Master's thesis

NTNU
Norwegian University of Science and Technology

Øystein Morken Linde and Magnus Kornelius Sæland

Estimating modal damping due to hysteric soil behaviour and its effect on fatigue response of offshore wind structures on jackets and monopiles

Master's thesis in Engineering and ICT, and Civil and Environmental Engineering

January 2022

NTNU
Norwegian University of
Science and Technology

Øystein Morken Linde and Magnus Kornelius Sæland

Estimating modal damping due to hysteric soil behaviour and its effect on fatigue response of offshore wind structures on jackets and monopiles

Master’s thesis in Engineering and ICT, and Civil and Environmental Engineering
January 2022

Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

Department of Structural Engineering
Faculty of Engineering
NTNU – Norwegian University of Science and Technology

ACCESSIBILITY
Open

MASTER THESIS 2022

SUBJECT AREA: Structural dynamics	DATE: 14.06.2022	NO. OF PAGES: 163

TITLE:

Estimating modal damping due to hysteric soil behaviour and its effect on fatigue response of offshore wind structures on jackets and monopiles

Estimering av modal demping fra jords hystereseoppførsel og dens effekt på utmattingsrespons av offshore vindturbiner på jacketer og monopeler

BY:

Øystein Morken Linde and Magnus Kornelius Sæland

SUMMARY:

Fatigue is a vital element in the design of offshore wind turbines. The effect of soil damping has been shown to impact fatigue life. However, no consensus exists on the value of soil damping or how to estimate it.

This thesis estimated the modal soil damping by developing a numerical methodology. Both hysteric and radiation damping were taken into account. The effect of the estimated soil damping was assessed by performing fatigue analyses.

Two structures were analysed: one 10 MW monopile structure, only including hysteric damping, and one 10 MW jacket structure, including hysteric and radiation damping. The weighted average damping in the fore-aft modes was estimated to be 3.50% and 4.44% for the monopile and the jacket -structure, respectively. The damping in the side-to-side modes were much smaller. By including hysteric soil-damping, fatigue life increased by 77% and 274% for the monopile and jacket -structure, respectively. Combined radiation and hysteric soil-damping in the jacket structure gave a 285% increase in fatigue life.

The hysteric damping was estimated by assessing the energy in each mode relative to the energy loss in the soil associated with the mode. Damping curves were used to estimate the energy loss in the soil of an Abaqus analysis simulating the pile-soil interaction. A modal analysis was implemented to find the energy of each mode and the associated representative displacement of the piles. Loads exciting the structure were extracted from FAST. FAST models with baseline damping were established for the two chosen structures. Then the estimated damping values from the modal analysis were added to compatible FAST modes. Finally, a fatigue analysis was carried out on the established models with both baselines and added soil damping.

A substantial increase in fatigue life can be achieved by including soil damping. An increase in fatigue life could, in turn, be exchanged for a longer operational life or reduced material usage. Therefore, a reduction in the cost and carbon footprint per unit of electricity produced by offshore wind turbines can be achieved by including soil damping.

RESPONSIBLE TEACHER: Amir M. Kaynia

SUPERVISOR: Amir M. Kaynia

CARRIED OUT AT: Department of Structural Engineering

NTNU
Kunnskap for en bedre verden

Department of Structural engineering

Master thesis

Estimating modal damping due to hysteric soil behaviour and its effect on fatigue response of offshore wind structures on jackets and monopiles

Authors:
Øystein Morken Linde
Magnus Kornelius Sæland

June 14, 2022

Abstract

Fatigue is a vital element in the design of offshore wind turbines. The effect of soil damping has been shown to impact fatigue life. However, no consensus exists on the value of soil damping or how to estimate it.

This thesis estimated the modal soil damping by developing a numerical methodology. Both hysteric and radiation damping were taken into account. The effect of the estimated soil damping was assessed by performing fatigue analyses.

Two structures were analysed: one 10 MW monopile structure, only including hysteric damping, and one 10 MW jacket structure, including hysteric and radiation damping. The weighted average damping in the fore-aft modes was estimated to be 3.50% and 4.44% for the monopile and the jacket -structure, respectively. The damping in the side-to-side modes were much smaller. By including hysteric soil-damping, fatigue life increased by 77% and 274% for the monopile and jacket -structure, respectively. Combined radiation and hysteric soil-damping in the jacket structure gave a 285% increase in fatigue life.

The hysteric damping was estimated by assessing the energy in each mode relative to the energy loss in the soil associated with the mode. Damping curves were used to estimate the energy loss in the soil of an Abaqus analysis simulating the pile-soil interaction. A modal analysis was implemented to find the energy of each mode and the associated representative displacement of the piles. Loads exciting the structure were extracted from FAST. FAST models with baseline damping were established for the two chosen structures. Then the estimated damping values from the modal analysis were added to compatible FAST modes. Finally, a fatigue analysis was carried out on the established models with both baselines and added soil damping.

A substantial increase in fatigue life can be achieved by including soil damping. An increase in fatigue life could, in turn, be exchanged for a longer operational life or reduced material usage. Therefore, a reduction in the cost and carbon footprint per unit of electricity produced by offshore wind turbines can be achieved by including soil damping.

Sammendrag

Utmatting er en vesentlig faktor i dimensjoneringen av offshore vindturbiner. Det har blitt vist at jorddemping påvirker utmattingslevetider. Til tross for viktigheten av jorddemping så er det ingen enighet i fagmiljøet hvor stor jorddempingen faktisk er, eller hvordan denne bør beregnes.

Denne oppgaven estimerer modal jorddemping ved å utvikle en numerisk metodikk. Både hysterese og radielldemping er inkludert i beregningen. Effekten av den estimerte jorddempingen ble evaluert ved å utføre flere utmattingsanalyser.

To konstruksjoner ble analysert: En 10 MW monopel-konstruksjon med bare hysterese demping, og en 10 MW jacket-konstruksjon med både hysterese og radielldemping. Det vektede snittet av dempingen i vibrasjonsmodene i vindretningen ble estimert til 3,50% for monopelen og 4,44% for jacketen. Dempingen til side-til-side modene var betydelig lavere. Monopel-konstruksjonens levetid ble økt med 77% da hysterese-jorddempingen ble inkludert. Jacket-konstruksjonen fikk 274% lenger levetid ved å inkludere hysterese-jorddempingene. Ved å i tillegg inkluderer radielldemping ble økningen i levetiden for jacketen ytterligere økt til 285%.

Hysterese dempingen ble beregnet ved å bruke forholdet mellom energien i hver mode og tilhørende energitap i jorden. Energitapet ble beregnet ved å bruke dempingskurver i postprosesseringen av en Abaqus-analyse, som simulerte pel-jordinteraksjonen. En modalanalyse ble implementert for å beregne energien i hver mode og den tilhørende representative forskyvningen av pelene. Lastene som virket på konstruksjonen, ble hentet fra FAST. FAST-modeller med en grunndemping ble etablert for de to konstruksjonene. Deretter ble den estimerte dempingen fra modalanalysen lagt til i ekvivalente moder i FAST. Til slutt ble det utført utmattingsanalyser på de to modellene med grunndemping og ekstra demping fra jord.

En betydelig økning i utmattingslevetiden kan oppnås ved å inkludere jorddemping. Dette kan igjen føre til økt levetid, eller redusert materialbruk. Dermed kan elektrisiteten fra offshore vindturbiner produseres billigere og med lavere karbonavtrykk ved å hensynta jorddempingen.

Contents

Nomenclature

$[\Phi]$	Eigenvectors
$[]$	Matrix
$[\bar{C}_{BB}]$	Guyan damping matrix
$[\bar{C}_{NN}]$	Craig-Bampton modes damping matrix
$[\tilde{K}]$	Stiffness matrix in modal space
$[\tilde{M}]$	Mass matrix in modal space
$[C]$	Damping matrix
$[K]$	Stiffness matrix
$[M]$	Mass matrix
$\{\tilde{r}\}$	Displacement vector in modal space
$\{\}$	Column vector
$\{\tilde{P}\}$	Load vector in modal space
$\{P\}$	Load vector
$\{r\}$	Displacement vector
β	ω/ω_n
ϵ	Strain
η	Loss factor
γ	Peak-enhandement factor for wave modelling
λ_0	Weibull scale parameter
ω	Load frequencies
ω_n	Eigenfrequencies
Π	Energy
σ	Stress
σ_R	Stress range
ξ	Damping ratio as share of critical damping
A_L	Absorbed energy in each element
D	Damping ratio from soil effects
H_S	Significant wave height

k	Weibull shape parameter
m	Mass in structural dynamics - Crack growth parameter in fatigue
W_L	Absorbed energy in soil
z_0	Terrain surface roughness parameter

Acronyms

CA constant amplitude

C-B Craig-Bampton

DEL damage equivalent load

DFF design fatigue factor

DLC design load condition

DNV Det Norske Veritas

DOF degree of freedom

E-B Euler-Bernoulli

EWM extreme wind model

FAST Fatigue, Aerodynamics, Structures and Turbulence

FE finite element

FEA finite element analysis

FEM finite element method

FFT fast Fourrier transform

IEA International Energy Agency

IEC International Electrotechnical Commission

JONSWAP Joint North Sea Wave Project

MDOF multiple degrees of freedom

NFEA nonlinear finite element analysis

NREL National Renewable Energy Laboratory

NTM normal turbulence model

OMA operational modal analysis

OWT offshore wind turbine

PDF probability density function

P-M Pierson-Moskowitz

PRNG psuedorandom number generator

RNA rotor nacelle assembly

SDOF single degrees of freedom

SLS serviceability limit state

S-N Stress-life

SSI soil-structure interaction

TP transition piece

ULS ultimate limit state

VA variable amplitude

Definitions

This section defines the non-standard definitions used in this thesis.

Soil damping is the damping from the soil-pile interaction on the system of interest. If nothing else is indicated the system of interest is the whole structure.

Dynamic energy is the energy in the dynamic part of the system that can be converted to kinetic energy. The expression is defined mathematically in Equation 2.61 in Section 2.5.1.

The foundation is the parts of the structure bellow the mudline.

Foundation damping is the damping of the foundation part of the system, not the effect of damping of the foundation on the overall system.

Preface

This Thesis has been written as the final project for the Master of Science. The work was conducted at the Department of Structural Engineering at the Norwegian University of Science and Technology (NTNU) during the spring semester of 2022. The authors have a background from the study programmes Civil and Environmental Engineering, and Engineering and ICT. Both with a specialisation in structural engineering - structural dynamics.

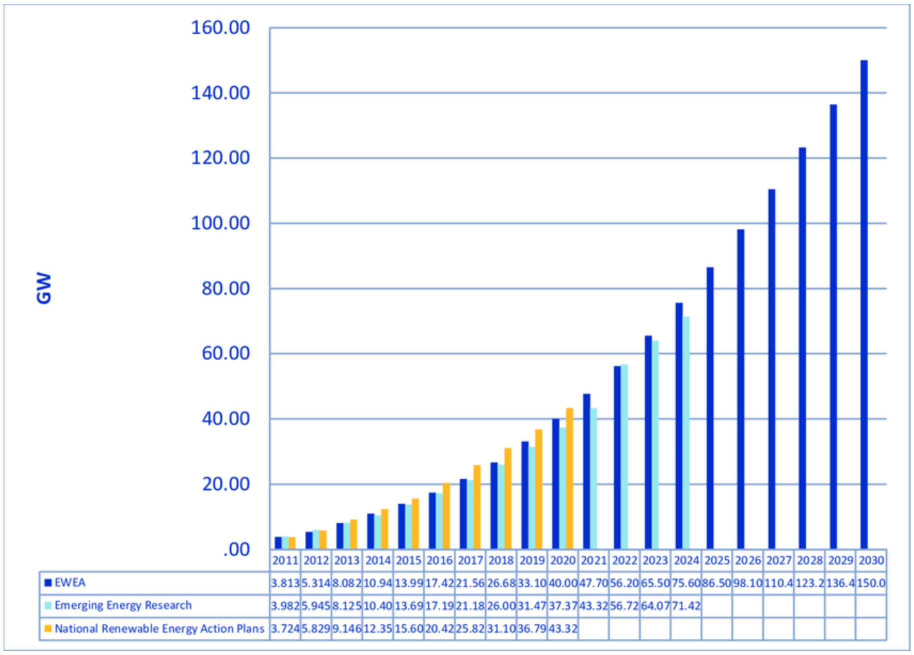
Acknowledgements

We want to express our gratitude to our supervisor, Professor Amir M. Kaynia. We highly appreciate his advice, interest and availability beyond what can be expected. Further, we want to thank Daniel Martens Pedersen and Henrik Askheim for providing us the model they built for their Master Thesis in 2021

Magnus Kornelius Sæland, Øystein Morken Linde

1. Introduction

Offshore wind turbines (OWT) are a relatively new technology, with the first OWTs being installed in 1991 off the coast of Denmark. In recent years there has been a significant increase in the number of installed OWTs. The total installed capacity is expected to continue to grow in the coming years, as illustrated in Figure 1.1. In response to the Russian attack on Ukraine in 2022, the EU commission has signalled that Europe should become independent from Russian fossil fuels and, in the process, increase renewable power production [1]. Moreover, the EU, UK, and Norway have recently increased their offshore wind committments. The countries currently aim to have an installed capacity of 450 GW of offshore wind by 2050 [2]. The recent increase in energy plans might make previous forecasts underestimated.



	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030
■ EWEA	3.813	5.314	8.082	10.94	13.99	17.42	21.56	26.68	33.10	40.00	47.70	56.20	65.50	75.60	86.50	98.10	110.4	123.2	136.4	150.0
■ Emerging Energy Research	3.982	5.945	8.125	10.40	13.69	17.19	21.18	26.00	31.47	37.37	43.32	56.72	64.07	71.42						
■ National Renewable Energy Action Plans	3.724	5.829	9.146	12.35	15.60	20.42	25.82	31.10	36.79	43.32										

Figure 1.1: Forecast of offshore wind installations (ref. [3])

Over the last couple of years, there has been a significant increase in the rated power of turbines. The Siemens Gamesa's SG 14-222 DD has the current biggest effect yet installed with a max capacity of 15 MW [4]. The increase in size is driven by the wish to reduce the cost per unit of electricity produced. Wind turbines have a power production proportional to the swept area of the turbine blades. This gives an incentive to build larger turbines as the power production increases with the square of the turbine diameter. This can be observed in Equation (1.1).

$$P = \frac{1}{2}\,\rho_{air}\;C_P\;A_S\;v_{Rel}^3 \tag{1.1}$$

where P is the power production, ρ_{Air} is the air density, C_P is the power coefficient, A_S is the swept area of the turbine and v_{Rel} is the velocity of the wind. As the power production is proportional to the wind speed cubed, there is a large incentive to put wind turbines in areas with good wind conditions. Areas at sea are ideal as the wind is less obstructed, leading to higher wind speed and less turbulence. In turn, the increased wind speed and the reduced turbulence lead to higher power output. In addition, areas at sea typically have fewer conflicts of interest than their land-based counterparts. However, the OWTs have the disadvantage of being subject to more demanding loading conditions, leading to greater challenges in engineering the structures.

The world installed a capacity of 6.1 GW of offshore wind in 2020, with the vast majority being installed in Europe and China [5]. That makes the total installed offshore capacity 35.3 GW, which is about 5% of the onshore capacity of 707.4 GW.

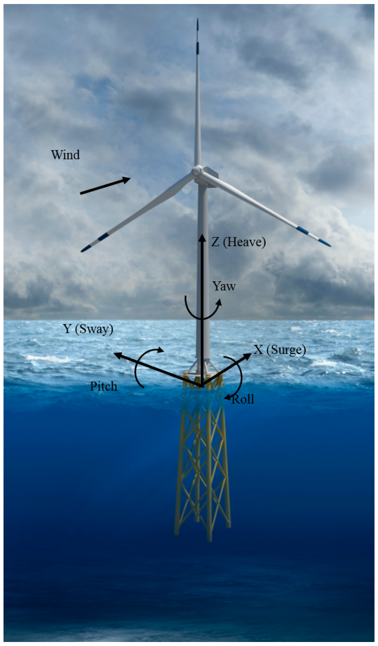


Figure 1.2: Coordinate system used throughout the thesis (ref. [6])

1.1 Future development and challenges

It is predicted in IEA's sustainable energy scenario that the world needs to produce 606 TWh of electricity from offshore wind in 2030 up from 67 TWh in 2018 [7]. Cost is of utmost importance to achieve this increase.

One of the design criteria for OWTs is fatigue, where a slight change in the damping can have a substantial impact on the overall life expectancy [8] as illustrated in Figure 1.3.

One of the biggest sources of damping is soil damping. The NEK EN IEC 61400-3-1:2019 [9], NS-EN ISO 19901-4:2016 [10], DNVGL-RP-C212 [11] and the DNVGL-ST-0126 [12] standards all express a need to take soil damping into account, but no specific method is recommended to find the damping effect from soil.

If it can be proven that the inclusion of soil damping can increase the overall damping compared to today's estimate, this could increase fatigue life and ULS load capacity. Increased fatigue life could, in turn, result in longer design life or less use of materials in construction, both leading to reduced costs and carbon emissions per unit of electricity produced.

Damping effects from soil damping as reported in literature vary greatly, spanning from 0.17% to 1.5% for monopiles [13–17].

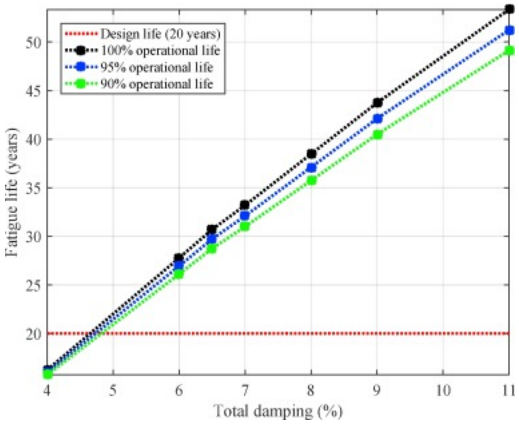


Figure 1.3: Effect of damping for fatigue life for a 5 MW monopile (ref. [8])

1.2 Previous work in the field

Rezaei et al. (2018) [8] performed a sensitivity analysis of a 5 MW monopile OWT using finite element simulations. The article states that the dynamic response of the structure is highly dependent on the combined aerodynamic, hydrodynamic, structural, and soil damping present. Furthermore, the article states that the aerodynamic damping is the largest contributor to overall damping. The model uses beam elements for the monopile and tower and includes nonlinear p-y curves for soil-structure interaction (SSI). The fatigue life of the structure was found to increase almost linearly with increased levels of overall damping.

Zhang et al. (2021) [18] presents a simplified model for estimating the soil damping due to nonlinear soil response for pile foundations. They used a finite element analysis (FEA) model of an offshore wind turbine (OWT) on a monopile to find the hysteric damping ratio for different levels of cyclic loading for the pile. They found an increased foundation damping ratio for increased displacement of the monopile. The proposed model is fundamentally linked to the damping response of the soil measured at element level. Therefore it offers design engineers an efficient and accurate way to estimate soil-pile interaction damping based on site-specific soil data.

Carswell et al. (2022) [13] evaluated the effect of hysteric foundation damping on a monopile for production, emergency stop and parked conditions using a rotational dashpot damper. A 5 MW reference turbine was modelled using the aero-hydro-elastic software FAST and included equivalent linear foundation stiffness and damping matrices. They found critical damping in the range 0.3-0.7%. Further, they found that the foundation damping significantly affected response during both the emergency shutdown and parked conditions.

Carswell et al. (2015) [14] used the logarithmic decrement method on a free vibration time history for a finite element model of a 5 MW monopile OWT with a rotational dashpot. They estimated that the foundation damping contributed 0.17-0.28% of the critical damping to total OWT damping. The authors used a 1.00% Rayleigh structural damping.

Damgaard et al. (2013) [15] analysed a series of rotor stop-tests from 2006 to 2011 on several monopile foundations in multiple wind parks. They found mean values of the logarithmic decrement δ_1 in the range 0.15-0.16 (damping ratio ξ between 2.39% and 2.55%). They estimated the modal soil damping to be 0.06 in terms of logarithmic decrement δ (damping ratio $\xi = 0.95\%$).

Versteijlen et al. (2011) [16] analysed 12 rotor stop tests on the BB16 test turbine. They estimated that the soil-pile interaction contributed 1.5% of critical damping in the first fore-aft mode.

Markou and Kaynia (2015) [19] performed a parametric study, which involves different generalised nonlinear mechanical formulations with different damping characteristics to account for the interaction between a monopile-supported OWT and the surrounding soil. They compared four different soil-pile interaction models and concluded that some commonly used models are unfit to describe damping in soil-pile interaction. The article emphasises the importance that proper simulation of the soil-pile interaction phenomena has for the estimation of the fatigue lifetime of OWTs.

Kaynia and Andersen (2015) [20] incorporated the nonlinear foundation response in dynamic SSI analyses using nonlinear soil springs. A nonlinear force-deformation curve represented the foundation. The methodology was applied to a gravity-based platform. Comparison with measured data confirmed the application of the adopted method.

Cruz and Miranda (2017) [21] evaluated the effects of SSI on damping ratios of buildings subjected to earthquake ground motions. Their results indicate that SSI effects may either increase or reduce the effective modal damping ratio of the fundamental period. Furthermore, they show that SSI effects lead to an approximately linear trend in effective modal damping ratios with increasing modal frequency.

Aasen et al. (2017) [22] studied the effect of foundation modelling on accumulated fatigue. They state that for the monopile foundation, the industry standard for pile-analysis has shown to be inaccurate. In the study, they investigated four different models. The study showed that stiffness and damping have a noticeable effect on fatigue damage, especially in idling cases. The accumulated fatigue damage at the mudline showed a variance up to 16%, depending on the foundation model used.

Shirzadeh et al. (2013) [17] used an Operational Modal Analysis (OMA) to identify the damping value of the fundamental fore-aft mode of a 3 MW OWT using both real-life measurements and simulations. They obtained experimental data from a measurement campaign on an OWT in the Belgian North Sea. Their measured results were compared with the numerical simulations carried out in HAWC2. Finally, they estimated the soil-damping contribution to be about 0.25%.

Malekjafarian et al. (2021) [23] performed a critical review on the recent work on foundation damping for monopile structures. They also reviewed the main methods used in numerical and experimental assessment of the foundation damping. Finally, they discussed the challenges to overcome, recommendations for the accurate estimation of foundation damping and the importance of damping to the OWTs' fatigue life.

Ishihara and Wang (2019) [24] studied modal damping ratios of OWTs systematically for different soil properties and foundation types. They used a numerical model where the foundation was modelled by using a Winkler model. Using this model they showed that the first and second mode depend strongly on the soil stiffness.

Chi-Yu et al. (2018) [25] investigated two different sampling methods for environmental conditions of an OWT. These were a traditional grid-based method and a Monte Carlo method. This was done in an attempt to reduce the amount of simulations performed during a fatigue analysis. They found a one-dimensional grid-based method have poor accuracy on their OWT.

Michel et al. (2017) [26] studied the impact of SSI on the fatigue design of the concrete foundation of a wind turbine. They found that SSI altered time domain effects such as eigenfrequency and bedding, but it did not change the fatigue design. Although SSI did not impact the design of the gravity base, Michel et al. highlighted the need to perform thorough analyses of other turbine parts.

Yeter et al. (2016) [27] evaluated spectral models to predict the fatigue life of a tripod OWT. The time domain rainflow counting method was used as a reference for several spectral models. The wide band solutions were deemed to be better than narrow banded solutions. The wide banded Tunna method would overestimate damage by 34%, while the narrow banded Rayleigh method overestimated damage by 244%.

Sørum et al. (2019) [28] examined how marine conditions impacted the fatigue damage. Specifically they studied crest length, wind-wave misalignment and soil stiffness. Opposed to previous consensus, it was determined that short-crested waves may deal more damage at greater depths and softer soils than long crested waves.

1.3 Scope of thesis research

This section presents the objectives of the thesis research and an overview of the methodology used to achieve the objectives.

1.3.1 Objectives

The objectives of the thesis are to:

1. Establish a theoretical model to calculate soil damping effect on overall damping
2. Use the theoretical model to calculate the soil damping effect on the overall system's modal damping values
3. Perform a fatigue analysis to quantify the effect of the calculated soil damping on the fatigue life of the structure

1.3.2 Methodology

This section gives an overview of the methodology used to achieve the goals of the thesis presented in Section 1.3.1. An overview of the thesis is given in the flow chart in Figure 1.4.

1.3.2.1 Step 1 - Theoretical model

The authors could not find a previous numerical method to establish the hysteric soil modal damping for OWTs using a full finite element (FE)-model. Therefore, the authors established a model to calculate this damping. The model is presented in Section 2.5.1, where damping curves and modal properties of the turbines under a representative displacement are used to come up with a formula for the modal damping. Incorporating the foundation damping using a dashpot was considered, but this feature was not present in FAST. FAST was the numerical tool for computation of the environmental loads and fatigue.

1.3.2.2 Step 2 - Damping estimation

Two OWTs were considered for the analysis: One monopile structure and one jacket structure. Two models of the structures had to be utilised: one FAST-model (Chapter 4) to generate environmental loads and one Abaqus model (Chapter 3) to estimate damping. The FAST model in this step was used to generate loading that an extension of the Abaqus model used to calculate a representative displacement for the piles for each mode exciting the soil. This representative displacement of the piles was, in turn, used to calculate the energy loss in the soil associated with the representative displacement. Together with the dynamic energy associated with each mode, the energy loss in the soil was used to calculate the hysteric soil damping of each mode.

1.3.2.3 Step 3 - Fatigue

Five fatigue analyses were performed using FAST: One analysis to set the baseline for each of the jacket and monopile-structure for a level of damping equal to 1.0%. Then, one additional analysis for each of the two structures adding the hysteric modal soil damping calculated in step 2, was performed. Finally, one analysis was performed for the jacket, including both radiation and hysteric damping. A comparison between the fatigue analyses with and without the soil damping was conducted to assess if the added level of soil damping increased the fatigue life.

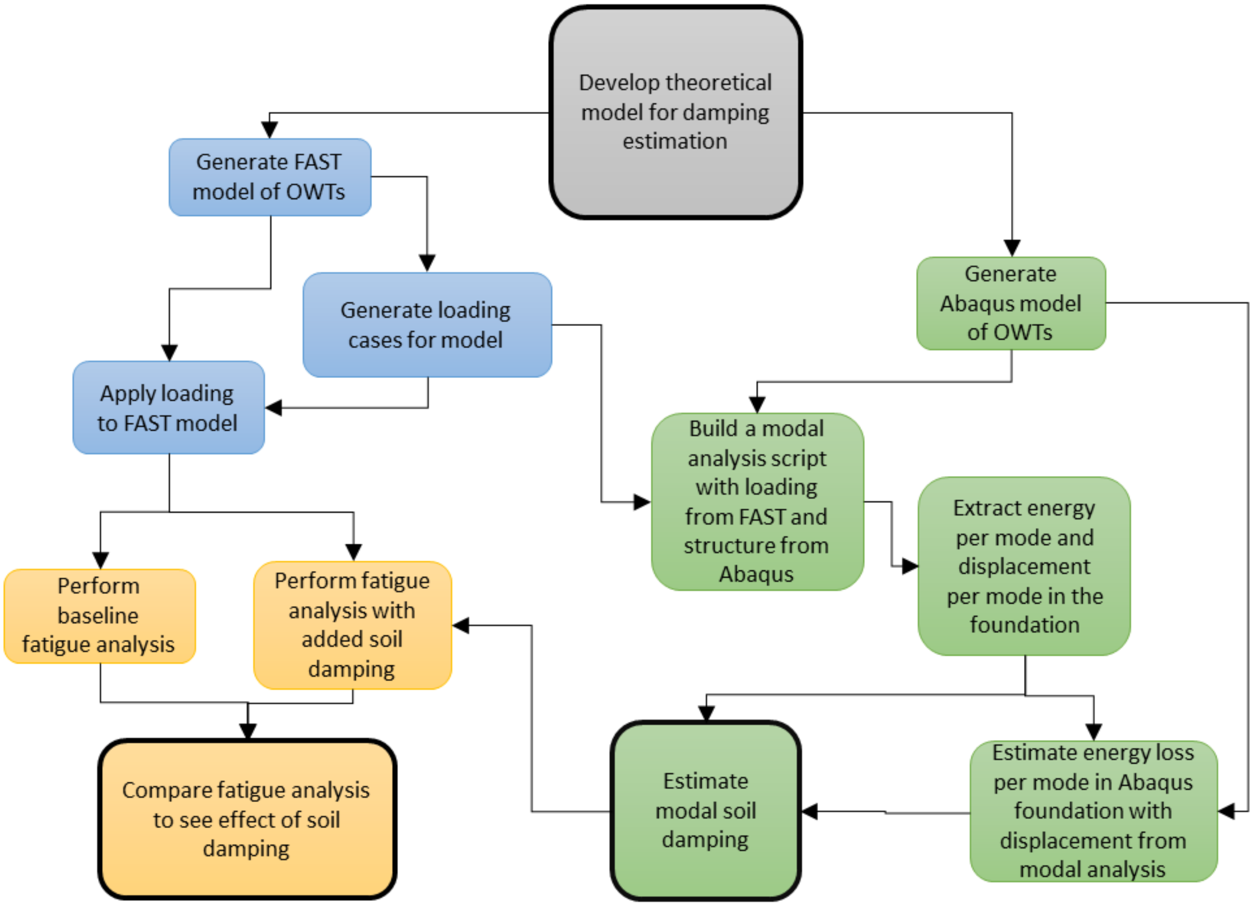

Develop theoretical model for damping estimation
Generate FAST model of OWTs
Generate loading cases for model
Apply loading to FAST model
Perform baseline fatigue analysis
Perform fatigue analysis with added soil damping
Compare fatigue analysis to see effect of soil damping
Generate Abaqus model of OWTs
Build a modal analysis script with loading from FAST and structure from Abaqus
Extract energy per mode and displacement per mode in the foundation
Estimate modal soil damping
Estimate energy loss per mode in Abaqus foundation with displacement from modal analysis

Figure 1.4: Flow chart of thesis

2. Theory

The theory used throughout the thesis is presented in this chapter.

2.1 Fatigue

Fatigue is a failure mechanism that often occurs in metals. It occurs when a member is subjected to many cycles of tensile stress. According to Barltrop [29], these stresses tend not to exceed the materials yield stress, but must exceed the endurance limit stress. At first, a crack initiates in the material, followed by its propagation until the member finally experiences a brittle failure.

In his state of the art review in 2003, Schijve [30] gives a brief overview of how fatigue was first discovered. In the 19th century, engineers noticed that thin steel members were experiencing premature failure in their structures. This problem was observed at the tail end of the industrial revolution and occurred in steam engines, locomotives and pumps. Investigators noticed that the material did not show any sign of plastic yielding, but rather a sudden failure. This observation led Rankine [31] to falsely believe that the cracks were caused by a transition from a fibrous to a crystalline material. Today, the phenomenon is known as fatigue. A fatigue fracture for a plastic cross-section can be seen in Figure 2.1.

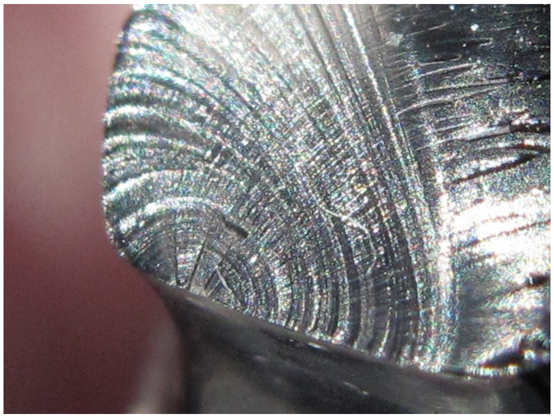


Figure 2.1: Fatigue fracture of plastic (ref. [32])

Knowledge about fatigue improved along with stronger microscopes and more cases to study. In 1870, Wöhler published a report stating his observations from studying railway axles. His findings are summarised by Schutz [33]. He noted that the stress amplitudes and mean tensile stress heavily influenced fatigue. Additionally, Wöhler formulated that components with low mean tensile stress may achieve an infinite design life. Wöhler suggests that the components subject to large stress amplitudes and tensile stress should be designed with higher safety factors to account for fatigue. The understanding of fatigue would develop, but it was not until the second half of the 20th century that modern design techniques started to take shape [30].

2.1.1 Stress-life curves

Stress-life (S-N) curves are used to determine the number of stress cycles that a component can take at a given stress range before failure due to fatigue damage. Wöhler conducted experiments to generate such curves; therefore, S-N curves are commonly referred to as Wöhler curves [30]. S-N curves are simple to use, but have limited applicability as they can only accurately predict the service life of members subjected to constant amplitude (CA) stress cycles. The stresses acting on CA components, such as machine parts, should not exceed the endurance limit stress.

S-N curves follow the Basquin equation. This relation is shown in Equation (2.1)

$$n = \sigma_0 \sigma_R^{-m} \tag{2.1}$$

where n is the number of cycles a member can withstand at stress range σ_R. Barltrop [29] calls σ_0 and m simply as material-dependent variables found through empirical tests. Identical members are subjected to CA stresses until failure. The number of cycles is counted and plotted against the stress amplitude. Repeating for a spectrum of different stress amplitudes allows researchers to make simplified curves of their results. As an example, steel has a slope of $m = 3$ for most cases and correction stress of $\sigma_0 \approx 10^{12}$. A S-N curve for steel is shown in Figure 2.2.

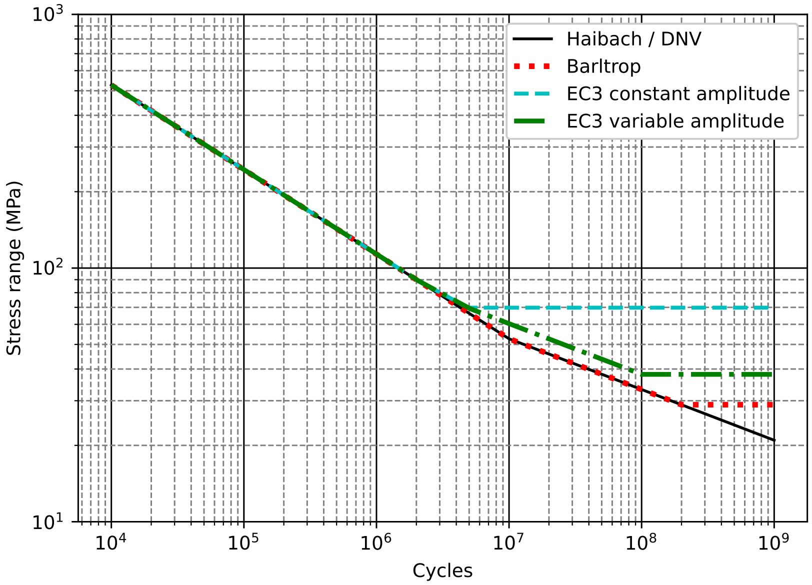


Figure 2.2: S-N curves of steel for different prediction models

The endurance limit alters the slope of the curve for lower stress ranges, but different standards have different ways of specifying it. Haibach proposed a general rule that the new slope should be $m = 2m + 1$ [30]. Det Norske Veritas (DNV) also uses a two-stage approach with the slope decreasing from $m = 3$ to $m = 5$ when the number of cycles exceeds either one or ten million, depending on the component and environmental conditions [34]. Barltrop [29] and Eurocode [35] operate with a three-stage approach. Barltrop states that at ten million cycles, the slope should increase by two ($m = m + 2$), and at 200 million, no fatigue damage is dealt ($m = \infty$). Eurocode uses the most complex method as it distinguishes between CA stresses and variable amplitude (VA) stresses. The slope factor is $m = 3$ for the first two million cycles. Next, the slope changes such that it hits the CA fatigue limit at five million cycles. Then for CA stress cases, the slope becomes constant at the CA fatigue limit. For VA stress cases, the slope becomes $m = 5$ until the cut-off limit at 100 million cycles. The cut-off limit is when even VA cases receive no fatigue damage. Suresh [36] explains that the three-stage model with an endurance limit is not valid for many high strength steels, which still receive some damage, even for small stress amplitudes.

S-N curves are essential for predicting the endurance of components that are subject to low-stress range and high-cycle fatigue loads. However, if such a curve was used for stresses approaching or exceeding the yield stress of steel, one might conclude that the component would last hundreds or thousands of cycles before failure. This is clearly wrong, as the member would sustain much more damage than the S-N curve suggests. This is because S-N is based on the assumption of purely elastic deformation. A common alternative to remedy this issue is strain-life curves, which are designed for low-cycle fatigue. Here a constant strain is applied as opposed to constant stress. As a result, strain-life tests give more accurate data on how a component reacts to stresses that would cause yielding and plastic deformation [30]. Lalanne [37] only considers S-N curves as applicable for stress ranges giving failure for more than 10^5 cycles.

Another shortcoming of S-N curves is that not all stress ranges are equal. Fatigue damage is primarily caused by cyclic tension. The damage must be corrected such that the member sustains more damage for higher mean stresses. The Goodman diagram [38] adjusts the S-N curve to account for varying mean stress. Smith [39] later modified the formula to make it easier to implement. Smith's Goodman correction is given as:

$$\sigma_R = \sigma_r \left(1 - \frac{\sigma_m}{\sigma_u}\right) \tag{2.2}$$

where σ_R is the stress range amplitude, σ_r is the endurance limit, σ_m is the mean stress, and σ_u is the ultimate tensile strength.

2.1.2 Miner's rule

According to Barltrop [29], offshore wind turbines are rarely subjected to CA loads, the type that the S-N curves are designed for. Instead, the environmental loading from wind and waves applies VA loads on the various turbine and tower components. Therefore, a method for calculating fatigue damage for a spectrum of stress ranges is required. Additionally, one must estimate the probability that a stress range might occur.

To solve these issues, one must change the interpretation of fatigue damage from "number of cycles until failure" to "damage per cycle". A linear interpretation of damage was proposed by Palmgren in 1924 and later refined by Miner in 1945 [37,40]. By employing the linear damage assumption, one may calculate the damage done by a single stress cycle as the inverse of the total number of cycles the member could withstand prior to failure. The total damage a member has received is the sum of all preceding damage dealt to the member by the various stress cycles. Miner's model predicts a failure when the sum exceeds 1. Barltrop [29] expresses this with the following equation:

$$D = \sum_{g=1}^{G} D_g = \sum_{g=1}^{G} \frac{1}{N_g} \tag{2.3}$$

where D is the total damage, D_g is the damage for one cycle, G is the total number of cycles, and N_g is the total number of cycles a member could withstand at stress range number g.

A common way to reduce the computing power needed to perform fatigue tests is to gather the stress ranges in bins. This modification reduces the number of summations from the total number of considered load cycles to the total number of bins. This gives us the Palmgren-Miner rule (referred to as Miner's rule):

$$D = \sum_{k=1}^{K} D_k = \sum_{k=1}^{K} \frac{n_k}{N_k} \tag{2.4}$$

where K is the total number of bins, n_k is the total number of cycles in bin k, and N_k is the total number of cycles the member can withstand at the mean stress range in bin k.

2.1.3 Rainflow counting

This section presents some stress range counting algorithms that Barltrop [29] and Lalanne [37] describe in their respective books.

To estimate the stress ranges a component is subject to, one might intuitively think to pair each crest with the following trough. However, this method will result in high-frequency noise filling the stress ranges with small values. As a result, the fatigue will be highly underestimated. An upper bound method is to pair similarly sized crests with troughs. Here, the lower frequency stress ranges with higher amplitudes are accounted for. However, the higher frequencies will cause an abundance of larger stress ranges as they increase the number of peaks at the crests and troughs.

The issue with the two methods described is that the high-frequency oscillations interrupt the stress series and cannot be accurately counted. A potential solution to the problem is to set a threshold for stress range amplitudes, thus ignoring the stress ranges stemming from high frequencies. There are numerous other peak counting methods, such as only counting the largest peak per mean-crossing or only counting peaks above a certain stress level.

The most common stress range counting algorithm is the rainflow method (or pagoda roof method), developed by Matsuishi and Endo in 1968. The name comes from rotating a time series of stresses by 90 degrees and imagining rain dripping down the time series like rain running down a pagoda roof. It considers first the largest stress ranges, and then the smaller ones are superimposed on the larger ones.

A rainflow counting method may be implemented with five simple steps:

1. Start stress range at a peak and follow rainflow down the time series.
2. Stop rainflow if path crosses with rainflow from a higher peak.
3. Stop rainflow if path crosses with rainflow from an earlier peak.
4. Stop rainflow if it reaches the end of the time history.
5. Repeat from step 1. with valleys instead of peaks.

A visualisation of the rainflow counting method for peaks is shown in Figure 2.3. Here it can be observed how some of the smaller half-cycles are stopped short of the next peak, allowing larger half-cycles to retain their continuity and give a more accurate stress range.

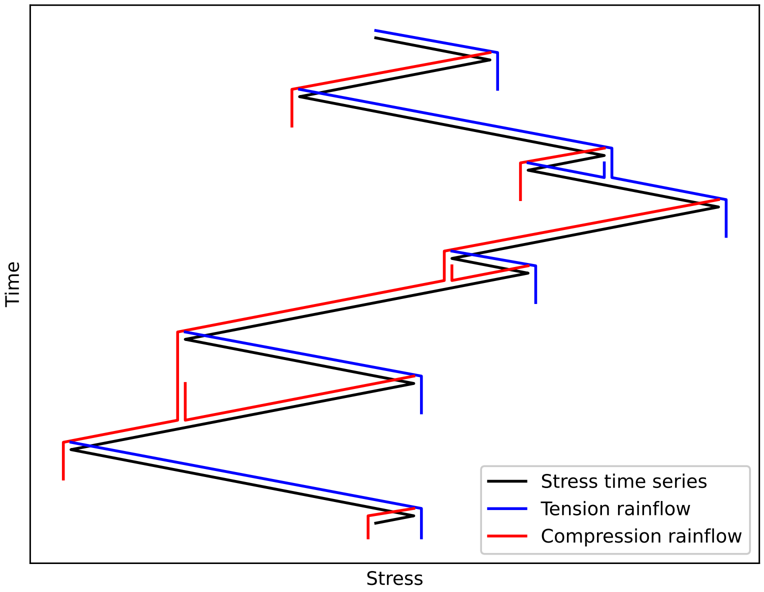


Figure 2.3: Rainflow counting method for tension and compression peaks

2.1.4 Fatigue in OWTs

OWTs are moderately complex structures with many different parts that experience different loads. The turbine blades are much thinner than the tower base, but they are primarily affected by the wind load and not the tower rocking, such as the base. Therefore, it is important to check different areas for fatigue damage to determine the endurance of each piece. Wind turbines with certain substructure types will also have parts that are more susceptible to fatigue. Monopile and gravity base design experience the largest stress ranges at the base where the largest overturning moment is experienced. Jacket and tripod designs have small members that may experience more stress than some larger members with larger forces acting on them. In addition, they might also be subject to high-frequency oscillations that may not individually deal much damage, but occur at such a rate that the damage accumulates. Michel et al. [26] highlights the importance of analysing different material types. For instance, the concrete foundation of gravity bases have different properties than the steel tower sitting atop it.

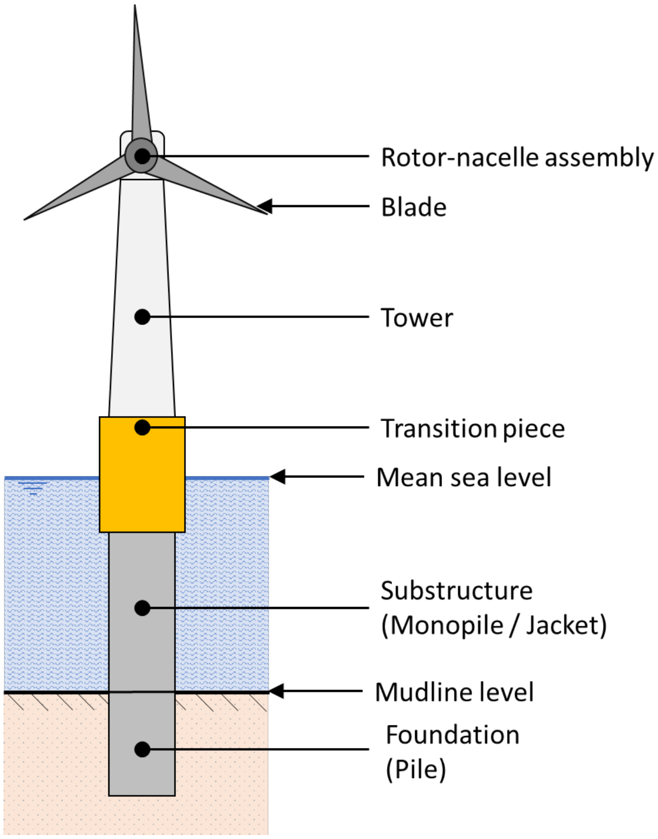


Figure 2.4: Various parts a generic OWT

Karimirad [41] highlights in his book the importance of checking the transition piece (TP). Although the TP is not subject to the largest forces, it may be a critical piece regarding fatigue. The grout connecting the TP to the substructure driven into the seabed is especially vulnerable for monopiles. This is due to the bending moments from aerodynamic thrust wearing down the cement mixture. TPs for jacket substructures will also receive significant fatigue loads from aerodynamic thrust in addition to shear forces generated by the ocean waves.

Standards require several design load condition (DLC)s to be checked when performing a fatigue analysis of an OWT [34,42]. Critical components must be checked for fatigue damage when the turbine is in a power-producing state, subject to moderate winds. This must be done both with and without ice formation on the substructure. In addition, fatigue controls must be carried out for several fault conditions that may occur during power production. Lastly, fatigue damage during start-up, shut-down, and idling must also be accounted for. Overall, since fatigue is a long term effect, only common occurrences are checked. This is despite large stress amplitudes causing more damage during a single cycle. It is just not economical to check the extreme cases as, overall, they do not contribute enough to the OWT durability.

2.2 Environmental loads

2.2.1 Wind loads

Wind is caused by varying air pressure. A place with a low density of air particles will be filled the area with air particles from a place with higher pressure, thus causing the wind to move from the high-pressure area to the low-pressure area. The pressure difference is caused by varying heating of the air molecules. Hot molecules rise and create lower pressure at the earth's surface, and cold air rushes to take its place.

According to Haddara and Guedes Soares [43], wind loads are not as impactful as wave loads for offshore structures. This is despite the smoother surface of the water in comparison to onshore structures, which are often rougher due to the wind-breaking effect of the earth's surface. OWTs are more subject to wind than other marine structures, such as platforms and ships, due to their lightweight design, tall and slender profile, and large blades at the top capturing wind and producing a significant overturning force at the base. In addition to the wind load produced by the mean wind velocity, Baniotopoulos [44] states that OWTs must also account for vortex excitation, ovalling, galloping and interference effects.

The wind's velocity, and therefore energy, increases with height above obstacles. This is due to the friction between the air and the surface it is in contact with. Eurocode 1 [45] employs a roughness parameter (z_0) to account for the terrain type and how much energy the surrounding area absorbs. Unlike in built areas, there are no obstacles to syphon the wind energy in the open sea. In an area with low vegetation and spread obstacles the roughness parameter would be 0.05 m, while at sea Eurocode suggests a value of 0.003 m. The base wind speed at height z is modified by the factor c_r as shown in Equation (2.5).

$$c_r(z) = k_r \ln\frac{z}{z_0} = 0.19\left(\frac{z_0}{z_{0,II}}\right)^{0.07}\ln\frac{z}{z_0} \tag{2.5}$$

$z_{0,II}$ is the roughness parameter for terrain category II which is for low vegetation. DNV's wind turbine design standards [34, 42] state that a roughness parameter of 0.003 m is on the high end and that it may be as low as 0.0001 m. An implicit equation should be solved to determine the value of z_0. It is important to note that the roughness for wind turbines is primarily used to determine turbulence parameters, not mean wind velocity.

The wind velocity fluctuation frequency content has two concentrations of frequencies with a spectral gap separating them. This can be seen in the Van der Hoven spectrum shown in Figure 2.5. Vorpahl et al. [46] state that this quality of wind loading is essential for the design procedures of structures influenced by wind. The low-frequency content is described by long-term wind statistics such as mean wind velocity and direction. The period of the wind is over two hours, with peaks at around twelve hours and four days. Data collection on long term wind statistics shows that a Weibull distribution may approximate the wind velocity. However, it is evident from data by Fischer et al. [47] that the direction cannot be decided that easily as it is wind velocity-dependent. Therefore, without site-specific data, one must conservatively assume that the wind acts in the least favourable direction. High-frequency wind content has a period of less than ten minutes. The load prediction model from DNV, due to the spectral gap, has simplified the wind parameters [34]. For a ten minute period, the wind energy is considered stationary. In practice, the mean wind velocity and the standard deviation of the fluctuating turbulence remain constant over ten minutes.

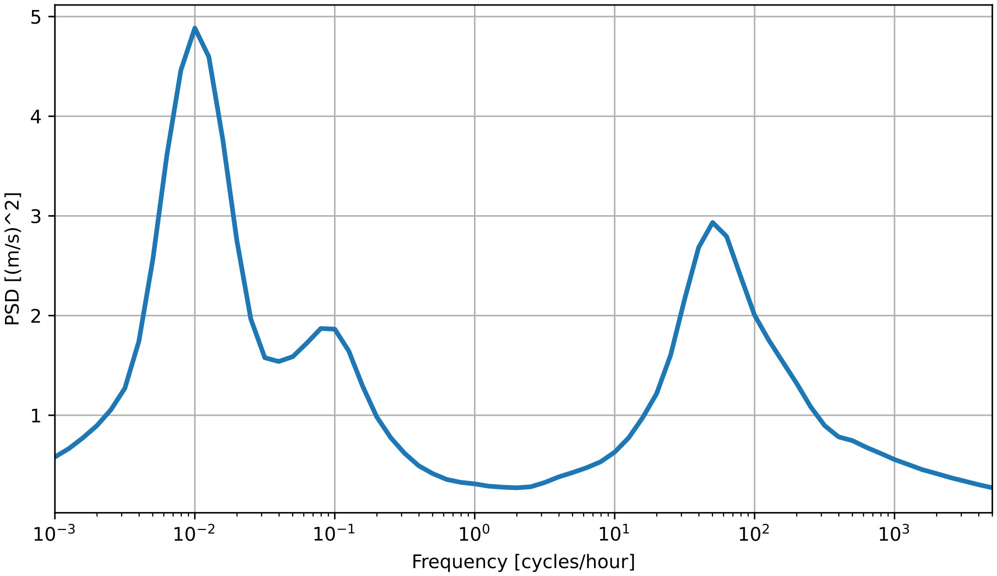


Figure 2.5: Wind velocity spectrum. Reproduced from Van der Hoven [48]

One of the most common turbulence spectrums is the Kaimal spectrum [49]. Kaimal and his co-authors proposed a general formula for the velocity spectra of wind in 1972. The original equation has been modified in newer standards to suit the desired application. The Norwegian Public Roads Administration [50] recommends a Kaimal spectrum for use when designing bridges subject to significant dynamic response due to wind load. This applies to bridges with eigenfrequencies shorter than 0.5 Hz, longer spans than 300 m or both. The International Electrotechnical Commission (IEC) [51], which is often used for OWT design, gives a modified Kaimal spectrum that fulfils their turbulence requirements for wind turbines. It is given by Equation (2.6).

$$\frac{fS_k(f)}{\sigma_k^2} = \frac{4fL_k/V_{hub}}{(1+6fL_k/V_{hub})^{5/3}} \tag{2.6}$$

where S_k is the one-sided velocity spectrum, σ_k is the velocity standard deviation, f is the frequency (Hz), V_{hub} is the mean wind velocity at the hub height, and L_k is the integral scale. The subscript k denotes the direction of the velocity component. The wind has the biggest standard deviation σ_k and integral scale L_k when moving longitudinally to the mean wind direction. σ_k and L_k are the smallest for wind moving in the vertical direction. That means more turbulence is experienced in the directions most aligned with the mean wind direction. The IEC model is repeated in DNV's old standard for OWTs [34] with modified standard deviation and integral scale. DNV's most recent wind turbine load standard [42] forgoes its own model in favour of using the power spectra models in IEC 61400-1 [51] or other more accurate or conservative models. It should also fulfil the Kolmogorov five-third law that states that the energy within the inertial subrange decreases by five-thirds on a logarithmic scale. This means that the spectral density in the longitudinal direction must approach the following equation for higher frequencies.

$$S_1(f) = 0.05\sigma_1^2(\Lambda_1/V_{hub})^{-2/3}f^{-5/3} \tag{2.7}$$

It is noted that the turbulence scale (Λ) is a parameter used in determining the integral scale (L_k) from Equation (2.6).

IEC 61400-1 [51] requires an OWT to be designed to withstand several different DLCs. The most common loading situation is during normal operating conditions when the turbine is subjected to a moderate wind force that it can use to generate power. The normal turbulence model (NTM) is utilised during this condition. The turbine must be parked and idling during extreme events to avoid excessive structural deterioration. Extreme events may be characterised by high mean wind velocities, strong gusts, larger turbulence standard deviation, and large and sudden direction changes.

2.2.1.1 Weibull distribution

Weibull described a cumulative distribution function in 1951 [52]. The probability density function (PDF) for a vanishing value of zero is given in Equation (2.8).

$$p(x) = \frac{k}{\lambda_0}\left(\frac{x}{\lambda_0}\right)^{k-1} \exp\left[-\left(\frac{x}{\lambda_0}\right)^{k}\right] \tag{2.8}$$

p is the probability of occurrence, k is the shape parameter and λ_0 is the scale parameter. The shape parameter decides the shape of the distribution peak. To alter the placement of the peak, the scale parameter must be changed. Figure 2.6 gives a better overview of the impact of each parameter. When fitting a Weibull distribution to wind velocity bins, it is necessary to multiply the probability by the distance between the bins. This is such that the probability for the sum of all the bins remains 1.00.

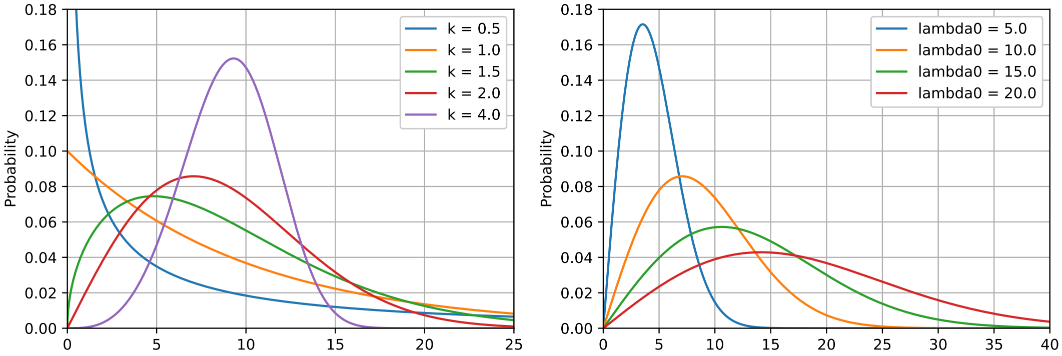


Figure 2.6: Weibull distribution for various shape k and scale parameters λ_0

2.2.2 Wave loads

On the surface, waves might seem like simple sine waves moving across the ocean. However, wave theories are complex descriptions of how the water surface behaves. According to Liu [53], numerous developments were made in the field of wave kinematics throughout the 19th century. Airy proposed a simple model that could describe wave motion for small waves in deep water locations. The following equation describes linear (Airy) wave theory according to DNV J101 [34].

$$\eta(x,y,t) = \frac{H}{2}\cos\Theta = \frac{H}{2}\cos\left[k\left(x\cos\left(\beta\right) + y\sin\left(\beta\right) - \omega t\right)\right] \tag{2.9}$$

where η is wave surface above mean sea level as a function of space and time, H is the wave height, and Θ is the phase angle. It must be noted that wave height is the vertical distance between wave trough and crest. This means that the wave height is double the size of the amplitude of the wave function. The parameters in the phase angle are the angular wavenumber (k), angular frequency (ω) and direction of propagation (β).

Linear wave theory is applicable to deep water with small wave heights. However, for shallower waters and bigger waves, it is observed that the crest of the wave is larger than the trough when compared to the still water level. Therefore, higher-order descriptions of the waves are required to depict their behaviour accurately. Stokes wave theory builds upon Airys theory to increase the order of the equation. A higher-order wave theory allows the equation to be valid for a larger spectrum of situations. DNV [34] allows for the fifth-order Stokes wave theory to be used for deep waters where the water is deeper than 15% of the wavelength. Cnoidal wave theory may be utilised for even shallower waters as it is better than Stokes theory for modelling the sharper peaks experienced in shallow waters. Lastly, there is the "Stream function" wave theory that numerically describes the waves and has the broadest use area provided that the correct number of terms have been included in the calculation [54]. These higher-order wave theories have not been described in detail because as more terms are included, the more complex the equations become. Likewise, when modelling waves, one should use the most straightforward description that can accurately recreate the desired marine condition.

The most popular formulation of the frequency content of waves is the Joint North Sea Wave Project (JONSWAP) spectrum. The spectrum is dependent on wave frequency (f), peak spectral frequency (f_p), significant wave height (H_S) and spectral width (σ). Equation (2.10) shows the general formulation of the JONSWAP spectrum.

$$S(f) = \frac{5}{16} H_S^2 \frac{f_p^4}{f^5} (1 - 0.287 \ln \gamma) \exp\left(-\frac{5 f_p^4}{4 f^4}\right) \gamma^{\exp\left(-\frac{(f-f_p)^2}{2\sigma^2 f_p^2}\right)} \tag{2.10}$$

γ is the peak-enhancement parameter which decides how broad band the spectrum is. The peak-enhancement parameter in DNV ST0437 [42] may be as high as five, where the result would be a narrow banded wave loading. On the low end, when $\gamma = 1$, a special case of the JONSWAP spectrum known as the Pierson-Moskowitz (P-M) spectrum is established. The value of the peak-enhancement is dependent on the the peak spectral frequency and the significant wave height as shown in Equation (2.11). The equation is only valid for $\gamma \in [1, 5]$ with γ set to the closest valid γ value when the peak-enhancement goes outside the bounds. Generally the P-M spectrum is used for low energy waves since the wave height is low without the peak wave frequency being too high.

$$\gamma = \exp\left(5.75 - \frac{1.15}{f_p \sqrt{H_S}}\right) \tag{2.11}$$

Figure 2.7 shows a JONSWAP spectrum for moderate wave height and frequency. It shows the P-M value, the value according to Equation (2.11) and extreme load case value for the peak-enhancement factor.

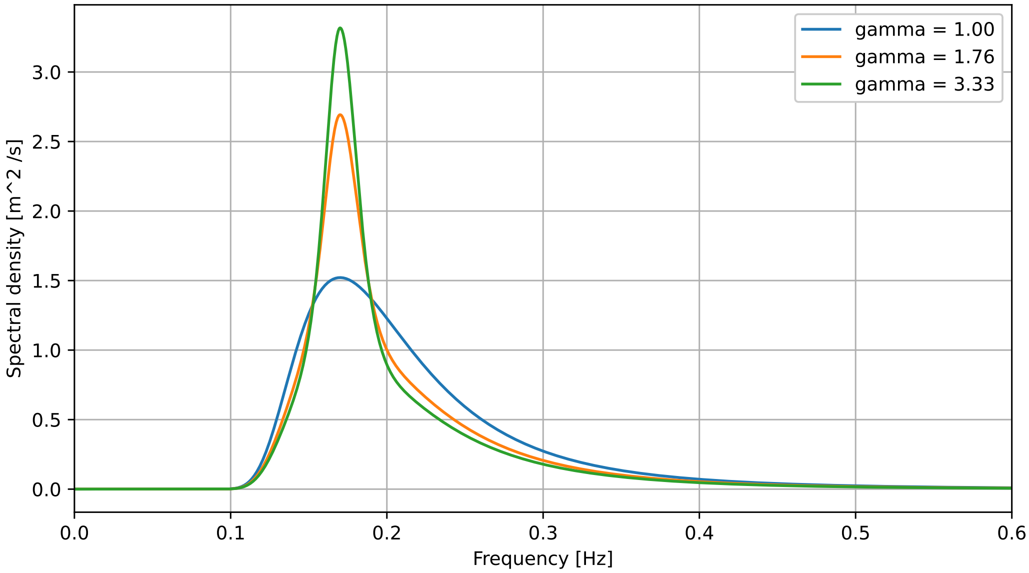


Figure 2.7: JONSWAP spectrum for various peak-enhancement factors.

2.3 Material models for soil

This section is based on Hopperstad and Børvik's compendium [55] and Abaqus documentation [56]

In everyday engineering it is common to assume that the material is linear elastic such that $\sigma = E\epsilon$ for materials such as steel, where σ is the stress, E Young's module and ϵ the strain. However, this assumption often fails for large strains, and a plastic model must be utilised.

For plastic 1D models the strain can be split into an elastic with subscript e and a plastic part with subscript p. Where the elastic part expresses the strain that will go back to zero if the part is relaxed. The plastic strain expresses the permanent deformation.

$$\epsilon = \epsilon_e + \epsilon_p \tag{2.12}$$

The stress is only dependent on the elastic strain and the elasticity modulus.

$$\sigma = E\epsilon_e \tag{2.13}$$

In an elastic plastic model the yield surface can be described by a function f

$$f(\sigma, R) = |\sigma| - (\sigma_0 + R) = 0 \tag{2.14}$$

where σ_0 is the initial yield capacity. While the work hardening R is a function of accumulated plastic strain p.

$$R = R(p) \tag{2.15}$$

It is important to notice that during yielding for an elastic-plastic material, the stress can only be on the yield surface and never on the outside.

A common work hardening functions is linear work hardening, where E_t is a material parameter

$$R = pE_t \tag{2.16}$$

Johnson-Cook hardening is an other used approach where A, B and n are material parameters. This material model is shown in Equation (2.17). By setting $A = 0$ Johnson-Cook reduces to the power-law.

$$R = (A + Bp^n) \tag{2.17}$$

When viewing the model in 3D, the yield surface becomes more complicated. Two common models that are able to describe the friction dependent yielding of soils are the Drucker-Prager yield criterion and the Mohr-Coulomb yield criterion.

2.3.1 Drucker-Prager

The Drucker-Prager yield criterion can be written as

$$f(\sigma) = \frac{\sqrt{3J_2} + \alpha I_\sigma}{1 + \alpha} - \sigma_Y = 0 \tag{2.18}$$

where J_2 is the second principal invariant of the stress deviator. With Einstein-notation it can be written:

$$J_2 = \frac{1}{2}\sigma_{ij}\sigma_{ij} - \frac{1}{6}\sigma_{kk}\sigma_{mm} = \frac{1}{2}(\sigma_{11}^2 + \sigma_{22}^2 + \sigma_{33}^2 + 2\sigma_{12}^2 + 2\sigma_{13}^2 + 2\sigma_{23}^2) - \frac{1}{6}(\sigma_{11} + \sigma_{22} + \sigma_{33})^2 \tag{2.19}$$

I_σ is 3 times the first invariant

$$I_\sigma = \sigma_{kk} = \sigma_{11} + \sigma_{22} + \sigma_{33} \tag{2.20}$$

σ_Y is the material's yield capacity and α is the materials pressure sensitivity. For $\alpha = 0$ Drucker-Prager is reduced to the well-known von Mises yield criterion.

2.3.2 Mohr-Coulomb

The Mohr-Coulomb yield criterion is given by

$$\frac{1}{2}(\sigma_1 - \sigma_3) + \frac{1}{2}(\sigma_1 + \sigma_3)sin(\phi) - \sigma_Y cos(\phi) = 0 \tag{2.21}$$

where ϕ is the friction angle. By setting the friction angle $\phi = 0$. Mohr-Coulomb is reduced to the Tresca yield criterion.

