NTNU – Trondheim
Norwegian University of
Science and Technology

Aero-Hydro Dynamic Analysis of Offshore Wind Turbine

Implementation of Nonlinear Soil-Structure Interaction in Software FAST

Veronica Liverud Krathe

Mechanical Engineering
Submission date: June 2015
Supervisor: Amir Kaynia, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Abstract

Bottom-mounted *offshore wind turbines* (OWTs) involve a wide range of engineering fields. Of these, modeling of foundation flexibility has been given little priority. This thesis looks at the modeling of a bottom-mounted offshore wind turbine in the non-linear aero-hydro-servo-elastic simulation tool *FAST v7*. The OWT considered is supported on a monopile. The main concern of the thesis was to implement a nonlinear foundation representation in this program. The *National Renewable Energy Laboratory*'s (NREL) fictitious 5MW reference turbine was used as a base for the analyses [19].

FAST v7 has an open source code that is written in the programming language *Fortran 90*. As a base for the execution of the task of this thesis, a thorough understanding of the FAST v7 source code and Fortran was required. Therefore, an introduction to the features and structure of FAST v7 is given. Relevant Fortran codes are presented in the appendices. In addition to FAST v7, NREL provides softwares such as *TurbSim* and *BModes* that were utilized. Input files for these are also provided in the appendices.

Moreover, a description of the modeling of environmental loads such as waves and wind, as well as foundation flexibility, is given. The history, current status and terminology of OWTs, in addition to other topics related to wind turbines, such as power generation, natural frequencies and fatigue, are also presented.

Regarding analyses, they are presented over three chapters. First, the base model of the NREL 5MW reference turbine is introduced. Properties of the OWT are given, and typical inputs and outputs are described. Default modeling of foundation in FAST v7 is by means of a rigid foundation, a setting which is kept in this chapter. This implies that soil stiffness and damping is disregarded. Damping may lead to lower design load estimates. A softer foundation, on the other hand, will reduce the natural periods of the system, shifting them closer to the frequencies of the environmental loads. This may in turn lead to amplified moments at the mudline. Therefore, including soil stiffness and damping in the analyses is important.

Correspondingly, the second part concerns a modified version of FAST v7, where *coupled linear springs* are applied for the modeling of the foundation flexibility. The deriva-

tion of an applied foundation stiffness matrix is described, provided by Passon and Kühn [31]. Response of a model with this stiffness is compared with two other models, having a softer and stiffer foundation, respectively.

Finally, a *nonlinear* foundation is introduced in FAST v7, by means of uncoupled parallel springs as proposed by Iwan [12]. To verify that the implementation was successful, load-displacement curves of the springs are presented. These show the typical hysteresis loops of an inelastic material, which confirms that a nonlinear foundation is implemented. Moreover, constant damping was added in the foundation.

Hence, the conclusion in this thesis is that the objective of implementing a nonlinear foundation representation in FAST v7 was successful.

Sammendrag

Bunnfaste offshore vindturbiner involverer et bredt spekter av ingeniørfelt. Av disse har modellering av fleksibilitet i fundamentet blitt nedprioritert. Denne oppgaven ser på modellering av bunnfaste vindturbiner i det ikkelineære aero-hydro-servoelastiske simuleringsverktøyet *FAST v7*. Vindturbinen som blir sett på her står på en monopel. Hovedformålet med denne oppgaven var å implementere en ikkelineær fremstilling av fundamentet i dette programmet. Det amerikanske *National Renewable Energy Laboratory* (NREL) har laget en fiktiv 5MW referanseturbin som ble benyttet som grunnlag for analysene [19].

FAST v7 har en åpen kildekode som er skrevet i programmeringsspråket *Fortran f90*. Som et grunnlag for utførelsen av denne oppgaven krevdes en grundig forståelse av kildekoden til FAST v7, og Fortran. Derfor gis en introduksjon til oppbyggingen av, og egenskapene til, FAST v7. Relevant Fortran-kode er presentert i vedleggene. I tillegg til FAST v7 distribuerer NREL programvare som *TurbSim* og *BModes*, som også ble benyttet. Input-filer tilhørende disse er også gitt i vedleggene.

Videre gis en beskrivelse av modellering av naturlaster som bølger og vind, i tillegg til fleksibilitet i fundamentet. Historie, gjeldende status, og terminologi vedrørende offshore vindturbiner presenteres, sammen med andre temaer relatert til vindturbiner, som kraftgenerering, naturlige frekvenser og utmattelse.

Når det gjelder analyser er disse presentert over tre kapitler. Først introduseres NREL 5MW referanseturbinen som en basismodell. Egenskaper ved vindturbinen gis, og typiske inndata og resultater beskrives. Standard modellering av fundament i FAST v7 er i form av en fast innspenning, en innstilling som er beholdt i dette kapittelt. Dette innebærer at stivhet og dempning i jorda sees bort fra. Dempning kan lede til lavere dimensjonerende last. Et mykere fundament vil på den andre siden redusere de naturlige periodene til systemet, slik at de kommer nærmere frekvensene til naturlastene. Dette kan føre til økt moment ved havbunnen. Derfor er det viktig å inkludere stivhet og dempning i jorda i analyser.

Tilsvarende tar den neste delen for seg en modifisert versjon av FAST v7, hvor *koblede*

lineære fjærer benyttes for å modellere fleksibilitet i jorda. Utledning av en stivhetsmatrise for fundamentet beskrives. Denne er beregnet av Passon and Kühn [31]. Responsen til en modell som innehar denne stivheten sammenlignes med to andre modeller, en med et mykere fundament og en med et stivere fundament.

Til slutt introduseres et *ikkelineært fundament* i FAST v7. Dette gjøres i form av ukoblede parallelle fjærer, som foreslått av Iwan [12]. For å verifisere at implementeringen var vellykket, presenteres last-forflytningskurver. Disse viser hystereseløkker som er typiske for et uelastisk materiale, noe som bekrefter at et ikkelineært fundament er implementer. I tillegg ble konstant dempning inkludert i fundamentet.

Resultatet av oppgaven var altså at formålet med å implementere en ikkelineær fundamentfremstilling i FAST v7 var vellykket.

Preface

This Master's thesis marks the end of a five year degree of Master of Science in Mechanical Engineering at the Norwegian University of Science and Technology (NTNU), Trondheim. The thesis was written over a period of 20 weeks during the spring of 2015, at the Department of Structural Engineering.

My specialization in Applied Mechanics, more specifically Computational Mechanics, has provided me with insight in a broad range of engineering fields, from aerodynamics and fluid dynamics, to structural dynamics, fracture mechanics, mechanics of materials and finite element analysis. This constituted a beneficial base for the challenges that this thesis demonstrated.

These 20 weeks have made up the most informative semester of my stay at NTNU. Including new and better insight in theoretical fields, I have also obtained experience and knowledge regarding software and programming, including FAST, Excel, Matlab, Fortran and Latex, the latter of which this thesis was written in.

Acknowledgements

I would like to award a huge thanks to my supervisor, Adjunct Professor Amir M. Kaynia of Norwegian Geotechnical Institute (NGI) and NTNU, for his genuine dedication and commitment. It has been an inspiring collaboration, from which I have obtained great knowledge and experience.

Great gratitude must also be given to senior engineer at the American *National Wind Technology Center* (NWTC), Jason Jonkman, for being helpful and accessible through emails, quickly replying when I had questions regarding FAST.

I also want to give credit to fellow student, Marte Talberg, for informative and entertaining discussions, of which both academic conversation and less serious business was appreciated.

Finally, I want to thank Ruth Liverud, for teaching me to appreciate the great value of knowledge.

Trondheim, 2015-06-09

Veronica Liverud Krathe

Contents

List of Tables

List of Figures

Acronyms

ADAMS Automatic Dynamic Analysis of Mechanical Systems

AF Apparent Fixity

BEM Blade Element Momentum

CAE Computer-Aided Engineering

CM Center of Mass

CLS Coupled Linear Springs

DLL Dynamic-Link-Library

DOE Department of Energy

DOF Degree of Freedom

ESS Extreme Sea State

ETM Extreme Turbulence Model

EWH Extreme Wave Height

EWM Extreme Wind Speed Model

FAST Fatigue, Aerodynamics, Structures, and Turbulence

GDW Generalized Dynamic Wake

HAWT Horizontal Axis Wind Turbine

IEC International Electrotechnical Commission

JONSWAP Joint North Sea Wave Project

MSL Mean Sea Level

NL NonLinear

NREL National Renewable Energy Laboratory

NSS Normal Sea State

NTM Normal Turbulence Model

NWH Normal Wave Height

NWP Normal Wind Profile

OWT Offshore Wind Turbine

PM Pierson-Moskowitz

RF Rigid Foundation

RNA Rotor and Nacelle Assembly

SSS Severe Sea State

SWH Severe Wave Height

UCLS UnCoupled Linear Springs

VAWT Vertical Axis Wind Turbine

Chapter 1

Introduction

Over the last years, offshore wind turbines have been installed at increasing water depths. This implies larger environmental loads, which again calls for stronger, more advanced support structures. Still, roughly 80% of offshore wind turbines in Europe are supported by the traditional monopile substructures [40]. One reason for this preference for monopiles is their simplicity when it comes to installation. Another reason is the proven success of piles for support in offshore oil and gas infrastructures, despite the geometric and practical differences between these two types of foundations [24].

The accuracy of a wind turbine simulation is dependent on, among other factors, how realistically the modeled foundation represents the true soil-structure interaction. Modeling the flexibility of the foundation is a shortcoming in most softwares for simulations of OWTs today, according to Passon and Kühn [31]. There is a lack of simplified models that provide sufficient accuracies for detailed design of monopiles. This absence of appropriate foundation models may lead to under-predicted or over-predicted fatigue loads. The purpose of this thesis is to improve the modeling of soil-structure interaction of bottom-mounted wind turbines by introducing a nonlinear foundation representation in an already existing software. Specifically, the aero-hydro-servo-elastic simulation tool *FAST v7* is modified, with a monopile-supported OWT in mind. *NREL* (National Renewable Energy Laboratory) has developed a fictitious 5MW reference wind turbine, together with its properties contained in input files appropriate for FAST v7.

This reference turbine will be used as a base model in the following, and will later be addressed as *NREL 5MW*. A nonlinear foundation is modeled by means of parallel elastic perfectly-plastic springs attached to the monopile at the mudline. This method was proposed by [12]. In addition, constant foundation damping is introduced.

Chapter 2

Offshore Wind Turbines

In this chapter, the history and status of OWTs is presented. Terminology is also presented. Additionally, topics like wind power generation, natural frequencies, and fatigue is presented.

2.1 History and Current Status

Human exploitation of wind energy goes far back in time. Yet, with the rise of the oil and gas industry, wind energy lost its market to this more efficient competitor. However, the world's politicians have increasingly agreed on the need for a shift towards more renewable energy, and wind energy has again obtained attention. The wind energy industry has developed and grown remarkably the past years. Being renewable and vastly spread across the planet, it is an important aid when faced with the world's increasing need for energy.

Some of the earliest cases where wind power was used for mechanical work were developed in Persia in the 10th century. From the 600s to the 1890s, windmills existed in a magnitude of hundreds of thousands in the northwestern Europe. They were used as mechanical drives for milling or grinding of grains, and for pumping water. One of the first windmills that was used to drive an electric generator, and thereby qualify as

an actual wind turbine, was made in 1888, in Cleveland, Ohio [25]. As years past, basic developments were achieved in fields such as aerodynamics. Wind turbines were then given less priority from around the 1930s due to cheaper energy from oil and gas. In 1973, however, wind turbines experienced a commercial breakthrough caused by the combination of energy crisis and environmental problems [34].

Since winds are stronger and more stable at sea than onshore, an increased focus on installing wind turbines offshore has been seen. In the 1990s, the world's first offshore wind turbine farm, *Vindeby*, was built in Denmark. The offshore wind turbine industry has increased significantly since then. About 90% of the world's OWTs are installed in Europe, with the UK as the leading country, according to RenewableUK [33]. Figure 2.1 shows the distribution of installed capacity (MW) in Europe by 2014, as given by The European Wind Energy Association [40]. European OWTs are mainly installed in the North Sea, namely 63.3%. However, 22.5% are installed in the Atlantic Ocean and 14.2% in the Baltic Sea. On average the size of a European OWT is about 4 MW, and this number is by the The European Wind Energy Association [40] not expected to increase significantly in the near future.

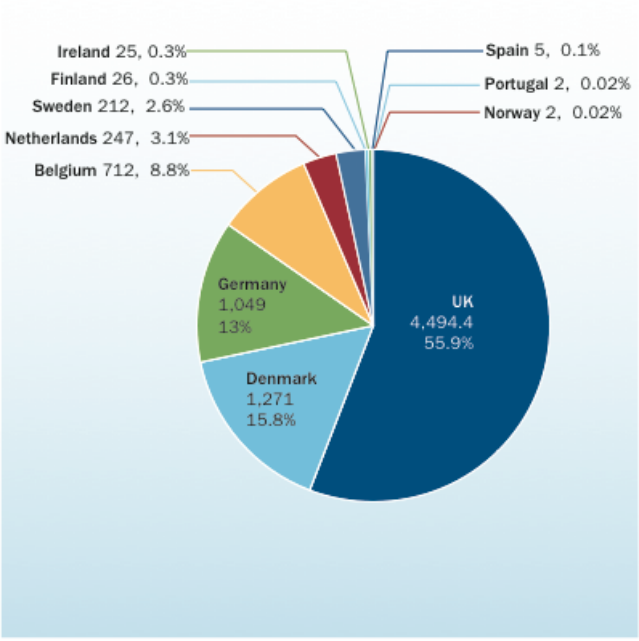

Figure 2.1: Installed capacity of OWTs in Europe. Cumulative share by country [40].

With the stronger winds come an increased potential for electricity production, but combined with large wave loads it also leads to new technological challenges. As the technology develops, turbines are expected to be installed further away from land, and at larger depths. By 2012, the average water depth for wind farms was about 22 m [42], covering a range of 2-45 m. While monopiles are the most popular support structures, for larger water depths other solutions may be applied, see Figure 2.2. Larger depths and stronger wind will generally lead to a larger load on the support structure.

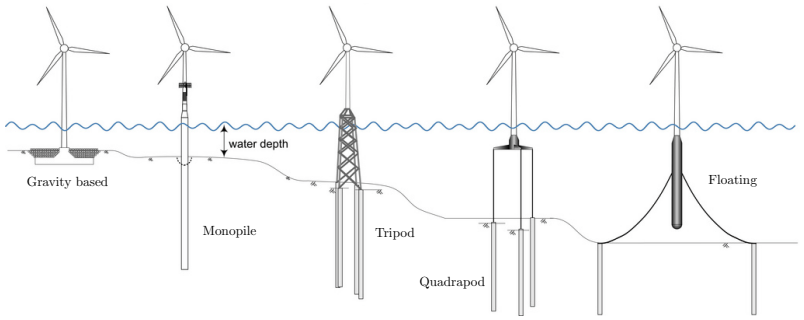

Figure 2.2: Support structures for offshore wind turbines [43].

2.2 Terminology

In this section a brief introduction to wind turbine terminology will be given. It will cover expressions that will be necessary to fully exploit this thesis.

During the development of wind turbines, both *vertical axis wind turbines* (VAWTs) and *horizontal axis wind turbines* (HAWTs) have been tried out, see Figure 2.3. Whereas HAWTs are well-known to most, the principle of VAWTs is not as familiar. HAWTs utilize lift, while VAWTs make use of drag. The blades of a VAWT are attached to a central vertical shaft. An alternator is positioned at the bottom of the shaft. When the blades rotate, the generator spins and produces electricity.

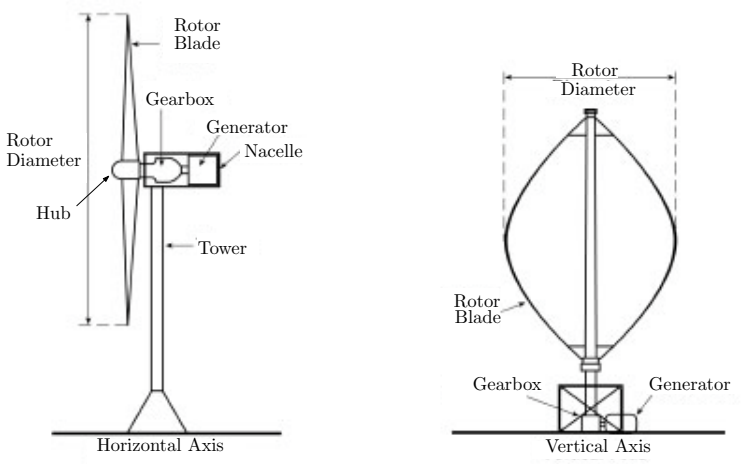

Figure 2.3: Example of a HAWT (left) and a VAWT (right) [41].

Today, the vast majority of wind turbines are two- or three-bladed HAWTs. In short they consist of a tower, hub, rotor, nacelle, see Figure 2.3. In addition, the substructure extends from the tower base, some meters above the *mean sea level* (MSL), down to the mudline, whereas the foundation continues below the mudline. Different types of substructures and foundations will be presented in Chapter 3. The nacelle works as support for the rotor, in addition to be housing a gearbox and an electrical generator. The *rotor and nacelle assembly* (RNA) constitutes everything carried by the tower. It comprises all physical parts such as blades, drive train and generator, as well as non-physical parts such as the control algorithms.

HAWTs can be characterized by whether their rotor is faced towards the wind (upwind) or away from the wind (downwind), see Figure 2.4. However, upwind is the most commonly applied configuration.

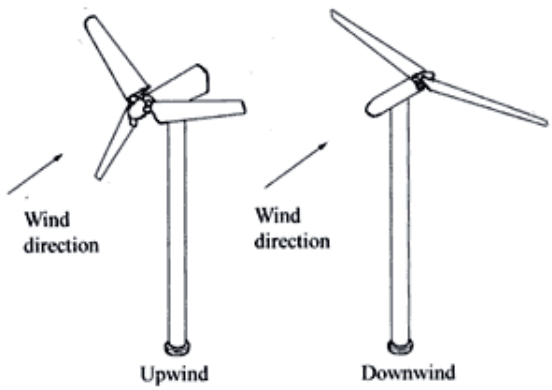

Figure 2.4: Upwind and downwind wind turbine.

Moreover, a generally accepted nomenclature regarding coordinate system, movements and rotations is presented in Figure 2.5. These *degrees of freedom* (DOFs) will continuously be used throughout the thesis. Other words used for surge and sway are *fore-aft* and *side-to-side*, respectively.

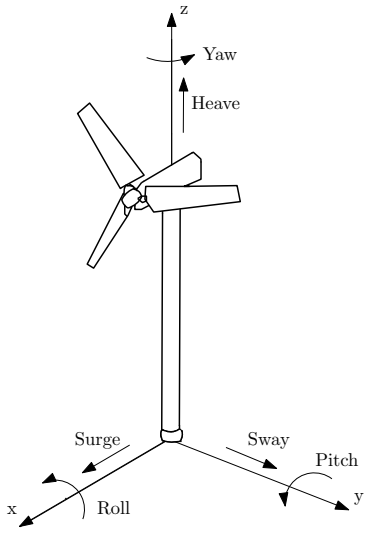

Figure 2.5: Coordinate system.

Many turbines utilize *active yaw*, meaning that the rotor is continuously oriented to face the wind.

2.3 Wind Turbine Power Generation

For a steady airstream, the power collected per square meter of the rotor is equal to:

$$P_{wind} = 0.5\rho_{air} AU^3 \tag{2.1}$$

where ρ_{air} is the density of air, A is the area intercepted by the rotor and U is the wind speed. The power related to the area intercepted by the rotor is not the amount actually extracted by a turbine rotor. Large modern turbines will typically capture up to about 50% of the wind power they are undergoing [25]. The turbine is designed for a certain range of wind speeds, with gears leading to an upper limit of rotor velocity, *cut-out wind speed*, corresponding to a maximum power output. Above this maximum rotating speed, the wind loads are considered too large, and could possibly lead to damage of the turbine if it is not shut down.

2.4 Natural Frequencies of Wind Turbines

Offshore wind turbines with monopile support structures are sensitive regarding dynamics because their full-system natural frequencies lie close to exciting frequencies of the environmental loads that act on them [24].

The most important natural frequencies of the wind turbine are the first tower bending frequencies. These natural frequencies are usually closest to the excitation frequency of the wind, wave and rotor [44]. If they coincide, dynamic amplification and large-amplitude stress variations will occur in the structure, leading to accelerated accumulation of fatigue [1]. In general, wind turbines have natural frequencies that relies on their operational states. A three-bladed HAWT will have operating natural frequencies of 1P and 3P. 1P is associated with the cyclic loading generated by mass imbalance. 3P is associated with wind deficiency generated loads that occur at the frequency of which the blades are passing the tower. Wave frequencies are often close to the rotor frequency, 1P. To avoid resonance with the wave frequency and the operating frequencies, the following three tower design approaches are proposed [1] and [2]:

- Soft-soft design. Tower frequency less than 1P and dominant wave frequencies.
- Soft-stiff design. Tower frequency lies between 1P and 3P.
- Stiff-stiff design. Tower frequency larger than 3P.

Figure 2.6 shows a simplified plot of the power spectral density of excitations of a typical three-bladed 3.6 MW OWT, having an operational interval of 5-13 rpm. The concept of power spectral density will be explained in Chapters 4 and 5. However, in short, it represents energy distribution of the different loads over different frequencies.

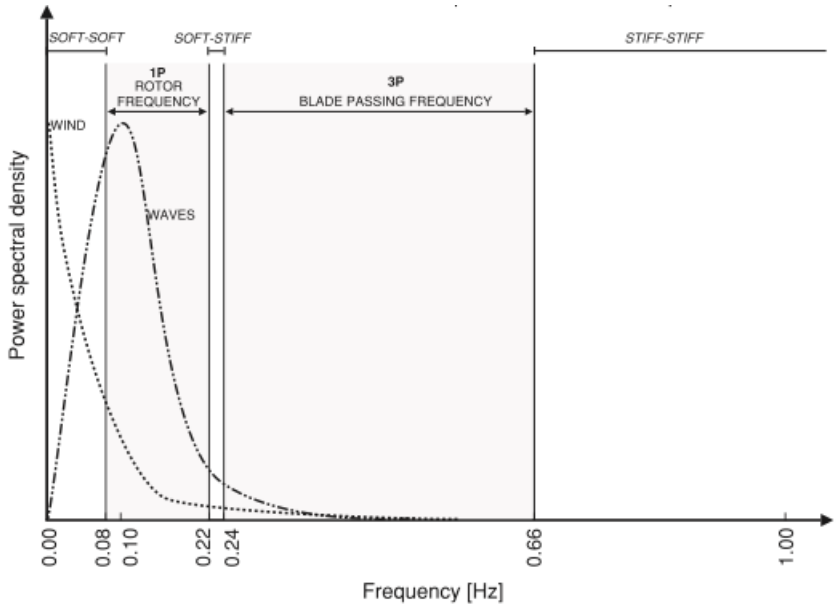

Figure 2.6: Simplified power spectral density of a typical OWT [24].

2.5 Fatigue in Wind Turbines

Today's wind turbines are designed for a lifetime of 20 years. In short, the lifetime of a structure considers how long the structure can be used while still remaining economically acceptable. For offshore wind turbines in high seas, up to almost 4000 full-load hours is expected per year [34]. The wind causes a significant bending moment on the base. On the other hand, the contribution from self-weight in stress calculation is small for a wind turbine compared to other structures.

A large portion of mechanical failures in metal structures are results of fatigue. Fatigue is failure in materials that occurs when the structure is subject to cyclic loading. A structure with a high static strength does not necessarily possess sufficient fatigue durability. To make sure the wind turbine can withstand the fatigue loads generated over its complete lifespan, analysis of these loads on the whole system is necessary.

Miner's rule, also known as the *Palmgren-Miner linear damage rule*, is an acknowledged measure of fatigue in wind turbines [38]. The damage rule is given as

$$D = \sum_{i=1}^{j} \frac{n_i}{N_i} \leq 1 \tag{2.2}$$

where D is the damage experienced by a structure that is subjected to j full stress-strain cycles, i.e. hysteresis loops. N_i is the number of cycles to failure at the i^{th} stress level σ_i, whereas n_i is the experienced number of stress cycles at the same stress level. According to Miner's rule, when D equals one, structural failure will occur [8]. The subject of load-displacement hysteresis loops is one that will be addressed later on in this thesis, in Section 3.3.

Chapter 3

Foundation Flexibility

Harnessing wind energy covers a wide range of engineering fields, from aerodynamics and structural dynamics to electrical generators and grid networks. Historically, modeling of foundation flexibility related to wind turbines has come in the shadow of other, more obvious aspects, such as aerodynamic efficiency, electricity generation, structural dynamics, and hydrodynamics. Currently, most wind turbine modeling tools are competent when it comes to computing aerodynamic and hydrodynamic loads, while soil-structure interactions are only considered in a very simplified manner based on results from external geotechnical results. According to Passon and Kühn [31], soil-pile interaction modeling for piled foundations is one of the most critical aspects regarding modeling of OWT substructures. This chapter presents support structure concepts, different ways of modeling foundation, and damping.

3.1 Support Structure Concepts

The traditional type of foundation for OWTs is the monopile; the support structure connects to a single pile foundation extending to a certain depth below the mudline, see Figures 3.1 and 3.2. For larger water depths, more complex structures such as jackets, tripods and trilets should be considered.

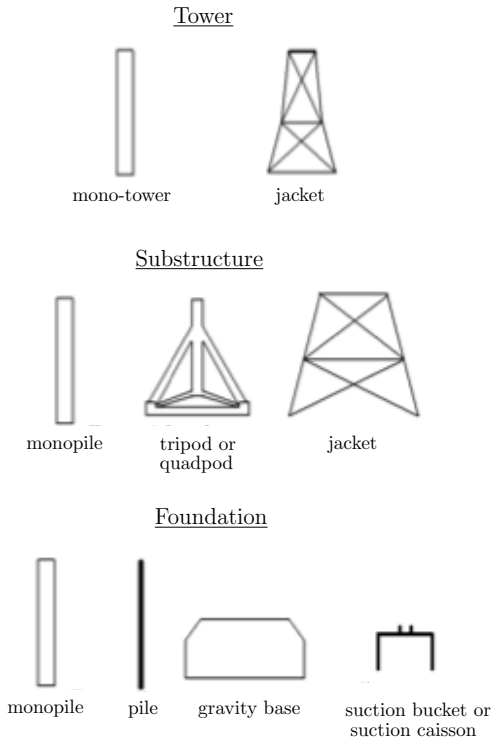

Figure 3.1: Support structure concepts for bottom-fixed offshore wind turbines [31].

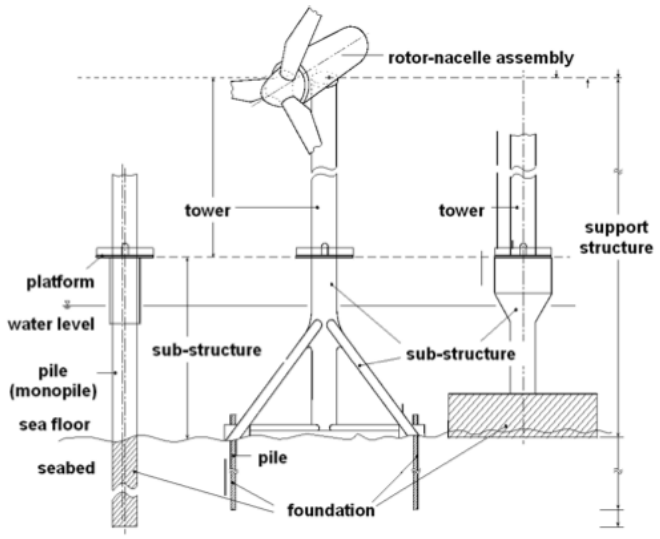

Figure 3.2: Terminology for the offshore support structure [31].

The difference between piles and monopiles is that while the monopile is used by itself and therefore is mainly laterally loaded, piles are arranged in a group and will additionally undergo significant axial loads.

3.2 Foundation Models

Pile foundations use lateral loading of the soil to withstand the loads induced in the supported structure [21]. FEM-modeling of the soil as a continuum may be the most correct way to describe the soil-pile system, but these models are very time consuming [31]. There are several methods for representing piled foundations in a simplified manner and some of them are illustrated in Figure 3.3. They are all different methods of representing the equivalent foundation model.

