷ 8.0	6 ÷ 10	716 T ÷ 718 T 250 ÷ 400 [kNm]	BN 132 ÷ 180 9.2 ÷ 22.0 [kW]	ACU size 3÷5 9.2 ÷ 22.0 [kW]
9.0 ÷ 12.0	8 ÷ 16	716 T ÷ 718 T 250 ÷ 400 [kNm]	BN 132 ÷ 180 9.2 ÷ 22.0 [kW]	ACU size 3÷5 9.2 ÷ 22.0 [kW]

TYPE	NOMINAL TORQUE Nm	PEAK STATIC TORQUE Nm	RANGE OF RATIOS 1:	AVAILABLE PINION MODULE m	WEIGHT kg
706 T	9.500	16.000	600-3.000	10 ÷ 14	120
707 T	12.500	25.000	600-3.000	12 ÷ 16	170
709 T	25.000	55.000	600-3.000	12 ÷ 20	300
710 T	30.000	60.000	600-3.000	14 ÷ 20	350
711 T	37.500	80.000	600-3.000	16 ÷ 20	400
712 T	50.000	110.000	600-3.000	16 ÷ 20	550
714 T	70.000	150.000	600-3.000	20 ÷ 24	800
716 T	100.000	250.000	600-3.000	22 ÷ 26	1.000
717 T	120.000	300.000	600-3.000	26 ÷ 30	1.800
718 T	150.000	400.000	600-3.000	30 ÷ 36	2.100

The indicated data are for reference only; please contact Bonfiglioli for more detailed information.

# TECHNICAL DATA

## PITCH DRIVE



**707T3N+BN132 Series**  
Gearboxes for 1.0 to  
2.0 MW wind turbines



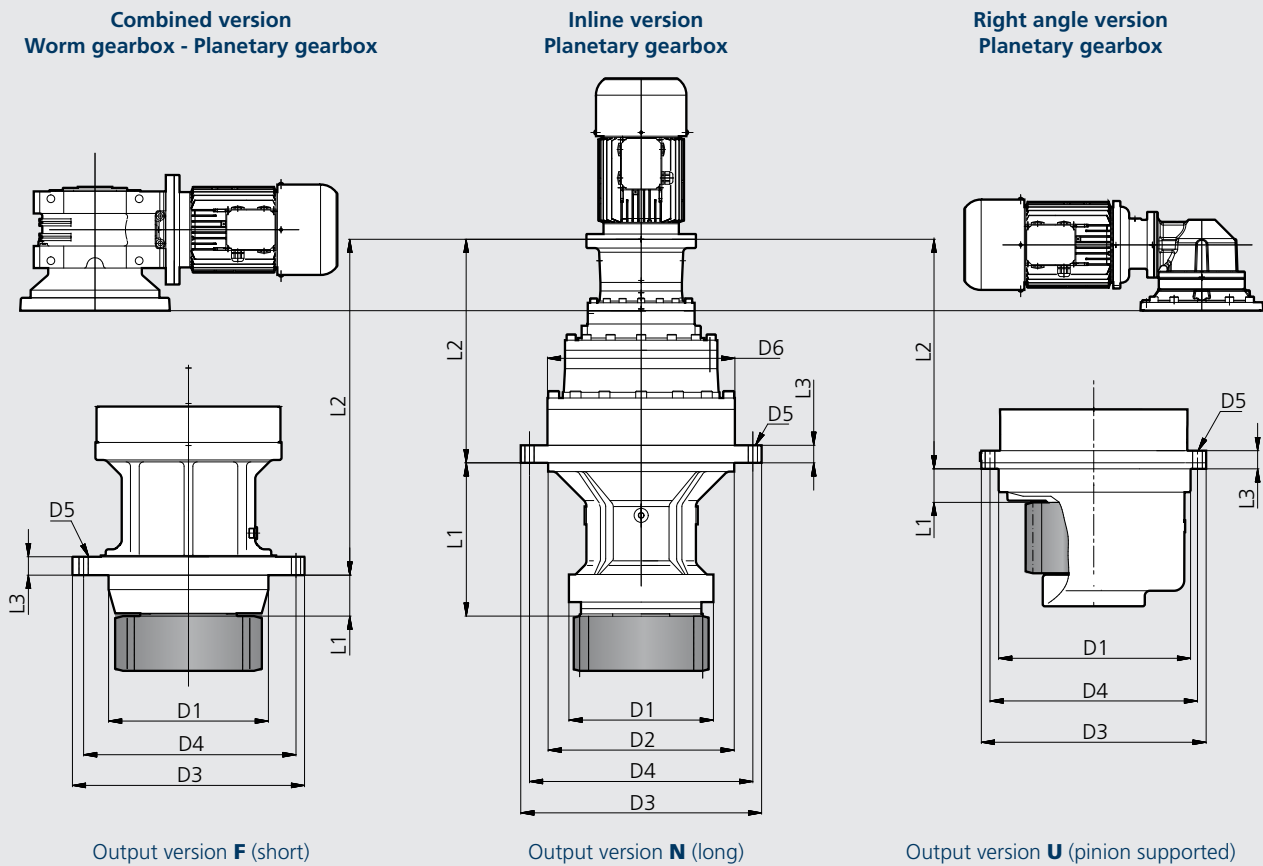
**707T3N Series**  
Gearboxes for 1.0 to  
2.0 MW wind turbines