2.4 Basic structural dynamics

2.4.1 Equation of motion

Consider a single degrees of freedom (SDOF) system. With a mass m, a stiffens k, a dash-pot damper c, a time-dependent load $p(t)$ and a displacement r. From equilibrium the equation of motion becomes

$$m\ddot{r} + c\dot{r} + kr = p(t) \tag{2.22}$$

The system has a load frequency ω and a natural frequency $\omega_n = \sqrt{\frac{k}{m}}$ The ratio between the load frequency and natural frequency $\beta = \frac{\omega}{\omega_n}$. The damping ratio $\xi = \frac{c}{2m\omega_n}$ The solution of the system can typically be split into a particular (steady state) solution r_p and a homogeneous solution r_h.

$$r = r_p + r_h \tag{2.23}$$

Over time the homogeneous solution is damped out. This thesis is primary interested in the long time response and thus only utilises the particular solution.

$$r = r_p = \frac{p(t)}{k\sqrt{(1-\beta^2)^2 + (2\xi\beta)^2}} \tag{2.24}$$

For a multiple degrees of freedom (MDOF) system the equation of motion is

$$[M]\{\ddot{r}\} + [C]\{\dot{r}\} + [K]\{r\} = \{P(t)\} \tag{2.25}$$

where $[M]$ is the mass matrix, $\{r\}$ is the response vector, $[C]$ is the damping matrix, $[K]$ is the stiffness matrix and $\{P(t)\}$ is the load vector

2.4.2 Modal analysis

One technique to solve a MDOF dynamic system problems is modal analysis. Consider solving the eigenvalue problem

$$det([K] - \{w_n^2\}[M]) = 0 \tag{2.26}$$

where $\{w_n^2\}$ is the eigenvalues. Use $[\Phi]$ to denote the matrix holding each eigenvector $\{\phi\}$.

Now move the system into the modal domain marked by $\sim$.

$$[\tilde{K}] = [\Phi]^T[K][\Phi] \tag{2.27}$$

$$[\tilde{M}] = [\Phi]^T[M][\Phi] \tag{2.28}$$

$$[\tilde{C}] = [\Phi]^T[C][\Phi] \tag{2.29}$$

$$\{\tilde{P}\} = [\Phi]^T\{P\} \tag{2.30}$$

where $[\tilde{K}]$, $[\tilde{M}]$, $[\tilde{C}]$ all are diagonal matrices. For $[\tilde{C}]$ this only holds if [C] has some particular properties, more on this in Section 2.4.3.

The particular solution of the system can now be solved for each mode i.

$$\{\tilde{r}\}_i = \{\tilde{r_p}\}_i = \frac{\{\tilde{P}\}_i(t)}{[\tilde{K}]_{ii}\sqrt{(1-\beta_i^2)^2 + (2\xi_i\beta_i)^2}} \tag{2.31}$$

The total physical response can then be calculated by Equation (2.32). In order to get each mode's contribution to the physical response, Equation (2.33) can be utilised.

$$\{r\} = [\Phi]\{\tilde{r}\} \tag{2.32}$$

$$\{r_i\} = \{\phi\}\{\tilde{r}\}_i \tag{2.33}$$

2.4.3 Damping

Damping is a process under oscillations that reduce or prevent the system's oscillations. The damping is, in effect, a measure of energy loss: potential and kinetic energy are dissipated to other forms of energy that have a negligible effect on the oscillations of the system. This can be from losing energy to surrounding media like fluid, losing energy as heat due to internal friction or other processes where energy is dissipated from the system.

The processes where the energy is dissipated can be extremely complex, and idealised solutions must be utilised to get a system that can be solved. The typical solution is to start with damping proportional to the velocity. This was also done in Section 2.4.1 and gives a modal damping ratio of $\xi = \frac{c}{2m\omega_n}$. It is important to notice that this approach is used because of its simple mathematical description rather than an approach that accurately describes the underlying physical processes causing energy dissipation.

In multiple degree of freedom systems it is a common approach to let $[C]$ be proportional to $[K]$, $[M]$ or both as this makes $[\tilde{C}]$ diagonal. A much used approach is the Rayleigh damping.

$$[C] = \alpha[M] + \beta[K] \tag{2.34}$$

$[\tilde{C}]$ is then a diagonal matrix and mode i have damping ratio ξ

$$\xi_{=}\alpha\frac{1}{2\omega_n} + \beta\frac{\omega_n}{2} \tag{2.35}$$

A high α will primarily damp out low-frequency modes, while β will primarily damp out high-frequency modes.

Another much-used definition for damping is D defined as $\frac{\eta}{2}$ where η is the share of total energy dissipated per radian for a cyclic response. Even though D is often referred to as hysteric damping, the D used in this thesis is the equivalent of viscous damping due to hysteric damping. Such that Equation (2.36) holds.

$$\frac{\eta}{2} = D = \xi \tag{2.36}$$

2.4.4 Craig-Bampton

The Craig-Bampton (C-B) method is a form of substructure coupling that lets the analyst split up a mesh and carry over most of the information from the reduced system [57]. It was developed to lessen the strain on computers and allow several engineering groups to work simultaneously on the same structure. This method improves on the coupling technique by Hurty [58]. The C-B method described in this section is based on the description used in FAST [6]. The structure is divided into several substructures with boundary nodes connecting them. Within each substructure are internal nodes intended to be eliminated through C-B reduction.

The equation of motion, such as (2.25), is partitioned such that the boundary (B) and interior (I) nodes are separated. This is shown in Figure 2.8 and Equation (2.37).

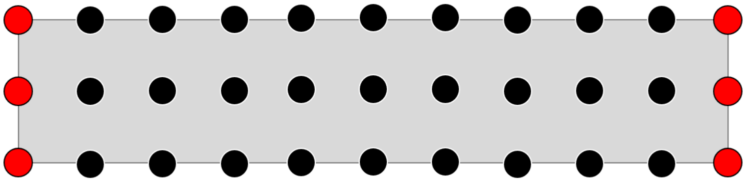


Figure 2.8: Boundary (red) and interior (black) nodes for a mesh with restrained sides. There are multiple ways to choose which nodes belong in each category depending on the structure surrounding the mesh.

$$\begin{bmatrix} M_{BB} & M_{BI} \\ M_{IB} & M_{II} \end{bmatrix} \begin{Bmatrix} \ddot{r}_B \\ \ddot{r}_I \end{Bmatrix} + \begin{bmatrix} C_{BB} & C_{BI} \\ C_{IB} & C_{II} \end{bmatrix} \begin{Bmatrix} \dot{r}_B \\ \dot{r}_I \end{Bmatrix} + \begin{bmatrix} K_{BB} & K_{BI} \\ K_{IB} & K_{II} \end{bmatrix} \begin{Bmatrix} r_B \\ r_I \end{Bmatrix} = \begin{Bmatrix} P_B \\ P_I \end{Bmatrix} \tag{2.37}$$

A coordinate transformation matrix, $[\alpha]$, is established to transform the system from physical degree of freedom (DOF)s to a modal mass-normalised DOFs without unwanted internal modes. The Guyan modes may be established by assuming a homogeneous, static version of the equation of motion. The bottom partition gives:

$$[K_{IB}]\{r_B\} + [K_{II}]\{r_I\} = 0 \tag{2.38}$$

Rearranging the expression with respect to $\{r_I\}$ yields:

$$\{r_I\} = -[K_{II}]^{-1}[K_{IB}]\{r_B\} = [\Phi_B]\{r_B\} \tag{2.39}$$

Next, the modal transformation matrix for the boundary modes is established. This transformation is found by solving the following eigenvalue problem:

$$[K_{II}][\Phi_I] = \omega^2[M_{II}][\Phi_I] \tag{2.40}$$

$[\Phi_I]$ is a mass normalised matrix. Unwanted DOFs are eliminated to the point where only N modes remain. Thus, leaving an $I \times N$ matrix that gives a relation similar those from Equation (2.27) and (2.28):

$$[\Phi_N]^T[M_{II}][\Phi_N] = [I] \tag{2.41}$$

$$[\Phi_N]^T[K_{II}][\Phi_N] = [K_{NN}] = [\Omega_N]^2 \tag{2.42}$$

The generalised modal DOFs are then established using the modal transformation matrices:

$$\{r_L\} = [\Phi_B]\{r_B\} + [\Phi_N]\{q_N\} \tag{2.43}$$

where q_N is the modal DOFs. This relation allows for the coordinate transformation matrix to be set up:

$$\begin{Bmatrix} r_B \\ r_I \end{Bmatrix} = [\alpha] \begin{Bmatrix} r_B \\ q_N \end{Bmatrix} = \begin{bmatrix} I & 0 \\ \Phi_B & \Phi_N \end{bmatrix} \begin{Bmatrix} r_B \\ q_N \end{Bmatrix} \tag{2.44}$$

Premultiplying each side of the equation of motion with the transposed coordinate transformation matrix gives the following equation:

$$[\alpha]^T \begin{bmatrix} M_{BB} & M_{BI} \\ M_{IB} & M_{II} \end{bmatrix} [\alpha] \begin{Bmatrix} \ddot{r}_B \\ \ddot{q}_N \end{Bmatrix} + [\alpha]^T \begin{bmatrix} C_{BB} & C_{BI} \\ C_{IB} & C_{II} \end{bmatrix} [\alpha] \begin{Bmatrix} \dot{r}_B \\ \dot{q}_N \end{Bmatrix} + [\alpha]^T \begin{bmatrix} K_{BB} & K_{BI} \\ K_{IB} & K_{II} \end{bmatrix} [\alpha] \begin{Bmatrix} r_B \\ q_N \end{Bmatrix} = [\alpha]^T \begin{Bmatrix} P_B \\ P_I \end{Bmatrix} \tag{2.45}$$

Multiplying terms and simplifying the equation of motion yields:

$$\begin{bmatrix} \bar{M}_{BB} & \bar{M}_{BN} \\ \bar{M}_{NB} & I \end{bmatrix} \begin{Bmatrix} \ddot{r}_B \\ \ddot{q}_N \end{Bmatrix} + \begin{bmatrix} \bar{C}_{BB} & \bar{C}_{BN} \\ \bar{C}_{NB} & \bar{C}_{NN} \end{bmatrix} \begin{Bmatrix} \dot{r}_B \\ \dot{q}_N \end{Bmatrix} + \begin{bmatrix} \bar{K}_{BB} & 0 \\ 0 & \Omega_N^2 \end{bmatrix} \begin{Bmatrix} r_B \\ q_N \end{Bmatrix} = \begin{Bmatrix} \bar{P}_B \\ \bar{P}_N \end{Bmatrix} \tag{2.46}$$

2.5 Soil damping

The damping of the foundation can be split into three parts.

Hysteric damping also known as material damping. The damping is caused by plastic dissipation of the soil during cyclic loading.

Radiation damping is the damping that is caused by elastic waves spreading away from the pile. This damping is negligible for frequencies smaller than 1 Hz [59]. As the response of monopile turbines typically has a low frequency, radiation damping is not taken into account in the calculations of the monopile.

Pore water dissipation damping is damping caused by water pores surrounding the piles. As for radiation damping, this type of damping is negligible for frequencies smaller than 1 Hz [59]. This damping is not taken into account in this thesis.

2.5.1 Estimation of hysteric soil damping

Two different methods for calculating hysteric soil damping are described: One method based on the dynamic energy of each mode (Method 1) and one method based on the energy related to the unit displacement in each mode (Method 2).

Method 1 is a physics-based complex approach and gives a complete guide on how to calculate modal soil damping. Method 2 is a simpler mathematical formulation and can be used to calculate the relationship between the foundation damping and the overall modal damping.

Only hysteric damping is taken into account. The hysteric damping of materials occurs due to plastic dissipation during cyclic loading. W_L is the energy loss per cycle. The relationship between shear stress τ, shear strain γ and W_L for one cycle is explained by Figure 2.9. Therefore, W_L can be calculated by knowing the maximum strain in the cycle. Unfortunately, the stress-strain curves in FE software are not accurate for cyclic loading. It is, therefore, necessary to post-process each element in the FE-analysis to get accurate results.

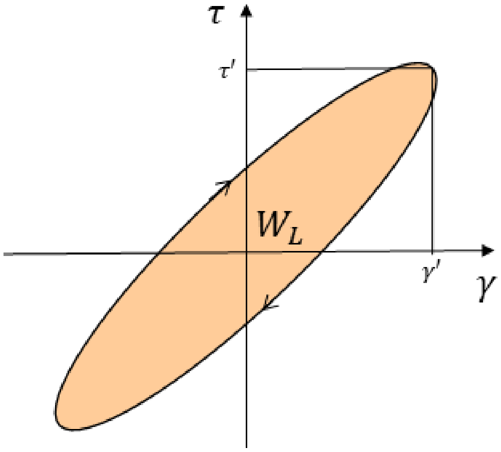


Figure 2.9: Visualisation of cyclic energy loss associated with a level of stress and strain. W_L is the shaded region

The damping D is related to the loss factor η, that is the share of the system's energy that is dissipated per radian. The relationship between D, η, and W_L is given by Equation (2.47).

$$D = \frac{\eta}{2} = \frac{W_L}{4\pi\Pi} \tag{2.47}$$

where Π is the energy in the system of interest.

In practice, for soil elements, the relationship between the strains and damping ratio for each element D_e are calculated by utilising shear strain. The relationship is given in Figure 2.10.

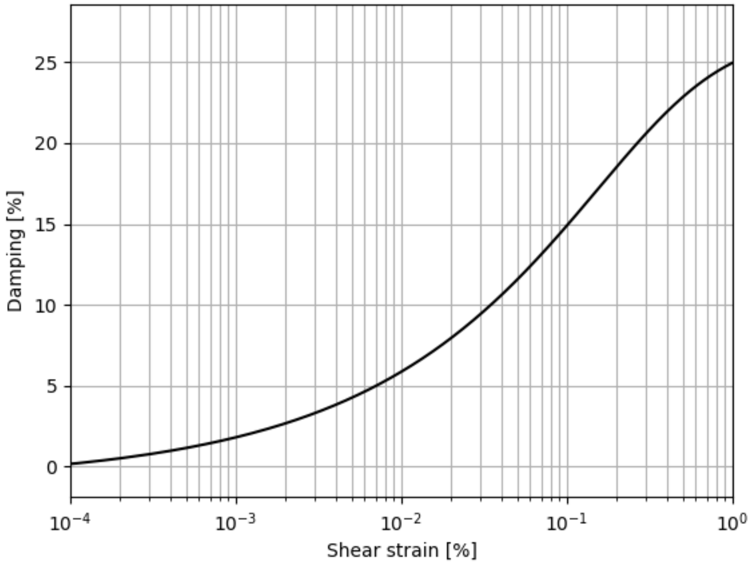


Figure 2.10: The relationship between damping ratio for a soil element and shear strain. Reproduced from Seed and Idris, ref. [60]

The energy loss estimation is calculated for each element and then summed to calculate the total energy loss. Each element has an energy loss per cycle A_L

$$A_L = 2\pi V[\gamma_1\tau_1 D_e(\gamma_1) + \gamma_2\tau_2 D_e(\gamma_2) + \gamma_3\tau_3 D_e(\gamma_3)] \tag{2.48}$$

where γ is the the shear strain from the strain vector (Equation 2.50) and D_e is the damping as a function of γ. τ is the shear stress from the stress vector (Equation: 2.49) and V the volume of each element.

$$\{\sigma\} = \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{Bmatrix} \tag{2.49}$$

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{Bmatrix} \tag{2.50}$$

Many different damping curves for soil are available. In this thesis, D_e is calculated for the mean limit of sand from Seed and Idriss ref. [60]. The equation is given in Equation (2.51) and shown in Figure 2.10. Both input and output of the formula are in percent.

$$D_e(\%) = [(12.6e^{0.9153log_{10}\gamma})(log_{10}(\gamma + 1.797))^{-0.8939}] - 0.9283 \tag{2.51}$$

The absorbed energy per cycle for the entire model can be calculated as the sum of the absorbed energy in each element.

$$W_L = \sum_e A_L \tag{2.52}$$

The damping from soil can now be calculated for the system of interest with a level of energy Π.

$$D = \frac{W_L}{4\pi\Pi} \tag{2.53}$$

2.5.1.1 Method 1

In method 1 the damping is calculated relative to the energy of each mode.

The system during harmonic oscillation will at any given time have a total energy Π_T consisting of the kinetic energy Π_K and the the potential energy Π_P.

$$\Pi_T = \Pi_K + \Pi_P. \tag{2.54}$$

The potential energy in turn consists of the two parts. The strain energy Π_U and the load potential Π_R

$$\Pi_P = \Pi_U + \Pi_R \tag{2.55}$$

$$\Pi_T = \Pi_K + \Pi_U + \Pi_R \tag{2.56}$$

Now move the system into the modal domain marked by $\sim$. In modal analysis, for mode i, the strain energy Π_U can be written as

$$\Pi_U = \frac{1}{2}\lfloor\tilde{r}\rfloor_i^2[\tilde{K}]_{ii} \tag{2.57}$$

This expression also includes the strain energy due to static loading, which is not interesting when calculating the global damping ratio. The energy of interest is the energy of the dynamic part of the system. Therefore the system is split into a dynamic part with subscript D and a static part with subscript S.

$$\Pi_T = \Pi_{TD} + \Pi_{TS} \tag{2.58}$$

For mode i the dynamic energy is

$$\Pi_{Di} = \Pi_{UDi} + \Pi_{Ki} = \frac{1}{2}\lfloor\tilde{r}\rfloor_i^{*2}[\tilde{K}]_{ii} + \Pi_{Ki} \tag{2.59}$$

where the normalised response is

$$\lfloor\tilde{r}\rfloor_i^* = \lfloor\tilde{r}\rfloor_i - \frac{\sum_{t=1}^{N}\lfloor\tilde{r}\rfloor_i}{N} \tag{2.60}$$

Consider the system when $\lfloor\tilde{r}\rfloor_i^*$ is at at local maximum. The system will at that point have 0 kinetic energy. So the dynamic energy for each mode is

$$\Pi_{Di} = \Pi_{UDi} = \frac{1}{2}\lfloor\tilde{r}\rfloor_i^{*2}[\tilde{K}]_{ii} \tag{2.61}$$

The system's energy loss per cycle is also split into a dynamic and a static part. For each mode, the dynamic average peak response height is calculated in the physical domain. In other words, the average of all peak response heights, for a single mode in the dynamic part of the system in the physical domain. In addition, the static response of the entire system is calculated. The energy loss of $W_{Lstatic}$ and $W_{Lstatic+dynamic}$ is calculated as shown in Figure 2.11. Then, the dynamic energy loss is calculated by Equation (2.62).

$$W_{Ldynamic} = W_{Lstatic+dynamic} - W_{Lstatic} \tag{2.62}$$

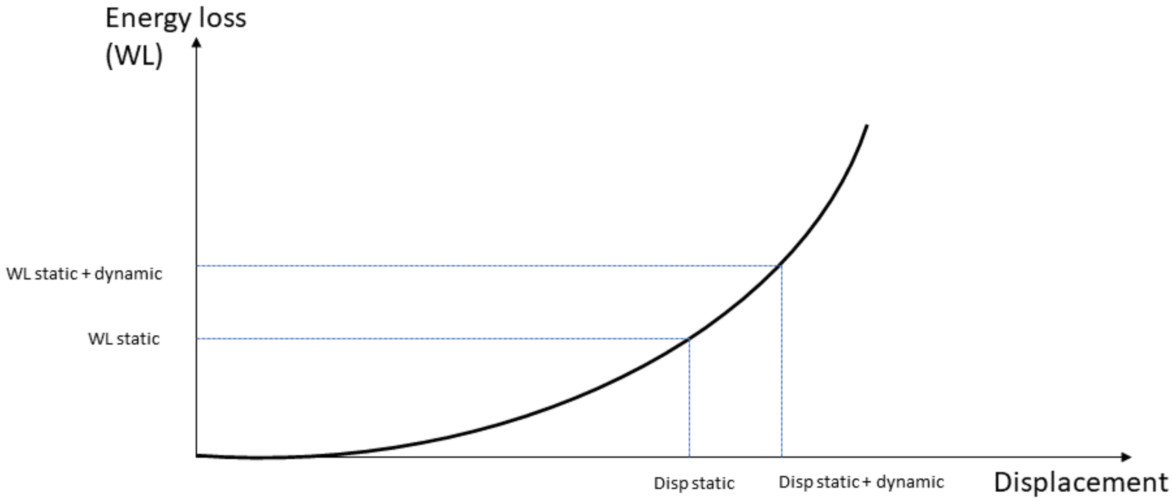


Figure 2.11: Calculations of dynamic and static energy loss

The global damping ratio D can now be calculated for each mode i

$$D_i = \frac{W_{Li}}{4\pi\Pi_{Di}} \tag{2.63}$$

where $W_{Li} = W_{L\,dynamic}$

2.5.1.2 Method 2

In method 2, the foundation damping D_f is assumed to be calculated first. The effect of each mode is then calculated from the mode shapes and the foundation damping.

Equation (2.64) can be expanded to modal damping. Consider having a unit displacement of each mode. The energy in the foundation Π_f and the energy of the entire structure Π are then given by Equation (2.65) and (2.66), respectively.

$$D = \frac{\Pi_f(4\pi D_f)}{4\pi\Pi} = \frac{\Pi_f D_f}{\Pi} \tag{2.64}$$

$$\Pi_f = \frac{1}{2}\{\phi_{fi}\}^T[K_f]\{\phi_{fi}\} \tag{2.65}$$

$$\Pi = \frac{1}{2}\{\phi_i\}^T[K]\{\phi_i\} \tag{2.66}$$

$\{\phi_{fi}\}$ is the mode-shape of mode i of the foundation, while $\{\phi_i\}$ is the mode-shape of mode i for the entire system.

By inserting Equation (2.65) and (2.66) into Equation (2.64) the overall damping for each mode D_i can be found.

$$D_i = \frac{\{\phi_{fi}\}^T[K_f]\{\phi_{fi}\}D_f}{\{\phi_i\}^T[K]\{\phi_i\}} \tag{2.67}$$

2.5.2 Estimation of radiation damping

Radiation damping is the result of waves generated from a vibrating foundation and propagating outwardly to the infinite soil domain. This represents a loss of energy which can be computed by special elastodynamic numerical models in the frequency domain. From such analyses, the radiation damping is computed from the imaginary part of the foundation impedance as a function of frequency from the expression $C = \frac{K_{imag}}{\omega}$ where K_{imag} is the imaginary part of the impedance and omega is the frequency of vibration. In this study, the foundation (pile) impedance was computed by PILES [61, 62]. PILES uses Green's functions in layered media together with the finite element model of the pile to compute the vibrations in a viscoelastic medium to vibrating piles. A Green's function is a mathematical expression for the response in an elastic medium due to a unit load at an arbitrary point in the medium.

The input data to PILES are the soil stratification with the shear wave velocity, mass density and Poisson's ratio in each layer, together with the geometrical and mechanical properties of the pile, namely, diameter, length, axial rigidity, EA, and bending rigidity, EI. PILES computes the stiffness and radiation damping in the horizontal and axial direction of pile vibration.

3. Abaqus model

The entire damping estimation process was performed in the Abaqus framework or based on files exported from the Abaqus framework.

The modelling was done in the Abaqus CAE module using the Python interface. The key parameters were loaded from CSV files copied from Excel, making the model partly parametric. The advantage of using the Python interface instead of the graphical CAE interface was increased modifiability, allowing the model to be easily changed.

The Abaqus model served primarily two purposes:

1. Generating the stiffness and mass matrix for the modal analysis and thus being the model foundation for the modal analysis.

2. Running an NFEA on the soil and pile part of the model to estimate stress and strains in the soil. In turn, the stress and strains are used to find energy dissipation due to soil plasticity.

The jacket structure and the soil profile were taken from Pedersen and Askheim [63] with some modifications as described in the following sections. The authors built the monopile Abaqus model based on an existing FAST model.

3.1 Soil profile

The same soil profile was used for both the jacket and the monopile. That being the soil profile from [63]. The soil was modelled with eight-node interpolated brick elements with reduced integration and hourglass control (C3D8R). The total height of the soil profile was 90 m with increasing Young modulus with depth spanning from 34.4 MPa to 572 MPa. The same was the case for the Yield stress spanning from 1.8 to 179.75 kPa. Details on the soil profile can be found in Table 3.2. The soil box was encastered at the bottom and had a locked horizontal position for the vertical sides. The material model was a perfect plastic Mohr-Coulomb model with a friction angle of 0 degrees, reducing the model to the Tresca model.

Table 3.2: Soil profile properties

Layer number	Bottom z-coordinate of layer [m]	Mass density [kg/m^3]	Young's modulus [Pa]	Poisson's ratio [-]	Depth [m]	Cohesion Yield Stress [kPa]
1	-2	1937	3.44E+07	0.3	2	1.80
2	-5.5	1937	6.66E+07	0.3	3.5	6.75
3	-9	1937	9.26E+07	0.3	3.5	13.1
4	-10	1937	1.06E+08	0.3	1	17.1
5	-15	1937	1.22E+08	0.3	5	22.5
6	-20	2039	1.92E+08	0.3	5	32.0
7	-22.5	2090	2.41E+08	0.3	2.5	39.6
8	-29	2141	3.00E+08	0.3	6.5	49.4
9	-34	2141	3.36E+08	0.3	5	62.1
10	-38	2141	3.62E+08	0.3	4	72.0
11	-40	2141	3.78E+08	0.3	2	78.6
12	-50	2141	4.09E+08	0.3	10	91.8
13	-60	2141	4.55E+08	0.3	10	114
14	-70	2141	4.97E+08	0.3	10	136
15	-80	2141	5.36E+08	0.3	10	158
16	-90	2141	5.72E+08	0.3	10	180

3.2 Gravity

The gravity led to substantial stresses and strains in the soil. The increased stress and strains led to significant nonphysical contributions in calculating the soil material and damping. Therefore, gravity was ignored in the Abaqus analysis. In general, gravity is not a major factor in the soil response in typical OWTs.

3.3 Rotor nacelle assembly

The rotor nacelle assembly (RNA) effect was included by adding a mass and a mass moment of inertia at the tower top. The values can be found in Table 3.4. The same values were used for both the jacket and the monopile design.

3.4 Jacket

The jacket was based on Rambøll's reference jacket design [64] from the European research project INNWIND. The Abaqus model was adopted from Pedersen and Askheim [63]. The jacket was located at 50 m depth, with a foundation of 4 piles penetrating 42 meters into the seabed. The top of the jacket and the transition piece were 26 m above sea level, and the top of the tower was 131.63 m above sea level. The entire model was modelled with linear interpolated Timoshenko beam elements (B31). The SSI stiffness is given in Equation (3.2).

The model was reused, except for the soil part. The soil profile was re-meshed to give better performance and allow for a larger base surface without a tremendous increase in computational time. The boundary conditions were changed to fixed condition at the bottom and a locked horizontal position for the vertical sides. In addition, some extra data sets were generated to help in later calculations. The model can be seen in Figure 3.1 and 3.2.

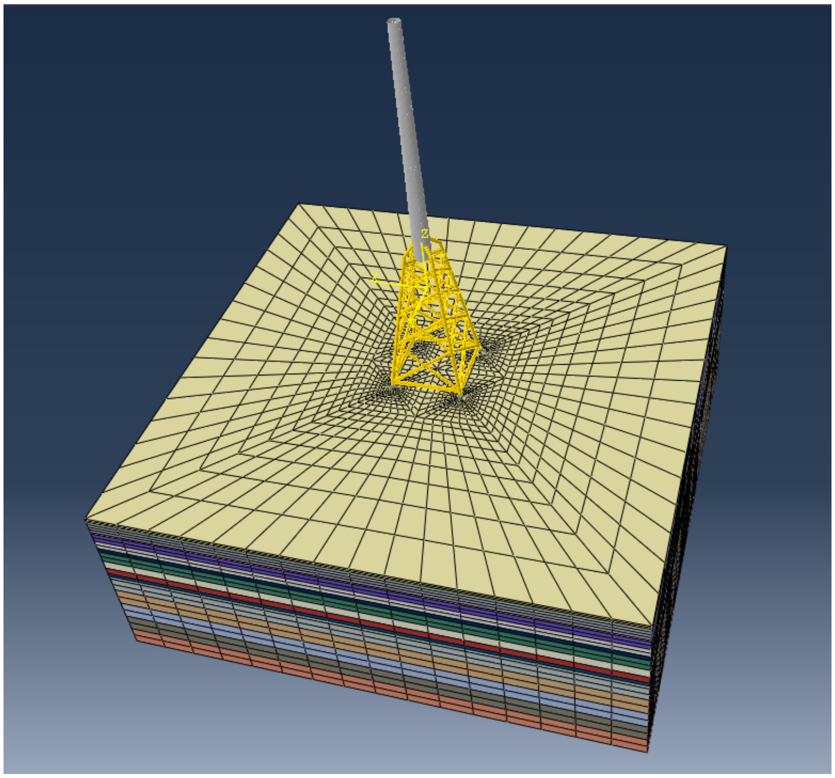


Figure 3.1: Full Abaqus model of jacket structure with soil

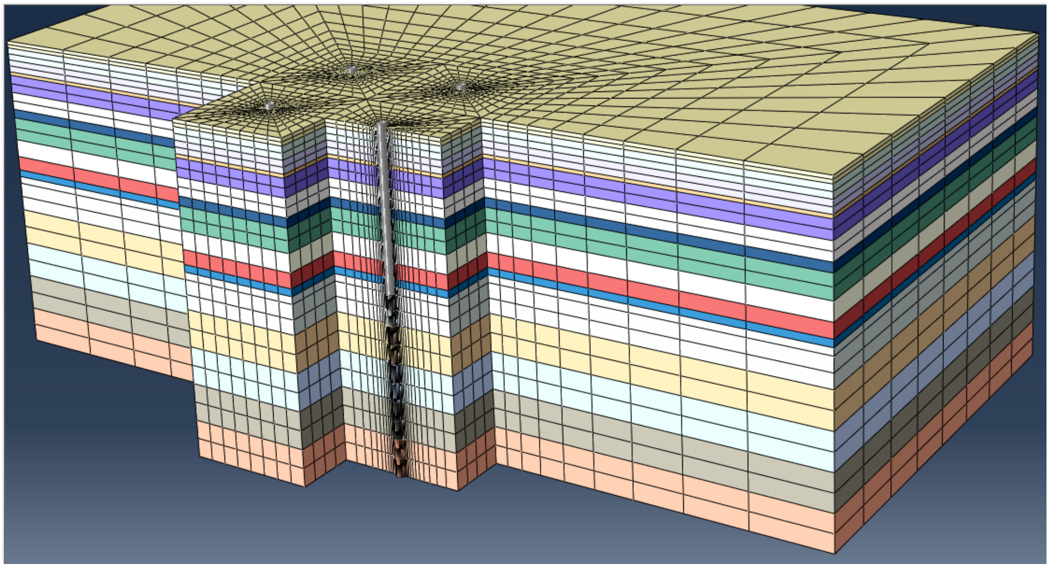


Figure 3.2: Abaqus model of piles of jacket structure in interaction with soil

The SSI stiffness matrix K_{SSI} is reused from Pedersen and Askheim [63].

$$K_{SSI} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} & k_{xr_x} & k_{xr_y} & k_{xr_z} \\ k_{xy} & k_{yy} & k_{yz} & k_{yr_x} & k_{yr_y} & k_{yr_z} \\ k_{xz} & k_{yz} & k_{zz} & k_{zr_x} & k_{zr_y} & k_{zr_z} \\ k_{xr_x} & k_{yr_x} & k_{zr_x} & k_{r_xr_x} & k_{r_xr_y} & k_{r_xr_z} \\ k_{xr_y} & k_{yr_y} & k_{zr_y} & k_{r_xr_y} & k_{r_yr_y} & k_{r_yr_z} \\ k_{xr_z} & k_{yr_z} & k_{zr_z} & k_{r_xr_z} & k_{r_yr_z} & k_{r_zr_z} \end{bmatrix} \tag{3.1}$$

$$= \begin{bmatrix} 4.69E+08 & 0 & 0 & 0 & -1.94E+09 & 0 \\ 0 & 4.69E+08 & 0 & 1.94E+09 & 0 & 0 \\ 0 & 0 & 2.45E+09 & 0 & 0 & 0 \\ 0 & 1.94E+09 & 0 & 1.52E+10 & 0 & 0 \\ -1.94E+09 & 0 & 0 & 0 & 1.52E+10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.97E+09 \end{bmatrix} \tag{3.2}$$

3.5 Monopile

The monopile was based on IEA's 10 MW-turbine [65]. The monopile is located at 30 m depth, with a foundation penetrating 42.6 meters into the seabed. The tower's hub height is 119 m. The dimensions and parameters are given in Table 3.3 and 3.4 respectively. The model can be seen in Figure 3.3 and 3.4.

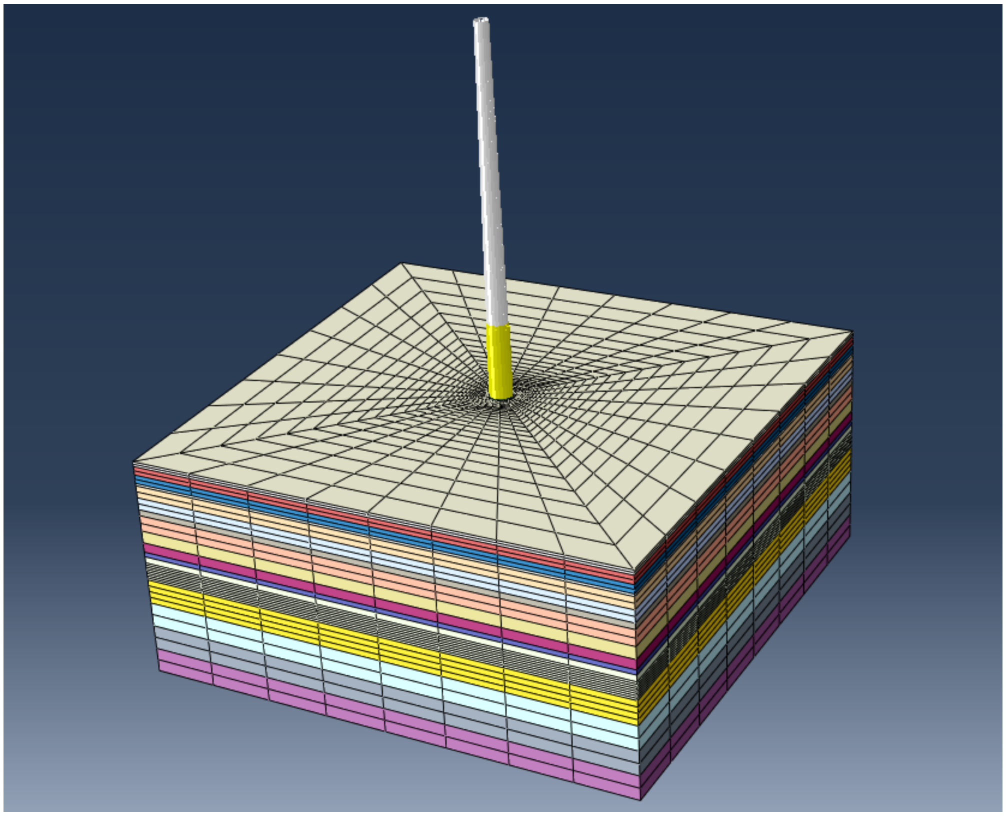


Figure 3.3: Abaqus model of monopile structure in interaction with soil

3.5.1 Tower and piles

The entire monopile was originally modelled with linear interpolated Timoshenko beam elements (B31) with linear varying tapered beams between the given dimensions listed in Table 3.3. Due to Abaqus' inability to handle tapered cross-sections a discretisation was used, splitting the beams up into several elements with a linearly tapering cross-section. This model was the basis for the SSI-stiffness matrix. Due to problems with this model, sub mudline parts of the model were later modified and used for the damping estimation. This is covered in greater detail in Section 3.5.4.

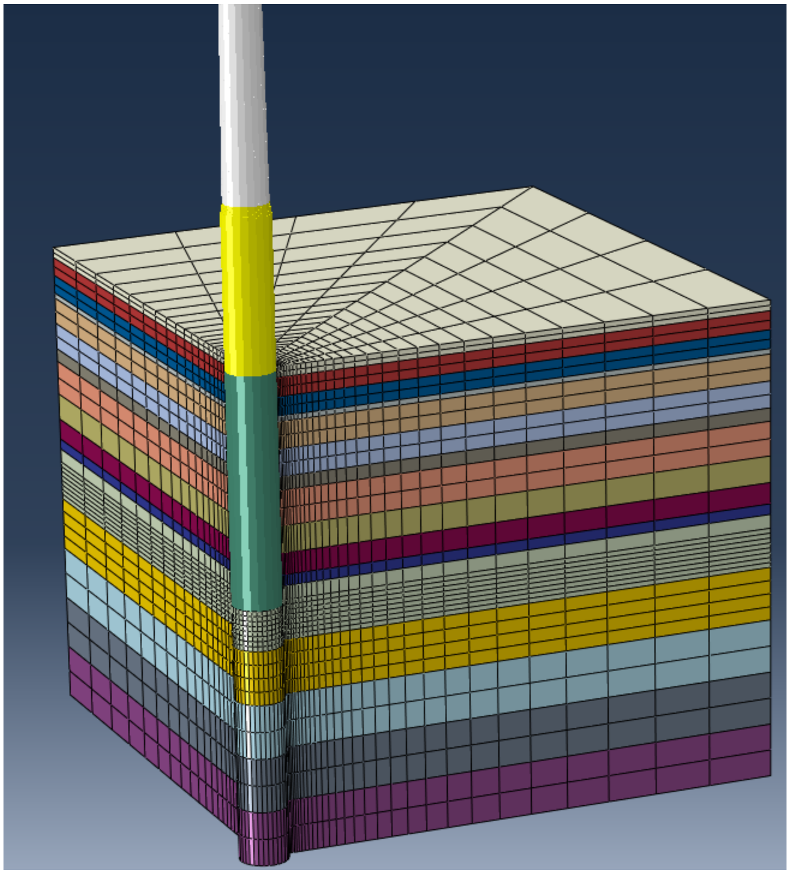


Figure 3.4: Abaqus model of monopile structure in interaction with soil

3.5.2 Effect of water

The effect of the water beam surrounding the submerged part of the structure was approximately accounted for by doubling the steel mass density in the parts of the tower below the sea surface.

Table 3.3: Dimensions monopile from [65] table 15

Location [-]	**z [m]**	$\mathbf{D_{outer}}$**[m]**	$\mathbf{D_{inner}}$**[m]**
Yaw bearing	145.63	5.5	5.44
	134.55	5.79	5.73
	124.04	6.07	6
	113.54	6.35	6.26
	103.03	6.63	6.53
	92.53	6.91	6.8
	82.02	7.19	7.07
	71.52	7.46	7.34
	61.01	7.74	7.61
↑	50.51	8.02	7.88
tower	40	8.3	8.16
foundation	40	9	8.7
↓	38	9	8.7
	36	9	8.7
	34	9	8.7
	32	9	8.69
waterline	30	9	8.69
	28	9	8.69
	26	9	8.69
	24	9	8.69
	22	9	8.69
	20	9	8.69
	16	9	8.8
	12	9	8.8
	8	9	8.8
	4	9	8.8
mudline	0	9	8.8
	-8.4	9	8.8
	-16.8	9	8.8
	-25.2	9	8.8
	-33.6	9	8.8
	-42.6	9	8.8

Table 3.4: Monopile parameters

Parameter	**Value**
Young's modulus [GPa]	210
Poison ratio [-]	0.3
Tower mass density [$\frac{kg}{m^3}$]	8500
RNA total mass [kg]	866 555
RNA total mass moment of inertia [kgm^2]	240 016 659
RNA total mass moment of inertia [kgm^2]	142 102 115
RNA total mass moment of inertia [kgm^2]	111 846 413

3.5.3 Generating SSI stiffnes

In order to generate the SSI stiffness matrix, all non-linear components in the soil material model were temporally removed. This soil model was combined with a beam model for the pile. The tower was ignored in this analysis.

Then a unit displacement of 1 was applied in each of the six principle directions at the pile top, using a linear analysis. Finally, the reaction forces in the top of the pile from all of the six unit displacements were fetched from the Abaqus ODB-files using a python script. The generated matrix is given in Equation (3.4).

$$K_{SSI} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} & k_{xr_x} & k_{xr_y} & k_{xr_z} \\ k_{xy} & k_{yy} & k_{yz} & k_{yr_x} & k_{yr_y} & k_{yr_z} \\ k_{xz} & k_{yz} & k_{zz} & k_{zr_x} & k_{zr_y} & k_{zr_z} \\ k_{xr_x} & k_{yr_x} & k_{zr_x} & k_{r_xr_x} & k_{r_xr_y} & k_{r_xr_z} \\ k_{xr_y} & k_{yr_y} & k_{zr_y} & k_{r_xr_y} & k_{r_yr_y} & k_{r_yr_z} \\ k_{xr_z} & k_{yr_z} & k_{zr_z} & k_{r_xr_z} & k_{r_yr_z} & k_{r_zr_z} \end{bmatrix} \tag{3.3}$$

$$= \begin{bmatrix} 3.27E+09 & 0 & 0 & 0 & -2.84E+10 & 0 \\ 0 & 3.27E+09 & 0 & 2.84E+10 & 0 & 0 \\ 0 & 0 & 9.73E+09 & 0 & 0 & 0 \\ 0 & 2.84E+10 & 0 & 5.80E+11 & 0 & 0 \\ -2.84E+10 & 0 & 0 & 0 & 5.80E+11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.18E+11 \end{bmatrix} \tag{3.4}$$

3.5.4 Model for damping estimation

The initial plan was to also model the pile with beam elements in the damping estimations. This model, however, had severe problems handling rotations at the top of the pile. A possible explanation might be the 4.5 m distance between the surfaces being tied together, where the pile had rotational DOFs in the nodes, while this was not the case for the solid soil elements. In addition, the monopile-pile had much higher stiffness and had close to rigid behaviour. This close to rigid behaviour led to large deformations when the pile was rotated at the top. This, in turn, led to large deformations in the soil layer directly below the pile.

Two different approaches were tested to overcome this problem: one shell-based pile and one solid-based pile. The solid-based pile was used for the damping estimations because it showed satisfactory behaviour and simplified calculation of the strain energy from the pile.

One of the advantages of using solid elements is that a soil column (Figure 3.10) can be added inside the pile as well, which gives a more realistic representation of the entire system. This same soil profile was used both inside and outside the pile. The soil column was tied to the inner pile surface. In addition, it was tied to the rest of the soil at the bottom of the soil column. The complete tie-relationships between the instances are described in Figure 3.8.

The solid pile (Figure 3.6) was modelled with the same dimension as in Table 3.3 and the same material parameters as the rest of the model. The pile was modelled with eight-node interpolated brick elements with reduced integration and hourglass control (C3D8R). One element was used over the pile's wall thickness in the meshing procedure. To be able to apply forced deformations to the pile, a rigid ring (Figure 3.7) was tied to the pile top with it's centre of rotation at the centre of the pile cross-section. For further details of the model, see Figure 3.5, 3.8, 3.9 and 3.10.

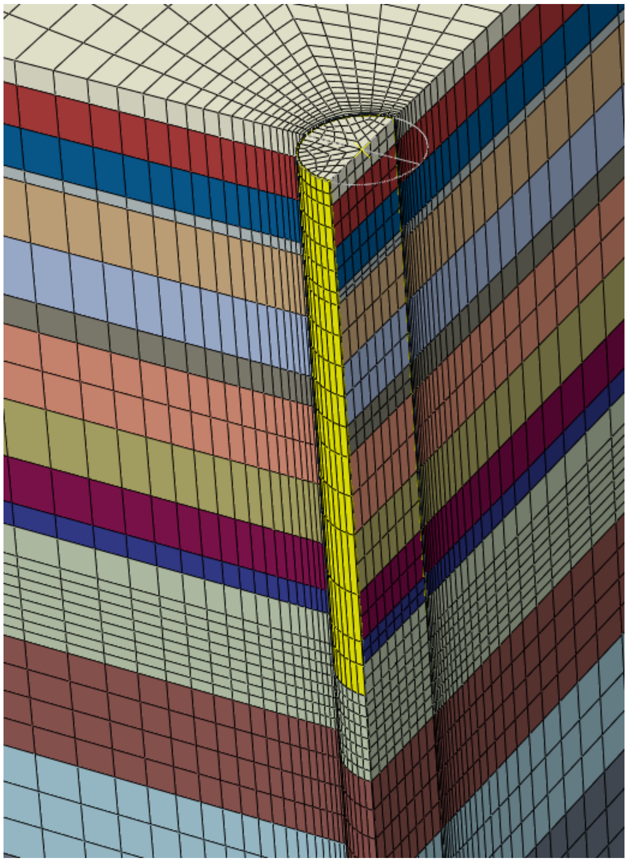


Figure 3.5: Abaqus model of solid-element-pile in interaction with soil. Pile elements are marked yellow.

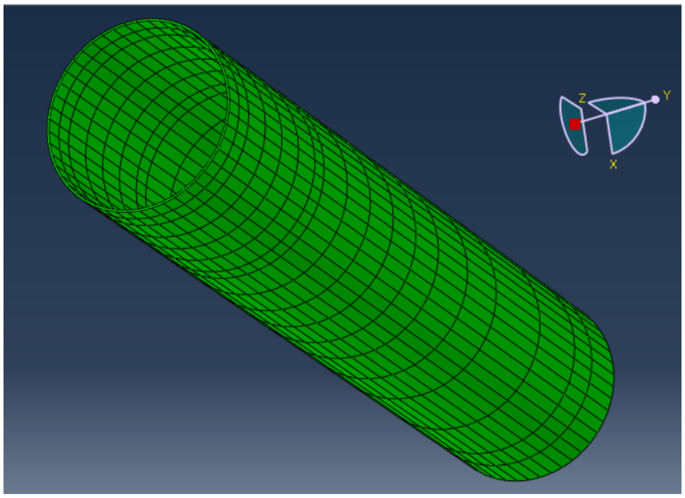


Figure 3.6: Monopile solid pile used in damping estimation

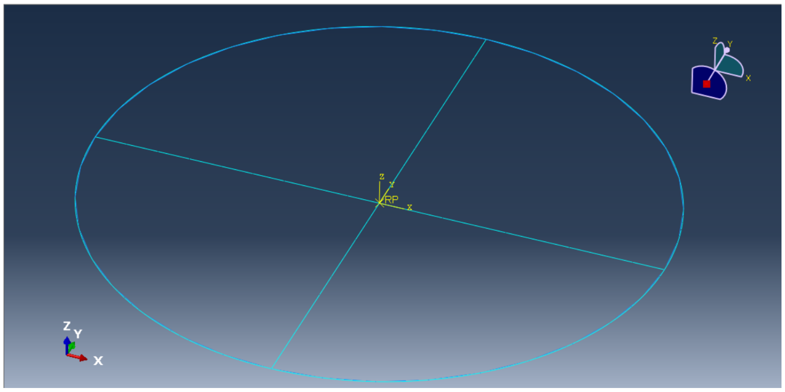


Figure 3.7: Rigid ring used to apply deformations to solid pile

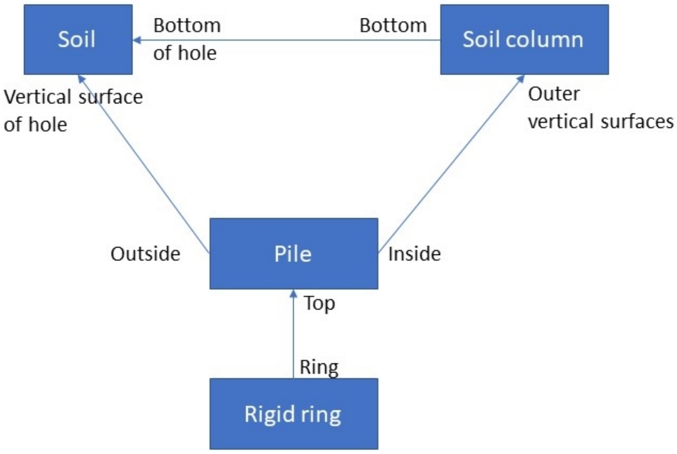


Figure 3.8: Relationship between instances. The source of the arrow indicates that the instance is the master in the relationship. Being pointed at indicating that the instance is the slave. The text outside the boxes indicates which surface on the instance is tied.

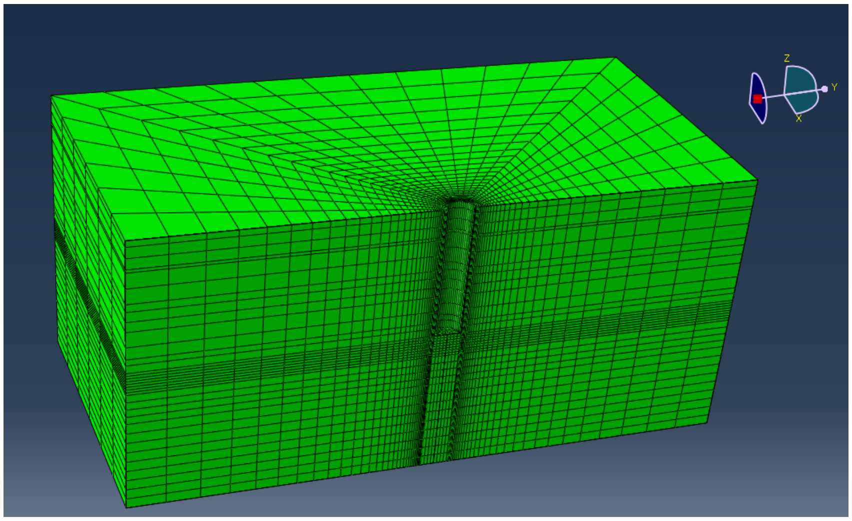


Figure 3.9: The soil used in the damping estimations (sliced). Dimension: 200x200x90m. Hole diameter: 9 m. Hole depth: 42.6 m.

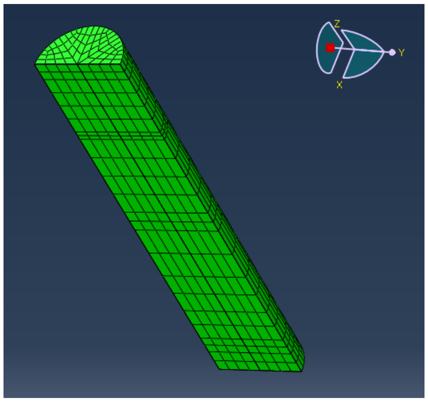


Figure 3.10: The soil column added inside the pile used in the damping estimations (slices). Diameter: 8.8 m. Height: 42.6 m

3.6 Calculation of damping

The damping calculation was done according to method 1 in Section 2.5.1.

First, a representative displacement was chosen for each mode at the top of the pile as described in Section 5. The representative displacement chosen was the average peak height for each mode. Then an analysis in Abaqus was initiated. The analysis applied the representative displacement added on top of the static displacement as a forced displacement at the top of the piles in a non-linear static analysis. These analyses were the basis for calculating the energy loss $W_{L static+dynamic}$ In addition, one analysis only including the static displacement was run. This analysis was the basis for calculating the energy loss $W_{L static}$. By finding $W_{L static}$ and $W_{L static+dynamic}$, $W_{L dynamic} = W_{L static+dynamic} - W_{L static}$ could be calculated.

The mode shapes can be scaled by either 1 or -1. This thesis decided the scale factor by calculating which scale factor would give the highest displacement given a rigid configuration at a given number of meters below the top. For the monopile, the pile was assessed 10 m below the top. For the jacket, pile 1 was assessed 5 m below the top.

After the analysis was run, the results were available in an Abaqus ODB-file. All relevant results were then extracted from the file using the python-interface. A geometry set was defined, including all the soil elements. From this set, each element's logarithmic strain-vector, stress-vector, and volume were extracted and put into a NumPy array.

The energy loss per element A_L was calculated as described in Section 2.5.1

To calculate the total energy loss, the energy loss per element was summed up. Combined with the dynamic modal energy calculated in the modal analysis in Section 5. The damping ratio for the entire structure can be calculated as described in Section 2.5.1.

The strain energy of only the foundation was used in calculating damping curves. The strain energy was requested and included in Abaqus.

The script used to calculate the damping can be found in Appendix B.

3.7 Exporting matrices

The matrices in Abaqus can not be exported using the CAE module in Abaqus. Instead, the input files have to be modified directly by pasting an extra step at the end of the procedure exporting the mass and stiffness matrix.

The matrices used in the modal analysis must be generated from a single part. This is not a problem for the monopile model as the substructure and the tower is modelled in one part. For the jacket model, the transition piece, tower, and jacket are three separate parts combined with a tie constraint. The tie-constraint does not merge the nodes being tied but instead sets the displacement of the slave node equal to the displacement of the master node. This leaves the matrices generated impossible to use in a simple modal analysis.

A new part was generated using the merge technique in Abaqus to solve the problem, merging the intersecting nodes for the three parts and creating a new part. Unfortunately, this process deletes information such as the orientation of the beams and inertia. This information must be reapplied to the new part before the matrices are generated.

3.8 Meshing of soil

When running the first few analyses, a too-small soil model made the soil "stiffer" than it should be and gave unrealistic stresses. Unrealistic stresses, in turn, gave incorrect estimations of damping. Therefore, it was necessary with a larger soil model such that the nonphysical effect of the boundary conditions was reduced. Furthermore, it was vital to reduce the number of elements with an efficient mesh that avoids fine meshing in areas where great details had little effect. The authors used an approach where the elements "radiate" from the holes, giving the elements an increased width with an increased distance from the holes. In addition, biasing was used to linearly increase the depth of the elements further away from the holes. Unfortunately, implementing this is a time-consuming task as Abaqus has no standard approach to achieve this mesh.

3.9 Verification of damping measurements

The model was compared to the results in Zhang et al. [18] (Figure 3.11). The figure shows the relationship between displacement and damping ratio in the foundation. Some of the parameters were set to match the model in [18]. The pile dimension was set to have a diameter of 2.14 m, a thickness of 60 mm, and a height of 40. In addition, a constant yield stress for soil was set to 20 kPa. It is important noticing that the two models did not match perfectly since Zhang et al. used a different soil damping ratio model. Therefore, the comparison will not be entirely applicable, but will instead provide a partial verification. The two plots are shown in Figure 3.11 and 3.12. They show a good correlation with some expected differences.

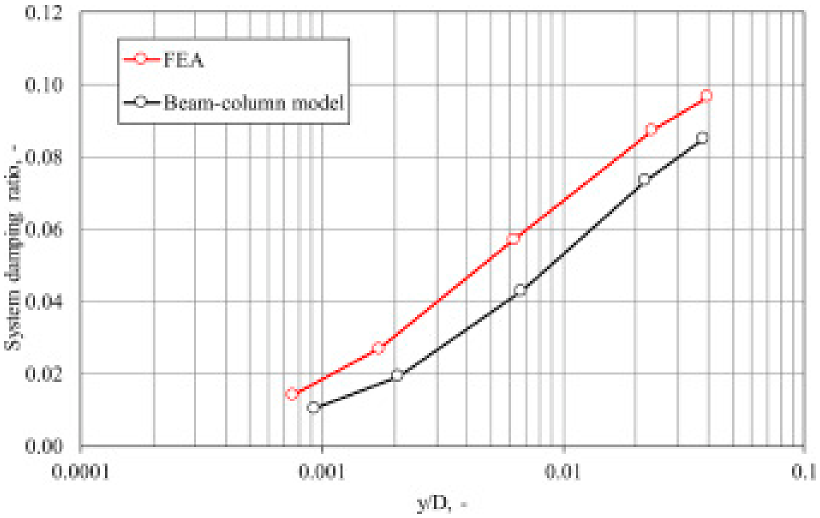


Figure 3.11: Damping curves from Zhang et al. [18] figure 13

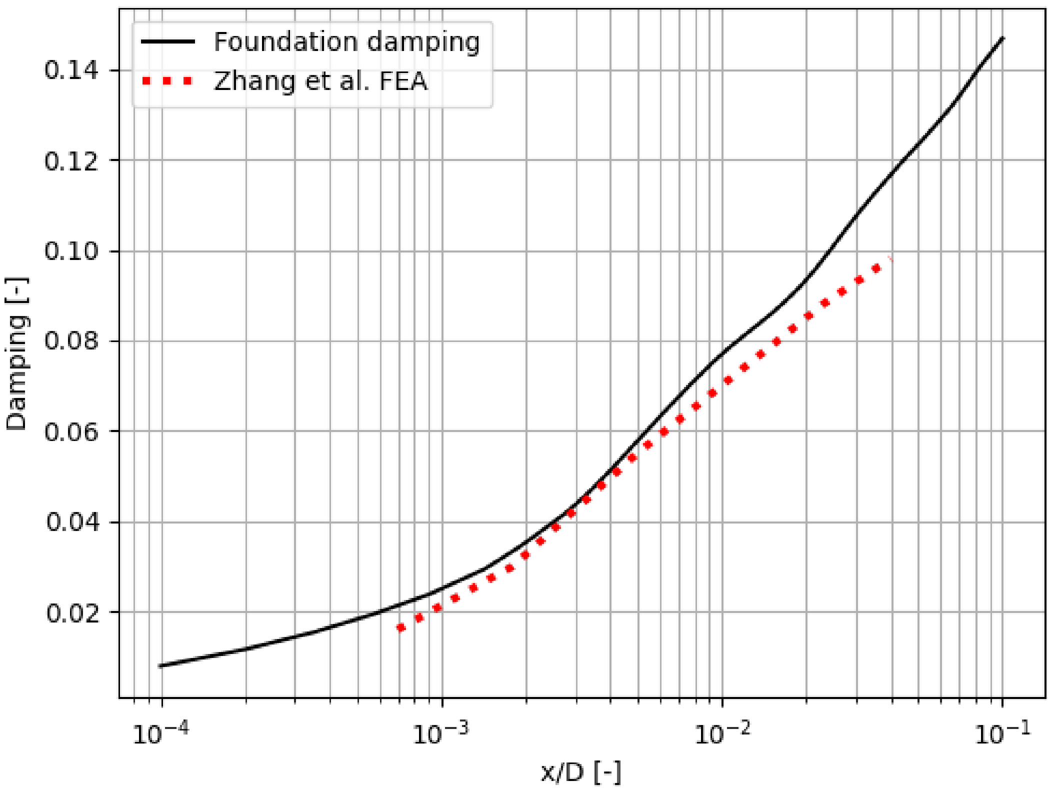


Figure 3.12: Comparison between damping curves estimated by the authors and Zhang et al. [18]

4. OpenFAST

The information in Section 4.1 and 4.2 is based on the OpenFAST Read the Docs page [6, 66] combined and adapted by the author's personal experience using the program. Sources for the modules whose complete documentation is not provided on Read the Docs will be provided along with the relevant sections. If the reader already understands how FAST operates, and does not desire a recap of the basics, then it is recommended to skip to Section 4.3.

4.1 Understanding FAST

Fatigue, Aerodynamics, Structures and Turbulence (FAST) is a program developed by National Renewable Energy Laboratory (NREL)s National Wind Technology Center in Colorado, USA. It is a physics-based tool for simulating the coupled aero-hydro-servo-elastic response of wind turbines. The program may be used for both commercial and research purposes. In 2017 FAST changed its name to OpenFAST, representing a shift to open-source. This change has allowed the program's inner workings to be more accessible and transparent to the end-user. Additionally, the development of the program is not limited to a single research group at NREL, but other teams and individuals may make improvements.

The OpenFAST code is written in Fortran, with some modules written in C and C++. There are several ways to download and run OpenFAST, such as simple precompiled binaries. However, the authors have chosen to build the required solution themselves as it gives a greater amount of control over the version used [67]. This method requires Intel Fortran and Microsoft Visual Studio to make a working version of OpenFAST. To assure that the program is functioning as it should, regression tests are included in the GitHub release of OpenFAST. The regression test compares the output from the user's computer to a verified output from the developer and ensures that they differ by no more than the required tolerance.

OpenFAST does not have a user interface which helps to give an intuitive understanding of what information the program needs to begin simulations. Instead it reads carefully constructed .dat and .fst input files. Many examples of the input files are provided along with the regression test or found online within various GitHub repositories. All input files are formatted similarly with three different columns. The first column is the variable, such as string, integer, float or flag. Column two is dedicated to variable names, followed lastly by a description of the variable. The description is often accompanied by valid values that can be used. This format is known as "key + value format", as the value or variable is assigned to a key. The format allows for better troubleshooting as it will be more obvious to the code when it reads a key it does not expect. Further explanations of the input files are found on the Read the Docs page. Some inputs do not follow the three column structure. This is because each variable name might have multiple variables associated with it, such as lists and matrices.

4.1.1 Main input

The main input file acts as a glue code for the rest of the input files. In this file, values used throughout the other input files are specified. These values include time step, analysis time, linearisation features and output format. Linearisation generates a linear model of the turbine that may be used for frequency-domain analyses and eigenvalue analyses. As the main input file does not include all the information about the turbine structure, it must also include the path to the other input files. In addition, different modules may be enabled through the main input file. This means that the module dealing with hydrodynamic forces may be activated for OWTs or deactivated for land-based wind turbines.

4.1.2 ElastoDyn

ElastoDyn computes the structural dynamics for the tower, blades, platform, rotor, nacelle and drivetrain. Out of all the modules, ElastoDyn is the only one that is required to run FAST. This is understandable as it handles most aspects of a basic wind turbine analysis. The module requires parameters such as the gravitational acceleration and tower integration method. The analyst may choose between three fourth-order integration

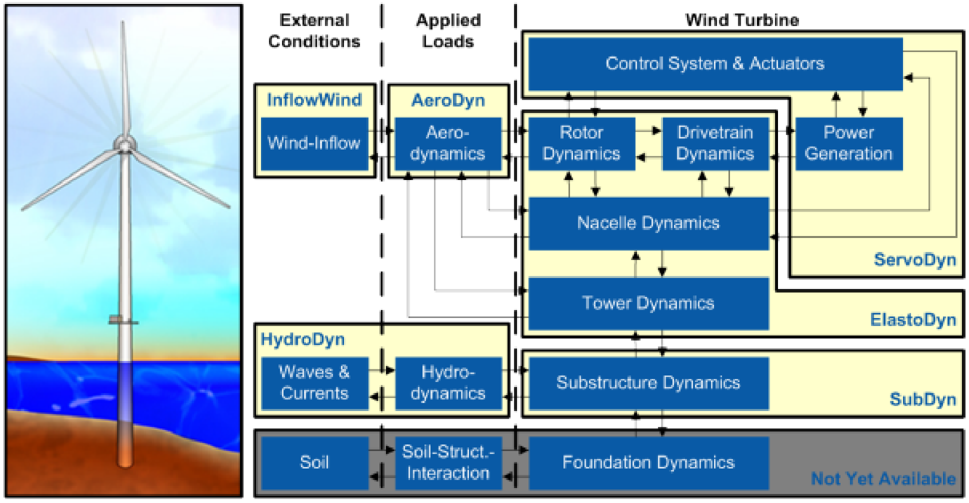


Figure 4.1: Schematic of the FAST modularisation framework for a fixed turbine (ref. [68])

methods, Runge-Kutta (RK), Adams-Bashfourth (AB), Adams-Bashfourth-Moulton (ABM) and second-order Adams-Moulton (AM). Cellier and Kofman [69] describes these integration methods. AB is an explicit method with a small stable region. AM is an implicit method that is not stiffly stable. Meanwhile, ABM uses both AB as the predictor and AM as the corrector to achieve a happy medium between the two. Fourth-order RK algorithm is a popular single step explicit method.