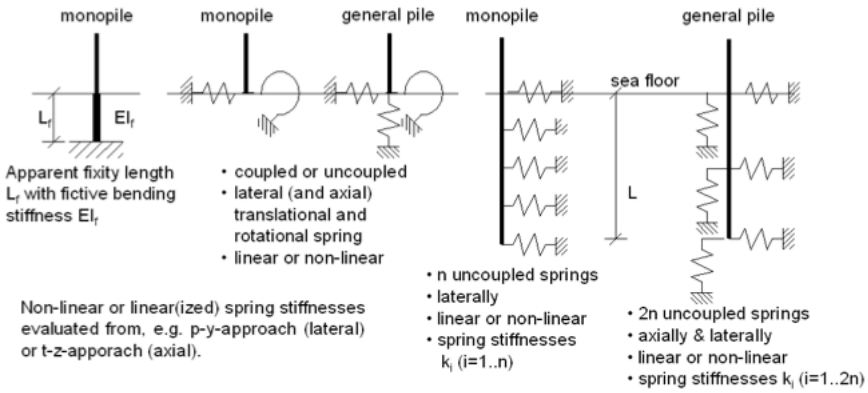

Figure 3.3: Typical models of piled foundations. Apparent fixity, single springs, and distributed springs [31].

A *fixed-base model* is the simplest model. Involving a rigid connection of the support structure to the sea bed, neither lateral nor rocking movements are allowed at the mudline. With this model, the soil profile at the turbine site is less critical in the turbine response. Both in FAST v7 and FAST v8, this representation is default.

In FAST v8, one can only model a rigidly connected monopile. However, an *apparent fixity* (AF) approach may typically be applied to deal with this issue. This is done by extending the monopile to a certain point below the mudline, where it is fixed, consti-

tuting a cantilevered beam. This length of extension is what is derived as the *apparent fixity length* for the cantilever. It makes sure that the fictive beam has the same stiffness as the actual pile-soil system, and reproduce the appropriate rotations and translations at the mudline. Specified levels of shear force and moment at the mudline are selected and used to calculate corresponding lateral displacements and rotations according to the soil profile, penetration depth and pile dimensions. From these data, the apparent fixity length and the flexural rigidity (EI) can be derived. The mass distribution is usually kept the same as for the monopile above the mudline. The latter estimation may be considered reasonable since the movement of the structure sub-soil is small compared to the movement of the tower top, so that inertial effects are assumed to be negligible [3]. If any other nonlinear soil model was to be applied, it would take a sizeable change to the FAST v8 source code to make it possible.

Letting the monopile end at the mudline, attached to lateral rotational and translational springs at the bottom, is another way of modeling the foundation response. The springs may be *coupled* or *uncoupled*, *linear* or *nonlinear*. A user-subroutine for implementing coupled linear springs like these in FAST v7 already exists. The physical meaning of coupled springs is that due to a lateral load, not only a lateral deflection, but also a rotation will occur, and vice versa. While coupled springs will be more accurate then uncoupled springs, coupled springs provide a stiffness matrix with terms located off the diagonal. A coupled stiffness matrix will in general make calculations more complicated.

An option that provides an equivalent representation of the loads to that of the coupled springs, while avoiding off-diagonal terms, is through the use of a *rigid link* [4]. By adding a rigid element to extend the pile to some length below the actual mudline, and applying the springs here instead, the loads at the mudline can be equivalently represented through a diagonal stiffness matrix. To illustrate, consider the two-dimensional model in Figure 3.4.

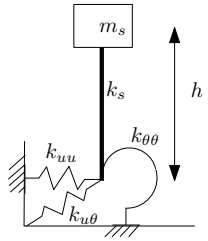

Figure 3.4: Beam with coupled springs foundation representation

The following coupled stiffness matrix representing forces applied by the coupled springs at the mudline:

$$\begin{bmatrix} k_{uu} & k_{u\theta} \\ k_{\theta u} & k_{\theta\theta} \end{bmatrix} \begin{bmatrix} u \\ \theta \end{bmatrix} = \begin{bmatrix} F \\ M \end{bmatrix} \tag{3.1}$$

k_{uu}, $k_{u\theta}$, $k_{\theta u}$ and $k_{\theta\theta}$ represent the stiffness of the springs in the figure, whereas F, M, u and θ are force, moment, translation and rotation in the springs. A proper length L of the applied rigid link is then given by Equation (3.2).

$$L = \frac{-k_{u\theta}}{k_{\theta\theta}} \tag{3.2}$$

A new, equivalent uncoupled stiffness matrix representing the uncoupled springs at the end of the rigid link is given as

$$\begin{bmatrix} k_{uu}{}' & 0 \\ 0 & k_{\theta\theta}{}' \end{bmatrix} \begin{bmatrix} u' \\ \theta' \end{bmatrix} = \begin{bmatrix} F' \\ M' \end{bmatrix} \tag{3.3}$$

where

$$k_{uu}{}' = k_{uu} \tag{3.4}$$

and

$$k_{\theta\theta}{}' = k_{\theta\theta} - L^2 k_{uu} \tag{3.5}$$

The equivalent model is illustrated in Figure 3.5

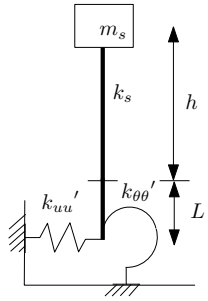

Figure 3.5: Rigid link approach with uncoupled springs.

Distributing a discrete number of uncoupled translational springs along the length of the sub-soil part of the pile is another common representation, which was used by for example Schløer [35]. The springs may be linear or nonlinear. The stiffness of the springs should be varied along the pile so that a large participation of the soil-structure interaction model in the dynamic response is fulfilled by the upper layer, while the lower layer ensures a proper overall stiffness of the foundation [30].

Soil experiencing loading will not act linearly elastic. When the load increases, the stiffness decreases. Nonlinear or linearized lateral spring stiffness may be evaluated using the *p-y-approach*. P-y curves describe the nonlinear elastic relationship between the integral value p of the mobilized resistance from the surrounding soil when the pile exhibits a lateral displacement y [8]. Applying the distributed springs approach in combination with p-y curves for the nonlinear springs is a common representation.

3.3 Damping

In recent times, OWT tower height and blade lengths have increased, without a corresponding increase in weight. Installation in deeper waters is one of the reasons for this. This moves the dynamic response of the wind turbine structure into a range of frequencies that is closer to that of wind and waves, and closer to the operating frequency. Therefore damping is important, to prevent large oscillation amplitudes [6].

Bottom-fixed OWTs experience damping in different forms. These main types of damping are aerodynamic damping, soil damping, hydrodynamic damping and structural

damping, in addition to tower dampers. Of these, the largest contributor is aerodynamic damping. During standstill, or when the wind turbine is idling, the aeroelastic damping becomes insignificant and the other types of damping therefore important. Next to aerodynamic damping, soil damping gives the largest contribution to overall damping, according to Schløer [35]. Soil damping results in energy dissipation and alters the cyclic response of the structure. The investigation of this influence on the structural response of OWTs is scarce [27].

Soil damping mainly exists in two types, namely *radiation damping* and *material damping*. The former concerns dissipation of energy by radiation of waves towards infinity, whereas the latter represents energy loss due to inelastic behavior of the material. Radiation damping may be neglected for frequencies below 1 Hz [39]. Since the environmental load frequencies normally lie below this frequency [24], together with operating frequencies and the first natural frequencies of the tower, material damping is the type of soil damping that will be examined here. When a structure or foundation with material damping is subjected to cyclic loading, the load-displacement curve forms a hysteresis loop. Therefore, material damping may also be referred to as *hysteretic damping*. The energy dissipated during one deformation cycle is the area under the hysteretic loop and is a consequence of inelastic behavior, see Figure 3.6. F represents either a moment or a force, whereas δ is rotation or translation.

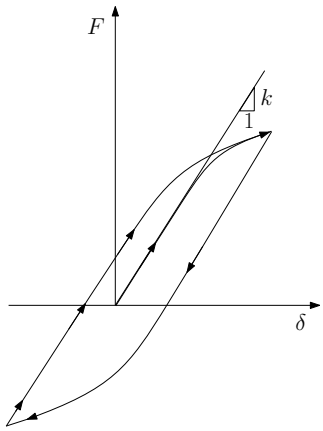

Figure 3.6: Load-displacement hysteresis loop. The area enclosed by the loop represents energy loss due to damping.

Hence, for a nonlinear relationship between load and displacement in a foundation represented by springs, damping is incorporated in the foundation by means of the hysteresis loop. Including soil damping in simulations is interesting because it may reduce the design loads for the tower, leading to a decreased amount of required material [4]. Moreover, the numerous cycles of hysteresis loops that a nonlinear foundation will experience, provide information about fatigue damage, as described in Section 2.5.

Chapter 4

Wind Modeling

In this chapter, a brief description of features that are important for wind modeling will be given.

4.1 The Nature of Wind

Wind speed and direction is highly variable. Fluctuations in wind velocity are present at a large range of time-scales, from seconds to years. First, wind experiences inter-annual changes, meaning that wind resources at a certain location will vary from year to year. An inter-annual variation of about 5% is not uncommon [25]. Therefore, basing long-term prediction of wind for a site on wind speed measurements from one single year, will lead to less accurate forecasts. Also, climate changes may introduce deviations from past measurements. Moreover, there will be seasonal variations of the wind. These are important when assessing how power output from wind turbines can meet seasonal electricity demand. Depending greatly on the geography of the site, distance to the sea, and altitude, daily (*diurnal*) changes in the wind power may also be significant. Usually, winds are weaker during the night. Finally, random fluctuations in wind velocity over time periods shorter than about 10 minutes are present. These short-term variations are known as *turbulence*.

4.2 Wind Statistics

To represent the fluctuating nature of the wind climate, a short term (i.e. over a 10-minute period) stationary wind condition may be assumed. It is governed by a constant mean wind speed, U_{10} and a constant standard deviation of the wind speed, σ_U. The random variation of the wind speed under these stationary conditions will follow a probability distribution described by the above two parameters. From measurements, it is observed that the distribution of wind velocity components in the *atmospheric boundary layer*, see Section 4.3.1, fits with a Gaussian probability density function. As a basis for describing long-term and short-term wind conditions, wind speed statistics of a period of, preferably, 10 years or more should be used [8].

According to Det Norske Veritas [8], it is customary to distinguish between normal wind conditions, concerning "recurrent structural loading conditions", and extreme wind conditions, which "represent rare external design conditions". The former is primarily basis for fatigue loads calculation, and also extreme loads through extrapolation of normal operation loads. Extreme wind conditions can lead to extreme loads in the wind turbine and its foundation. For an offshore wind turbine, parameters used to specify normal wind conditions are air density, a long-term distribution of the 10-minute mean wind speed, a *wind shear* level, and turbulence. The wind shear is given as the rate of change of mean wind speed with height [10].

While the probability distribution describes the mean wind speed and the deviation of the wind speed, there is still a lack of information regarding the pace at which these variations happens. Through the *spectral density* (spectrum) of the wind, the distribution of turbulence with frequency can be described. Hence, the short term 10-minute stationary wind climates described here may be represented by a wind spectrum, see Section 4.3.3.

4.2.1 Reference Wind Conditions and Reference Wind Speeds

For load combinations for design, a number of reference wind conditions and wind speeds are defined.

While the *Normal Wind Profile* (NWP) describes how the mean wind speed varies with height above sea level, the *Normal Turbulence Model* (NTM) represents turbulent wind speed. The latter representation is done through a characteristic standard deviation of the wind speed $\sigma_{U,c}$, which by definition is the 90% quantile in the probability distribution of the standard deviation σ_U of the wind speed conditioned on the 10-minute mean wind speed at the hub height.

Other models that may be appropriate are the *Extreme Wind Speed Model* (EWM) and the *Extreme Turbulence Model* (ETM). The former is applied to describe extreme wind conditions with a specific return period (e.g. 1 year or 50 years), either with a steady or turbulent wind model. The latter combines the NPM with a turbulent wind speed.

4.3 Wind Representation

4.3.1 Mean Wind Speed Profiles

A mean wind speed profile, or height profile, is a representation of the time-averaged wind speed variation with height above ground or seawater level.

Close to the surface of the earth, the air flow will be retarded by frictional effects. This region of wind flow affected by friction is known as the *atmospheric boundary layer*. Here, the wind velocity depends on the distance from the ground. Beyond a certain height above the surface, the *gradient height*, the wind speed is no longer affected by ground, or ocean, obstruction. The corresponding, fully developed wind speed is called *gradient wind speed*. Due to the Coriolis force, the mean wind speed may also slightly change direction with height. However, this change of direction is small relative the the height range of normal structures, and can therefore normally be neglected [10].

One way of representing the wind profile is by an approximation through the *power law* [8]:

$$U(z) = U_{10}(H)\left(\frac{z}{H}\right)^{\alpha} \tag{4.1}$$

Here, U is the mean wind speed at height z. U_{10} is the 10-minute mean wind speed, occurring at the reference height H. When modeling wind for wind turbines, the reference height should be at the wind turbine hub height [8]. The power law exponent, α, is dependent on the *surface roughness* of the landscape below. Landscapes with trees or buildings will have a higher roughness surface than a snow-covered flat, and therefore a larger α. The roughness across the ocean surface, despite waves, is very small [34], contributing to high wind velocities over the ocean. The power law is easy to integrate over height. This is suitable when for example bending moments at the base of a structure are to be determined.

The *logarithmic law* is another wind profile type commonly used. In strong wind conditions, it is the most accurate mathematical expression [10] and it applies to turbulent boundary layers on both a small and large scale [34]. The logarithmic law states that

$$U(z) \propto \ln\left(\frac{z}{z_0}\right) \tag{4.2}$$

where z_0 is known as the *roughness length*. For offshore locations it depends on the wind speed, the upstream distance to land, water depth and wave field, and typically varies between 0.0001 m in open sea without waves, and 0.01 m in coastal areas with onshore wind [7].

Some drawbacks are related to the logarithmic law. It cannot be evaluated for negative z's, e.g. below the zero-plane, and it is difficult to integrate. For that reason, wind engineers have commonly chosen to use the power law.

The two presented laws describe very similar profiles [34]. In Figure 4.1 the logarithmic law and power law are presented in the same graph, with $z_0 = 0.02$ m and $\alpha = 0.128$. Due to the similarities, the power law is in general recognized as adequate for engineering purposes [10].

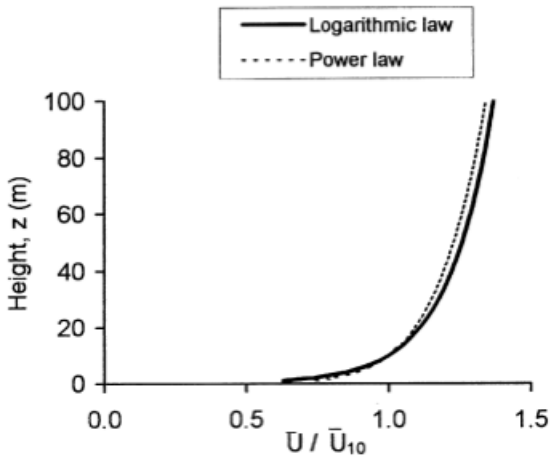

Figure 4.1: Comparison of the logarithmic law and the power law, as presented in Holmes [10].

4.3.2 Turbulence

It is useful to consider the wind speed as a combination of two components: the mean wind speed, and a fluctuating part. The latter is what is known as turbulence, see Figure 4.2. U is the mean wind speed, whereas u, v and w are fluctuating components. In general, most flows observed in engineering applications are turbulent. In wind, it is caused by convective movement in the atmosphere, friction due to ground roughness, or both. Turbulence will increase with higher ground roughness and decrease with distance from the ground.

As already mentioned, a fundamental characteristic of turbulent flows is significant, irregular variation of the fluid velocity, both in time and space. Periodicity will not be distinct, and fluctuations will be characterized by randomness. However, the velocity and its mean will in some sense be "stationary", as huge changes in the velocity u(t) will not occur over a shorter timescale. Neither will the velocity exhibit large deviations from the mean velocity U_{10} for longer time periods [32]. This "steadiness" can be explained by the fact that generation of the mean flow occurs at time scales much larger than 10 minutes, or 1 hour for that sake. Choosing an averaging time period of between 1 hour and 10 minutes provides quite stable mean values [36]. However, over longer time periods, some variation in the mean velocity will occur. It is common to distinguish

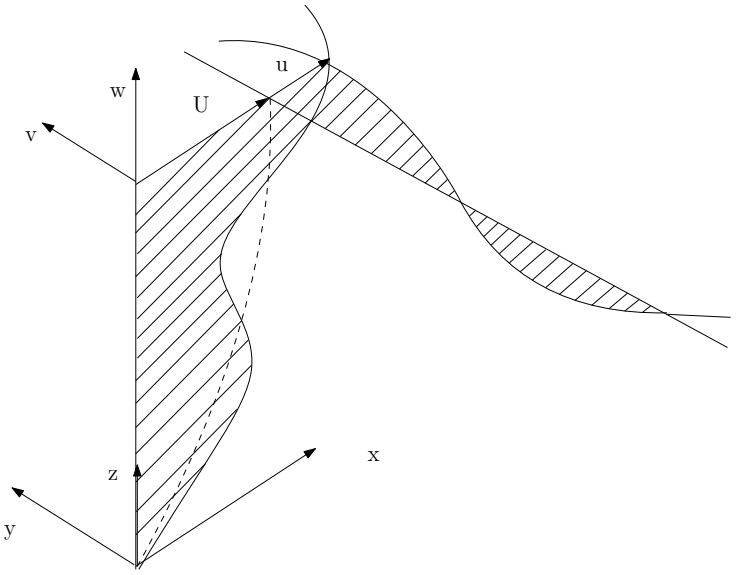

Figure 4.2: Turbulent wind field [37].

between small-scale turbulence, and the large-scale motions in turbulent flows [32]. While the large-scale motions are largely dependent on the boundary conditions, i.e. geometry, of the flow, the small-scale motions have a universal character, which is not influenced by the flow geometry. The random variations of the wind speed are created by unique eddies or vortices in the air flow travelling along at mean wind speed.

A convenient parameter describing a turbulent wind field is the *turbulence intensity*. At height z in along wind direction it is expressed as

$$I_u(z) = \frac{\sigma_U(z)}{U_{10}(z)} \tag{4.3}$$

Lateral and vertical turbulence components will in general be smaller than the corresponding longitudinal value [10].

4.3.3 Wind Spectra

A way of representing the short term, stationary wind conditions is through a wind spectrum. From such a spectrum, software like TurbSim, which will be presented more

thoroughly in Chapter 6, can generate time series.

The power spectral density function of the wind speed process, $S(f)$, shows the distribution of wind energy between various frequencies. $S(f)$ is expressed in terms of U_{10} and σ_U. Such spectral densities of the wind speed process can be represented for specific sites through available measured wind data. A spectral density model may be applied. TurbSim offers models such as the *Kaimal spectrum*, the *Von Karman* model, the *Risø Smooth-Terrain* Model, and NREL's own *National Wind Technology Center* Model (NWTCUP). According to Det Norske Veritas [8], unless data indicate otherwise, the spectral density of the wind speed process is appropriately represented by the Kaimal spectrum. For a given wind component this may be given as,

$$S_U(f) = \sigma_U^2 \frac{4\frac{L_k}{U_{10}}}{(1+6\frac{fL_k}{U_{10}})^{5/3}} \tag{4.4}$$

where f is the cyclic frequency and L_k is an integral scale parameter. For design of wind turbine generators, L_k may be independent of terrain roughness [7] and taken as

$$L_k = \begin{cases} 5.67z \text{ for } z < 60 \text{ m} \\ 340.2 \text{ m for } z \geq 60 \text{ m} \end{cases} \tag{4.5}$$

where z denotes height above seawater level. The different spectra will in general match in the high frequency range, but may differ largely in the low frequency range. Moreover, most spectra are calibrated for use with wind data measured over land, whereas only a few are fit for wind data obtained over water.

4.3.4 Correlation, Co-spectrum and Coherence. Atmospheric Stability

When considering wind loads and wind turbulence, *covariance* and *correlation* are important characteristics. In this case, the covariance is the time-averaged product of the variation of the velocities at two different points in space. The correlation coefficient ρ is then defined as the covariance divided by the product of the standard deviation in the

velocities at the two points. The value of ρ must lie between -1 and +1. If ρ is 0, there is no correlation between the velocity in the two points. If the correlation coefficient is +1, there is full correlation, and if it is -1, there is perfect negative correlation. The latter implies that if one of them increases, the other decreases equally. These two terms become useful when calculating the varying loads on structures with a large span [10].

Another important aspect is the correlation of velocity fluctuations at two separated points at a given frequency, for example the natural frequency of the structure at which the wind is acting. The frequency-dependence of the correlation can be expressed in terms of the *cross-spectral density* (cross-spectrum), *co-spectral density* (co-spectrum) and *coherence*, which are all functions of frequency. The cross-spectrum consists of a real and imaginary part. The real part is what is called the co-spectral density, and may be interpreted as a frequency-dependent covariance. Coherence, on the other hand, is obtained by normalizing the cross-spectrum, so that only the real part is included. Coherence can more or less be regarded as a frequency-dependent cross-correlation coefficient.

Chapter 5

Wave Modeling

In this chapter, an introduction to wave theory, wave representation and wave statistics is given. Waves are by nature irregular and arbitrary when it comes to shape, height, length and propagation speed. Hence, the best way of representing a sea state is through a random wave model. A *linear* irregular wave model is produced through superposition of several wave components, each with a different amplitude A, frequency f, and direction. On the contrary, a *nonlinear* wave model accounts for nonlinear interaction between the wave components. The latter is necessary when the waves are no longer small compared to the water depth [31]. Wave theory is developed to express the relationship between the wave period T and the wave length λ, and the motion of fluid particles in the water.

5.1 Wave Theory and Kinematics

Wave kinematics can be represented in several ways. One of them, the simplest one, is *Airy linear wave theory*. It is suitable for both deep waters and shallow waters, but assumes that the wave height is small compared to both the wave length and the water depth. According to Det Norske Veritas [8], the elevation of the water surface η is for regular waves then given by

$$\eta(x, y, t) = \frac{H}{2}\cos\Theta \tag{5.1}$$

where $\Theta = k(x\cos\beta + y\sin\beta) - \omega t)$ is the phase of the wave and β is the direction of propagation. H is the wave height, see Figure 5.1. It is equal to twice the amplitude A of the regular wave, $H = 2A$, and k is the wave number. By definition, the wave height H reaches from the lowest through to the highest crest in a wave cycle. The period T is the time period between two following zero-upcrossings of the sea elevation.

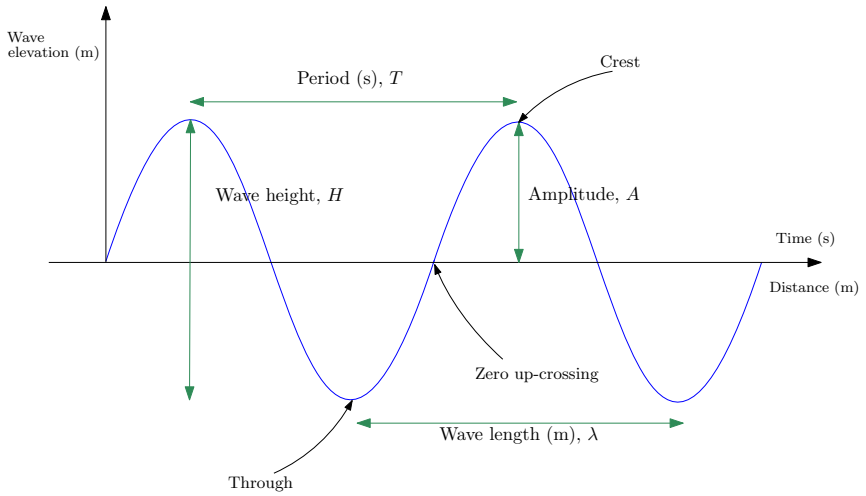

Figure 5.1: Wave description.

A regular wave is characterized by propagating with a permanent form, and having a certain wave length γ, period T and height H [8]. Moreover, the *dispersion relationship* relates the wave period T and the wavelength λ. The dispersion relationship for constant depth d is given in Equation (5.2).

$$\lambda = \frac{gT^2}{2\pi}\tanh\left(\frac{2\pi d}{\lambda}\right) \tag{5.2}$$

Other theories describing the kinematics of regular waves are *Stokes theory*, which is suitable for high waves and not applicable for particularly shallow waters, the *stream function theory*, which is accurate for a broad range of water depths, *Boussinesq higher-order theory* for shallow water, and *solitary wave theory* appropriate for very shallow

water.

According to Schløer [35], deep water is classified as $kH > \pi$ and shallow water as $kh < \frac{\pi}{10}$.

To obtain kinematics for irregular waves, superposition of kinematics from linear kinematics may be applied. The water surface elevation may then be given by

$$\eta(t) = \sum_{k=1}^{N} A_k \cos(\omega_k t + \varepsilon_k) \tag{5.3}$$

where ε_k are arbitrary phase angles with a uniform distribution between 0 and 2π, and A_k are random amplitudes with corresponding angular frequency ω_k.

Airy wave theory and Stokes wave theory provide wave kinematics below $z = 0$. However, to obtain predictions of the fluid velocity and acceleration, i.e. kinematics, above MSL, appropriate stretching or extrapolation methods of the kinematics profile to the wave surface should be applied [8].

5.1.1 Wave Stretching

Kinematic stretching is the process of extending linear Airy wave theory to provide predictions of fluid velocity and acceleration at points above the MSL [29]. The *Wheeler stretching method* is widely used, and offered in FAST. It accounts for the fact that the fluid velocity at the still water level is reduced compared to linear theory [7]. The velocity at the free surface is computed using linear theory, and for each time step in a time series, the vertical coordinate is stretched according to

$$z = \frac{z_s - \eta}{1 + \eta/d} \text{ for } \begin{cases} -d < z < 0 \\ -d < z_s < \eta \end{cases} \tag{5.4}$$

z_s is the stretched z-coordinate, η is the free surface elevation and d is the mean water depth. A sketch is given in Figure 5.2

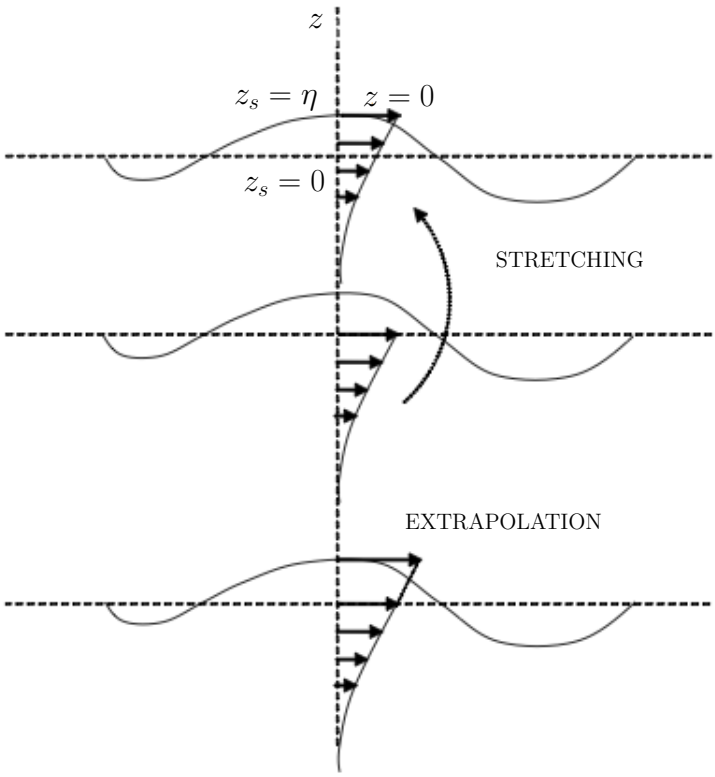

Figure 5.2: Stretching and extrapolation of velocity profile, as seen in Det Norske Veritas [7].