WIND TURBINE SIZE	PITCH DRIVE NUMBER	GEARBOX TYPE & MAX STATIC TORQUE	ELECTRIC MOTOR TYPE & POWER
[MW]			
up to 1.0	3	703 T ÷ 706 T 4.5 ÷ 16 [kNm]	BN 90 ÷ 100 1.1 ÷ 3.0 [kW]
1.0 ÷ 1.5	3	705 T ÷ 707 T 8 ÷ 25 [kNm]	BN 100 ÷ 132 2.2 ÷ 5.5 [kW]
1.5 ÷ 2.0	3	706 T ÷ 709 T 16 ÷ 55 [kNm]	BN 100 ÷ 132 3.0 ÷ 7.5 [kW]
3.0 ÷ 4.0	3	707 T ÷ 711 T 25 ÷ 80 [kNm]	BN 132 ÷ 160 5.5 ÷ 15.0 [kW]
5.0 ÷ 6.0	3	711 T ÷ 712 T 80 ÷ 110 [kNm]	BN 132 ÷ 180 9.2 ÷ 22.0 [kW]
7.0 ÷ 8.0	3 ÷ 6	712 T 110 [kNm]	BN 160 ÷ 200 11.0 ÷ 30.0 [kW]
9.0 ÷ 12.0	3 ÷ 6	712 T 110 [kNm]	BN 160 ÷ 200 11.0 ÷ 30.0 [kW]

TYPE	NOMINAL TORQUE	PEAK STATIC TORQUE	RANGE OF RATIOS	AVAILABLE PINION MODULE	WEIGHT
	Nm	Nm	1:	m	kg
703 T	2.500	4.500	100-250	10 ÷ 12	60
705 T	4.000	8.000	100-250	10 ÷ 12	90
706 T	7.500	16.000	100-250	10 ÷ 14	120
707 T	10.000	25.000	100-250	12 ÷ 16	170
709 T	20.000	55.000	100-250	12 ÷ 20	300
710 T	25.000	60.000	100-250	14 ÷ 20	350
711 T	30.000	80.000	100-250	16 ÷ 20	400
712 T	40.000	110.000	100-250	18 ÷ 20	500

The indicated data are for reference only; please contact Bonfiglioli for more detailed information.

# OVERALL DIMENSIONS



TYPE	VERSION	D1	D2	D3	D4	D5	D6	L1	L2	L3
703T	F	175		275	245	∅ 18 n°10	244	41	370	20
705T	F	175		275	245	∅ 18 n°10	244	41	400	20
706T	F	250		360	320	∅ 18 n°24	292	130	460	35
706T	N	200	250	360	325	∅ 17 n°10	292	225	350	25
707T	F	310		410	360	∅ 22 n°12	348	70	540	30
707T	N	230	280	348	314	∅ 17 n°12	348	300	360	98
709T	F	310		410	360	∅ 22 n°12	348	70	540	30
709T	N	230	280	348	314	∅ 17 n°24	348	300	400	125
709T	U	340		405	375	∅ 17 n°24	348	90	450	40
710T	F	320		410	370	∅ 21 n°21	400	75	600	35
710T	N	300	425	500	450	∅ 22 n°12	400	360	500	40
710T	U	340		400	370	∅ 17 n°24	400	36	550	176
711T	F	390		520	480	∅ 17 n°30	428	60	700	35
711T	N	300	425	500	460	∅ 22 n°12	428	350	520	40
712T	F	410		490	450	∅ 21 n°24	428	125	660	40
712T	N	400	425	520	470	∅ 21 n°24	428	318	580	40
712T	U	415		530	480	∅ 26 n°16	428	140	500	45
714T	F	420		530	490	∅ 22 n°24	490	160	870	40
714T	U	555		645	600	∅ 30 n°32	490	97	760	100
716T	F	555		650	600	∅ 30 n°32	542	70	900	50
716T	U	555		650	600	∅ 30 n°32	542	70	900	50
717T	F	630		740	680	∅ 27 n°32	695	112	1250	50
718T	U	750		900	830	∅ 32 n°24	695	60	900	100

The indicated data are for reference only; please contact Bonfiglioli for more detailed information.