One of the primary tasks of ElastoDyn is setting up the initial conditions of the tower and blades. It uses six DOFs per node with three translational and three rotational. Translational DOFs are referred to as surge, sway and heave for movement in the axis longitudinal, transverse and vertical to the wind's direction, respectively. Similarly, roll, pitch and yaw refer to rotation about the surge, sway and heave directions. See Figure 1.2 for a visual representation of the coordinate directions. For the ElastoDyn tower, one may set surge, sway and yaw for the tower top, and surge, sway, heave, roll, pitch and yaw for the platform/TP. The final initial conditions include setting blade rotational frequency and various other characteristics of the blades. The old FAST user's guide [70] is the best place to find out more about the various coordinate systems that FAST employs.

To simplify the turbine model and reduce computing power, ElastoDyn's primary input file discretises a three-bladed turbine model into a reduced number of DOFs. Sixteen of these DOFs can be enabled or disabled depending on the function of the analysis. These are the six platform DOFs that represent the state at the bottom of the tower that is read by the modules covering the substructure. Four DOFs are reserved for the four first tower modes. Two in the fore-aft (surge) direction and two in the side-to-side (sway) direction. Three additional modes are represented by the blades, two in the flapwise (surge) direction and one in the edgewise (roll) direction. The final three DOFs are for the nacelle yaw, drivetrain- and generator rotation.

The subsequent inputs ElastoDyn requires are the geometry of various turbine parts. In addition to the placement of the parts, the mass and inertia must also be supplied to the program. Finally, the analyst must input some drivetrain characteristics before deciding which time-domain outputs ElastoDyn should produce.

There are two additional input files that ElastoDyn requires. Namely, the blade files and the tower file. These are almost identical to each other in setup and function. A matrix containing density and stiffness properties for nodes distributed along the blade/tower must be provided. Three (blade) or four (tower) mode shapes must be given in the polynomial form. This is done through BModes and ModeShapePolyFitting.xls as described in Section 4.2.1. The modes may also receive damping values. Only one tower input file may be provided. However, up to three unique blade files can be defined, one for each blade. Although the functionality for different blades is present, in all but some special analyses, symmetry dictates that repeating the same blade file across all blades will be sufficient.

4.1.3 InflowWind

InflowWind [71] is OpenFAST's driver for generating wind fields to be applied to turbines. It can either be used coupled to FAST or as a standalone driver. This module can create a wind field in seven different ways. They are steady wind, uniform wind file, TurbSim file, BLADED file, HAWC file, user-specified, or native BLADED. Most of these import wind fields from other programs such that they can be used within the FAST framework. The exceptions are steady wind, where parameters are specified within the InflowWind file, and the user-specified wind, where the end-user edits the Fortran code to represent their desired wind conditions.

Only three input parameters are required to generate the steady wind field that remains constant with time. The first two are the horizontal wind speed (U_{ref}) at a given height (z_{ref}). The final parameter is the power-law exponent ($PLexp$), used to scale the wind velocity over the height (z). Equation (4.1) expresses how these parameters generate a steady wind field.

$$U(z) = U_{ref} \left(\frac{z}{z_{ref}} \right)^{PLexp} \tag{4.1}$$

A uniform wind file imports a simple text file containing eight columns. The columns include data at different time steps, such as wind velocities, direction, gust velocity, power-law exponent and wind-shear parameters. Reference height and -length is given in the InflowWind file since they are constant over time. Steady wind and uniform wind profiles are compatible with FASTs linearisation features due to their more consistent wind fields that the other profiles lack. The latter profiles use more complex turbulence models such as the Mann uniform shear model or the Kaimal spectrum that was covered in Section 2.2.1.

4.1.4 AeroDyn

AeroDyn handles the aerodynamic loading and response of wind turbine tower and blades. It used to have InflowWind integrated, but has in later FAST releases become separated. Therefore AeroDyn receives the wind field and turbine data via the FAST driver. As a standalone module, the wind- and turbine data must be stated in the input file. The wind field is converted to loading through by utilising actuator lines. This principle turns the three-dimensional flow around the turbine bodies into a two-dimensional flow around cross-sections.

Before running an AeroDyn simulation, the user must choose which four models best describe their turbine. For rotor wake/induction one may choose between blade-element momentum theory (BEMT), dynamic blade-element momentum theory (DBEMT) and convecting lagrangian filaments (OLAF). As for blade airfoil aerodynamics, the choice is simply between a steady model and three various Beddoes-Leisham models, including the original 1989 model, as well as improvements from Gonzáles and Minnema/Pierce. Next, the analyst must decide whether or not to include tower influence of wind based on potential flow, and if the Bak correction should be applied. Lastly, which of Powles or Eames model should be used if one wants to account for tower drag.

In addition to all the model parameters, the input file must also include basic environmental parameters such as air density, pressure and viscosity. Each blade must also have a blade definition input file that gives cross-sectional properties at nodal points along the length of a blade. The extra input file is only a matrix with seven columns giving geometry, twist and cord. It also includes a reference to airfoil data, which is in its separate file. Airfoil input files give lift, drag and pitch for different parts of a turbine blade.

4.1.5 ServoDyn

As the name suggests, ServoDyn manages the servo dynamics within the FAST framework. This includes controlling the blade-pitch, nacelle-yaw and torque. The other functionalities of the module are to apply the brakes, regulate the tuned-mass dampers, and override pitch and yaw. A submodule for ServoDyn is Structural Control which controls the tuned-mass dampers and allows the user to prescribe forces and moments at specific nodes.

4.1.6 HydroDyn

HydroDyn models the hydrodynamic forces and damping for both fixed and floating substructures of OWTs. This is achieved through strip-theory modelling. It is one of the modules that may be used as a standalone driver. In addition to the hydrodynamic response, HydroDyn also generates waves and current, making it the primary module for simulating the ocean's influence on OWTs. The geometry of the substructure is constructed as node-, member- and cross-section tables within the HydroDyn input file.

The required environmental conditions, that HydroDyn needs, include water density and depth from the seabed to mean sea level. After that, the analyst must apply one of eight wave kinematics models. The simplest of these is the still water model which does not model waves, only current. To keep it simple, but also include waves, either of the two periodic wave models might be best. The JONSWAP spectrum can be used to generate waves with varying height, frequency and direction. Recall Equation (2.10) for frequency-dependent the description of the JONSWAP spectrum. HydroDyn uses code-compliant values for the variables except for the peak spectral period (T_p), significant wave height (H_S) and peak-enhancement factor (γ). Since the peak-enhancement factor is user decided, setting it to 1 will inform HydroDyn to generate a P-M spectrum. For some simulations, a white noise spectrum is desired, and then HydroDyn will not read the peak spectral period and instead produce a constant frequency spectrum. Similar to InflowWind, the user can edit some code to describe their own spectrum that HydroDyn can run. Finally, the analyst can import externally generated wave elevations and kinematics.

To repeat the same time-domain wave conditions across multiple HydroDyn simulations, seeds may be specified that tell the psuedorandom number generator (PRNG) to create an identical time series. Specific structures require the use of second-order wave theory. HydroDyn has the capabilities to implement both the difference-frequency method and the summation frequency method. For floating turbines, the module can use the potential-flow model, radiation memory-effect and second-order forces to describe the OWT's response accurately. There are also preload, stiffness, damping and drag matrices to give the platform additional characteristics. Any OWT may be subject to currents. HydroDyn has three current sub-models to apply. A power-law model, a linear model down to a reference depth, or a constant current. Alternatively, users can implement their own model by editing the HydroDyn code and recompiling the driver. The last functionality HydroDyn provides is to augment members with marine growth or fill them with water if they are submerged while not in a closed environment.

4.1.7 SubDyn

SubDyn is the FAST framework's primary substructure dynamics modelling tool. It utilises the displacements, velocities, accelerations and forces from ElastoDyn at the TP to interpolate the response to the rest of the substructure. Since HydroDyn and SubDyn cover the substructure, that means they also have overlapping input data. HydroDyn imports the displacement, velocities and accelerations from SubDyn in exchange for the hydrodynamic forces for the given response situation. This close relationship between the modules means that it is highly recommended to use the same nodes, members and cross-sections to describe the substructure geometry. However, the tables do not contain the same information despite describing the geometry. For example, when SubDyn constructs nodes, it requires joint type and stiffness in addition to its coordinates. However, it does not need axial coefficients or the joint overlap flag like HydroDyn.

The SubDyn model can be discretised in two separate ways, full finite element method (FEM) DOFs or a reduced dynamic system. Doing a full linear FEM simulation requires a lot of computing power and can realistically only be performed on simple structures with a limited number of DOFs. SubDyn does offer functionality to divide members into smaller members for a more accurate model. To avoid time-consuming simulations, the analyst should enable substructure coupling to greatly reduce the number of DOFs. This is achieved through C-B reduction as described in Section 2.4.4. For a frame structure, the number DOFs may be reduced from several thousand to a couple dozen. The exact number of DOFs are chosen by the analyst. The first few are inherited from the TP (which acts as the interface node during the C-B reduction), and the rest are the number of modes that the analyst wants to include, starting from the modes with the lowest frequency. If no additional modes should be included, then a special case of C-B reduction is performed, called Guyan reduction.

Substructure damping is specified through the Guyan and C-B modes ($\left[\bar{C}_{BB}\right]$ and $\left[\bar{C}_{NN}\right]$ in Equation (2.46) respectively). The retained C-B modes can be given specific damping values. Meanwhile, the Guyan modes can

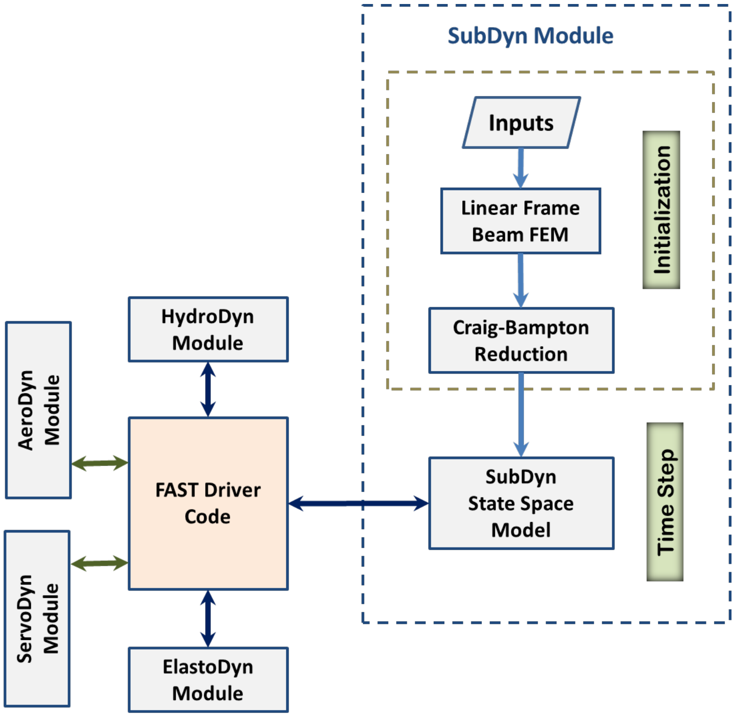


Figure 4.2: Schematic of the SubDyn within the FAST modularisation framework (ref. [66])

be damped through Rayleigh damping by providing the two Rayleigh parameters. Alternatively the entire Guyan damping matrix ($\left[\bar{C}_{BB}\right]$) can be stated within SubDyn. The user decides which DOFs of the base reaction- and interface joints are free or fixed. An additional file may be provided for the reaction joints, including the full 6x6 mass and stiffness matrix for each joint. This matrix means that SSI may be included through SubDyn. Any joint in the module can also have a concentrated mass added to it.

Members within SubDyn can be more than just beams, but also rigid links and pretension cables. The beams must be tubular and can be modelled as both Euler-Bernoulli (E-B) or Timoshenko beams. They require outer diameter and wall thickness to understand the size, and Young's modulus, shear modulus and material density to complete the cross-section. Pretensioned cables need material density, pre-tensioning force and elongation stiffness. Rigid links only need to know the material density to serve their function.

4.1.8 Other modules

The following section is an overview of modules that are part of the FAST modulisation framework. They are only being covered briefly as they were not enabled during the simulations carried out for this thesis.

4.1.8.1 BeamDyn

BeamDyn is an alternative to ElastoDyn blade for modelling the blade response. ElastoDyn provides the displacements, velocities, accelerations and reaction forces at the root of the blade, and BeamDyn calculates the response of the blade. ElastoDyn blade is more straightforward and only provides sufficiently accurate results for straight isotropic blades dominated by bending. Meanwhile, BeamDyn uses geometrically exact beam theory that can handle geometric nonlinearity and large deflection.

4.1.8.2 ExtPtfm

ExtPtfm replaces SubDyn and HydroDyn for substructure response by using a superelement approach. It requires the mass, damping, stiffness and load matrices for the C-B modes. The user must use their own FE program to generate these matrices. ExtPtfm gives the user more freedom by allowing the analyst to make the C-B matrices how they desire, but it lacks many of the features of SubDyn.

4.1.8.3 Mooring

MAP++, FEAM, MoorDyn and OrcaFlex are all options to simulate the mooring lines for floating platforms. They receive the six DOFs for the TP from ElastoDyn, and then they compute the forces and geometry relating to the mooring lines.

4.1.8.4 Ice load

IceFloe [72] and IceDyn are modules designed to calculate the forces applied by ice at the water surface. Ice has several failure modes, which depend on the thickness of the ice and the geometry of the member subjected to the force. The modules account for these varying conditions and calculate the external force to apply to the OpenFAST driver.

4.2 Other FAST tools

There are several other useful programs to aid with wind turbine design that comes in addition to the modules that are part of the FAST modularisation framework. These programs help pre- or post-process data for OpenFAST. The following sections cover some of the programs used in this thesis.

4.2.1 BModes

BModes [73] was designed to obtain rotating blade coupled modes. It is the preferred method for obtaining mode shapes as it uses the FEM with linearisation that can account for both the Coriolis and centrifugal effects. BModes works by first idealising the blade as a E-B beam, then calculating the dynamic beam energy and establishing the partial differential equations via Hamilton's principle. The blade is then discretised into 15 DOF beam elements, followed by applying the boundary conditions to find the ordinary differential equations. Next, analytical linearisation is carried out to find the inertial, stiffness and gyroscopic matrices. Finally, the modes are established by conducting an eigenvalue analysis of the matrices. Although designed for wind turbine blades, BModes can also be applied to other beam-like structures such as the wind turbine tower.

The program needs inputs that describe the beam in question. After choosing whether BModes is modelling a blade or tower, a description of the beam size is required. BModes needs rotation speed and pitch for the blades. The connection to the hub or substructure can be cantilevered or partially fixed through a mass and stiffness matrix. At the free end, the designer can apply mass and inertial properties. This function is helpful for tower modelling such that the significant contribution of the off-centred mass of the hub is included. A separate input file is needed to describe the blade and tower geometry. A matrix with the mass and stiffness properties at different sampling points along the beam must be supplied such that BModes can interpolate the properties for the entire beam. Finally, the analyst must decide the location and number of discretised beam elements the blade or tower needs for an accurate description.

After executing BModes, the user is supplied with an output file with the number of modes specified in the input file. The output file gives the frequency for each mode as well as the fore-aft/flap displacement and slope, side-to-side/lag displacement and slope, and the twist about the centerline. These values are provided at the previously entered nodes along the beam. The mode shapes are human-understandable, but ElastoDyn requires they be given in the polynomial form. ModeShapePolyFitting.xls is an Excel spreadsheet designed to translate the mode shapes to the required polynomial coefficients. These allow ElastoDyn to use the low-frequency modes it requires.

4.2.2 TurbSim

TurbSim [74] is a turbulent wind simulator that can produce full-field stochastic winds for InlfowWind. It has several built-in turbulence models that can produce accurate wind fields with relatively few parameters. The output from TurbSim are three vector components that show wind velocity and direction, as shown in Figure 4.3.

A grid of vectors are given in a two-dimensional plane at equally spaced time intervals. InlfowWind interpolates the vectors through time and space to produce a smooth wind field.

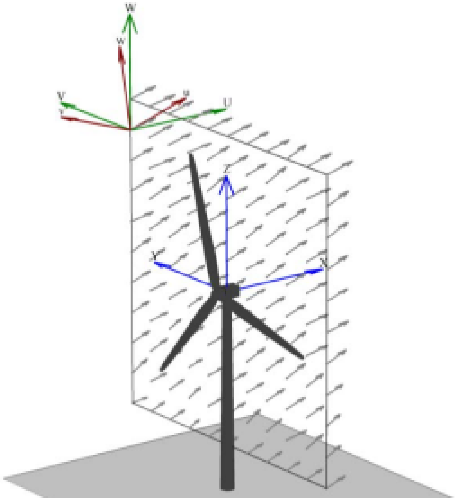


Figure 4.3: Plane of vector components producing a wind field in TurbSim (ref. [74])

The first thing the analyst must do in TurbSim is provide the required geometrical parameters to describe the plane such that it covers the entire swept area of the rotors. That includes the hub height, height and width of the grid, and the number of sampling points in each direction. The timestep and analysis time is also required as it does not retrieve these parameters from FAST.

Most of the wind parameters are set by the type of turbulence model chosen. The turbulence models include the Kaimal, von Karman, Risø smooth, wind farm, and even user-defined. After setting a model, the user must decide which IEC standard to utilise. Different standards should be used for turbines of varying size and location. TurbSim needs the reference height and turbulence intensity to create a simulated time series based on measured conditions. The analyst chooses parameters such as the roughness length or power-law exponent themselves or lets TurbSim pick values from chosen IEC standard. One must alter the mean wind velocity and turbulence characteristics when simulating different load conditions. An ultimate limit state (ULS) analysis has higher mean wind and different turbulence characteristics than a serviceability limit state (SLS) analysis. Similar to the HydroDyn wave generation, TurbSim uses seeds to influence the PRNG such that an identical time series may be generated on a different simulation.

4.2.3 MLife

MLife [75] is a fatigue calculation tool designed to post-process FAST time-series outputs. It calculates three fatigue adjacent factors. First is time until failure, which is the number of seconds the given member could survive. The second is lifetime damage, the fraction of damage taken during the designed lifespan. Lastly there is the damage equivalent load (DEL), the exact amplitude of the force or moment needed at a given frequency to cause equal damage to the VA time-series inputted.

The MLife input file does require some setup to correctly read the OpenFAST output file by setting the headers, units, times-series data, wind-series data and which data columns should be analysed for fatigue. MLife uses rainflow counting as described in Section 2.1.3 to calculate the load ranges. The load ranges are sorted into bins of user-defined size using a method similar to Miner's rule (Section 2.1.2). Damage is calculated by reading S-N curves (Section 2.1.1) for the given stress range. To compute the fatigue damage from a single channel MLife requires the logarithmic crack growth parameter (m), bin size, how to determine the load mean, and the ultimate load. The mean loads are required to apply a Goodman correction to the load. The ultimate load is used to determine the damage of a cycle.

Section 2.2.1 describes how specific wind parameters such as the mean wind velocity and standard deviation for turbulence remain stationary for a ten minute period. This characteristic is accepted by MLife when doing fatigue analyses as it can calculate the combined damage across multiple different time-series data. It calculates the mean wind and uses a Weibull distribution to apply a weight to the damage based on the probability of occurrence during the turbine lifetime. This means it requires at least two Weibull parameters, the shape factor and the scale factor. These are roughly correlated with the standard deviation and mean, respectively.

4.2.4 OpenFAST Matlab toolbox

The FAST team has developed an add-on [76] for Matlab that provides various useful functions and scripts. The primary functionality of the toolbox is to plot meshes, draw mode shapes, translate blade coordinates, and update FAST input files. The latter function proved the most useful for this thesis as it imports the FAST input files to Matlab as data structures that can be edited in scripts before the data structure is written to new FAST input files. This function opens up an efficient method to make minor edits to an existing template file. This is very practical when the analyst must run many slightly different OpenFAST simulations.

4.2.5 pyDatView

pyDatView [77] is a simple tool that displays plots of tabulated data. It was designed for use with FAST and provides a quick way to analyse plots of data. In addition to plotting the time-series data, it also can display the PDF and fast Fourrier transform (FFT). The PDF gives a graphical representation of the most common values in the plot as well as the distribution between them. pyDatView also provides the mean and standard deviation for the displayed plots. FFT is useful to see the frequency content of time-series. Frequency content is important for both the response of the turbine and turbine loading. The program can load in multiple FAST output files to plot the time-series, PDF or FFT within the same graph or in different subplots. When doing a free decay analysis, pyDatView can even estimate the damping of the system.

4.3 OpenFAST fatigue model

The OpenFAST models used in this thesis are based on reference OWTs. These reference turbines have downloadable OpenFAST input files, making for a quick setup. The reference monopile files for the International Energy Agency (IEA) Task 37 10 MW reference monopile [65] can be downloaded from GitHub [78]. Pedersen and Askheim kindly shared the jacket model [63]. Their model had the substructure of INNWIND's Deliverable 4.31 reference jacket [64] with the IEA Task 37 10 MW reference monopile on top of the jacket.

For this thesis, OpenFAST 3.0.0 was used [6, 67]. This version was the most recent main version at the beginning of this project, and it was backwards compatible with the dev version used by Pedersen and Askheim. That means the jacket model was usable with only a few tweaks. However, the monopile model was designed for an even older version of OpenFAST that was not directly compatible with v3.0.0. Instead of debugging and updating the old input files, it was elected to edit the files inherited from Pedersen and Askheim with the info from the IEA files. This methodology was chosen because the errors experienced when debugging the IEA files were not always intuitive to fix. Unless otherwise specified, the FAST models are assumed to be identical to those described above.

4.3.1 Choosing turbine modes

A key to calculating correct modal damping values is to have identical modes in OpenFAST and Abaqus. The modes in OpenFAST are not straightforward as the FAST framework couples several modules together. This means that OpenFAST uses different modes for different parts of the structure. Notably, the tower modes and the substructure modes are not handled by the same module when setting up OpenFAST the traditional way. ElastoDyn, which handles the tower modes, is designed for large tubular structures subject to geometric nonlinearities. SubDyn, which controls the substructure, can however include more than the first two modes in each direction. It was considered to model the tower in SubDyn to get greater coherence for the modes of the entire support structure. However, this idea was discarded as the C-B assumptions of a fixed TP would make those modes unusable without ElastoDyn. Therefore it was decided to model the OpenFAST modes in the traditional method, using ElastoDyn and SubDyn. The process is described in Section 4.3.1.1 and 4.3.1.2.

4.3.1.1 ElastoDyn modes

ElastoDyn has a rather tedious way to import mode shapes as described in sections 4.1.2 and 4.2.1. In brief, ElastoDyn does not calculate the mode shapes itself, but rather it requires a sixth-order polynomial description of the mode shapes as input. The model is defined in BModes, which makes the mode shapes, and then ModeShapePolyFitting.xls is used to translate the shapes to polynomial form. Luckily IEA had already provided the polynomial mode shapes for the monopile tower and turbine blades in their FAST model [65]. Pedersen and Askheim had also calculated the mode shapes and provided them for the tower of the jacket model [64]. The tower mode shapes were checked by recreating the model in BModes and putting in the new polynomials. These polynomials proved to be different from the ones provided. However, the tower response for the different polynomials is indistinguishable from each other. Therefore it was concluded that the polynomials provided for the jacket and monopile were correct. The calculated polynomials were discarded in favour of the ones provided. In retrospect, this may not have been the correct choice as any change in the stiffness matrix at the pile would yield a slightly different mode shape.

4.3.1.2 SubDyn modes

Both the jacket and the monopile substructure took too long to run to perform fatigue analyses with the full FEM DOFs. Therefore, the C-B method must be utilised to reduce the number of DOFs. Six Guyan modes are inherited from the TP, and any additional C-B modes may be chosen. Although additional modes in themselves do not directly increase computing time, they do decrease the time step needed for a stable solution. More modes lead to higher frequency modes, which lead to smaller time steps. Table 4.1 shows an overview of the modes and time steps for the different models. It is worth noting that the monopile frequencies increase much faster than the jacket, which means it can include much fewer modes. The jacket's more complex structure still gives a long simulation time despite larger steps.

Table 4.1: Overview of C-B modes for monopile and jacket
*Run time ratio is the amount of time OpenFAST can simulate in one second. Higher run time ratios means less time spent running simulations

Model		**Monopile**	**Jacket**
C-B modes	$[-]$	8	24
Max frequency	$[Hz]$	49.01602	8.73066
Time step	$[s]$	0.001	0.005
Run time ratio*	$[s/s]$	$<$0.008	$<$0.003

4.3.2 Model verification

Some tests were carried out to ensure that the models downloaded were behaving the same as those used by Pedersen and Askheim and IEA.

4.3.2.1 Jacket natural frequency

One of the simplest tests for the jacket is comparing the natural frequency. Pedersen and Askheim found the natural frequency of their jacket by removing the external loads and setting an initial displacement to perform a free decay analysis. That means disabling all modules except ElastoDyn and SubDyn. The gravity constant in ElastoDyn is set to zero, and the initial displacement of the tower top is set to 2.0 m. Additionally, only the DOFs relating to the tower movement are enabled.

Figure 4.4 shows the results of these simulations. The first mode is clearly visible in the time-domain displacements of the tower top translations. The second mode is only visible through the TP translations in the first few seconds of the analysis. Performing a FFT on the displacements clearly shows the structure's natural periods. The FFT results are shown in Figure 4.5.

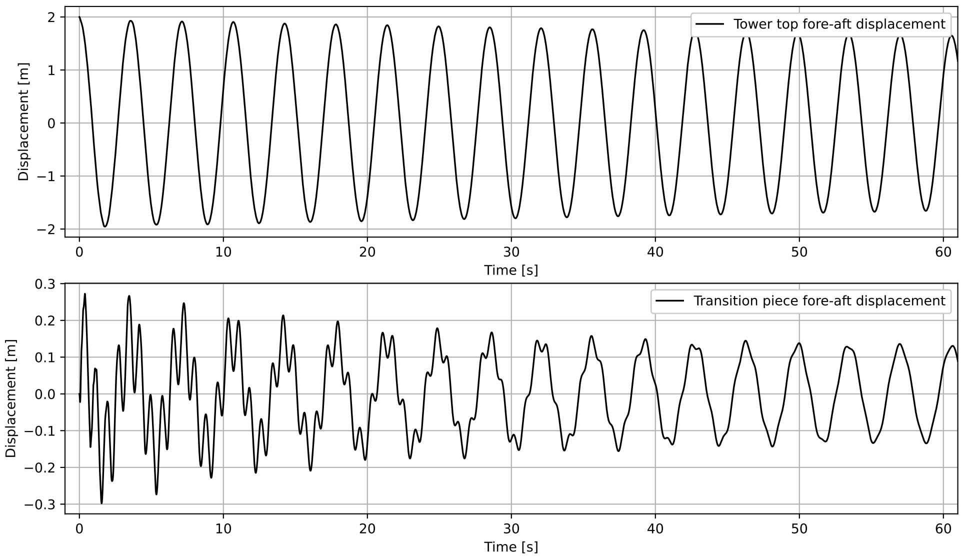


Figure 4.4: Jacket structure free decay in fore-aft direction

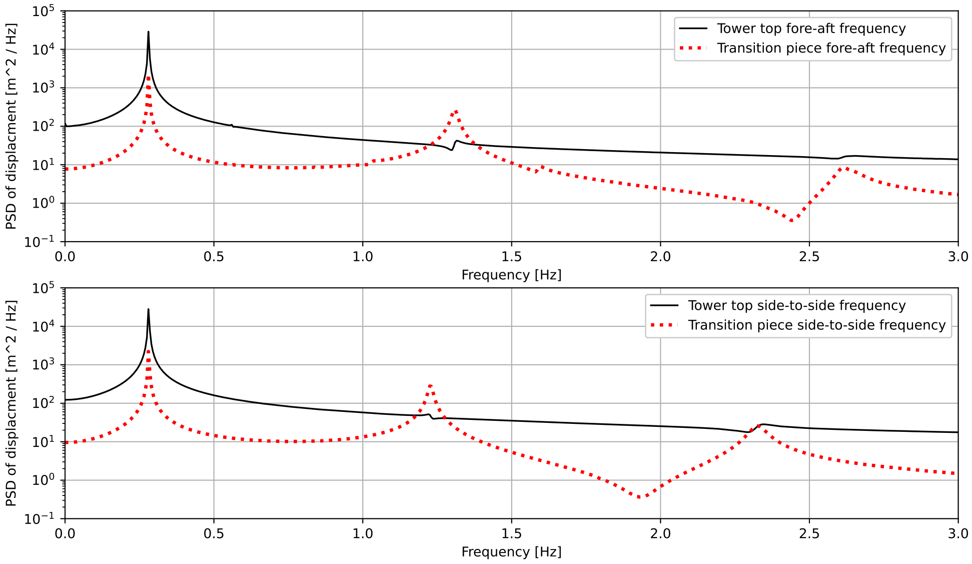


Figure 4.5: FFT of jacket structure during free decay

Table 4.2 shows a comparison of the results from the simulations to ensure the first four natural frequencies are in agreement. It is evident that the present model has a higher second natural frequency. Unfortunately, this discrepancy is hard to remedy as even the unaltered model from Pedersen and Askheim exhibits the same behaviour as the present model.

Table 4.2: Natural frequency comparison of jacket structure

Mode	**Pedersen and Askheim [63]** $[Hz]$	**Present model** $[Hz]$
First fore-aft	0.281	0.28
Second fore-aft	1.256	1.31
First side-to-side	0.280	0.28
Second side-to-side	1.174	1.23

Pedersen and Askheim also performed a linearisation analysis in OpenFAST, which gave the frequencies and mode shapes of the entire turbine, including the blades. These analyses were found to be quite time-consuming to set up and run. This is because linearisation requires the turbine to be in a steady-state. Only specific options within the input files are available when linearising, and even more must be altered to reach the steady-state in a reasonable amount of time. Without ideal conditions, it might take hours of simulation time to get linearisation results. These experiences led to the decision not to perform linearisation on the models.

4.3.2.2 Monopile natural frequency

The natural frequency of the monopile was found with the same methodology as for the jacket in Section 4.3.2.1. Only the first natural frequency was provided by IEA and was given as 0.24 Hz. The FFT provided in Figure 4.6 shows the natural frequencies. It is clear from the plot that the first natural frequency is 0.24 Hz in the fore-aft direction.

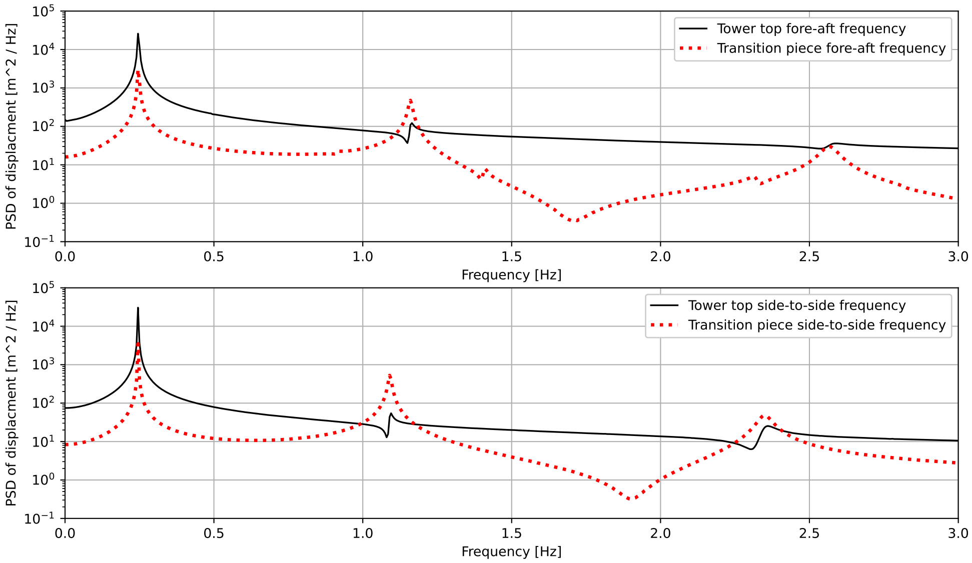


Figure 4.6: FFT of monopile structure during free decay

4.3.3 Environmental load generation

For this research, it was desirable to find some measured data to base the environmental conditions on. Therefore, the measurements by Fischer et al. [47] during the European research project UpWind were chosen. The site in question is the K13-alpha 3 deep water site in the Dutch North Sea. Fischer et al. have reported 22 years of measured wind and wave data at the site. The wind was measured at the height of 90.55 m above sea level, and the waves were measured at 25 m depth. However, the waves were verified to be depth independent, so a 50 m deep site was also presented.

The most important wind and wave parameters from table 59 from Fischer's report are repeated in Table 4.3.

Table 4.3: Wind and wave parameters from table 59 in UpWind report [47]

Wind velocity $[m/s]$	**Wave height** $[m]$	**Wave period** $[s]$	**Probability of occurrence** $[-]$
2	1.07	6.03	0.06071
4	1.10	5.88	0.08911
6	1.18	5.76	0.14048
8	1.31	5.67	0.13923
10	1.48	5.74	0.14440
12	1.70	5.88	0.12806
14	1.91	6.07	0.10061
16	2.19	6.37	0.07554
18	2.47	6.71	0.04878
20	2.76	6.99	0.03151
22	3.09	7.40	0.01924
24	3.42	7.80	0.00977
26	3.76	8.14	0.00474
28	4.17	8.49	0.00243
30	4.46	8.86	0.00093
32	4.79	9.12	0.00053
34-42	4.90	9.43	0.00019

4.3.3.1 Wind load generation

TurbSim was used to generate the wind field for the FAST simulation. The IEC Kaimal spectrum was chosen to model the turbulence. Most parameters were left at the IEC 61400-3 offshore wind turbine default, with the exceptions described in this paragraph [54]. The turbine class was set to class B and the turbulence model was set to NTM. Out of fear that TurbSim would not pick an appropriate roughness parameter, it was set as the parameter specified in the UpWind report ($z0 = 0.002$). TurbSim had to generate 17 different wind fields, one for each mean wind velocity provided by Fischer et al. The physical size of the wind fields was set to be big enough to envelop both the monopile and the 16 m taller jacket OWT. Accurate wind fields for the jacket were achieved by setting the reference wind height to the hub height of the K13 turbine ($RefHt = 90.55$) and setting the hub height of the jacket ($HubHt = 131.63$). The hub height of the taller jacket must be used because TurbSim automatically extends the excess vertical wind field downwards.

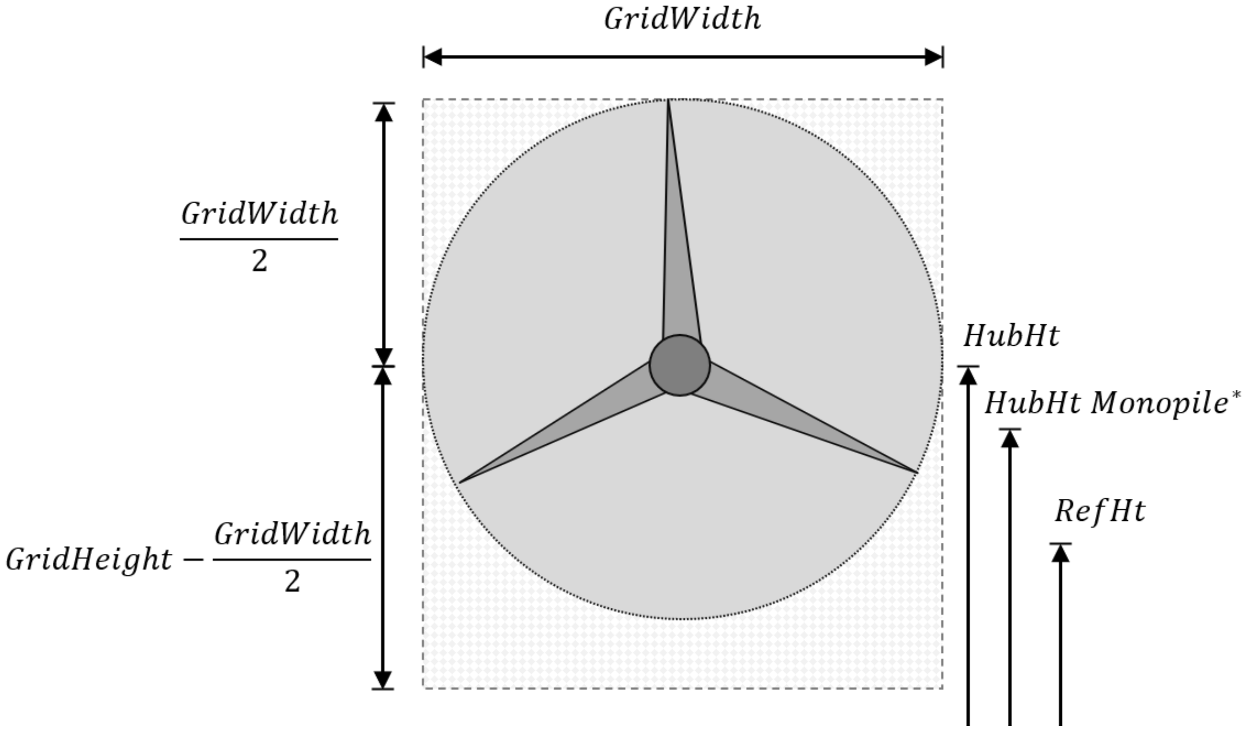


Figure 4.7: TurbSim grid geometry
*Monopile hub height is not used, but difference between jacket hub and monopile hub is the amount the wind field is extended below the swept area of the blades

The TurbSim wind field was imported into the FAST framework through InflowWind. Since TurbSim has all the information InflowWind needs, a simple file path to the TurbSim .bts file was the only input needed.

4.3.3.2 Wave load generation

Wave geometry was specified in HydroDyn. Wave height H_S and period T_p were set in within the waves section of HydroDyn. A JONSWAP spectrum for the wave representation was picked with a peak enhancement factor equal to the default from IEC 61400-3 [54]. The wave spectrums for some of the load cases are shown in Figure 4.8. The same wave period and height had to be chosen for the monopile and jacket despite the distinct water depths of the models. The fidelity of the waves was also increased to a time step of 0.01 s.

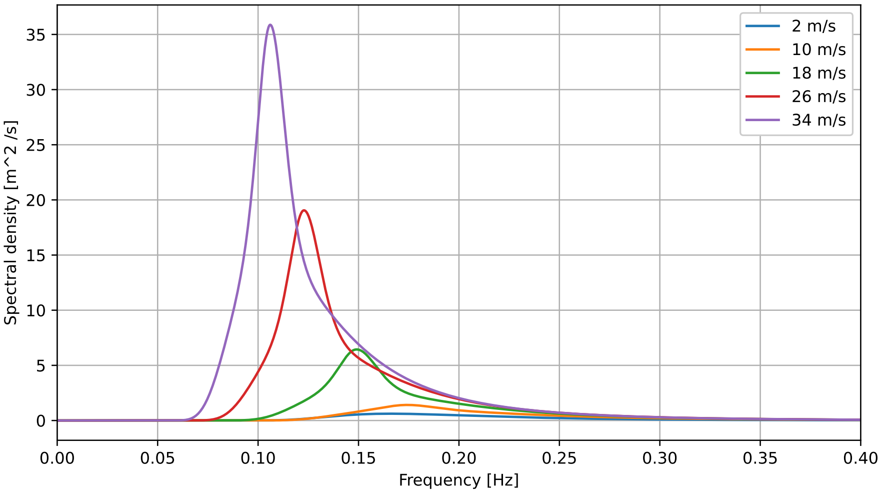


Figure 4.8: JONSWAP spectrum for various mean wind velocities

4.3.4 Damping specification

There are two primary modules where damping may be applied ElastoDyn Tower for the tower and SubDyn for the substructure. In ElastoDyn Tower, damping can only be applied to the first four modes. The four modes must be the two fore-aft modes and the two side-to-side modes. A damping ratio can be provided for each of the four modes. In SubDyn, damping can only be applied to the substructure. Damping is applied to either the Guyan or the C-B modes through damping ratios, Rayleigh damping and a damping matrix. A complete guide to SubDyn damping is described in Section 4.1.7. Blade damping may be expressed in ElastoDyn blade. However, this was left unchanged from the IEA default throughout the simulations.

A damping ratio of 1.0% was chosen as the base damping from all structural damping sources, excluding soil damping. This damping was applied to all four ElastoDyn modes, all retained SubDyn C-B modes and the Rayleigh damping of the Guyan modes. PyDatView's decay analysis function can estimate the first mode damping by performing a free decay analysis as described in Section 4.3.2.1. Applying 1.0% damping in the OpenFAST modules gave almost 1.2% damping in the first mode of both the fore-aft and side-to-side mode. Several discoveries were made through iterating the damping of various modes. The first one is that the first ElastoDyn modes had the most significant impact on the damping. Secondly, setting the damping of all modes to 0.8% yielded 1.01% damping. This configuration was used as the baseline damping for future analyses. This method was also employed to correct the first mode damping when additional damping was applied.

Soil damping was applied using a similar method described in the above paragraph. However, soil damping is not uniform across all modes. An approach to apply accurate damping in modes higher than the first in each direction must be employed because the damping is not measurable in a free decay test. For the monopile model, it was decided to find the correct damping ratio for the first mode in each direction. Then modify the following modes so that the proportion between the damping ratios of each mode remains the same. Meanwhile, the vibration modes directions of the jacket substructure in FAST do not align with the those present when calculating the modal damping in Abaqus. Therefore, it was chosen to use the weighted average soil damping in the fore-aft direction for all modes in SubDyn. Section 6.1 and 6.2 provides values for soil damping of the monopile and jacket structure and a brief description of the weighted average.

4.3.5 Matlab fatigue setup

Due to the sheer number of simulations that needed to be run, a Matlab script utilising the FAST Matlab toolbox was established to create and update input files. The jacket model was set as the template file, with the

Matlab script applying any changes that needed to be performed. The changes can be roughly divided into three different groups. They are parameters dependent on the substructure, damping value and wind velocity. The substructure dependent parameters include placement of various turbine parts, tower mode shapes, water depth, substructure geometry and substructure discretisation. Damping dependent parameters are the damping of ElastoDyn Tower, C-B and Guyan modes. Wind velocity dependent parameters include paths to TurbSim .bts files, blade rotation and pitch, wave height and period, and AeroDyn effects.

Some of the functions within Matlab had to be altered to be compatible with the chosen version of OpenFAST. This change was because some of the new and less standardised FAST inputs were not always correctly translated from FAST to Matlab and back to FAST. Although lots of time and frustration was involved with setting up the Matlab script, it proved valuable when modifying files in bulk. There were hundreds of input files despite creating them with the most efficient file structure. The Matlab script can, within seconds, make the necessary changes to them without needing to worry about any human error made when manually navigating between them.

4.4 OpenFAST ULS model

The monopile from Section 4.3 was used as a base for a new OpenFAST model subject to an extreme weather event. Since the monopile structure is meant to be the same, only the environmental conditions were changed. The assumed loading condition is a storm with a 50 year return period not too different from DLC 6.1 from DNV-ST-0437 [42]. Unlike DLC 6.1, the modelled turbine will not have any yaw misalignment or skewed wind and wave directions.

Updates to the TurbSim input file included changing the turbulence characteristic from B to A. The change gives a more turbulent wind field. Another change was to implement a 50-year extreme wind model (EWM) for turbine class I as opposed to a NTM. The next change was to set the mean wind velocity to the IEC 61400-1 standard for class I ($URef = 50$) [51]. And finally, some of the heights and grid sizes were changed to fit the monopile better. In HydroDyn, the only changes made were to the wave height and period. The values from the UpWind report were used ($WaveHS = 9.4$ and $WaveTp = 10.87$) [47]. No changes were made to the other modules besides utilising the input file for a parked rotor.

4.5 MLife configuration

Some of the MLife input parameters are challenging to obtain. Therefore, the setup and assumptions made will be described in this section.

4.5.1 Choosing loads

MLife uses load-series to predict the fatigue damage. Loads for up to six DOFs can be obtained for specific nodes in the OpenFAST model. Some have predefined names, such as at the tower top and TP. Other nodes must be specified by the analyst, such as nodes along the tower or substructure. However, the number of nodes that can be outputted is limited. This limitation is an issue for the jacket substructure, which has hundreds of potential nodes which may be critical for the durability of the substructure. Jacket forces and moments were sampled at eight nodes near the bottom of the substructure: one node at each of the four legs, three nodes on the braces and one on a column leg. Force and moment at the bottom of the tower (top of the TP) were also sampled. The much more simplistic monopile only needs to be sampled where the forces are highest (mudline) and where the structure is weakest (TP). At the TP and at the monopile mudline: axial, shear and moment are only observed in the most critical directions. Meanwhile, for the jacket substructure: axial, shear and moment forces are checked in most directions. Load roses are not used to expand the number of directions. However, shear forces were added to get a more conservative estimate for that force in particular. Torsions about the axial axis were not considered as closed cross-sections have a high twisting moment capacity.

There are various ways to obtain the ultimate load ($LUlt$) used to determine the total durability of the section that the load series refers to. The recommended approach is to perform a FEA for the given load. Another approach is to increase the ultimate load until the DEL asymptomatically reaches its true value. For research meant to compare fatigue lives, it has also been observed that analysts set the ultimate load to a value that gives the desired durability. However, since these methods can be time-consuming and have varying degrees of accuracy, it was decided to simply use the yield strength of steel to determine the ultimate load. This method is simple, understandable and not too far from the real solution. INNWIND [64] uses S355 steel on their jacket and while IEA [65] does not specify steel quality. S355 steel with a safety factor of 1.05 was chosen to determine the elastic yield strength of the cross-sections. Equation (4.2) to (4.4) are based on the axial strength, shear strength, and moment resistance of a circular cross-section in Eurocode 3 [79].

$$LUlt_{axial} = Af_{yd} = \pi \left(r_o^2 - r_i^2\right) f_{yd} \tag{4.2}$$

$$LUlt_{shear} = \frac{A_v f_{yd}}{\sqrt{3}} = \frac{\frac{2A}{\pi} f_{yd}}{\sqrt{3}} = \frac{2}{\sqrt{3}} \left(r_o^2 - r_i^2\right) f_{yd} \tag{4.3}$$

$$LUlt_{moment} = \frac{I}{r_o} f_{yd} = \frac{\pi}{4} \frac{r_o^4 - r_i^4}{r_o} f_{yd} \tag{4.4}$$

where A is the cross sectional area, I is the second moment of area, f_{yd} is the design yield strength, r_o is the outer radius of the given member, and r_i is the inner radius of the member.

4.5.2 Weibull distribution

MLife requires certain Weibull parameters to determine the probability of each load case occurring. These parameters are the scale factor, shape factor and distance between sampling points. These are given by the UpWind report [47]. However, these are for the wind velocity at the reference height. The reference height is not insignificantly lower than the hub heights of the monopile and jacket. That makes the parameters in the report inaccurate for this purpose as MLife would calculate the wind velocity as slightly higher than the reference wind velocity. There was not any tool that would accurately recreate a Weibull distribution from so few data points and with a varying step length between wind velocity bins. The Weibull parameters had to be estimated based on a graphical representation. By plotting a Weibull distribution with various parameters, it was decided to keep the shape factor at 2.04, but increase the scale factor from 11.75 to 12.50 and increase the jumps between wind velocity bins from 2 to 2.12. This distribution was deemed acceptable for both the monopile and the jacket. Relevant distributions can be seen in Figure 4.9 and 4.10.

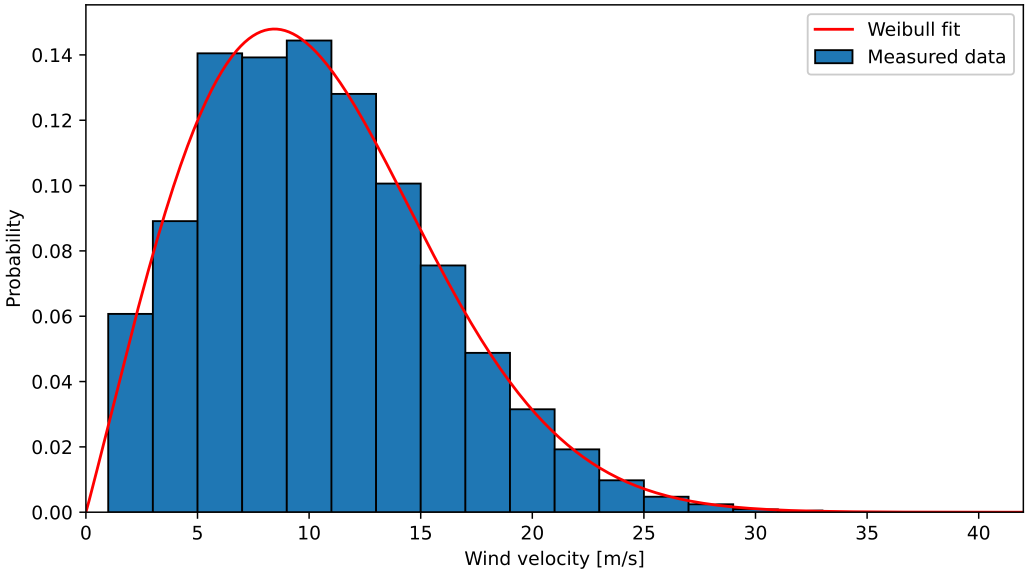


Figure 4.9: Weibull distribution for wind velocities at deep water site. Reproduced from Fischer et al. [47]

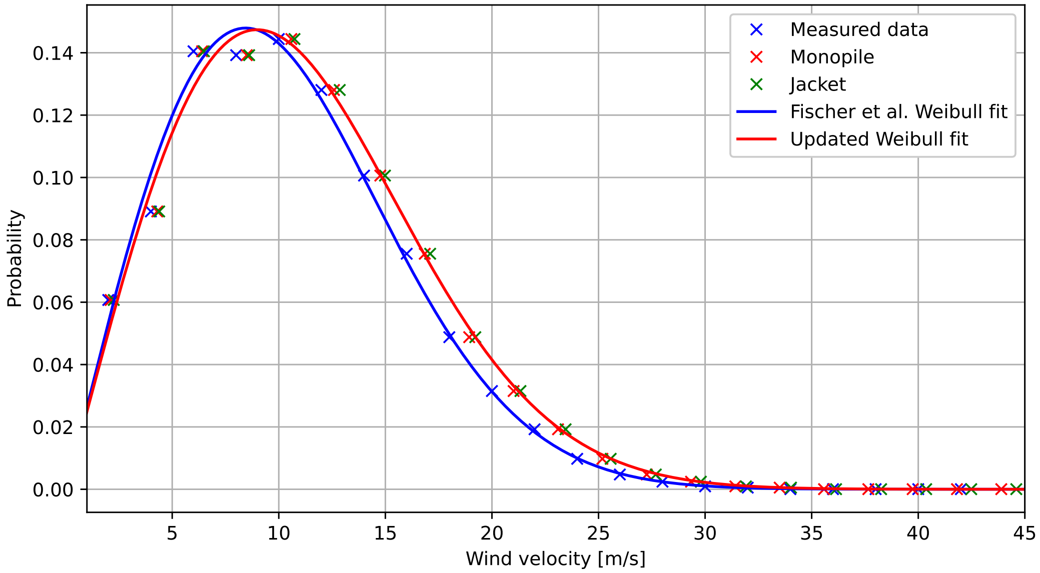


Figure 4.10: Wind velocity Weibull distribution for taller structures. Partially reproduced from Fischer et al. [47]

4.5.3 Other MLife parameters

The cut-in, cut-out and maximum wind velocity were set to the values from the IEA report [65]. Rainflow binning was disabled as the MLife fatigue analysis remained brief, despite the extra calculations. The Goodman correction was applied to account for the tension or compression state of the load series. The design life of the turbines was assumed to be 20 years. DELs were calculated assuming they were applied at 1 Hz. The S-N logarithmic slope factor (m) was set to 3, 4 and 5. This decision is due to DNV's recommendation to use 3 for low cycle fatigue loads and 5 for high cycle fatigue loads [80]. Predicted durability is reported for $m = 5$ since loads at the first natural frequency are expected to occur over 10^8 times during the turbine lifetime. This suggests that most loads will be within the high cycle fatigue range.

5. Modal analysis

The primary goal of the modal analysis was to find the properties of each mode needed for the estimation of the soil damping, including:

- The dynamic excitation in the foundation in each mode
- The dynamic energy in each mode

Because of the necessity to find non-standard sizes, such as each mode's dynamic energy and physical response, a different approach than the standard analysis in Abaqus was used. The mass and stiffness matrix was exported from the Abaqus model to a Python script performing the actual modal analysis. The SSI-stiffness matrix' off-diagonal terms could not be included directly in Abaqus. Therefore, the SSI-stiffness-matrix was added directly into the stiffness matrix in the foundations in the Python script. The nodes of interest were defined as a data set in Abaqus and could be read directly from the Abaqus input file.

5.1 Loading

The only loading considered in this analysis was the loading from the turbine. The loading was applied as concentrated force and moments to the top node of the tower. The loads were collected from a FAST analysis for the jacket and monopile -model with a mean wind speed of 12 m/s. For details, see Section 4.3.3.

The loading was then split into one dynamic and one static part. The static part was the mean of the load series, while the dynamic part was the original load series subtracted by the mean. The dynamic load was, in turn, written as a sum of sinusoidal terms using Fourier series. The number of terms was 2000 for the monopile and 2700 for the jacket construction. The number of terms made the Fourier series include terms up to 3 and 4 Hz for the monopile and jacket, respectively. From Figure 5.1 and 5.2 it is clear that frequencies higher than 3 Hz and 4 Hz have little impact on the movement of the foundation. Therefore, including higher frequencies, parts from the loading would have had minimal effect on the damping estimation.

The modal analysis did not take gravity into account. As gravity is a static load, the effect of gravity could be accounted for by applying the average displacement from the FAST analysis instead of the average displacement from the author's modal analysis. This was done to calculate the static response for the top of the pile. The static response from FAST would also include any other static effects from the FAST analysis.

5.2 Response computation

The dynamic response was calculated using a modal analysis with 1.00% damping and Fourier series loading. The particular solution was found by applying the loading of each frequency, calculating the frequency response, and summing up responses. This approach did not take phases into account. An attempt was made to input the phases by using complex numbers, but the computational cost was too big. The modal analysis approach was meant only to calculate a typical level of displacement per mode and was sufficient for this purpose, even without faces. The physical response was calculated for each mode before a signal analysis was performed to locate the peaks. The peaks were then averaged to give an average modal peak height in the physical domain.

The average peak height is used as the basis to calculate the damping as described in Section 2.5.1. The average peak height is also used to calculate the modal dynamic energy Π_{TDi}.

The scripts can be found in Appendix C.

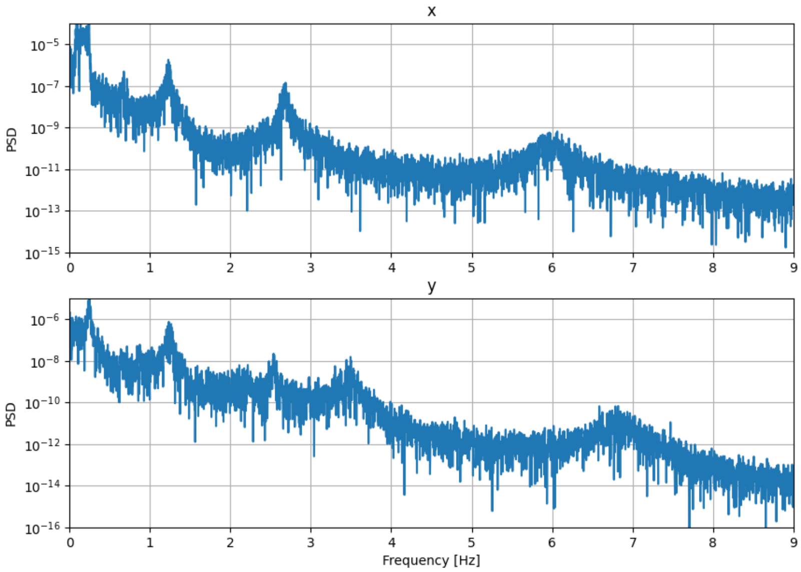


Figure 5.1: Power spectral density of displacement from FAST in the foundation of the monopile in the x and y-direction. 12 m/s loading

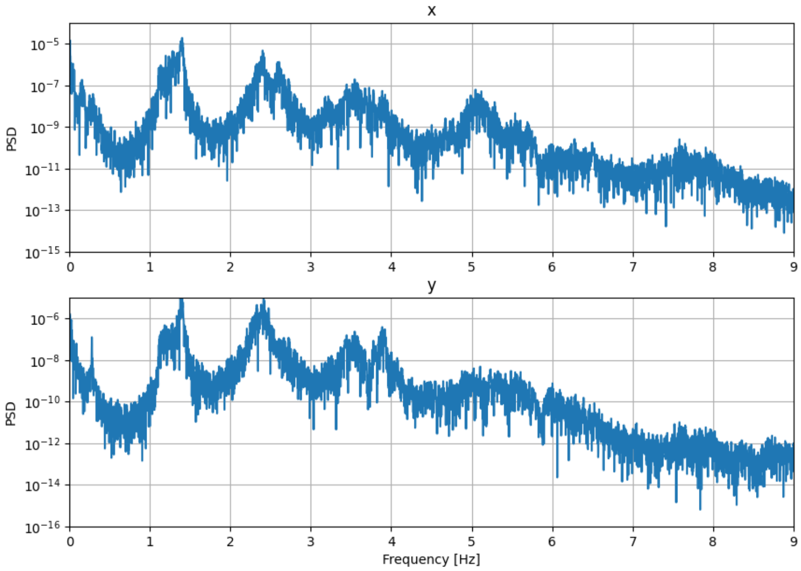


Figure 5.2: Power spectral density of displacement from FAST in the foundation of the jacket in the x and y direction. 12 m/s loading

5.3 Verification

To verify the implementation, the calculated modal response from the modal analysis was compared to the response calculated in FAST for the monopile and jacket structure in Figure 5.3, 5.4, 5.5 and 5.6. The figures show a good fit between the response calculated in the modal analysis and FAST. The modal response is more high-frequent than the FAST response. This is possibly caused by the aerodynamic damping of the rotors damping out high-order frequencies in the FAST-analysis.

The difference between the FAST analyses, including waves and gravity, is relatively small for the jacket. This may indicate that the waves have a relatively small impact on the tower's movement. The difference between the FAST analysis, including and excluding waves and gravity, is larger for the monopile structure.

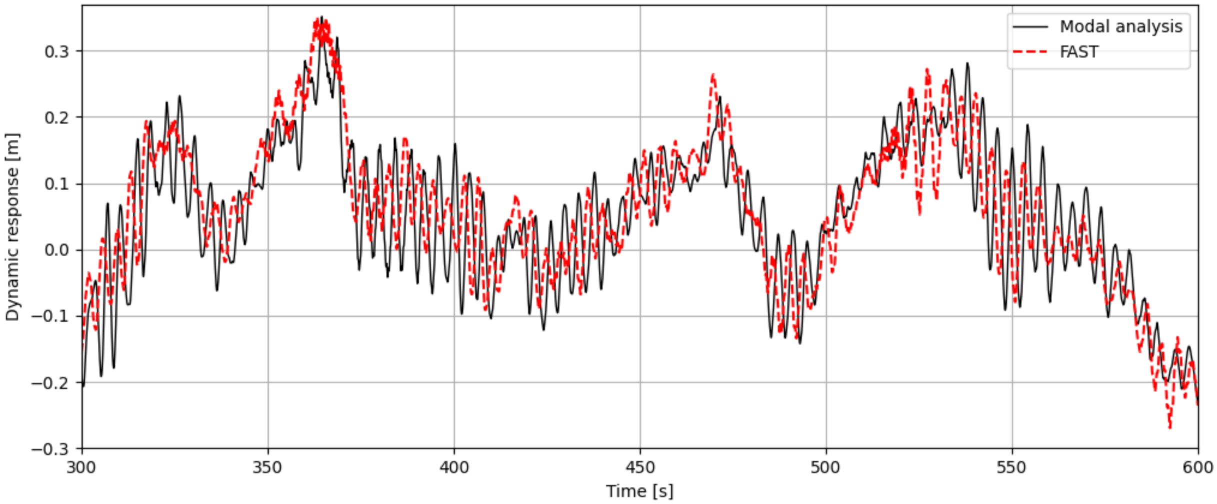


Figure 5.3: Dynamic response at the tower top for monopile structure in the x-direction. Comparing results between FAST and modal response implementation in this study. Only wind is included in modal implementation. Wind, waves and gravity are included in FAST. Wind speed = 12m/s

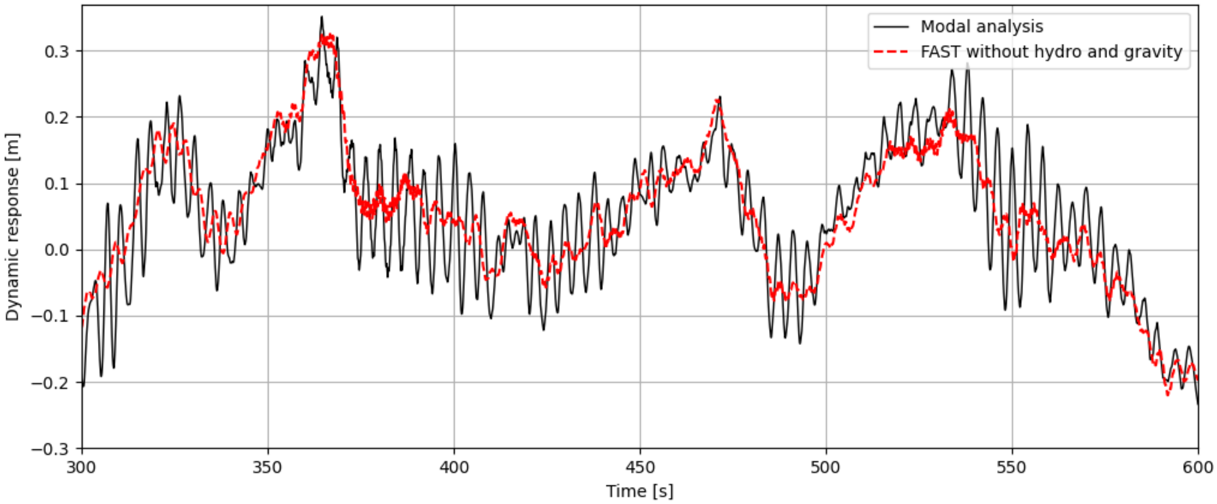


Figure 5.4: Dynamic response in top of tower for monopile structure in x-direction. Comparing results between FAST and modal response implementation in this study. Only wind is included in modal implementation. Only wind is included in FAST. Wind speed = 12m/s

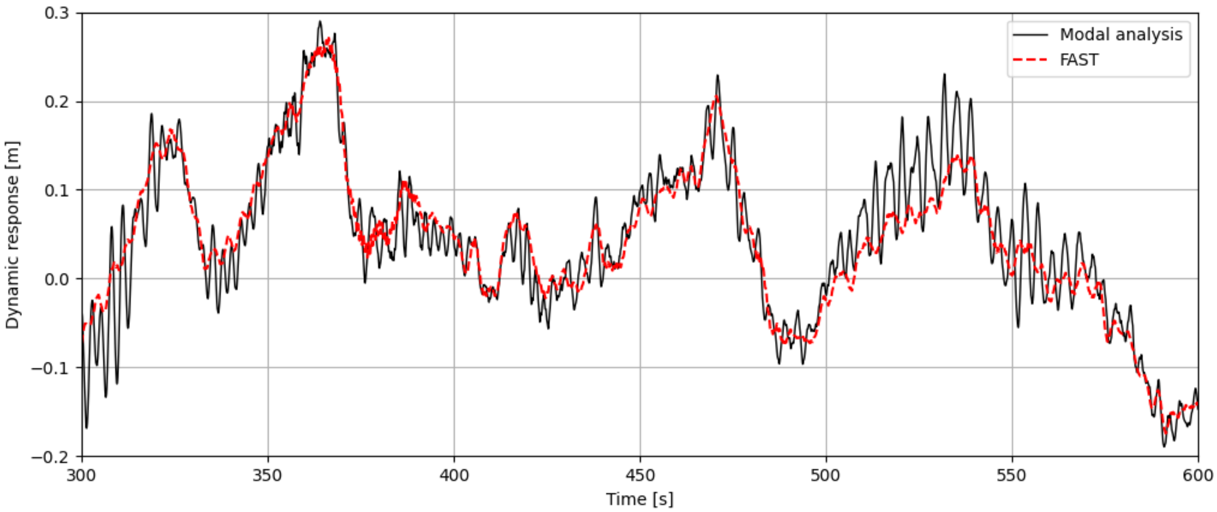


Figure 5.5: Dynamic response in top of tower for jacket structure in x-direction. Comparing results between FAST and modal response implementation in this study. Only wind is included in modal implementation. Wind, waves and gravity are included in FAST. Wind speed = 12m/s

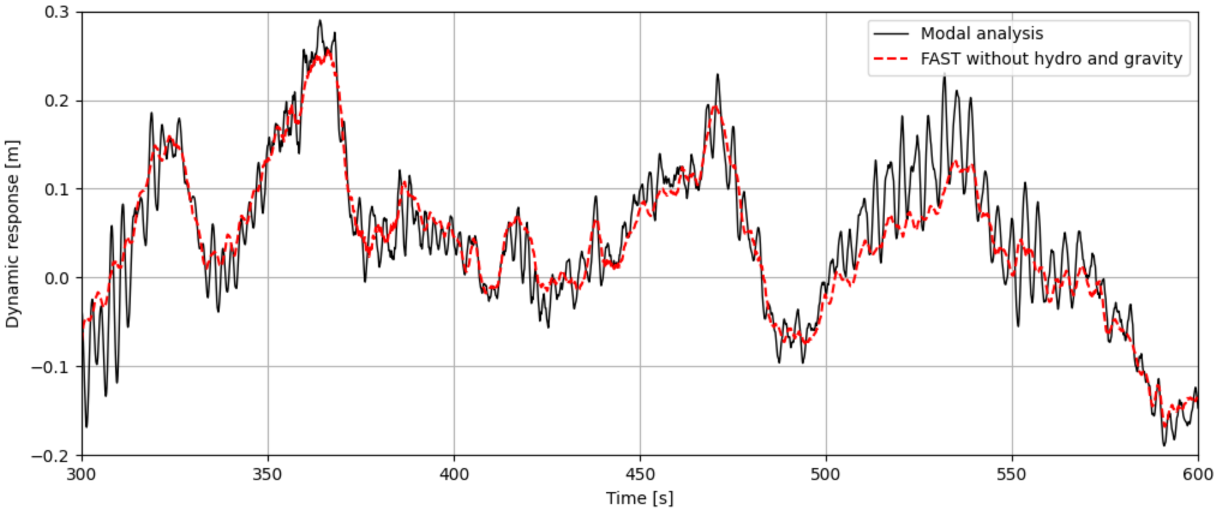


Figure 5.6: Dynamic response in top of tower for jacket structure in x-direction. Comparing results between FAST and modal response implementation in this study. Only wind is included in modal implementation. Only wind is included in FAST. Wind speed = 12m/s

5.4 Data from monopile analysis

The static response from FAST in the foundation is given in Figure 5.7. The figure shows that the primary displacement is in the fore-aft direction in addition to some vertical displacement. The side-to-side displacement is negligible. From Figure 5.8 showing the average peak in the foundation, it is clear that also the dynamic response is dominated by the fore-aft modes. By analysing the dynamic energy per mode in Figure 5.9 combined with Figure 5.8, it is clear that the system is dominated by the fore-aft modes and a yaw mode.