5.2 Wave Representation

The wave climate may be expressed in terms of the *significant wave height* H_S and the *spectral peak* T_p. The definition of the significant wave height is according to Det Norske Veritas [8] as follows: H_S is four times the standard deviation of the sea surface elevation. However, if the sea state only has a narrow band of frequencies, the significant wave height is approximately equal to the mean height of the highest third of the zero up-crossing waves, as stated in IEC 61400-3 [11]. It represents the intensity of the wave conditions and the fluctuations in the random wave heights. Over a short period of time, both parameters may be assumed to be constant, implying stationary wave conditions. These short time periods may be in the range of three to six hours [8]. During short-term stationary conditions the arbitrary wave height H will follow a probability distribution that is dependent on H_S. The wave period T will also follow

a probability distribution, however this distribution is a function of both H_S, T_p and H. Another important parameter is the *wave crest height* H_C. This is the highest crest between two successive zero-upcrossings of the sea elevation. H_C follows a probability distribution that is dependent on the significant wave height. The short-term sea conditions may be expressed in terms of a wave spectrum. *S(f)* is the power spectral density of the elevation of the ocean. It depends on H_S and T_p. The wave spectrum represents the energy distribution of the sea elevation at different frequencies.

5.2.1 Wave Spectrum

A sea state should be represented by a wave frequency spectrum and a corresponding wave height, a typical frequency, a mean propagation direction and a spreading function, according to Det Norske Veritas [7].

From available wave statistics, a wave spectrum for the specific site may be established. Both wind sea and swell should be considered. Wind seas are directly produced by local wind, whereas swell are waves that have traveled from the location at which they were generated. Swell is not dependent on local wind conditions. The most siginificant wave loads are the ones caused by waves that are induced by wind [31]. According to Det Norske Veritas [8], unless other is indicated by data, an appropriate representation of the sea state may be given by the *Joint North Sea Wave Project* (JONSWAP) spectrum:

$$S(f) = \frac{\alpha g^2}{(2\pi)^4} f^{-5} exp\left(-\frac{5}{4}\left(\frac{f}{f_p}\right)^{-4}\right)\gamma^{exp\left(-0.5\left(\frac{f-f_p}{\sigma f_p}\right)^2\right)} \tag{5.5}$$

where

$$\alpha = 5\left(\frac{H_S^2 f_p^4}{g^2}\right)(1-0.287\ln\gamma)\pi^4 \tag{5.6}$$

$f = 1/T$ is the wave frequency, $f_p = 1/T_p$ is the spectral peak frequency and g is the acceleration of gravity. γ is the *peak-enhancement factor*, and depends on the significant wave height and peak period. The spectral width parameter σ is given as

$$\sigma = \begin{cases} 0.07 \text{ for} f \leq f_p \\ 0.09 \text{ for } f > f_p \end{cases} \tag{5.7}$$

T_p, T, H_S are defined previously in this chapter.

When γ is 1, the JONSWAP spectrum is reduced to the *Pierson-Moskowitz* (PM) spectrum. For the same total energy of the spectra, the JONSWAP spectrum will in general have a higher and more narrow peak than the PM spectrum [17].

Both the Pierson-Moskowitz spectrum and the JONSWAP spectrum express wind sea conditions that are common for the most severe sea states [7]. The PM spectrum was initially developed for fully-developed sea, while the JONSWAP spectrum is an expansion of the former to account for sea states that are still developing. In Figure 5.3 the JONSWAP spectrum for different values of γ is presented as a function of frequency.

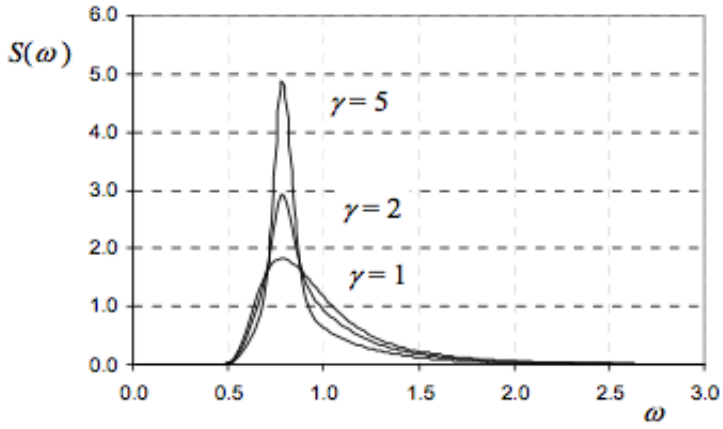

Figure 5.3: JONSWAP spectrum for H_S = 4.0 m, T_p = 8.0 s for γ = 1 (Pierson-Moskowitz spectrum), γ = 2 and γ = 5, as presented in [7].

If the wave conditions has a pronounced swell component, the JONSWAP spectrum may not be sufficient. Then a two-peaked spectrum, such as the *Torsethaugen* spectrum, can better express the wave climate.

5.3 Wave Statistics

As with wind statistics, wave statistics used as a basis for design should ideally cover a time period of 10 years or more [8]. This applies to both long-term and short-term conditions.

Wave and wind conditions are in general correlated. However, within a period of stationary wind and wave conditions, wind speeds and wave heights may be assumed to be independent and uncorrelated [8].

The probability distribution of the long-term wave climate parameters H_S and T_p can be expressed through a generic distribution, or through scattergrams. Scattergrams provide frequency of appearance of pairs of H_S and T_p. Combining a Weibull distribution for H_S with a lognormal distribution of T_p conditional on H_S generates a typical generic distribution representation of the wave climate. Det Norske Veritas [8] suggests the following Weibull distribution for the significant wave height.

$$F_{H_S}(h) = 1 - exp(-\left(\frac{h-\gamma}{\alpha}\right)^{\beta}) \tag{5.8}$$

Typically a deterministic approach for the wave description is applied to calculate extreme loads, while a stochastic approach is applied for determining fatigue load. Deterministic approaches are commonly preferred for extreme load calculations, while stochastic approaches are preferred for fatigue loads [31].

5.3.1 Reference Sea States and Reference Wave Heights

As for wind statistics, several reference sea states and -wave heights exist. The *Normal Sea State* (NSS) is described in terms of significant wave height, peak period, and wave direction [8]. Importantly, it is related to a simultaneous mean wind speed. The significant wave height, $H_{S,NSS}$ is then the expected value of the significant wave height conditioned on the corresponding 10-minute mean wind speed. This is also known as the *Normal Wave Height* (NWH)). The NSS is applied to calculate ultimate loads and fatigue loads. For the latter purpose, a sequence of normal sea states must be looked at,

associated with different mean wind speeds. Also, a range of appropriate peak periods should be considered for each significant wave height. Other reference sea states are the *Severe Sea State* (SSS) and the *Extreme Sea State* (ESS), while other reference wave heights are such as the *Severe Wave Height* (SWH) and the *Extreme Wave Height* (EWH).

Chapter 6

Introduction to FAST

FAST is a time-domain *computer-aided engineering* (CAE) tool for simulating dynamic response of horizontal-axis wind turbines, including both hydrodynamic and aerodynamic loads. It is developed by the United States *Department of Energy's* (DOE's) National Renewable Energy Laboratory. It models structural dynamics of the support structure, tower, rotor and nacelle, as well as control and electrical drive dynamics. Both land-based, offshore bottom-mounted and offshore floating wind turbines can be modeled, and they may be two- or three-bladed. As opposed to softwares like *Abaqus FEA* and *Seismostruct*, FAST does not exhibit a graphical user interface. All information given to the user and by the user is in the form of text files and scripts.

The latest version of this tool is FAST v8. Currently, however, FAST v8 can only model a bottom-fixed monopile with rigid connections between the substructure and the seabed. For modeling flexible foundation with nonlinear springs, the aim of this thesis, this is not suitable. Instead, FAST v7 is used. FAST v7 provides the possibility of substituting the rigid connection at the bottom of the monopile with linear springs in each dof. For monopiles, the advantages of FAST v8 are minor, while FAST v7 has several advantages, e.g. the option to include wave stretching in the hydrodynamic load calculation, which is important in shallow water, and the ability to make easy customizations for soil models. Therefore, when subsequently referring to FAST in this thesis, FAST v7 is the version that is meant.

6.1 FAST v7 Source Code

FAST is an open source software, written in the programming language of Fortran. In addition to the FAST source files, it also requires compilation with *NWTC Subroutine Library*, *AeroDyn* and *InFlowWind*. AeroDyn is an aerodynamics software module for use by designers of horizontal-axis wind turbines. It is written to be interfaced with a number of dynamics analysis software packages, such as FAST, ADAMS (Automatic Dynamic Analysis of Mechanical Systems), SIMPACK, and FEDEM, for aero-elastic analysis of wind turbine models. AeroDyn will be more thoroughly investigated in Section 6.5. The InflowWind module is used to return wind speeds at a given time and location requested from a calling program. The NWTC Subroutine Library consists of several general-purpose routines and their source files.

Fortran files contained in the FAST source, and their respective functions are listed below, see Table 6.1.

In addition, FAST includes "hooks" for 10 user-specified subroutines. Examples of these are contained in the file *UserSubs.f90*. The routines are listed in Table 6.2.

The dummy routines can be commented out and replaced with the user's own logic.

It is also possible to interface FAST with a master controller implemented as *dynamic-link-library* (DLL) in the style of Garrad Hassan's Bladed wind turbine software package. To simulate the NREL 5MW reference turbine, this must be implemented. It is then necessary to recompile FAST to include another source file, *BladedDLLInterface.f90*. This is described by Jonkman [13]. Certain parameter settings are also required, see Table 6.3 below.

Additionally, some subroutines must be commented out in *UserSubs.f90* and *UserVSCont_KP.f90*, which then are renamed to *UserSubs_forBladedDLL.f90* and *UserVSCont_KP_forBladedDLL.f90* [13].

Table 6.1: FAST source files [20].

Source File	Description
Fast_Prog.f90	Contains PROGRAM Fast(), which guides the program's execution.
Fast_Mods.f90	Contains MOUDLEs that store variables used by FAST's routines.
Fast_IO.f90	Contains routines related to program input and output.
Fast.f90	Contains routines that make up the "guts" of FAST, including the equations of motion and their solution.
AeroCalc.f90	Contains the interface routines between FAST and AeroDyn.
FAST2ADAMS.f90	Contains routines that make up the FAST-to-ADAMS preprocessor.
Fast_Lin.f90	Contains routines used during a linearization analysis.
SetVersion.f90	Contains a routine that sets the program version number.
GenUse.f90	Contains general-purpose routines.
NoiseMods.f90	Contains a MODULE that stores variables used by aeroacoustic routines.
NoiseSubs.f90	Contains routines related to aeroacoustics.
ModCVF.f90	Contains a MODULE that stores compiler-dependent variables.
SysCVF.f90	Contains compiler-dependent routines.
UserSubs.f90	Contains dummy placeholders of **all** available user-specified routines.
PitchCntrl_ACH.f90	Contains an example pitch control routine written by A. Craig Hansen.
UserVSCont_KP.f90	Contains an example variable-speed torque control routine written by Kirk Pierce.

Table 6.2: Optional user-specified subroutines [20].

Routine	Description
PitchCntrl()	User-specified blade pitch control (either independent or rotor-collective).
UserGen()	User-specified generator torque and power model.
UserHSSBr()	User-specified high-speed shaft brake model.
UserPtfmLd()	User-specified platform loading.
UserRFrl()	User-specified rotor-furl spring/damper model.
UserTeet()	User-specified rotor-teeter spring/damper model.
UserTFin()	User-specified tail fin aerodynamics model.
UserTFrl()	User-specified tail-furl spring/damper model.
UserVSCont()	User-specified variable-speed torque and power control model.
UserYawCont()	User-specified nacelle-yaw control model.

Table 6.3: Parameter settings for BladedDLL-Interface [20].

Parameter	Setting	Reason
YCMode	1	Tells FAST to use routine UserYawCont() for active yaw control.
TYCOn	0.0	Tells FAST to start active yaw control at the beginning of the simulation.
PCMode	1	Tells FAST to use routine PitchCntrl() for active pitch control.
TPCOn	0.0	Tells FAST to start active pitch control at the beginning of the simulation.
VSContrl	2	Tells FAST to use routine UserVSCont() for active variable-speed torque control.
GenTiStr	True	Tells FAST to start torque control based on time TimGenOn.
GenTiStp	True	Tells FAST to stop torque control based on time TimGenOf.
TimGenOn	0.0	Tells FAST to start torque control at the beginning of the simulation.
TimeGenOf	TMax	Tells Fast to stop controlling torque throughout the simulation.
HSSBrMode	2	Tells FAST to use routine UserHSSBr() for control of the HSS brake.
THSSBrDp	0.0	Tells FAST to start HSS brake torque control at the beginning of the simulation.

6.2 Input

The FAST primary input file is named *filename.fst*. It describes the wind turbine operating parameters and basic turbine geometry. An example of an input file can be seen in Appendix A. However, *filename.fst* also refers to data from a number of other input files, which are also fed to FAST, see Figure 6.1:

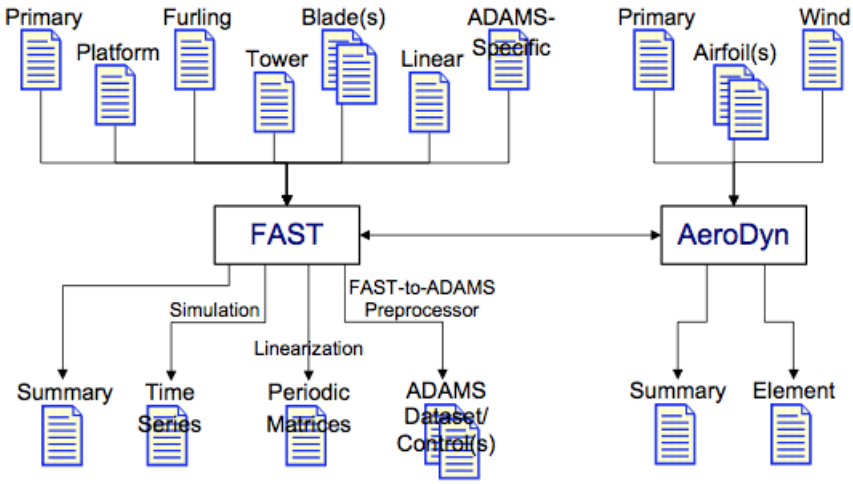

Figure 6.1: Input and output files [20].

The information contained in the different files are listed in Table 6.4.

The input files are written in simple text formats readable by any text editor. Examples of a platform file, a tower, and an AeroDyn input file may also be found in Appendix A. Turbulent wind files are an exception, but when these are created with TurbSim, see chapter 4, TurbSim also generates a *filename.sum*-file containing a summary of the *filename.wnd*-file. An input file to TurbSim, used for simulation in this thesis, is presented in Appendix B.

A list of input and possible output parameters can be found in [20].

Table 6.4: Input files [20].

Input File	Description
Tower input file	Contains information about distributed tower properties such as stiffness and mass density, and mode shapes. These parameters are specified at fractions of the tower height and FAST will linearly interpolate these data for the analysis nodes.
Platform input file	Contains, among other, information about the platform degrees of freedom, initial conditions, together with the tower draft and specification of the platform reference point. It tells what platform loading model and tower loading model should be used. The wave and current conditions are also specified in the platform file.
Furling input file	Parameters such as lateral offset and skew angle of rotor shaft, rotor-furling, tail-furling and tail inertia and aerodynamics are specified here. Furling is the process of turning the rotor or tail away from the wind direction at high wind speeds to protect the wind turbine from being exposed to large forces.
Blade input file(s)	This file contains information about blade characteristics. Distributed blade properties and blade mode shapes are specified. Aerodynamic coefficients of the blades are not listed here.
AeroDyn input files	The primary AeroDyn input file (*filename.ipt*) contains information about what kind of aerodynamic analysis tools to use. It also refers to a wind file (e.g. *filename.wnd*), and several airfoil files (*filename.dat*) containing aerodynamic coefficients of the blades. Blade discretization is set in AeroDyn (see Section 6.5), while tower discretization used by AeroDyn is set in the primary input file.
Linearization control-input file	This file lists input parameters related to FAST linearization analysis.
ADAMS-specific input file	This file lists input parameters related to ADAMS-specific functionalities, see Section 6.3.

6.3 Analysis Modes

FAST exhibits two kinds of analysis modes. One of them is *time-marching* of the nonlinear equations of motions, i.e. simulation. In this mode, the wind turbine response to aero- and hydrodynamic loads are calculated along time steps. Several aspects of tur-

bine operation may be considered through active controls. Outputs of loads and deflections are given as time-series data that can be used for example to determine extreme and fatigue loads. It is possible to run the simulation analyses as a *dynamic-link-library* (DLL) interfaced with *Simulink*, a simulation tool that introduces significant flexibility in wind turbine controls implementation during simulation. The second analysis mode is *linearization*. In this case, linear representations of the nonlinear wind turbine model is extracted. This is useful for finding the full system modes of the operating or stationary turbine through simple eigenanalysis. It can also be used to develop state matrices of a wind turbine "plant" as a tool in controls design and analysis. Moreover, the ADAMS preprocessor is available for use in FAST, and is separate from the two analysis modes available. Input parameters from FAST are used to construct an ADAMS dataset of the complete aeroelastic wind turbine. This way, the ADAMS code is the code where turbine simulation or linearizations are performed. This ensures that the ADAMS and FAST models are consistent, and provides quick and easy creation of ADAMS datasets [20].

6.4 Model Description

6.4.1 General

As mentioned above, FAST can model both two-bladed and three-bladed horizontal-axis wind turbines. Rotor-furling, tail-furling and tail aerodynamics are features that may be useful.

FAST is governed by a combination of multi-body and modal-dynamics formulation. The multi-body formulation applies to the following rigid bodies: the earth, platform, base plate, nacelle, armature, gears, hub, tail and structure furling with the rotor. The tower, blades and drive shaft are represented using a linear modal formulation. FAST uses an assumed-modes approach; it needs blade and tower modes as well as structural properties to compute modal integrals for its equations of motion.

A two-bladed wind turbine utilizes 22 DOFs, while a three-bladed wind turbine requires

24 DOFs. Several coordinate systems are used, to suitably represent the many possible motions. The parametrization of a conventional, upwind, three-bladed turbine is shown in Figure 6.2 below.

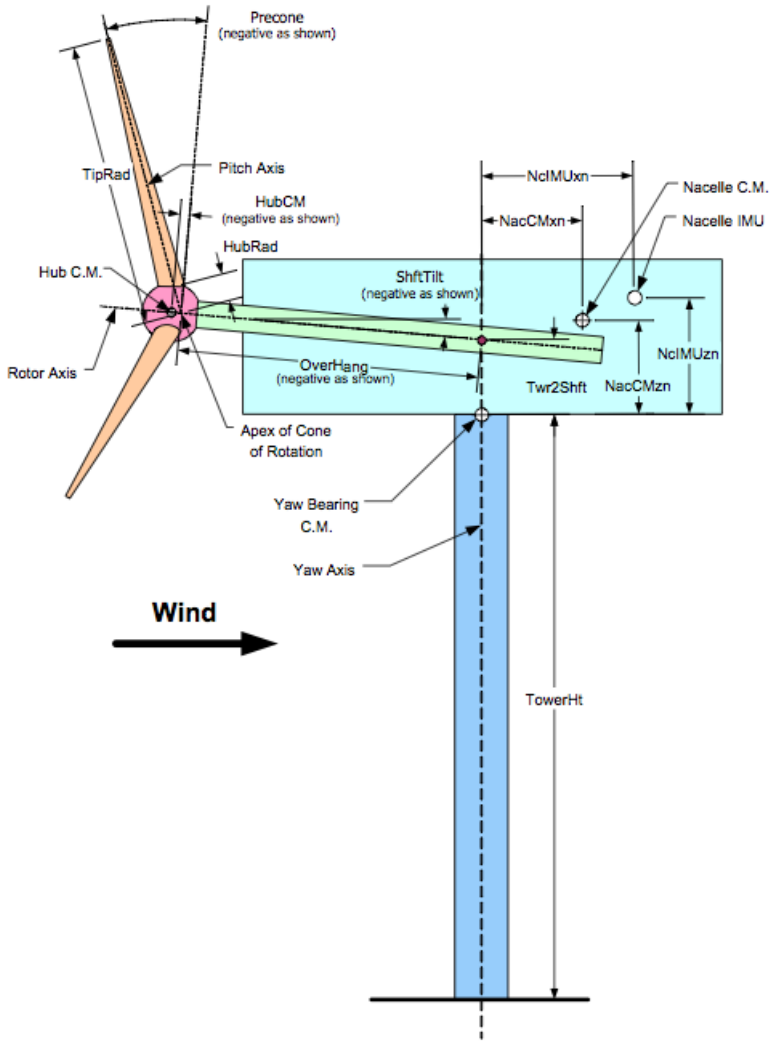

Figure 6.2: Parametrization of an upwind, three-bladed wind turbine [20].

A 4^{th}-order explicit Runge-Kutta integration algorithm is used for the first three time steps of the simulation, while a combined Adams-Bashforth predictor and Adams-Moulton corrector integration scheme is used for all other time steps.

FAST holds many different features involving many fields of expertise. However, only

the features that are most relevant to this thesis will be investigated further. Areas such as drivetrain, generator, nacelle yaw, rotor-furl, tail-furl, rotor-teeter and tail fin aerodynamics will not be investigated further.

6.4.2 Tower and Blades

Flexible elements in the tower and blades are characterized using a linear modal representation that assumes small deflections. As already mentioned, FAST uses an assumed-modes approach, meaning that it requires blade and tower mode shapes as input. A software called *BModes* is developed primarily to provide coupled mode shapes for the user, so that the user can provide these to FAST.

For the tower, four different shape modes can be specified in FAST: two fore-aft modes and two side-to-side modes. For each of the blades, FAST can use two flapwise modes and one edgewise mode. The blade modes are defined relative to the local structural twist, so that the shapes twist with the blade. Straight Bernoulli-Euler beams are used for the model. This implies no axial or torsional DOF and no shear deformation. Otherwise, all terms include full nonlinearity: Mode shapes are used as shape functions in a nonlinear beam model (Rayleigh-Ritz method), and motions include radial shortening effect (geometric nonlinearity), which is important for centrifugal stiffening. Also, inertial loads include nonlinear centrifugal, Coriolis, and gyroscopic terms.

6.4.3 Support Platform

The tower is attached to a support platform through a cantilever connection. However, the six degrees of freedom of the platform reference point can be fixed or free, see Figure 6.3, enabling the modeling of foundation flexibility. The support platform may be in an onshore foundation, in a bottom-fixed offshore foundation, or may have a floating offshore configuration.

As can be seen in Figure 6.3, the *tower base* is in FAST defined as the point where the tower is attached to the platform. In reality, the monopile will extend from the platform to some location above MSL, where it is attached to the actual tower.

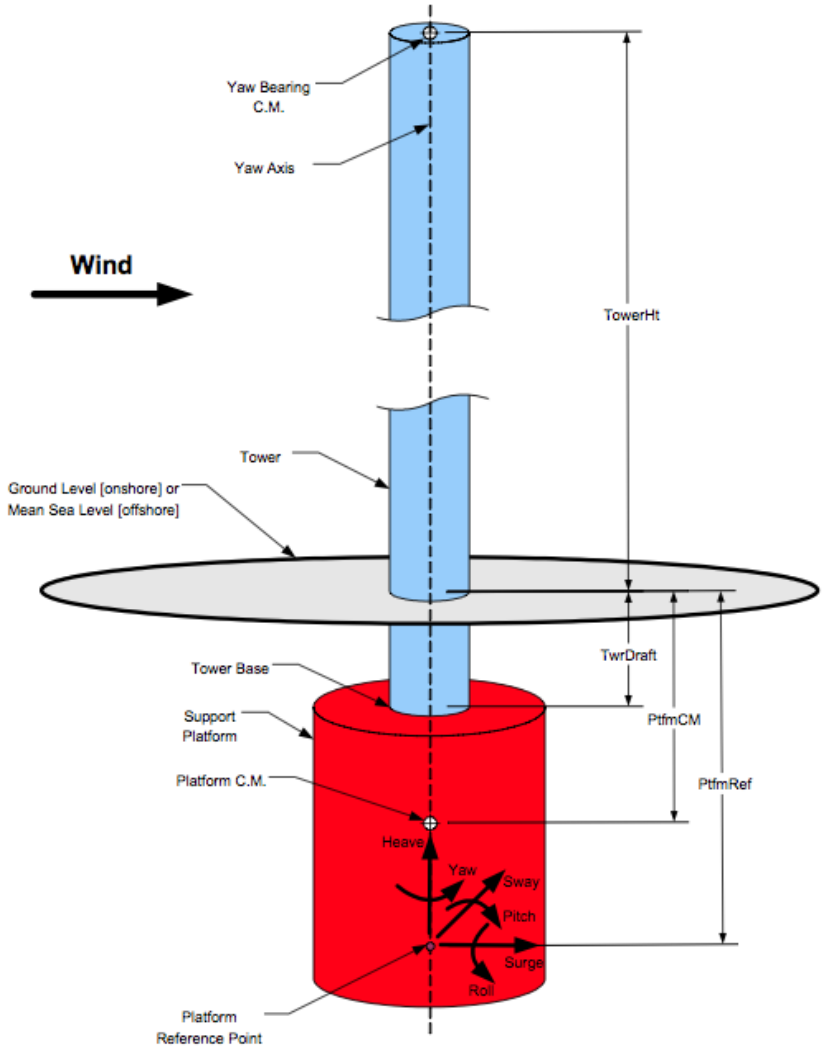

Figure 6.3: Support platform layout [20].

As well as being the origin of the platform, and the point about which the DOFs of the platform are defined, the platform reference point is also the point where external loads are applied to the platform.

The platform loads may be specified by the user to include contributions from foundation stiffness and damping, or mooring line restoring and hydrodynamic contributions for a bottom-fixed or floating turbine, respectively. This is done through a user defined subroutine, UserPtfmLd (). This subroutine is an important tool when implementing

nonlinear soil-structure interaction later in this thesis.

6.4.4 HydroDyn

HydroDyn is the part of FAST that concerns hydrodynamic loads on the substructure. It is applicable to both bottom-mounted and floating offshore wind turbines. For FAST v8 it exists as a standalone module. However, in FAST v7, the input parameters of HydroDyn are specified in the platform input file, and HydroDyn is less extensive than in FAST v8.

For monopiles, the relative form of *Morison's equation*, which is a semi-empirical equation for force on a body in oscillatory flow, is used to calculate loads. For floating platforms, both a *potential-flow theory solution*, a *strip-theory solution* (including the relative form of Morison's equation), or a combination of these two can be used for load calculation. Regular and irregular linear waves can be modeled, with or without wave stretching, and with or without sea currents. The irregular waves may be represented with a JONSWAP or Pierson-Moskowitz spectrum, or through a user-defined spectrum, or with GH Bladed wave data (or some other wave kinematics software).

The FAST v7 UserGuide is not updated to include the hydrodynamics part of the software, but a quick review regarding monopiles can be found in Jonkman and Jonkman [14].

6.5 AeroDyn

AeroDyn calculates the aerodynamic lift, drag and pitching moment of airfoil sections along the wind turbine blades. Each blade is broken into a number of segments along the span, which is specified in the AeroDyn input file. Aerodynamic forces influence the turbine deflections and vice versa. Two-dimensional localized flow is considered, and characteristics of the airfoils along the blade are represented by lift, drag, and pitching moment coefficients measured in wind tunnel tests.

The user must specify parameters such as geometry, airfoil data, undisturbed wind inflow, and air density.

To calculate the forces on the blades, it is necessary to calculate the induced velocities from the wake in the rotor plane. In AeroDyn, this can be done using one of the following two models: The *blade element momentum* (BEM) theory, or the *generalized dynamic-wake* (GDW) theory. The former may also be referred to as equilibrium wake.

BEM theory discretizes blades into elements and assumes momentum balance in ring-shaped sections in the rotor field. The blade-element loads are obtained using airfoil data, see Figure 6.4. The GDW theory, on the other hand, is based on a potential flow solution to Laplace's equation. The transient loading leads to a dynamic wake, and the GDW models the time- and spatial-varying induction across the rotor. For BEM theory, modifications must be done regarding tip losses, hub losses and skewed wake, whereas these are automatically included in the GDW model. This makes the GDW model more appropriate for most applications.