## D. MATLAB script for Rainflow counting algorithm

```

1 %% Rainflow counting
2
3 %% Selecting YawBrMzp output
4 YawBrMzp = YawMom.t1(721:end)*10^6; % Nmm (First 8 seconds are unstable)
5
6 %% Dimensions of Yaw Pinion Gear and Yaw Gear along with loads
7 d_b = 2000; % meters diameter bearing [mm]
8 d_p = 425; % diameter pinion gear [mm]
9 r = d_b/2; % radi bearing [mm]
10 m = 17; % module spur gear [mm]
11 b1 = 100; % width [mm]
12 z = d_p/m; %% number of teeth
13 Y = 0.359; %% Lewis shape factor
14
15 F_t = YawBrMzp / (r); % N
16 sigma_f = F_t / ((m) * (b1) * Y)/10; % Stress load by 10 pinion gears N/mm^2 (
    MPa)
17
18 %% Rainflow algorithm and plotting results
19 [c,hist,edges,mmm,idx] = rainflow(sigma_f);
20
21 G = array2table(c, 'VariableNames', {'Count', 'Range', 'Mean', 'Start', 'End'});
22
23 figure(1)
24 histogram('BinEdges',edges, 'BinCounts',sum(hist,2))
25 xlabel('Stress Range')
26 ylabel('Cycle Counts')
27 t = Time(721:end,1);
28 figure(2)
29 rainflow(sigma_f,t); % Rainflowcounting algorithm
30
31 B = length(hist);
32 sigma_fatigue = 1220; %MPa fatigue strength coefficient
33 sigma_ult = 517; % MPa Ultimate strength
34 sigma_e = 0.5 * sigma_ult; % Endurance limit
35 b = (log(sigma_e)-log(0.9*sigma_ult))/3; % Fatigue strength exponent
36 A = sigma_e/(10^(6*b));
37 S = 0;
38
39 %% Cycle counting and summarize damage
40 for k = 1:B
41     N_k = sum(hist(k:k,:)); % Cycle Counts
42     sigma_mean = edges(k+1)/2; % Mean stress cycle k
43     sigma_a = edges(k+1) - edges(k); % Stress amplitude for N_k
44     N_f_k = (sigma_a/A)^(1/b);
45     %N_f_k = (1/2)*(((sigma_a)/(sigma_fatigue))*(1/(1-(sigma_mean/sigma_ult))))
        ^ (1/b);
46     S = S + (N_k/N_f_k);
47 end
48
49 %% Damage accumulation 1 hour of simulation
50 D1 = B * S

```

## E. MATLAB script for plotting simulated FAST data

```

1 %% Fast fourier transform on rotorspeed
2 % 01. Mars 2018 - 26. October 2018
3
4

```



```

57 V_y = Mainout(721:end,3); % m/s
58 V_z = Mainout(721:end,4); % m/s
59 subplot(3,1,1)
60 plot(time,[V_x,V_y,V_z]);
61 legend('V_x','V_y','V_z')
62 ylim([-5,15]);
63 xlabel('time(s)')
64 title('Wind Speed (m/s)');
65
66 %% Plotting Moments
67 subplot(3,1,2)
68 plot(time,[YawBrMxp,YawBrMyp,YawBrMzp]);
69 title('Yaw-bearing moments')
70 ylim([-6500,6500]);
71 xlabel('time(s)')
72 legend(['YawBrMxp(kN\cdotm)'; 'YawBrMyp(kN\cdotm)'; 'YawBrMzp(kN\cdotm)'])
73
74 %% Plotting FFT
75 subplot(3,1,3)
76 plot(Fbins,X_mag-2);
77 xlim([0,6]);
78 xlabel('Frequency(Hz)')
79 ylabel('PSD (m^2\cdot s^-3)')
80 title('FFT YawBrMyp')

```