The complete data set for modal response in the foundation can be found in Appendix A.

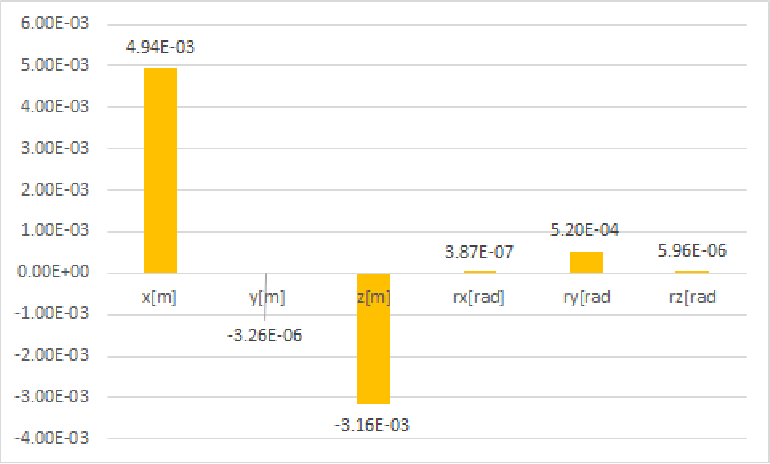


Figure 5.7: The static displacement in the foundation for the monopile. Units are meters for displacements and radians for rotations.

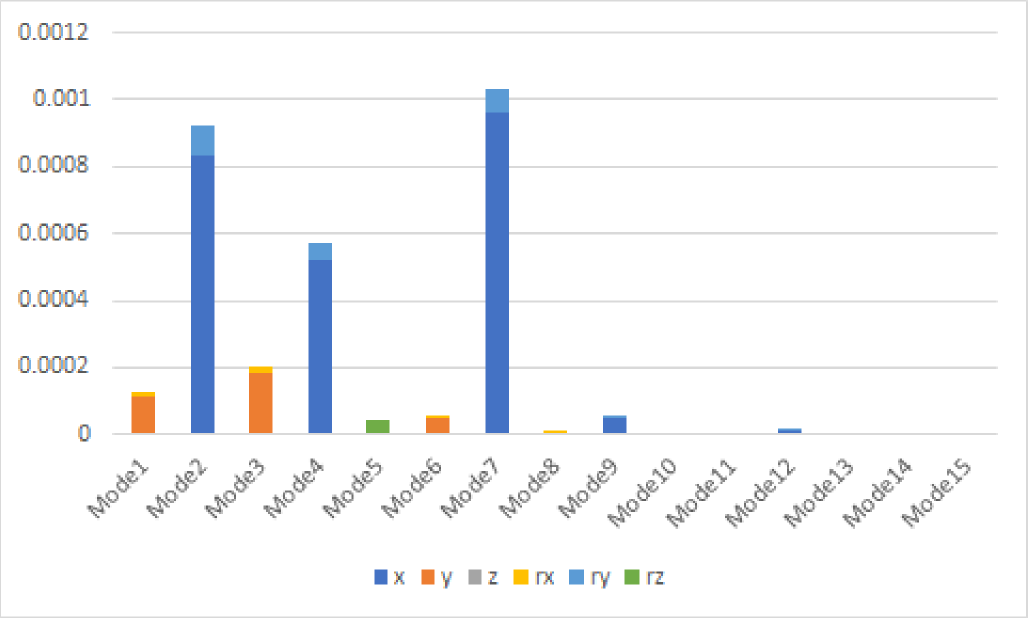


Figure 5.8: The average peak displacement in physical space for monopile modes at foundation. Units are meters for displacements and radians for rotations.

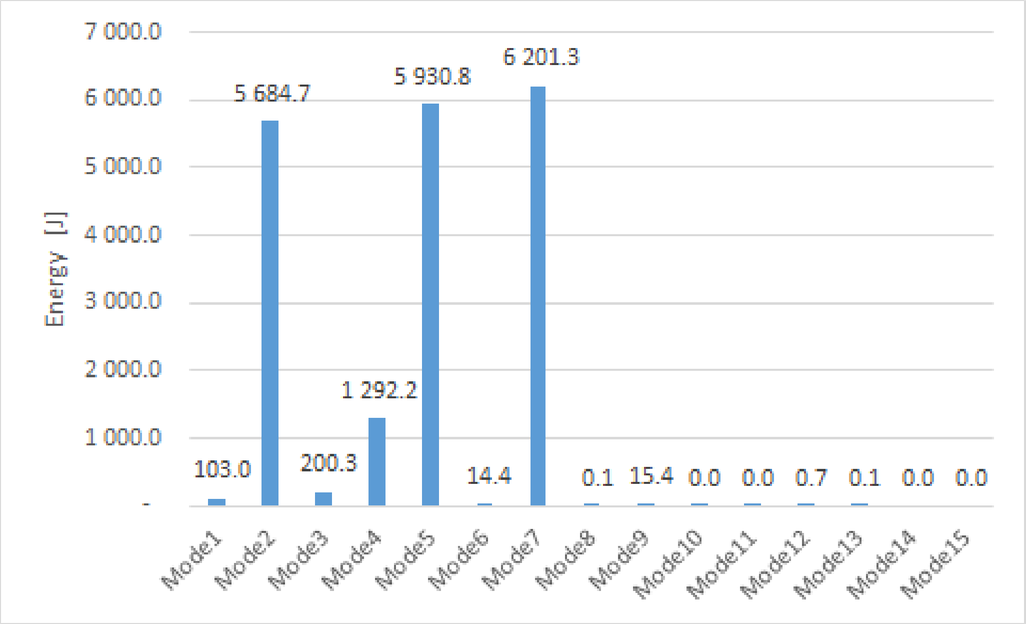
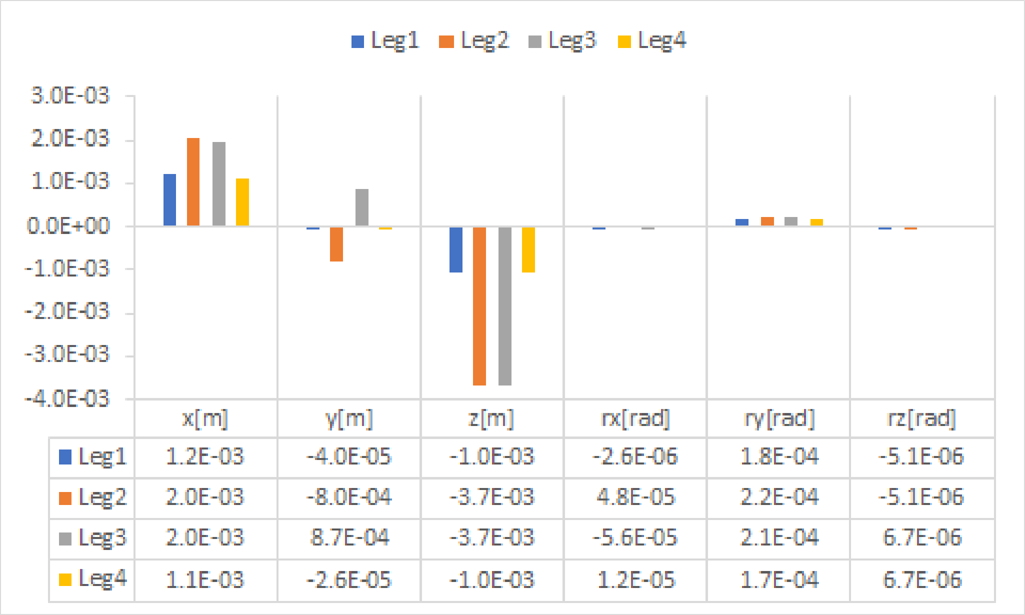


Figure 5.9: The dynamic energy held by each mode in the monopile structure. The energy is calculated from the average peak in the fore-aft-direction.

5.5 Data from jacket analysis

The static response from FAST in the foundation for each of the legs is given in Figure 5.10. The figure shows that much of the static displacement is in the fore-aft-direction. The difference between the vertical displacements is believed to be caused by the tilting movement of the jacket. All piles are expected to sink some due to gravity.

The absolute value of the average peak displacements for the different modes for Leg1 is shown in Figure 5.11. The absolute value of average peaks does not vary between the different legs to any major extent. Nevertheless, the signs do vary in accordance with the vertical movement associated with rocking. The dynamic energy in each of these modes is shown in Figure 5.12.



	x[m]	y[m]	z[m]	rx[rad]	ry[rad]	rz[rad]
■ Leg1	1.2E-03	-4.0E-05	-1.0E-03	-2.6E-06	1.8E-04	-5.1E-06
■ Leg2	2.0E-03	-8.0E-04	-3.7E-03	4.8E-05	2.2E-04	-5.1E-06
■ Leg3	2.0E-03	8.7E-04	-3.7E-03	-5.6E-05	2.1E-04	6.7E-06
■ Leg4	1.1E-03	-2.6E-05	-1.0E-03	1.2E-05	1.7E-04	6.7E-06

Figure 5.10: The static displacement in the foundation of the jacket. Units are meters for displacements and radians for rotations.

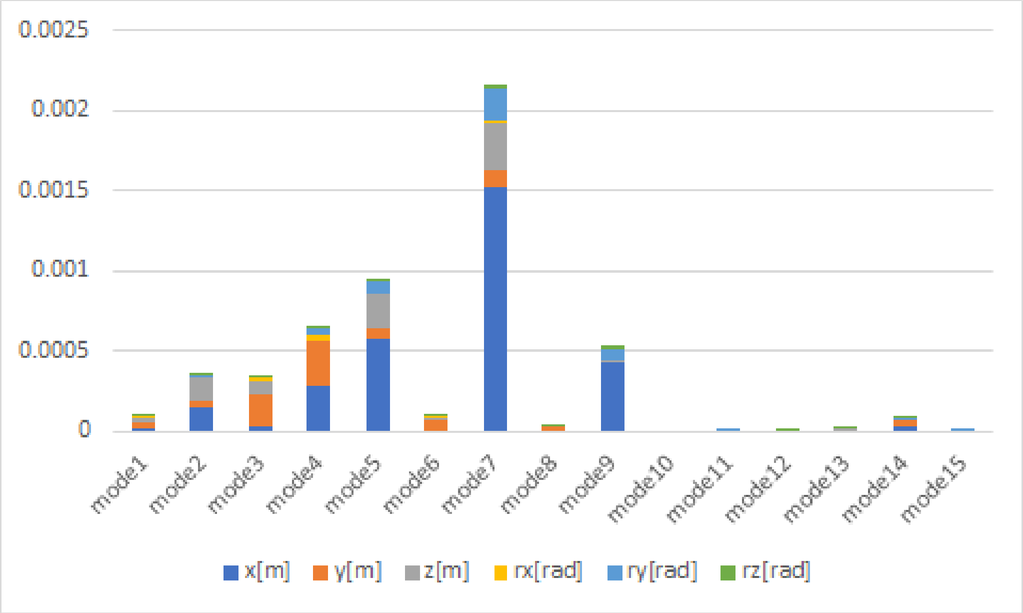


Figure 5.11: The absolute value of average peaks in physical space for one of the jacket legs. Units are meters for displacements and radians for rotations. The absolute value of average peaks do not vary in any major extent between the different legs. But the signs do vary in accordance with the vertical movement associated with rocking.

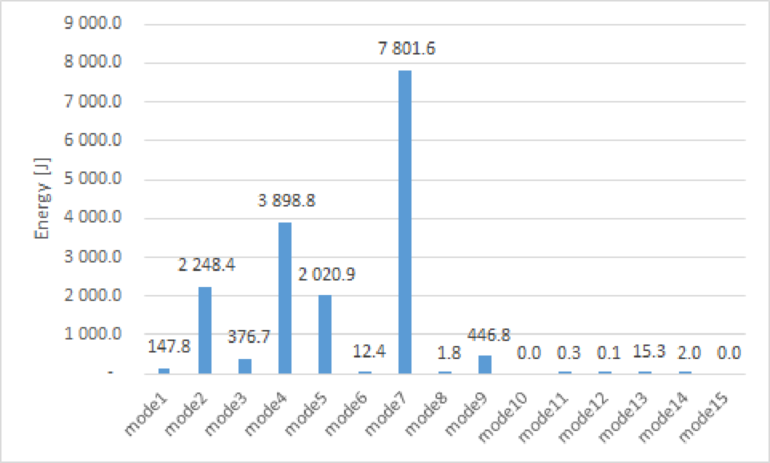


Figure 5.12: The dynamic energy in each mode in the jacket structure.

6. Results and discussion

6.1 Estimated damping for monopile

This section will present the results for soil damping for the monopile structure.

6.1.1 Hysteric damping - Method 1

The seven first modes are chosen for the basis of the analysis. The results are summarised in Table 6.1. These modes are the only ones associated with enough energy that there is any sense to do hysteric damping estimations. However, the side-to-side modes have a low level of dynamic energy. The low level of dynamic energy cause even a very low energy loss per cycle to result in a significant level of damping. Therefore, these results have to be interpreted with caution.

The weighted average is based on the seven first modes' dynamic energy Π_{Di} and hysteric damping D_i. The weighted average is calculated as $\frac{\Sigma \Pi_{Di} D_i}{\Sigma \Pi_{Di}}$. Giving a total weighted average of:

- 2.39% including all modes
- 3.50% including only fore-aft modes
- 0.21% including only side-to-side modes

Table 6.1: Hysteric soil damping estimated for monopile structure using method 1

Mode	**Total loss per cycle [J]**	**Static loss per cycle [J]**	**Dynamic loss per cycle [J]**	**Dynamic energy total structure [J]**	**Damping ratio [%]**	**Frequency [Hz]**	**Mode type**
1	20 149	20 148	1.3	103	0.10	0.252	Side-to-side
2	22 256	20 148	2 108.4	5 685	2.95	0.253	Fore-aft
3	20 155	20 148	6.7	200	0.27	1.063	Side-to-side
4	21 431	20 148	1 283.3	1 292	7.90	1.164	Fore-aft
5	20 180	20 148	31.7	5 931	0.04	1.249	Yaw
6	20 148	20 148	0.2	14	0.11	1.980	Side-to-side
7	22 558	20 148	2 409.5	6 201	3.09	2.245	Fore-aft

The side-to-side modes contribute little to energy loss in the soil, resulting in low levels of hysteric-soil damping compared to the fore-aft modes.

6.1.2 Relationship between foundation and total damping - Method 2

The relationship between the combined damping of the foundation and the damping of the entire structure is calculated using method 2. Setting a unit foundation damping $D_f = 1$. The results for the monopile structure are displayed in Table 6.2.

First, the unit damping of 1 was applied to both the foundation excluded structure and the foundation to verify the correct implementation. This resulted in total damping of 1 per mode, equal to the overall damping applied. Indicating a correct implementation.

Table 6.2: The foundation damping as share of total damping per mode for the monopile structure. Calculated using method 2

Mode	**Frequency [Hz]**	**Foundation damping as share of total damping [%]**
1	0.252	22.6
2	0.253	23.0
3	1.063	24.3
4	1.164	29.7
5	1.249	3.5
6	1.980	20.6
7	2.245	16.4
8	4.042	15.6
9	4.163	14.4
10	5.388	36.8
11	7.603	13.5
12	7.649	13.4
13	7.921	46.5
14	10.845	40.1
15	12.042	12.1

6.2 Estimated damping for jacket

In this section, the damping calculated for the jacket structure is presented. This includes hysteric damping, radiation damping and the relationship between the foundation and the entire structure.

6.2.1 Hysteric damping - Method 1

The hysteric damping of the jacket structure is calculated per mode using method 1. The results can be found in Table 6.3. The nine first modes are chosen for the basis of the analysis. These modes are the only ones containing enough energy to justify performing hysteric damping estimations. However, the side-to-side modes have a low level of dynamic energy. The low level of dynamic energy cause even a very low energy loss per cycle to result in a significant level of damping. Therefore, these results have to be interpreted with caution.

The weighted average is based on the nine first modes' dynamic energy Π_{Di} and hysteric damping D_i. The weighted average is calculated as $\frac{\Sigma \Pi_{Di} D_i}{\Sigma \Pi_{Di}}$. Giving a total weighted average of:

- 2.44% including all modes
- 3.29% including only fore-aft modes
- 0.29% including only side-to-side modes

Table 6.3: Hysteric soil damping estimated for jacket structure using method 1

Mode	**Total loss per cycle [J]**	**Static loss per cycle [J]**	**Dynamic loss per cycle [J]**	**Dynamic energy total structure [J]**	**Damping ratio [%]**	**Frequency [Hz]**	**Mode type**
1	11 075	11 073	2.4	148	0.13	0.272	Side-to-side
2	12 062	11 073	989.6	2 248	3.50	0.274	Fore-aft
3	11 087	11 073	14.4	377	0.30	1.176	Side-to-side
4	11 086	11 073	13.3	3 899	0.03	1.185	Yaw
5	12 686	11 073	1 612.8	2 021	6.35	1.299	Fore-aft
6	11 075	11 073	2.4	12	1.52	2.005	Side-to-side
7	13 620	11 073	2 546.9	7 802	2.60	2.220	Fore-aft
8	11 073	11 073	0.1	2	0.64	3.441	Side-to-side
9	11 099	11 073	25.7	447	0.46	3.529	Fore-aft

The jacket foundation is more complex than that of the monopile as the tilting movement of the structure will lead to vertical excitation of the piles. From the results it is clear that the damping of the side-to-side modes is significantly lower than the damping in the fore-aft modes.

6.2.2 Relationship between foundation and total damping - Method 2

The relationship between the jacket foundation's damping, consisting of the four piles, and the damping of the entire structure is calculated using method 2. Setting a unit foundation damping $D_f = 1$. The results for the jacket structure are displayed in Table 6.4.

First, the unit damping of 1 was applied to both the foundation excluded structure and the foundation to verify the correct implementation. This resulted in total damping of 1 per mode, equal to the overall damping applied. Indicating a correct implementation.

Table 6.4: The foundation damping as share of total damping per mode for the jacket structure. Calculated using method 2

Mode	**Frequency [Hz]**	**Foundation damping as share of total damping [%]**
1	0.272	4.6
2	0.274	4.7
3	1.176	13.3
4	1.185	1.8
5	1.299	18.4
6	2.005	22.2
7	2.220	19.2
8	3.441	22.1
9	3.529	20.7
10	3.581	9.7
11	3.736	15.7
12	4.244	4.4
13	4.306	3.9
14	4.347	51.0
15	4.702	0.6

6.2.3 Radiation damping

The effect of radiation damping on the entire jacket structure can be found in Table 6.5. The results are calculated by first calculating the radiation damping per pile, using the PILES program. By assuming that the piles do not interact, the radiation damping of each pile is equal to the radiation damping of the foundation, as described in Equation 6.1. The total damping can then be obtained by multiplying damping per pile by foundation damping as a share of total damping. Foundation damping as share of total damping is calculated in Table 6.4 using method 2. The weighted average is based on the nine first modes' dynamic energy Π_{Di} and radiation damping D_{ri}. The weighted average is calculated as $\frac{\Sigma\Pi_{Di}D_{ri}}{\Sigma\Pi_{Di}}$. Giving a total weighted average of:

- 0.87% including all modes
- 1.15% including only fore-aft modes
- 0.29% including only side-to-side modes

$$D_f = \frac{W_{Lf}}{4\pi\Pi_f} = \frac{4\pi D_{pile}\Pi_{pile} \times 4}{4\pi\Pi_{pile} \times 4} = D_{pile} \tag{6.1}$$

Table 6.5: The effect of radiation damping for the entire jacket structure.

Mode	**Frequency [Hz]**	**Foundation damping as share of total damping [%]**	**Radiation damping per pile [%]**	**Total damping [%]**
1	0.272	4.6	0.50	0.02
2	0.274	4.7	0.69	0.03
3	1.176	13.3	2.57	0.34
4	1.185	1.8	2.64	0.05
5	1.299	18.4	3.24	0.60
6	2.005	22.2	7.02	1.56
7	2.220	19.2	7.98	1.53
8	3.441	22.1	13.00	2.87
9	3.529	20.7	13.27	2.74
10	3.581	9.7	13.56	1.31
11	3.736	15.7	13.92	2.18
12	4.244	4.4	15.40	0.68
13	4.306	3.9	15.50	0.60
14	4.347	51.0	15.50	7.90
15	4.702	0.6	16.40	0.09

6.2.4 Combined hysteric and radiation damping

The hysteric damping and the radiation damping are combined in Table 6.6. The weighted average is based on the nine first modes' dynamic energy Π_{Di} and total damping D_{Ti}. The weighted average is calculated as $\frac{\Sigma\Pi_{Di}D_{Ti}}{\Sigma\Pi_{Di}}$. Giving a total weighted average of:

- 3.32% including all modes
- 4.44% including only fore-aft modes
- 0.58% including only side-to-side modes

Table 6.6: Combined radiation and hysteric damping for entire jacket-structure

Mode	Frequency [Hz]	Hysteric damping [%]	Radiation damping [%]	Combined damping [%]	Mode type
1	0.272	0.13	0.02	0.15	Side-to-side
2	0.274	3.50	0.03	3.53	Fore-aft
3	1.176	0.30	0.34	0.65	Side-to-side
4	1.185	0.03	0.05	0.07	Yaw
5	1.299	6.35	0.60	6.95	Fore-aft
6	2.005	1.52	1.56	3.07	Side-to-side
7	2.220	2.60	1.53	4.13	Fore-aft
8	3.441	0.64	2.87	3.51	Side-to-side
9	3.529	0.46	2.74	3.20	Fore-aft

6.3 Damping results summarised

The results of the damping estimation is summarised in Table 6.7. The weighted average is calculated by $\frac{\Sigma\Pi_{Di}D_i}{\Sigma\Pi_{Di}}$, where Π_{Di} is the dynamic energy in each mode and D_i is the damping in each mode.

Table 6.7: Summation of damping results

Side-to-side modes	Fore-aft modes	Yaw modes

	Monopile		Jacket			
Mode	Frequency [Hz]	Hysteric damping [%]	Frequency [Hz]	Hysteric damping [%]	Radiation damping [%]	Total damping [%]
1	0.252	0.10	0.272	0.13	0.02	0.15
2	0.253	2.95	0.274	3.50	0.03	3.53
3	1.063	0.27	1.176	0.30	0.34	0.65
4	1.164	7.90	1.185	0.03	0.05	0.07
5	1.249	0.04	1.299	6.35	0.60	6.95
6	1.980	0.11	2.005	1.52	1.56	3.07
7	2.245	3.09	2.22	2.60	1.53	4.13
8	-	-	3.441	0.64	2.87	3.51
9	-	-	3.529	0.46	2.74	3.20
	Monopile weighted avg damping		Jacket weighted avg damping			
Mode type	Hysteric [%]		Hysteric [%]	Radiation [%]	Total [%]	
Side-to-side	0.21		0.29	0.29	0.58	
Fore-aft	3.50		3.29	1.15	4.44	
Yaw	0.04		0.03	0.05	0.08	
All modes	2.39		2.44	0.87	3.32	

6.4 Comparison between damping results and literature

The results from the damping estimation are significantly higher in the first fore-aft mode than previous estimations [13–17]. The authors' estimation for the monopile's soil damping in the first fore-aft mode is almost twice the highest reported damping in literature (1.5 % from Versteijlen et al. [16]). Since the damping in literature typically is estimated for low order modes, it is difficult to compare the estimated damping of the high order modes. Two contributing factors might be the cause of the observed difference in the first fore-aft mode:

1. The damping estimated in literature is typically for smaller and older turbines with more conservatively designed foundations. This would lead to less movement in the foundation and thus less damping from the

soil. The monopile in the present study is for a modern design with large diameters and a low aspect ratio which will lead to larger motions in the foundations.

2. In literature, the damping is typically estimated from free-decay tests. A wind turbine in free decay is a different dynamic system than that of the structure during normal operations, on which the author's damping estimation is based.

6.5 Fatigue results

A MLife fatigue analysis was performed on five cases. Three analyses for the jacket and two for the monopile. The first monopile case is one with 1.0% damping, which is considered the baseline. Another case is when the effects of hysteric damping are added to the baseline damping. These same damping cases are performed on the jacket structure, in addition to one case with radiation damping added to the previous two damping contributions.

6.5.1 Monopile fatigue results

The expected durability for the monopile case is presented in Table 6.8.

Table 6.8: Durability of monopile for selected sections.
* % increase over baseline case due to additional damping

Section	**Baseline**	**Hysteric**	**Hysteric increase***
$[-]$	$[years]$	$[years]$	[%]
Transition piece	135	156	15.6
Mudline	65.6	116	76.8

6.5.2 Jacket fatigue results

During the fatigue design, it was discovered that the durability of the jacket was dangerously low. This prompted an investigation into finding out why a single brace had such a low lifespan. Several factors were modified to troubleshoot the issue. However, any variable that increased the lifetime of the critical member would also extend the durability of other members to much higher proportions. Therefore, an additional fatigue analysis was performed on randomly selected jacket members in an attempt to detect other members exposed to fatigue. About half of these members had similarly low durability. However, no clear pattern emerged as to which parts of the jacket were vulnerable to fatigue. The results from the additional fatigue analysis on the jacket substructure will not be reported further in this study due to the computing costs associated with performing these simulations.

After analysing the implemented model, the focus shifted to the source of the reference jacket. The fatigue analyses in Deliverable 4.31 in the European research project INNWIND [64] showed poor performance for certain joints in the reference jacket. D4.31 describes how the reference jacket had a crucial design flaw where the 3P blade passing frequency ($0.30Hz = 6rpm$) was too close to the first natural frequency (0.287 Hz). This would induce resonance for lower wind velocities (5-7 m/s). As a result, select K-joints would fail after about four years, far short of the target of 25 years. The best explanation for the frighteningly short time to failure experienced during our analyses would be that the critical member was excited by the blade passing frequency and received excessive damage as a result. Deliverable 4.34 [81] in the project addresses the fatigue issues by making extensive changes to the reference jacket. To avoid making these changes in this study, it was decided to subdivide the fatigue results of the jacket into two parts. The first part considers all the members (resonant) with the results given in Table 6.10. The second part disregards the faulty member such that it only encompasses the members deemed to be acceptable (ideal). In the rest of this study, the ideal case will be considered the best representation of the jacket's durability. Therefore, unless otherwise stated, the ideal case will be used as the jacket's point of comparison.

Table 6.9: Durability of ideal jacket for selected sections.
* % increase over baseline case due to additional damping

Section	**Baseline**	**Hysteric**	**Hysteric increase***	**Radiation**	**Radiation increase***
[−]	[*years*]	[*years*]	[%]	[*years*]	[%]
Transition piece	219	241	10.0	241	10.0
Mudline	104	409	293	418	302
Braces	20.7	683	3300	844	3980
Leg	18.5	69.2	274	71.2	285

6.5.3 Fatigue results overview and discussion

Table 6.10: Durability of critical section of monopile and jacket.
* Increase over baseline case due to additional damping

Turbine	**Baseline**	**Hysteric**	**Hysteric increase***		**Radiation**	**Radiation increase***	
[−]	[*years*]	[*years*]	[*years*]	[%]	[*years*]	[*years*]	[%]
Monopile	65.6	116	50.4	76.8	-	-	-
Jacket ideal	18.5	69.2	50.7	274	71.2	52.7	285
Jacket resonant	2.26	6.45	4.19	185	9.57	7.31	323

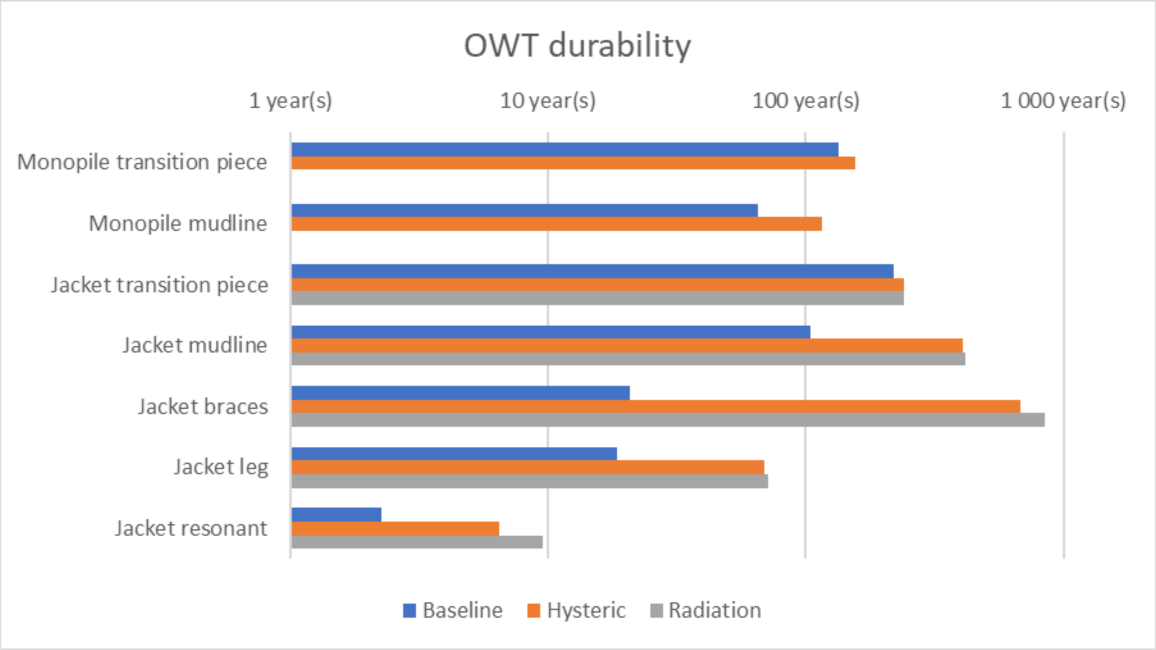


Figure 6.1: Fatigue lifetime of various turbine parts

We observe reasonable durability of the jacket at 18.5 years of service life. It is only slightly lower than the most common design goal of 20 years. The monopile, however, is expected to last an impressive 66 years before a fatigue failure occurs. There are four possible reasons why this may be the case:

1. Using the cross-sectional strength does not give an accurate representation of the MLife ultimate load ($LUlt$).

2. The fatigue limit state may not be the governing design situation as the monopile may be more exposed to other actions.

3. Fatigue performance was not mentioned in the Task 37 report [65]. Therefore, it may be the case that the turbine was not designed to have economic fatigue durability.

4. It could be a deliberate design decision to have a durability several times higher than design life. DNV RP-C203 uses design fatigue factor (DFF) to increase the safety of the section [80]. According to INNWIND's Deliverable 4.34, a DFF of three would give such a small probability of fatigue fault occurring that fatigue service inspection does not need to be carried out [81]. Sixty-six years is coincidentally about three times higher than a service life of 20 years.

From Table 6.10 and Figure 6.1 it is evident that the additional soil damping had a significant impact on the durability of the OWTs. The monopile turbine model saw at least 50 extra years until failure resulting in a 77% increase. The jacket also received about 50 more years, giving a remarkable 274% increase. The resonance-exposed jacket brace saw a lower percentage increase compared to the jacket leg. However, the damping had a much larger impact on the other braces as they will last at least 30 times longer. On the flip side, the hysteric damping only increases the time until failure at the TP by a couple of years and percent.

The fatigue life of the jacket is further increased by including radiation damping in addition to the hysteric damping. Radiation damping gave the most significant fatigue life increase to the braces. The braces possess the longest fatigue life of 844 years. Thus improving the fatigue life almost 40 times what the baseline damping provided. The TP did not experience any more durability due to the inclusion of radiation damping in addition to hysteric damping. Both the piles at mudline and jacket legs have a 280-300% improvement over the baseline damping. The jacket may attain about two additional years to its service life by including radiation damping, over only considering hysteric damping.

6.5.4 Environmental load impact on durability

The short term DEL is computed for each mean wind velocity to compare the damage done for various environmental states. To compare the DELs across different cross-sections and substructure models, a load-weighted DEL ratio was established. This is the ratio between the short term DEL and the ultimate load ($LUlt$). Figure 6.2 shows the load-weighted DELs for all the analysed member groups without soil damping. There is a clear difference between the jacket and the monopile as to which wind velocities are the most damaging. When the rotor is parked, the monopile sustains much more damage than the jacket. This is likely caused by the loss of aerodynamic damping contribution from the spinning blades. When the blades are not spinning, the hydrodynamic forces become more important. Therefore, it is not surprising that the DEL is higher for the monopile as it is much more exposed to hydrodynamic forces than the jacket. A trend for both the monopile and jacket is a significant DEL between 8 and 12 m/s. Additionally, there is not any significant difference between the DELs for 14-24 m/s and 26-38 m/s.

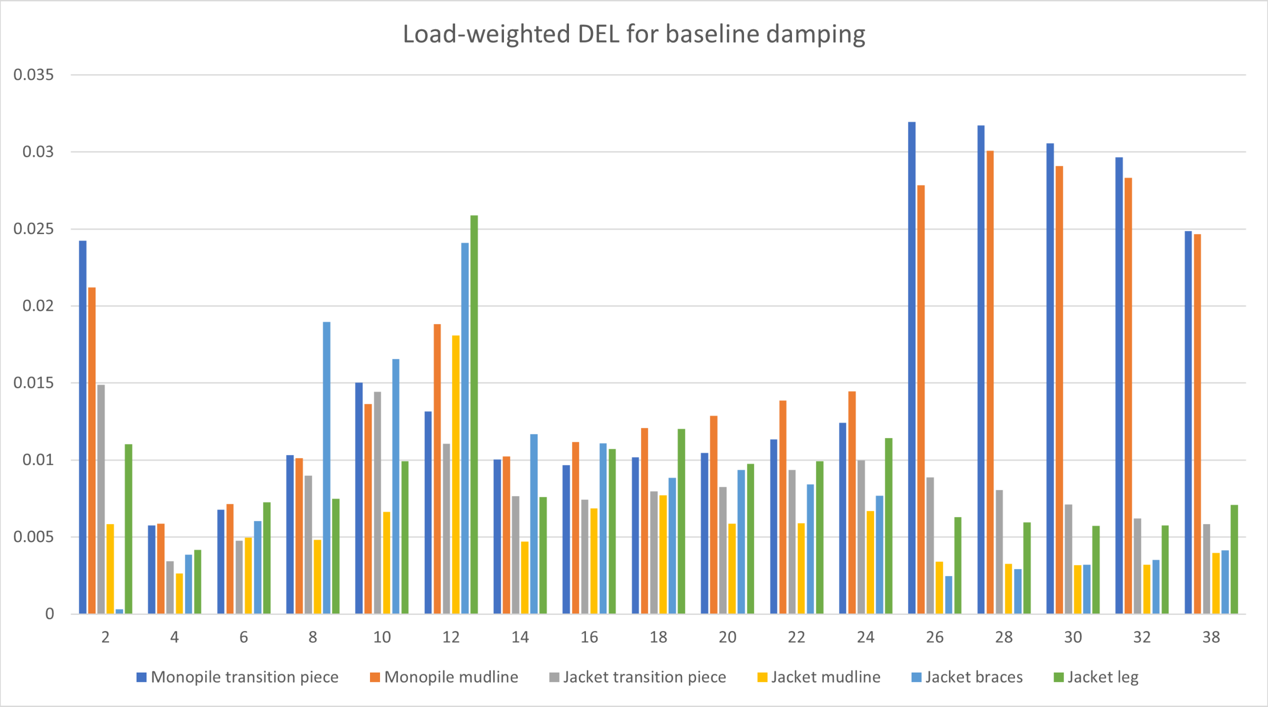


Figure 6.2: Load-weighted DEL for baseline damping at various reference mean wind velocities

When the load-weighted DEL is multiplied by the probability that the wind velocity is within the respective wind-bin, the load- and probability-weighted DEL is established. This DEL gives a better representation of the damages dealt by each mean wind velocity throughout the OWTs service life. Figure 6.3 shows this new relation. It is evident that the wind velocities around the turbine rated wind speed deal the greatest damage. Winds below the cut-in velocity also deal significant damage. The probability of winds faster than 20 m/s is low enough to make them almost negligible to the overall durability of the OWTs. This is even the case for idling monopile rotors.

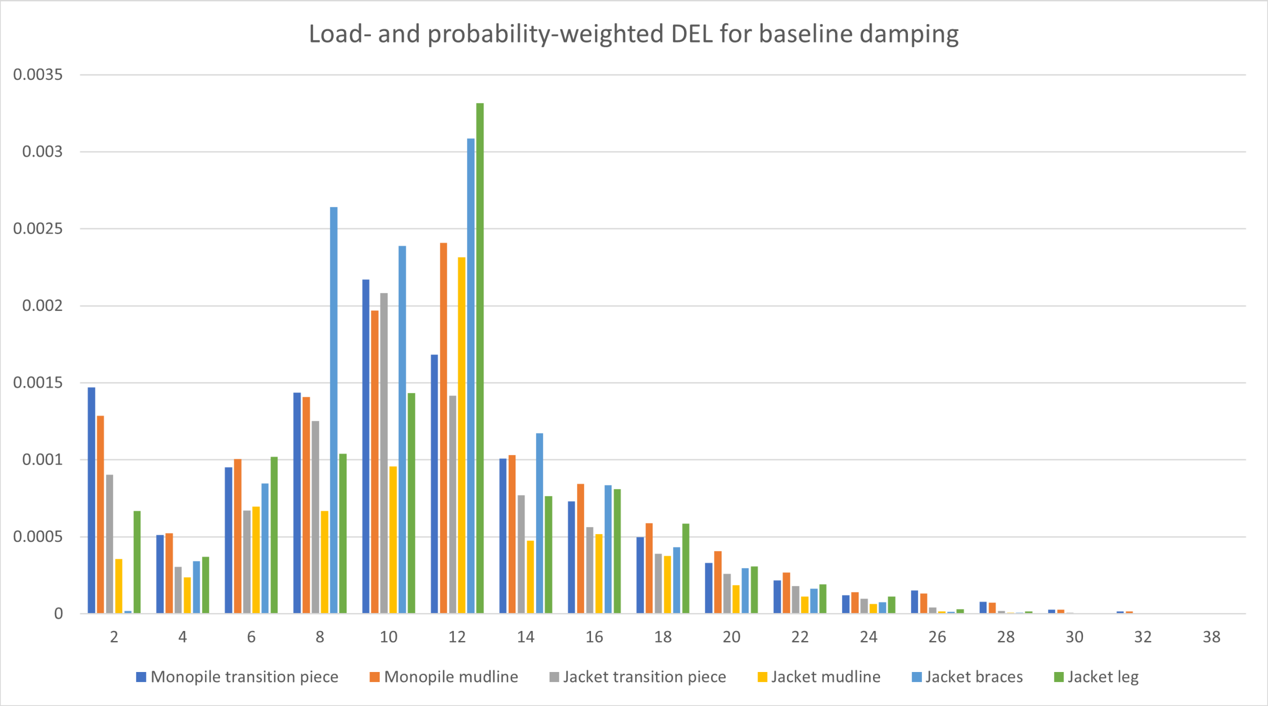


Figure 6.3: Load- and probability-weighted DEL for baseline damping at various reference mean wind velocities

6.5.5 Damping impact on durability

MLife has also produced the DELs for the damped simulations. By plotting these beside the DELs for the OWTs without soil damping, it will be more obvious how each section achieved its fatigue life. Figure 6.4.a shows the DELs at the monopile TP, but the same damping trend is also present for the jacket TP. There is no obvious effect of soil damping while the turbine is producing power. However, once the turbine is parked, large improvements can be seen. This is very inefficient damping measure as the turbine should be spinning about 92% of the time. The monopile mudline (Figure 6.4.b) shows a very similar damping situation to the TP, except it shows moderate effect of soil damping around the rated wind velocity. As a result, the mudline experienced a meaningful improvement by adding the soil damping. Figure 6.4.c shows the jacket braces, which are in stark contrast to the TP as soil damping is prominent while the turbine is producing power. The spinning turbine evidently excites the modes being damped, giving an extremely favourable fatigue life extension. The jacket mudline and leg have an almost indistinguishable DEL shape and are represented by Figure 6.4.d. The soil damping is present throughout most of the wind bins, resulting in an impressive durability increase. It is observed that the additional radiation damping does not significantly lower the DELs compared to only hysteric damping.

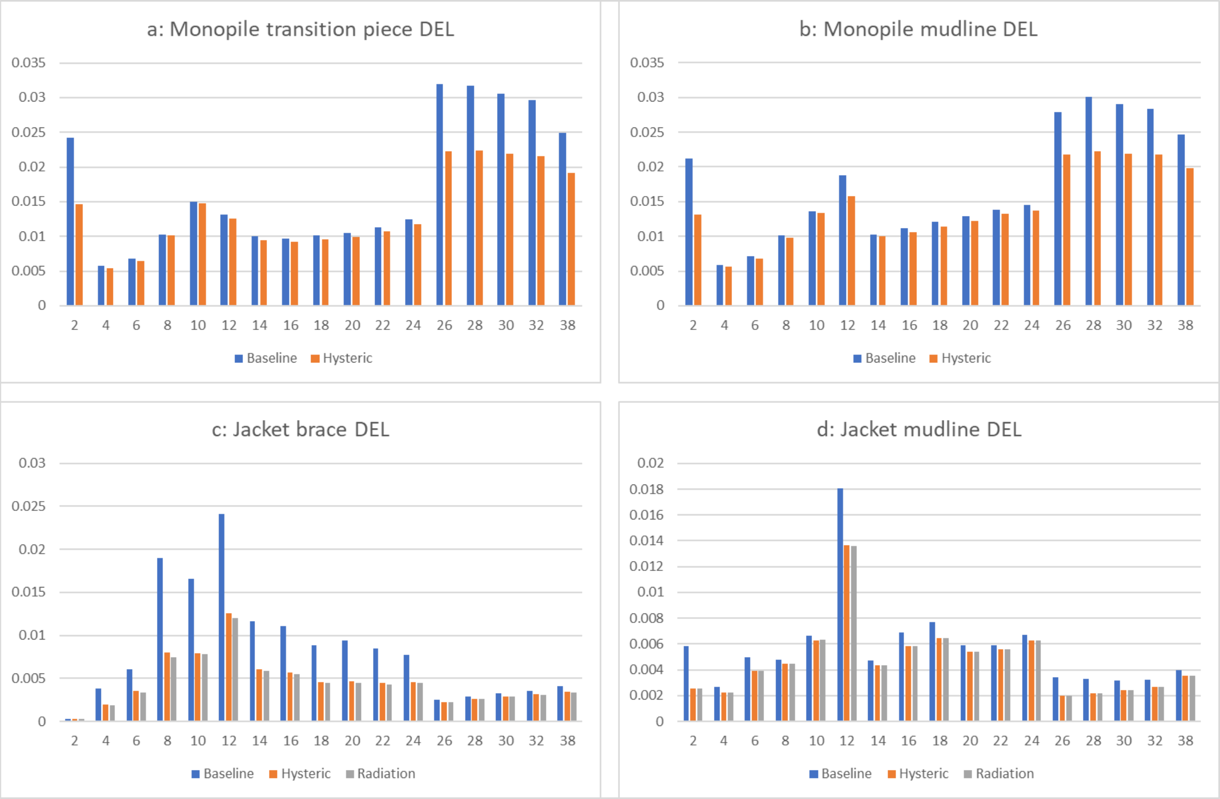


Figure 6.4: DELs for various damping configurations

6.6 OpenFAST results

6.6.1 Damping impact

The effect of applying soil damping to the structures is, in some cases, almost undetectable in the time series of the OWTs. In other cases the impact of hysteric damping far outweighs any additional radiation damping. For these types of situations, the damping cases might be hard to distinguish from each other. Despite this, all damping configurations will be displayed on the plot to represent the differences visually.

Figure 6.5 shows the fore-aft displacement at the tower top during a power-producing state during a 12 m/s mean wind velocity. Here all three damping values are indistinguishable from each other. This means that the inclusion of soil damping will not improve the turbine with respect to the SLS conditions. However, the fore-aft damping is noticeable when the blades are parked. This can be seen in Figure 6.6. This relation is valid for both the monopile and the jacket.

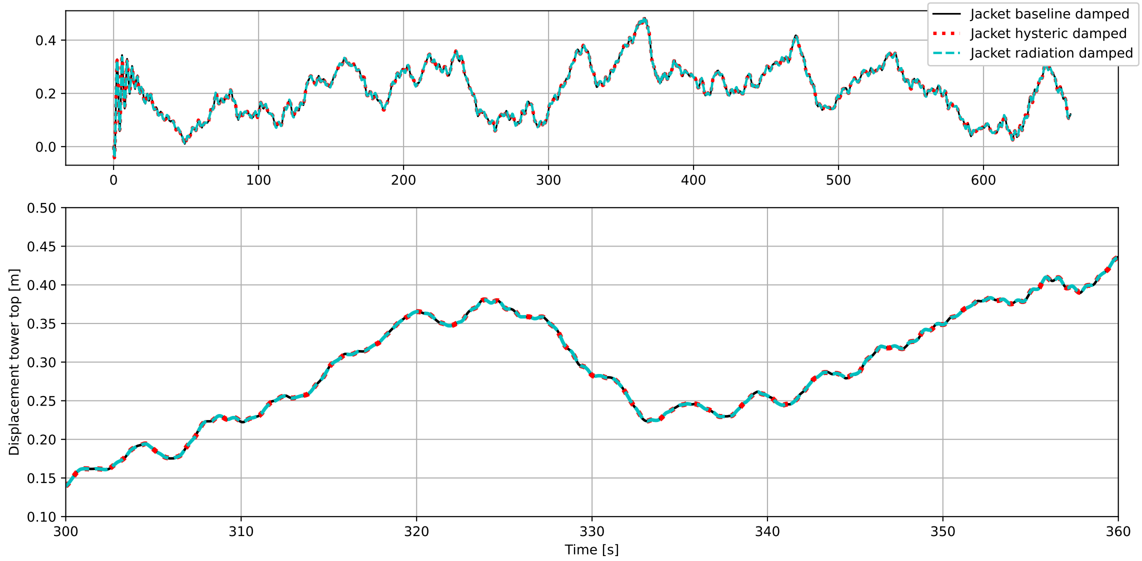


Figure 6.5: Fore-aft displacement for jacket structure at tower top during 12 m/s mean wind

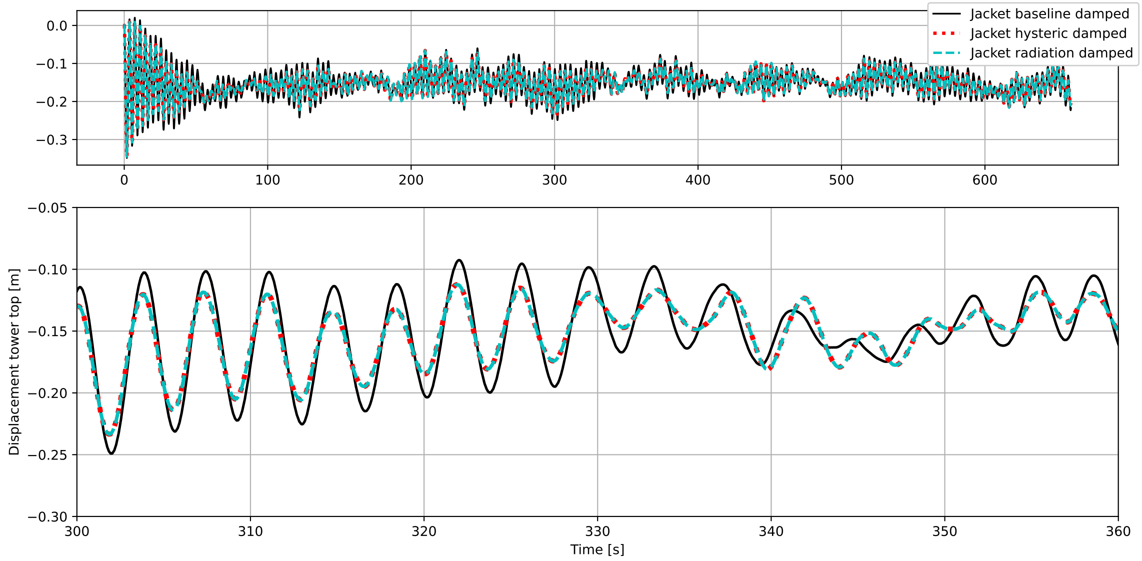


Figure 6.6: Fore-aft displacement for jacket structure at tower top during 38 m/s mean wind

Shifting attention to the jacket's mudline moments, a more substantial effect of the damping can be observed. Figure 6.7 plots the moment for the jacket with the three different damping configurations. The damping clearly reduces the larger moments. In fact, the moment range is almost 30% smaller when radiation and hysteric damping is included. However, the impact of damping is less prominent for smaller moment amplitudes.

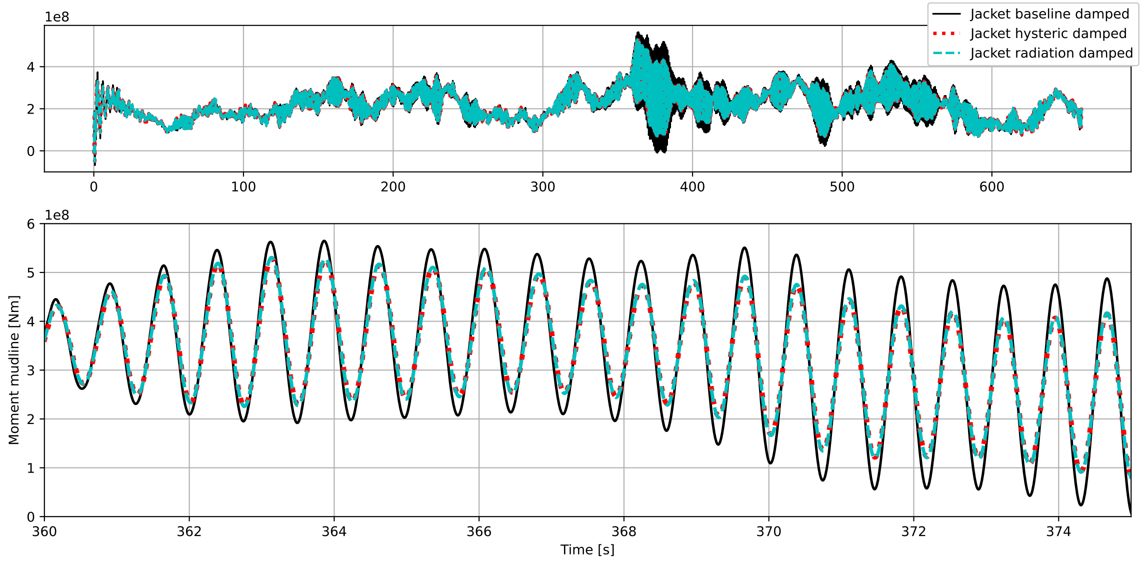


Figure 6.7: Mudline moments for jacket structure during 12 m/s mean wind

The most significant improvement hysteric damping had on the monopile is when the rotor is idling. Moments at the TP and mudline decreased about 20% when the turbine blades are idling, but closer to 10% when producing power. A plot of the monopile mudline moment is displayed in Figure 6.8.

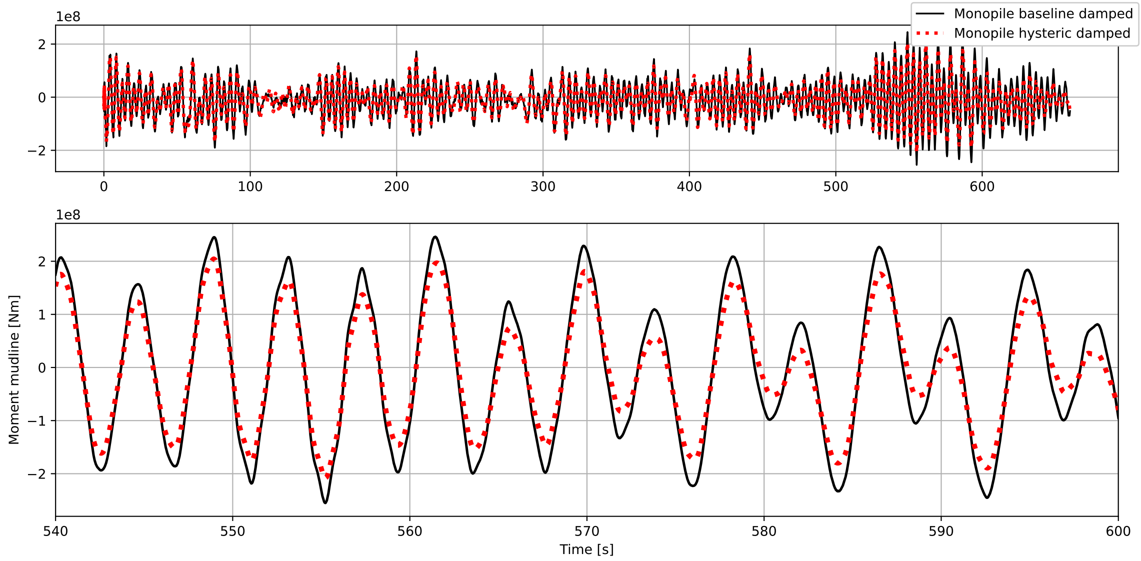


Figure 6.8: Mudline moments for monopile structure during 38 m/s mean wind

A jacket brace showed almost 50% reduction in moment range when damping was applied. This is the point which is the most influenced by the applied damping. The time series for this point is shown in Figure 6.9.

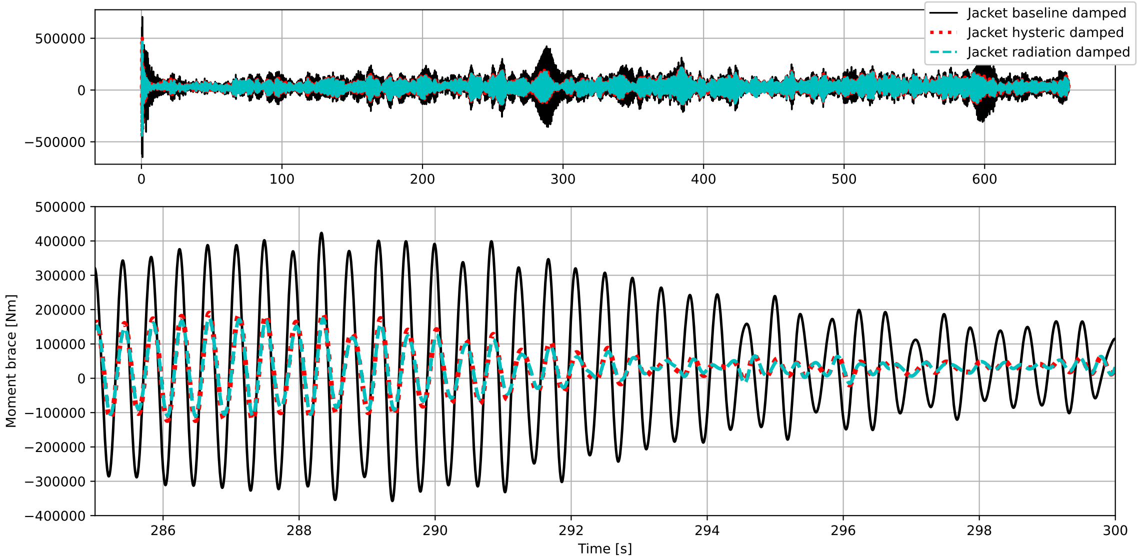


Figure 6.9: Brace moments for jacket structure during 12 m/s mean wind

6.6.2 Substructure comparison

Figure 6.10 shows the displacement at the top of the tower. It shows that the monopile experiences larger deflections and has a pronounced vibration of its first natural frequency.

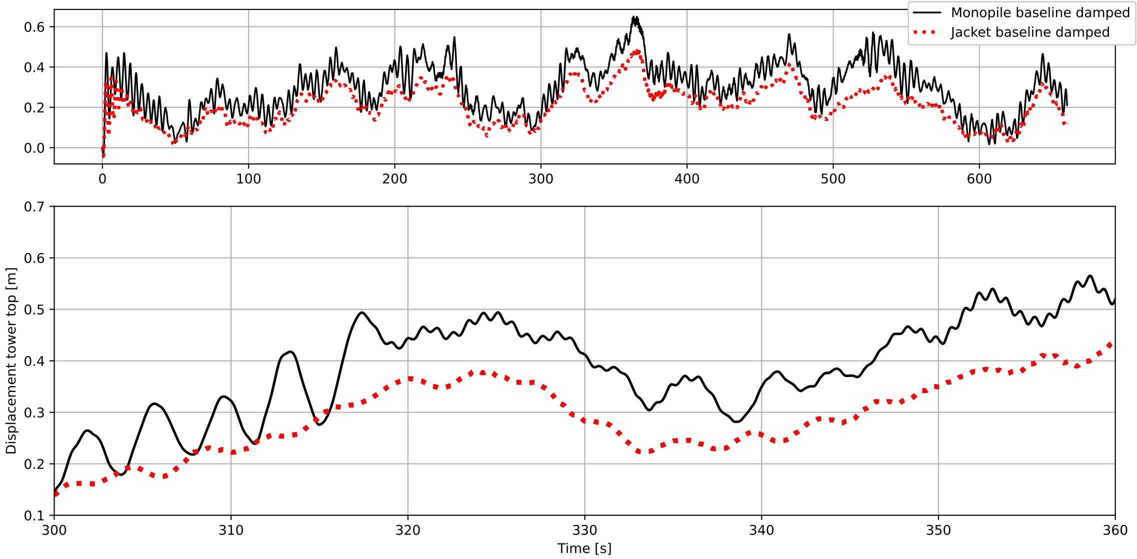


Figure 6.10: Displacement at tower top for baseline damping and 12 m/s mean wind velocity

Despite the higher displacement of the monopile, the jacket structure shows overall larger overturning moments at the mudline. This is plotted in Figure 6.11. The mudline moments for the monopile and jacket act differently from one another. Due to the moment arm between the jacket legs, moments are primarily translated to pure tension and compression at the piles.

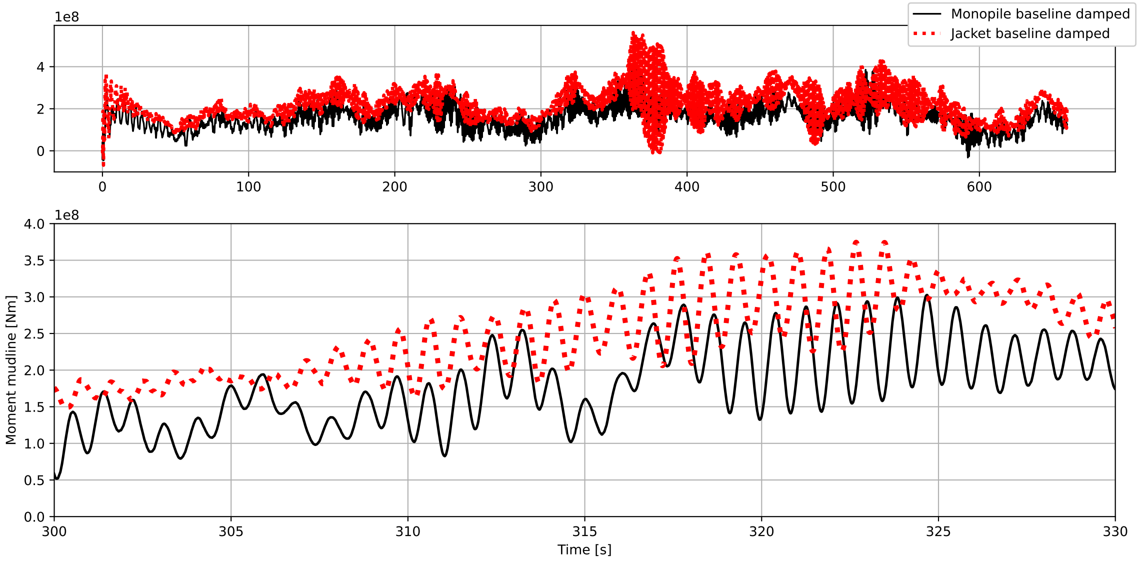


Figure 6.11: Moment at mudline for baseline damping and 12 m/s mean wind velocity

6.6.2.1 Impact of wind and wave loads

We see some interesting relations by shifting focus to the environmental loads acting on the turbines. Figure 6.12 shows the forces from the rotors acting on the hub for both an electricity producing wind velocity (12 m/s) and a parked case (38 m/s). The forces on the jacket and monopile are similar in size. However, it is clear that the rotating blades have a substantial impact on the aerodynamic forces. When the blades are parked, the wind does not transfer nearly as much energy to the hub as when they rotate at maximum velocity.