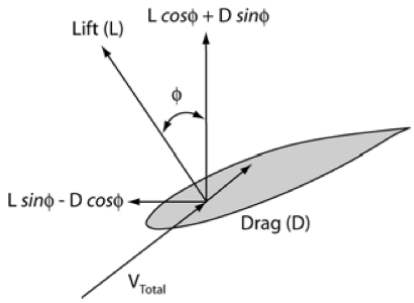

Figure 6.4: Local element forces for the BEM model.

The airfoil characteristics may be given as static airfoil tables, or they may include dynamic stall. The latter method is based on the semi-empirical Beddoes-Leishman model [23].

AeroDyn also includes a tower shadow model, both for upwind and downwind conditions. It provides the influence of the tower on the local velocity field at all points around the tower, which in turn influences the blade aerodynamics.

AeroDyn obtains wind input data and information about the velocity field as a func-

tion of time, from either *hub-height* wind files or from *full-field* wind files. Hub-height wind files contain hub-height wind speed and direction. Linear interpolation in time is performed to obtain a uniform, but time-varying wind field. Full-field turbulence wind files contain turbulent velocity components at points on a grid. This grid covers an area slightly larger than the rotor disc and tower.

6.6 TurbSim

As already mentioned in Section 6.2, TurbSim can, and will, be used to generate wind files for this thesis. This software is not includded in FAST, but is recommended by NREL for generation of wind files. This stochastic, full-field turbulent-wind simulator applies a statistical model to produce a numerical simulation of the time series of wind-speed vectors. The vectors are three-dimensional and located at points in a two-dimensional, vertical, fixed grid. An example of a TurbSim wind field with 15° horizontal and 8° vertical mean flow angles is shown in Figure 6.5. The capital letters represent the initial reference frame, while the small letters are vectors aligned with the mean wind.

In the TurbSim input file a number of parameters and properties are specified. These are categorized into five sections. The first one, *Runtime Options*, contains choices regarding output types, and initializes a random generator. In the second one, *Turbine/Model Specifications*, dimensions of the grid, analysis time specifications, hub height of the turbine, and mean flow angles are stated. Under the *Meteorological Boundary Conditions* section, wind characteristics are specified. These include what spectral models to apply, wind profile type, reference height, mean wind speed, and so on. TurbSim offers several wind profile types, including the power-law wind profile, logarithmic wind profiles, low-level jet wind-profile, and the option to apply power-law profile on the rotor disk and a logarithmic profile elsewhere. The Kaimal or von Karman spectral model are both based on *International Electrotechnical Commission* (IEC) standards. Thus, if one of these is applied, some parameters related to these standards need also be specified. If an alternative to these two turbulence models is applied, the section *Non-IEC Meteorological Boundary Conditions* must be considered as well. The final category refers to *Coherent Turbulence Scaling Parameters*, which are used with some

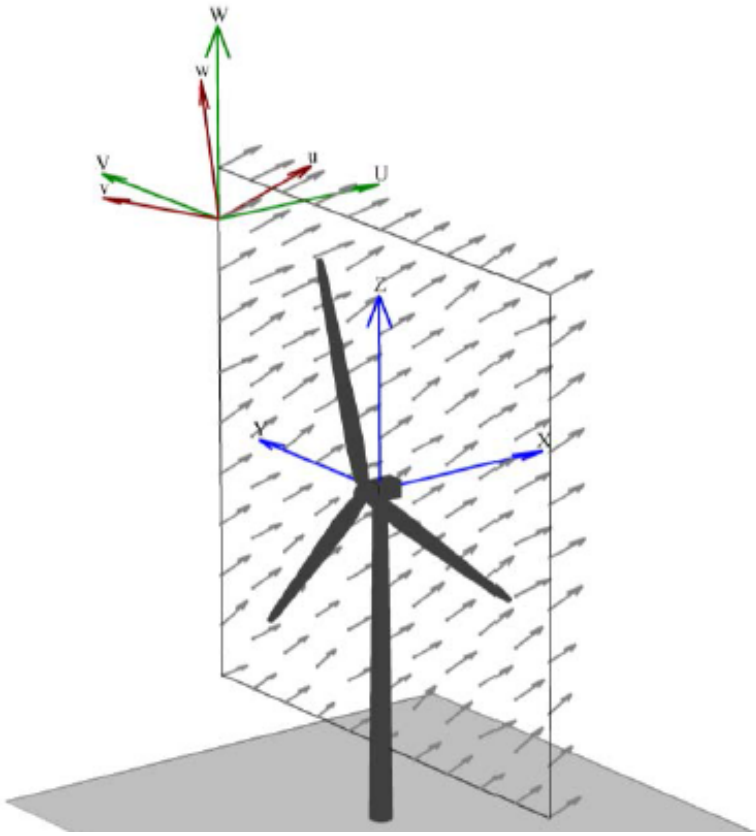

Figure 6.5: Coordinates of a TurbSim wind field [15].

of the non-IEC spectral models.

When TurbSim is used to generate full-field wind files, in the input file for TurbSim, the parameters *RandSeed1* and *RandSeed2* may be specified. RandSeed1 is a random number, whereas RandSeed 2 is either a second random number, or set to *RNSNLW* or *RANLUX*. RANLUX is used in this thesis. It is a *pseudorandom number generator* (pRNG). A pRNG algorithm generates a sequence of numbers with properties that are approximately those of sequences of random numbers. The sequence generated by the pRNG is not completely unsystematic because it must be initialized by a number, which in this context is specified in RandSeed1. In TurbSim, the pRNG is used to create random phases for the velocity components: one per frequency per grid point per wind component [15]. If RandSeed1 and RandSeed2 are not changed the user is able to repeat

the phases for different runs, so results will be comparable when other parameters are varied.

An example of an input file for TurbSim can be found in Appendix B.

Chapter 7

Base Model with Rigid Foundation

To simplify the study of offshore wind technology, NREL has developed a standard set of turbine specifications for a fictitious offshore 5 MW baseline wind turbine supported on a monopile. NREL has provided a model of this reference turbine in FAST. This model will be used as a base for analyses in this thesis, and it will subsequently be referred to as the *NREL 5MW*. In this chapter, the reference turbine will presented, along with typical simulation inputs and results.

7.1 Properties of the NREL 5MW OWT

The NREL 5MW is meant to be representative of a large land- or sea-based multimegawatt wind turbine, also appropriate for deep water sites. The choice of a power rating of 5 MW is based on the idea that a cost-efficient deep-water turbine should be rated at at least 5 MW [19]. Its properties are obtained from publicly available information on real wind turbine prototypes and conceptual models. Table 7.1 lists gross properties of the turbine.

Table 7.1: Properties of the NREL 5-MW Baseline OWT with a monopile support structure [19].

Rating	5 MW
Rotor orientation, configuration	Upwind, 3 blades
Rotor diameter	126 m
Hub-height	90 m
Tower top diameter, wall thickness	3.87 m, 0.019 m
Tower base diameter, wall thickness	6.0 m, 0.027 m
Substructure diameter, wall thickness	6.0 m, 0.06 m
Cut-in, rated, cut-out wind speed	3 m/s, 11.4 m/s, 25 m/s
Rated rotor speed	12.1 rpm
Rated tip speed	80 m/s
Rotor mass	110,000 kg
Nacelle mass	240,000 kg
Tower mass	346,460 kg
Mean sea level	20.0 m

The tower ends at the tower base, 10 m above MSL. From there, the monopile extends down to the mudline at 20 m below MSL. The diameter and wall-thickness of the tower are linearly tapered from the top to the tower base. The monopile has a constant diameter and wall-thickness of 6 m and 0.060 m, respectively [21]. Since the default model in FAST attaches the monopile to the mudline through a rigid foundation, the depth of the monopile into the ground is not given. However, in Chapter 8 the flexibility of the foundation is modeled based on a monopile penetration depth of 36 m below the mudline [30].

In addition, blade structural and aerodynamic properties, hub and nacelle properties, drivetrain properties, tower properties, and control system properties are given. The turbine utilizes active yaw, meaning that the rotor is continuously oriented to face the wind.

The full-system natural frequencies of the fore-aft and side-to-side tower mode shapes are [19]:

Table 7.2: First natural frequencies of the tower [19].

Mode	Description	Natural frequency (Hz)
1	1st Tower fore-aft	0.324
2	1st Tower side-to-side	0.312
:	:	:
:	(Blades)	:
12	2nd Tower fore-aft	2.900
13	2nd Tower side-to-side	2.936

It is important to notice that the aforementioned natural frequencies and mode shapes are related to the full system - that is, how the blades couple with the drivetrain and tower. They are obtained using a FAST linearization analysis followed by an eigenanalysis. These mode shapes are outputs from FAST, whereas individual mode shapes for tower and blades are input to FAST.

As mentioned in Section 2.4 wind turbines also have natural frequencies that are highly related to their operational state. For the NREL 5MW turbine, with a rated rotor speed of 12.1 rpm, this implies the following system natural frequencies:

$$1P = \frac{12.1 min^{-1}}{60 s/min} = 0.20 s^{-1} = 0.20 Hz \tag{7.1}$$

$$3P = 3 \cdot 0.20 Hz = 0.60 Hz \tag{7.2}$$

Evidently, the wind turbine has a soft-stiff design, since the natural frequency of the first tower modes are larger than 1P but less than 3P, see Figure 2.6.

7.2 Standard Simulation of the NREL 5 MW in FAST

The input files for the NREL 5MW OWT were supplied by NREL in a format suitable for FAST. The primary input file (*filename.fst*), as well as the tower file, platform file,

and AeroDyn input file, is given in Appendix A. Only parameters such as time step and simulation duration were changed. However, to make us of these input files, the original version of FAST v7 must be recompiled to include a DLL-Bladed Interface, as described in Chapter 6. As a default, FAST v7 assumes a rigid foundation. Correspondingly, all of the degrees of freedom at the end of the monopile are fixed. Modeling the monopile with a rigid foundation is a non-conservative method because it is not able to reflect the flexibility of the foundation. Still, this is the standard setting of FAST v7, Fast v8, and many other softwares.

To illustrate a typical output scenario when analyzing the NREL 5MW OWT in FAST v7 with a monopile support structure and a rigid foundation, a typical load scenario was applied. In this context, a realistic load situation would imply a full-field, turbulent wind-field in combination with irregular waves.

Simulation time was set to 630 s, or 10 min and 30 s. However, the outputs for the first 30 seconds of the simulations were omitted, to allow for a steady-state situation to occur.

7.2.1 Input

A full-field wind file was generated by TurbSim, see Section 6.6. It was based on the IEC Kaimal turbulence model. The wind speed at the hub height of the turbine was set to 12 m/s, with turbulence intensity $I_{ref} = 0.14$ corresponding to class B of IEC61400-1 [26]. The input file used in TurbSim to generate the wind file can be found in Appendix B. A time series of the wind in horizontal (x- and y-direction) and vertical direction can be seen in Figure 7.1. The coordinate system presented in Figure 6.5 applies here.

As prescribed, the velocities in y- and z-direction merely fluctuate around a zero wind speed, whereas the mean velocity of the wind in the x-direction is 12 m/s. Even though the average velocity is 12 m/s in the x-direction, the random fluctuations in all three directions cover relatively large spans, with deviations in the range of ±4-6 m/s.

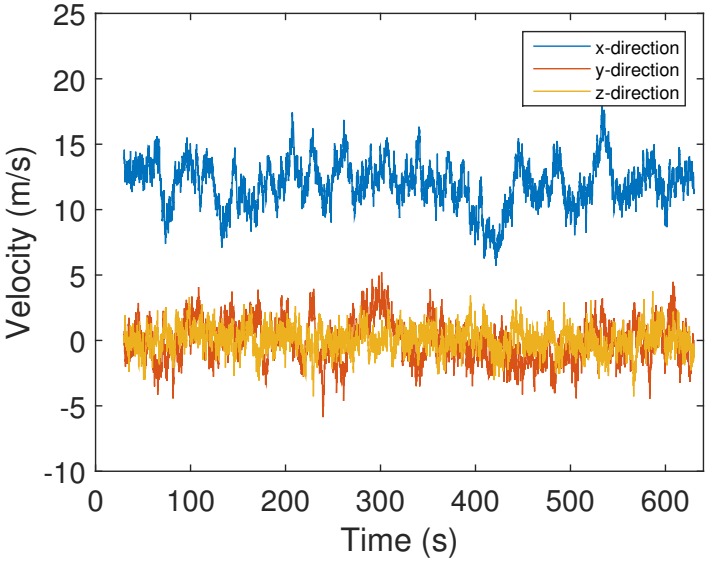

Figure 7.1: Wind field generated using TurbSim.

Irregular waves were generated based on a JONSWAP spectrum. Significant wave height, H_s was set to 5 m and the peak spectral period T_p was set to 12 s. No current was applied. The wave elevation above the MSL is plotted in Figure 7.2.

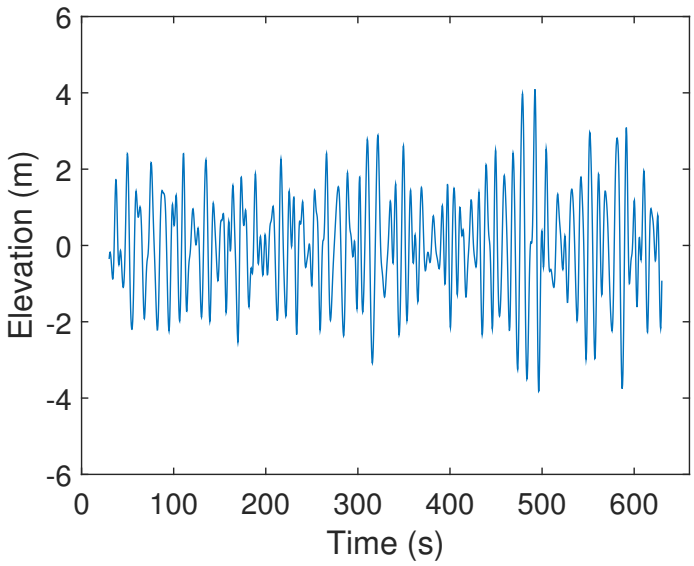

Figure 7.2: Wave elevation (m) above MSL. JONSWAP irregular waves. $H_s = 5m$ and $T_p = 12s$.

As mentioned in Chapter 5, the significant wave height is defined to be four times the standard deviation of the elevation above the sea surface. The standard deviation of the wave elevation in Figure 7.2 is 1.3 m. This implies a significant wave height of approximately 5 m, as prescribed.

7.2.2 Outputs

The outputs of the simulation are given as time histories. As explained in Section 2.5, cyclic load changes in bending moments in the turbine support structure at the mudline are critical when designing wind turbines. Therefore, bending moments of the monopile at the mudline are interesting outputs. These are plotted in Figures 7.3 and 7.4. Moment about the z-axis, i.e. yaw moment, is plotted in 7.5, and in Figures 7.6, 7.7 and 7.8 time histories of forces in x-,y- and z-direction at the same location can be seen. Moreover, it could be of interest to plot time histories of bending moments further up along the support structure. Fore-aft bending moments in the monopile halfway between the mudline and the MSL, at MSL, and at the monopile/tower connection are plotted in Figure 7.9, together with the already mentioned bending moment at the mudline. For a more visual illustration of the movements of the turbine, time histories of the displacements of the tower-top are shown in Figures 7.10 and 7.11.

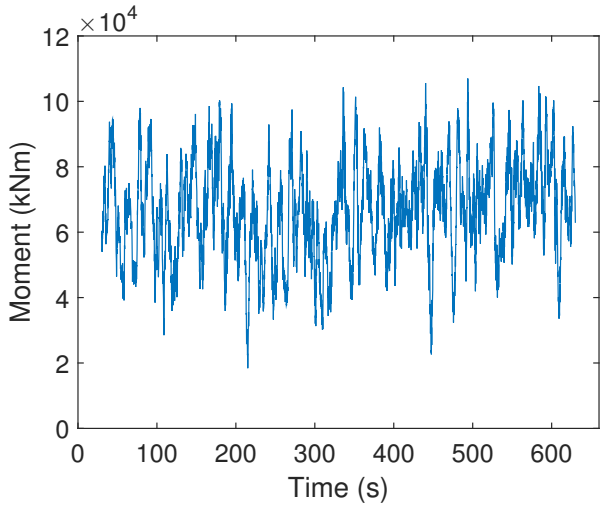

Figure 7.3: Fore-aft bending moment in monopile at mudline.

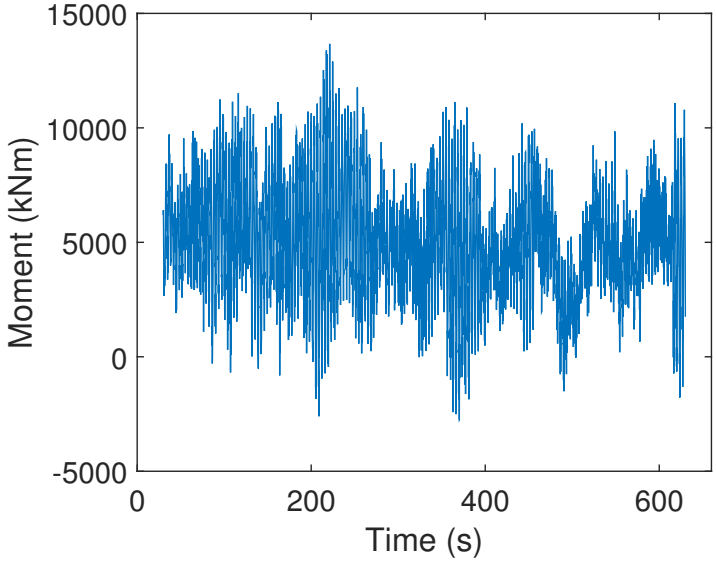

Figure 7.4: Side-to-side bending moment in monopile at mudline.

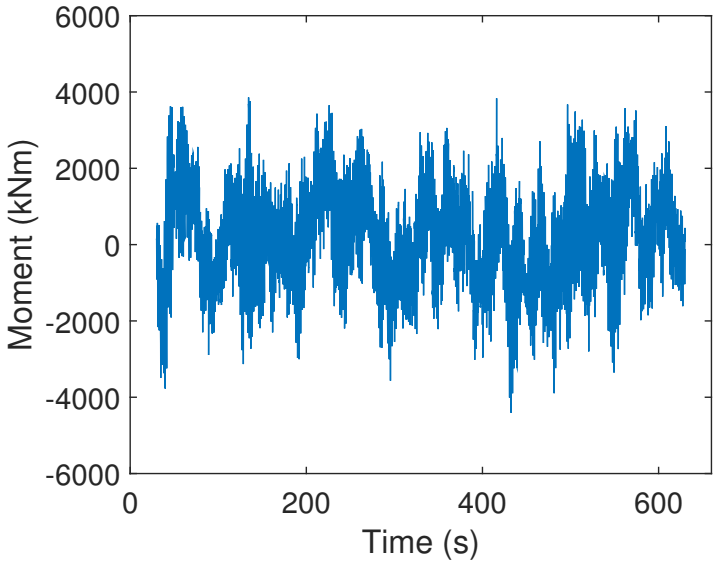

Figure 7.5: Yaw moment in monopile at mudline.

Whereas the fore-aft bending moment reaches a maximum value of $108.5 \cdot 10^3$ kNm, the maximum side-to-side moment is less by a factor of ten, namely $13.8 \cdot 10^3$. Considering

that the main wind and wave load is applied in the x-direction, this is expected. As for the yaw moment at the mudline, the maximum is decreased by yet another factor of ten. Hence, fore-aft bending moments are more significant, and therefore kept in focus. The side-to-side bending moment vibrates with very short periods of 3.5 s, which may be due to smaller short-period turbulent oscillations in the wind.

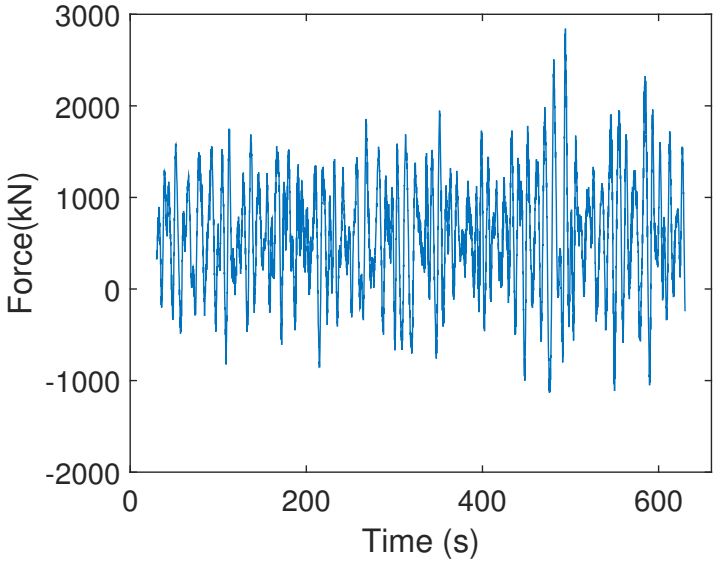

Figure 7.6: Fore-aft shear force at mudline.

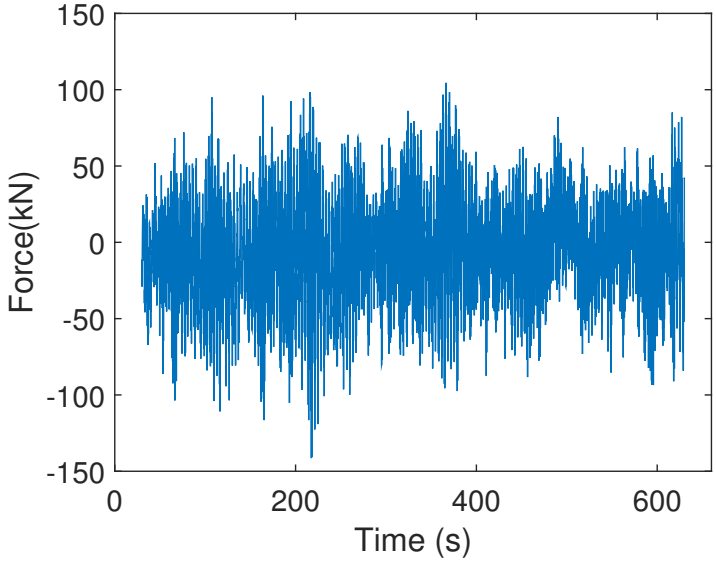

Figure 7.7: Side-to-side shear force at mudline.

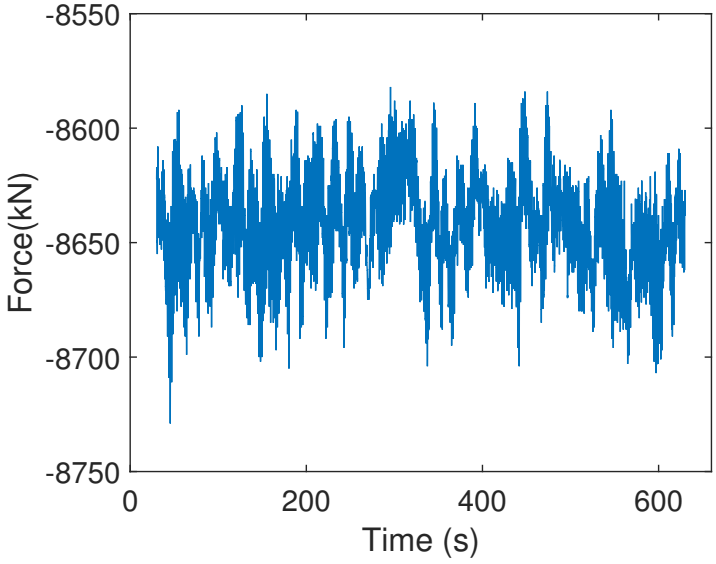

Figure 7.8: Vertical force at mudline.

Looking at the three graphs in Figures 7.6, 7.7 and 7.8 above, some of the same trends seen for the bending moments at the mudline are also applicable to the shear forces at

the same location. The fore-aft shear force reaches a maximum of $2.45 \cdot 10^3$ kN, while the side-to-side shear force shows rapid oscillations with much smaller amplitude. However, due to the large weight of the tower and RNA, the mean vertical force is much larger than both lateral forces. The standard deviations are however small, namely 20 kN. The large vertical force may therefore be considered as more or less static.

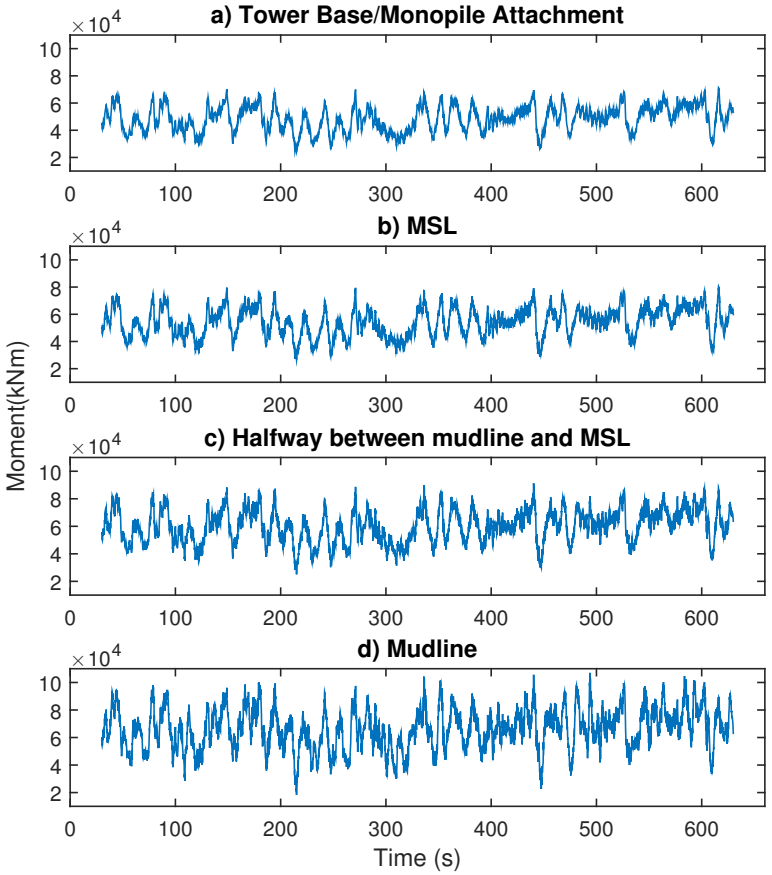

Figure 7.9: Fore-aft bending moments along lower part of tower.

As seen in Figure 7.9, it is evident that the fore-aft bending moments decrease as output locations are moved upwards. The maximum bending moment is $108.5 \cdot 10^3$ kNm at the mudline and $70.9 \cdot 10^3$ kNm at the location where the tower base is attached to the monopile, 10 m above MSL. In other words, the largest bending moments in the

monopile occur at the mudline, as expected. Moreover, the standard deviations decrease further up the tower, namely from $14.4 \cdot 10^3$ kNm to $9.2 \cdot 10^3$ kNm.

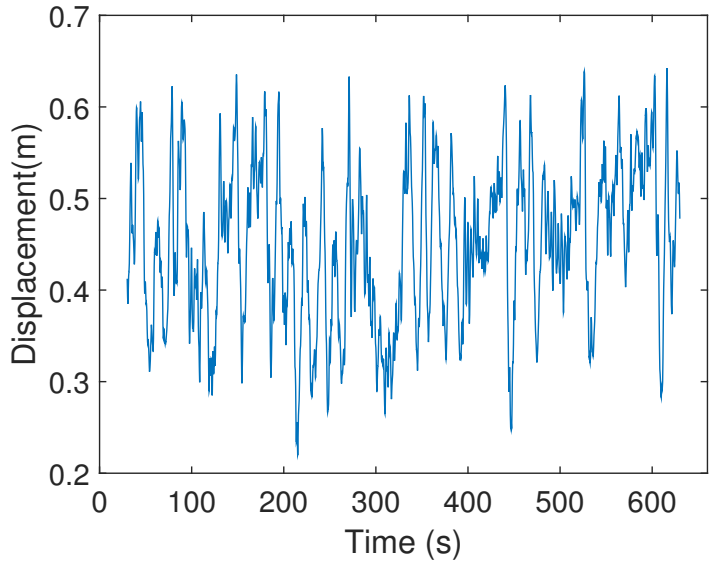

Figure 7.10: Fore-aft tower-top displacement.

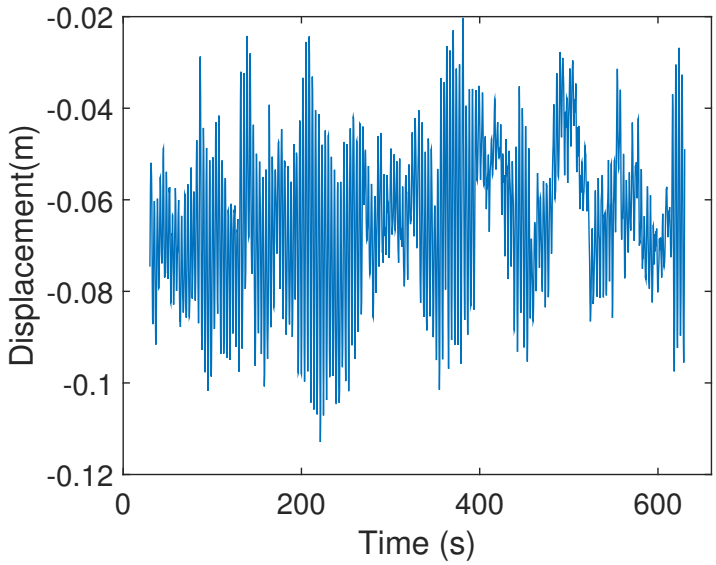

Figure 7.11: Side-to-side tower-top displacement.

The maximum fore-aft displacement of the tower-top is 0.66 m, while the standard deviation is 0.08 m about a mean value of 0.45 m. Side-to-side translations fluctuate rapidly close to a zero mean displacement. However, the mean velocity of the wind in the y-direction is slightly less than zero, namely -0.066 m/s. This small accidental imbalance in wind velocity in the y-direction is recognized in the tower-top side-to-side displacement as a mean displacement of -0.065 m.

Chapter 8

Coupled Linear Springs Foundation Model

In this chapter, the NREL 5MW is modeled using coupled linear springs as a representation of the foundation. This concept is described in Section 3.2. The stiffness of the foundation is varied to investigate how this will influence the wind turbine response. The model will subsequently be refered to as the CLS (coupled linear springs) model.

8.1 CLS in FAST

To implement the coupled linear springs as a model for the foundation flexibility, some changes had to be done in the original source code of FAST v7, and the input files for NREL 5MW.

8.1.1 The User-Specified Subroutine UserPtfmLd()

FAST v7 possesses the option to implement a user-specified subroutine, *UserPtfmLd()*, for platform loading, as mentioned in Sections 6.1 and 6.1. The platform is a reference point to which the monopile is rigidly connected. The platform loads applied by this

subroutine are supposed to include contributions from external loads acting on the platform other than those transferred from the wind turbine itself. Contributions from foundation stiffness should therefore be applied here.

The subroutine for this foundation flexibility modeling approach was supplied by NREL and can be found in Appendix D. Also included in the code is a template to incorporate constant added mass and damping matrices. However these parameters were not in focus in this thesis, and are therefore not utilized in this chapter. The damping implementation feature will be investigated further in Chapter 9. The input given from the FAST program to the UserPtfmLd() subroutine in each time step is a vector containing the current deflections and rotations at the platform reference point. The output from UserPtfmLd() is a vector holding the corresponding forces and moments at the platform reference point.

When implementing this subroutine some parameters in the platform input file must be altered. The monopile is no longer rigidly attached to the mudline, but is instead limited in movement along, and against rotation about, the horizontal axes by linear springs. For these springs to be applied, motion in these DOFs, namely surge, sway, pitch and roll, must be allowed for. This is done by applying the following settings in the platform input file, see Table 8.1.1.

Table 8.1: Settings in platform input file when coupled springs are applied.

PtfmSgDOF	True
PtfmSwDOF	True
PtfmHvDOF	False
PtfmRDOF	True
PtfmPDOF	True
PtfmYDOF	False

8.1.2 Mode Shapes for Coupled Springs

Mode shapes for the tower must be given as input to FAST. For the NREL 5MW Offshore monopile with a rigid foundation, mode shapes are included with the input files that NREL supplies. However, when using the CLS approach, the boundary conditions of the tower are changed, requiring an update of the mode shapes. By using the software BModes, see Chapter 6, combined with the spreadsheet *ModeShapePolyfitting.xls* that was given with the FAST archive, and NREL provided BModes input files for coupled springs [16], mode shapes were derived for the CLS approach. Files applied in BMode are presented in Appendix C. As proposed by NWTC [28], the parameters *edge_iner* and *flp_iner* were set to very small numbers, while the parameters *tor_stff* and *axial_stff* were set to very high numbers in the *filename.bmi*-input file to BModes. It may be worth noticing that when the boundary conditions, and thereby mode shapes and natural frequencies of the structures, are changed it may be necessary to change the simulation time step as well. According to [28], the time step should be larger than, or equal to $\frac{1}{10\omega_{max}}$, where ω_{max} is the largest natural frequency of the system.

The first four natural frequencies found for the tower using the coupled springs approach and the aforementioned stiffness matrix are as follows:

Table 8.2: First natural frequencies of the tower.

1st fore-aft	0.240 Hz
1st side-to-side	0.242 Hz
2nd side-to-side	1.359 Hz
2nd fore-aft	1.528 Hz

These are not full-system natural frequencies as those described in Chapter 7, but merely natural frequencies of a single tower with boundary conditions supplied by the CLS foundation.

8.2 Foundation Stiffness

For the current foundation flexibility model, coupled linear springs are used, see Figure 8.1. The springs are applied to the platform for movements in surge, sway, roll and pitch DOFs, e.g. all lateral motions. Heave translation and yaw rotation DOFs are fixed, and this is specified in the platform input file, as already mentioned in Section 8.1.

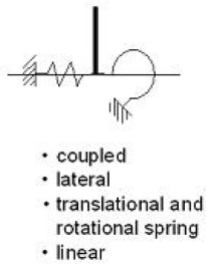

Figure 8.1: The CLS model [14].

The springs are linear, meaning that the displacements and rotations of the platform are linearly related to the forces and moments at the same location through a constant stiffness matrix. Coupled linear springs imply that a coupled foundation stiffness matrix is used. The physical meaning of such a coupled stiffness matrix is that a lateral load in the x-direction F_x will cause not only a lateral deflection u_x, but also a rotation θ_y. Similarly, a moment about the x-axis, M_x will cause both a rotation θ_x and a deflection u_y.

The foundation stiffness matrix relates the movements and respective loads in the springs as seen in Equation (8.2).

$$\begin{bmatrix} \boldsymbol{k_{uu}} & \boldsymbol{k_{u\theta}} \\ \boldsymbol{k_{\theta u}} & \boldsymbol{k_{\theta\theta}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \end{bmatrix} \tag{8.1}$$

For six degrees of freedom at the spring attachment location, the matrix expands to the following:

$$\boldsymbol{k_{uu}} = \begin{bmatrix} k_{u_x u_x} & k_{u_y u_x} & k_{u_z u_x} \\ k_{u_x u_y} & k_{u_y u_y} & k_{u_z u_y} \\ k_{u_x u_z} & k_{u_y u_z} & k_{u_z u_z} \end{bmatrix} \tag{8.2}$$

,

$$\boldsymbol{k_{u\theta}} = \begin{bmatrix} k_{u_x \theta_x} & k_{u_y \theta_x} & k_{u_z \theta_x} \\ k_{u_x \theta_y} & k_{u_y \theta_y} & k_{u_z \theta_y} \\ k_{u_x \theta_z} & k_{u_y \theta_z} & k_{u_z \theta_z} \end{bmatrix} \tag{8.3}$$

$$\boldsymbol{k_{\theta u}} = \begin{bmatrix} k_{\theta_x u_x} & k_{\theta_y u_x} & k_{\theta_z u_x} \\ k_{\theta_x u_y} & k_{\theta_y u_y} & k_{\theta_z u_y} \\ k_{\theta_x u_z} & k_{\theta_y u_z} & k_{\theta_z u_z} \end{bmatrix} \tag{8.4}$$

and

$$\boldsymbol{k_{\theta\theta}} = \begin{bmatrix} k_{\theta_x \theta_x} & k_{\theta_y \theta_x} & k_{\theta_z \theta_x} \\ k_{\theta_x \theta_y} & k_{\theta_y \theta_y} & k_{\theta_z \theta_y} \\ k_{\theta_x \theta_z} & k_{\theta_y \theta_z} & k_{\theta_z \theta_z} \end{bmatrix} \tag{8.5}$$

Moreover, $\boldsymbol{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$, $\boldsymbol{\theta} = \begin{bmatrix} \theta_x & \theta_y & \theta_z \end{bmatrix}^T$, $\boldsymbol{F} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T$ and $\boldsymbol{M} = \begin{bmatrix} M_x & M_y & u_z \end{bmatrix}^T$. u_x is displacement in x-direction, θ_x is rotation about the x-axis, F_x is force in the x-direction, M_x is moment about the x-axis, and so on.

Note that loads acting on the platform from the springs will be in the opposite direction of the movement of the platform. This means that when the platform translates or rotates in a positive direction, the forces experienced by the springs, and thereby the platform, will be negative. Stiffness coefficients were derived by Passon [30] based on the soil profile and pile depth presented in Figure 8.2.

The stiffness coefficients were then applied in the stiffness matrix used for the simulations. The method that Passon [30] used to derive this stiffness matrix is briefly presented here.

The stiffness coefficients were based on a fictitious bending stiffness and length derived using the apparent fixity approach and the stiffness matrix for a Bernoulli beam:

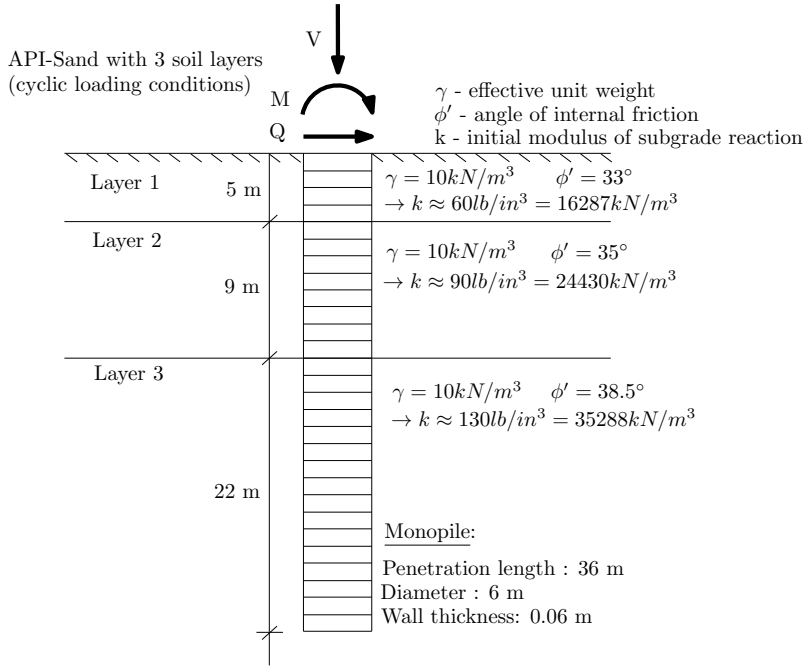

Figure 8.2: The soil profile from which the foundation stiffness matrix was derived [30].

$$\begin{bmatrix} \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{bmatrix} u \\ \theta \end{bmatrix} = \begin{bmatrix} F \\ M \end{bmatrix} \tag{8.6}$$

The fictive parameters are:

$$\begin{aligned} EI_{fictive} &= 1.15 \cdot 10^{12} Nm^2 \\ L_{fictive} &= 1.75 \cdot 10m \end{aligned} \tag{8.7}$$

These fictive parameters were computed based on a given set of translations and rotations corresponding to a pair of force and moment, see Equation (8.2). That is, for a given force and moment, corresponding translation and rotation was calculated for a complex model in the geo-technical software *LPILE 4.0*.

$$\begin{aligned} u &= 0.0226m \\ \theta &= 0.0024rad \\ F &= 3910kN \\ M &= 124 \cdot 10^3 kNm \end{aligned} \tag{8.8}$$

Correspondingly, the stiffness matrix below was obtained for a pile in three dimensions with fixed yaw and heave.

$$\begin{bmatrix} \boldsymbol{k_{uu}} & \boldsymbol{k_{u\theta}} \\ \boldsymbol{k_{\theta u}} & \boldsymbol{k_{\theta\theta}} \end{bmatrix} = \begin{bmatrix} 2.57 \cdot 10^9 & 0.0 & 0.0 & 0.0 & -22.5 \cdot 10^9 & 0.0 \\ 0.0 & 2.57 \cdot 10^9 & 0.0 & 2.25 \cdot 10^{10} & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.25 \cdot 10^{10} & 0.0 & 2.63 \cdot 10^{11} & 0.0 & 0.0 \\ -22.5 \cdot 10^9 & 0.0 & 0.0 & 0.0 & 2.63 \cdot 10^{11} & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{8.9}$$

where the matrix at the left is as described above. The units of $\boldsymbol{k_{uu}}$, $\boldsymbol{k_{u\theta}}$, $\boldsymbol{k_{\theta u}}$ and $\boldsymbol{k_{\theta\theta}}$ are $[N/m]$, $[N/rad]$, $[Nm/m]$ and $[Nm/rad]$, respectively.

8.2.1 Foundation Stiffness According to Eurocode 8

Eurocode 8, Part 5 [9] proposes some alternative expressions to obtain a foundation stiffness matrix for use in the coupled springs approach. It assumes a soil profile and applies a foundation stiffness of piles using closed-form solutions, as listed in Table 8.2.1.

Table 8.3: Expressions for static stiffness of flexible piles embedded in three soil models [9].

Soil model	$\frac{k_{uu}}{dE_s}$	$\frac{k_{\theta\theta}}{d^3E_s}$	$\frac{k_{u\theta}}{d^2E_s}$
$E = E_s z/d$	$0.60\left(\frac{E_p}{E_s}\right)^{0.35}$	$0.14\left(\frac{E_p}{E_s}\right)^{0.80}$	$-0.17\left(\frac{E_p}{E_s}\right)^{0.60}$
$E = E_s\sqrt{z/d}$	$0.79\left(\frac{E_p}{E_s}\right)^{0.28}$	$0.15\left(\frac{E_p}{E_s}\right)^{0.77}$	$-0.24\left(\frac{E_p}{E_s}\right)^{0.53}$
$E = E_s$	$1.08\left(\frac{E_p}{E_s}\right)^{0.21}$	$0.16\left(\frac{E_p}{E_s}\right)^{0.75}$	$-0.22\left(\frac{E_p}{E_s}\right)^{0.50}$

Here, k_{uu} applies to the stiffness of the two horizontal translational springs, $k_{\theta\theta}$ concerns the two rotational springs and $k_{u\theta}$ is the off-diagonal term. Moreover, z is the actual pile depth while d is the pile diameter. E and E_p are the Young's modulus of the soil model and of the pile material, respectively. E_s is the Young's modulus of the soil at a depth equal to the pile diameter.

8.3 Variation of Foundation Stiffness

To investigate how the stiffness of the foundation influences the response of the wind turbine, three different sets of stiffness coefficients were applied for the springs. One of them, *Case1*, uses the stiffness matrix already described in Section 8.2. In addition, one model representing a softer foundation and one model representing a stiffer foundation are used. They were obtained by multiplying the foundation stiffness coefficients of Case1 with factors of 0.6 and 1.4. Subsequently these are called *Case06* and *Case14*, Case06 being the softer one. An alternative way of varying the foundation stiffness matrix that also could be used, is by varying the soil and pile properties using one of the pile stiffness expressions from Eurocode 8, Part 5 [9] as described in Section 8.2.1.

All three models were run with equivalent input so as to enable comparison. The inputs applied are those that were already presented in Chapter 7. That is, full-field wind files with a mean velocity of U=12 m/s and irregular JONSWAP spectrum waves with T_p=12 s and H_s=5 m were applied. They both act mainly along the positive direction of the x-axis. Time series of the wind velocity and the wave elevation are given in Section 7, in Figures 7.1 and 7.2, respectively.

As before, simulation time was set to 630 s, or 10 min and 30 s. The time histories plotted start after 30 seconds has passed, to ignore the transient period associated with applying loads from a zero start. This way, only steady-state time histories are considered.

8.3.1 Results

The combined wind and wave load case resulted in the tower-top deflection along the x-axis plotted in Figure 8.3 for each of the cases.

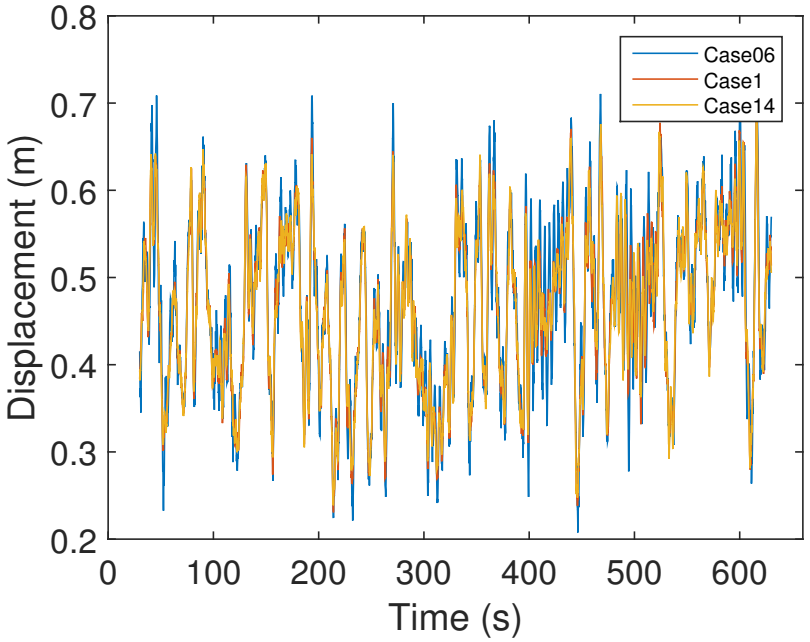

Figure 8.3: Tower-top fore-aft displacement for the various stiffness matrices.

It is clear from the graph above that the model with the lowest stiffness foundation flexibility, namely Case06, experiences the largest tower-top displacements. On the other hand, the stiffest foundation, Case14, observes the smallest amplitudes.

In Figure 8.4, a smaller time window of the tower-top displacement is shown.

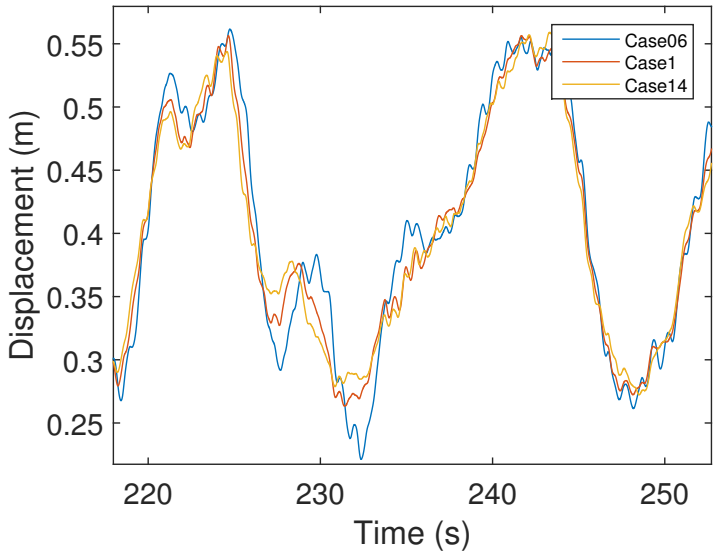

Figure 8.4: Time window of the tower-top fore-aft displacement.

Observing the graph in Figure 8.4, one can see how Case06 continuously experiences larger amplitudes, while Case1 and Case14 seem to have several smaller amplitudes. The fact that the curves of the two latter cases lie so close together implies that the foundation stiffness from Section 8.2 was already large, so that increasing it lead only to a small change in influence on the structural response.

Time histories for the fore-aft bending moment and shear force at the mudline are also plotted. They can be seen in Figure 8.5 and Figure 8.6, respectively.

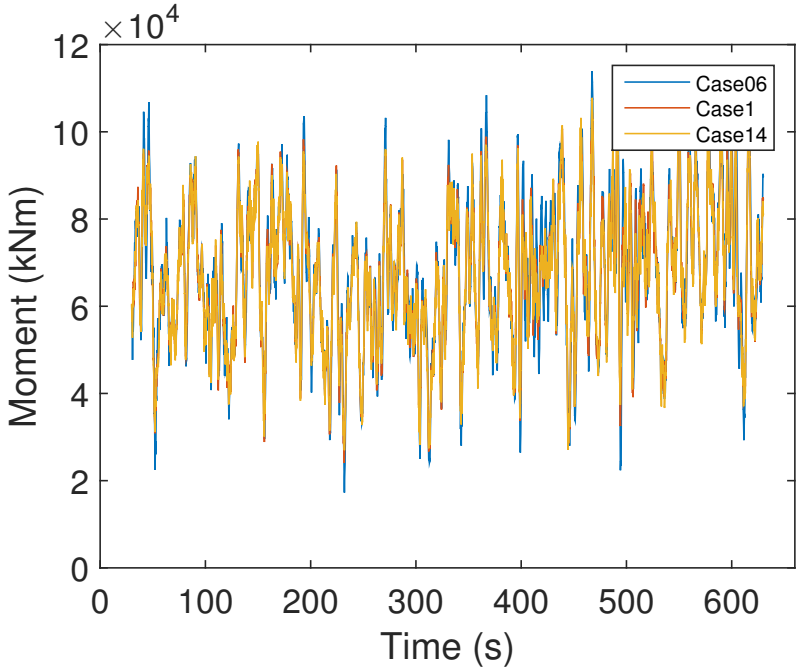

Figure 8.5: Bending moment at mudline for the various stiffness matrices.

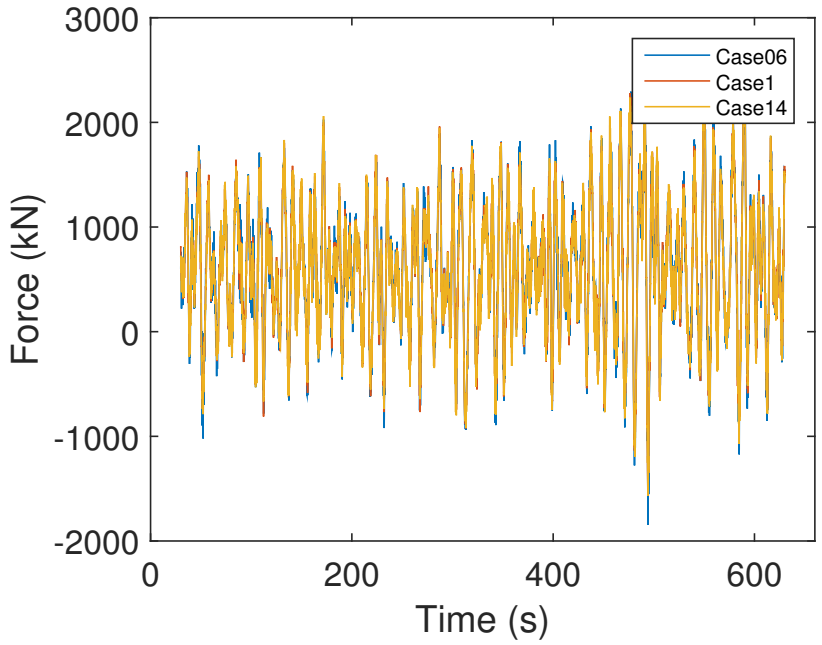

Figure 8.6: Shear force at mudline for the various stiffness matrices.

Differences between the models are most evident in the plot of the bending moment, where the softest foundation provides the largest bending moments.

Zooming in on a smaller time window of the shear force time history, see Figure 8.7, high frequency oscillations are evident for the models with stiffer foundations.

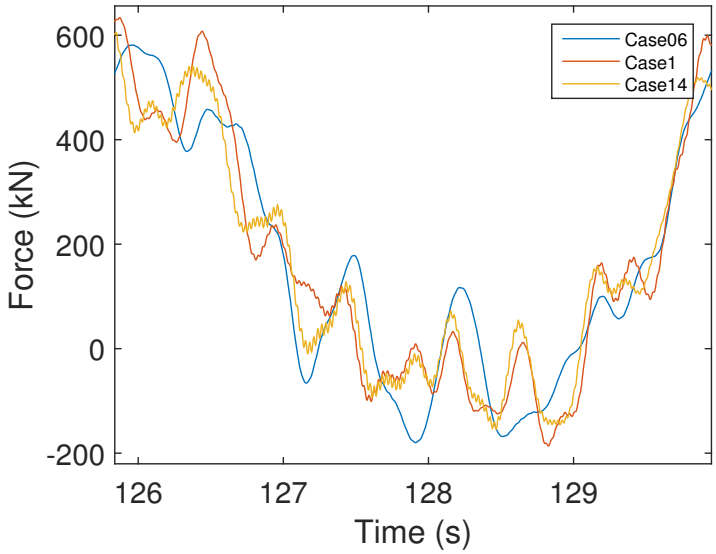

Figure 8.7: Time window of shear force at mudline for the various stiffness matrices.

These oscillations do not occur for the softer foundation model. To explain this phenomenon, consider a simple two-dimensional vertical beam with stiffness k_s. It is constrained against vertical movement, and has a horizontal and a rotational spring attached at the bottom, with stiffness k_{uu} and $k_{\theta\theta}$, respectively. Then, according to Kramer [22], the equivalent natural frequency of the complete system is given as

$$\omega_e = \frac{\omega_0}{\sqrt{1 + \frac{k_s}{k_{uu}} + \frac{k_s h^2}{k_{\theta\theta}}}} \tag{8.10}$$

where ω_0 is the natural frequency of the beam. Hence, when the springs get stiffer, i.e. k_{uu} and $k_{\theta\theta}$ increase, the natural frequency of the system increases and the eigenperiods are decreased. On the other hand, when the foundation gets softer, the natural

frequencies of the structure are decreased. As can be seen in Figure 2.6 in Chapter 2 they will then move closer to the excitation frequency of external loads. The latter explains why larger bending moments, shear forces and displacements occur for the softer foundation.

8.3.2 Summary

In short, lower values for the foundation stiffness matrix imply a softer foundation. The model with a softer foundation will in general experience larger tower-top displacements and bending moments because the natural period will approach the excitation frequencies of the environmental loads, see Equation (8.3.1). Because of the same relation, high frequency oscillations are observed for the stiffest foundation, whereas the softer foundation experiences larger amplitudes for the lower frequencies. Moreover, increasing the foundation stiffness presented in Section 8.2 with a factor of 1.4 will not change the behavior of the wind turbine significantly, implying that the foundation was already quite stiff.

All of these results were expected, and gives an indication of how the foundation derived by Passon [30] behaves, and what the main effects of a softer or stiffer foundation are. It also shows that the implementation of CLS in FAST works as planned.

Chapter 9

Implementing a Nonlinear Spring in FAST v7

The main focus of this master thesis was to implement a nonlinear foundation flexibility in FAST v7. This was done by applying nonlinear properties to an uncoupled version of the coupled foundation stiffness matrix described in Chapter 8. In general, the loads and displacements in the springs will be nonlinearly related, i.e. the stiffness of the spring depends on the load and displacement of which it undergoes.

9.1 The Parallel Springs Model

Iwan [12] presents an approach for modeling the yielding behavior of materials and structures by means of parallel springs. The method is briefly presented in this section.

By applying several, parallel elastic perfectly-plastic springs at the mudline, one can obtain a smooth and realistic load response to the deflection, see Figure 9.1. Here, F represents either a moment or a force, and correspondingly δ may represent a rotation or a translation.