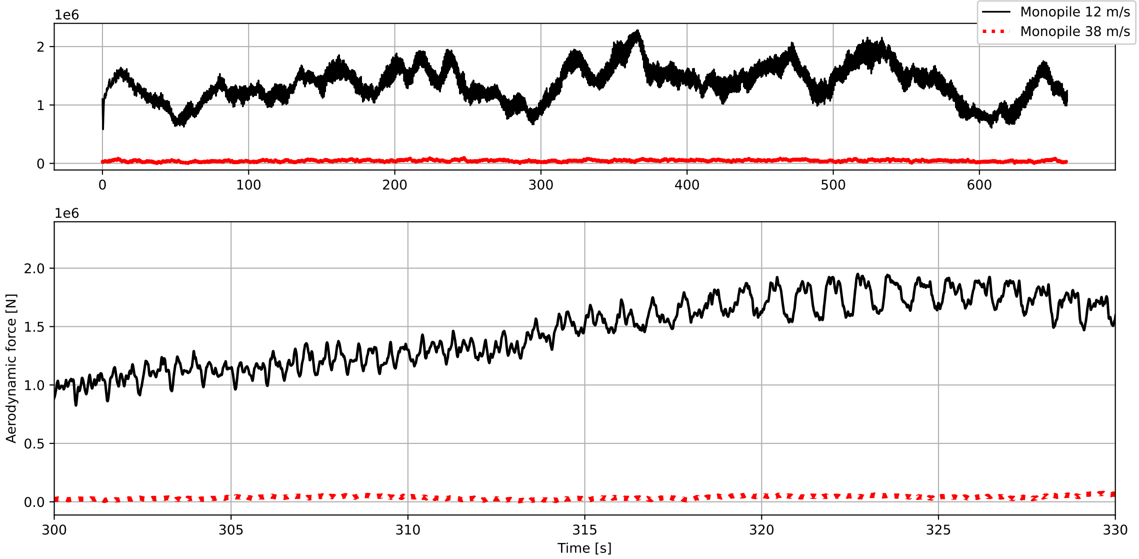


Figure 6.12: Rotor aerodynamic load for select mean wind velocities

Figure 6.13 shows the total integrated hydrodynamic loads for the substructures. These are the total forces acting on the substructure due to waves and current. The time slice is representative of the forces during the entire eleven-minute time series. The plot shows that the monopile is much more exposed to the hydrodynamic forces than the jacket. It also reveals that the hydrodynamic forces are applied at a much lower frequency for higher wind velocities.

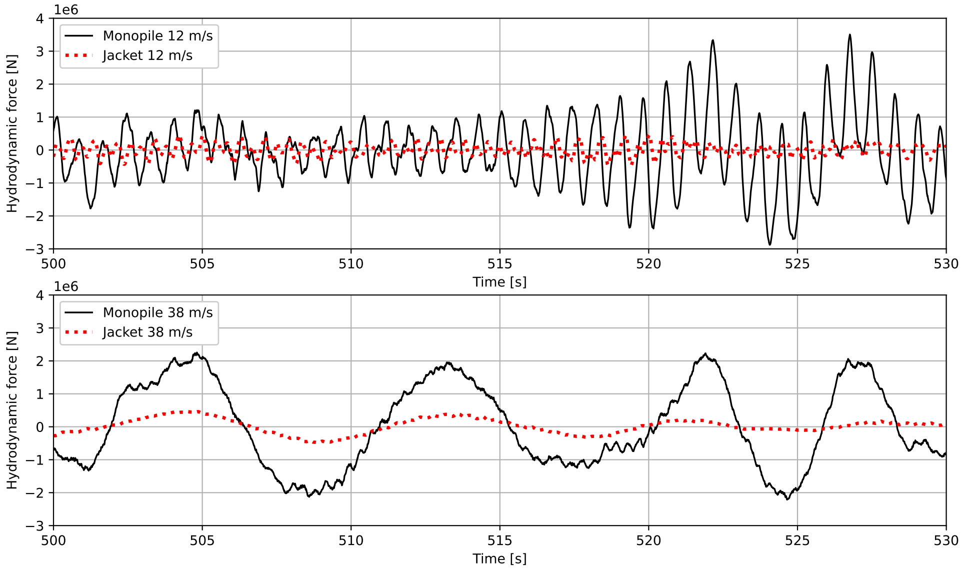


Figure 6.13: Integrated hydrodynamic load for select mean wind velocities

Due to the negligible aerodynamic forces for parked turbines, the hydrodynamic forces are expected to be closely related to the base reaction. Figure 6.14 shows the reaction forces at the mudline for an idling turbine. For this case, the monopile structure has a greater response than the jacket structure. Therefore, it does show a strong correlation to the integrated hydrodynamic loads.

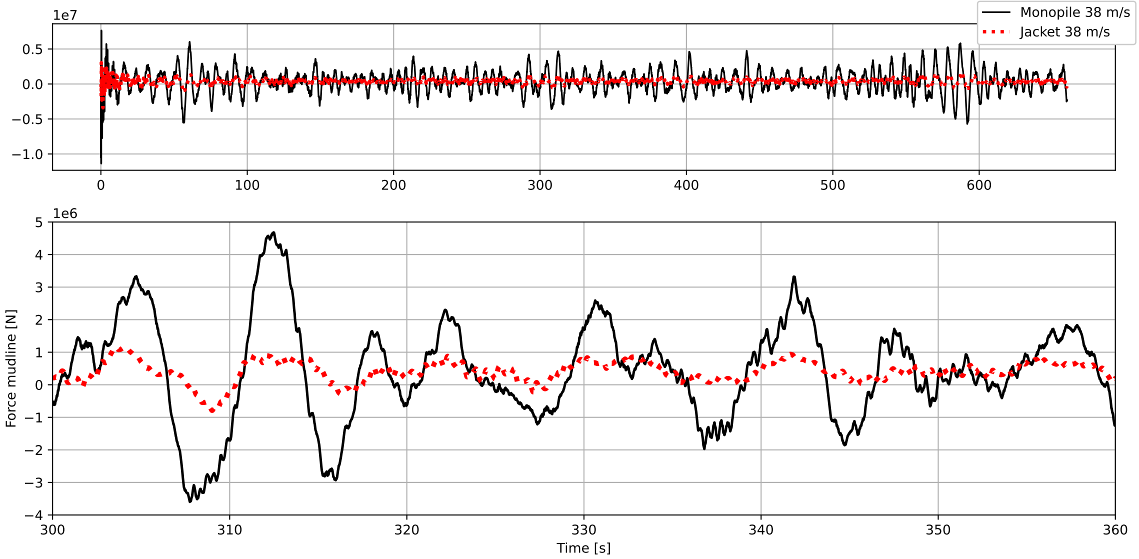


Figure 6.14: Force at mudline for baseline damping and 38 m/s mean wind velocity

6.7 Effect of damping in ULS

A parked monopile turbine subject to a 50 m/s mean wind velocity for ten minutes is the basis for the ULS case. Two monopile structures were created for comparison. One with only baseline damping and one with added hysteric damping. That is the hysteric damping for a 12 m/s wind case which was calculated in Section 6.1. A more accurate damping estimation may be achieved by basing the damping calculations on the 50 m/s wind case. The 50 m/s case has a lower static displacement and higher standard deviation than the 12 m/s case. Issues establishing a reliable model for higher wind velocities are discussed in Section 6.8.

The maximum forces and moments for the monopile were checked with and without soil damping. Some of these results are shown in Table 6.11. Overall, the monopile experienced a 12-18% decrease in the moments at the mudline and at the TP. The force at the TP was reduced by 13%, but the maximum force at the mudline was only reduced by 0.9%. When checking the range and standard deviation change of the force and moment, it was observed that they are reduced by a similar percentage amount to the values seen for the absolute maximum force and moment.

Table 6.11: Maximum absolute forces and moments for monopile during 50 m/s wind

Case	**Transition piece**		**Mudline**	
	Force $[kN]$	**Moment** $[kNm]$	**Force** $[kN]$	**Moment** $[kNm]$
Baseline damping	2639	171 600	10 270	361 400
Hysteric damping	2304	140 800	10 180	317 300
Difference	335	30 800	90	44 100

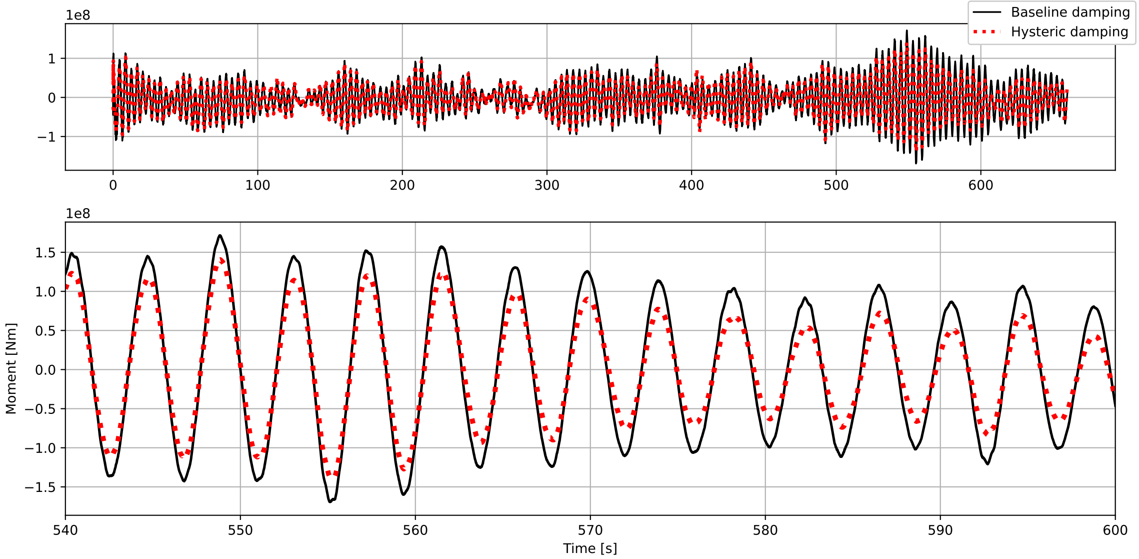


Figure 6.15: Time-series of fore-aft moment at the transition piece

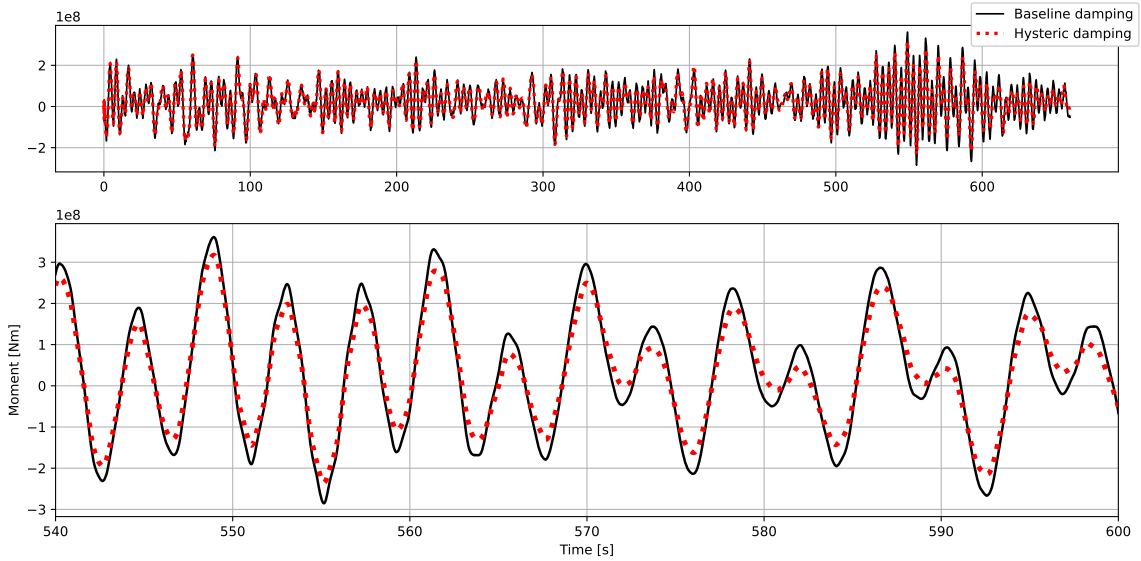


Figure 6.16: Time-series of fore-aft moment at mudline with focus on the largest moments

The displacement and rotation about the top of the tower underwent a 12-16% reduction when damping was included. Based on expected turbine behaviour, it is predicted that the damping would have a similar effect during SLS wind conditions when the turbine is parked.

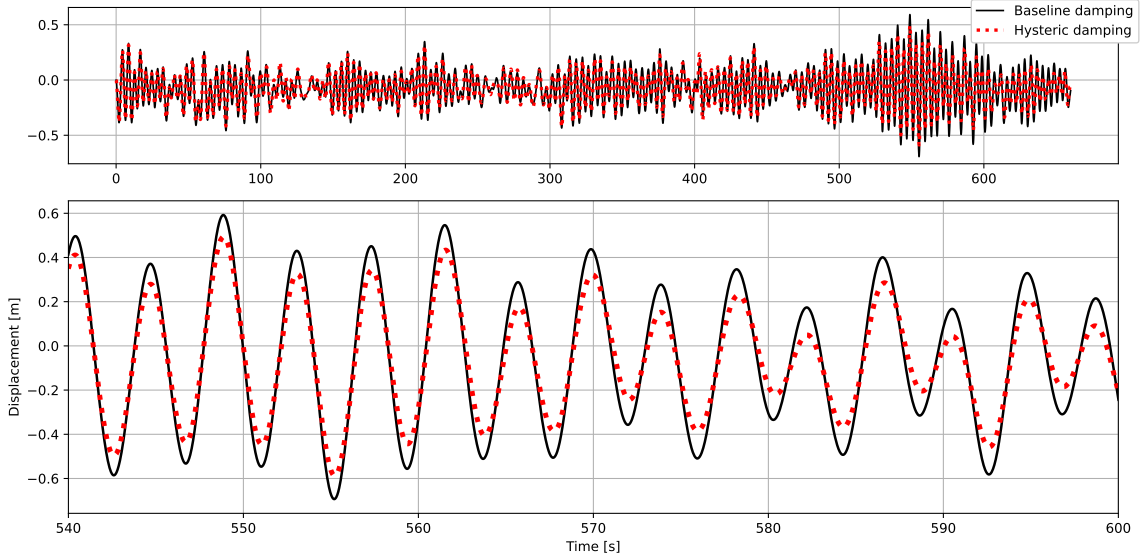


Figure 6.17: Time-series of fore-aft displacement at tower top

It is evident that the extra damping affected the maximum forces and moments of the monopile structure. The meagre reduction in forces does not significantly expand the number of viable designs. Especially since the moments were already about ten times lower than the yield stresses. Therefore, the damping may increase the safety margin of the section. Results from the fatigue analysis showed that certain jacket members see a much more significant reduction in forces due to damping. Therefore, it may be economical to include soil damping for a ULS analysis of a jacket.

6.8 Effect of waves, hydrodynamic and aerodynamic damping

The effect of waves, hydrodynamic and aerodynamic damping has not been included in the modal analysis performed on the Abaqus model. As shown in Figure 5.3 5.4, 5.5 and 5.6 the effect on the dynamic movement is limited in the top of the tower. In the foundation, the dynamic impact of waves and fluids are significant, as shown in Figure 6.18 and 6.19.

By ignoring the effect of waves, the analysis will likely, all else equal, underestimate the foundation's dynamic movement and thus underestimate the damping.

By ignoring the effect of hydrodynamic and aerodynamic damping, all else equal, the analysis will overestimate the movement of the structure. The overestimated movement due to lack of damping is, in turn, likely to result in an overestimation of the soil damping.

Abaqus has a fluid module available that can include fluid effects. In turn, this model can be used to generate and export a load matrix, including the effect of waves and other fluid effects. This approach would give a more realistic representative displacement for each mode. However, the option has not been examined further in this thesis and remains for further work.

A better approach would be to calculate the modal properties directly from FAST. Insuring an accurate representation of loading and damping. This is discussed in Section 6.9

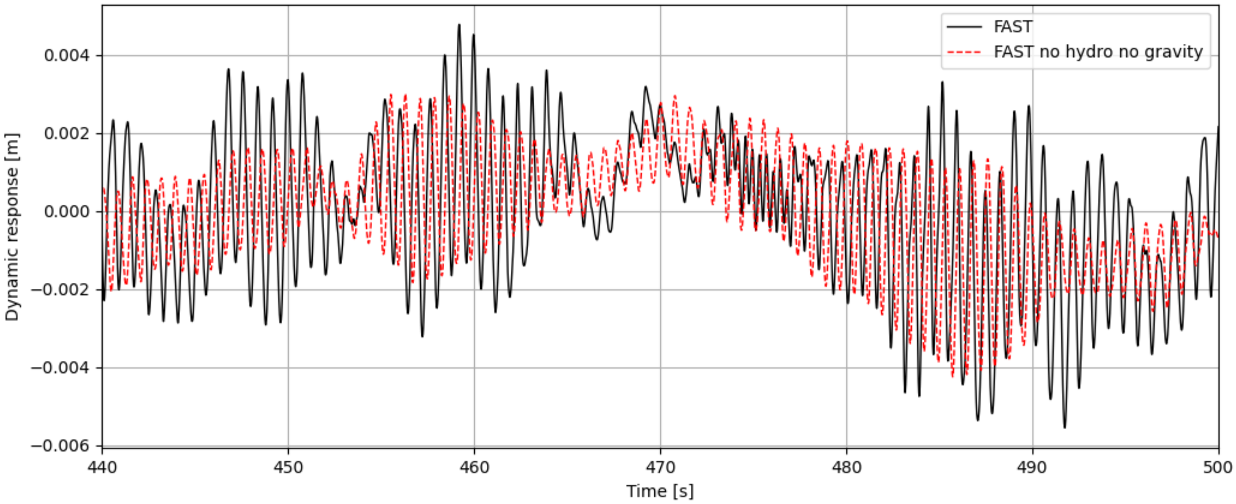


Figure 6.18: Response in foundation in the x-direction for the monopile structure. Comparison between FAST including hydro and gravity effects and FAST excluding hydro and gravity effects. 12 m/s wind

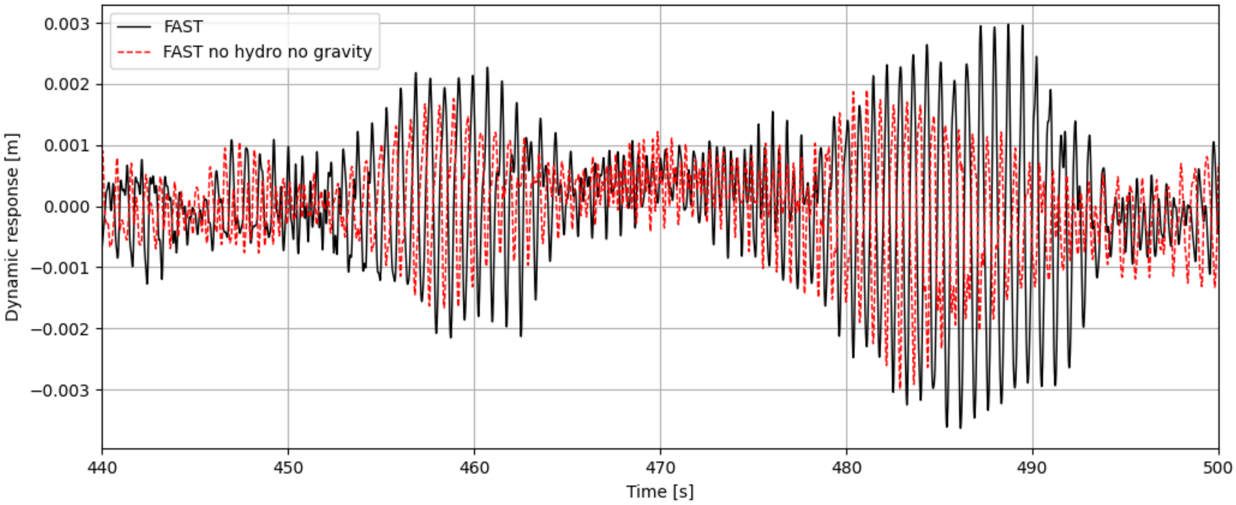


Figure 6.19: Response in one of the piles in the foundation in the x-direction for the jacket structure. Comparison between FAST including hydro and gravity effects and FAST excluding hydro and gravity effects. 12 m/s wind

6.8.1 Attempt estimating soil damping in ULS

An attempt was made to estimate hysteric soil damping for a ULS-load case. In the ULS-state, the wind turbine is restrained from spinning, resulting in a greatly reduced load from the turbine blades. As the wind load from the turbine is no longer dominating, wave loads must be included to get an accurate response. An attempt was made by distributing the total wave loads equally among the nodes below the sea surface. The total wave loads were distributed equally among nodes ranging from 10 to 30 m above the mudline for the monopile structure. The displacement in the fore-aft mode at top of the tower is shown in Figure 6.20. The figure shows a bad fit indicating that this approach is not suitable. Because of the bad fit, the authors chose not to estimate the damping as the results would have been untrustworthy.

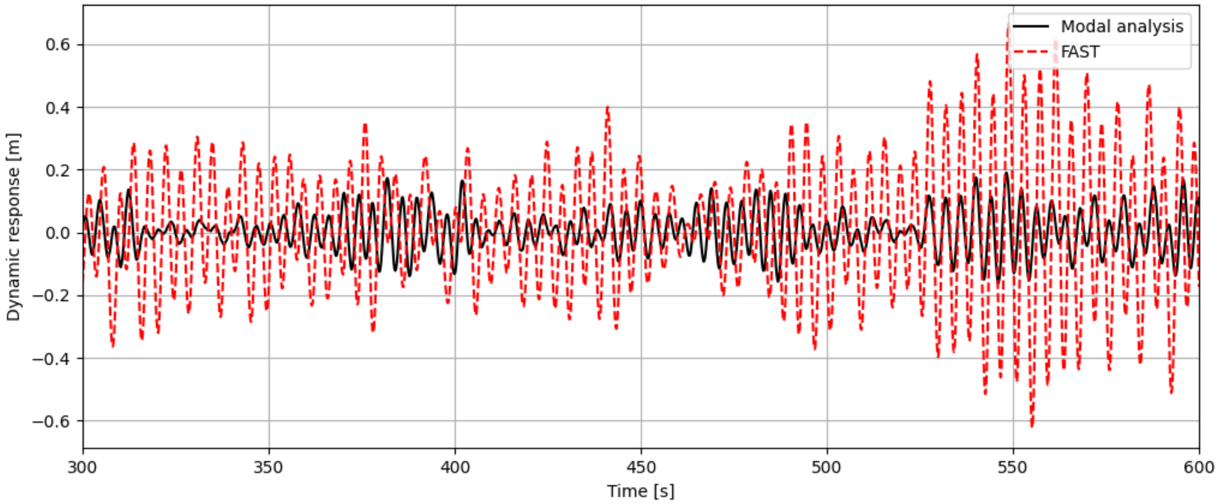


Figure 6.20: Displacement in the top of the monopile structure in the x-direction for ULS comparing FAST and modal results. The fit is non-existing. The results has not been used any further.

6.9 Different approach to obtain modal quantities

The movement and the energy of the modes were in this thesis obtained by creating a separate model. A more straightforward approach would be to extract these sizes directly from FAST. This approach has the advantage of giving a more accurate representation of the loading and damping, in addition to reducing the workload associated with the analysis. However, only the total response is easily available. Therefore a operational modal analysis (OMA) would have to be performed to split the system into different modes. In this case, splitting the movement into fore-aft and side-to-side directions would be recommended prior to extracting modal movement. The separation is recommended because pairs of modes in each direction have frequencies very close to each other. Therefore, the modes can be difficult to separate without an initial separation of movement in the fore-aft and side-to-side direction. After the movement in one node is found and separated into separate modes. The excitation of the modes can be found by comparing the relationship between each mode shape and the modal excitation in the nodes of interest. By also extracting the stiffness matrix, and the eigenvectors, the modal quantities can be calculated as described in method 1 in Section 2.5.1.

A third option is quantifying the modal properties and exporting them directly from the modal analysis in FAST. This approach would likely be the most accurate, result in the lowest computational time, and be the easiest to use of the methods proposed in this thesis. However, this approach will require intricate domain knowledge of the FAST framework and modifications of the FAST source code.

An inconvenience with these new approaches is that they require the full stiffness and mode shape matrices for the entire structure. The modular nature of OpenFAST limits the generation of these crucial components, but it might be achieved by modelling the entire OWT in Subdyn with full FEM DOFs. If this issue can be circumvented, then both of the suggested methods above are likely to result in a more accurate damping estimation than the method used in this thesis. In addition, the problem of converting the damping of the modes from the modal analysis to FAST modes would disappear. The authors, therefore, recommend using one of the suggested methods above instead of exporting the model to an external modal analysis.

6.10 Gravity

Gravity has only been partially included in this thesis. This is somewhat unfortunate as analysis from FAST is affected by whether the parameter is active or not, as seen in Figure 6.21 and 6.22.

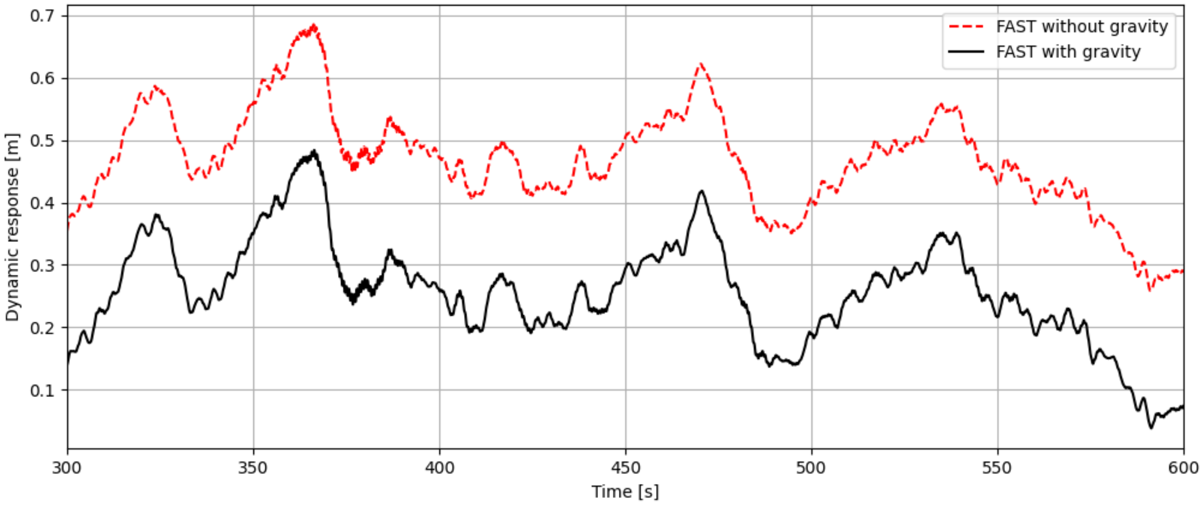


Figure 6.21: The jacket-structure's response in the top of the tower with and without gravity included in calculation. 12 m/s wind

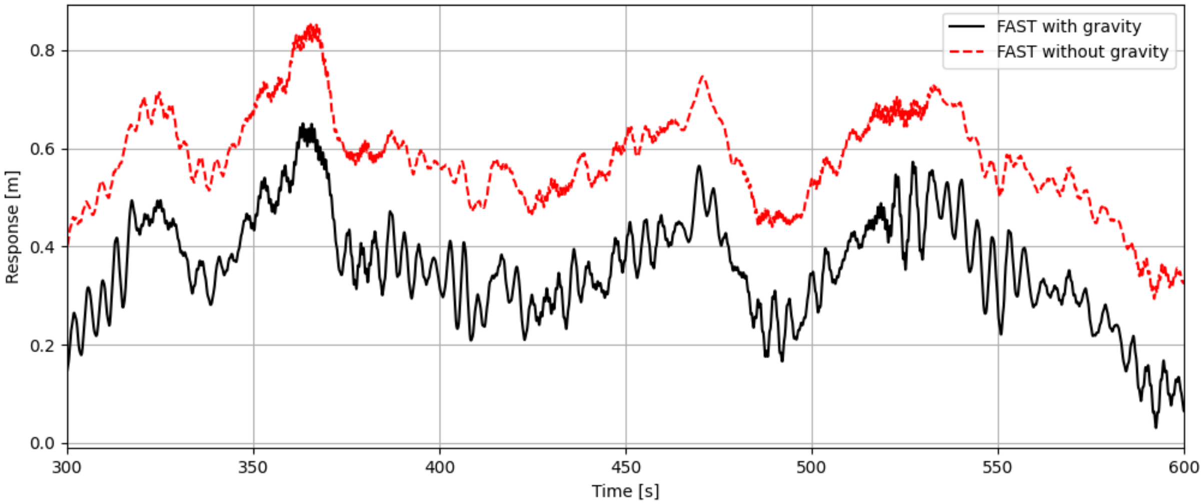


Figure 6.22: The monopile-structure's response in the top of the tower with and without gravity included in calculation. 12 m/s wind

In the modal analysis, gravity is not included in the loading. Assuming linear geometry, the gravity, being a static load, will only affect the static part of the response, not the dynamic. A simple way to include gravity's effect is to use the static response calculated from FAST. This is the approach used in this thesis. A different approach is defining gravity as a load in Abaqus and combining the Abaqus gravity load with the wind load from FAST. Both loads can be exported from Abaqus in the same manner as the mass and stiffness matrix in Abaqus. In turn, they can be used as input in the modal analysis without further processing.

For the damping estimation in Abaqus, the gravity effect can be easily included. The authors also attempted this. However, the gravity will lead to significant strains and stresses in the soil that are difficult to separate from the strains and stresses from the movement of the offshore wind turbine. By including strains and stresses from gravity, the estimated energy loss from the soil will be heavily overestimated and lead to an unrealistically high level of damping. A possible solution might be first to estimate the energy loss from gravity and subtract this energy from the energy loss estimated from both gravity and the movement of the structure. However, it is not certain that this solution will yield accurate results, and further analysis must be performed to ensure this approach's correctness. If the approach can be performed, this would allow for setting a more realistic friction angle in the Mohr-Coulomb yield criteria. The angle could be set larger than 0 in the Mohr-Coulomb model, making the overall material model more realistic. In addition to the increased accuracy of including the gravity isolated.

6.11 Dynamic energy

The equation for dynamic modal energy was established in Section 2.5.1. It can be argued that since the total energy of the system is $\frac{1}{2}\{r\}^T[K]\{r\}$. The energy of the system cannot be split into a dynamic energy part and a static energy part as energy is not linearly dependent on {r}. However, the damping only applies to the dynamic part of the system. It would be unreasonable to include terms from the static part of the system in this estimation.

This can be understood easier by a 1 DOF-system analogy. The system displacement r is split into a static and dynamic displacement: r_s and r_d respectively. Such that the strain energy is:

$$\frac{1}{2}Kr^2 = \frac{1}{2}K(r_s + r_d)^2 = \frac{1}{2}K(r_s^2 + r_d^2 + 2r_d r_s) \tag{6.2}$$

Since the soil damping is proportional to $\frac{Energy\ dissipated\ per\ cycle}{Total\ energy\ of\ relevant\ system}$ and the relevant system is the system corresponding to dynamic displacement r_d. This system have strain energy $\frac{1}{2}Kr_d^2$. Therefore all terms containing r_s should be removed. The dynamic energy per mode can be calculated as in Section 2.5.1 Equation 2.61

6.12 Comparison between methods

Both method 1 and method 2 comes from the same line of energy-based reasoning of systems. Where method 2 assumes the knowledge of some foundation damping D_f, this assumption is unnecessary for method 1. Method 1 is a physics-based approach, and all sizes in the damping estimation are physical sizes. The drawback is a higher computational cost and complexity when calculating the damping and a loss of generalisation. The damping estimated is strictly speaking only valid for the system on which the damping calculation is based. It cannot in any simple way be transformed to a different system. In addition, a certain level of loading is assumed in the calculations. It is uncertain how well these loading levels generalise to other levels of loading. Therefore, an analysis should be performed on how much the damping varies with different loading levels.

Method 2 could, because of its simplicity, easily be implemented in any modal analysis tool without any external input other than D_f. However, a methodology to find D_f has to be developed in this case. A possible approach to calculating hysteric modal damping for only a subpart of the system is discussed in Section 6.16. An approach to calculate non-modal damping in the foundation is discussed in Section 6.13

6.13 Damping estimation for time-domain integration

Most FE-software typically use numerical time integration such as the Hilber-Hughes-Taylor alpha algorithm to solve dynamical problems. The time integration approach solves the problem in the time domain, while the modal-analysis approach solves the system in the frequency domain. Time integration algorithms do not use modal damping in their calculations, but use the damping matrix directly. Method 1 in Section 2.5.1 can be expanded to provide this damping.

Instead of calculating the representative displacement and dynamic energy per mode, the sizes can be calculated for the foundation as a whole. The representative dynamic displacement of the piles can be calculated by finding the mean-subtracted average peak height in each direction in the foundation. Each mean-subtracted average peak height in each direction can, in turn, be used to calculate the dynamic energy in the foundation and the corresponding energy loss. The estimate can be used to calculate the foundation damping in each direction. The generation of the damping matrix can be achieved by transforming the foundation damping into a dash-pot damper.

6.14 Soil energy loss

The energy loss in the soil shows exponential growth. As described in detail in Section 2.5.1 and Figure 2.11, the exponential behaviour of the energy loss at different levels of displacement (Figure 6.23) involve calculating the energy loss as $W_{Ldynamic} = W_{Lstatic+dynamic} - W_{Lstatic}$. This will give higher estimated damping than when ignoring the static part. The validity of this approach depends on the soil's material model not changing due to long term static loading. How the physical material behaviour of soil behaves after being exposed to static loading over time remains unclear and requires an understanding of soils beyond the scope of this thesis research.

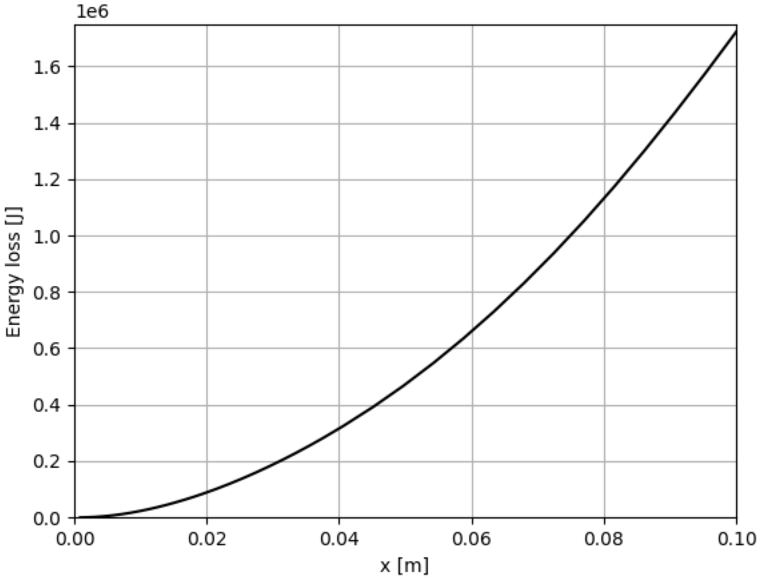


Figure 6.23: Energy loss per cycle for different levels of displacement in x-direction for the monopile structure

6.15 Work hardening

Most materials experience some work hardening when exposed to plastic strains. In most cases, ignoring work-hardening and using a perfect plastic model would be a conservative approximation. In damping estimation, an approximation that underestimates the material yield capacity would give higher damping, making the approach non-conservative. This effect is likely to be minor. A possible solution would be to include work-hardening in the model. If work hardening is applied, a forced displacement should be applied at the top of the pile equal to the highest displacement recorded. After, when the representative loading is later added, the material model should be more realistic.

6.16 Damping subsystem only

In FAST, the calculation of the structure is divided into Subdyn (substructure) and Elastodyn (tower). It might be easier to apply the damping if only one of the substructures is damped. The substructure damping can be calculated by using only the dynamic energy of the system in question. An approach to calculating the modal damping of only the subsystem is given as:

1. Calculate the representative modal response
2. Transfer the modal response to physical coordinates for each mode
3. Calculate the modal dynamic energy for the part of interest by only including the subsystem's DOFs in the dynamic energy calculation
4. Calculate the damping ratio for the subsystem by using the energy loss in the foundation and the dynamic energy of the subsystem.

6.17 Sources of error

This section presents this thesis's known sources of error and possible solutions to fix them.

- The modal analysis (Section 5) did not include aerodynamic damping, hydrodynamic damping or wave forces. This led to an inaccurate estimation of the OWTs response and thus to inaccuracies in the estimated soil damping. This is discussed in Section 6.8 and possible solutions are proposed in Section 6.9.

- As mentioned in Section 6.12 the estimated soil damping is strictly speaking only valid for the load case for it was estimated. As a result, the estimated soil damping can be inappropriate for many load cases used in the fatigue analysis. Therefore, it would be interesting to see how much soil damping varies between loading cases. If the soil damping varies greatly between different load cases, it might be necessary to update the soil damping for each load case.

- It may be inappropriate to link the modes from FAST directly to the modes in Abaqus, as these modes are slightly different. Therefore, it may be slightly inaccurate to use the estimated soil damping from the modes in the modal analysis (Section 5) on the modes in FAST. A possible solution is proposed in Section 6.9.

- FAST has limitations on which modes can be damped. Therefore, the authors used an approximate approach, as described in Section 4.3.4. However, this approximation might lead to inaccuracies in the applied soil damping.

- Certain fatigue parameters may have been improperly set up. The ultimate load ($LUlt$) is particularly challenging to determine. Section 4.5.1 mentions several additional methods for establishing the ultimate load.

7. Conclusion

The authors developed a model for estimating the overall hysteric modal soil damping. The model was applied on a 10 MW jacket structure and a 10 MW monopile structure. Damping estimation was implemented using Abaqus, and wind loads were extracted from FAST for a 12 m/s wind. An estimate for the radiation damping was included for the jacket structure. The hysteric damping in the first fore-aft mode for the monopile is significantly higher than the literature in the field. No previous damping estimation was found for OWT jacket structures.

The soil-damping estimates of the two structures in the fore-aft modes are given in Table 7.1. Giving a weighted average of total soil damping of the fore-aft modes of 3.50% and 4.44%, based on the 3 and 4 first fore-aft modes, of the monopile and jacket, respectively. The hysteric damping had the largest impact on the fore-aft modes. The effect on the side-to-side modes were much smaller.

Table 7.1: Damping estimated for fore-aft modes

	Monopile		**Jacket**			
Fore-aft mode	**Frequency [Hz]**	**Hysteric damping [%]**	**Frequency [Hz]**	**Hysteric damping [%]**	**Radiation damping [%]**	**Total damping [%]**
1	0.253	2.95	0.274	3.50	0.03	3.53
2	1.164	7.90	1.299	6.35	0.60	6.95
3	2.245	3.09	2.22	2.60	1.53	4.13
4	-	-	3.529	0.46	2.74	3.20

The authors performed five different fatigue analyses on the model using five different levels of damping. The 10 MW jacket structure and the 10 MW monopile structure were the basis for the fatigue estimation. First, an analysis was performed for each structure with a 1.0% base level of overall damping, forming the baseline fatigue life. A subsequent analysis was then performed by adding the hysteric soil damping to the base level damping for both structures. Finally, an analysis was performed, adding both hysteric and radiation damping to the baseline damping for the jacket structure.

The analysis showed a high increase in fatigue life of 77% for the monopile. Critical sections in the jacket substructure experienced a very high 274% increase in fatigue durability by simply adding hysteric damping. Adding hysteric and radiation to the jacket increased fatigue life by 285% over the baseline situation. These results suggest that a turbine with a 20-year service life could, on the low end, produce electricity for 15 additional years. The jacket, which sees greater benefits from the foundation damping, could last 57 additional years.

There is a level of uncertainty connected to the estimated damping. Partially because wave loads, hydrodynamic damping and aerodynamic damping are not included in the hysteric damping estimation, resulting in an inaccurate response in the foundation. It proved difficult to translate the estimated damping values to OpenFAST simulation accurately. As for the fatigue analysis, there were uncertainties associated with establishing a faithful ultimate load for each cross-section.

The estimated soil damping for fatigue was applied in a ULS load case with a 50 m/s wind load affecting the monopile turbine. The maximum moment for the critical section was reduced by 12%.

The soil contributes a significant level of damping. Including this damping significantly increases the fatigue life. In addition, it slightly reduces moments associated with ULS. Increased fatigue life could result in longer service life for the structure or reduced use of materials. Reduced ULS toll on the structure could result in reduced use of materials. The cost and carbon footprint per unit of electricity produced by OWTs could therefore be reduced by including soil damping in the engineering of OWTs.

8. Suggestions for further work

Based on the performed research the authors believe that several of the following subjects could be interesting to explore further.

- To estimate the soil damping from physical measurements and using these results to validate and calibrate a damping estimation technique would be incredibly beneficial. It would give a much higher validity to the estimation than that of a purely numerical approach.
- In order to further validate the model used in this study, it might be interesting to apply the model to a free decay case. The results could then be more accurately compared to the results of Damgaard et al. [15] and Versteijlen et al. [16].
- Incorporate a more accurate loading and damping representation by using the FAST response instead of the modal analysis as discussed in Section 6.9. By using the FAST response the analysis would include wave loading, and hydrodynamic and aerodynamic damping. Thus solving the problems discussed in Section 6.8. Using the FAST approach would also make it possible to estimate the damping in ULS. A more accurate estimation of loading and damping could also be solved by accounting for these directly in the modal analysis. However, the authors expect it to be difficult to implement the loading and damping accurately in practice.
- As discussed in Section 6.10, the model could become more accurate by fully accounting for gravity.
- Choose a better ultimate load in MLife to achieve a more realistic durability.
- Run several versions of the same model with variation in loading and soil parameters to create statistical data points. In essence, a sensitivity analysis. Performing a sensitivity analysis would be vital in understanding the damping estimation model's uncertainty and which parameters have an effect. Much of the model is parameterised, which would make it possible to automate much of the process.
- Run a more rigorous fatigue analysis with more wind and wave seeds and check different members for increased analysis accuracy.
- Estimate the damping for different foundation types, such as Caisson and gravity-based foundations. Furthermore, review the jacket damping with the updated design, with improved fatigue life.
- Could be interesting to see if there is any effect from implementing work-hardening as discussed in Section 6.15. If work hardening is implemented, it could be beneficial to run the model through some realistic loading first to get a more accurate level of accumulated plastic strain.
- To explore the relationship between turbine size and foundation damping.
- It might be intriguing to see how the results vary with different soil profiles and damping models, for instance, using Darandelli curves [82].

Bibliography

[1] European commission. (2022, March) Repowereu: Joint european action for more affordable, secure and sustainable energy. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1511

[2] J. O. G. Tande. (2022, March) Offshore wind: the eu needs to invest now. [Online]. Available: https://blog.sintef.com/sintefenergy/offshore-wind-the-eu-needs-to-invest-now/

[3] A. Athanasia, G. Anne-Bénédicte, and M. Jacopo, "The offshore wind market deployment: Forecasts for 2020, 2030 and impacts on the european supply chain development," *Energy Procedia*, vol. 24, pp. 2–10, 2012. [Online]. Available: https://doi.org/10.1016/j.egypro.2012.06.080

[4] A. Durakovic. (2021) World's largest, most powerful wind turbine stands complete. [Online]. Available: https://www.offshorewind.biz/2021/11/12/worlds-largest-most-powerful-wind-turbine-stands-complete/

[5] "Global wind report 2021," Global Wind Energy Council, Tech. Rep., 2021. [Online]. Available: https://gwec.net/global-wind-report-2021/

[6] "Openfast documentation," National Renewable Energy Laboratory, Tech. Rep., Mar. 2022. [Online]. Available: https://openfast.readthedocs.io/_/downloads/en/v3.0.0/pdf/

[7] IEA, "Tracking offshore wind 2020," 2020. [Online]. Available: https://www.iea.org/reports/tracking-offshore-wind-2020

[8] R. Rezai, P. Fromme, and P. Duffor, "Fatigue life sensitivity of monopile-supported offshore wind turbine to damping," *Renewable Energy*, pp. 450–459, 2018. [Online]. Available: https://doi.org/10.1016/j.renene.2018.02.086

[9] "Wind energy generation systems - part 3-1: Design requirements for fixed offshore wind turbines," International Electrotechnical Commission, Geneva, Standard, 2020.

[10] "Petroleum and natural gas industries — specific requirements for offshore structures — part 4: Geotechnical and foundation design considerations," International Organization for Standardization, Geneva, Switzerland, Standard, Jul. 2016.

[11] "Offshore soil mechanics and geotechnical engineering," Det Norske Veritas, Norway, Standard, Sep. 2019.

[12] "Support structures for wind turbines," Det Norske Veritas, Norway, Standard, Dec. 2021.

[13] W. Carswell, S. R. Arwade, J. Johansson, and D. J. DeGroot, "Influence of foundation damping on offshore wind turbine monopile design loads," *Marine Structures*, 2022. [Online]. Available: https://doi.org/10.1016/j.marstruc.2021.103154

[14] W. Carswell, J. Johansson, F. Løvholt, S. Arwade, C. Madshus, D. DeGroot, and A. Myers, "Foundation damping and the dynamics of offshore wind turbine monopiles," *Renewable Energy*, pp. 724–736, 2015. [Online]. Available: https://doi.org/10.1016/j.renene.2015.02.058

[15] M. Damgaard, L. Ibsen, L. Andersen, and J. Andersen, "Cross-wind modal properties of offshore wind turbines identified by full scale testing," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 116, pp. 94–108, 2013. [Online]. Available: https://doi.org/10.1016/j.jweia.2013.03.003

[16] W. G. Versteijlen, A. Metrikine, J. Hoving, E. Smidt, and W. de Vries, "Estimation of the vibration decrement of an offshore wind turbine support structure caused by its interaction with soil," *Knowledge, Technology Policy*, 2011.

[17] R. Shirzadeh, C. Devriendt, M. Bidakhvidi, and P. Guillaume, "Experimental and computational damping estimation of an offshore wind turbine on a monopile foundation," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 120, pp. 96–106, 2013. [Online]. Available: https://doi.org/10.1016/j.jweia.2013.07.004

[18] Y. Zhang, K. K. Aamodt, and A. M. Kaynia, "Hysteric damping model laterally loaded piles," *Marine Structures*, 2021. [Online]. Available: https://doi.org/10.1016/j.marstruc.2020.102896

[19] A. M. Markou, Athanasios A.; Kaynia, "Nonlinear soil-pile interaction for offshore wind turbines," *Wind Energy*, vol. 21, no. 7, pp. 558–574, 2018. [Online]. Available: https://doi.org/10.1002/we.2178

[20] A. Kaynia and K. Andersen, "Development of nonlinear foundation springs for dynamic analysis of platforms," pp. 1067–1072, May 2015. [Online]. Available: https://doi.org/10.1201/b18442

[21] C. Cruz and E. Miranda, "Evaluation of soil-structure interaction effects on the damping ratios of buildings subjected to earthquakes," *Soil Dynamics and Earthquake Engineering*, vol. 100, pp. 183–195, 2017. [Online]. Available: https://doi.org/10.1016/j.soildyn.2017.05.034

[22] S. Aasen, A. Page, K. Skau, and T. Nygaard, "Effect of the foundation modelling on the fatigue lifetime of a monopile-based offshore wind turbine," *Wind Energy Science Discussions*, pp. 361–376, 2017. [Online]. Available: https://doi.org/10.5194/wes-2016-37

[23] A. Malekjafarian, S. Jalilvand, P. Doherty, and D. Igoe, "Foundation damping for monopile supported offshore wind turbines: A review," *Marine Structures*, vol. 77, p. 102937, 2021. [Online]. Available: https://doi.org/10.1016/j.marstruc.2021.102937

[24] T. Ishihara and L. Wang, "A study of modal damping for offshore wind turbines considering soil properties and foundation types," *Wind energy*, vol. 22, pp. 1760–1778, 2019. [Online]. Available: https://doi.org/10.1002/we.2401

[25] C. Chi-Yu, Z. Yi-Qing, T.-Y. Lin, B. Nelson, and H.-H. Huang, "Comparative study of time-domain fatigue assessments for an offshore wind turbine jacket substructure by using conventional grid-based and monte carlo sampling methods," *Energies*, vol. 11, no. 11, Nov. 2018. [Online]. Available: https://www.proquest.com/scholarly-journals/comparative-study-time-domain-fatigue-assessments/docview/2316359782/se-2

[26] Michel, Klein, Butenweg, and Klinkel, "The influence of soil-structure-interaction on the fatigue analysis in the foundation design of onshore wind turbines," *Procedia engineering*, vol. 199, pp. 3218–3223, 2017.

[27] B. Yeter, Y. Garbatov, and C. Guedes Soares, "Evaluation of fatigue damage model predictions for fixed offshore wind turbine support structures," *International Journal of Fatigue*, vol. 87, pp. 71–80, Jun. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142112316000086

[28] S. H. Sørum, J. R. Krokstad, and J. Amdahl, "Wind-wave directional effects on fatigue of bottom-fixed offshore wind turbine," 2019. [Online]. Available: http://hdl.handle.net/11250/2635996

[29] N. Barltrop, *Dynamics of fixed marine structures*, 3rd ed. Oxford: Butterworth-Heinemann, 1991.

[30] J. Schijve, "Fatigue of structures and materials in the 20th century and the state of the art," *International Journal of Fatigue*, vol. 25, no. 8, pp. 679–702, 2003.

[31] W. J. M. Rankine, "On the causes of the unexpected breakage of the journals of railway axles; and on the means of preventing such accidents by observing the law of continuity in their construction." Jan. 1843.

[32] D. Drang, "Bridge failure detail," Jan. 2014. [Online]. Available: https://www.flickr.com/photos/drdrang/12140071645/

[33] W. Schutz, "A history of fatigue," *Engineering fracture mechanics*, vol. 54, no. 2, pp. 263–300, 1996.

[34] "Design of offshore wind turbine structures," Det Norske Veritas, Norway, Standard, May 2014.

[35] “Eurocode 3: Design of steel structures - part 1-9: Fatigue,” European Committee for Standarization, Brussels, Standard, Feb. 2010.

[36] S. Suresh, *Fatigue of materials*, 2nd ed. Cambridge: Cambridge University Press, 1998.

[37] C. Lalanne, *Fatigue damage*, 3rd ed., ser. Mechanical Vibration and Shock Analysis ;. London, England: ISTE Ltd : John Wiley Sons, 2014, vol. 4.

[38] J. Goodman, *Mechanics applied to engineering*, 8th ed. London: Longmans, Green, and Co, 1914.

[39] J. O. Smith, *The effect of range of stress on the fatigue strength of metals*, ser. Bulletin (University of Illinois. Engineering Experiment Station), 1942, vol. 334.

[40] M. A. Miner, “Cumulative Damage in Fatigue,” *Journal of Applied Mechanics*, vol. 12, no. 3, pp. A159–A164, Sep. 1945. [Online]. Available: https://doi.org/10.1115/1.4009458

[41] M. Karimirad, *Offshore Energy Structures: For Wind Power, Wave Energy and Hybrid Marine Platforms.* Cham: Springer International Publishing AG, 2014.

[42] “Loads and site conditions for wind turbines,” Det Norske Veritas, Norway, Standard, Nov. 2021.

[43] M. Haddara and C. Guedes Soares, “Wind loads on marine structures,” vol. 12, no. 3, pp. 199–209, 1999.

[44] C. C. Baniotopoulos, *Design of Wind-Sensitive Structures.* Vienna: Springer Vienna, 2007, pp. 201–227. [Online]. Available: https://doi.org/10.1007/978-3-211-73076-8_7

[45] “Eurocode 1: Actions on structures, part 1-4: General actions, wind actions,” European Committee for Standarization, Brussels, Standard, Apr. 2009.

[46] F. Vorpahl, H. Schwarze, T. Fischer, M. Seidel, and J. Jonkman, “Offshore wind turbine environment, loads, simulation, and design,” *Wiley Interdisciplinary Reviews: Energy and Environment*, vol. 2, Sep. 2013. [Online]. Available: https://doi.org/10.1002/wene.52

[47] T. Fischer, W. De Vries, and B. Schmidt, “Upwind design basis (wp4: Offshore foundations and support structures),” Stuttgart, Germany, Oct. 2010.

[48] I. Van der Hoven, “Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour,” *Journal of Atmospheric Sciences*, vol. 14, no. 2, pp. 160–164, Apr. 1957. [Online]. Available: https://doi.org/10.1175/1520-0469(1957)014<0160:PSOHWS>2.0.CO;2

[49] J. C. Kaimal, J. C. Wyngaard, Y. Izumi, and O. R. Cote, *Spectral Characteristics of Surface-Layer Turbulence.* Quarterly Jnl. of the Royal Meteorological Society, Jul. 1972, vol. v98.

[50] “N400 bruprosjektering,” Norwegian Public Roads Administration, Norway, Standard, Jan. 2022.

[51] “Wind energy generation systems - part 1: Design requirements,” International Electrotechnical Commision, Geneva, Standard, Feb. 2019.

[52] W. Weibull, “A Statistical Distribution Function of Wide Applicability,” *Journal of Applied Mechanics*, vol. 18, no. 3, pp. 293–297, Sep. 1951. [Online]. Available: https://doi.org/10.1115/1.4010337

[53] P. Liu, *A general theory of fluid mechanics.* Singapore: Springer, 2021.

[54] “Wind energy generation systems - part 3-1: Design requirements for fixed offshore wind turbines,” International Electrotechnical Commision, Geneva, Standard, Apr. 2019.

[55] O. S. Hopperstad and T. Børvik, *Materials Mechanics.* Compendium from TKT4135 at NTNU, 2017.

[56] Abaqus. Mohr-coulomb model. [Online]. Available: https://abaqus-docs.mit.edu/2017/English/SIMACAETHERefMap/simathe-c-mohrcoulomb.htm

[57] R. R. Craig and M. C. C. Bampton, "Coupling of substructures for dynamic analyses," *AIAA journal*, vol. 6, no. 7, pp. 1313–1319, 1968.

[58] W. C. Hurty, "Dynamic analysis of structural systems using component modes," *AIAA journal*, vol. 3, no. 4, pp. 678–685, 1965.

[59] L. Arany, S. Bhattacharya, J. Macdonald, and S. Hogan, "Closed form solution of eigen frequency of monopile supported offshore wind turbines in deeper waters incorporating stiffness of substructure and ssi," *Soil Dynamics and Earthquake Engineering*, vol. 83, pp. 18–32, 2016. [Online]. Available: https://doi.org/10.1016/j.soildyn.2015.12.011

[60] B. Amir-Faryar, M. S. Aggour, and R. H. McCuen, "Universal model forms for predicting the shearmodulus and material damping of soils," *Geomechanics and Geoengineering*, vol. 12:1, pp. 60–71, 2017. [Online]. Available: http://dx.doi.org/10.1080/17486025.2016.1162332

[61] A. M. Kaynia, "Dynamic stiffness and seismic response of pile groups," MIT, Dept. of Civil Eng, MIT, Cambridge, USA., Tech. Rep., 1982.

[62] A. M. Kaynia and E. Kausel, "Dynamics of piles and pile groups in layered soil media," *Soil Dynamics and Earthquake Engineering*, vol. 10, no. 8, pp. 386–401, 1991. [Online]. Available: https://doi.org/10.1016/0267-7261(91)90053-3

[63] D. M. Pedersen and H. Askheim, "Implementation of seismic soil-structure interaction in openfast and application to a 10mw offshore wind turbine on jacket structure," 2021. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2824677

[64] T. von Borstel, "Innwind.eu design report -reference jacket," Oct. 2013. [Online]. Available: http://www.innwind.eu/publications/deliverable-reports

[65] P. Bortolotti, H. C. Tarres, K. Dykes, K. Merz, L. Sethuraman, D. Verelst, and F. Zahle, "Iea wind task 37 on systems engineering in wind energy wp2.1 reference wind turbines," 2019. [Online]. Available: https://www.nrel.gov/docs/fy19osti/73492.pdf

[66] "Openfast documentation," National Renewable Energy Laboratory, Tech. Rep., Apr. 2022. [Online]. Available: https://openfast.readthedocs.io/_/downloads/en/main/pdf/

[67] "Openfast v3.0.0," National Renewable Energy Laboratory, Tech. Rep., Jun. 2021. [Online]. Available: https://github.com/OpenFAST/openfast/releases/tag/v3.0.0

[68] B. Jonkman and J. Jonkman, "Fast v8.16.00a-bjj," Jul. 2016. [Online]. Available: https://www.nrel.gov/wind/nwtc/assets/downloads/FAST/FASTv8.16/README_FAST8.pdf

[69] F. E. Cellier and E. Kofman, *Continuous System Simulation.* Boston, MA: Springer US, 2006. [Online]. Available: https://doi.org/10.1007/0-387-30260-3_4

[70] J. M. Jonkman and M. L. Buhl Jr., "Fast user's guide," Oct. 2005. [Online]. Available: https://www.nrel.gov/docs/fy06osti/38230.pdf

[71] A. Platt, B. Jonkman, and J. Jonkman, "Inflowwind user's guide," Jul. 2016. [Online]. Available: https://www.nrel.gov/wind/nwtc/inflowwind.html

[72] "Ice load project final technical report," Det Norske Veritas, Washington, Report, Oct. 2014.

[73] G. S. Bir, "User's guide to bmodes (software for computing rotating beam coupled modes)," Sep. 2007. [Online]. Available: https://www.nrel.gov/wind/nwtc/assets/pdfs/bmodes.pdf

[74] B. J. Jonkman, “Turbsim user’s guide,” Jun. 2016. [Online]. Available: https://www.nrel.gov/wind/nwtc/assets/downloads/TurbSim/TurbSim_v2.00.pdf

[75] G. Hayman and M. Buhl Jr., “Mlife user’s guide for version 1.00,” Oct. 2012. [Online]. Available: https://www.nrel.gov/wind/nwtc/assets/pdfs/mlife-user.pdf

[76] B. Jonkman and E. Branlard, “Matlab toolbox for openfast, including mbc3,” Feb. 2022. [Online]. Available: https://github.com/OpenFAST/matlab-toolbox

[77] E. Branlard, “pydatview,” Jun. 2021. [Online]. Available: https://github.com/ebranlard/pyDatView

[78] P. Bortolotti, “Iea-10.0-198-rwt,” May 2020. [Online]. Available: https://github.com/IEAWindTask37/IEA-10.0-198-RWT/tree/offshore

[79] “Eurocode 3: Design of steel structures - part 1-1: General rules and rules for buildings,” European Committee for Standarization, Brussels, Standard, Feb. 2015.

[80] “Fatigue design of offshore steel structures,” Det Norske Veritas, Norway, Standard, Sep. 2021.

[81] M. Stolpe, W. N. Wandji, A. Natarajan, R. Shirzadeh, M. Kühn, and D. Kaufer, “Deliverable d4.34 - innovative design of a 10mw steel-type jacket,” Jun. 2016. [Online]. Available: http://www.innwind.eu/publications/deliverable-reports

[82] M. B. Darendeli, “Development of a new family of normalized modulus reduction and material damping curves,” 2001.