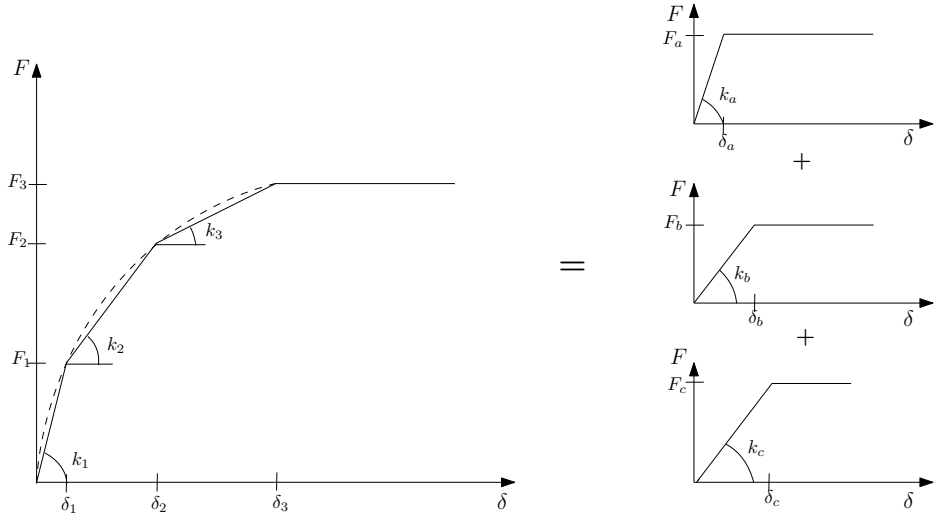

Figure 9.1: Representation of nonlinear foundation by use of parallel elastic perfectly-plastic springs.

Given that

$$\delta a < \delta b < \delta c \tag{9.1}$$

the following equations apply:

$$\begin{aligned} k_1 &= k_a + k_b + k_c \\ k_2 &= k_b + k_c \\ k_3 &= k_c \end{aligned} \tag{9.2}$$

$$\begin{aligned} F_1 &= F_a + k_b\delta_a + k_c\delta_a \\ F_2 &= F_a + F_b + k_c\delta_b \\ F_3 &= F_a + F_b + F_c \end{aligned} \tag{9.3}$$

$$\begin{aligned} \delta_1 &= \delta_a \\ \delta_2 &= \delta_b \\ \delta_3 &= \delta_c \end{aligned} \tag{9.4}$$

When unloading from F_3, the slope of the curve will again start with a slope of k_1. When the unloading covers a displacement of $2\delta_1$, the slope changes from k_1 to k_2. When the unloading in total covers $2\delta_2$, the slope changes from k_2 to k_3. From there the slope has a constant value of k_3 until -F_3 is reached. This is illustrated in Figure 9.2.

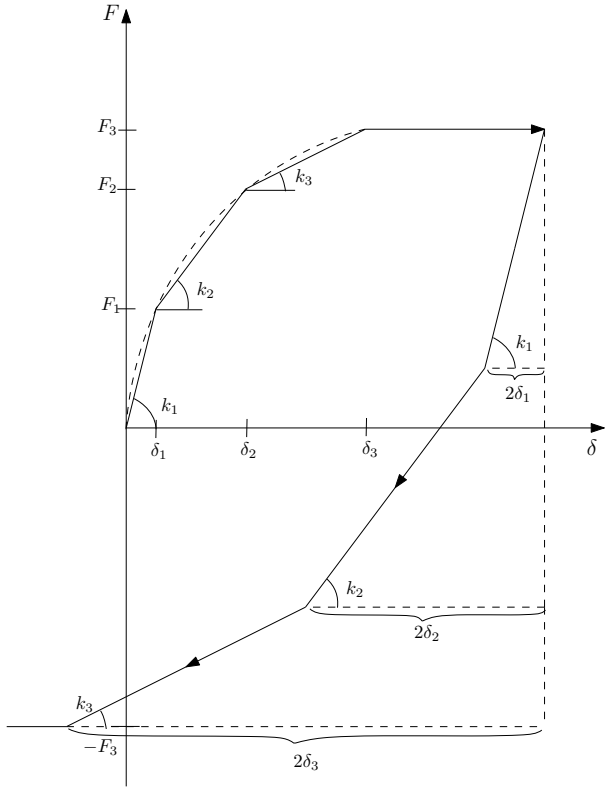

Figure 9.2: Unloading of nonlinear foundation.

This approach was implemented in FAST using 20 parallel springs, varying their yield load and stiffness. The restrictions on these parameters were given by Equations (9.1), (9.1) and (9.1).

The shape of the backbone curve is given by the hyperbolic function as

$$F = \frac{k_{max}\delta}{1 + \frac{k_{max}\delta}{F_{max}}} \tag{9.5}$$

where the parameters are as presented in Figure 9.3.

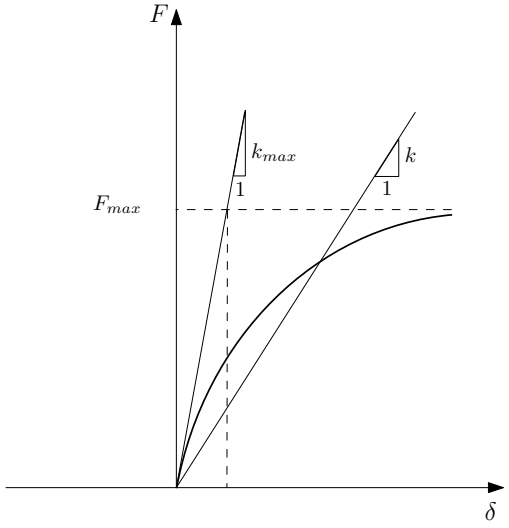

Figure 9.3: Hyperbolic backbone curve.

9.1.1 Values for the Parameters of the Backbone Curve

The initial stiffness k_{max} for each DOF, was taken from the stiffness matrix presented in Chapter 8. To obtain a reasonable value for the yield moment, a simulation using the NREL 5MW with uncoupled springs at the bottom was made. The same full-field wind and irregular waves as presented in Chapter 7 were applied, and simulation time was set to 630 s. The maximum moment caused by the loads was thereby found to be $M_{max,linear}$=109,000 kNm, omitting the first 30 s of the simulation for the same reasons as described in Chapters 7 and 8. To prevent the springs from getting too close to yield, which is not a realistic scenario, the maximum value of the backbone curve was set to M_{max}=5$M_{max,linear}$=545,000 kNm. Here, M_{max} corresponds to F_{max} in Figure 9.3 and δ is equivalent to the rotation, θ, of the spring. Maximum horizontal forces were set equal to the maximum moments, to avoid yield and maintain linear translational

springs. This model will subsequently be referred to as the *NL* (nonlinear) model. A program using a reduced yield of M_{max}=3M_{max}=327,000 kNm was also compiled, for comparison. This program will be named NL_{red} in the following text.

9.2 Introducing the Model in FAST

As described in Chapter 8, NREL provides a subroutine for implementation of linear springs at the bottom of the wind turbine model, namely *UserPtfmLd*. This subroutine was used as a template for the new nonlinear foundation representation.

The code includes nonlinear springs in both lateral translational DOFs and both lateral rotational DOFs. For this thesis, nonlinear springs are most relevant in the rotational DOFs, because at the bottom of the wind turbine large moments are dominating compared to lateral forces. Therefore, the yield forces for the translational forces are set so high that the translational springs will behave linearly.

9.2.1 FAST Foundation Representation

In this subsection, a brief review of the way the foundation stiffness is represented by springs in FAST, is given.

As an example, consider the two-dimensional, one-element beam with a rotational spring at the soil intersection and an external load at the top, see Figure 9.4.

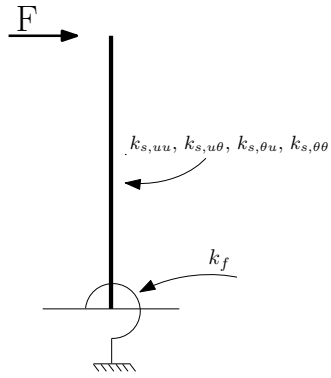

Figure 9.4: Two-dimensional one-element beam.

The equilibrium equation is given as

$$\begin{bmatrix} k_{s,uu} & k_{s,u\theta} \\ k_{s,\theta u} & k_{s,\theta\theta} + k_f \end{bmatrix} \begin{bmatrix} u \\ \theta \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \tag{9.6}$$

where $k_{s,uu}$, $k_{s,u\theta}$, $k_{s,\theta u}$ and $k_{s,\theta\theta}$ are stiffnesses related to the structure, whereas k_f is the rotational spring stiffness, i.e. the stiffness related to the foundation. In combination they constitute the stiffness matrix for the whole assembly, seen to the very left in Equation (9.2.1). u and θ are translational and rotational displacements. Let the external force vary in time. Rearranging the terms in Equation (9.2.1), the following equations are obtained:

$$\begin{aligned} k_{s,uu}u + k_{s,u\theta}\theta &= F \\ k_{s,u\theta}u + k_{s,\theta\theta}\theta &= -k_f\theta \end{aligned} \tag{9.7}$$

Now, denote the displacements from the previous time step u_{t-1} and θ_{t-1}, whereas displacements in the current time step are denoted u_t and θ_t.

$$\begin{bmatrix} k_{s,uu} & k_{s,u\theta} \\ k_{s,\theta u} & k_{s,\theta\theta} \end{bmatrix} \begin{bmatrix} u_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} F_t \\ -k_{f,t-1}\theta_{t-1} \end{bmatrix} \tag{9.8}$$

Accordingly, FAST treats the foundation term as an external load calculated based on the previous displacement. It should be noted that these equations are for a simplified system with one beam element. For a large wind turbine model, the stiffness matrix would be much larger, including the DOFs in all direction for the foundation stiffness and beam stiffness. However, the idea is the same.

When the foundation stiffness is nonlinear, this approach is somewhat debatable. Chopra [5] describes how numerical evaluation of nonlinear systems should be treated. Ideally, for each time step the stiffness matrix of the whole assembly should be updated according to the previous stiffness. Then, the equilibrium equation should be solved for the current displacements based on the updated stiffness matrix and the current external loads. The reason why such a method was not used in this thesis is that the structure of

FAST is not consistent with that type of incremental solution. If the nonlinear foundation was to be implemented rigorously correct, a fundamental change in the algorithm of FAST would how to be made. Moreover, for small nonlinearities the method applied in this thesis should be acceptable.

9.2.2 FAST Source Code Modifications and Changes

As mentioned above, the subroutine *UserPtfmLd* was changed to represent a nonlinear foundation in terms of parallel springs. While the procedure applied is only briefly presented here, a more thorough description, together with the Fortran codes used, are available in Appendix F.

The Structure of FAST

FAST_Prog.f90 is the actual FAST Program, where calls to different subroutines and modules are made. In short, it opens and reads input files, sets up the initial values for all the DOFs, decides what analysis mode is to be run, runs it, and then ends the program. For the time-marching analysis mode, FAST_Prog.f90 calls the subroutine *TimeMarch*. TimeMarch controls the execution of a typical time-marching simulation of the FAST code. It contains an *infinite DO-loop* in which a call to a predictor-corrector subroutine, namely *Solver* is made. After each call to Solver, an advancement in time is made by the statement

$$\begin{aligned} Step &= Step + 1 \\ ZTime &= Step \cdot DT \end{aligned} \tag{9.9}$$

After this is done, all outputs are calculated in another subroutine, *CalcOuts*, using information obtained in the Solver subroutine. The infinite DO-loop is exited when TMax is reached. Both of these routines are located in FAST.f90.

For the purpose of implementing foundation flexibility in FAST, Solver is the subroutine that is interesting. It "solves the equations of motion by marching in time using a predictor-corrector scheme. Fourth order Runge-Kutta is used to get the first four points from the initial degrees of freedom and velocities. Adams-Bashforth predictor

and Adams-Moulton corrector (ABAM) integration scheme are used for all other time steps" [20]. So, for each time step, Solver solves the equations of motion by means of estimating next displacements and velocities in all DOFs through one of the above mentioned integration methods. The current estimations are correspondingly called QT and QDT. These are fed into the subroutine *RtHS*, which calculates the equations of motion for a particular time step, and outputs an estimate for the acceleration vector. For each time step, these iterations are made four times when using Runge-Kutta, and one time (predictor and corrector) for the ABAM integration method. Then, a best estimate for the displacement and velocity vector is obtained, leading to a best estimate of the acceleration vector. RtHS calls many other subroutines. One of them is the subroutine *PtfmLoading*, which checks what platform model is supposed to be used. In the current scenario, the user-specified subroutine UserPtfmLd should be applied as a platform model. This is the subroutine where the nonlinear foundation has been implemented. Also, adjustments had to be made to Solver, and to the modules *Platform* and *RtHndSid*. Finally, the UserPtfmLd subroutine had to be introduced in the UserSubs_forBladedDLL.f90 source file.

UserPtfmLd

For each of the 20 elastic perfect-plastic springs, an uncoupled stiffness matrix and a yield force or moment was defined. The modified subroutine takes in the following variables:

Table 9.1: Input variables for UserPtfmLd subroutine.

X(6), XD(6)	Vectors containing current estimate for platform displacements and velocities.
QTC(6), QDTC(6)	Vectors containing platform displacements and velocities from previous time step.
PtfmFtC1(6), PtfmFtC2(6), ..., PtfmFtC20(6)	Vectors containing forces and moments experienced by each spring in the previous time step.

The outputs from the subroutine are:

Table 9.2: Output variables for UserPtfmLd subroutine.

PtfmFt1(6), PtfmFt2(6), ..., PtfmFt20(6)	Vectors containing forces and moments experienced by each spring in the current time step.
PtfmFt(6)	Vector containing total forces and moments experienced by the platform in the current time step. This is the sum of vectors PtfmFt1, PtfmFt2,..., PtfmFt20.

As seen in Tables 9.1 and 9.2, in general the subroutine takes in forces and moments experienced by each of the springs in the last time step, together with previous and current estimates of displacements and velocities of the platform reference point. Using this information, the code estimates the current load in each of the elastic perfectly-plastic parallel springs. These forces are summed up in the vector PtfmFt, which is output from the subroutine. To keep track of the status of each of the springs, current loads for each of the springs are also outputted and saved. Thereby, information about the loads in each spring at the current time step is available in the next time step.

Interaction Between UserPtfmLd and Remaining Part of FAST

As described previously, status of each of the springs are kept by means of loads in each spring, and displacements and velocities for the platform from the previous time step. To manage this, the loads experienced by the springs are outputted from UserPtfmLd as PtfmFt1, PtfmFt2 and so on. These loads are saved in the Platform module, together with the total load on the platform, PtfmFt. PtfmFt is the only output used by the rest of the program. After each time step, in Solver, these loads are renamed PtfmFtC1, PtfmFtC2, etc. These variables are stored in the RtHndSid module. When the next time step is encountered, these forces are thereby available for the next calculations.

A complete description of the codes and changes can be found in Appendix F.

9.3 Assumptions and Simplifications

As described in Chapter 3.2, a rigid link should be applied when going from a coupled to an uncoupled stiffness matrix. Since the length of the rigid link depends on the stiffness of the foundation, for a nonlinear foundation the rigid link approach is difficult to handle. Iteration processes are usually required. One way of dealing with it is presented by Carswell et al. [4]. There, a finite element program is applied to obtain an appropriate length based on small load perturbations. For this thesis, a rigid link for a nonlinear foundation is out of scope. Hence, to simplify the simulations, in the following the length of the rigid link is set to zero.

Moreover, mode shapes of the tower should be obtained based on its boundary conditions, which include the foundation stiffness, see Section 8.1.2. For a nonlinear foundation, in theory the mode shapes of the tower should be updated for each new tangent stiffness of the foundation. However, this is not the focus of this thesis, and for the following analyzes, the mode shapes of the tower are based on the initial foundation stiffness.

9.4 Results

9.4.1 Verification

To verify that the nonlinear foundation was appropriately implemented, the moment in the rotational springs was plotted against the corresponding rotation. To get a clear picture of expected hysteresis curves, a harmonic wind load was used. Moreover, no waves were applied. The wind acts in the x-direction, therefore only the rotation of the platform about the y-axis is considered.

For a first visualization of the behavior of the foundation, a wind load that oscillates harmonically around zero was used. The values of the amplitude and period were set to mimic realistic behavior of wind. Periods of 30 s were applied, with velocity amplitudes ranging from 5 m/s to 10 m/s, see Figure 9.5.

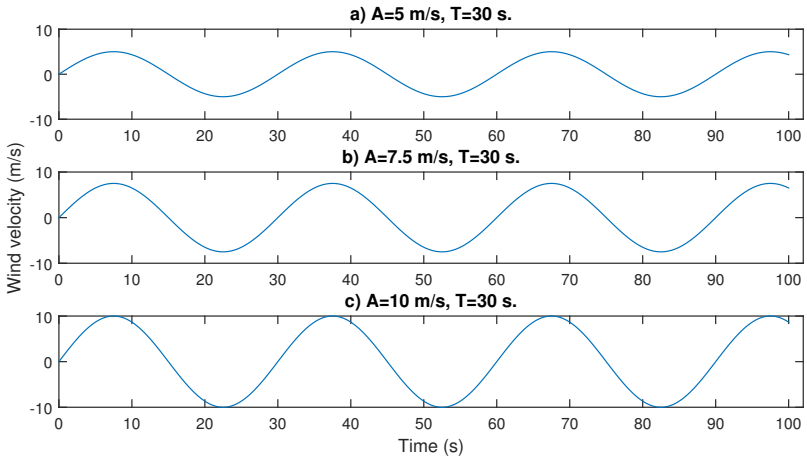

Figure 9.5: Harmonic wind loads.

In Figures 9.6, 9.7 and 9.8 are plots of the hysteresis curves for NL with the harmonic loads in Figure 9.5a), b), and c), respectively. As a comparison, according to Lombardi et al. [24] wind turbines typically should not be allowed a tilt of more than 0.5°. This calls for the need for software that models the turbine behavior correctly.

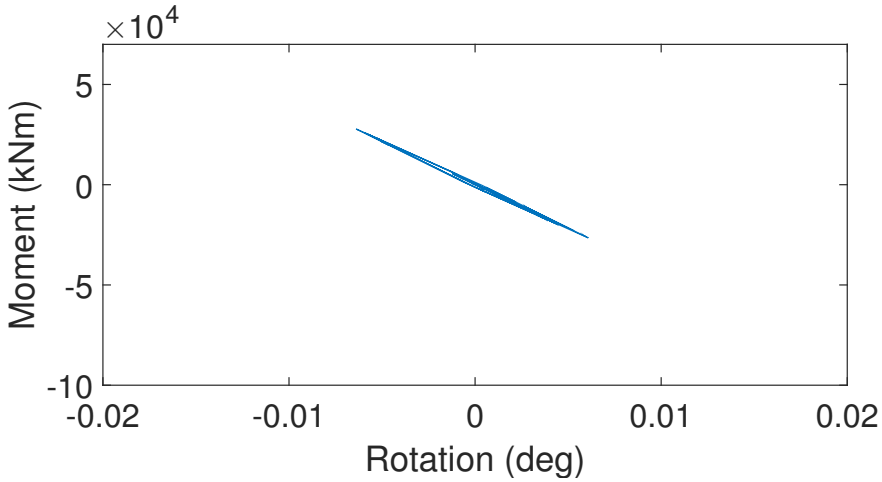

Figure 9.6: Hysteretic curve for the NL model subjected to wind load in Figure 9.5 a).

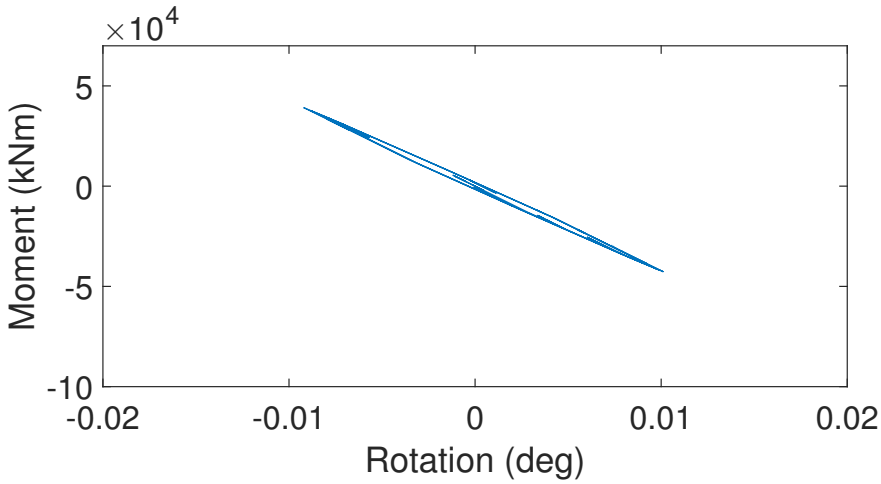

Figure 9.7: Hysteretic curve for the NL model subjected to wind load in Figure 9.5 b).

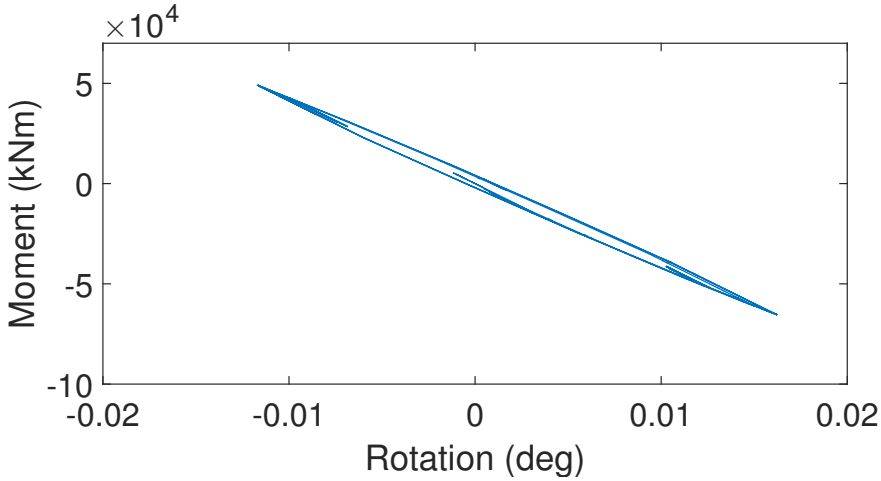

Figure 9.8: Hysteretic curve for the NL model subjected to wind load in Figure 9.5 c).

The shape of the hysteresis curves above clearly shows that a nonlinear behavior is successfully implemented. The area under the loop represents damping, as described in Section 3.3. Moreover, these hysteresis loops are related to fatigue as given by Equation (2.5) in Section 2.5. Note that these are hysteresis loops for the springs, not the tower. That is, a positive displacement in the spring will lead to a negative load. This is the reason why they are opposite of what the hysteresis curves of an inelastic material usually are.

For verification, the load displacement curve of a model with *uncoupled linear springs* (UCLS) subjected to the wind load in Figure 9.5c) is plotted in Figure 9.9.

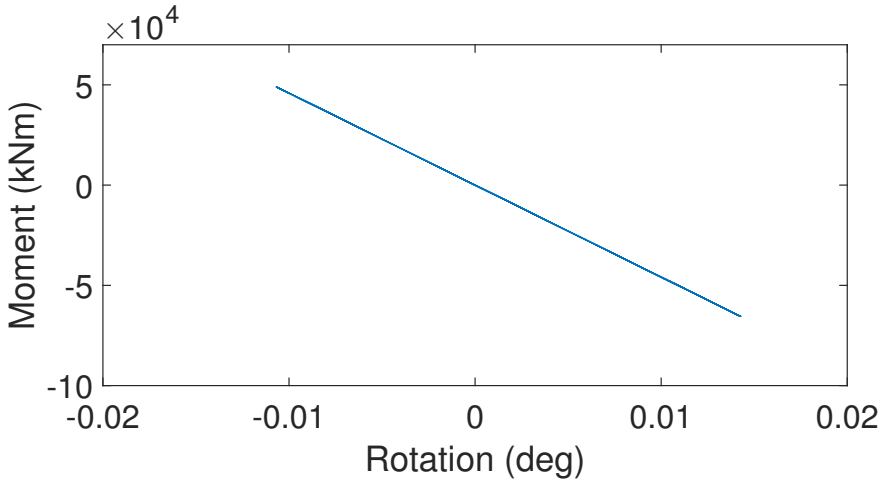

Figure 9.9: Moment plotted against rotation for the model with uncoupled linear springs subjected to the load in Figure 9.5c).

It is clear that the linear model does not exhibit the same hysteresis loop.

For each of the wind loads in Figure 9.5, hysteresis curves are plotted in the the graph in Figure 9.10 for NL_{red} as well.

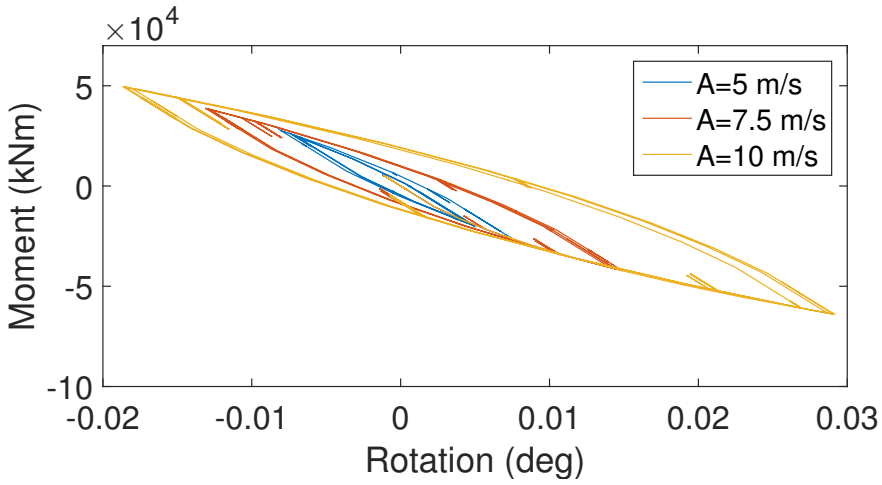

Figure 9.10: Hysteretic curve for the NL_{red} model subjected to wind loads in Figure 9.5.

NL_{red} exhibits a softer foundation where nonlinearities are observed at a stage of smaller rotations. Correspondingly, the area under the hysteresis loop is larger, indicating more damping, as described in Section 3.3.

For a load closer to the natural frequencies of the turbine, namely with a period of T=10 s, the nonlinearities are more visible in the hysteresis curves plotted in Figures 9.12, 9.13 and 9.14 for the NL model. The wind loads applied are illustrated in Figure 9.11.

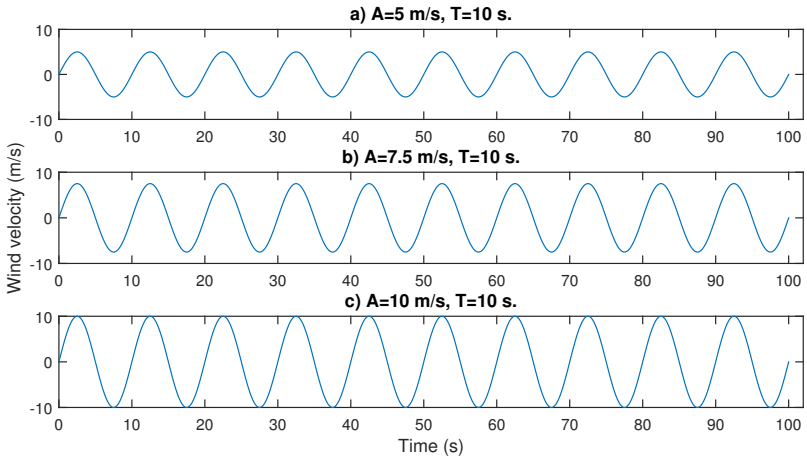

Figure 9.11: Harmonic wind loads.

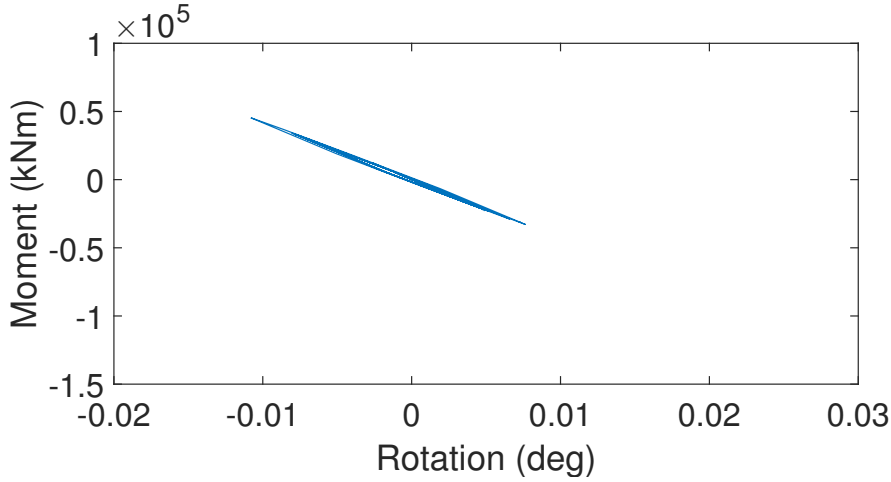

Figure 9.12: Hysteretic curve for the NL model subjected to wind load in Figure 9.11 a).

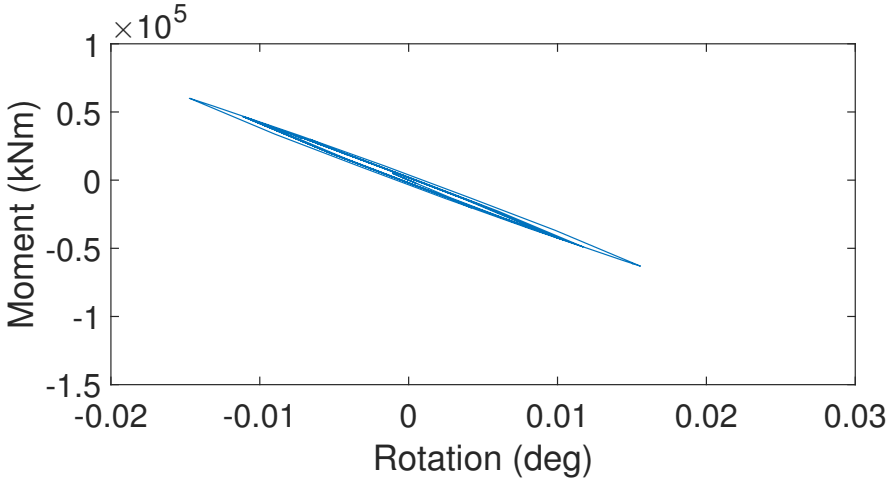

Figure 9.13: Hysteretic curve for the NL model subjected to wind load in Figure 9.11 b).

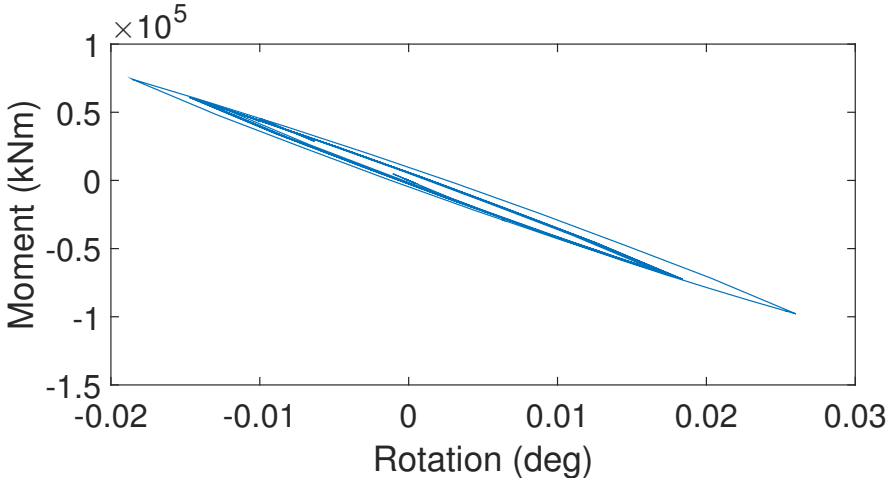

Figure 9.14: Hysteretic curve for the NL model subjected to wind load in Figure 9.11 c).

As can be seen in Chapter 7, in practice, the wind velocity will oscillate around a mean wind speed of some value larger than zero. To obtain more realistic results while still being able to observe the hysteresis curves clearly, simulations were made for the NL model with harmonic wind having a mean velocity of 10 m/s, a period of 30 s and the same variety of amplitudes as previously, see Figure 9.15

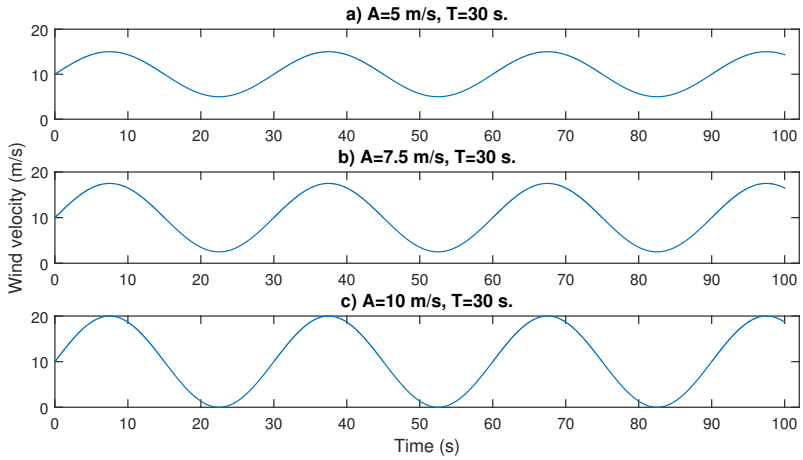

Figure 9.15: Harmonic wind loads. Mean velocity of 10 m/s.

Due to the larger mean velocity in one direction, larger rotations can now be observed, see Figures 9.16, 9.17 and 9.18.

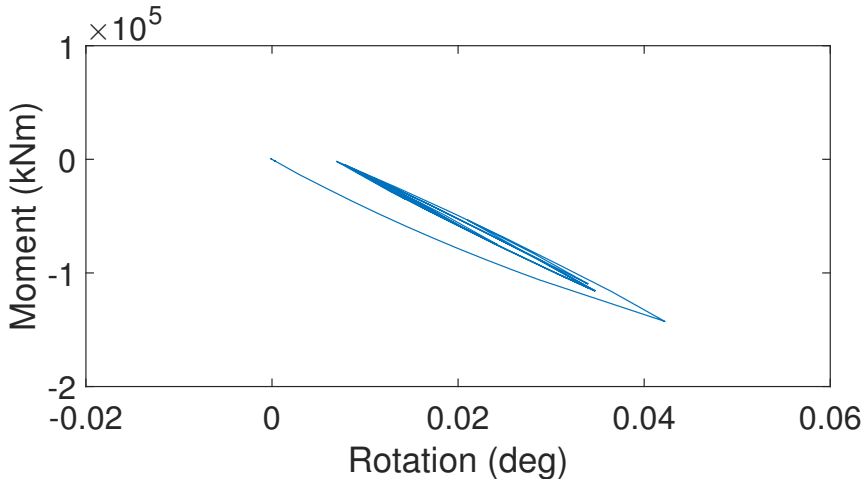

Figure 9.16: Hysteretic curve for the NL model subjected to wind load in Figure 9.15 a).

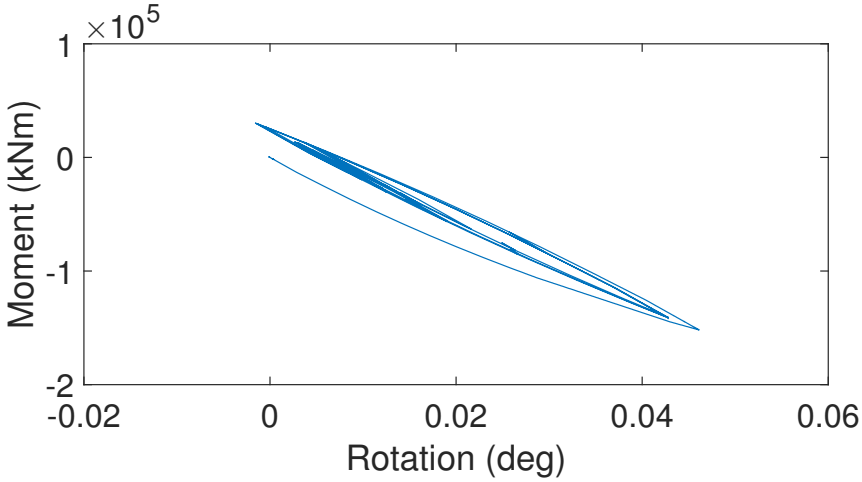

Figure 9.17: Hysteretic curve for the NL model subjected to wind load in Figure 9.15 b).

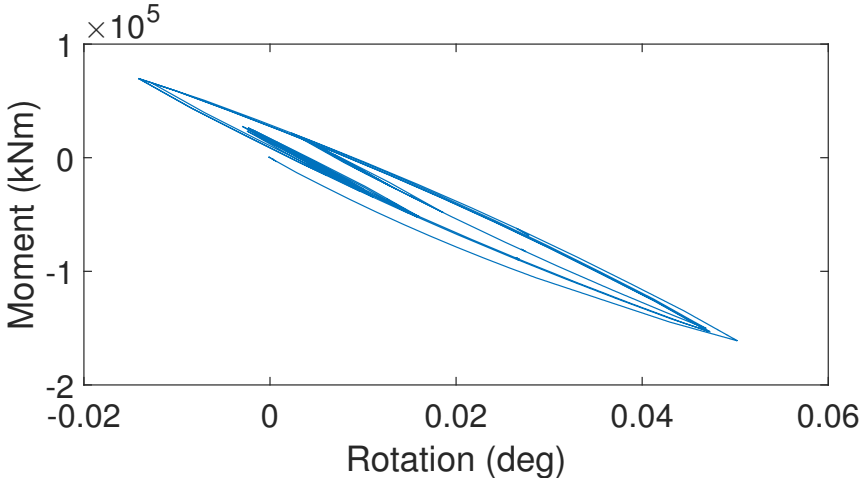

Figure 9.18: Hysteretic curve for the NL model subjected to wind load in Figure 9.15 c).

The same procedure presented above was applied for a wind load with a period of 10 s, see Figure 9.19.

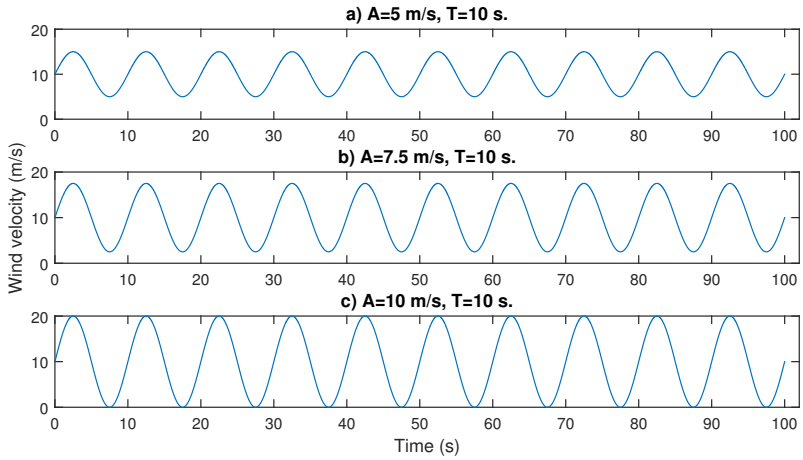

Figure 9.19: Harmonic wind loads. Mean velocity of 10 m/s.

The results may be observed in Figures 9.20, 9.21 and 9.22. For a period of 10 s the rotations are even larger, due to the fact that the period of the load is now closer to the natural periods of the turbine.

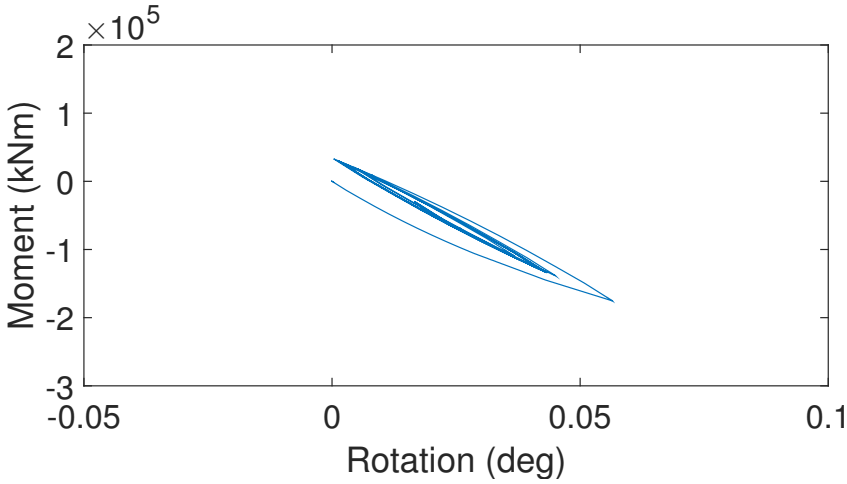

Figure 9.20: Hysteretic curve for the NL model subjected to wind load in Figure 9.19 a).

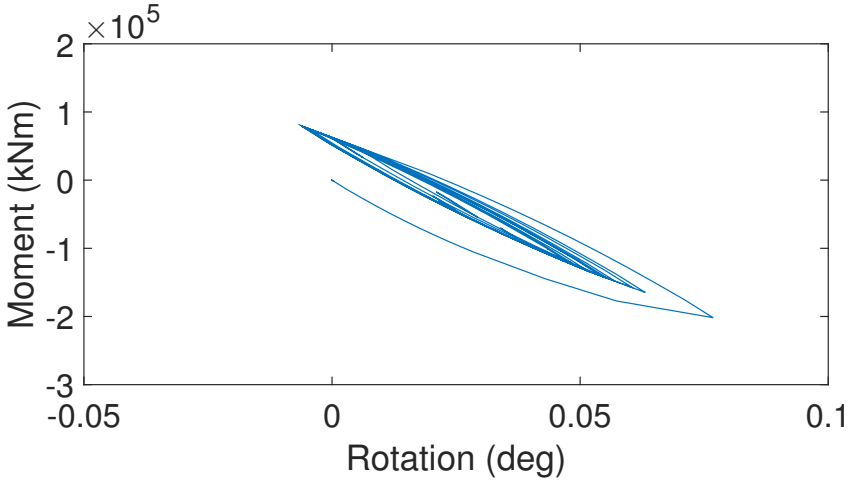

Figure 9.21: Hysteretic curve for the NL model subjected to wind load in Figure 9.19 b).

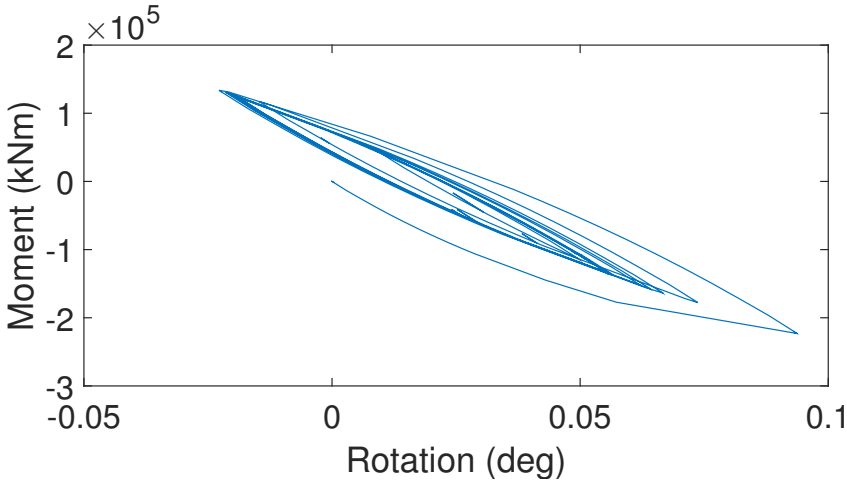

Figure 9.22: Hysteretic curve for the NL model subjected to wind load in Figure 9.19 c).

As mentioned in Section 2.5, fatigue is described by the number of cycles enclosed by hysteresis loops. The curves presented above may therefore be an indication of the fatigue damage in the structure.

Zooming in on the graph plotted in Figure 9.18, one can see small branches of loading- and unloading curves. These are shown in Figure 9.23.

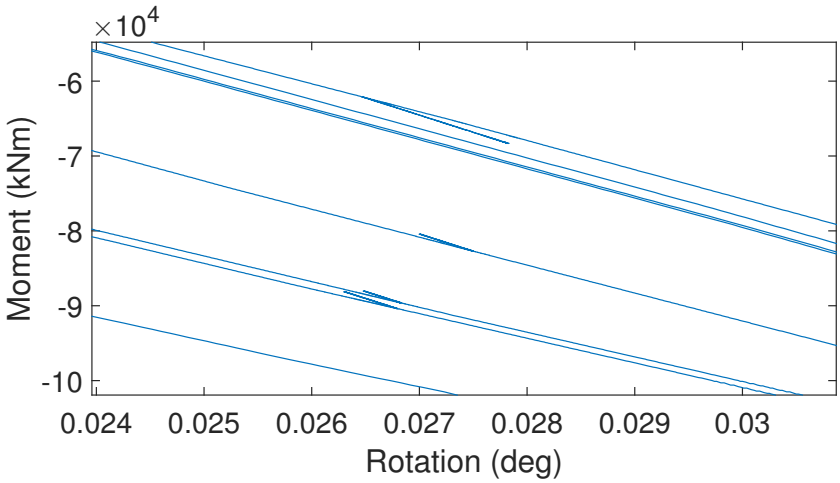

Figure 9.23: Close up from Figure 9.18.

These small cycles have a stiffness corresponding to the initial stiffness of the moment-rotation curve, i.e. they are linear. Whereas the large cycles of the hysteresis represent the way the structure moves with the load, the smaller cycles enclosed are related to short period vibrations. The periods of these cycles may be associated with the natural periods of the structure. While their amplitudes are small, they may be many, and thereby be significant to fatigue damage. With the correct tools, which will not be further investigated here, this information may be significant when fatigue damage is examined. Moreover, the fact that these cycles are seen is yet another verification of the successful implementation of the foundation, because the code is supposed to see these higher frequency oscillations.

9.4.2 Damping

Since the high frequency cycles in Figure 9.23 are linear, they do not exhibit damping. Therefore, to obtain more realistic models, damping was implemented for the foundation models. This was done both for the UCLS model and the NL model. Although damping is already represented in the hysteresis loop of the load-displacement curve of the NL model, this incorporation was made to account for damping that occurs when the nonlinear springs are in the linear regime. The load cases applied will in general be

acting in the x-direction. Therefore, as a simplification, all foundation damping here has been assigned to the rotational DOF about the y-axis.

Material damping is frequency dependent and may be given through the relationship in Equation (9.4.2) [5]. The damping coefficient is an indication of the energy dissipated in a cycle of vibration.

$$c_{\theta\theta} = \frac{2k_{\theta\theta}\xi}{2\pi f} \tag{9.10}$$

$c_{\theta\theta}$ is the damping coefficient in the fore-aft rotational dof. $k_{\theta\theta}$ is the corresponding stiffness coefficient. f is taken as the natural frequency of the first fore-aft bending mode, since this is the most dominant fore-aft bending mode. ξ is the damping ratio. According to Carswell et al. [4], experiments have obtained estimates of the damping ratio of everything from 0.25% to 1.5%. Here, to obtain merely an indication of its influence, it is taken as 1.0%. The natural frequency of the first fore-aft bending mode for the wind turbine model with UCLS is 0.2670 Hz. $k_{\theta\theta}$ is taken as the rotational stiffness coefficient of the linear spring in Section 8.2, that is, $k_{\theta\theta}$=2.63 $\cdot 10^{11}$Nm/rad.

Using these values, a damping coefficient of $c_{\theta\theta}$=3.14 $\cdot 10^{9}$Nms/rad is obtained.

Implementation of Foundation Damping in FAST

To include damping in FAST, the code in the subroutine UserPtfmLd had to be rewritten. This applies both to the subroutine used for the uncoupled linear springs, and the one used for nonlinear springs.

As mentioned in Section 8.1, NREL provides a template for including damping. This template is applied for both cases, and is in short presented here. In addition to the load provided by the springs attached to the platform, a load provided by a rotational dashpot is hereby also modeled. The moment applied by the dashpot is equal to the negative product of the damping coefficient $c_{\theta\theta}$ just described, and the rotational ve-

locity $\omega_{platform}$ with which the platform is moving:

$$M_{dashpot} = -c_{\theta\theta}\omega_{platform} \tag{9.11}$$

The total moment on the platform from the foundation is then given as

$$M_{total} = M_{dashpot} + M_{springs} \tag{9.12}$$

The complete Fortran code for the implementation of damping is given in Appendix F for the NL model, and in Appendix E for the UCLS model.

Hysteretic Curve

The hysteretic curve for the NL model with and without additional damping, and for the UCLS model with damping, is plotted in Figure 9.24. The load applied is the one presented in Figure 9.15c). As seen in Section 9.4.1, the moment-rotation curve of the UCLS model without damping only followed a single, straight line.

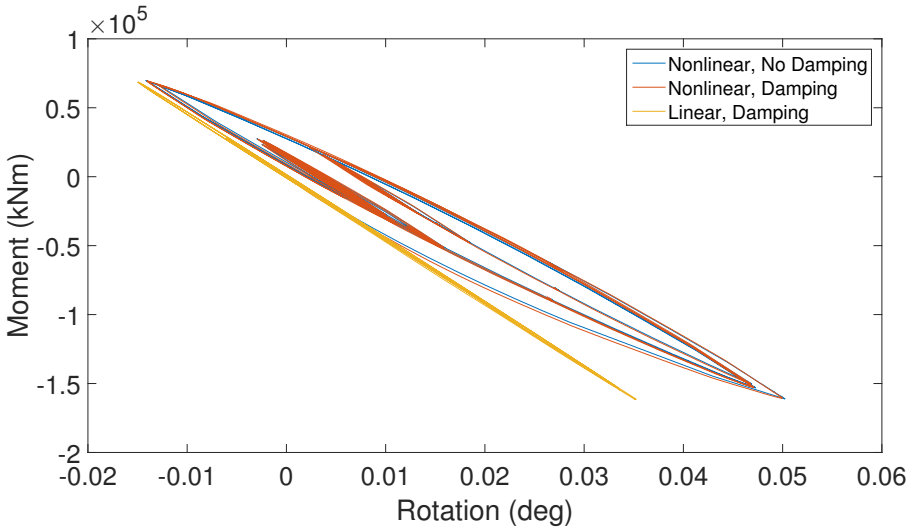

Figure 9.24: Hysteretic curve, i.e. moment and rotation about the y-axis at the bottom of the monopile.

It is evident that for the curve representing the foundation exhibiting both damping and nonlinear stiffness, i.e. the *Nonlinear, Damping* curve, more energy is dissipated. That is, the area under the hysteresis loop is larger. This also applies to the UCLS model, where the moment-rotation graph covers an area slightly larger than that of the same model without damping, the latter of which can bee seen in Figure 9.5.

9.4.3 Time Series

Simulations were run using the load case from Figure 9.15 c). The NL model and the UCLS model were used, both with damping included.

Time series of fore-aft tower-top displacement and bending moments were plotted, and are shown in Figures 9.25 and 9.26, respectively.

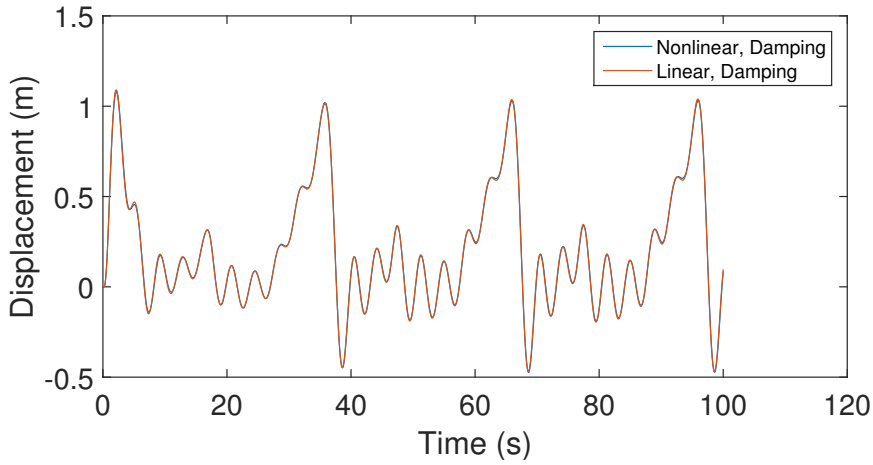

Figure 9.25: Tower-top fore-aft displacement for linear and nonlinear foundation, both with damping included.

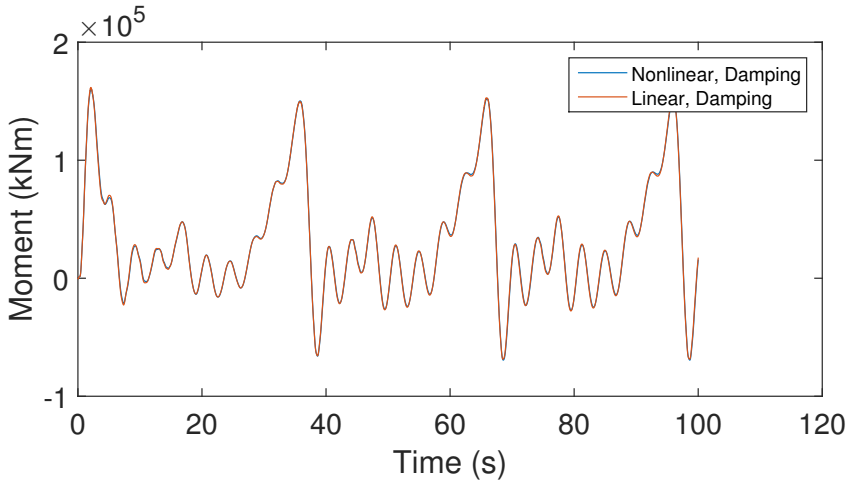

Figure 9.26: Fore-aft bending moment for tower at mudline for linear and nonlinear foundation, both with damping included.

As can be observed in both plots, differences between the nonlinear and linear foundation model are barely there. This is not unexpected, as this load situation is quasi-static. That is, the period of the wind load is much larger than the period of the construction, implying that the tower is merely following the motion of the wind. Hysteretic damping and dynamics is in general not important during operation. However, according to Det Norske Veritas [8], contributions from damping may be significant when the frequencies of steady-state loads are in the vicinity of the natural periods of the structure. This is also applicable during transient loads such as start-up from standstill or from idling, shutdown, yawing, or faults in either control system, electrical network connections, in protection system or in electrical system. The aforementioned situations are characterized by large loads at higher frequencies. This subject will not be discussed further, as the main concern of this section was only to implement the nonlinear foundation with damping, not to investigate how this will influence the turbine response.

However, one aspect that may be interesting is observed when plotting the rotation of the monopile at the mudline level. A comparison of the damped versions of the UCLS model and the NL model can be seen in Figure 9.27. It is evident that the mudline rotation of the NL model consequently lies above the UCLS model.

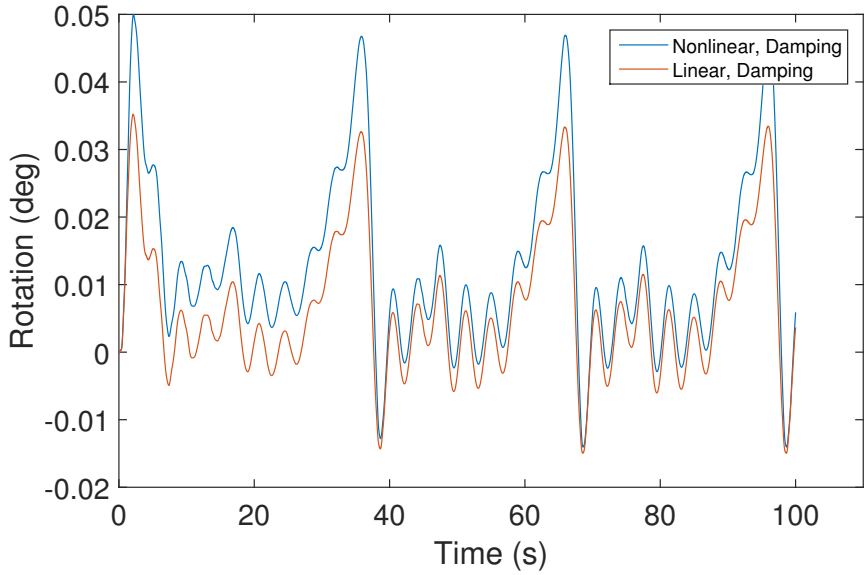

Figure 9.27: Rotation at the bottom of the monopile, i.e. at mudline level.

Considering the rule mentioned in Section 9.4.1, which states that a turbine should not experience a tilt larger than 0.5°, the necessity of modeling the foundation correctly is supported.