Appendices

A. Modal data

Table A.1: The average peak in physical space in the foundation for monopile modes

Mode	**x** **[m]**	**y** **[m]**	**z** **[m]**	**rx** **[rad]**	**ry** **[rad]**	**rz** **[rad]**
1	-9.17E-16	1.12E-04	5.77E-12	-1.18E-05	1.95E-15	-3.09E-13
2	8.35E-04	-1.13E-11	-6.70E-11	8.47E-13	8.82E-05	-2.68E-12
3	1.86E-15	-1.84E-04	-1.27E-11	1.66E-05	-3.50E-15	5.62E-13
4	5.24E-04	-1.37E-12	-5.45E-11	1.13E-14	4.65E-05	-1.24E-12
5	1.68E-13	6.46E-12	4.48E-12	8.31E-14	-3.64E-13	-4.35E-05
6	6.17E-14	-5.07E-05	-4.46E-12	3.88E-06	4.14E-15	1.38E-13
7	-9.59E-04	1.63E-12	1.29E-10	4.81E-14	-6.99E-05	1.98E-12
8	-7.33E-14	3.78E-06	6.20E-13	-2.01E-07	-3.82E-15	-1.21E-14
9	4.84E-05	8.72E-13	-1.17E-11	-5.17E-14	2.52E-06	-1.10E-13
10	4.19E-13	-3.34E-13	-1.06E-06	-1.23E-14	-1.69E-14	6.55E-15
11	1.46E-14	-6.57E-07	-5.56E-14	2.06E-08	4.63E-16	1.20E-14
12	-9.31E-06	-1.96E-13	1.11E-12	6.79E-15	-2.90E-07	1.31E-13
13	-2.38E-13	-2.85E-13	6.02E-14	1.29E-14	-9.12E-15	-7.67E-07
14	-2.14E-13	1.63E-13	3.35E-07	3.02E-15	3.88E-15	-1.03E-16
15	-9.36E-15	2.01E-07	-4.52E-14	-3.30E-09	-1.58E-16	-1.69E-16

Table A.2: The average peak in physical space in the foundation for jacket modes leg 1

	Leg 1					
	x[m]	**y[m]**	**z[m]**	**rx[rad]**	**ry[rad]**	**rz[rad]**
mode1	-1.10E-05	-3.77E-05	-3.53E-05	4.62E-06	-7.93E-07	-1.57E-07
mode2	-0.0001498	-4.35E-05	-0.00013911	3.13E-06	-1.84E-05	6.22E-07
mode3	-2.57E-05	-0.0002013	-8.05E-05	2.63E-05	-1.82E-06	-1.08E-06
mode4	-0.00027938	0.00027938	-1.19E-11	-3.80E-05	-3.80E-05	-6.59E-06
mode5	0.00057237	6.83E-05	0.00021342	-4.85E-06	7.51E-05	-3.41E-06
mode6	4.75E-06	6.16E-05	1.45E-05	-8.26E-06	3.51E-07	7.09E-07
mode7	-0.0015231	-0.0001006	-0.00030219	7.66E-06	-0.00020574	2.17E-05
mode8	-1.93E-07	-2.84E-05	1.07E-08	4.06E-06	-9.27E-08	-1.66E-06
mode9	-0.00043271	-6.11E-07	8.25E-06	1.46E-06	-6.23E-05	2.91E-05
mode10	-1.95E-16	1.95E-16	1.07E-20	-8.77E-18	-8.78E-18	-1.60E-15
mode11	1.04E-06	1.04E-06	3.17E-06	-7.16E-08	7.16E-08	5.48E-13
mode12	1.97E-07	-2.59E-07	7.46E-07	9.75E-08	-2.29E-08	-6.29E-07
mode13	4.10E-06	1.97E-06	7.35E-06	2.30E-07	-3.79E-08	6.21E-06
mode14	3.33E-05	-3.33E-05	6.31E-11	4.69E-06	4.69E-06	2.15E-07
mode15	1.06E-10	1.09E-10	2.19E-10	1.42E-10	-1.43E-10	2.72E-14

Table A.3: The average peak in physical space in the foundation for jacket modes leg 2

	Leg 2					
	x[m]	**y[m]**	**z[m]**	**rx[rad]**	**ry[rad]**	**rz[rad]**
mode1	1.10E-05	-3.77E-05	-3.53E-05	4.62E-06	7.94E-07	1.57E-07
mode2	-0.0001498	4.35E-05	0.00013911	-3.13E-06	-1.84E-05	6.21E-07
mode3	2.57E-05	-0.0002013	-8.05E-05	2.63E-05	1.82E-06	1.08E-06
mode4	-0.00027938	-0.00027938	-1.19E-11	3.80E-05	-3.80E-05	-6.59E-06
mode5	0.00057237	-6.83E-05	-0.00021342	4.85E-06	7.51E-05	-3.41E-06
mode6	-4.75E-06	6.16E-05	1.45E-05	-8.26E-06	-3.50E-07	-7.09E-07
mode7	-0.0015231	0.00010063	0.00030219	-7.67E-06	-0.00020574	2.17E-05
mode8	1.92E-07	-2.84E-05	1.06E-08	4.06E-06	9.25E-08	1.66E-06
mode9	-0.00043271	6.27E-07	-8.25E-06	-1.46E-06	-6.23E-05	2.91E-05
mode10	1.95E-16	1.95E-16	-1.07E-20	-8.78E-18	8.77E-18	1.60E-15
mode11	-1.04E-06	1.04E-06	3.17E-06	-7.16E-08	-7.16E-08	5.48E-13
mode12	-1.97E-07	-2.59E-07	7.46E-07	9.75E-08	2.29E-08	6.29E-07
mode13	4.10E-06	-1.97E-06	-7.35E-06	-2.30E-07	-3.79E-08	6.20E-06
mode14	3.33E-05	3.33E-05	6.31E-11	-4.69E-06	4.69E-06	2.15E-07
mode15	-1.09E-10	1.06E-10	2.19E-10	1.43E-10	1.42E-10	2.72E-14

Table A.4: The average peak in physical space in the foundation for jacket modes leg 3

	Leg 3					
	x[m]	**y[m]**	**z[m]**	**rx[rad]**	**ry[rad]**	**rz[rad]**
mode1	-1.10E-05	-3.77E-05	3.53E-05	4.62E-06	-7.93E-07	1.57E-07
mode2	-0.0001498	-4.35E-05	0.00013911	3.13E-06	-1.84E-05	-6.22E-07
mode3	-2.57E-05	-0.0002013	8.05E-05	2.63E-05	-1.82E-06	1.08E-06
mode4	0.00027938	-0.00027938	-1.22E-11	3.80E-05	3.80E-05	-6.59E-06
mode5	0.00057237	6.83E-05	-0.00021342	-4.85E-06	7.51E-05	3.41E-06
mode6	4.75E-06	6.16E-05	-1.45E-05	-8.26E-06	3.51E-07	-7.09E-07
mode7	-0.0015231	-0.0001006	0.00030219	7.66E-06	-0.00020574	-2.17E-05
mode8	-1.93E-07	-2.84E-05	-1.07E-08	4.06E-06	-9.27E-08	1.66E-06
mode9	-0.00043271	-6.11E-07	-8.25E-06	1.46E-06	-6.23E-05	-2.91E-05
mode10	1.95E-16	-1.95E-16	1.07E-20	8.77E-18	8.78E-18	-1.60E-15
mode11	-1.04E-06	-1.04E-06	3.17E-06	7.16E-08	-7.16E-08	5.48E-13
mode12	1.97E-07	-2.59E-07	-7.46E-07	9.75E-08	-2.29E-08	6.29E-07
mode13	4.10E-06	1.97E-06	-7.35E-06	2.30E-07	-3.79E-08	-6.21E-06
mode14	-3.33E-05	3.33E-05	6.31E-11	-4.69E-06	-4.69E-06	2.15E-07
mode15	-1.06E-10	-1.09E-10	2.19E-10	-1.42E-10	1.43E-10	2.72E-14

Table A.5: The average peak in physical space in the foundation for jacket modes leg 4

	Leg 4					
	x[m]	**y[m]**	**z[m]**	**rx[rad]**	**ry[rad]**	**rz[rad]**
mode1	1.10E-05	-3.77E-05	3.53E-05	4.62E-06	7.94E-07	-1.57E-07
mode2	-0.0001498	4.35E-05	-0.00013911	-3.13E-06	-1.84E-05	-6.21E-07
mode3	2.57E-05	-0.0002013	8.05E-05	2.63E-05	1.82E-06	-1.08E-06
mode4	0.00027938	0.00027938	-1.21E-11	-3.80E-05	3.80E-05	-6.59E-06
mode5	0.00057237	-6.83E-05	0.00021342	4.85E-06	7.51E-05	3.41E-06
mode6	-4.75E-06	6.16E-05	-1.45E-05	-8.26E-06	-3.50E-07	7.09E-07
mode7	-0.0015231	0.00010063	-0.00030219	-7.67E-06	-0.00020574	-2.17E-05
mode8	1.92E-07	-2.84E-05	-1.06E-08	4.06E-06	9.25E-08	-1.66E-06
mode9	-0.00043271	6.27E-07	8.25E-06	-1.46E-06	-6.23E-05	-2.91E-05
mode10	-1.95E-16	-1.95E-16	-1.07E-20	8.78E-18	-8.77E-18	1.60E-15
mode11	1.04E-06	-1.04E-06	3.17E-06	7.16E-08	7.16E-08	5.48E-13
mode12	-1.97E-07	-2.59E-07	-7.46E-07	9.75E-08	2.29E-08	-6.29E-07
mode13	4.10E-06	-1.97E-06	7.35E-06	-2.30E-07	-3.79E-08	-6.20E-06
mode14	-1.80E-04	1.15E-05	-1.53E-05	-1.60E-06	-2.50E-05	-7.95E-06
mode15	-0.00019187	1.15E-05	-1.6305E-05	-1.63E-06	-2.67E-05	-8.39E-06

B. Python script damping estimation

Script to estimate damping in monopile. The logic is the same for the jacket

```python
from odbAccess import *

import numpy as np
import pickle
from operator import attrgetter

files_to_run = open("files_to_run.txt", "r")
step = 'displacement_load'


def values_to_array(values, dim=2, item='data', dtype=np.float64):
    '''Thanks to https://stackoverflow.com/a/46925902/8935243'''
    array = np.array(
        map(attrgetter(item), values),
        dtype=dtype,
    )
    return array


def get_local_damping(shear_array):
    # Sand- Mean Limit Seed and Idriss (1970)B. AMIR-FARYAR ET AL.
    shear_array *= 100
    damping = (12.6 * np.exp(0.9153 * np.log10(shear_array)) * (np.log10(shear_array + 1.797)) ** (-0.8939)) - 0.9283
    damping = damping * 10 ** -2
    return damping


for file_to_run in files_to_run.readlines():
    file_to_run = file_to_run.rstrip("\n")
    odb = openOdb(path=file_to_run + ".odb")
    print("")
    print(file_to_run)
    sett_soil = odb.rootAssembly.instances['SOIL'].elementSets['ALL']
    sett_pile1_top = odb.rootAssembly.instances['PILES'].nodeSets['PILE_1_TOP_NODE']
    sett_pile_tops_sett = np.array([sett_pile1_top])

    f = open("damping_data_" + file_to_run + ".txt", "w")
    f.write("Damping WT WL WT_top")
    for i in range(1, 2):
        f.write(" x" + str(i) + " y" + str(i) + " z" + str(i) + " rx" + str(i) + " ry" + str(i) + " rz" + str(i))
    for i in range(1, 2):
        f.write(" Fx" + str(i) + " Fy" + str(i) + " Fz" + str(i) + " Mx" + str(i) + " My" + str(i) + " Mz" + str(i))
    f.write("\n")
    for frame_index in range(len(odb.steps[step].frames) - 1,
                             len(odb.steps[step].frames)):  # skip first to avoid zero-division
        # Collect data from abaqus and convert to array
        frame = odb.steps[step].frames[frame_index]
        values_soil_stress = frame.fieldOutputs['S'].getSubset(region=sett_soil).values
        stress_soil = values_to_array(values_soil_stress)
        values_soil_strain = frame.fieldOutputs['LE'].getSubset(region=sett_soil).values
        strain_soil = values_to_array(values_soil_strain)
        strain_soil_org = strain_soil.copy()
        values_volume = frame.fieldOutputs['EVOL'].getSubset(region=sett_soil).values
        volume = values_to_array(values_volume, dim=1)
        elastic_strain_energy_abaqus = \
            odb.steps[step].historyRegions['Assembly ASSEMBLY'].historyOutputs['ALLSE'].data[frame_index][1]

        # Calculate damping
        gamma_1 = strain_soil_org[:, 3]
        gamma_2 = strain_soil_org[:, 4]
        gamma_3 = strain_soil_org[:, 5]
        D1 = get_local_damping(np.abs(gamma_1))
        D2 = get_local_damping(np.abs(gamma_2))
        D3 = get_local_damping(np.abs(gamma_3))
        shear_stress_1 = stress_soil[:, 3]
        shear_stress_2 = stress_soil[:, 4]
        shear_stress_3 = stress_soil[:, 5]
        AL = volume * (4 * np.pi) * 0.5 * (shear_stress_1 * gamma_1 * D1
                                           + shear_stress_2 * gamma_2 * D2
                                           + shear_stress_3 * gamma_3 * D3)
        WL = np.sum(AL)

        # Calculate energy loss from pile movement
        R = np.empty([1, 6])  # columns Fx Fy Fz Mx My Mz:  rows pile1
        r = np.empty([1, 6])  # columns  x y x rx ry rz:  rows pile1
        for i in range(1):
            # Get forces and moments
            values_pile_top_reaction_force = frame.fieldOutputs['RF'].getSubset(region=sett_pile_tops_sett[i]).values
            values_pile_top_reaction_moment = frame.fieldOutputs['RM'].getSubset(region=sett_pile_tops_sett[i]).values
            R[i, [0, 1, 2]] = values_to_array(values_pile_top_reaction_force)[0]
            R[i, [3, 4, 5]] = values_to_array(values_pile_top_reaction_moment)[0]
            # Get displacements and rotations
            values_pile_top_displacement = frame.fieldOutputs['U'].getSubset(region=sett_pile_tops_sett[i]).values
            values_pile_top_rotation = frame.fieldOutputs['UR'].getSubset(region=sett_pile_tops_sett[i]).values
            r[i, [0, 1, 2]] = values_to_array(values_pile_top_displacement)[0]
            r[i, [3, 4, 5]] = values_to_array(values_pile_top_rotation)[0]
        WT_wrong = 0.5 * np.sum(np.abs(r * R))  # see 9.3 appendix I

        DS = WL * (4 * np.pi * elastic_strain_energy_abaqus) ** -1
        print(
            "DS: " + str(DS) + "  WL: " + str(WL) + "  WT: " + str(elastic_strain_energy_abaqus) + "  WT_pile: " + str(
                WT_wrong))
        # Write results to file. See header for order  f.write("Damping WT WL WT_top")
        f.write(str(DS) + " " + str(elastic_strain_energy_abaqus) + " " + str(WL) + " " + str(WT_wrong))
        # Write reaction forces and displacements to file
        for disp in np.nditer(r[0]):
            f.write(" " + str(disp))
        for force in np.nditer(R[0]):
            f.write(" " + str(force))
        f.write("\n")
    f.close()
```

C. Modal analysis python scripts

C.1 Modal analysis for monopile structure

The modal analysis for monopile structure. All damping estimation with the exception of ULS was estimated without wave load. Method 2 is referred to as Kaynia damping in the script.

```python
import pandas as pd
import numpy as np
from scipy import linalg as LA
from scipy import signal as sig
from matplotlib import pyplot as plt
from scipy.integrate import simps

n_first_modes = 50  # number of modes to use in signal processing
terms = 2000  # terms in load fourier series
modal_damping_coefficient = 0.01
reduction_factor = 1  # extract every reduction_factor th element

# Load load series and convert to fourier series with zero mean
file_name = "10MW_Monopile_Base_EWM.out"
load_data = pd.read_csv(file_name, skiprows=[1, 2, 3, 4, 5, 7], sep="\t")
loads = ["RtAeroFxh", "RtAeroFyh", "RtAeroFzh", "RtAeroMxh", "RtAeroMyh", "RtAeroMzh"]
wave_loads = ["HydroFxi", "HydroFyi", "HydroFzi", "HydroMxi", "HydroMyi", "HydroFzi"]

fourier_loads = dict.fromkeys(loads, 0)
t = load_data["Time"].to_numpy()[::reduction_factor]

load_frequencies = np.zeros([terms])
static_load = np.array([])
for load in loads:
    L = t[len(t) - 1]
    y = load_data[load].to_numpy()[::reduction_factor]
    static_load = np.append(arr=static_load, values=np.mean(y))
    y = y - np.mean(y)
    samples = len(t)
    # Calculation of Co-efficients
    a0 = 2. / L * simps(y, t)
    an = lambda n: 2.0 / L * simps(y * np.cos(2 * np.pi * n * t / L), t)
    bn = lambda n: 2.0 / L * simps(y * np.sin(2 * np.pi * n * t / L), t)
    # Sum of the series
    load_fourier_series = np.zeros([terms, len(t)])
    for i in range(terms):
        k = i + 1
        load_fourier_series[i] = an(k) * np.cos(2. * np.pi * k * t / L) + bn(k) * np.sin(2. * np.pi * k * t / L)
        load_frequencies[i] = 2 * np.pi * k / L
    fourier_loads.update({load: load_fourier_series})
    s = load_fourier_series.sum(axis=0)
    correlation = np.corrcoef(y, s)[1, 0]
    # Plotting
    plt.plot(t, s, label="Fourier series")
    plt.plot(t, load_data[load].to_numpy()[::reduction_factor], "--", label="Original load")
    plt.xlabel("$Time[s]$    " + "correlation = " + str(np.corrcoef(y, s)[1, 0]))
    plt.ylabel("$Load[N] / Load[Nm]$")
    plt.legend(loc='best', prop={'size': 10})
    plt.title(load)
    plt.show()

fourier_wave_loads = dict.fromkeys(loads, 0)
wave_load_frequencies = np.zeros([terms])
static_wave_load = np.array([])
for wave_load in wave_loads:
    L = t[len(t) - 1]
    y = load_data[wave_load].to_numpy()[::reduction_factor]
    static_wave_load = np.append(arr=static_wave_load, values=np.mean(y))
    y = y - np.mean(y)
    samples = len(t)
    # Calculation of Co-efficients
    a0 = 2. / L * simps(y, t)
    an = lambda n: 2.0 / L * simps(y * np.cos(2 * np.pi * n * t / L), t)
    bn = lambda n: 2.0 / L * simps(y * np.sin(2 * np.pi * n * t / L), t)
    # Sum of the series
    wave_load_fourier_series = np.zeros([terms, len(t)])
    for i in range(terms):
        k = i + 1
        wave_load_fourier_series[i] = an(k) * np.cos(2. * np.pi * k * t / L) + bn(k) * np.sin(2. * np.pi * k * t / L)
        wave_load_frequencies[i] = 2 * np.pi * k / L
    fourier_wave_loads.update({wave_load: wave_load_fourier_series})
    s = wave_load_fourier_series.sum(axis=0)
    correlation = np.corrcoef(y, s)[1, 0]
    # Plotting
    plt.plot(t, s, label="Fourier series")
    plt.plot(t, load_data[wave_load].to_numpy()[::reduction_factor], "--", label="Original load")
    plt.xlabel("$Time[s]$    " + "correlation = " + str(np.corrcoef(y, s)[1, 0]))
    plt.ylabel("$Load[N] / Load[Nm]$")
    plt.legend(loc='best', prop={'size': 10})
    plt.title(wave_load)
    plt.show()

# See input file to find bottom and top node
bottom_node = 2
bottom_node_pos = 6 * (bottom_node - 1)  # positon of x
top_node = 383  # top_node = 383 real top node
top_node_pos = 6 * (top_node - 1)  # position of x
wave_top_node = 76
wave_top_node_pos = 6 * (wave_top_node-1)
wave_bottom_node = 26
wave_bottom_node_pos = 6 * (wave_bottom_node-1)

# import global matrixes generate by abaqus from tower only SSI added later
df_K = pd.read_csv("../monopile_matrixes/tower_matrixes_STIF1.mtx", header=None, delimiter=',')
df_M = pd.read_csv("../monopile_matrixes/tower_matrixes_MASS1.mtx", header=None, delimiter=',')
format = ['node_1', 'dof_node_1', 'node_2', 'dof_node_2', 'value']
df_K.columns = format
df_M.columns = format
dofs = [1, 2, 3, 4, 5, 6]  # x, y, z, rx, ry, rz
dim = df_K['node_1'].nunique() * df_K['dof_node_1'].nunique()
# Matrixes sorted after node number first 6 dofs is for node 1 next 6 for node 2 and so on
K = np.zeros([dim, dim])
for index, row in df_K.iterrows():
    x = (row['node_1'] - 1) * len(dofs) + dofs.index(row['dof_node_1'])
    y = (row['node_2'] - 1) * len(dofs) + dofs.index(row['dof_node_2'])
    K[int(x), int(y)] = row['value']
    K[int(y), int(x)] = row['value']
# Add SSI matrix to simulate soil structure interaction top left indicate bottom of tower
SSI_K = np.genfromtxt('../SSI_stiffness_matrix/ODB_SSI_stiffness_output.txt', delimiter=',')
K[bottom_node_pos:bottom_node_pos + len(dofs), bottom_node_pos:bottom_node_pos + len(dofs)] += SSI_K.copy()
M = np.zeros([dim, dim])
for index, row in df_M.iterrows():
    x = (row['node_1'] - 1) * len(dofs) + dofs.index(row['dof_node_1'])
    y = (row['node_2'] - 1) * len(dofs) + dofs.index(row['dof_node_2'])
    M[int(x), int(y)] = row['value']
    M[int(y), int(x)] = row['value']
# transfer to modal space
eig_values, Phi = LA.eigh(K, M)
w_n = np.sqrt(eig_values)
print("wn")
print(w_n[0:30])
K_star = np.matmul(np.transpose(Phi), np.matmul(K, Phi))
K_star_diag = np.diag(K_star)
# M_star = np.matmul(np.transpose(Phi), np.matmul(M, Phi)) only 1, so drop it
U_star = np.zeros([dim, len(t)])

# load handling
P = np.zeros([dim, len(t)])
xi = np.ones(dim) * modal_damping_coefficient
for i in range(terms):
    print("Term" + str(i))
    w = load_frequencies[i]
    for dof in range(6):  # insert loads to top of tower for x y x rx ry rz
        P[top_node_pos + dof] = fourier_loads[loads[dof]][i]  # fourier series load for frequency w returns 1d array
    for node in range(wave_top_node-wave_bottom_node):
        for dof in range(6):
            P[wave_bottom_node_pos + node * 6 + dof] = fourier_wave_loads[wave_loads[dof]][i]/(wave_top_node-wave_bottom_node)

    P_star = np.matmul(np.transpose(Phi), P)
    # calculate response
    beta = w / w_n
    H = 1 / (K_star_diag * np.sqrt((1 - beta ** 2) ** 2 + (2 * xi * beta) ** 2))
    U_star += np.transpose(np.transpose(P_star) * H)

# transfer modal response to physical response
U = np.matmul(Phi, U_star)

# set up tables to store avg peak for each dof for each mode
U_mode_sum = np.zeros([dim, len(t)])
header = []
for i in range(1, int(dim / len(dofs)) + 1):
    header.append('x' + str(i))
    header.append('y' + str(i))
    header.append('z' + str(i))
    header.append('rx' + str(i))
    header.append('ry' + str(i))
    header.append('rz' + str(i))
U_mode_avg_peak = pd.DataFrame(columns=header, data=np.zeros([n_first_modes, dim]))
U_mode_max_peak = U_mode_avg_peak.copy()
pot_energy_each_mode = np.empty([n_first_modes])
# calculate response in physical space for each mode and find energy for each mode
for i in range(n_first_modes):  # for each mode
    print("mode " + str(i))
    U_mode_i = np.matmul(Phi[:, [i]], U_star[i:i + 1, :])  # mode i contribution to displacement
    peaks_U_star, _ = sig.find_peaks(U_star[i])  # find peaks in modal space for mode i
    U_star_avg_peak = np.average(U_star[i][peaks_U_star])
    pot_energy_each_mode[i] = 1 / 2 * U_star_avg_peak ** 2 * K_star[i][i]
    for dofNumb in range(dim):  # find average peak for each dof
        # peaks_U_mode_i, _ = sig.find_peaks(U_mode_i[dofNumb])
        U_mode_avg_peak.iloc[i, dofNumb] = np.average(U_mode_i[dofNumb][peaks_U_star])  # [peaks_U_mode_i])
        U_mode_max_peak.iloc[i, dofNumb] = np.amax(U_mode_i[dofNumb][peaks_U_star])  # [peaks_U_mode_i])
    U_mode_sum += U_mode_i

    for dof in range(6):  # plot for rz ry rx z y x at top of tower and at bottom
        if np.amax(U_mode_avg_peak.iloc[i, bottom_node_pos + dof].max() < 1e-4):
            continue
        plt.figure()
        plt.plot(t, U_mode_i[top_node_pos + dof], color='b', label="top of tower dof=" + str(dof + 1))
        plt.title("mode: " + str(i) + ",  dof: " + str(dof + 1) + " Top")
        plt.show()
        plt.figure()
        plt.plot(t, U_mode_i[bottom_node_pos + dof], color='g', label="bottom of tower dof=" + str(dof + 1))
        plt.title("mode: " + str(i) + ",  dof: " + str(dof + 1) + " Bottom")
        plt.show()

diff = U - U_mode_sum  # Should be equal to zero, might be a bit higher as u_mode_sum only include n_first_mode
print("error in calculation" + str(np.amax(diff)))

numOfNodes = int(dim / len(dofs))
U_mode_avg_peak[['x' + str(bottom_node), 'y' + str(bottom_node), 'z' + str(bottom_node), 'rx' + str(bottom_node),
                 'ry' + str(bottom_node), 'rz' + str(bottom_node)]].to_csv(path_or_buf="U_mode_avg_peak.csv",
                                                                           header=None,
                                                                           index=False)
U_mode_max_peak[['x' + str(bottom_node), 'y' + str(bottom_node), 'z' + str(bottom_node), 'rx' + str(bottom_node),
                 'ry' + str(bottom_node), 'rz' + str(bottom_node)]].to_csv(path_or_buf="U_mode_max_peak.csv",
                                                                           header=None,
                                                                           index=False)
np.savetxt('pot_energy_each_mode.csv', pot_energy_each_mode, delimiter=',')

# Handle static loading
P_stat = np.zeros([dim, 1])
for dof in range(6):  # insert loads to top of tower for x y z rx ry rz
    P_stat[top_node_pos + dof] = static_load[dof]  # average load for each loadseriesEEEEE
static_U = np.matmul(np.linalg.inv(K), P_stat)
static_U_DF = pd.DataFrame()
static_U_top_DF = pd.DataFrame()
for dof in range(6):
    # static_U_DF['TopPos dof' +str(dof)] = static_U[top_node_pos+dof]
    static_U_DF['BottomPos dof' + str(dof)] = static_U[bottom_node_pos + dof]
    static_U_top_DF['TopPos dof' + str(dof)] = static_U[top_node_pos + dof]
static_U_DF.to_csv(path_or_buf="U_static_response.csv", header=None, index=False)
static_U_top_DF.to_csv(path_or_buf="U_static_response_top.csv", header=None, index=False)
np.save("U_all", U)
np.save("U_star_all", U_star)
np.save("U_static", static_U)
# need to use FAST static as our modal analysis don't take gravity into account, which have effect on static
# due to uneven mass distribution
U_static_FAST = np.array(
    [np.mean(load_data['M1N1TDxss']), np.mean(load_data['M1N1TDyss']), np.mean(load_data['M1N1TDzss']),
     np.mean(load_data['M1N1RDxe']), np.mean(load_data['M1N1RDye']), np.mean(load_data['M1N1RDze'])])
np.savetxt('U_static_FAST.txt', U_static_FAST, delimiter=' ')

# Method 2. The relationship between foundation dating and total structure damping
# Kaynia damping = damping from method 2
kaynia_damping_from_foundation = np.zeros([n_first_modes])
foundation_damping_estimate = 1
SSI_K = np.genfromtxt('../SSI_stiffness_matrix/ODB_SSI_stiffness_output.txt', delimiter=',')
K_st = K.copy()
K_st[bottom_node_pos:bottom_node_pos + len(dofs), bottom_node_pos:bottom_node_pos + len(dofs)] -= SSI_K  # subtract SSI
K_f = SSI_K  # K for foundation
# Calculate relationship for each mode
for i in range(n_first_modes):
    phi_st = Phi[:, i]
    phi_f = Phi[bottom_node_pos:bottom_node_pos + len(dofs), i]
    phi_all = Phi[:, i]
    kaynia_damping_from_foundation[i] = (0.5 * np.matmul(np.transpose(phi_st), np.matmul(K_st, phi_st)) * 0
                                         + 0.5 * np.matmul(np.transpose(phi_f),
                                                           np.matmul(K_f, phi_f)) * foundation_damping_estimate) / (
                                                0.5 * np.matmul(np.transpose(Phi[:, i]), np.matmul(K, Phi[:, i])))

for i in range(15):
    print("mode: "+str(i+1))
    print(Phi[top_node_pos:top_node_pos + 6, i])
a = 0
```

C.2 Modal analysis for jacket structure

The modal analysis for the jacket structure. Method 2 is referred to as Kaynia damping in the script.

```
import pandas as pd
import numpy as np
from scipy import linalg as LA
from scipy import signal as sig
from matplotlib import pyplot as plt
from scipy.integrate import simps

n_first_modes = 50  # number of modes to use in signal processing
terms = 2700  # terms in load fourier series
modal_damping_coefficient = 0.01
reduction_factor = 1  # extract every reduction_factor th element

# Load load series and convert to fourier series with zero mean
file_name = "10MW_Jacket_Base_12.out"
load_data = pd.read_csv(file_name, skiprows=[1, 2, 3, 4, 5, 7], sep="\t")
loads = ["RtAeroFxh", "RtAeroFyh", "RtAeroFzh", "RtAeroMxh", "RtAeroMyh", "RtAeroMzh"]
fourier_loads = dict.fromkeys(loads, 0)
t = load_data["Time"].to_numpy()[::reduction_factor]
load_frequencies = np.zeros([terms])
static_load = np.array([])
for load in loads:
    L = t[len(t) - 1]
    y = load_data[load].to_numpy()[::reduction_factor]
    static_load = np.append(arr=static_load, values=np.mean(y))
    y = y - np.mean(y)
    samples = len(t)
    # Calculation of Co-efficients
    a0 = 2. / L * simps(y, t)
    an = lambda n: 2.0 / L * simps(y * np.cos(2 * np.pi * n * t / L), t)
    bn = lambda n: 2.0 / L * simps(y * np.sin(2 * np.pi * n * t / L), t)
    # Sum of the series
    load_fourier_series = np.zeros([terms, len(t)])
    for i in range(terms):
        k = i + 1
        load_fourier_series[i] = an(k) * np.cos(2. * np.pi * k * t / L) + bn(k) * np.sin(2. * np.pi * k * t / L)
        load_frequencies[i] = 2 * np.pi * k / L
    fourier_loads.update({load: load_fourier_series})
    s = load_fourier_series.sum(axis=0)
    correlation = np.corrcoef(y, s)[1, 0]
    # Plotting
    plt.plot(t, s, label="Fourier series")
    plt.plot(t, load_data[load].to_numpy()[::reduction_factor], "--", label="Original load")
    plt.xlabel("$Time[s]$    " + "correlation = " + str(np.corrcoef(y, s)[1, 0]))
    plt.ylabel("$Load[N] / Load[Nm]$")
    plt.legend(loc='best', prop={'size': 10})
    plt.title(load)
    plt.show()

# See inp file to find bottom and top node
bottom_nodes = np.array([4, 579, 643, 68])  # In order leg1 leg2 leg3 leg4
bottom_node_positions = 6 * (bottom_nodes - 1)  # positon of x
top_node = 349
top_node_pos = 6 * (top_node - 1)  # position of x
# import global matrixes generate by abaqus from tower only SSI added later
df_K = pd.read_csv("Generate_matrices_STIF2.mtx", header=None, delimiter=',')
df_M = pd.read_csv("Generate_matrices_MASS2.mtx", header=None, delimiter=',')
format = ['node_1', 'dof_node_1', 'node_2', 'dof_node_2', 'value']
df_K.columns = format
df_M.columns = format
dofs = [1, 2, 3, 4, 5, 6]  # x, y, z, rx, ry, rz
dim = df_K['node_1'][len(df_K['node_1']) - 1] * df_K['dof_node_1'].nunique()
# Matrixes sorted after node number first 6 dofs is for node 1 next 6 for node 2 and so on
K = np.zeros([dim, dim])
for index, row in df_K.iterrows():
```

```python
    x = (row['node_1'] - 1) * len(dofs) + dofs.index(row['dof_node_1'])
    y = (row['node_2'] - 1) * len(dofs) + dofs.index(row['dof_node_2'])
    K[int(x), int(y)] = row['value']
    K[int(y), int(x)] = row['value']
# Add SSI matrix to simulate soil structure interaction top left indicate bottom of tower
SSI_K = np.genfromtxt('ODB_SSI_stiffness_output.txt', delimiter=',')
for bottom_node_pos in bottom_node_positions:
    K[bottom_node_pos:bottom_node_pos + len(dofs), bottom_node_pos:bottom_node_pos + len(dofs)] += SSI_K.copy()
M = np.zeros([dim, dim])
for index, row in df_M.iterrows():
    x = (row['node_1'] - 1) * len(dofs) + dofs.index(row['dof_node_1'])
    y = (row['node_2'] - 1) * len(dofs) + dofs.index(row['dof_node_2'])
    M[int(x), int(y)] = row['value']
    M[int(y), int(x)] = row['value']

# transfer to modal space
eig_values, Phi = LA.eigh(K, M)
w_n = np.sqrt(eig_values)
print("wn")
print(w_n[0:30])
K_star = np.matmul(np.transpose(Phi), np.matmul(K, Phi))
K_star_diag = np.diag(K_star)
# M_star = np.matmul(np.transpose(Phi), np.matmul(M, Phi)) only 1, so drop it
U_star = np.zeros([dim, len(t)])

# load handling
P = np.zeros([dim, len(t)])
xi = np.ones(dim) * modal_damping_coefficient
for i in range(terms):
    print("Term" + str(i))
    w = load_frequencies[i]
    for dof in range(6):  # insert loads to top of tower for x y z rx ry rz
        P[top_node_pos + dof] = fourier_loads[loads[dof]][i]  # fourier series load for frequency w returns 1d array
    P_star = np.matmul(np.transpose(Phi), P)
    # calculate response
    beta = w / w_n
    H = 1 / (K_star_diag * np.sqrt((1 - beta ** 2) ** 2 + (2 * xi * beta) ** 2))
    U_star += np.transpose(np.transpose(P_star) * H)

# transfer modal response to physical response
U = np.matmul(Phi, U_star)

# set up tables to store avg peak for each dof for each mode
U_mode_sum = np.zeros([dim, len(t)])
header = []
for i in range(1, int(dim / len(dofs)) + 1):
    header.append('x' + str(i))
    header.append('y' + str(i))
    header.append('z' + str(i))
    header.append('rx' + str(i))
    header.append('ry' + str(i))
    header.append('rz' + str(i))
U_mode_avg_peak = pd.DataFrame(columns=header, data=np.zeros([n_first_modes, dim]))
U_mode_max_peak = U_mode_avg_peak.copy()
pot_energy_each_mode = np.empty([n_first_modes])
# calculate response in physical space for each mode and find energy for each mode
for i in range(n_first_modes):  # for each mode
    print("mode " + str(i))
    U_mode_i = np.matmul(Phi[:, [i]], U_star[i:i + 1, :])  # mode i contribution to displacement
    peaks_U_star, _ = sig.find_peaks(U_star[i])  # find peaks in modal space for mode i
    U_star_avg_peak = np.average(U_star[i][peaks_U_star])
    pot_energy_each_mode[i] = 1 / 2 * U_star_avg_peak ** 2 * K_star[i][i]
    for dofNumb in range(dim):  # find average peak for each dof
        # peaks_U_mode_i, _ = sig.find_peaks(U_mode_i[dofNumb])
        U_mode_avg_peak.iloc[i, dofNumb] = np.average(U_mode_i[dofNumb][peaks_U_star])  # [peaks_U_mode_i])
        U_mode_max_peak.iloc[i, dofNumb] = np.amax(U_mode_i[dofNumb][peaks_U_star])  # [peaks_U_mode_i])
    U_mode_sum += U_mode_i

    for dof in range(6):  # plot for rz ry rx z y x at top of tower and at bottom
        if np.amax(U_mode_avg_peak.iloc[i, (bottom_node_positions + dof)].max() < 1e-4):
            continue
        plt.figure()
        plt.plot(t, U_mode_i[top_node_pos + dof], color='b', label="top of tower dof=" + str(dof + 1))
        plt.title("mode: " + str(i) + ",  dof: " + str(dof + 1) + " Top")
        plt.show()
        for bottom_node_pos in bottom_node_positions:
            plt.figure()
            plt.plot(t, U_mode_i[bottom_node_pos + dof], color='g', label="bottom of tower dof=" + str(dof + 1))
            plt.title("mode: " + str(i) + ",  dof: " + str(dof + 1) + " Bottom")
            plt.show()

diff = U - U_mode_sum  # Should be equal to zero, might be a bit higher as u_mode_sum only include n_first_mode
```

```
print("error in calculation" + str(np.amax(diff)))

numOfNodes = int(dim / len(dofs))
for i in range(4):
    bottom_node = bottom_nodes[i]
    U_mode_avg_peak[['x' + str(bottom_node), 'y' + str(bottom_node), 'z' + str(bottom_node), 'rx' + str(bottom_node),
                     'ry' + str(bottom_node), 'rz' + str(bottom_node)]].to_csv(
        path_or_buf="U_mode_avg_peak_leg_" + str(i + 1) + ".csv",
        header=None,
        index=False)
    U_mode_max_peak[['x' + str(bottom_node), 'y' + str(bottom_node), 'z' + str(bottom_node), 'rx' + str(bottom_node),
                     'ry' + str(bottom_node), 'rz' + str(bottom_node)]].to_csv(
        path_or_buf="U_mode_max_peak" + str(i + 1) + ".csv",
        header=None,
        index=False)
np.savetxt('pot_energy_each_mode.csv', pot_energy_each_mode, delimiter=',')

# Handle static loading
P_stat = np.zeros([dim, 1])
for dof in range(6):  # insert loads to top of tower for x y z rx ry rz
    P_stat[top_node_pos + dof] = static_load[dof]  # average load for each loadseriesEEEEE
static_U = np.matmul(np.linalg.inv(K), P_stat)
static_U_DF = pd.DataFrame()
static_U_top_DF = pd.DataFrame()
for i in range(4):
    for dof in range(6):
        # static_U_DF['TopPos dof' +str(dof)] = static_U[top_node_pos+dof]
        static_U_DF['BottomPos leg' + str(i + 1) + 'dof' + str(dof)] = static_U[bottom_node_positions[i] + dof]
        static_U_top_DF['TopPos dof' + str(dof)] = static_U[top_node_pos + dof]
    static_U_DF.to_csv(path_or_buf="U_static_response_leg_" + str(i + 1) + ".csv", header=None, index=False)
    static_U_DF = pd.DataFrame()

static_U_top_DF.to_csv(path_or_buf="U_static_response_top.csv", header=None, index=False)
np.save("U_all", U)
np.save("U_star_all", U_star)
np.save("U_static", static_U)
# need to use FAST static as our modal analysis don't take gravity into account, which have major effect on static
# due to uneven mass distribution
mapping_Abaqus_leg_to_FAST = np.array([[1, 3], [2, 4], [3, 2], [4, 1]])
for i in range(4):
    FAST_leg = mapping_Abaqus_leg_to_FAST[i][1]
    U_static_FAST = np.array(
        [np.mean(load_data['M' + str(FAST_leg) + 'N1TDxss']),
         np.mean(load_data['M' + str(FAST_leg) + 'N1TDyss']),
         np.mean(load_data['M' + str(FAST_leg) + 'N1TDzss']),
         np.mean(load_data['M' + str(FAST_leg) + 'N1RDxe']),
         np.mean(load_data['M' + str(FAST_leg) + 'N1RDye']),
         np.mean(load_data['M' + str(FAST_leg) + 'N1RDze'])])
    print("ab= "+str(i+1)+"      FAST= " + str(FAST_leg))
    np.savetxt('U_static_FAST_leg_' + str(i + 1) + '.txt', U_static_FAST, delimiter=' ')

# Method 2 damping estimation
# Kaynia damping = damping from method 2
kaynia_damping_from_foundation = np.zeros([n_first_modes])
foundation_damping_estimate = 1
SSI_K = np.genfromtxt('ODB_SSI_stiffness_output.txt', delimiter=',')
K_st = K.copy()
for bottom_node_pos in bottom_node_positions:
    K_st[bottom_node_pos:bottom_node_pos + len(dofs), bottom_node_pos:bottom_node_pos + len(dofs)] -= (SSI_K)
for i in range(n_first_modes):
    phi_st = Phi[:, i]
    foundation_E_loss = 0
    for bottom_node_pos in bottom_node_positions:
        foundation_E_loss += 0.5 * np.matmul(np.transpose(Phi[bottom_node_pos:bottom_node_pos + len(dofs), i]),
                                             np.matmul(SSI_K, Phi[bottom_node_pos:bottom_node_pos + len(dofs), i]))
    kaynia_damping_from_foundation[i] = (0.5 * np.matmul(np.transpose(phi_st), np.matmul(K_st, phi_st)) * 0
                                          + foundation_E_loss * foundation_damping_estimate) / (
                                                 0.5 * np.matmul(np.transpose(Phi[:, i]), np.matmul(K, Phi[:, i])))
```

D. OpenFAST jacket template

The following appendices are the OpenFAST 3.0.0 input files for the fatigue analysis template. The files are designed for the 10MW turbine with jacket substructure. Matlab script in Appendix E modifies the files for various wind velocities, damping values and substructures.

D.1 Main input file

```
------- OpenFAST EXAMPLE INPUT FILE -------------------------------------------
10MW reference turbine on reference jacket
---------------------- SIMULATION CONTROL --------------------------------------
FALSE       Echo           - Echo input data to <RootName>.ech (flag)
"FATAL"     AbortLevel     - Error level when simulation should abort (string) {"WARNING", "SEVERE", "
    FATAL"}
       660  TMax           - Total run time (s)
     0.001  DT             - Recommended module time step (s)
         2  InterpOrder    - Interpolation order for input/output time history (-) {1=linear, 2=
             quadratic}
         0  NumCrctn       - Number of correction iterations (-) {0=explicit calculation, i.e., no
             corrections}
     99999  DT_UJac        - Time between calls to get Jacobians (s)
     1E+06  UJacSclFact    Scaling factor used in Jacobians (-)
---------------------- FEATURE SWITCHES AND FLAGS ------------------------------
         1  CompElast      - Compute structural dynamics (switch) {1=ElastoDyn; 2=ElastoDyn + BeamDyn
             for blades}
         1 CompInflow     - Compute inflow wind velocities (switch) {0=still air; 1=InflowWind; 2=
             external from OpenFOAM}
         2  CompAero       - Compute aerodynamic loads (switch) {0=None; 1=AeroDyn v14; 2=AeroDyn v15}
         1  CompServo      - Compute control and electrical-drive dynamics (switch) {0=None; 1=
             ServoDyn}
         1  CompHydro      - Compute hydrodynamic loads (switch) {0=None; 1=HydroDyn}
         1  CompSub        - Compute sub-structural dynamics (switch) {0=None; 1=SubDyn; 2=External
             Platform MCKF}
         0  CompMooring    - Compute mooring system (switch) {0=None; 1=MAP++; 2=FEAMooring; 3=MoorDyn
             ; 4=OrcaFlex}
         0  CompIce        - Compute ice loads (switch) {0=None; 1=IceFlow; 2=IceDyn}
---------------------- INPUT FILES ---------------------------------------------
".\ElastoDyn\10MW_ElastoDyn.dat" EDFile        - Name of file containing ElastoDyn input parameters (
    quoted string)
"unused"   BDBldFile(1)   - Name of file containing BeamDyn input parameters for blade 1 (quoted
    string)
"unused"   BDBldFile(2)   - Name of file containing BeamDyn input parameters for blade 2 (quoted
    string)
"unused"   BDBldFile(3)   - Name of file containing BeamDyn input parameters for blade 3 (quoted
    string)
".\InflowWind\10MW_InflowFile.dat" InflowFile    - Name of file containing inflow wind input
    parameters (quoted string)
".\AeroDyn\10MW_AeroDyn15.dat" AeroFile      - Name of file containing aerodynamic input parameters (
    quoted string)
".\ServoDyn\10MW_ServoDyn_temp.dat" ServoFile     - Name of file containing control and electrical-
    drive input parameters (quoted string)
".\HydroDyn\Jacket_HydroDyn-Kopi.dat" HydroFile     - Name of file containing hydrodynamic input
    parameters (quoted string)
".\SubDyn\Jacket_SubDyn-Kopi.dat" SubFile       - Name of file containing sub-structural input
    parameters (quoted string)
"unused"    MooringFile    - Name of file containing mooring system input parameters (quoted string)
"unused"    IceFile        - Name of file containing ice input parameters (quoted string)
---------------------- OUTPUT --------------------------------------------------
False       SumPrint       - Print summary data to "<RootName>.sum" (flag)
```

```
        30   SttsTime       - Amount of time between screen status messages (s)
      9999   ChkptTime      - Amount of time between creating checkpoint files for potential restart (s
          )
     0.005   DT_Out         - Time step for tabular output (s) (or "default")
         0   TStart         - Time to begin tabular output (s)
         1   OutFileFmt     - Format for tabular (time-marching) output file (switch) {0: uncompressed
              binary [<RootName>.outb], 1: text file [<RootName>.out], 2: binary file [<RootName>.outb
              ], 3: both 1 and 2}
True         TabDelim       - Use tab delimiters in text tabular output file? (flag) {uses spaces if
    false}
"ES10.3E2"   OutFmt         - Format used for text tabular output, excluding the time channel.
    Resulting field should be 10 characters. (quoted string)
---------------------- LINEARIZATION -------------------------------------------
False        Linearize      - Linearization analysis (flag)
False        CalcSteady     - Calculate a steady-state periodic operating point before linearization? [
    unused if Linearize=False] (flag)
         3   TrimCase       - Controller parameter to be trimmed {1:yaw; 2:torque; 3:pitch} [used only
              if CalcSteady=True] (-)
     0.0001 TrimTol         - Tolerance for the rotational speed convergence [used only if CalcSteady=
          True] (-)
      0.001 TrimGain        - Proportional gain for the rotational speed error (>0) [used only if
           CalcSteady=True] (rad/(rad/s) for yaw or pitch; Nm/(rad/s) for torque)
         0   Twr_Kdmp       - Damping factor for the tower [used only if CalcSteady=True] (N/(m/s))
         0   Bld_Kdmp       - Damping factor for the blades [used only if CalcSteady=True] (N/(m/s))
         1   NLinTimes      - Number of times to linearize (-) [>=1] [unused if Linearize=False]
        60   LinTimes       - List of times at which to linearize (s) [1 to NLinTimes] [used only when
             Linearize=True and CalcSteady=False]
         1   LinInputs      - Inputs included in linearization (switch) {0=none; 1=standard; 2=all
              module inputs (debug)} [unused if Linearize=False]
         1   LinOutputs     - Outputs included in linearization (switch) {0=none; 1=from OutList(s); 2=
              all module outputs (debug)} [unused if Linearize=False]
False        LinOutJac      - Include full Jacobians in linearization output (for debug) (flag) [unused
     if Linearize=False; used only if LinInputs=LinOutputs=2]
False        LinOutMod      - Write module-level linearization output files in addition to output for
    full system? (flag) [unused if Linearize=False]
---------------------- VISUALIZATION ------------------------------------------
         0   WrVTK          - VTK visualization data output: (switch) {0=none; 1=initialization data
              only; 2=animation; 3=mode shapes}
         2   VTK_type       - Type of VTK visualization data: (switch) {1=surfaces; 2=basic meshes (
              lines/points); 3=all meshes (debug)} [unused if WrVTK=0]
true         VTK_fields     - Write mesh fields to VTK data files? (flag) {true/false} [unused if WrVTK
    =0]
         5   VTK_fps        - Frame rate for VTK output (frames per second){will use closest integer
              multiple of DT} [used only if WrVTK=2 or WrVTK=3]
```

D.2 ElstoDyn

D.2.1 ElastoDyn input file

```
------- ELASTODYN v1.03.* INPUT FILE -------------------------------------------
Generated with AeroElasticSE FAST driver
---------------------- SIMULATION CONTROL --------------------------------------
False                Echo       - Echo input data to "<RootName>.ech" (flag)
3                    Method     - Integration method: {1: RK4, 2: AB4, or 3: ABM4} (-)
default              DT         - Integration time step (s)
---------------------- ENVIRONMENTAL CONDITION ---------------------------------
9.80665              Gravity    - Gravitational acceleration (m/s^2)
---------------------- DEGREES OF FREEDOM --------------------------------------
True                 FlapDOF1   - First flapwise blade mode DOF (flag)
True                 FlapDOF2   - Second flapwise blade mode DOF (flag)
True                 EdgeDOF    - First edgewise blade mode DOF (flag)
False                TeetDOF    - Rotor-teeter DOF (flag) [unused for 3 blades]
True                 DrTrDOF    - Drivetrain rotational-flexibility DOF (flag)
True                 GenDOF     - Generator DOF (flag)
True                 YawDOF     - Yaw DOF (flag)
True                 TwFADOF1   - First fore-aft tower bending-mode DOF (flag)
True                 TwFADOF2   - Second fore-aft tower bending-mode DOF (flag)
True                 TwSSDOF1   - First side-to-side tower bending-mode DOF (flag)
True                 TwSSDOF2   - Second side-to-side tower bending-mode DOF (flag)
True                 PtfmSgDOF  - Platform horizontal surge translation DOF (flag)
True                 PtfmSwDOF  - Platform horizontal sway translation DOF (flag)
True                 PtfmHvDOF  - Platform vertical heave translation DOF (flag)
True                 PtfmRDOF   - Platform roll tilt rotation DOF (flag)
True                 PtfmPDOF   - Platform pitch tilt rotation DOF (flag)
True                 PtfmYDOF   - Platform yaw rotation DOF (flag)
---------------------- INITIAL CONDITIONS -------------------------------------
0.0                  OoPDefl    - Initial out-of-plane blade-tip displacement (meters)
0.0                  IPDefl     - Initial in-plane blade-tip deflection (meters)
0.0                  BlPitch(1) - Blade 1 initial pitch (degrees)
0.0                  BlPitch(2) - Blade 2 initial pitch (degrees)
0.0                  BlPitch(3) - Blade 3 initial pitch (degrees) [unused for 2 blades]
0.0                  TeetDefl   - Initial or fixed teeter angle (degrees) [unused for 3 blades]
0.0                  Azimuth    - Initial azimuth angle for blade 1 (degrees)
6.0                  RotSpeed   - Initial or fixed rotor speed (rpm)
0.0                  NacYaw     - Initial or fixed nacelle-yaw angle (degrees)
0.0                  TTDspFA    - Initial fore-aft tower-top displacement (meters)
0.0                  TTDspSS    - Initial side-to-side tower-top displacement (meters)
0.0                  PtfmSurge  - Initial or fixed horizontal surge translational displacement of
    platform (meters)
0.0                  PtfmSway   - Initial or fixed horizontal sway translational displacement of
    platform (meters)
-0.0156              PtfmHeave  - Initial or fixed vertical heave translational displacement of
    platform (meters)
0.0                  PtfmRoll   - Initial or fixed roll tilt rotational displacement of platform (
    degrees)
0.0                  PtfmPitch  - Initial or fixed pitch tilt rotational displacement of platform (
    degrees)
0.0                  PtfmYaw    - Initial or fixed yaw rotational displacement of platform (degrees)
---------------------- TURBINE CONFIGURATION -----------------------------------
3                    NumBl      - Number of blades (-)
99.055               TipRad     - The distance from the rotor apex to the blade tip (meters)
2.3                  HubRad     - The distance from the rotor apex to the blade root (meters)
-4.0                 PreCone(1) - Blade 1 cone angle (degrees)
-4.0                 PreCone(2) - Blade 2 cone angle (degrees)
-4.0                 PreCone(3) - Blade 3 cone angle (degrees) [unused for 2 blades]
0.349                HubCM      - Distance from rotor apex to hub mass [positive downwind] (meters)
0.0                  UndSling   - Undersling length [distance from teeter pin to the rotor apex] (
    meters) [unused for 3 blades]
0.0                  Delta3     - Delta-3 angle for teetering rotors (degrees) [unused for 3 blades]
0.0                  AzimB1Up   - Azimuth value to use for I/O when blade 1 points up (degrees)
-10.039              OverHang   - Distance from yaw axis to rotor apex [3 blades] or teeter pin [2
    blades] (meters)
3.55                 ShftGagL   - Distance from rotor apex [3 blades] or teeter pin [2 blades] to
    shaft strain gages [positive for upwind rotors] (meters)
-6.0                 ShftTilt   - Rotor shaft tilt angle (degrees)
-4.84                NacCMxn    - Downwind distance from the tower-top to the nacelle CM (meters)
0.0                  NacCMyn    - Lateral distance from the tower-top to the nacelle CM (meters)
3.39                 NacCMzn    - Vertical distance from the tower-top to the nacelle CM (meters)
-3.09528             NcIMUxn    - Downwind distance from the tower-top to the nacelle IMU (meters)
```

```
0.0                 NcIMUyn    - Lateral distance from the tower-top to the nacelle IMU (meters)
2.23336             NcIMUzn    - Vertical distance from the tower-top to the nacelle IMU (meters)
2.96                Twr2Shft   - Vertical distance from the tower-top to the rotor shaft (meters)
131.63              TowerHt    - Height of tower above ground level [onshore] or MSL [offshore] (
    meters)
26.0                TowerBsHt  - Height of tower base above ground level [onshore] or MSL [offshore]
    (meters)
0.0                 PtfmCMxt   - Downwind distance from the ground level [onshore] or MSL [offshore]
    to the platform CM (meters)
0.0                 PtfmCMyt   - Lateral distance from the ground level [onshore] or MSL [offshore]
    to the platform CM (meters)
26.0                PtfmCMzt   - Vertical distance from the ground level [onshore] or MSL [offshore]
    to the platform CM (meters)
26.0                PtfmRefzt  - Vertical distance from the ground level [onshore] or MSL [offshore]
    to the platform reference point (meters)
---------------------- MASS AND INERTIA ----------------------------------------
0.0                 TipMass(1) - Tip-brake mass, blade 1 (kg)
0.0                 TipMass(2) - Tip-brake mass, blade 2 (kg)
0.0                 TipMass(3) - Tip-brake mass, blade 3 (kg) [unused for 2 blades]
81707.0             HubMass    - Hub mass (kg)
476512.0            HubIner    - Hub inertia about rotor axis [3 blades] or teeter axis [2 blades] (
    kg m^2)
3801700.0           GenIner    - Generator inertia about HSS (kg m^2)
545623.0            NacMass    - Nacelle mass (kg)
19006380.6          NacYIner   - Nacelle inertia about yaw axis (kg m^2)
93457.0             YawBrMass  - Yaw bearing mass (kg)
0.0                 PtfmMass   - Platform mass (kg)
0.0                 PtfmRIner  - Platform inertia for roll tilt rotation about the platform CM (kg m
    ^2)
0.0                 PtfmPIner  - Platform inertia for pitch tilt rotation about the platform CM (kg m
    ^2)
4.0513389E+07       PtfmYIner  - Platform inertia for yaw rotation about the platform CM (kg m^2)
---------------------- BLADE ---------------------------------------------------
50                  BldNodes   - Number of blade nodes (per blade) used for analysis (-)
"10MW_ElastoDyn_blade.dat" BldFile1 - Name of file containing properties for blade 1 (quoted string)
"10MW_ElastoDyn_blade.dat" BldFile2 - Name of file containing properties for blade 2 (quoted string)
"10MW_ElastoDyn_blade.dat" BldFile3 - Name of file containing properties for blade 3 (quoted string) [
    unused for 2 blades]
---------------------- ROTOR-TEETER --------------------------------------------
0                   TeetMod    - Rotor-teeter spring/damper model {0: none, 1: standard, 2: user-
    defined from routine UserTeet} (switch) [unused for 3 blades]
0.0                 TeetDmpP   - Rotor-teeter damper position (degrees) [used only for 2 blades and
    when TeetMod=1]
0.0                 TeetDmp    - Rotor-teeter damping constant (N-m/(rad/s)) [used only for 2 blades
    and when TeetMod=1]
0.0                 TeetCDmp   - Rotor-teeter rate-independent Coulomb-damping moment (N-m) [used
    only for 2 blades and when TeetMod=1]
0.0                 TeetHStP   - Rotor-teeter hard-stop position (degrees) [used only for 2 blades
    and when TeetMod=1]
0.0                 TeetSStP   - Rotor-teeter soft-stop position (degrees) [used only for 2 blades
    and when TeetMod=1]
0.0                 TeetSSSp   - Rotor-teeter soft-stop linear-spring constant (N-m/rad) [used only
    for 2 blades and when TeetMod=1]
0.0                 TeetHSSp   - Rotor-teeter hard-stop linear-spring constant (N-m/rad) [used only
    for 2 blades and when TeetMod=1]
---------------------- DRIVETRAIN ----------------------------------------------
100.0               GBoxEff    - Gearbox efficiency (%)
1.0                 GBRatio    - Gearbox ratio (-)
2317025000.0        DTTorSpr   - Drivetrain torsional spring (N-m/rad)
9240560.0           DTTorDmp   - Drivetrain torsional damper (N-m/(rad/s))
```

```
---------------------- FURLING -------------------------------------------------
False               Furling    - Read in additional model properties for furling turbine (flag) [must
     currently be FALSE)
"unused"            FurlFile   - Name of file containing furling properties (quoted string) [unused
    when Furling=False]
---------------------- TOWER ---------------------------------------------------
20                  TwrNodes   - Number of tower nodes used for analysis (-)
10MW_ElastoDyn_Tower.dat TwrFile - Name of file containing tower properties (quoted string)
---------------------- OUTPUT --------------------------------------------------
False               SumPrint   - Print summary data to "<RootName>.sum" (flag)
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)
True                TabDelim   - Use tab delimiters in text tabular output file? (flag) (currently
    unused)
"ES10.3E2"          OutFmt     - Format used for text tabular output (except time). Resulting field
    should be 10 characters. (quoted string) (currently unused)
0.0                 TStart     - Time to begin tabular output (s) (currently unused)
1                   DecFact    - Decimation factor for tabular output {1: output every time step} (-)
     (currently unused)
5                   NTwGages   - Number of tower nodes that have strain gages for output [0 to 9] (-)
2, 6, 10, 14, 18                  TwrGagNd   - List of tower nodes that have strain gages [1 to
    TwrNodes] (-) [unused if NTwGages=0]
6                   NBlGages   - Number of blade nodes that have strain gages for output [0 to 9] (-)
6, 11, 16, 21, 26, 30 BldGagNd - List of blade nodes that have strain gages [1 to BldNodes] (-) [
    unused if NBlGages=0]
                 OutList           - The next line(s) contains a list of output parameters. See
                     OutListParameters.xlsx for a listing of available output channels, (-)
"PtfmTDxt"
"PtfmTDyt"
"PtfmTDzt"
"PtfmRDxi"
"PtfmRDyi"
"PtfmRDzi"
"TwrTpTDxi"
"TwrTpTDyi"
"TwrTpTDzi"
"TwrBsFxt, TwrBsFyt, TwrBsFzt, TwrBsMxt, TwrBsMyt, TwrBsMzt"
"TwHt1FLxt, TwHt2FLxt, TwHt3FLxt, TwHt4FLxt, TwHt5FLxt"
"TwHt1FLyt, TwHt2FLyt, TwHt3FLyt, TwHt4FLyt, TwHt5FLyt"
"TwHt1FLzt, TwHt2FLzt, TwHt3FLzt, TwHt4FLzt, TwHt5FLzt"
"TwHt1MLxt, TwHt2MLxt, TwHt3MLxt, TwHt4MLxt, TwHt5MLxt"
"TwHt1MLyt, TwHt2MLyt, TwHt3MLyt, TwHt4MLyt, TwHt5MLyt"
"TwHt1MLzt, TwHt2MLzt, TwHt3MLzt, TwHt4MLzt, TwHt5MLzt"
"TwHt1TDxt, TwHt2TDxt, TwHt3TDxt, TwHt4TDxt, TwHt5TDxt"
"TwHt1TDyt, TwHt2TDyt, TwHt3TDyt, TwHt4TDyt, TwHt5TDyt"
"TwHt1TDzt, TwHt2TDzt, TwHt3TDzt, TwHt4TDzt, TwHt5TDzt"
"TwHt1RDxt, TwHt2RDxt, TwHt3RDxt, TwHt4RDxt, TwHt5RDxt"
"TwHt1RDyt, TwHt2RDyt, TwHt3RDyt, TwHt4RDyt, TwHt5RDyt"
"TwHt1RDzt, TwHt2RDzt, TwHt3RDzt, TwHt4RDzt, TwHt5RDzt"
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

D.2.2 ElastoDyn blade input file

```
------- ELASTODYN V1.00.* INDIVIDUAL BLADE INPUT FILE --------------------------
Generated with AeroElasticSE FAST driver
---------------------- BLADE PARAMETERS ----------------------------------------
30                  NBlInpSt   - Number of blade input stations (-)
0.4                 BldFlDmp1  - Blade flap mode #1 structural damping in percent of critical (%)
0.4                 BldFlDmp2  - Blade flap mode #2 structural damping in percent of critical (%)
```

```
0.3                 BldEdDmp1  - Blade edge mode #1 structural damping in percent of critical (%)
---------------------- BLADE ADJUSTMENT FACTORS --------------------------------
1.0                 FlStTunr1  - Blade flapwise modal stiffness tuner, 1st mode (-)
1.0                 FlStTunr2  - Blade flapwise modal stiffness tuner, 2nd mode (-)
1.0                 AdjBlMs    - Factor to adjust blade mass density (-)
1.0                 AdjFlSt    - Factor to adjust blade flap stiffness (-)
1.0                 AdjEdSt    - Factor to adjust blade edge stiffness (-)
---------------------- DISTRIBUTED BLADE PROPERTIES ----------------------------
   BlFract     PitchAxis     StrcTwst     BMassDen      FlpStff       EdgStff
     (-)          (-)         (deg)         (kg/m)       (Nm^2)        (Nm^2)
 0.000000000000000e+00 5.000000000000000e-01 1.200002806121996e+01 2.371932011770502e+03
     1.199028948725793e+11 1.199064840689547e+11
 3.448051701163937e-02 4.992152755398002e-01 1.199964011912278e+01 1.286670292880901e+03
     6.148299413684251e+10 4.680332593325961e+10
 6.896739751917620e-02 4.672485136741974e-01 1.199671430769378e+01 1.210002060001184e+03
     4.811493915810210e+10 3.949145513777705e+10
 1.034469135766283e-01 4.288897840799227e-01 1.202241256026795e+01 1.090244794430454e+03
     3.697599613119305e+10 3.550538202178642e+10
 1.379339990205685e-01 3.912335423197492e-01 1.156869232147211e+01 9.707264168439144e+02
     2.554482191445094e+10 3.339533472007076e+10
 1.724181855255898e-01 3.606002388416181e-01 1.003883227305816e+01 8.533689005250673e+02
     1.489698805398057e+10 3.154939020265875e+10
 2.068986193664772e-01 3.402398741105638e-01 8.076609005387132e+00 7.589503735636732e+02
     9.852819867259695e+09 2.892568799431439e+10
 2.413775792954272e-01 3.306748768472906e-01 6.583589890889979e+00 6.711033341469282e+02
     7.822494294802562e+09 2.401909319207922e+10
 2.758624208035038e-01 3.302603283023026e-01 5.661206923495265e+00 6.102039059553290e+02
     6.496072288841572e+09 1.991980484738091e+10
 3.103451056153142e-01 3.361433319484836e-01 5.010257468512085e+00 5.575750780467753e+02
     5.356472909835671e+09 1.608948590892418e+10
 3.448321547903672e-01 3.459405469678954e-01 4.447057061811484e+00 5.182519825659979e+02
     4.425385473590683e+09 1.310863522274828e+10
 3.793123394443066e-01 3.588053915031434e-01 3.930526877993271e+00 4.880983929220806e+02
     3.643107012589824e+09 1.076409740822400e+10
 4.137941726258467e-01 3.742787356321839e-01 3.440409048815762e+00 4.588849512518002e+02
     2.945830911876412e+09 8.692055389344172e+09
 4.482818342296974e-01 3.920532253662087e-01 2.937699901176175e+00 4.315563802175020e+02
     2.334313851620291e+09 6.953052354981404e+09
 4.827644125346460e-01 4.102244087380393e-01 2.390684243696569e+00 4.129008730336116e+02
     1.818120326314945e+09 5.446480253092859e+09
 5.172465736612203e-01 4.282815639424382e-01 1.800014974504586e+00 3.863704614264516e+02
     1.371100961374134e+09 4.300160045340353e+09
 5.517259133457584e-01 4.463827495700969e-01 1.181890536278555e+00 3.589306571652718e+02
     9.649576921568537e+08 3.345010216811661e+09
 5.862086652444077e-01 4.629447909024211e-01 5.507512667346930e-01 3.328621291759838e+02
     6.701775805484936e+08 2.604034379874705e+09
 6.206949652273239e-01 4.781259370314843e-01 -7.688715647492510e-02 3.079117529295063e+02
     4.551419670273628e+08 2.024047369607347e+09
 6.551719841142299e-01 4.919919378338999e-01 -6.869854123077352e-01 2.825760758579560e+02
     3.120563202590153e+08 1.559647926259672e+09
 6.896582879742414e-01 5.036629588431591e-01 -1.262800892452079e+00 2.591066718550961e+02
     2.282282406112023e+08 1.199317866360565e+09
 7.241436828138220e-01 5.135660919540230e-01 -1.792359064289039e+00 2.338554327437469e+02
     1.660570163934705e+08 9.184074710487620e+08
 7.586231597836172e-01 5.217514745916515e-01 -2.262491400035471e+00 2.086567449582570e+02
     1.186636830166743e+08 6.997481094482968e+08
 7.931011025897087e-01 5.253728858004458e-01 -2.647842454204785e+00 1.841105691170818e+02
     8.379747737868544e+07 5.274058943422229e+08
 8.275897060231471e-01 5.264592476489028e-01 -2.942933952928573e+00 1.593507162753616e+02
     5.827137276418268e+07 3.910002030405779e+08
 8.620721712768565e-01 5.227474695488077e-01 -3.129337812859181e+00 1.340423816482642e+02
     3.989110925118160e+07 2.821294133288149e+08
 8.965547308306578e-01 5.138115831586969e-01 -3.135739832811044e+00 1.068398520238115e+02
     2.590873895987872e+07 1.926219854489844e+08
 9.310400808627222e-01 4.983891246684350e-01 -2.863327401352850e+00 7.448752035471513e+01
     1.418925809630283e+07 1.127660729273758e+08
 9.655187699072447e-01 4.762865900383142e-01 -2.046041506422669e+00 4.162980623490174e+01
     4.479243006045787e+06 4.438108703913335e+07
 1.000000000000000e+00 4.460000000000000e-01 -3.724225668350351e-02 2.378849807754251e+00
     7.285024349360181e+02 2.464563907062745e+04
---------------------- BLADE MODE SHAPES ---------------------------------------
-0.36659891275548917 BldFl1Sh(2) - Flap mode 1, coeff of x^2
3.9964234627642856   BldFl1Sh(3) -         , coeff of x^3
-9.971511882614983   BldFl1Sh(4) -         , coeff of x^4
12.712153836321718   BldFl1Sh(5) -         , coeff of x^5
-5.370466503715531   BldFl1Sh(6) -         , coeff of x^6
-0.7821582550747335  BldFl2Sh(2) - Flap mode 2, coeff of x^2
7.660260929624139    BldFl2Sh(3) -         , coeff of x^3
-25.870446123900223  BldFl2Sh(4) -         , coeff of x^4
33.60271388191876    BldFl2Sh(5) -         , coeff of x^5
-13.610370432567944  BldFl2Sh(6) -         , coeff of x^6
0.20422563911176705  BldEdgSh(2) - Edge mode 1, coeff of x^2
0.5943215941350376   BldEdgSh(3) -         , coeff of x^3
3.084040947006545    BldEdgSh(4) -         , coeff of x^4
-4.265508602723565   BldEdgSh(5) -         , coeff of x^5
1.3829204224702147   BldEdgSh(6) -         , coeff of x^6
```

D.2.3 ElastoDyn tower input file

```
------- ELASTODYN V1.00.* TOWER INPUT FILE -------------------------------------
Generated with AeroElasticSE FAST driver
---------------------- TOWER PARAMETERS ----------------------------------------
11                    NTwInpSt   - Number of input stations to specify tower geometry
0.8                  TwrFADmp(1) - Tower 1st fore-aft mode structural damping ratio (%)
0.8                  TwrFADmp(2) - Tower 2nd fore-aft mode structural damping ratio (%)
0.8                  TwrSSDmp(1) - Tower 1st side-to-side mode structural damping ratio (%)
0.8                  TwrSSDmp(2) - Tower 2nd side-to-side mode structural damping ratio (%)
---------------------- TOWER ADJUSTMUNT FACTORS --------------------------------
1.0                  FAStTunr(1) - Tower fore-aft modal stiffness tuner, 1st mode (-)
1.0                  FAStTunr(2) - Tower fore-aft modal stiffness tuner, 2nd mode (-)
1.0                  SSStTunr(1) - Tower side-to-side stiffness tuner, 1st mode (-)
1.0                  SSStTunr(2) - Tower side-to-side stiffness tuner, 2nd mode (-)
1.0                  AdjTwMa    - Factor to adjust tower mass density (-)
1.0                  AdjFASt    - Factor to adjust tower fore-aft stiffness (-)
1.0                  AdjSSSt    - Factor to adjust tower side-to-side stiffness (-)
---------------------- DISTRIBUTED TOWER PROPERTIES ----------------------------
  HtFract     TMassDen       TwFAStif      TwSSStif
   (-)         (kg/m)         (Nm^2)        (Nm^2)
  0.000       15383.91          3.2182E+12   3.2182E+12
  0.099       14860.52          2.9008E+12   2.9008E+12
  0.199       13321.73          2.4236E+12   2.4236E+12
  0.298       11856.37          2.0052E+12   2.0052E+12
  0.398       11423.77          1.7936E+12   1.7936E+12
  0.497       10067.90          1.4611E+12   1.4611E+12
  0.597       8785.46       1.1748E+12       1.1748E+12
  0.696       7576.46       9.3018E+11       9.3018E+11
  0.796       5640.45       6.3444E+11       6.3444E+11
  0.895       4614.37       4.7280E+11       4.7280E+11
  1.000       4382.05       4.0493E+11       4.0493E+11
---------------------- TOWER FORE-AFT MODE SHAPES ------------------------------
```

```
    -0.8575  TwFAM1Sh(2) - Mode 1, coefficient of x^2 term
     8.2745  TwFAM1Sh(3) -       , coefficient of x^3 term
   -15.1777  TwFAM1Sh(4) -       , coefficient of x^4 term
    13.0613  TwFAM1Sh(5) -       , coefficient of x^5 term
    -4.3007  TwFAM1Sh(6) -       , coefficient of x^6 term
     1.1996  TwFAM2Sh(2) - Mode 2, coefficient of x^2 term
    -3.8005  TwFAM2Sh(3) -       , coefficient of x^3 term
     8.3850  TwFAM2Sh(4) -       , coefficient of x^4 term
    -7.0279  TwFAM2Sh(5) -       , coefficient of x^5 term
     2.2438  TwFAM2Sh(6) -       , coefficient of x^6 term
---------------------- TOWER SIDE-TO-SIDE MODE SHAPES --------------------------
    -0.8393  TwSSM1Sh(2) - Mode 1, coefficient of x^2 term
     8.1446  TwSSM1Sh(3) -       , coefficient of x^3 term
   -14.9227  TwSSM1Sh(4) -       , coefficient of x^4 term
    12.8376  TwSSM1Sh(5) -       , coefficient of x^5 term
    -4.2202  TwSSM1Sh(6) -       , coefficient of x^6 term
     1.1827  TwSSM2Sh(2) - Mode 2, coefficient of x^2 term
    -4.1511  TwSSM2Sh(3) -       , coefficient of x^3 term
     9.0914  TwSSM2Sh(4) -       , coefficient of x^4 term
    -7.7153  TwSSM2Sh(5) -       , coefficient of x^5 term
     2.5922  TwSSM2Sh(6) -       , coefficient of x^6 term
```

D.3 InfowWind input file

TurbSim input file that generates the .bts files can be found in Appendix G.1.

```
------- InflowWind v3.01.* INPUT FILE
    -------------------------------------------------------------------------
Generated with AeroElasticSE FAST driver
--------------------------------------------------------------------------------------------------------------

False                Echo       - Echo input data to <RootName>.ech (flag)
3                    WindType   - switch for wind file type (1=steady; 2=uniform; 3=binary TurbSim FF;
     4=binary Bladed-style FF; 5=HAWC format; 6=User defined; 7=native Bladed FF)
0                    PropagationDir - Direction of wind propagation (meteoroligical rotation from
    aligned with X (positive rotates towards -Y) -- degrees)
0                    VFlowAng   - Upflow angle (degrees) (not used for native Bladed format WindType
    =7)
1                    NWindVel   - Number of points to output the wind velocity (0 to 9)
0.0                  WindVxiList - List of coordinates in the inertial X direction (m)
0.0                  WindVyiList - List of coordinates in the inertial Y direction (m)
131.63               WindVziList - List of coordinates in the inertial Z direction (m)
================== Parameters for Steady Wind Conditions [used only for WindType = 1]
    =========================
12                   HWindSpeed - Horizontal windspeed                       (m/s)
131.63               RefHt      - Reference height for horizontal wind speed (m)
0.2                  PLexp      - Power law exponent                         (-)
================== Parameters for Uniform wind file [used only for WindType = 2]
    ============================
"none"               Filename_Uni - Filename of time series data for uniform wind field. (-)
119.0                RefHt_Uni  - Reference height for horizontal wind speed         (m)
1.0                  RefLength  - Reference length for linear horizontal and vertical sheer (-)
================== Parameters for Binary TurbSim Full-Field files [used only for WindType = 3]
    ==============
"..\..\TurbSim\TurbSim11.bts"  FileName_BTS - Name of the Full field wind file to use (.bts)
================== Parameters for Binary Bladed-style Full-Field files [used only for WindType = 4]
    =========
"none"               FilenameRoot - Rootname of the full-field wind file to use (.wnd, .sum)
False                TowerFile  - Have tower file (.twr) (flag)
```

```
================== Parameters for HAWC-format binary files [Only used with WindType = 5]
    ======================
"none"              FileName_u - name of the file containing the u-component fluctuating wind (.bin)
"none"              FileName_v - name of the file containing the v-component fluctuating wind (.bin)
"none"              FileName_w - name of the file containing the w-component fluctuating wind (.bin)
2                   nx         - number of grids in the x direction (in the 3 files above) (-)
2                   ny         - number of grids in the y direction (in the 3 files above) (-)
2                   nz         - number of grids in the z direction (in the 3 files above) (-)
10                  dx         - distance (in meters) between points in the x direction (m)
10                  dy         - distance (in meters) between points in the y direction (m)
10                  dz         - distance (in meters) between points in the z direction (m)
0.0                 RefHt_Hawc - reference height; the height (in meters) of the vertical center of
    the grid (m)
------------- Scaling parameters for turbulence
    ---------------------------------------------------------
0                   ScaleMethod - Turbulence scaling method [0 = none, 1 = direct scaling, 2 =
    calculate scaling factor based on a desired standard deviation]
1.0                 SFx        - Turbulence scaling factor for the x direction (-) [ScaleMethod=1]
1.0                 SFy        - Turbulence scaling factor for the y direction (-) [ScaleMethod=1]
1.0                 SFz        - Turbulence scaling factor for the z direction (-) [ScaleMethod=1]
1.0                 SigmaFx    - Turbulence standard deviation to calculate scaling from in x
    direction (m/s) [ScaleMethod=2]
1.0                 SigmaFy    - Turbulence standard deviation to calculate scaling from in y
    direction (m/s) [ScaleMethod=2]
1.0                 SigmaFz    - Turbulence standard deviation to calculate scaling from in z
    direction (m/s) [ScaleMethod=2]
------------- Mean wind profile parameters (added to HAWC-format files)
    ---------------------------------
0.0                 URef       - Mean u-component wind speed at the reference height (m/s)
0                   WindProfile - Wind profile type (0=constant;1=logarithmic,2=power law)
0.0                 PLExp_Hawc - Power law exponent (-) (used for PL wind profile type only)
0.0                 Z0         - Surface roughness length (m) (used for LG wind profile type only)
0                   XOffset    - Initial offset in +x direction (shift of wind box) (-)
====================== OUTPUT ==================================================
False               SumPrint   - Print summary data to <RootName>.IfW.sum (flag)
                                      OutList     - The next line(s) contains a list of output
                                          parameters. See OutListParameters.xlsx for a listing of
                                          available output channels, (-)
"Wind1VelX"          X component of wind at user selected wind point 1 Directed along the xi-axis
"Wind1VelY"          Y component of wind at user selected wind point 1 Directed along the yi-axis
"Wind1VelZ"          Z component of wind at user selected wind point 1 Directed along the zi-axis

END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

D.4 AeroDyn

D.4.1 AeroDyn input file

```
------- AERODYN v15.03.* INPUT FILE ------------------------------------------------
IEA-10.0-198 Reference Wind Turbine aerodynamic input properties
====== General Options ============================================================================
False               Echo       - Echo the input to "<rootname>.AD.ech"? (flag)
default             DTAero     - Time interval for aerodynamic calculations {or "default"} (s)
1                   WakeMod    - Type of wake/induction model (switch) {0=none, 1=BEMT, 2=DBEMT, 3=
    OLAF} [WakeMod cannot be 2 or 3 when linearizing]
2                   AFAeroMod  - Type of blade airfoil aerodynamics model (switch) {1=steady model,
    2=Beddoes-Leishman unsteady model} [AFAeroMod must be 1 when linearizing]
```

```
1                   TwrPotent  - Type tower influence on wind based on potential flow around the
    tower (switch) {0=none, 1=baseline potential flow, 2=potential flow with Bak correction}
0                   TwrShadow  - Calculate tower influence on wind based on downstream tower shadow (
    switch) {0=none, 1=Powles model, 2=Eames model}
True                TwrAero    - Calculate tower aerodynamic loads? (flag)
False               FrozenWake - Assume frozen wake during linearization? (flag) [used only when
    WakeMod=1 and when linearizing]
False               CavitCheck - Perform cavitation check? (flag) [AFAeroMod must be 1 when
    CavitCheck=true]
False               CompAA     - Flag to compute AeroAcoustics calculation [only used when WakeMod=1
    or 2]
AeroAcousticsInput.dat AA_InputFile - AeroAcoustics input file [used only when CompAA=true]
====== Environmental Conditions ===================================================================
1.225                         AirDens    - Air density (kg/m^3)
1.464e-05                     KinVisc    - Kinematic air viscosity (m^2/s)
340.3                         SpdSound   - Speed of sound (m/s)
101325                        Patm       - Atmospheric pressure (Pa) [used only when CavitCheck=True
    ]
1700                          Pvap       - Vapour pressure of fluid (Pa) [used only when CavitCheck=
    True]
0.5                                  FluidDepth - Water depth above mid-hub height (m) [used only
    when CavitCheck=True]
====== Blade-Element/Momentum Theory Options ====================================================== [
    used only when WakeMod=1]
2                   SkewMod    - Type of skewed-wake correction model (switch) {1=uncoupled, 2=Pitt/
    Peters, 3=coupled} [unused when WakeMod=0 or 3]
"default"                  SkewModFactor - Constant used in Pitt/Peters skewed wake model {or "
    default" is 15/32*pi} (-) [used only when SkewMod=2; unused when WakeMod=0 or 3]
True                TipLoss    - Use the Prandtl tip-loss model? (flag) [unused when WakeMod=0 or 3]
True                HubLoss    - Use the Prandtl hub-loss model? (flag) [unused when WakeMod=0 or 3]
True                TanInd     - Include tangential induction in BEMT calculations? (flag) [unused
    when WakeMod=0 or 3]
True                AIDrag     - Include the drag term in the axial-induction calculation? (flag) [
    unused when WakeMod=0 or 3]
True                TIDrag     - Include the drag term in the tangential-induction calculation? (flag
    ) [unused when WakeMod=0,3 or TanInd=FALSE]
"default"           IndToler   - Convergence tolerance for BEMT nonlinear solve residual equation {or
     "default"} (-) [unused when WakeMod=0 or 3]
100                 MaxIter    - Maximum number of iteration steps (-) [unused when WakeMod=0]
====== Dynamic Blade-Element/Momentum Theory Options
    ==================================================== [used only when WakeMod=1]
1                   DBEMT_Mod  - Type of dynamic BEMT (DBEMT) model {1=constant tau1, 2=time-
    dependent tau1} (-) [used only when WakeMod=2]
20                  tau1_const - Time constant for DBEMT (s) [used only when WakeMod=2 and DBEMT_Mod
    =1]
====== OLAF -- cOnvecting LAgrangian Filaments (Free Vortex Wake) Theory Options ================== [
    used only when WakeMod=3]
IEA-10.0-198-RWT_OLAF.dat OLAFInputFileName - Input file for OLAF [used only when WakeMod=3]
====== Beddoes-Leishman Unsteady Airfoil Aerodynamics Options ===================================== [
    used only when AFAeroMod=2]
3                   UAMod      - Unsteady Aero Model Switch (switch) {1=Baseline model (Original), 2=
    Gonzalez's variant (changes in Cn,Cc,Cm), 3=Minnema/Pierce variant (changes in Cc and Cm)} [used
    only when AFAeroMod=2]
True                FLookup    - Flag to indicate whether a lookup for f' will be calculated (TRUE)
    or whether best-fit exponential equations will be used (FALSE); if FALSE S1-S4 must be provided
    in airfoil input files (flag) [used only when AFAeroMod=2]
====== Airfoil Information =========================================================================
1                   AFTabMod   - Interpolation method for multiple airfoil tables {1=1D interpolation
     on AoA (first table only); 2=2D interpolation on AoA and Re; 3=2D interpolation on AoA and
    UserProp} (-)
```

```
1                   InCol_Alfa - The column in the airfoil tables that contains the angle of attack
    (-)
2                   InCol_Cl   - The column in the airfoil tables that contains the lift coefficient
    (-)
3                   InCol_Cd   - The column in the airfoil tables that contains the drag coefficient
    (-)
4                   InCol_Cm   - The column in the airfoil tables that contains the pitching-moment
    coefficient; use zero if there is no Cm column (-)
0                   InCol_Cpmin - The column in the airfoil tables that contains the Cpmin
    coefficient; use zero if there is no Cpmin column (-)
30                  NumAFfiles - Number of airfoil files used (-)
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_00.dat" AFNames       - Airfoil file names (NumAFfiles
    lines) (quoted strings)
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_01.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_02.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_03.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_04.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_05.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_06.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_07.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_08.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_09.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_10.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_11.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_12.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_13.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_14.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_15.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_16.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_17.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_18.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_19.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_20.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_21.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_22.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_23.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_24.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_25.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_26.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_27.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_28.dat"
"Airfoils/IEA-10.0-198-RWT_AeroDyn15_Polar_29.dat"
====== Rotor/Blade Properties =================================================================
True                UseBlCm    - Include aerodynamic pitching moment in calculations? (flag)
"10MW_AeroDyn15_blade.dat" ADBlFile(1) - Name of file containing distributed aerodynamic properties
    for Blade #1 (-)
"10MW_AeroDyn15_blade.dat" ADBlFile(2) - Name of file containing distributed aerodynamic properties
    for Blade #2 (-) [unused if NumBl < 2]
"10MW_AeroDyn15_blade.dat" ADBlFile(3) - Name of file containing distributed aerodynamic properties
    for Blade #3 (-) [unused if NumBl < 3]
====== Tower Influence and Aerodynamics ========================================================
     [used only when TwrPotent/=0, TwrShadow/=0, or TwrAero=True]
11                 NumTwrNds  - Number of tower nodes used in the analysis (-) [used only when
    TwrPotent/=0, TwrShadow/=0, or TwrAero=True]
TwrElev      TwrDiam      TwrCd       TwrTI
(m)            (m)         (-)          (-)
26.00            8.300              0.5               0.1
36.51            8.020              0.5               0.1
47.01            7.740              0.5               0.1
57.52            7.460              0.5               0.1
```

```
68.02           7.190                 0.5                0.1
78.53           6.910                 0.5                0.1
89.03           6.630                 0.5                0.1
99.54           6.350                 0.5                0.1
110.04      6.070             0.5                0.1
120.55      5.790             0.5                0.1
131.63      5.500             0.5                0.1
====== Outputs ====================================================================================
False               SumPrint   - Generate a summary file listing input options and interpolated
    properties to "<rootname>.AD.sum"? (flag)
9                   NBlOuts    - Number of blade node outputs [0 - 9] (-)
4, 7, 10, 13, 15, 18, 21, 24, 27 BlOutNd - Blade nodes whose values will be output (-)
9                   NTwOuts    - Number of tower node outputs [0 - 9] (-)
1, 2, 3, 4, 5, 6, 7, 8, 9                TwOutNd    - Tower nodes whose values will be output (-)
                OutList           - The next line(s) contains a list of output parameters. See
                    OutListParameters.xlsx for a listing of available output channels, (-)
"TwN1Fdx"
"TwN2Fdx"
"TwN3Fdx"
"TwN4Fdx"
"TwN5Fdx"
"TwN6Fdx"
"TwN7Fdx"
"TwN8Fdx"
"TwN9Fdx"
"TwN1Fdy"
"TwN2Fdy"
"TwN3Fdy"
"TwN4Fdy"
"TwN5Fdy"
"TwN6Fdy"
"TwN7Fdy"
"TwN8Fdy"
"TwN9Fdy"
"RtAeroFxh"
"RtAeroFyh"
"RtAeroFzh"
"RtAeroMxh"
"RtAeroMyh"
"RtAeroMzh"
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
-------------------------------------------------------------------------------------
```

D.4.2 AeroDyn blade input file

```
------- AERODYN v15.00.* BLADE DEFINITION INPUT FILE ------------------------------------
Generated with AeroElasticSE FAST driver
====== Blade Properties =================================================================
30         NumBlNds   - Number of blade nodes used in the analysis (-)
   BlSpn       BlCrvAC      BlSwpAC      BlCrvAng     BlTwist      BlChord        BlAFID
    (m)          (m)          (m)          (deg)        (deg)         (m)            (-)
 0.000000000000000e+00 0.000000000000000e+00 0.000000000000000e+00 -2.841341264563080e-01
     1.200002806121996e+01 4.600000000000000e+00 1
 3.336162423461167e+00 -1.654421527553980e-02 0.000000000000000e+00 -3.377268435011394e-01
     1.199964011912278e+01 4.602856719303900e+00 2
 6.672940546967894e+00 -3.933305588585017e-02 0.000000000000000e+00 -4.210207359220099e-01
     1.199671430769378e+01 4.722267657550535e+00 3
 1.000900612310667e+01 -6.557715114405413e-02 0.000000000000000e+00 -4.651652887327193e-01
     1.202241256026795e+01 5.008074089590719e+00 4
```

```
1.334580407523511e+01 -9.350721003134796e-02 0.000000000000000e+00 -4.830551455935536e-01
    1.156869232147211e+01 5.411514106583073e+00 5
1.668232154052844e+01 -1.218385579937304e-01 0.000000000000000e+00 -4.877219150971492e-01
    1.003883227305816e+01 5.800776847290640e+00 6
2.001847591680350e+01 -1.503066639299398e-01 0.000000000000000e+00 -4.938541126732519e-01
    8.076609005387132e+00 6.015708470169677e+00 7
2.335448768472906e+01 -1.793477832512315e-01 0.000000000000000e+00 -5.017469868135497e-01
    6.583589890889979e+00 5.982229064039409e+00 8
2.669106852484301e+01 -2.087387242480996e-01 0.000000000000000e+00 -5.126107923364652e-01
    5.661206923495265e+00 5.827155062245235e+00 9
3.002744069380972e+01 -2.390482031574574e-01 0.000000000000000e+00 -5.391128507636488e-01
    5.010257468512085e+00 5.608710832987120e+00 10
3.336423513674197e+01 -2.715275862068965e-01 0.000000000000000e+00 -5.942422772358332e-01
    4.447057061811484e+00 5.345638585017836e+00 11
3.670036540293389e+01 -3.082550962088016e-01 0.000000000000000e+00 -6.817513298831223e-01
    3.930526877993271e+00 5.053029053152982e+00 12
4.003665517241379e+01 -3.509195402298850e-01 0.000000000000000e+00 -7.966951995003160e-01
    3.440409048815762e+00 4.745833333333334e+00 13
4.337350887089437e+01 -4.010418615230644e-01 0.000000000000000e+00 -9.417872586551786e-01
    2.937699901176175e+00 4.434493039759803e+00 14
4.670987073478967e+01 -4.606041704278749e-01 0.000000000000000e+00 -1.133138363624353e+00
    2.390684243696569e+00 4.125517078895109e+00 15
5.004619223459137e+01 -5.329988867770838e-01 0.000000000000000e+00 -1.385754310382543e+00
    1.800014974504586e+00 3.822383518870486e+00 16
5.338224074576885e+01 -6.219662141370261e-01 0.000000000000000e+00 -1.695430119589656e+00
    1.181890536278555e+00 3.529777853199384e+00 17
5.671861940572266e+01 -7.304127842993395e-01 0.000000000000000e+00 -2.059721193813486e+00
    5.507512667346930e-01 3.252281841526046e+00 18
6.005534136056972e+01 -8.618052848575714e-01 0.000000000000000e+00 -2.497308875779408e+00
    -7.688715647492510e-02 2.991705734632684e+00 19
6.339116532297232e+01 -1.021152044681885e+00 0.000000000000000e+00 -3.005546348335876e+00
    -6.869854123077352e-01 2.748427838756678e+00 20
6.672788765294773e+01 -1.211664404894328e+00 0.000000000000000e+00 -3.591966938519594e+00
    -1.262800892452079e+00 2.523873414905450e+00 21
7.006452203065135e+01 -1.439241794380588e+00 0.000000000000000e+00 -4.299034625217793e+00
    -1.792359064289039e+00 2.317451053639846e+00 22
7.340058382486387e+01 -1.711862556715063e+00 0.000000000000000e+00 -5.171394237073920e+00
    -2.262491400035471e+00 2.128054548548095e+00 23
7.673649718106726e+01 -2.040622669463747e+00 0.000000000000000e+00 -6.228703097646942e+00
    -2.647842454204785e+00 1.957865831912941e+00 24
8.007344200626959e+01 -2.435850156739812e+00 0.000000000000000e+00 -7.439338711743959e+00
    -2.942933952928573e+00 1.802591379310345e+00 25
8.340979293189226e+01 -2.904657231085948e+00 0.000000000000000e+00 -8.994870020906601e+00
    -3.129337812859181e+00 1.660044089895351e+00 26
8.674615298152030e+01 -3.479102038483521e+00 0.000000000000000e+00 -1.103253271356095e+01
    -3.135739832811044e+00 1.521990683939798e+00 27
9.008278302387268e+01 -4.181642891246685e+00 0.000000000000000e+00 -1.374480804418788e+01
    -2.863327401352850e+00 1.342706976127321e+00 28
9.341876858237546e+01 -5.064501302681990e+00 0.000000000000000e+00 -1.766379073748617e+01
    -2.046041506422669e+00 1.050590574712644e+00 29
9.675500000000000e+01 -6.206200000000000e+00 0.000000000000000e+00 -2.001170345264212e+01
    -3.724225668350351e-02 9.619999999999999e-02 30
```

D.4.3 Airfoils

Airfoils from IEA Task 37 10MW reference monopile GitHub repository [78]

D.5 ServoDyn input file

```
------- SERVODYN v1.05.* INPUT FILE --------------------------------------------
Generated with AeroElasticSE FAST driver
---------------------- SIMULATION CONTROL --------------------------------------
False               Echo         - Echo input data to <RootName>.ech (flag)
default             DT           - Communication interval for controllers (s) (or "default")
---------------------- PITCH CONTROL -------------------------------------------
0                   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)
0.0                 TPCOn        - Time to enable active pitch control (s) [unused when PCMode=0]
99999.0             TPitManS(1) - Time to start override pitch maneuver for blade 1 and end standard
    pitch control (s)
99999.0             TPitManS(2) - Time to start override pitch maneuver for blade 2 and end standard
    pitch control (s)
99999.0             TPitManS(3) - Time to start override pitch maneuver for blade 3 and end standard
    pitch control (s) [unused for 2 blades]
10.0                PitManRat(1) - Pitch rate at which override pitch maneuver heads toward final
    pitch angle for blade 1 (deg/s)
10.0                PitManRat(2) - Pitch rate at which override pitch maneuver heads toward final
    pitch angle for blade 2 (deg/s)
10.0                PitManRat(3) - Pitch rate at which override pitch maneuver heads toward final
    pitch angle for blade 3 (deg/s) [unused for 2 blades]
90.0                BlPitchF(1) - Blade 1 final pitch for pitch maneuvers (degrees)
90.0                BlPitchF(2) - Blade 2 final pitch for pitch maneuvers (degrees)
90.0                BlPitchF(3) - Blade 3 final pitch for pitch maneuvers (degrees) [unused for 2
    blades]
---------------------- GENERATOR AND TORQUE CONTROL ----------------------------
0                   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)
1                   GenModel     - Generator model {1: simple, 2: Thevenin, 3: user-defined from
    routine UserGen} (switch) [used only when VSContrl=0]
94.4                GenEff       - Generator efficiency [ignored by the Thevenin and user-defined
    generator models] (%)
True                GenTiStr     - Method to start the generator {T: timed using TimGenOn, F: generator
     speed using SpdGenOn} (flag)
True                GenTiStp     - Method to stop the generator {T: timed using TimGenOf, F: when
    generator power = 0} (flag)
99999.0             SpdGenOn     - Generator speed to turn on the generator for a startup (HSS speed) (
    rpm) [used only when GenTiStr=False]
0.0                 TimGenOn     - Time to turn on the generator for a startup (s) [used only when
    GenTiStr=True]
99999.0             TimGenOf     - Time to turn off the generator (s) [used only when GenTiStp=True]
---------------------- SIMPLE VARIABLE-SPEED TORQUE CONTROL --------------------
99999.0             VS_RtGnSp    - Rated generator speed for simple variable-speed generator control (
    HSS side) (rpm) [used only when VSContrl=1]
99999.0             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]
99999.0             VS_Rgn2K     - Generator torque constant in Region 2 for simple variable-speed
    generator control (HSS side) (N-m/rpm^2) [used only when VSContrl=1]
99999.0             VS_SlPc      - Rated generator slip percentage in Region 2 1/2 for simple variable-
    speed generator control (%) [used only when VSContrl=1]
---------------------- SIMPLE INDUCTION GENERATOR ------------------------------
99999.0             SIG_SlPc     - Rated generator slip percentage (%) [used only when VSContrl=0 and
    GenModel=1]
99999.0             SIG_SySp     - Synchronous (zero-torque) generator speed (rpm) [used only when
    VSContrl=0 and GenModel=1]
99999.0             SIG_RtTq     - Rated torque (N-m) [used only when VSContrl=0 and GenModel=1]
99999.0             SIG_PORt     - Pull-out ratio (Tpullout/Trated) (-) [used only when VSContrl=0 and
    GenModel=1]
---------------------- THEVENIN-EQUIVALENT INDUCTION GENERATOR -----------------
```

```
99999.0              TEC_Freq   - Line frequency [50 or 60] (Hz) [used only when VSContrl=0 and
    GenModel=2]
0                    TEC_NPol   - Number of poles [even integer > 0] (-) [used only when VSContrl=0
    and GenModel=2]
99999.0              TEC_SRes   - Stator resistance (ohms) [used only when VSContrl=0 and GenModel=2]
99999.0              TEC_RRes   - Rotor resistance (ohms) [used only when VSContrl=0 and GenModel=2]
99999.0              TEC_VLL    - Line-to-line RMS voltage (volts) [used only when VSContrl=0 and
    GenModel=2]
99999.0              TEC_SLR    - Stator leakage reactance (ohms) [used only when VSContrl=0 and
    GenModel=2]
99999.0              TEC_RLR    - Rotor leakage reactance (ohms) [used only when VSContrl=0 and
    GenModel=2]
99999.0              TEC_MR     - Magnetizing reactance (ohms) [used only when VSContrl=0 and GenModel
    =2]
---------------------- HIGH-SPEED SHAFT BRAKE ----------------------------------
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)
99999.0              THSSBrDp   - Time to initiate deployment of the HSS brake (s)
99999.0              HSSBrDT    - Time for HSS-brake to reach full deployment once initiated (sec) [
    used only when HSSBrMode=1]
99999.0              HSSBrTqF   - Fully deployed HSS-brake torque (N-m)
---------------------- NACELLE-YAW CONTROL ------------------------------------
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)
0.0                  TYCOn      - Time to enable active yaw control (s) [unused when YCMode=0]
0.0                  YawNeut    - Neutral yaw position--yaw spring force is zero at this yaw (degrees)
9028320000.0            YawSpr     - Nacelle-yaw spring constant (N-m/rad)
19160000.0              YawDamp    - Nacelle-yaw damping constant (N-m/(rad/s))
9999.0               TYawManS   - Time to start override yaw maneuver and end standard yaw control (s)
1                    YawManRat  - Yaw maneuver rate (in absolute value) (deg/s)
0.0                  NacYawF    - Final yaw angle for override yaw maneuvers (degrees)
---------------------- TUNED MASS DAMPER ---------------------------------------
0                                      NumBStC - Number of blade structural controllers
"none"                          BStCfiles [-] - Name of the files for blade structural controllers (
    quoted strings on one line) [unused when NumBStC==0]
0                                      NumNStC [integer] - Number of nacelle structural controllers
"none"                          NStCfiles [-] - Name of the files for nacelle structural controllers (
    quoted strings on one line) [unused when NumNStC==0 ]
0                                      NumTStC [integer] - Number of tower structural controllers
"none"                          TStCfiles [-] - Names of the file for tower structural control damping
     (quoted strings on one line) [unused when NumTStC==0 ]
1                               NumSStC [integer] - Number of substructure structural controllers
"StC-Substructure1.dat" "StC-Substructure9.dat" "StC-Substructure33.dat" "StC-Substructure41.dat"
                     SStCfiles [-] - Name of the files for substructure structural controllers (
    quoted strings on one line) [unused when NumSStC==0 ]
---------------------- BLADED INTERFACE ---------------------------------------- [used only with
    Bladed Interface]
"controller\DISCON_IEA_offshore.dll" DLL_FileName - Name/location of the dynamic library {.dll [
    Windows] or .so [Linux]} in the Bladed-DLL format (-) [used only with Bladed Interface]
"IEA-10.0-198-RWT_DISCON.IN" DLL_InFile - Name of input file sent to the DLL (-) [used only with
    Bladed Interface]
"DISCON"             DLL_ProcName - Name of procedure in DLL to be called (-) [case sensitive; used
    only with DLL Interface]
default              DLL_DT     - Communication interval for dynamic library (s) (or "default") [used
    only with Bladed Interface]
False                DLL_Ramp   - Whether a linear ramp should be used between DLL_DT time steps [
    introduces time shift when true] (flag) [used only with Bladed Interface]
99999.0              BPCutoff   - Cuttoff frequency for low-pass filter on blade pitch from DLL (Hz) [
    used only with Bladed Interface]
```

```
0                    NacYaw_North - Reference yaw angle of the nacelle when the upwind end points due
    North (deg) [used only with Bladed Interface]
0                    Ptch_Cntrl - Record 28: Use individual pitch control {0: collective pitch; 1:
    individual pitch control} (switch) [used only with Bladed Interface]
0.0                  Ptch_SetPnt - Record 5: Below-rated pitch angle set-point (deg) [used only with
    Bladed Interface]
0.0                  Ptch_Min   - Record 6: Minimum pitch angle (deg) [used only with Bladed Interface
    ]
0.0                  Ptch_Max   - Record 7: Maximum pitch angle (deg) [used only with Bladed Interface
    ]
0.0                  PtchRate_Min - Record 8: Minimum pitch rate (most negative value allowed) (deg/s)
     [used only with Bladed Interface]
0.0                  PtchRate_Max - Record 9: Maximum pitch rate (deg/s) [used only with Bladed
    Interface]
0.0                  Gain_OM    - Record 16: Optimal mode gain (Nm/(rad/s)^2) [used only with Bladed
    Interface]
0.0                  GenSpd_MinOM - Record 17: Minimum generator speed (rpm) [used only with Bladed
    Interface]
0.0                  GenSpd_MaxOM - Record 18: Optimal mode maximum speed (rpm) [used only with Bladed
     Interface]
0.0                  GenSpd_Dem - Record 19: Demanded generator speed above rated (rpm) [used only
    with Bladed Interface]
0.0                  GenTrq_Dem - Record 22: Demanded generator torque above rated (Nm) [used only
    with Bladed Interface]
0.0                  GenPwr_Dem - Record 13: Demanded power (W) [used only with Bladed Interface]
---------------------- BLADED INTERFACE TORQUE-SPEED LOOK-UP TABLE -------------
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]
GenSpd_TLU           GenTrq_TLU
(rpm)                 (Nm)
---------------------- OUTPUT --------------------------------------------------
False                SumPrint   - Print summary data to <RootName>.sum (flag) (currently unused)
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)
True                 TabDelim   - Use tab delimiters in text tabular output file? (flag) (currently
    unused)
"ES10.3E2"           OutFmt     - Format used for text tabular output (except time). Resulting field
    should be 10 characters. (quoted string) (currently unused)
0.0                  TStart     - Time to begin tabular output (s) (currently unused)
                                       OutList     - The next line(s) contains a list of output
                                           parameters. See OutListParameters.xlsx for a listing of
                                           available output channels, (-)
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

D.6 HydroDyn input file

```
------- HydroDyn v2.03.* Input File --------------------------------------------
INNWIND.EU 10 MW Offshore Wind Turbine with Reference Jacket HydroDyn input properties       Jan
    Haefele (Leibniz Universitaet Hannover), j.haefele@isd.uni-hannover.de
False          Echo         - Echo the input file data (flag)
---------------------- ENVIRONMENTAL CONDITIONS --------------------------------
         1025  WtrDens      - Water density (kg/m^3)
         48.5  WtrDpth      - Water depth (meters)
                      0   MSL2SWL      - Offset between still-water level and mean sea level (meters)
                        [positive upward; must be zero if HasWAMIT=TRUE]
---------------------- WAVES ---------------------------------------------------
```

```
            2  WaveMod       - Incident wave kinematics model {0: none=still water, 1: regular (
                periodic), 1P#: regular (periodic) with user-specified phase, 2: JONSWAP/Pierson-
                Moskowitz spectrum (irregular), 3: White noise spectrum (irregular), 4: user-defined
                spectrum from routine UserWaveSpctrm (irregular), 5: GH Bladed wave data [option 5 is
                invalid for HasWAMIT = TRUE]} (switch)
            0  WaveStMod     - Model for stretching incident wave kinematics to instantaneous free
                surface {0: none=no stretching, 1: vertical stretching, 2: extrapolation stretching,
                3: Wheeler stretching} (switch) [unused when WaveMod=0 or when HasWAMIT = TRUE]
          660  WaveTMax      - Analysis time for incident wave calculations (sec) [unused when WaveMod
              =0] [determines WaveDOmega=2Pi/WaveTMax in the IFFT]
         0.01  WaveDT        - Time step for incident wave calculations (sec) [unused when WaveMod=0]
             [0.1<=WaveDT<=1.0 recommended] [determines WaveOmegaMax=Pi/WaveDT in the IFFT]
          6.0  WaveHs        - Significant wave height of incident waves (meters) [used only when
              WaveMod=1, 2, or 3]
           12  WaveTp        - Peak-spectral period of incident waves  (sec) [used only when WaveMod=1
                or 2]
"DEFAULT"      WavePkShp     - Peak-shape parameter of incident wave spectrum (-) or DEFAULT (string)
    [used only when WaveMod=2] [use 1.0 for Pierson-Moskowitz]
            0  WvLowCOff     - Low cut-off frequency or lower frequency limit of the wave spectrum
                beyond which the wave spectrum is zeroed (rad/s) [used only when WaveMod=2, 3, or 4]
          500  WvHiCOff      - High cut-off frequency or upper frequency limit of the wave spectrum
              beyond which the wave spectrum is zeroed (rad/s) [used only when WaveMod=2, 3, or 4]
            0  WaveDir       - Incident wave propagation heading direction                 (degrees)
                [unused when WaveMod=0 or 5]
            0  WaveDirMod    - Directional spreading function {0: none, 1: COS2S}          (-)      [
                only used when WaveMod=2,3,4]
            1  WaveDirSpread - Wave direction spreading coefficient ( > 0 )                (-)      [
                only used when WaveMod=2,3,4 and WaveDirMod=1]
            3  WaveNDir      - Number of wave directions                                   (-)      [
                odd number only, may be adjusted within HydroDyn]
           45  WaveDirRange  - Range of wave directions (full range: WaveDir +/- 1/2*WaveDirRange) (
               degrees) [only used when WaveMod=2,3,4 and WaveDirMod=1]
    123456789  WaveSeed(1)   - First random seed of incident waves [-2147483648 to 2147483647] (-)   [
         unused when WaveMod=0 or 5]
   1011121314  WaveSeed(2)   - Second random seed of incident waves [-2147483648 to 2147483647] (-)  [
        unused when WaveMod=0 or 5]
False          WaveNDAmp     - Flag for normally distributed amplitudes (flag)
""             WvKinFile     - Root name of GH Bladed files containing wave data (quoted string)     [
    used only when WaveMod=5]
            1  NWaveElev     - Number of points where the incident wave elevations can be computed (-)
                 [maximum of 9 output locations]
            0  WaveElevxi    - List of xi-coordinates for points where the incident wave elevations
                can be output (meters) [NWaveElev points, separated by commas or white space; usused
                if NWaveElev = 0]
            0  WaveElevyi    - List of yi-coordinates for points where the incident wave elevations
                can be output (meters) [NWaveElev points, separated by commas or white space; usused
                if NWaveElev = 0]
---------------------- 2ND-ORDER WAVES ---------------------------------------- [unused with WaveMod
    =0 or 6]
False          WvDiffQTF     - Full difference-frequency 2nd-order wave kinematics (flag)
False          WvSumQTF      - Full summation-frequency 2nd-order wave kinematics (flag)
            0  WvLowCOffD    - Low frequency cutoff used in the difference-frequencies (rad/s) [Only
                used with a difference-frequency method]
          3.5  WvHiCOffD     - High frequency cutoff used in the difference-frequencies (rad/s) [Only
              used with a difference-frequency method]
          0.1  WvLowCOffS    - Low frequency cutoff used in the summation-frequencies (rad/s) [Only
              used with a summation-frequency method]
          3.5  WvHiCOffS     - High frequency cutoff used in the summation-frequencies (rad/s) [Only
              used with a summation-frequency method]
---------------------- CURRENT ------------------------------------------------ [unused with WaveMod
```

```
    =6]
           0   CurrMod       - Current profile model {0: none=no current, 1: standard, 2: user-defined
                from routine UserCurrent} (switch)
           0   CurrSSV0      - Sub-surface current velocity at still water level (m/s) [used only when
                CurrMod=1]
"DEFAULT"      CurrSSDir     - Sub-surface current heading direction (degrees) or DEFAULT (string) [
    used only when CurrMod=1]
          20   CurrNSRef     - Near-surface current reference depth       (meters) [used only when
               CurrMod=1]
           0   CurrNSV0      - Near-surface current velocity at still water level (m/s) [used only
               when CurrMod=1]
           0   CurrNSDir     - Near-surface current heading direction    (degrees) [used only when
               CurrMod=1]
           0   CurrDIV       - Depth-independent current velocity           (m/s) [used only when
               CurrMod=1]
           0   CurrDIDir     - Depth-independent current heading direction (degrees) [used only when
               CurrMod=1]
---------------------- FLOATING PLATFORM --------------------------------------- [unused with WaveMod
    =6]
           0   PotMod        - Potential-flow model {0: none=no potential flow, 1: frequency-to-time-
               domain transforms based on WAMIT output, 2: fluid-impulse theory (FIT)} (switch)
0                            ExtnMod
           0   RdtnMod       - Radiation memory-effect model {0: no memory-effect calculation, 1:
               convolution, 2: state-space} (switch) [only used when PotMod=1; STATE-SPACE REQUIRES
               *.ss INPUT FILE]
           0   RdtnTMax      - Analysis time for wave radiation kernel calculations (sec) [only used
               when PotMod=1 and RdtnMod>0; determines RdtnDOmega=Pi/RdtnTMax in the cosine transform
               ; MAKE SURE THIS IS LONG ENOUGH FOR THE RADIATION IMPULSE RESPONSE FUNCTIONS TO DECAY
               TO NEAR-ZERO FOR THE GIVEN PLATFORM!]
           0   RdtnDT        - Time step for wave radiation kernel calculations (sec) [only used when
               PotMod=1 and RdtnMod=1; DT<=RdtnDT<=0.1 recommended; determines RdtnOmegaMax=Pi/RdtnDT
                in the cosine transform]
0                            NBody
0                            NBodyMod
""             PotFile       - Root name of potential-flow model data; WAMIT output files containing
    the linear, nondimensionalized, hydrostatic restoring matrix (.hst), frequency-dependent
    hydrodynamic added mass matrix and damping matrix (.1), and frequency- and direction-dependent
    wave excitation force vector per unit wave amplitude (.3) (quoted string) [MAKE SURE THE
    FREQUENCIES INHERENT IN THESE WAMIT FILES SPAN THE PHYSICALLY-SIGNIFICANT RANGE OF FREQUENCIES
    FOR THE GIVEN PLATFORM; THEY MUST CONTAIN THE ZERO- AND INFINITE-FREQUENCY LIMITS!]
           1   WAMITULEN     - Characteristic body length scale used to redimensionalize WAMIT output
               (meters) [only used when PotMod=1]
                0.0  PtfmRefxt - The xt offset of the body reference point(s) from(0,0,0) (meters)
                    [1 to NBody] [only used when PotMod=1]
                0.0  PtfmRefyt - The yt offset of the body reference point(s) from (0,0,0) (meters)
                    [1 to NBody] [only used when PotMod=1]
                0.0  PtfmRefzt - The zt offset of the body reference point(s) from (0,0,0) (meters)
                    [1 to NBody] [only used when PotMod=1. If NBody-Mod=2,PtfmRefzt=0.0]
                0.0  PtfmRefztRot - The rotation about zt of the body reference frame(s) from xt/yt
                    (degrees) [1 to NBody] [only used when PotMod=1]
           0   PtfmVol0      - Displaced volume of water when the platform is in its undisplaced
               position (m^3) [only used when PotMod=1; USE THE SAME VALUE COMPUTED BY WAMIT AS
               OUTPUT IN THE .OUT FILE!]
           0   PtfmCOBxt     - The xt offset of the center of buoyancy (COB) from the platform
               reference point (meters) [only used when PotMod=1]
           0   PtfmCOByt     - The yt offset of the center of buoyancy (COB) from the platform
               reference point (meters) [only used when PotMod=1]
---------------------- 2ND-ORDER FLOATING PLATFORM FORCES ---------------------- [unused with WaveMod
    =0 or 6, or PotMod=0 or 2]
           0   MnDrift       - Mean-drift 2nd-order forces computed                               {0:
                None; [7, 8, 9, 10, 11, or 12]: WAMIT file to use} [Only one of MnDrift, NewmanApp,
               or DiffQTF can be non-zero]
           0  NewmanApp    - Mean- and slow-drift 2nd-order forces computed with Newman's
               approximation {0: None; [7, 8, 9, 10, 11, or 12]: WAMIT file to use} [Only one of
               MnDrift, NewmanApp, or DiffQTF can be non-zero. Used only when WaveDirMod=0]
           0  DiffQTF      - Full difference-frequency 2nd-order forces computed with full QTF  {0:
                None; [10, 11, or 12]: WAMIT file to use}    [Only one of MnDrift, NewmanApp, or
               DiffQTF can be non-zero]
           0  SumQTF       - Full summation -frequency 2nd-order forces computed with full QTF  {0:
                None; [10, 11, or 12]: WAMIT file to use}
---------------------- PLATFORM ADDITIONAL STIFFNESS AND DAMPING --------------
           0           0           0           0           0           0  AddF0 - Additional
               preload (N, N-m) [If NBodyMod=1, one size 6*NBody x 1 vector; if NBodyMod>1, NBody
               size 6 x 1 vectors]
           0           0           0           0           0           0  AddCLin - Additional
               linear stiffness (N/m, N/rad, N-m/m, N-m/rad)
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0  AddBLin - Additional
               linear damping(N/(m/s), N/(rad/s), N-m/(m/s), N-m/(rad/s))
           0           0           0           0           0           0
           0           0       1.115919E+6     0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0  AddBQuad - Additional
               quadratic drag(N/(m/s)^2, N/(rad/s)^2, N-m(m/s)^2, N-m/(rad/s)^2)
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
           0           0           0           0           0           0
---------------------- AXIAL COEFFICIENTS --------------------------------------
           1  NAxCoef      - Number of axial coefficients (-)
AxCoefID AxCd    AxCa    AxCp
   (-)   (-)     (-)     (-)
    1    0.00    1.00    1.00
---------------------- MEMBER JOINTS -------------------------------------------
           62  NJoints      - Number of joints (-) [must be exactly 0 or at least 2]
JointID  Jointxi    Jointyi    Jointzi JointAxID  JointOvrlp  [JointOvrlp= 0: do nothing at joint, 1:
    eliminate overlaps by calculating super member]
   (-)    (m)        (m)        (m)       (-)      (switch)
1      -17    17     -48.6  1      0
2      -17    17     -47.5  1      0
3      -16.80366412  16.80366412   -46.214 1      0
4      -13.58381679  13.58381679   -25.124 1      0
5      -10.98229008  10.98229008   -8.084  1      0
6      -8.881526718  8.881526718   5.676   1      0
7      -7.183206107  7.183206107   16.8    1      0
8      -7     7      18     1      0
9      17     17     -48.6  1      0
10     17     17     -47.5  1      0
11     16.80366412   16.80366412   -46.214 1      0
12     13.58381679   13.58381679   -25.124 1      0
13     10.98229008   10.98229008   -8.084  1      0
14     8.881526718   8.881526718   5.676   1      0
15     7.183206107   7.183206107   16.8    1      0
```

```
16      7       7       18      1       0
17      -15.02315348    0       -34.513 1       0
18      -12.14530387    0       -15.704 1       0
19      -9.820821823    0       -0.509  1       0
20      -7.942595455    0       11.901  1       0
21      0       15.02315348     -34.513 1       0
22      0       12.14530387     -15.704 1       0
23      0       9.820821823     -0.509  1       0
24      0       7.942595455     11.901  1       0
25      15.02315348     0       -34.513 1       0
26      12.14530387     0       -15.704 1       0
27      9.820821823     0       -0.509  1       0
28      7.942595455     0       11.901  1       0
29      -16.80366412    0       -46.214 1       0
30      16.80366412     0       -46.214 1       0
31      0       16.80366412     -46.214 1       0
32      0       -16.80366412    -46.214 1       0
33      -17     -17     -48.6   1       0
34      -17     -17     -47.5   1       0
35      -16.80366412    -16.80366412    -46.214 1       0
36      -13.58381679    -13.58381679    -25.124 1       0
37      -10.98229008    -10.98229008    -8.084  1       0
38      -8.881526718    -8.881526718    5.676   1       0
39      -7.183206107    -7.183206107    16.8    1       0
40      -7      -7      18      1       0
41      17      -17     -48.6   1       0
42      17      -17     -47.5   1       0
43      16.80366412     -16.80366412    -46.214 1       0
44      13.58381679     -13.58381679    -25.124 1       0
45      10.98229008     -10.98229008    -8.084  1       0
46      8.881526718     -8.881526718    5.676   1       0
47      7.183206107     -7.183206107    16.8    1       0
48      7       -7      18      1       0
49      0       -15.02315348    -34.513 1       0
50      0       -12.14530387    -15.704 1       0
51      0       -9.820821823    -0.509  1       0
52      0       -7.942595455    11.901  1       0
53      0       0       18      1       0
54      -6.541984733    6.541984733     22      1       0
55      -2.711754506    2.711754506     26      1       0
56      2.711754506     2.711754506     26      1       0
57      6.541984733     6.541984733     22      1       0
58      -6.541984733    -6.541984733    22      1       0
59      -2.711754506    -2.711754506    26      1       0
60      2.711754506     -2.711754506    26      1       0
61      6.541984733     -6.541984733    22      1       0
62      0       0       26      1       0
---------------------- MEMBER CROSS-SECTION PROPERTIES -------------------------
           13   NPropSets     - Number of member property sets (-)
PropSetID   PropD        PropThck
   (-)       (m)           (m)
1       1.4     0.12
2       1.4     0.07
3       1.4     0.042
4       1.4     0.042
5       1.4     0.042
6       1.4     0.066
7       1.04    0.02
8       1.06    0.03
9       0.936   0.018
```

```
10     0.84   0.02
11     0.832  0.016
12     1.4    0.08
13     8.3    0.07
---------------------- SIMPLE HYDRODYNAMIC COEFFICIENTS (model 1) --------------
    SimplCd   SimplCdMG   SimplCa   SimplCaMG   SimplCp   SimplCpMG  SimplAxCd SimplAxCdMG SimplAxCa
        SimplAxCaMG SimplAxCp SimplAxCpMG
      (-)        (-)        (-)        (-)        (-)        (-)        (-)        (-)        (-)
                  (-)        (-)        (-)
      1.00       1.00       1.00       1.00       1.00       1.00       1.00       0.00       0.00
                1.00       1.00       1.00
---------------------- DEPTH-BASED HYDRODYNAMIC COEFFICIENTS (model 2) ---------
           0   NCoefDpth     - Number of depth-dependent coefficients (-)
Dpth     DpthCd  DpthCdMG  DpthCa  DpthCaMG     DpthCp  DpthCpMG  DpthAxCa  DpthAxCaMG     DpthAxCp
    DpthAxCpMG
(m)      (-)     (-)       (-)     (-)          (-)     (-)        (-)       (-)            (-)
          (-)
---------------------- MEMBER-BASED HYDRODYNAMIC COEFFICIENTS (model 3) --------
           0   NCoefMembers     - Number of member-based coefficients (-)
MemberID  MemberCd1   MemberCd2   MemberCdMG1  MemberCdMG2   MemberCa1   MemberCa2   MemberCaMG1
    MemberCaMG2   MemberCp1   MemberCp2   MemberCpMG1  MemberCpMG2  MemberAxCa1  MemberAxCa2
    MemberAxCaMG1 MemberAxCaMG2 MemberAxCp1 MemberAxCp2 MemberAxCpMG1 MemberAxCpMG2
   (-)       (-)         (-)         (-)         (-)         (-)         (-)         (-)
       (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)
               (-)         (-)         (-)         (-)         (-)
-------------------- MEMBERS -------------------------------------------------
         117   NMembers     - Number of members (-)
MemberID MJointID1 MJointID2 MPropSetID1 MPropSetID2 MDivSize MCoefMod PropWAMIT  [MCoefMod=1: use
    simple coeff table, 2: use depth-based coeff table, 3: use member-based coeff table] [ PropWAMIT
    = TRUE if member is modeled in WAMIT]
  (-)       (-)       (-)        (-)         (-)       (m)     (switch)  (flag)
1      1      2      1      1      1      1      FALSE
2      2      3      1      1      1      1      FALSE
3      3      4      2      2      1      1      FALSE
4      4      5      3      3      1      1      FALSE
5      5      6      4      4      1      1      FALSE
6      6      7      5      5      1      1      FALSE
7      7      8      6      6      1      1      FALSE
8      9      10     1      1      1      1      FALSE
9      10     11     1      1      1      1      FALSE
10     11     12     2      2      1      1      FALSE
11     12     13     3      3      1      1      FALSE
12     13     14     4      4      1      1      FALSE
13     14     15     5      5      1      1      FALSE
14     15     16     6      6      1      1      FALSE
15     33     34     1      1      1      1      FALSE
16     34     35     1      1      1      1      FALSE
17     35     36     2      2      1      1      FALSE
18     36     37     3      3      1      1      FALSE
19     37     38     4      4      1      1      FALSE
20     38     39     5      5      1      1      FALSE
21     39     40     6      6      1      1      FALSE
22     41     42     1      1      1      1      FALSE
23     42     43     1      1      1      1      FALSE
24     43     44     2      2      1      1      FALSE
25     44     45     3      3      1      1      FALSE
26     45     46     4      4      1      1      FALSE
27     46     47     5      5      1      1      FALSE
28     47     48     6      6      1      1      FALSE
29     3      29     7      7      1      1      FALSE
```

```
30      29      35      7       7       1       1       FALSE
31      3       17      8       8       1       1       FALSE
32      35      17      8       8       1       1       FALSE
33      4       17      8       8       1       1       FALSE
34      36      17      8       8       1       1       FALSE
35      4       18      9       9       1       1       FALSE
36      36      18      9       9       1       1       FALSE
37      5       18      9       9       1       1       FALSE
38      37      18      9       9       1       1       FALSE
39      5       19      10      10      0.2     1       FALSE
40      37      19      10      10      0.2     1       FALSE
41      6       19      10      10      0.2     1       FALSE
42      38      19      10      10      0.2     1       FALSE
43      6       20      11      11      1       1       FALSE
44      38      20      11      11      1       1       FALSE
45      7       20      11      11      1       1       FALSE
46      39      20      11      11      1       1       FALSE
47      11      30      7       7       1       1       FALSE
48      30      43      7       7       1       1       FALSE
49      11      25      8       8       1       1       FALSE
50      43      25      8       8       1       1       FALSE
51      12      25      8       8       1       1       FALSE
52      44      25      8       8       1       1       FALSE
53      12      26      9       9       1       1       FALSE
54      44      26      9       9       1       1       FALSE
55      13      26      9       9       1       1       FALSE
56      45      26      9       9       1       1       FALSE
57      13      27      10      10      0.2     1       FALSE
58      45      27      10      10      0.2     1       FALSE
59      14      27      10      10      0.2     1       FALSE
60      46      27      10      10      0.2     1       FALSE
61      14      28      11      11      1       1       FALSE
62      46      28      11      11      1       1       FALSE
63      15      28      11      11      1       1       FALSE
64      47      28      11      11      1       1       FALSE
65      3       31      7       7       1       1       FALSE
66      11      31      7       7       1       1       FALSE
67      3       21      8       8       1       1       FALSE
68      11      21      8       8       1       1       FALSE
69      4       21      8       8       1       1       FALSE
70      12      21      8       8       1       1       FALSE
71      4       22      9       9       1       1       FALSE
72      12      22      9       9       1       1       FALSE
73      5       22      9       9       1       1       FALSE
74      13      22      9       9       1       1       FALSE
75      5       23      10      10      0.2     1       FALSE
76      13      23      10      10      0.2     1       FALSE
77      6       23      10      10      0.2     1       FALSE
78      14      23      10      10      0.2     1       FALSE
79      6       24      11      11      1       1       FALSE
80      14      24      11      11      1       1       FALSE
81      7       24      11      11      1       1       FALSE
82      15      24      11      11      1       1       FALSE
83      35      32      7       7       1       1       FALSE
84      43      32      7       7       1       1       FALSE
85      35      49      8       8       1       1       FALSE
86      43      49      8       8       1       1       FALSE
87      36      49      8       8       1       1       FALSE
88      44      49      8       8       1       1       FALSE
89      36      50      9       9       1       1       FALSE
```

```
90     44     50     9      9      1      1      FALSE
91     37     50     9      9      1      1      FALSE
92     45     50     9      9      1      1      FALSE
93     37     51     10     10     0.2    1      FALSE
94     45     51     10     10     0.2    1      FALSE
95     38     51     10     10     0.2    1      FALSE
96     46     51     10     10     0.2    1      FALSE
97     38     52     11     11     1      1      FALSE
98     46     52     11     11     1      1      FALSE
99     39     52     11     11     1      1      FALSE
100    47     52     11     11     1      1      FALSE
101    8      53     12     12     1      1      FALSE
102    16     53     12     12     1      1      FALSE
103    40     53     12     12     1      1      FALSE
104    48     53     12     12     1      1      FALSE
105    53     62     13     13     1      1      FALSE
106    8      54     12     12     1      1      FALSE
107    16     57     12     12     1      1      FALSE
108    40     58     12     12     1      1      FALSE
109    48     61     12     12     1      1      FALSE
110    55     62     12     12     1      1      FALSE
111    56     62     12     12     1      1      FALSE
112    59     62     12     12     1      1      FALSE
113    60     62     12     12     1      1      FALSE
114    54     55     12     12     1      1      FALSE
115    57     56     12     12     1      1      FALSE
116    58     59     12     12     1      1      FALSE
117    61     60     12     12     1      1      FALSE
----------------------- FILLED MEMBERS -----------------------------------------
           1   NFillGroups    - Number of filled member groups (-) [If FillDens = DEFAULT, then
               FillDens = WtrDens; FillFSLoc is related to MSL2SWL]
FillNumM FillMList  FillFSLoc    FillDens
(-)     (-)           (m)          (kg/m^3)
66     1      2      3      4      5      8      9      10     11     12     29     30     31
    32     33     34     35     36     37     38     39     40     41     42     47     48     49
         50     51     52     53     54     55     56     57     58     59     60     65     66
         67     68     69     70     71     72     73     74     75     76     77     78     83
         84     85     86     87     88     89     90     91     92     93     94     95     96
         0      0
---------------------- MARINE GROWTH -------------------------------------------
           0   NMGDepths     - Number of marine-growth depths specified (-)
MGDpth    MGThck      MGDens
(m)       (m)        (kg/m^3)
---------------------- MEMBER OUTPUT LIST --------------------------------------
           0   NMOutputs     - Number of member outputs (-) [must be < 10]
MemberID  NOutLoc   NodeLocs [NOutLoc < 10; node locations are normalized distance from the start of
    the member, and must be >=0 and <= 1] [unused if NMOutputs=0]
  (-)       (-)       (-)
---------------------- JOINT OUTPUT LIST ---------------------------------------
           0   NJOutputs     - Number of joint outputs [Must be < 10]
   0          JOutLst       - List of JointIDs which are to be output (-)[unused if NJOutputs=0]
---------------------- OUTPUT --------------------------------------------------
False          HDSum         - Output a summary file [flag]
False          OutAll        - Output all user-specified member and joint loads (only at each member
    end, not interior locations) [flag]
           2   OutSwtch      - Output requested channels to: [1=Hydrodyn.out, 2=GlueCode.out, 3=both
               files]
"ES11.4e2"     OutFmt        - Output format for numerical results (quoted string) [not checked for
    validity!]
"A11"          OutSFmt       - Output format for header strings (quoted string) [not checked for
```

```
    validity!]
---------------------- OUTPUT CHANNELS ---------------------------------------
"Wave1Elev"
"HydroFxi"
"HydroFyi"
"HydroFzi"
"HydroMxi"
"HydroMyi"
"HydroMzi"
END of output channels and end of file. (the word "END" must appear in the first 3 columns of this
    line)
```

D.7 SubDyn

D.7.1 SubDyn input file

```
----------- SubDyn v1.03.x MultiMember Support Structure Input File ------------
INNWIND.EU 10MW Reference (Steel) Jacket SubDyn input properties
-------------------------- SIMULATION CONTROL ---------------------------------
False          Echo       - Echo input data to "<rootname>.SD.ech" (flag)
"DEFAULT"      SDdeltaT   - Local Integration Step. If "default", the glue-code integration step will
     be used.
           3   IntMethod  - Integration Method [1/2/3/4 = RK4/AB4/ABM4/AM2].
True           SttcSolve  - Solve dynamics about static equilibrium point
False                 GuyanLoadCorection - Include extra moment from lever arm at interface and
    rotate FEM for floating
-------------------- FEA and CRAIG-BAMPTON PARAMETERS---------------------------
           3   FEMMod     - FEM switch: element model in the FEM. [1= Euler-Bernoulli(E-B); 2=Tapered
                 E-B (unavailable); 3= 2-node Timoshenko; 4= 2-node tapered Timoshenko (unavailable)]
           3   NDiv       - Number of sub-elements per member
True           CBMod      - [T/F] If True perform C-B reduction, else full FEM dofs will be retained.
     If True, select Nmodes to retain in C-B reduced system.
          24   Nmodes     - Number of internal modes to retain (ignored if CBMod=False). If Nmodes=0
               --> Guyan Reduction.
         0.8   JDampings  - Damping Ratios for each retained mode (% of critical) If Nmodes>0, list
              Nmodes structural damping ratios for each retained mode (% of critical), or a single
              damping ratio to be applied to all retained modes. (last entered value will be used for
               all remaining modes).
                      1      GuyanDampMod Guyan damping [0=none, 1=Rayleigh Damping, 2= user
                         specified 6x6 matrix]
0.0225, 0.0016  RayleighDamp Mass and stiffness proportional damping coefficients ((,     ) Rayleigh
    damping) [only if GuyanDampMod=1]
                      6      GuyanDampSize - Guyan damping matrix size (square, 6x6) [only if
                         GuyanDampMod=2]
0          0            0            0            0            0    GuyanDampMat1
0          0            0            0            0            0    GuyanDampMat2
0          0            0            0            0            0    GuyanDampMat3
0          0            0            0            0            0    GuyanDampMat4
0          0            0            0            0            0    GuyanDampMat5
0          0            0            0            0            0    GuyanDampMat6
---- STRUCTURE JOINTS: joints connect structure members (~Hydrodyn Input File)---
          62   NJoints    - Number of joints (-)
JointID        JointXss            JointYss            JointZss    JointType JointDirX JointDirY
    JointDirZ JointStiff
  (-)             (m)                 (m)                 (m)         (-)       (-)       (-)       (-)
         (Nm/rad)
1      -17     17      -48.5   1      0      0      0      0
2      -17     17      -47.5   1      0      0      0      0
3      -16.80366412  16.80366412   -46.214 1      0      0      0      0
```

```
4      -13.58381679  13.58381679   -25.124 1      0      0      0      0
5      -10.98229008  10.98229008   -8.084  1      0      0      0      0
6      -8.881526718  8.881526718   5.676   1      0      0      0      0
7      -7.183206107  7.183206107   16.8    1      0      0      0      0
8      -7     7      18     1      0      0      0      0
9      17     17     -48.5  1      0      0      0      0
10     17     17     -47.5  1      0      0      0      0
11     16.80366412   16.80366412   -46.214 1      0      0      0      0
12     13.58381679   13.58381679   -25.124 1      0      0      0      0
13     10.98229008   10.98229008   -8.084  1      0      0      0      0
14     8.881526718   8.881526718   5.676   1      0      0      0      0
15     7.183206107   7.183206107   16.8    1      0      0      0      0
16     7      7      18     1      0      0      0      0
17     -15.02315348  0      -34.513 1      0      0      0      0
18     -12.14530387  0      -15.704 1      0      0      0      0
19     -9.820821823  0      -0.509  1      0      0      0      0
20     -7.942595455  0      11.901  1      0      0      0      0
21     0      15.02315348   -34.513 1      0      0      0      0
22     0      12.14530387   -15.704 1      0      0      0      0
23     0      9.820821823   -0.509  1      0      0      0      0
24     0      7.942595455   11.901  1      0      0      0      0
25     15.02315348   0      -34.513 1      0      0      0      0
26     12.14530387   0      -15.704 1      0      0      0      0
27     9.820821823   0      -0.509  1      0      0      0      0
28     7.942595455   0      11.901  1      0      0      0      0
29     -16.80366412  0      -46.214 1      0      0      0      0
30     16.80366412   0      -46.214 1      0      0      0      0
31     0      16.80366412   -46.214 1      0      0      0      0
32     0      -16.80366412  -46.214 1      0      0      0      0
33     -17    -17    -48.5  1      0      0      0      0
34     -17    -17    -47.5  1      0      0      0      0
35     -16.80366412  -16.80366412  -46.214 1      0      0      0      0
36     -13.58381679  -13.58381679  -25.124 1      0      0      0      0
37     -10.98229008  -10.98229008  -8.084  1      0      0      0      0
38     -8.881526718  -8.881526718  5.676   1      0      0      0      0
39     -7.183206107  -7.183206107  16.8    1      0      0      0      0
40     -7     -7     18     1      0      0      0      0
41     17     -17    -48.5  1      0      0      0      0
42     17     -17    -47.5  1      0      0      0      0
43     16.80366412   -16.80366412  -46.214 1      0      0      0      0
44     13.58381679   -13.58381679  -25.124 1      0      0      0      0
45     10.98229008   -10.98229008  -8.084  1      0      0      0      0
46     8.881526718   -8.881526718  5.676   1      0      0      0      0
47     7.183206107   -7.183206107  16.8    1      0      0      0      0
48     7      -7     18     1      0      0      0      0
49     0      -15.02315348  -34.513 1      0      0      0      0
50     0      -12.14530387  -15.704 1      0      0      0      0
51     0      -9.820821823  -0.509  1      0      0      0      0
52     0      -7.942595455  11.901  1      0      0      0      0
53     0      0      18     1      0      0      0      0
54     -6.389313     6.389313      22     1      0      0      0      0
55     -2.9344931    2.9344931     26     1      0      0      0      0
56     2.9344931     2.9344931     26     1      0      0      0      0
57     6.3893123     6.389313      22     1      0      0      0      0
58     -6.389313     -6.389313     22     1      0      0      0      0
59     -2.9344931    -2.9344931    26     1      0      0      0      0
60     2.9344931     -2.9344931    26     1      0      0      0      0
61     6.389313      -6.389313     22     1      0      0      0      0
62     0      0      26     1      0      0      0      0
------------------- BASE REACTION JOINTS: 1/0 for Locked/Free DOF @ each Reaction Node
```

```
    ---------------------
           4   NReact     - Number of Joints with reaction forces; be sure to remove all rigid motion
                DOFs of the structure (else det([K])=[0])
RJointID  RctTDXss   RctTDYss   RctTDZss   RctRDXss   RctRDYss   RctRDZss               SSIfile [
    Global Coordinate System]
  (-)      (flag)     (flag)     (flag)     (flag)     (flag)     (flag)                 (string)
1           0          0          0          0          0          0                   "SSIfile_Jacket.
    txt"
9           0          0          0          0          0          0                   "SSIfile_Jacket.
    txt"
33          0          0          0          0          0          0                   "SSIfile_Jacket.
    txt"
41          0          0          0          0          0          0                   "SSIfile_Jacket.
    txt"
------- INTERFACE JOINTS: 1/0 for Locked (to the TP)/Free DOF @each Interface Joint (only Locked-to-TP
     implemented thus far (=rigid TP)) ---------
           1   NInterf    - Number of interface joints locked to the Transition Piece (TP): be sure
               to remove all rigid motion dofs
IJointID  ItfTDXss   ItfTDYss   ItfTDZss   ItfRDXss   ItfRDYss   ItfRDZss    [Global Coordinate System]
  (-)      (flag)     (flag)     (flag)     (flag)     (flag)     (flag)
  62         1          1          1          1          1          1
----------------------------------- MEMBERS -------------------------------------
         117   NMembers   - Number of frame members
MemberID  MJointID1  MJointID2  MPropSetID1  MPropSetID2  MType    COSMID
  (-)        (-)        (-)         (-)          (-)        (-)          (-)
1      1      2      1      1      1
2      2      3      1      1      1
3      3      4      2      2      1
4      4      5      3      3      1
5      5      6      4      4      1
6      6      7      5      5      1
7      7      8      6      6      1
8      9      10     1      1      1
9      10     11     1      1      1
10     11     12     2      2      1
11     12     13     3      3      1
12     13     14     4      4      1
13     14     15     5      5      1
14     15     16     6      6      1
15     33     34     1      1      1
16     34     35     1      1      1
17     35     36     2      2      1
18     36     37     3      3      1
19     37     38     4      4      1
20     38     39     5      5      1
21     39     40     6      6      1
22     41     42     1      1      1
23     42     43     1      1      1
24     43     44     2      2      1
25     44     45     3      3      1
26     45     46     4      4      1
27     46     47     5      5      1
28     47     48     6      6      1
29     3      29     7      7      1
30     29     35     7      7      1
31     3      17     8      8      1
32     35     17     8      8      1
33     4      17     8      8      1
34     36     17     8      8      1
35     4      18     9      9      1
```

36	36	18	9	9	1
37	5	18	9	9	1
38	37	18	9	9	1
39	5	19	10	10	1
40	37	19	10	10	1
41	6	19	10	10	1
42	38	19	10	10	1
43	6	20	11	11	1
44	38	20	11	11	1
45	7	20	11	11	1
46	39	20	11	11	1
47	11	30	7	7	1
48	30	43	7	7	1
49	11	25	8	8	1
50	43	25	8	8	1
51	12	25	8	8	1
52	44	25	8	8	1
53	12	26	9	9	1
54	44	26	9	9	1
55	13	26	9	9	1
56	45	26	9	9	1
57	13	27	10	10	1
58	45	27	10	10	1
59	14	27	10	10	1
60	46	27	10	10	1
61	14	28	11	11	1
62	46	28	11	11	1
63	15	28	11	11	1
64	47	28	11	11	1
65	3	31	7	7	1
66	11	31	7	7	1
67	3	21	8	8	1
68	11	21	8	8	1
69	4	21	8	8	1
70	12	21	8	8	1
71	4	22	9	9	1
72	12	22	9	9	1
73	5	22	9	9	1
74	13	22	9	9	1
75	5	23	10	10	1
76	13	23	10	10	1
77	6	23	10	10	1
78	14	23	10	10	1
79	6	24	11	11	1
80	14	24	11	11	1
81	7	24	11	11	1
82	15	24	11	11	1
83	35	32	7	7	1
84	43	32	7	7	1
85	35	49	8	8	1
86	43	49	8	8	1
87	36	49	8	8	1
88	44	49	8	8	1
89	36	50	9	9	1
90	44	50	9	9	1
91	37	50	9	9	1
92	45	50	9	9	1
93	37	51	10	10	1
94	45	51	10	10	1
95	38	51	10	10	1

```
96      46      51      10      10      1
97      38      52      11      11      1
98      46      52      11      11      1
99      39      52      11      11      1
100     47      52      11      11      1
101     8       53      12      12      1
102     16      53      12      12      1
103     40      53      12      12      1
104     48      53      12      12      1
105     53      62      13      13      1
106     8       54      12      12      1
107     16      57      12      12      1
108     40      58      12      12      1
109     48      61      12      12      1
110     55      62      12      12      1
111     56      62      12      12      1
112     59      62      12      12      1
113     60      62      12      12      1
114     54      55      12      12      1
115     57      56      12      12      1
116     58      59      12      12      1
117     61      60      12      12      1
------------------ MEMBER X-SECTION PROPERTY data 1/2 [isotropic material for now: use this table for
    circular-tubular elements] ------------------------
           13  NPropSets  - Number of structurally unique x-sections (i.e. how many groups of X-
               sectional properties are utilized throughout all of the members)
PropSetID    YoungE         ShearG         MatDens        XsecD         XsecT
  (-)       (N/m2)         (N/m2)         (kg/m3)         (m)           (m)
1      2.10E+11      8.08E+10      7850   1.40   0.12
2      2.10E+11      8.08E+10      7850   1.40   0.07
3      2.10E+11      8.08E+10      7850   1.40   0.042
4      2.10E+11      8.08E+10      7850   1.40   0.042
5      2.10E+11      8.08E+10      7850   1.40   0.042
6      2.10E+11      8.08E+10      7850   1.40   0.066
7      2.10E+11      8.08E+10      7850   1.04   0.02
8      2.10E+11      8.08E+10      7850   1.06   0.03
9      2.10E+11      8.08E+10      7850   0.936  0.018
10     2.10E+11      8.08E+10      7850   0.84   0.02
11     2.10E+11      8.08E+10      7850   0.832  0.016
12     1.05E+12      4.04E+11      7850   1.40   0.08
13     1.05E+12      4.04E+11      7850   8.3    0.07
------------------ MEMBER X-SECTION PROPERTY data 2/2 [isotropic material for now: use this table if
    any section other than circular, however provide COSM(i,j) below] ------------------------
           0   NXPropSets - Number of structurally unique non-circular x-sections (if 0 the following
                table is ignored)
PropSetID    YoungE         ShearG         MatDens        XsecA         XsecAsx      XsecAsy
    XsecJxx      XsecJyy       XsecJ0
  (-)       (N/m2)         (N/m2)         (kg/m3)         (m2)           (m2)         (m2)         (m4)
             (m4)         (m4)
-------------------------- CABLE PROPERTIES -------------------------------------
0 NCablePropSets - Number of cable cable properties
PropSetID EA MatDens T0 CtrlChannel
(-) (N) (kg/m) (N) (-)
----------------------- RIGID LINK PROPERTIES ------------------------------------
0 NRigidPropSets - Number of rigid link properties
PropSetID MatDens
(-) (kg/m)
---------------------- MEMBER COSINE MATRICES COSM(i,j) ------------------------
           0   NCOSMs     - Number of unique cosine matrices (i.e., of unique member alignments
               including principal axis rotations); ignored if NXPropSets=0 or 9999 in any element
```

```
                below
COSMID   COSM11   COSM12   COSM13   COSM21   COSM22   COSM23   COSM31   COSM32   COSM33
 (-)      (-)      (-)      (-)      (-)      (-)      (-)      (-)      (-)      (-)
------------------------ JOINT ADDITIONAL CONCENTRATED MASSES--------------------------
           0  NCmass     - Number of joints with concentrated masses; Global Coordinate System
CMJointID     JMass           JMXX           JMYY           JMZZ        JMXY           JMXZ
           JMYZ    MCGX          MCGY    MCGZ
  (-)          (kg)         (kg*m^2)       (kg*m^2)       (kg*m^2)      (kg*m^2)   (kg*m^2)   (kg*m^2)
            (kg*m^2) (kg*m^2) (kg*m^2)
---------------------------- OUTPUT: SUMMARY & OUTFILE ------------------------------
False          SDSum      - Output a Summary File (flag).It contains: matrices K,M and C-B reduced
    M_BB, M-BM, K_BB, K_MM(OMG^2), PHI_R, PHI_L. It can also contain COSMs if requested.
False          OutCOSM    - Output cosine matrices with the selected output member forces (flag)
False          OutAll     - [T/F] Output all members' end forces
           2  OutSwtch   - [1/2/3] Output requested channels to: 1=<rootname>.SD.out; 2=<rootname>.
               out (generated by FAST); 3=both files.
False          TabDelim   - Generate a tab-delimited output in the <rootname>.SD.out file
           1  OutDec     - Decimation of output in the <rootname>.SD.out file
"ES11.4e2"     OutFmt     - Output format for numerical results in the <rootname>.SD.out file
"A11"          OutSFmt    - Output format for header strings in the <rootname>.SD.out file
------------------------- MEMBER OUTPUT LIST ------------------------------------------
           8  NMOutputs  - Number of members whose forces/displacements/velocities/accelerations
               will be output (-) [Must be <= 9].
MemberID  NOutCnt   NodeCnt [NOutCnt=how many nodes to get output for [< 10]; NodeCnt are local
    ordinal numbers from the start of the member, and must be >=1 and <= NDiv+1] If NMOutputs=0 leave
     blank as well.
  (-)       (-)       (-)
  1         1         1
  8         1         1
  15        1         1
  22        1         1
  31        1         1
  67        1         1
  33        1         4
  3         1         1
------------------------- SSOutList: The next line(s) contains a list of output parameters that will
    be output in <rootname>.SD.out or <rootname>.out. ------
"ReactFXss, ReactFYss, ReactFZss, ReactMXss, ReactMYss, ReactMZss" - Base reactions: fore-aft shear,
    side-to-side shear, side-to-side moment, fore-aft moment, yaw moment, vertical force
"M1N1TDxss, M1N1TDyss, M1N1TDzss, M1N1RDxe, M1N1RDye, M1N1RDze" - Mudline translations
"M2N1TDxss, M2N1TDyss, M2N1TDzss, M2N1RDxe, M2N1RDye, M2N1RDze" - Mudline translations
"M3N1TDxss, M3N1TDyss, M3N1TDzss, M3N1RDxe, M3N1RDye, M3N1RDze" - Mudline translations
"M4N1TDxss, M4N1TDyss, M4N1TDzss, M4N1RDxe, M4N1RDye, M4N1RDze" - Mudline translations
"M1N1FKxe, M2N1FKxe, M3N1FKxe, M4N1FKxe, M5N1FKxe, M6N1FKxe, M7N1FKxe, M8N1FKxe" - Static forces along
     local member coordinate system
"M1N1FKye, M2N1FKye, M3N1FKye, M4N1FKye, M5N1FKye, M6N1FKye, M7N1FKye, M8N1FKye" - Static forces along
     local member coordinate system
"M1N1FKze, M2N1FKze, M3N1FKze, M4N1FKze, M5N1FKze, M6N1FKze, M7N1FKze, M8N1FKze" - Static forces along
     local member coordinate system
"M1N1MKxe, M2N1MKxe, M3N1MKxe, M4N1MKxe, M5N1MKxe, M6N1MKxe, M7N1MKxe, M8N1MKxe" - Static moments
    along local member coordinate system
"M1N1MKye, M2N1MKye, M3N1MKye, M4N1MKye, M5N1MKye, M6N1MKye, M7N1MKye, M8N1MKye" - Static moments
    along local member coordinate system
"M1N1MKze, M2N1MKze, M3N1MKze, M4N1MKze, M5N1MKze, M6N1MKze, M7N1MKze, M8N1MKze" - Static moments
    along local member coordinate system
"M1N1FMxe, M2N1FMxe, M3N1FMxe, M4N1FMxe, M5N1FMxe, M6N1FMxe, M7N1FMxe, M8N1FMxe" - Dynamic forces
    along local member coordinate system
"M1N1FMye, M2N1FMye, M3N1FMye, M4N1FMye, M5N1FMye, M6N1FMye, M7N1FMye, M8N1FMye" - Dynamic forces
    along local member coordinate system
"M1N1FMze, M2N1FMze, M3N1FMze, M4N1FMze, M5N1FMze, M6N1FMze, M7N1FMze, M8N1FMze" - Dynamic forces
```

```
    along local member coordinate system
"M1N1MMxe, M2N1MMxe, M3N1MMxe, M4N1MMxe, M5N1MMxe, M6N1MMxe, M7N1MMxe, M8N1MMxe" - Dynamic moments
    along local member coordinate system
"M1N1MMye, M2N1MMye, M3N1MMye, M4N1MMye, M5N1MMye, M6N1MMye, M7N1MMye, M8N1MMye" - Dynamic moments
    along local member coordinate system
"M1N1MMze, M2N1MMze, M3N1MMze, M4N1MMze, M5N1MMze, M6N1MMze, M7N1MMze, M8N1MMze" - Dynamic moments
    along local member coordinate system
END of output channels and end of file. (the word "END" must appear in the first 3 columns of this
    line)
```

D.7.2 SSI file jacket

```
!---------------- Pile Head K and M elements -------------------!
!Equivalent Stiffness Constants: Kxx, Kyy, Kzz, Kxtx, Kxty..Kztx,Kzty,Kztz in any order; max 21
    elements
4.69155E+08                         Kxx
0.00000E+00                         Kxy
4.69155E+08                         Kyy
0.00000E+00                         Kxz
0.00000E+00                         Kyz
2.44494E+09                         Kzz
0.00000E+00                         Kxtx
1.93452E+09                         Kytx
0.00000E+00                         Kztx
1.52446E+10                         Ktxtx
-1.93452E+09                        Kxty
0.00000E+00                         Kyty
0.00000E+00                         Kzty
0.00000E+00                         Ktxty
1.52446E+10                         Ktyty
0.00000E+00                         Kxtz
0.00000E+00                         Kytz
0.00000E+00                         Kztz
0.00000E+00                         Ktxtz
0.00000E+00                         Ktytz
3.96802E+09                         Ktztz
```

D.7.3 SSI file monopile

```
!---------------- Pile Head K and M elements -------------------!
!Equivalent Stiffness Constants: Kxx, Kyy, Kzz, Kxtx, Kxty..Kztx,Kzty,Kztz in any order; max 21
    elements
3.271681280000000000e+09            Kxx
0.00000E+00                         Kxy
3.271681280000000000e+09            Kyy
0.00000E+00                         Kxz
0.00000E+00                         Kyz
9.727485952000000000e+09            Kzz
0.00000E+00                         Kxtx
2.844383641600000000e+10            Kytx
0.00000E+00                         Kztx
5.804918046720000000e+11            Ktxtx
-2.844383641600000000e+10           Kxty
0.00000E+00                         Kyty
0.00000E+00                         Kzty
0.00000E+00                         Ktxty
5.804918046720000000e+11            Ktyty
0.00000E+00                         Kxtz
0.00000E+00                         Kytz
```

```
0.00000E+00                         Kztz
0.00000E+00                         Ktxtz
0.00000E+00                         Ktytz
2.176937000960000000e+11            Ktztz
```

E. Fatigue analysis input file generation in Matlab

This is a script that generates OpenFAST input files for use in a fatigue analysis. Template files are given in Appendix D. Information not present in the script should be accessible from other sources. Most jacket substructure information is found in the template files. Monopile substructure information is available from the IEA Task 37 GitHub repository [78].

```matlab
% Modify FAST input files for the subcategories

%% Documentation
% This script creates and modifies FAST input files for fatigue analysis
% Some files within the FAST Matlab Toolbox have been modified to
%  accomodate for the datastrucutre in this script.
% Text files contain information that can be interpreted from Pedersen and
%  Askheim jacket or IEA Task 37 10MW reference monopile.
% Folders and files must be set in accordance this file structure. Analyse
%  file paths and destinations to figure this out.

%% Initialization
clear all; close all; clc;
restoredefaultpath;
addpath(genpath('PATH2TOOLBOX/matlab-toolbox')); % TODONE adapt me


%% Set up names and paths to files
oldFSTName = '.\_Template\10MW.fst';

[templateDir, baseName, ext ] = fileparts(oldFSTName);
if strcmp(templateDir, filesep)
    templateDir = ['.' filesep];
end


% Read template FST file
FP = FAST2Matlab(oldFSTName, 2); %FP are the FST parameters, specify 2 lines of header

% Path and basename for modified files
TwrFilePath_Root = ['.\10MW_ElastoDyn_Tower'];
FileName_BTS_root = '".\TurbSim\TurbSim';
HeaderBase = ['IEA Task 37 10MW reference OWT'];


%% Read or modify a sub file
% Read the sub file
SubFiles = ["EDFile" "InflowFile" "AeroFile" "ServoFile" "HydroFile" "SubFile"]; % EDFile InflowFile AeroFile ServoFile HydroFile SubFile
%SubFiles = ["InflowFile"];
windVels = [2:2:34]; % 2 4 ... 32 34
Damps = ["Base" "Damped"]; % Damped Base Radiation
SubSs = ["Monopile" "Jacket"]; % Monopile Jacket JacketBrace

% Parameters for windspeed
turbPark = [2 26:2:34];


for SubS = SubSs % Iterate through Substructure types
    % Parameters for substructure
    TowProp = ['TowPropTable' '_' convertStringsToChars(SubS) '.txt'];
    AddBLin = ['AddBLin' '_' convertStringsToChars(SubS) '.txt'];
    Joints = ['Joints' '_' convertStringsToChars(SubS) '.txt'];
    ReactionJoints = ['ReactionJoints' '_' convertStringsToChars(SubS) '.txt'];
    InterfaceJoints = ['InterfaceJoints' '_' convertStringsToChars(SubS) '.txt'];
    Members = ['Members' '_' convertStringsToChars(SubS) '.txt'];
    FillGroupsTableMList = ['FillGroupsTableMList' '_' convertStringsToChars(SubS) '.txt'];
    MemberSectionProp = ['MemberSectionProp' '_' convertStringsToChars(SubS) '.txt'];
    MemberOuts = ['MemberOuts' '_' convertStringsToChars(SubS) '.txt'];
    if (SubS == "Jacket")
        SubSHdr = ' atop INNWIND 10MW reference jacket';
        DT = '0.005';
        NDiv = '3';
        Nmodes = '24';
        TowerHt = '131.63';
        TowerBsHt = '26.0';
        PtfmYIner = '4.0513389E7';
        PtfmHeave = '-0.0156';
        TwFAM1Sh2 = '-0.8575';
        TwFAM1Sh3 = '8.2745';
        TwFAM1Sh4 = '-15.1777';
        TwFAM1Sh5 = '13.0613';
        TwFAM1Sh6 = '-4.3007';
        TwFAM2Sh2 = '1.1996';
        TwFAM2Sh3 = '-3.8005';
        TwFAM2Sh4 = '8.385';
        TwFAM2Sh5 = '-7.0279';
        TwFAM2Sh6 = '2.2438';
        TwSSM1Sh2 = '-0.8393';
        TwSSM1Sh3 = '8.1446';
        TwSSM1Sh4 = '-14.9227';
        TwSSM1Sh5 = '12.8376';
        TwSSM1Sh6 = '-4.2202';
        TwSSM2Sh2 = '1.1827';
        TwSSM2Sh3 = '-4.1511';
        TwSSM2Sh4 = '9.0914';
        TwSSM2Sh5 = '-7.7153';
        TwSSM2Sh6 = '2.5922';
        WtrDpth = '48.5';
        NJoints = '62';
        NReact = '4';
        NMembers = '117';
        NFillGroups = '1';
        FillGroupsTableNumM = '66';
        NPropSets = '13';
    elseif SubS == "Monopile"
        SubSHdr = ' atop IEA Task 37 reference monopile';
        DT = '0.001';
        NDiv = '4';
        Nmodes = '8';
        TowerHt = '115.63';
        TowerBsHt = '10.0';
        PtfmYIner = '1.327E7';
        PtfmHeave = '-0.0156';
        TwFAM1Sh2 = '0.9090';
        TwFAM1Sh3 = '0.4134';
        TwFAM1Sh4 = '-0.8094';
        TwFAM1Sh5 = '0.8126';
        TwFAM1Sh6 = '-0.3256';
        TwFAM2Sh2 = '-0.9741';
        TwFAM2Sh3 = '1.7876';
        TwFAM2Sh4 = '-0.2284';
        TwFAM2Sh5 = '0.5943';
        TwFAM2Sh6 = '-0.1793';
        TwSSM1Sh2 = '0.9025';
        TwSSM1Sh3 = '0.4146';
        TwSSM1Sh4 = '-0.8024';
        TwSSM1Sh5 = '0.8036';
        TwSSM1Sh6 = '-0.3183';
        TwSSM2Sh2 = '-1.0739';
        TwSSM2Sh3 = '1.6306';
        TwSSM2Sh4 = '0.1647';
        TwSSM2Sh5 = '0.1024';
        TwSSM2Sh6 = '0.1762';
        WtrDpth = '30.0';
        NJoints = '16';
        NReact = '1';
        NMembers = '15';
        NFillGroups = '0';
        NPropSets = '3';
    end

    for Damp = Damps % Iterate through Damping values
        % Parameters for damping
        TwrFilePath = [TwrFilePath_Root '_' convertStringsToChars(SubS) '_' convertStringsToChars(Damp) '.dat'];
        if Damp == "Damped" %Old values in comments
            DampHdr = ' with foundation damping included';
            RayleighDamp = '0.0225, 0.0016';
            if SubS == "Jacket"
                TwrDmpFA1 = '3.97'; %'1.60'
                TwrDmpFA2 = '7.35'; %'2.25'
                TwrDmpSS1 = '0.93'; %'0.8'
                TwrDmpSS2 = '1.1'; %'0.8'
                JDampings = '3.2'; %'1.67'
            elseif SubS == "Monopile"
                TwrDmpFA1 = '4.67'; %'1.66'
                TwrDmpFA2 = '9.50'; %'2.94'
                TwrDmpSS1 = '0.91'; %'0.8'
                TwrDmpSS2 = '1.17'; %'0.8'
                JDampings = '0.9 3.9 0.8 8.9 1.1 0.8 4.1 0.9'; %'0.8 1.66 0.8 2.94 0.8 0.8 1.77 1.0'
            end
        elseif Damp == "Base"
            DampHdr = ' with only baseline damping';
            RayleighDamp = '0.0225, 0.0016';
            if SubS == "Jacket"
                TwrDmpFA1 = '0.8';
                TwrDmpFA2 = '0.8';
                TwrDmpSS1 = '0.8';
                TwrDmpSS2 = '0.8';
                JDampings = '0.8';
            elseif SubS == "Monopile"
                TwrDmpFA1 = '0.8';
                TwrDmpFA2 = '0.8';
                TwrDmpSS1 = '0.8';
                TwrDmpSS2 = '0.8';
                JDampings = '0.8';
            end
        elseif (Damp == "Radiation") && (SubS == "Jacket")
            DampHdr = ' with hysteric and radiation damping included';
            RayleighDamp = '0.0225, 0.0016';
            TwrDmpFA1 = '4.39'; %'1.63'
            TwrDmpFA2 = '6.65'; %'2.85'
            TwrDmpSS1 = '0.84'; %'0.82'
            TwrDmpSS2 = '1.55'; %'1.19'
            JDampings = '4.2'; %'2.82'
        end
        newFSTFile = [baseName '_' convertStringsToChars(SubS) '_' convertStringsToChars(Damp)];

        for windVel = windVels % Iterate through Windspeeds
            FP_mod = FP;

            % For parked turbine
            if ismember(windVel, turbPark)
                ParkHdr = '(parked)';
                ParkChar = 'Park';
                GenDOF = 'False';
                RotSpeed = '0';
                BlPitch = '90';
                WakeMod = '0';
                AFAeroMod = '1';
            else % For producing turbines
                ParkHdr = '(producing power)';
                ParkChar = 'Prod';
                GenDOF = 'True';
                WakeMod = '1';
                AFAeroMod = '2';
            end

            % Find TurbSim .bts file
            windVelChar = int2str(windVel);
            if windVel < 10
                windVelChar = ['0' windVelChar];
            elseif windVel == 34
                windVelChar = '38';
            end
            FileName_BTS = [FileName_BTS_root windVelChar '.bts"'];

            % For wind/wave parameters for different windspeeds
            WindHdr = [' during ' int2str(windVel) 'm/s wind'];
            if windVel == 2
                WaveHs = '1.07';
                WaveTp = '6.03';
            elseif windVel == 4
                RotSpeed = '6.000';
                BlPitch = '2.589';
                WaveHs = '1.10';
                WaveTp = '5.88';
            elseif windVel == 6
                RotSpeed = '6.000';
                BlPitch = '2.589';
                WaveHs = '1.18';
                WaveTp = '5.76';
            elseif windVel == 8
                RotSpeed = '8.165';
                BlPitch = '0';
                WaveHs = '1.31';
                WaveTp = '5.67';
            elseif windVel == 10
                RotSpeed = '8.684';
                BlPitch = '0';
                WaveHs = '1.48';
                WaveTp = '5.74';
            elseif windVel == 12
                RotSpeed = '8.684';
                BlPitch = '5.975';
                WaveHs = '1.70';
                WaveTp = '5.88';
            elseif windVel == 14
```

```
            RotSpeed = '8.684';
            BlPitch = '10.121';
            WaveHs = '1.91';
            WaveTp = '6.07';
        elseif windVel == 16
            RotSpeed = '8.684';
            BlPitch = '13.280';
            WaveHs = '2.19';
            WaveTp = '6.37';
        elseif windVel == 18
            RotSpeed = '8.684';
            BlPitch = '16.018';
            WaveHs = '2.47';
            WaveTp = '6.71';
        elseif windVel == 20
            RotSpeed = '8.684';
            BlPitch = '18.513';
            WaveHs = '2.76';
            WaveTp = '6.99';
        elseif windVel == 22
            RotSpeed = '8.684';
            BlPitch = '20.752';
            WaveHs = '3.09';
            WaveTp = '7.40';
        elseif windVel == 24
            RotSpeed = '8.684';
            BlPitch = '22.991';
            WaveHs = '3.42';
            WaveTp = '7.80';
        elseif windVel == 26
            WaveHs = '3.76';
            WaveTp = '8.14';
        elseif windVel == 28
            WaveHs = '4.17';
            WaveTp = '8.49';
        elseif windVel == 30
            WaveHs = '4.46';
            WaveTp = '8.86';
        elseif windVel == 32
            WaveHs = '4.79';
            WaveTp = '9.12';
        elseif windVel == 34
            WaveHs = '4.90';
            WaveTp = '9.43';
        end

        for SubFile = SubFiles % Iterate through input files
            oldPathSub = RebaseFile(GetFASTPar(FP, SubFile), templateDir);
            if ismember(SubFile, ["EDFile" "InflowFile" "AeroFile" "ServoFile"])
                [paramSub, templateFilenameSub] = GetFASTPar_Subfile(FP, SubFile, templateDir, templateDir);
                if SubFile == "EDFile"
                    SubFileBase = [baseName '_ElastoDyn_' convertStringsToChars(SubS) '_' convertStringsToChars(Damp) '_' windVelChar];
                    SubHeader = [HeaderBase SubSHdr DampHdr WindHdr '.'];
                    paramSub = SetFASTPar(paramSub, 'GenDOF', GenDOF);
                    paramSub = SetFASTPar(paramSub, 'RotSpeed', RotSpeed);
                    paramSub = SetFASTPar(paramSub, 'BlPitch(1)', BlPitch);
                    paramSub = SetFASTPar(paramSub, 'BlPitch(2)', BlPitch);
                    paramSub = SetFASTPar(paramSub, 'BlPitch(3)', BlPitch);

                    oldTwrPath = [fileparts(templateFilenameSub) filesep erase(GetFASTPar(paramSub,'TwrFile'), '"')];
                    paramSubSub = FAST2Matlab(oldTwrPath, 2);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwrFADmp(1)', TwrDmpFA1);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwrFADmp(2)', TwrDmpFA2);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwrSSDmp(1)', TwrDmpSS1);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwrSSDmp(2)', TwrDmpSS2);

                    paramSub = SetFASTPar(paramSub, 'TowerHt', TowerHt);
                    paramSub = SetFASTPar(paramSub, 'TowerBsHt', TowerBsHt);
                    paramSub = SetFASTPar(paramSub, 'PtfmCMzt', TowerBsHt);
                    paramSub = SetFASTPar(paramSub, 'PtfmRefzt', TowerBsHt);
                    paramSub = SetFASTPar(paramSub, 'PtfmYIner', PtfmYIner);
                    paramSub = SetFASTPar(paramSub, 'PtfmHeave', PtfmHeave);
                    paramSub = SetFASTPar(paramSub, 'TwrFile', TwrFilePath);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM1Sh(2)', TwFAM1Sh2);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM1Sh(3)', TwFAM1Sh3);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM1Sh(4)', TwFAM1Sh4);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM1Sh(5)', TwFAM1Sh5);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM1Sh(6)', TwFAM1Sh6);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM2Sh(2)', TwFAM2Sh2);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM2Sh(3)', TwFAM2Sh3);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM2Sh(4)', TwFAM2Sh4);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM2Sh(5)', TwFAM2Sh5);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwFAM2Sh(6)', TwFAM2Sh6);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM1Sh(2)', TwSSM1Sh2);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM1Sh(3)', TwSSM1Sh3);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM1Sh(4)', TwSSM1Sh4);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM1Sh(5)', TwSSM1Sh5);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM1Sh(6)', TwSSM1Sh6);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM2Sh(2)', TwSSM2Sh2);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM2Sh(3)', TwSSM2Sh3);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM2Sh(4)', TwSSM2Sh4);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM2Sh(5)', TwSSM2Sh5);
                    paramSubSub = SetFASTPar(paramSubSub, 'TwSSM2Sh(6)', TwSSM2Sh6);
                    Matlab2FAST(paramSubSub, oldTwrPath, [fileparts(templateFilenameSub) filesep TwrFilePath], 2);

                elseif SubFile == "InflowFile"
                    SubFileBase = [baseName '_InflowWind_' convertStringsToChars(SubS) '_' windVelChar];
                    SubHeader = [HeaderBase SubSHdr WindHdr '.'];
                    paramSub = SetFASTPar(paramSub, 'FileName_BTS', FileName_BTS);

                    paramSub = SetFASTPar(paramSub, 'WindVziList', TowerHt);
                elseif SubFile == "AeroFile"
                    SubFileBase = [baseName '_AeroDyn_' convertStringsToChars(SubS) '_' ParkChar];
                    SubHeader = [HeaderBase SubSHdr ParkHdr '.'];
                    paramSub = SetFASTPar(paramSub, 'WakeMod', WakeMod);
                    paramSub = SetFASTPar(paramSub, 'AFAeroMod', AFAeroMod);

                    paramSub.TowProp = setfield(paramSub.TowProp, 'Table', readmatrix([fileparts(oldPathSub) filesep TowProp]));
                elseif SubFile == "ServoFile"
                    SubFileBase = [baseName '_ServoDyn'];
                    SubHeader = [HeaderBase '.'];
                end
                newPathSub = [fileparts(oldPathSub) filesep SubFileBase '.dat'];
                paramSub.HdrLines(2) = {SubHeader};
                Matlab2FAST(paramSub, templateFilenameSub, newPathSub, 2); %contains 2 header lines
            elseif SubFile == "HydroFile"
                SubFileBase = [baseName '_HydroDyn_' convertStringsToChars(SubS) '_' windVelChar];
                SubHeader = [HeaderBase SubSHdr WindHdr '.'];
                paramSub = HD2Matlab(oldPathSub, 2); %contains 2 header lines

                paramSub = SetFASTPar(paramSub, 'WaveHs', WaveHs);
                paramSub = SetFASTPar(paramSub, 'WaveTp', WaveTp);

                paramSub = SetFASTPar(paramSub, 'WtrDpth', WtrDpth);
                paramSub = setfield(paramSub, 'AddBLin', readmatrix([fileparts(oldPathSub) filesep AddBLin]));
                paramSub = SetFASTPar(paramSub, 'NJoints', NJoints);
                paramSub.Joints = setfield(paramSub.Joints, 'Table', readmatrix([fileparts(oldPathSub) filesep Joints]));
                paramSub = SetFASTPar(paramSub, 'NPropSets', NPropSets);
                paramSub.MemberSectionProp = ...
                    setfield(paramSub.MemberSectionProp, 'Table', readmatrix([fileparts(oldPathSub) filesep MemberSectionProp]));
                paramSub = SetFASTPar(paramSub, 'NMembers', NMembers);
                paramSub.Members = setfield(paramSub.Members, 'Table', readmatrix([fileparts(oldPathSub) filesep Members]));
                paramSub = SetFASTPar(paramSub, 'NFillGroups', NFillGroups);
                if str2num(NFillGroups) > 0
                    paramSub.FillGroups.Table.NumM = str2double(FillGroupsTableNumM);
                    paramSub.FillGroups.Table = ...
                        setfield(paramSub.FillGroups.Table, 'MList', readmatrix([fileparts(oldPathSub) filesep FillGroupsTableMList]));
                    paramSub.FillGroups.Table.FSLoc = 0;
                    paramSub.FillGroups.Table.Dens = 0;
                else
                    paramSub.FillGroups.Table(:) = [];
                end

                newPathSub = [fileparts(oldPathSub) filesep SubFileBase '.dat'];
                paramSub.HdrLines(2) = {SubHeader};
                Matlab2HD(paramSub, oldPathSub, newPathSub, 2); % there are 2 header lines
            elseif SubFile == "SubFile"
                SubFileBase = [baseName '_SubDyn_' convertStringsToChars(SubS) '_' convertStringsToChars(Damp)];
                SubHeader = [HeaderBase SubSHdr DampHdr '.'];
                paramSub = SD2Matlab(oldPathSub, 2); %contains 2 header lines

                paramSub = SetFASTPar(paramSub, 'NDiv', NDiv);
                paramSub = SetFASTPar(paramSub, 'Nmodes', Nmodes);

                paramSub = SetFASTPar(paramSub, 'JDampings', JDampings);
                paramSub = SetFASTPar(paramSub, 'RayleighDamp', RayleighDamp);

                paramSub = SetFASTPar(paramSub, 'NJoints', NJoints);
                paramSub = setfield(paramSub, 'Joints', readmatrix([fileparts(oldPathSub) filesep Joints]));
                paramSub = SetFASTPar(paramSub, 'NReact', NReact);
                paramSub = setfield(paramSub, 'ReactionJoints', readcell([fileparts(oldPathSub) filesep ReactionJoints]));
                paramSub = setfield(paramSub, 'InterfaceJoints', readmatrix([fileparts(oldPathSub) filesep InterfaceJoints]));
                paramSub = SetFASTPar(paramSub, 'NMembers', NMembers);
                paramSub = setfield(paramSub, 'Members', readmatrix([fileparts(oldPathSub) filesep Members]));
```

```
                    paramSub = SetFASTPar(paramSub, 'NPropSets', NPropSets);
                    paramSub = setfield(paramSub, 'MemberSectionProp', readmatrix([fileparts(oldPathSub) filesep MemberSectionProp]));
                    M = readmatrix([fileparts(oldPathSub) filesep MemberOuts]);
                    paramSub = setfield(paramSub, 'MemberOuts', ...
                        table2struct(table(M(:,1),M(:,2),M(:,3),'VariableNames',{'ID' 'NOutLoc' 'NodeLocs'})));

                    newPathSub = [fileparts(oldPathSub) filesep SubFileBase '.dat'];
                    paramSub.HdrLines(2) = {SubHeader};
                    Matlab2SD(paramSub, oldPathSub, newPathSub, 2); % there are 2 header lines
                end

                % Change file path in FST
                FP_mod = SetFASTPar(FP_mod, SubFile, ['"' '..\' newPathSub '"']);
            end

            % Modify FST file
            FP_mod.HdrLines(2) = {[HeaderBase SubSHdr DampHdr WindHdr ParkHdr '.']};
            FP_mod = SetFASTPar(FP_mod, 'DT', DT);
            newFSTName = [newFSTFile filesep newFSTFile '_' windVelChar ext];
            Matlab2FAST(FP_mod, oldFSTName, newFSTName, 2); %contains 2 header lines
        end
    end
end
disp("Files generated")
```

F. MLife input files

MLife input file for a fatigue analysis of jacket structure. Change channels and input files to make similar .mlif file for monopile.

```
----- MLife version 1.0 Input File -------------------------------------------
10MW-Jacket_Base
----- Job Options ------------------------------------------------------------
false               EchoInp         Echo input to <rootname>.echo as this file is being read.
true            StrNames        Use channel names following a "$" instead of numbers when specifying
    channels in this input file.
false               OutData         Output modified data array after scaling and calculated channels.
    (currently unavailable)
"%6.2e"         RealFmt         Format for outputting floating-point values.
"10MW_Jacket_Base"   RootName        Root name for aggregate output files.
----- Input-Data Layout ------------------------------------------------------
5               TitleLine       The row with the file title on it (zero if no title is available).
7               NamesLine       The row with the channel names on it (zero if no names are available
    or are specified below).
8               UnitsLine       The row with the channel units on it (zero if no units are available
    or are specified below).
12009           FirstDataLine   The first row of data.
0               NumChans       The number of channels in each input file.
ChanTitle   ChanUnits   Scale Offset  PSF_Type  NumCols rows of data follow. Title and units strings
    must be 10 characters or less.
----- Calculated Channels ----------------------------------------------------
8               NumCChan        The number calculated channels to generate.
1234567890      Seed            The integer seed for the random number generator (-2,147,483,648 to
    2,147,483,647)
Col_Title  Units   Equation     Put each field in quotes. Titles and units are limited to 10
    characters. NumCChan rows of data follow.S
"M1N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M1N1FKxe$).^2 + timeSeriesData (:,$M1N1FKye$).^2 )"
"M2N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M2N1FKxe$).^2 + timeSeriesData (:,$M2N1FKye$).^2 )"
"M3N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M3N1FKxe$).^2 + timeSeriesData (:,$M3N1FKye$).^2 )"
"M4N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M4N1FKxe$).^2 + timeSeriesData (:,$M4N1FKye$).^2 )"
"M5N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M5N1FKxe$).^2 + timeSeriesData (:,$M5N1FKye$).^2 )"
"M6N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M6N1FKxe$).^2 + timeSeriesData (:,$M6N1FKye$).^2 )"
"M7N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M7N1FKxe$).^2 + timeSeriesData (:,$M7N1FKye$).^2 )"
"M8N1FKxye" "(N)"  "sqrt(timeSeriesData(:,$M8N1FKxe$).^2 + timeSeriesData (:,$M8N1FKye$).^2 )"
----- Load Roses -------------------------------------------------------------
0               NumRoses        The number of load roses to generate.
Rose Name     Units   Channel1      Channel2    nSectors
----- Time and Wind Speed ----------------------------------------------------
1               TimeChan        The channel containing time.
2               WSChan          The primary wind-speed channel (used for mean wind speed and
    turbulence intensity, 0 for none)
----- Statistics and Extreme Events ------------------------------------------
true                    DoStats         Generate statistics of all the channels.
true            WrStatsTxt      Write the stats to a text file?
false                   WrStatsXLS      Write the stats to an Excel file?
0               NumSFChans      Number of channels that will have summary statistics generated for
    them.
0               SFChans         List of channels that will have summary statistics generated for them.
     Must number NumSFChans.
----- Distributions -----------------------------------------------------------
false           UserDistrib       User defined distribution? true = load user-specified distribution,
     false = only use Weibull wind distribution
2.04            WeibullShape      Weibull shape factor. If WeibullShape=2, enter the mean wind speed
    for WeibullScale.
```

```
12.50           WeibullScale      Weibull scale factor. If WeibullShape<>2. Otherwise, enter the mean
     wind speed.
4               WSin              Cut-in wind speed for the turbine.
25              WSout             Cut-out wind speed for the turbine.
42              WSmax             Maximum wind speed value for the wind-speed bins.
2.12            WSMaxBinSize      Maximum width of a wind-speed bin.
0               nDistribVars      Number of independent variables in the user-specified distribution,
     ignored if UserDistrib = false
""              DistribName       Filename of the user-supplied distribution table, ignored if
    UserDistrib = false
----- Fatigue -------------------------------------------------------------
35              nFatigueChannels  The number of fatigue channels. Next six lines ignored if zero.
0.0             FiltRatio         The fraction of the maximum range of each channel used as a cutoff
    range for the racetrack filter. Use zero for no filter.
630720000       DesignLife        Number of seconds in the design lifetime (20 years = 630720000
    seconds).
1               Availability      Fraction of the design life the turbine is operating when winds are
     between Vin and Vout
false                  BinCycles         Bin the rainflow cycles?
0.5             UCMult            Multiplier for binning unclosed cycles. (0 discards, 1 counts as a
    full cycle)
true            DoShortTerm       Compute simple (unweighted) damage-equivalent loads and damage
    rates.
true            DoLife            Do lifetime-related calculations?
true            DoAggregate       Compute a DELs and a damage result based on an aggregate of all the
     input files (does not use the wind spped distribution)
true            WrShortTermTxt    Write short-term results to plain-text files?
false           WrShortTermXLS    Write short-term resultsto an Excel workbook?
true            WrLifeTxt         Write lifetime results to plain-text files?
false           WrLifeXLS         Write lifetime results to an Excel workbook?
1               EquivalentFrequency The frequency of the damage equivalent load (Hz)
false           DEL_AsRange       true = report DELs as a range value, false = report as a one-sided
    amplitude
1               DEL_Type          1 = fixed mean, 2 = zero mean, 3 = both
1               GoodmanFlag       0 = no Goodman correction, 1 = use Goodman correction, 2 = compute
    results with and without Goodman correction
Channel# NSlopes SNslopeLst    BinFlag BinWidth/Number TypeLMF   LUlt   BinWidth not used when
    BinCycles is false. nFatigueChannels rows of data follow. LUlt >> LMF
$TwrBsFxt$      3       3 4 5         BN      40.0           WM       445992
$TwrBsFzt$      3       3 4 5         BN      40.0           WM       1213408
$TwrBsMyt$      3       3 4 5         BN      40.0           WM       2434317
$M1N1FKxye$      3       3 4 5         BN      40.0           WM        59965249
$M1N1FKze$      3       3 4 5         BN      40.0           WM        163147394
$M1N1MKxe$      3       3 4 5         BN      40.0           WM        48151788
$M1N1MKye$      3       3 4 5         BN      40.0           WM        48151788
$M2N1FKxye$      3       3 4 5         BN      40.0           WM        59965249
$M2N1FKze$      3       3 4 5         BN      40.0           WM        163147394
$M2N1MKxe$      3       3 4 5         BN      40.0           WM        48151788
$M2N1MKye$      3       3 4 5         BN      40.0           WM        48151788
$M3N1FKxye$      3       3 4 5         BN      40.0           WM        59965249
$M3N1FKze$      3       3 4 5         BN      40.0           WM        163147394
$M3N1MKxe$      3       3 4 5         BN      40.0           WM        48151788
$M3N1MKye$      3       3 4 5         BN      40.0           WM        48151788
$M4N1FKxye$      3       3 4 5         BN      40.0           WM        59965249
$M4N1FKze$      3       3 4 5         BN      40.0           WM        163147394
$M4N1MKxe$      3       3 4 5         BN      40.0           WM        48151788
$M4N1MKye$      3       3 4 5         BN      40.0           WM        48151788
$M5N1FKxye$      3       3 4 5         BN      40.0           WM        12063321
$M5N1FKze$      3       3 4 5         BN      40.0           WM        32820667
$M5N1MKxe$      3       3 4 5         BN      40.0           WM        8219100
```

```
$M5N1MKye$      3       3 4 5         BN      40.0           WM         8219100
$M6N1FKxye$      3       3 4 5         BN      40.0           WM         12063321
$M6N1FKze$      3       3 4 5         BN      40.0           WM         32820667
$M6N1MKxe$      3       3 4 5         BN      40.0           WM         8219100
$M6N1MKye$      3       3 4 5         BN      40.0           WM         8219100
$M7N1FKxye$      3       3 4 5         BN      40.0           WM         12063321
$M7N1FKze$      3       3 4 5         BN      40.0           WM         32820667
$M7N1MKxe$      3       3 4 5         BN      40.0           WM         8219100
$M7N1MKye$      3       3 4 5         BN      40.0           WM         8219100
$M8N1FKxye$      3       3 4 5         BN      40.0           WM         36346124
$M8N1FKze$      3       3 4 5         BN      40.0           WM         98886865
$M8N1MKxe$      3       3 4 5         BN      40.0           WM         31322414
$M8N1MKye$      3       3 4 5         BN      40.0           WM         31322414
4                   nGroups
Group Name NChannels    ChannelList
"TP"       3            1 2 3
"Mudline"      16            4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
"Braces"      12            20 21 22 23 24 25 26 27 28 29 30 31
"Column"      4            32 33 34 35
----- Input Files -----------------------------------------------------------
1               FileFormat       Format of input files. 1 = FAST ascii, 2 = FAST binary
11  1.1  1.3  1.5  1.7   (Weibull-Weighted Normal Operation: NumNormFiles, PSF1, PSF2, PSF3, PSF4)
"..\10MW_Jacket_Base\10MW_Jacket_Base_04.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_06.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_08.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_10.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_12.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_14.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_16.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_18.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_20.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_22.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_24.out"
6  1.1   1.3   1.5   1.7   (Weibull-Weighted Idling: NumIdleFiles, PSF1, PSF2, PSF3, PSF4)
"..\10MW_Jacket_Base\10MW_Jacket_Base_02.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_26.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_28.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_30.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_32.out"
"..\10MW_Jacket_Base\10MW_Jacket_Base_38.out"
0  1.2   1.3   1.4   1.6   (Discrete Events: NumDiscFiles, PSF1, PSF2, PSF3, PSF4)
==EOF==                         DO NOT REMOVE OR CHANGE. MUST COME JUST AFTER LAST LINE OF VALID
    INPUT.
```

G. TurbSim input files

G.1 Fatigue analysis template

```
---------TurbSim v2.00.* Input File------------------------
Input file for TurbSim.
---------Runtime Options-----------------------------------
False    Echo          - Echo input data to <RootName>.ech (flag)
123456   RandSeed1     - First random seed (-2147483648 to 2147483647)
789012   RandSeed2     - Second random seed (-2147483648 to 2147483647) for intrinsic pRNG, or an
    alternative pRNG: "RanLux" or "RNSNLW"
False    WrBHHTP       - Output hub-height turbulence parameters in binary form? (Generates RootName.
    bin)
False    WrFHHTP       - Output hub-height turbulence parameters in formatted form? (Generates
    RootName.dat)
False    WrADHH        - Output hub-height time-series data in AeroDyn form? (Generates RootName.hh)
True     WrADFF        - Output full-field time-series data in TurbSim/AeroDyn form? (Generates
    RootName.bts)
False    WrBLFF        - Output full-field time-series data in BLADED/AeroDyn form? (Generates
    RootName.wnd)
True     WrADTWR       - Output tower time-series data? (Generates RootName.twr)
False    WrFMTFF       - Output full-field time-series data in formatted (readable) form? (Generates
    RootName.u, RootName.v, RootName.w)
False    WrACT         - Output coherent turbulence time steps in AeroDyn form? (Generates RootName.
    cts)
True     Clockwise     - Clockwise rotation looking downwind? (used only for full-field binary files -
     not necessary for AeroDyn)
 0       ScaleIEC      - Scale IEC turbulence models to exact target standard deviation? [0=no
     additional scaling; 1=use hub scale uniformly; 2=use individual scales]

--------Turbine/Model Specifications-----------------------
65       NumGrid_Z     - Vertical grid-point matrix dimension
60       NumGrid_Y     - Horizontal grid-point matrix dimension
0.005    TimeStep      - Time step [seconds]
720      AnalysisTime  - Length of analysis time series [seconds] (program will add time if necessary:
     AnalysisTime = MAX(AnalysisTime, UsableTime+GridWidth/MeanHHWS) )
660      UsableTime    - Usable length of output time series [seconds] (program will add GridWidth/
    MeanHHWS seconds unless UsableTime is "ALL")
131.63   HubHt         - Hub height [m] (should be > 0.5*GridHeight)
225      GridHeight    - Grid height [m]
210      GridWidth     - Grid width [m] (should be >= 2*(RotorRadius+ShaftLength))
0        VFlowAng      - Vertical mean flow (uptilt) angle [degrees]
0        HFlowAng      - Horizontal mean flow (skew) angle [degrees]

--------Meteorological Boundary Conditions-------------------
"IECKAI" TurbModel     - Turbulence model ("IECKAI","IECVKM","GP_LLJ","NWTCUP","SMOOTH","WF_UPW","
    WF_07D","WF_14D","TIDAL","API","USRINP","TIMESR", or "NONE")
"unused" UserFile      - Name of the file that contains inputs for user-defined spectra or time
    series inputs (used only for "USRINP" and "TIMESR" models)
"3"      IECstandard   - Number of IEC 61400-x standard (x=1,2, or 3 with optional 61400-1 edition
    number (i.e. "1-Ed2") )
"B"      IECturbc      - IEC turbulence characteristic ("A", "B", "C" or the turbulence intensity in
    percent) ("KHTEST" option with NWTCUP model, not used for other models)
"NTM"    IEC_WindType  - IEC turbulence type ("NTM"=normal, "xETM"=extreme turbulence, "xEWM1"=
    extreme 1-year wind, "xEWM50"=extreme 50-year wind, where x=wind turbine class 1, 2, or 3)
"default" ETMc         - IEC Extreme Turbulence Model "c" parameter [m/s]
"default" WindProfileType - Velocity profile type ("LOG";"PL"=power law;"JET";"H2L"=Log law for TIDAL
    model;"API";"USR";"TS";"IEC"=PL on rotor disk, LOG elsewhere; or "default")
```

```
"unused" ProfileFile    - Name of the file that contains input profiles for WindProfileType="USR" and/
    or TurbModel="USRVKM" [-]
90.55    RefHt          - Height of the reference velocity (URef) [m]
12       URef           - Mean (total) velocity at the reference height [m/s] (or "default" for JET
    velocity profile) [must be 1-hr mean for API model; otherwise is the mean over AnalysisTime
    seconds]
350      ZJetMax        - Jet height [m] (used only for JET velocity profile, valid 70-490 m)
default  PLExp          - Power law exponent [-] (or "default")
0.002    Z0             - Surface roughness length [m] (or "default")

--------Non-IEC Meteorological Boundary Conditions------------
default  Latitude       - Site latitude [degrees] (or "default")
0.05     RICH_NO        - Gradient Richardson number [-]
default  UStar          - Friction or shear velocity [m/s] (or "default")
default  ZI             - Mixing layer depth [m] (or "default")
default  PC_UW          - Hub mean u'w' Reynolds stress [m^2/s^2] (or "default" or "none")
default  PC_UV          - Hub mean u'v' Reynolds stress [m^2/s^2] (or "default" or "none")
default  PC_VW          - Hub mean v'w' Reynolds stress [m^2/s^2] (or "default" or "none")

--------Spatial Coherence Parameters----------------------------
default  SCMod1          - u-component coherence model ("GENERAL", "IEC", "API", "NONE", or "default")
default  SCMod2          - v-component coherence model ("GENERAL", "IEC", "NONE", or "default")
default  SCMod3          - w-component coherence model ("GENERAL", "IEC", "NONE", or "default")
default  InCDec1         - u-component coherence parameters for general or IEC models [-, m^-1] (e.g.
    "10.0 0.3e-3" in quotes) (or "default")
default  InCDec2         - v-component coherence parameters for general or IEC models [-, m^-1] (e.g.
    "10.0 0.3e-3" in quotes) (or "default")
default  InCDec3         - w-component coherence parameters for general or IEC models [-, m^-1] (e.g.
    "10.0 0.3e-3" in quotes) (or "default")
default  CohExp          - Coherence exponent for general model [-] (or "default")

--------Coherent Turbulence Scaling Parameters-------------------
"C:\Users\oysteiml\TurbSim\CertTest\EventData" CTEventPath - Name of the path where event data files
    are located
"Random" CTEventFile    - Type of event files ("LES", "DNS", or "RANDOM")
true     Randomize      - Randomize the disturbance scale and locations? (true/false)
 1.0     DistScl        - Disturbance scale [-] (ratio of event dataset height to rotor disk). (
     Ignored when Randomize = true.)
 0.5     CTLy           - Fractional location of tower centerline from right [-] (looking downwind) to
      left side of the dataset. (Ignored when Randomize = true.)
 0.5     CTLz           - Fractional location of hub height from the bottom of the dataset. [-] (
     Ignored when Randomize = true.)
30.0     CTStartTime    - Minimum start time for coherent structures in RootName.cts [seconds]

====================================================
! NOTE: Do not add or remove any lines in this file!
====================================================
```

G.2 ULS analysis

```
---------TurbSim v2.00.* Input File------------------------
Input file for TurbSim.
---------Runtime Options-----------------------------------
False    Echo           - Echo input data to <RootName>.ech (flag)
123456   RandSeed1      - First random seed (-2147483648 to 2147483647)
789012   RandSeed2      - Second random seed (-2147483648 to 2147483647) for intrinsic pRNG, or an
    alternative pRNG: "RanLux" or "RNSNLW"
False    WrBHHTP        - Output hub-height turbulence parameters in binary form? (Generates RootName.
    bin)
```

```
False    WrFHHTP       - Output hub-height turbulence parameters in formatted form? (Generates
    RootName.dat)
False    WrADHH        - Output hub-height time-series data in AeroDyn form? (Generates RootName.hh)
True     WrADFF        - Output full-field time-series data in TurbSim/AeroDyn form? (Generates
    RootName.bts)
False    WrBLFF        - Output full-field time-series data in BLADED/AeroDyn form? (Generates
    RootName.wnd)
True     WrADTWR       - Output tower time-series data? (Generates RootName.twr)
False    WrFMTFF       - Output full-field time-series data in formatted (readable) form? (Generates
    RootName.u, RootName.v, RootName.w)
False    WrACT         - Output coherent turbulence time steps in AeroDyn form? (Generates RootName.
    cts)
True     Clockwise     - Clockwise rotation looking downwind? (used only for full-field binary files -
     not necessary for AeroDyn)
 0       ScaleIEC      - Scale IEC turbulence models to exact target standard deviation? [0=no
     additional scaling; 1=use hub scale uniformly; 2=use individual scales]

--------Turbine/Model Specifications-----------------------
60       NumGrid_Z     - Vertical grid-point matrix dimension
60       NumGrid_Y     - Horizontal grid-point matrix dimension
0.005    TimeStep      - Time step [seconds]
720      AnalysisTime  - Length of analysis time series [seconds] (program will add time if necessary:
     AnalysisTime = MAX(AnalysisTime, UsableTime+GridWidth/MeanHHWS) )
660      UsableTime    - Usable length of output time series [seconds] (program will add GridWidth/
    MeanHHWS seconds unless UsableTime is "ALL")
115.63   HubHt         - Hub height [m] (should be > 0.5*GridHeight)
210      GridHeight    - Grid height [m]
210      GridWidth     - Grid width [m] (should be >= 2*(RotorRadius+ShaftLength))
0        VFlowAng      - Vertical mean flow (uptilt) angle [degrees]
0        HFlowAng      - Horizontal mean flow (skew) angle [degrees]

--------Meteorological Boundary Conditions-------------------
"IECKAI" TurbModel     - Turbulence model ("IECKAI","IECVKM","GP_LLJ","NWTCUP","SMOOTH","WF_UPW","
    WF_07D","WF_14D","TIDAL","API","USRINP","TIMESR", or "NONE")
"unused" UserFile      - Name of the file that contains inputs for user-defined spectra or time
    series inputs (used only for "USRINP" and "TIMESR" models)
"3"      IECstandard   - Number of IEC 61400-x standard (x=1,2, or 3 with optional 61400-1 edition
    number (i.e. "1-Ed2") )
"A"      IECturbc      - IEC turbulence characteristic ("A", "B", "C" or the turbulence intensity in
    percent) ("KHTEST" option with NWTCUP model, not used for other models)
"1EWM50" IEC_WindType  - IEC turbulence type ("NTM"=normal, "xETM"=extreme turbulence, "xEWM1"=
    extreme 1-year wind, "xEWM50"=extreme 50-year wind, where x=wind turbine class 1, 2, or 3)
"default" ETMc         - IEC Extreme Turbulence Model "c" parameter [m/s]
"default" WindProfileType - Velocity profile type ("LOG";"PL"=power law;"JET";"H2L"=Log law for TIDAL
    model;"API";"USR";"TS";"IEC"=PL on rotor disk, LOG elsewhere; or "default")
"unused" ProfileFile   - Name of the file that contains input profiles for WindProfileType="USR" and/
    or TurbModel="USRVKM" [-]
115.63   RefHt         - Height of the reference velocity (URef) [m]
50       URef          - Mean (total) velocity at the reference height [m/s] (or "default" for JET
    velocity profile) [must be 1-hr mean for API model; otherwise is the mean over AnalysisTime
    seconds]
350      ZJetMax       - Jet height [m] (used only for JET velocity profile, valid 70-490 m)
default  PLExp         - Power law exponent [-] (or "default")
0.002    Z0            - Surface roughness length [m] (or "default")

--------Non-IEC Meteorological Boundary Conditions------------
default  Latitude      - Site latitude [degrees] (or "default")
0.05     RICH_NO       - Gradient Richardson number [-]
default  UStar         - Friction or shear velocity [m/s] (or "default")
default  ZI            - Mixing layer depth [m] (or "default")
default  PC_UW         - Hub mean u'w' Reynolds stress [m^2/s^2] (or "default" or "none")
default  PC_UV         - Hub mean u'v' Reynolds stress [m^2/s^2] (or "default" or "none")
default  PC_VW         - Hub mean v'w' Reynolds stress [m^2/s^2] (or "default" or "none")

--------Spatial Coherence Parameters----------------------------
default  SCMod1         - u-component coherence model ("GENERAL", "IEC", "API", "NONE", or "default")
default  SCMod2         - v-component coherence model ("GENERAL", "IEC", "NONE", or "default")
default  SCMod3         - w-component coherence model ("GENERAL", "IEC", "NONE", or "default")
default  InCDec1        - u-component coherence parameters for general or IEC models [-, m^-1] (e.g.
    "10.0 0.3e-3" in quotes) (or "default")
default  InCDec2        - v-component coherence parameters for general or IEC models [-, m^-1] (e.g.
    "10.0 0.3e-3" in quotes) (or "default")
default  InCDec3        - w-component coherence parameters for general or IEC models [-, m^-1] (e.g.
    "10.0 0.3e-3" in quotes) (or "default")
default  CohExp         - Coherence exponent for general model [-] (or "default")

--------Coherent Turbulence Scaling Parameters-------------------
"C:\Users\oysteiml\TurbSim\CertTest\EventData" CTEventPath - Name of the path where event data files
    are located
"Random" CTEventFile   - Type of event files ("LES", "DNS", or "RANDOM")
true     Randomize     - Randomize the disturbance scale and locations? (true/false)
 1.0     DistScl       - Disturbance scale [-] (ratio of event dataset height to rotor disk). (
     Ignored when Randomize = true.)
 0.5     CTLy          - Fractional location of tower centerline from right [-] (looking downwind) to
      left side of the dataset. (Ignored when Randomize = true.)
 0.5     CTLz          - Fractional location of hub height from the bottom of the dataset. [-] (
     Ignored when Randomize = true.)
30.0     CTStartTime   - Minimum start time for coherent structures in RootName.cts [seconds]

====================================================
! NOTE: Do not add or remove any lines in this file!
====================================================
```

NTNU
Norwegian University of
Science and Technology