9.5 Summary

A nonlinear foundation was implemented in FAST v7 by means of the concept of parallel springs [12]. In addition, damping was introduced in both the nonlinear foundation model, and in a linear foundation with uncoupled springs. Verification of this is made through plots of moment against rotation at the bottom of the fictitious monopile, i.e. at the mudline.

Chapter 10

Summary and Recommendations for Further Work

In this chapter a summary of the thesis will be made. Moreover, recommendations for further work will be given.

10.1 Summary and Conclusions

As is appropriate, the thesis starts with an introduction, where the problem that will be considered is presented, including motivating aspects. As given in Chapter 1, the main intention of this thesis was to implement a nonlinear foundation representation in FAST v7. An OWT with a monopile support structure was considered, and the corresponding NREL 5MW reference turbine was used as a base for the analyses.

The history, current status and terminology of OWTs, in addition to relevant topics such as power generation, natural frequencies and fatigue, is also presented in brief.

FAST v7 is an open source software written in the programming language *Fortran 90*. As a base for the execution of the task of this thesis, a thorough understanding of the FAST v7 source code, and thereby Fortran, was required. Therefore, an introduction

to the features and structure of FAST v7 is given in Chapter 6. NREL provided softwares BModes and TurbSim were also utilized and are briefly mentioned or presented, respectively.

Chapter 3 provides a review of existing approaches for modeling of foundation flexibility, and presents different support structure concepts.

An insight in the way environmental loads such as waves and wind are modeled was needed, to be able to generate suitable load cases for the simulations. Therefore, theory of wind and waves, and the way these may be represented, is given in Chapters 4 and 5, respectively. Due to the large extent of these subjects, aerodynamics and hydrodynamics, i.e. the loads generated on the turbine by these external forces is not considered. However, a brief description on the way FAST treats them is given in Chapter 6.

In Chapter 7 the base model, namely the NREL 5MW reference turbine is presented. Properties of this fictitious OWT are given, and typical inputs and outputs are described. Default modeling of foundation in FAST v7 is by means of a rigid foundation, disregarding soil stiffness and damping. In this chapter, the rigid foundation representation is kept.

However, foundation damping may lead to lower design load estimates. A softer foundation, on the other hand, will reduce the natural periods of the system, shifting them closer to the frequencies of the environmental loads. This may in turn lead to amplified overturning moments at the mudline. Therefore, including soil stiffness and damping in the analyses is important.

Chapter 8 considers a modified version of FAST v7, where *coupled linear springs* (CLS) are applied for the modeling of the foundation flexibility. The derivation of a foundation stiffness matrix is described in Section 8.2. It was obtained by Passon and Kühn [31]. The response of a model with this stiffness is compared with two other models, each having a softer and stiffer foundation, respectively.

Finally, in Chapter 9, a *nonlinear* (NL) foundation is successfully introduced in FAST v7, by means of uncoupled parallel elastic perfectly-plastic springs, as proposed by Iwan [12]. The mathematical model, the implementation into FAST, and simulation outputs are considered. Moreover, constant damping was added in the foundation. To verify

that the implementation was successful, load-displacement curves of the springs are presented, showing the typical hysteresis loops of an inelastic material.

A summary and recommendations for further work is given in the next section.

In the appendices, the applied Fortran Codes are given, as well as representative input files to FAST, TurbSim and BModes.

10.2 Discussion

Considering the given circumstances regarding the construction of FAST source code and the time limits associated with a master thesis, the purpose of this thesis was fulfilled. However, there is room for improvements, which will be investigated in this section. These are all challenges that have been given awareness, but which have been regarded as out of scope for this thesis.

For each time step, the stiffness matrix of the nonlinear foundation is updated by means of change of status of the 20 parallel springs applied at the bottom of the monopile. However, for each of these updates, the stiffness matrix of the foundation should be included in the overall stiffness matrix of the complete wind turbine system, as outlined in Section 9.2.1. Currently, influence of the foundation on the support structure is merely applied as an external force, disregarding the fact that the soil-pile interaction should be considered as a part of whole turbine system. This requires an extensive change in the source code of FAST.

Another subject that is of a challenging character is the application of a rigid link for a nonlinear foundation, as described in Section 9. The rigid link should be applied for uncoupled stiffness matrices so that their response equals that of a coupled stiffness matrix.

Moreover, FAST relies on the user to input mode shapes for the tower and the blades. The mode shapes of the tower depends on its boundary conditions. For a nonlinear foundation, these boundary conditions are updated for each time step. Hence, for a more correct model, the mode shapes of the tower should also be updated for each

time step. BModes is the software that NREL recommends users of FAST to consult for obtaining mode shapes for the tower. NREL has stated that in the future, BModes should be incorporated as a part of FAST. This may introduce the possibility of continuous updating of mode shapes. For small nonlinearities, the inaccuracies related to this may be acceptable, since changes of the foundation stiffness will be minor.

Also, the Fortran code that considers the status of each of the elastic perfectly-plastic springs should be investigated more thoroughly for pitfalls that may not have been discovered during the short time period of which it was written, and for the limited amount of load cases it was tested for. Whereas FAST v7 is perfectly suitable for modeling monopile-supported offshore wind turbines, as discussed in Chapter 6, FAST v8 is the software version of which NREL is focusing on improving. Whether these improvements are crucial regarding OWTs with monopile substructures is not known. However, it would probably be advantageous to implement a nonlinear foundation in FAST v8. NREL has itself stated a wish to extend FAST v8 to include nonlinear foundation, but according to Jonkman [18], they have not received funding yet to do so.

On the positive side, it should be repeated that translational springs may also be nonlinear simply by using a lower limit for the hyperbolic force-displacement curve, and rewriting the code using these values. Moreover, the yield load, initial stiffness, and damping of the foundation may be changed according to the user's preferences.

10.3 Recommendations for Further Work

Based on the challenges described in Section 10.2, some recommendations for possible extensions of the work in this thesis will be presented. As explained, the rigid link approach is not as straightforward when the uncoupled springs are no longer linear. One way of avoiding this problem is by means of a stiffness matrix with off-diagonal items, that is, a coupled stiffness matrix. Nonlinear springs with a coupled stiffness matrix will require a great deal of changes to the code applied for uncoupled springs, so this would probably be a long-term aim.

On the other hand, the rigid link could be dealt with by means of updating it for each

time step. The calculations would then have to be able to exchange information with the FAST program in the same way that HydroDyn and AeroDyn are currently doing. This is also a very ambitious task.

Moreover, it would be advantageous to apply the stiffness matrix of the foundation directly to the global stiffness matrix of the whole system. This will require an even greater understanding of the way the FAST source is build up, and is also suitable to consider a long-term goal.

However, providing a way for the user to specify the stiffness of the foundation in the input files to FAST, instead of rewriting the source code for each new nonlinear stiffness curve, would improve efficiency a lot. This accounts for damping as well. For example, for a hyperbolic shape of the load-displacement curve, the user should ideally be able to specify only initial stiffness and ultimate yield load for the foundation. In this thesis, an Excel sheet was applied to calculate corresponding stiffness and yield load of the 20 parallel springs, which were then manually placed in the source code. It should be possible to implement these calculations in the FAST source code, so that it automatically updates the foundation stiffness. This task is considered as possible over a shorter time period.

Including BModes directly in FAST, so that the user is not required to specify the mode shapes of the tower by first consulting BModes, will save a great deal of work for the user. This applies specifically when working with a variety of foundations. Ideally, BModes would provide tower mode shapes to FAST for every time step, corresponding to the updated nonlinear stiffness of the foundation. This is a task that would probably be best fit for NREL itself, since a great understanding of BModes would also be required.

Research and simulation should be done to obtain information regarding what load situations foundation flexibility and damping are most important.

Bibliography

[1] Andersen, L. V., Vahdatirad, M. J., Sichani, M. T., and Sørensen, J. D. (2012). Natural frequencies of wind turbines on monopile foundations in clayey soils—a probabilistic approach. *Computers and Geotechnics*, 43(0):1–11.

[2] Applied Physical Sciences Corp. (2010). Evaluate the effect of turbine period of vibration requirement on structural design parameters: Technical report of findings. Report, US Bureau of Ocean Energy Management, Regulation, and Enforcement (BOEMRE).

[3] Bush, E. and Manuel, L. (2009). *Foundation models for offshore wind turbines*. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2009-1037.

[4] Carswell, W., Johansson, J., Løvholt, F., Arwade, S. R., Madshus, C., DeGroot, D. J., and Myers, A. T. (2015). Foundation damping and the dynamics of offshore wind turbine monopiles. *Renewable Energy*, 80(0):724–736.

[5] Chopra, A. K. *Dynamics of Structures*. Prentice Hall, University of California, Berkely, 4th edition.

[6] Damgaard, M., Andersen, L. V., Ibsen, L. B., and Andersen, J. Time-varying dynamic properties of offshore wind turbines evaluated by modal testing.

[7] Det Norske Veritas (2010). DNV-RP-C205. Environmental conditions and environmental loads - Recommended Practice. Standard.

[8] Det Norske Veritas (2014). Offshore Standard DNV-OS-J101. Design of Offshore Wind Turbine Structures.

[9] Eurocode 8, Part 5 (2014). Eurocode 8, Part 5: Design of structures for earthquake resistance.

[10] Holmes, J. D. . (2001). *The atmospheric boundary layer and wind turbulence*. Spon Press. doi:10.4324/9780203301647.ch3.

[11] IEC 61400-3 (2009). Wind Turbines - Part 3: Design requirements for offshore wind turbines.

[12] Iwan, W. D. (1967). On a class of models for the yielding behavior of continuous and composite systems. *Journal of Applied Mechanics*, 34(3):612–617. 10.1115/1.3607751.

[13] Jonkman, B. (2012). Instructions for compiling FAST using IVF for Windows - Excerpt from a draft of the NWTC Programming Handbook. Report, National Renewable Energy Laboratory.

[14] Jonkman, B. and Jonkman, J. (2013). Addendum to the User's Guides for FAST, A2AD, and AeroDyn Released March 2010 - February 2013. Report, National Renewable Energy Laboratory.

[15] Jonkman, B. and Kilcher, L. (2012). TurbSim User's Guide: Version 1.06.00 DRAFT VERSION. Report, National Renewable Energy Laboratory.

[16] Jonkman, J. `http://wind.nrel.gov/public/jjonkman/BModes/` [Accessed: March 2015].

[17] Jonkman, J. (2007). Dynamics modeling and loads analysis of an offshore floating wind turbine. Report, National Renewable Energy Laboratory.

[18] Jonkman, J. (2015).

[19] Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-MW reference wind turbine for offshore system development. Report, National Renewable Energy Laboratory.

[20] Jonkman, J. M. and Buhl Jr, M. L. (2005). FAST User's Guide. Report, National Renewable Energy Laboratory.

[21] Jonkman, J.; Musial, W. (2010). Offshore Code Comparison Collaboration (OC3) for IEA Task 23 Offshore Wind Technology and Deployment. Report, National Renewable Energy Laboratory (NREL).

[22] Kramer, S. L. (1996). *Ground Response Analysis*, book section 7. International Series in Civil Engineering and Engineering Mechanics. Prentice-Hall.

[23] Leishman, J. G. and Beddoes, T. S. (1989). A semi-empirical model for dynamic stall. *Journal of the American Helicopter Society*, 34(3):3–17.

[24] Lombardi, D., Bhattacharya, S., and Muir Wood, D. (2013). Dynamic soil–structure interaction of monopile supported wind turbines in cohesive soil. *Soil Dynamics and Earthquake Engineering*, 49(0):165–180.

[25] Lynn, P. A. (2011). *Onshore and Offshore Wind Energy : An Introduction*. John Wiley and Sons.

[26] Madsen, P. H. Introduction to the IEC 61400-1 standard. `http://www.windpower.org/download/461/Introduction_to_the_IEC.pdf` [Accessed: May 2015].

[27] Meyer, V. Overview of offshore wind turbine foundation research at NGI. `http://www.ngi.no/upload/77442/03-OWT%20Foundation%20Research%20at%20NGI_vme.pdf` [Accessed: May 2015].

[28] NWTC. NWTC Forums - NWTC Information Portal. `https://wind.nrel.gov/forum/wind/` [Accessed: 2015].

[29] Orcina Ltd. Orcaflex manual version 9.8a. Report. `http://www.orcina.com/SoftwareProducts/OrcaFlex/Documentation/OrcaFlex.pdf` [Accessed: February 2015].

[30] Passon, P. (2006). Memorandum: Derivation and description of the soil-pile-interaction models. Report, University of Stuttgart.

[31] Passon, P. and Kühn, M. (2005). State-of-the-art and development needs of simulation codes for offshore wind turbines. http://www.ieawind.org/task_23/Subtask_2S_docs/Meeting%2004_Roskilde%20II/Risoe%202005-State_of_the_art.pdf [Accessed: February 2015].

[32] Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press.

[33] RenewableUK. Offshore wind. http://www.renewableuk.com/en/renewable-energy/wind-energy/offshore-wind/index.cfm [Accessed: May 2015].

[34] Schaffarczyk, A. (2014). *Understanding Wind Power Technology : Theory, Development and Optimisation*. Wiley, Somerset, NJ, USA.

[35] Schløer, S. (2013). *Fatigue and extreme wave loads on bottom fixed offshore wind turbines - Effects from fully nonlinear wave forcing on the structural dynamics*. Thesis.

[36] Stathopoulos, T. and Baniotopoulos, C. C. (2007). *Wind Effects on Buildings and Design of Wind-Sensitive Structures*. Springer Vienna, Vienna.

[37] Strømmen, E. N. (2014). *Structural Dynamics*. Springer Series in Solid and Structural Mechanics. Springer International Publishing.

[38] Sutherland, H. On the fatigue analysis of wind turbines. Report, Sandia National Laboratories.

[39] Tarp-Johansen, N. J., Andersen, L., Christensen, E., Mørch, C., and Frandsen, S. Comparing sources of damping of cross-wind motion.

[40] The European Wind Energy Association (2014). The European offshore wind industry - key trends and statistics 2014. Report, The European Wind Energy Association.

[41] The Scottish Government (2006). Planning for Micro Renewables Annex to PAN 45 Renewable Energy Technologies: Part 2. http://www.gov.scot/Publications/2006/10/03093936/2 [Accessed: April 2015].

[42] Wilkes, J. and Moccia, J. (2013). Wind in power: 2012 European statistics. Report. http://www.ewea.org/fileadmin/files/library/publications/statistics/Wind_in_power_annual_statistics_2012.pdf [Accessed: February 2015].

[43] Yauck, J. (2009). Turning great lakes wind into energy. *Bayview Compass*. http://bayviewcompass.com/category/h20/page/2/ [Accessed: June 2015].

[44] Zaaijer, M. B. (2006). Foundation modelling to assess dynamic behaviour of offshore wind turbines. *Applied Ocean Research*, 28(1):45–57.

Appendix A

Input Files for NREL 5MW Turbine

Main input file, platform file, tower file and AeroDyn input file is included, in that sequence.

```
---------------------------------------------------------------------------
------- FAST INPUT FILE -------------------------------------------------
NREL 5.0 MW Baseline Wind Turbine for Use in Offshore Analysis.
Properties from Dutch Offshore Wind Energy Converter (DOWEC) 6MW Pre-Design (10046_009.pdf) and REpower 5M 5MW (5m_uk.pdf); Compatible with FAST v7.02.
---------------------- SIMULATION CONTROL --------------------------------
False       Echo        - Echo input data to "echo.out" (flag)
   1        ADAMSPrep   - ADAMS preprocessor mode {1: Run FAST, 2: use FAST as a preprocessor to create an ADAMS model, 3: do both} (switch)
   1        AnalMode    - Analysis mode {1: Run a time-marching simulation, 2: create a periodic linearized model} (switch)
   3        NumBl       - Number of blades (-)
 630.0      TMax        - Total run time (s)
   0.012    DT          - Integration time step (s)
---------------------- TURBINE CONTROL -----------------------------------
   1        YCMode      - Yaw control mode {0: none, 1: user-defined from routine UserYawCont, 2: user-defined from Simulink/Labview} (switch)
   0.0      TYCOn       - Time to enable active yaw control (s) [unused when YCMode=0]
   1        PCMode      - Pitch control mode {0: none, 1: user-defined from routine PitchCntrl, 2: user-defined from Simulink/Labview} (switch)
   0.0      TPCOn       - Time to enable active pitch control (s) [unused when PCMode=0]
   2        VSContrl    - Variable-speed control mode {0: none, 1: simple VS, 2: user-defined from routine UserVSCont, 3: user-defined from Simulink/Labview} (switch)
9999.9      VS_RtGnSp   - Rated generator speed for simple variable-speed generator control (HSS side) (rpm) [used only when VSContrl=1]
9999.9      VS_RtTq     - Rated generator torque/constant generator torque in Region 3 for simple variable-speed generator control (HSS side) (N-m) [used only when VSContrl=1]
9999.9      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]
9999.9      VS_SlPc     - Rated generator slip percentage in Region 2 1/2 for simple variable-speed generator control (%) [used only when VSContrl=1]
   2        GenModel    - Generator model {1: simple, 2: Thevenin, 3: user-defined from routine UserGen} (switch) [used only when VSContrl=0]
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)
9999.9      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]
9999.9      TimGenOf    - Time to turn off the generator (s) [used only when GenTiStp=True]
   2        HSSBrMode   - HSS brake model {1: simple, 2: user-defined from routine UserHSSBr, 3: user-defined from Labview} (switch)
   0.0      THSSBrDp    - Time to initiate deployment of the HSS brake (s)
9999.9      TiDynBrk    - Time to initiate deployment of the dynamic generator brake [CURRENTLY IGNORED] (s)
9999.9      TTpBrDp(1)  - Time to initiate deployment of tip brake 1 (s)
9999.9      TTpBrDp(2)  - Time to initiate deployment of tip brake 2 (s)
9999.9      TTpBrDp(3)  - Time to initiate deployment of tip brake 3 (s) [unused for 2 blades]
9999.9      TBDepISp(1) - Deployment-initiation speed for the tip brake on blade 1 (rpm)
9999.9      TBDepISp(2) - Deployment-initiation speed for the tip brake on blade 2 (rpm)
9999.9      TBDepISp(3) - Deployment-initiation speed for the tip brake on blade 3 (rpm) [unused for 2 blades]
9999.9      TYawManS    - Time to start override yaw maneuver and end standard yaw control (s)
9999.9      TYawManE    - Time at which override yaw maneuver reaches final yaw angle (s)
   0.0      NacYawF     - Final yaw angle for override yaw maneuvers (degrees)
9999.9      TPitManS(1) - Time to start override pitch maneuver for blade 1 and end standard pitch control (s)
9999.9      TPitManS(2) - Time to start override pitch maneuver for blade 2 and end standard pitch control (s)
9999.9      TPitManS(3) - Time to start override pitch maneuver for blade 3 and end standard pitch control (s) [unused for 2 blades]
9999.9      TPitManE(1) - Time at which override pitch maneuver for blade 1 reaches final pitch (s)
9999.9      TPitManE(2) - Time at which override pitch maneuver for blade 2 reaches final pitch (s)
9999.9      TPitManE(3) - Time at which override pitch maneuver for blade 3 reaches final pitch (s) [unused for 2 blades]
   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      BlPitchF(1) - Blade 1 final pitch for pitch maneuvers (degrees)
   0.0      BlPitchF(2) - Blade 2 final pitch for pitch maneuvers (degrees)
   0.0      BlPitchF(3) - Blade 3 final pitch for pitch maneuvers (degrees) [unused for 2 blades]
---------------------- ENVIRONMENTAL CONDITIONS --------------------------
   9.80665  Gravity     - Gravitational acceleration (m/s^2)
---------------------- FEATURE FLAGS -------------------------------------
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        CompAero    - Compute aerodynamic forces (flag)
False       CompNoise   - Compute aerodynamic noise (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      TeetDefl    - Initial or fixed teeter angle (degrees) [unused for 3 blades]
   0.0      Azimuth     - Initial azimuth angle for blade 1 (degrees)
  12.1      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)
---------------------- TURBINE CONFIGURATION -------------------------------------
  63.0      TipRad      - The distance from the rotor apex to the blade tip (meters)
   1.5      HubRad      - The distance from the rotor apex to the blade root (meters)
   1        PSpnElN     - Number of the innermost blade element which is still part of the pitchable portion of the blade for partial-span pitch control [1 to BldNodes] [CURRENTLY IGNORED] (-)
   0.0      UndSling    - Undersling length [distance from teeter pin to the rotor apex] (meters) [unused for 3 blades]
   0.0      HubCM       - Distance from rotor apex to hub mass [positive downwind] (meters)
  -5.01910  OverHang    - Distance from yaw axis to rotor apex [3 blades] or teeter pin [2 blades] (meters)
   1.9      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)
   1.75     NacCMzn     - Vertical distance from the tower-top to the nacelle CM (meters)
  87.6      TowerHt     - Height of tower above ground level [onshore] or MSL [offshore] (meters)
   1.96256  Twr2Shft    - Vertical distance from the tower-top to the rotor shaft (meters)
   0.0      TwrRBHt     - Tower rigid base height (meters)
  -5.0      ShftTilt    - Rotor shaft tilt angle (degrees)
   0.0      Delta3      - Delta-3 angle for teetering rotors (degrees) [unused for 3 blades]
  -2.5      PreCone(1)  - Blade 1 cone angle (degrees)
  -2.5      PreCone(2)  - Blade 2 cone angle (degrees)
  -2.5      PreCone(3)  - Blade 3 cone angle (degrees) [unused for 2 blades]
   0.0      AzimB1Up    - Azimuth value to use for I/O when blade 1 points up (degrees)
---------------------- MASS AND INERTIA ------------------------------------------
   0.0      YawBrMass   - Yaw bearing mass (kg)
 240.00E3   NacMass     - Nacelle mass (kg)
  56.78E3   HubMass     - Hub mass (kg)
   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]
2607.89E3   NacYIner    - Nacelle inertia about yaw axis (kg m^2)
 534.116    GenIner     - Generator inertia about HSS (kg m^2)
 115.926E3  HubIner     - Hub inertia about rotor axis [3 blades] or teeter axis [2 blades] (kg m^2)
---------------------- DRIVETRAIN ------------------------------------------------
 100.0      GBoxEff     - Gearbox efficiency (%)
  94.4      GenEff      - Generator efficiency [ignored by the Thevenin and user-defined generator models] (%)
  97.0      GBRatio     - Gearbox ratio (-)
False       GBRevers    - Gearbox reversal {T: if rotor and generator rotate in opposite directions} (flag)
  28.1162E3 HSSBrTqF    - Fully deployed HSS-brake torque (N-m)
   0.6      HSSBrDT     - Time for HSS-brake to reach full deployment once initiated (sec) [used only when HSSBrMode=1]
"Dummy"     DynBrkFi    - File containing a mech-gen-torque vs HSS-speed curve for a dynamic brake [CURRENTLY IGNORED] (quoted string)
 867.637E6  DTTorSpr    - Drivetrain torsional spring (N-m/rad)
   6.215E6  DTTorDmp    - Drivetrain torsional damper (N-m/(rad/s))
---------------------- SIMPLE INDUCTION GENERATOR --------------------------------
9999.9      SIG_SlPc    - Rated generator slip percentage (%) [used only when VSContrl=0 and GenModel=1]
9999.9      SIG_SySp    - Synchronous (zero-torque) generator speed (rpm) [used only when VSContrl=0 and GenModel=1]
9999.9      SIG_RtTq    - Rated torque (N-m) [used only when VSContrl=0 and GenModel=1]
9999.9      SIG_PORt    - Pull-out ratio (Tpullout/Trated) (-) [used only when VSContrl=0 and GenModel=1]
---------------------- THEVENIN-EQUIVALENT INDUCTION GENERATOR -------------------
9999.9      TEC_Freq    - Line frequency [50 or 60] (Hz) [used only when VSContrl=0 and GenModel=2]
9998        TEC_NPol    - Number of poles [even integer > 0] (-) [used only when VSContrl=0 and GenModel=2]
9999.9      TEC_SRes    - Stator resistance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9      TEC_RRes    - Rotor resistance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9      TEC_VLL     - Line-to-line RMS voltage (volts) [used only when VSContrl=0 and GenModel=2]
9999.9      TEC_SLR     - Stator leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9      TEC_RLR     - Rotor leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9      TEC_MR      - Magnetizing reactance (ohms) [used only when VSContrl=0 and GenModel=2]
---------------------- PLATFORM --------------------------------------------------
   2        PtfmModel   - Platform model {0: none, 1: onshore, 2: fixed bottom offshore, 3: floating offshore} (switch)
"NRELOffshrBsline5MW_Platform_Monopile_RF.dat"   PtfmFile    - Name of file containing platform properties (quoted string) [unused when PtfmModel=0]
---------------------- TOWER -----------------------------------------------------
  99        TwrNodes    - Number of tower nodes used for analysis (-)
"NRELOffshrBsline5MW_Tower_Monopile_RF.dat"      TwrFile     - Name of file containing tower properties (quoted string)
---------------------- NACELLE-YAW -----------------------------------------------
9028.32E6   YawSpr      - Nacelle-yaw spring constant (N-m/rad)
  19.16E6   YawDamp     - Nacelle-yaw damping constant (N-m/(rad/s))
   0.0      YawNeut     - Neutral yaw position--yaw spring force is zero at this yaw (degrees)
---------------------- FURLING ---------------------------------------------------
False       Furling     - Read in additional model properties for furling turbine (flag)
"Dummy"     FurlFile    - Name of file containing furling properties (quoted string) [unused when Furling=False]
---------------------- 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      TeetSStP    - Rotor-teeter soft-stop position (degrees) [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      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]
---------------------- TIP-BRAKE ---------------------------------------------
   0.0      TBDrConN    - Tip-brake drag constant during normal operation, Cd*Area (m^2)
   0.0      TBDrConD    - Tip-brake drag constant during fully-deployed operation, Cd*Area (m^2)
   0.0      TpBrDT      - Time for tip-brake to reach full deployment once released (sec)
---------------------- BLADE -------------------------------------------------
"NRELOffshrBsline5MW_Blade.dat"                 BldFile(1)  - Name of file containing properties for blade 1 (quoted string)
"NRELOffshrBsline5MW_Blade.dat"                 BldFile(2)  - Name of file containing properties for blade 2 (quoted string)
"NRELOffshrBsline5MW_Blade.dat"                 BldFile(3)  - Name of file containing properties for blade 3 (quoted string) [unused for 2 blades]
---------------------- AERODYN -----------------------------------------------
"NRELOffshrBsline5MW_AeroDyn.ipt"               ADFile      - Name of file containing AeroDyn input parameters (quoted string)
---------------------- NOISE -------------------------------------------------
"Dummy"     NoiseFile   - Name of file containing aerodynamic noise input parameters (quoted string) [used only when CompNoise=True]
---------------------- ADAMS -------------------------------------------------
"NRELOffshrBsline5MW_ADAMSSpecific.dat"         ADAMSFile   - Name of file containing ADAMS-specific input parameters (quoted string) [unused when ADAMSPrep=1]
---------------------- LINEARIZATION CONTROL ---------------------------------
"NRELOffshrBsline5MW_Linear.dat"                LinFile     - Name of file containing FAST linearization parameters (quoted string) [unused when AnalMode=1]
---------------------- OUTPUT ------------------------------------------------
True        SumPrint    - Print summary data to "<RootName>.fsm" (flag)
1           OutFileFmt  - Format for tabular (time-marching) output file(s) (1: text file [<RootName>.out], 2: binary file [<RootName>.outb], 3: both) (switch)
True        TabDelim    - Use tab delimiters in text tabular output file? (flag)
"ES10.3E2"  OutFmt      - Format used for text tabular output (except time).  Resulting field should be 10 characters. (quoted string)  [not checked for validity!]
  30.0      TStart      - Time to begin tabular output (s)
   1        DecFact     - Decimation factor for tabular output {1: output every time step} (-)
   1.0      SttsTime    - Amount of time between screen status messages (sec)
  -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)
   1.912    ShftGagL    - Distance from rotor apex [3 blades] or teeter pin [2 blades] to shaft strain gages [positive for upwind rotors] (meters)
   3        NTwGages    - Number of tower nodes that have strain gages for output [0 to 9] (-)
10,19,28    TwrGagNd    - List of tower nodes that have strain gages [1 to TwrNodes] (-) [unused if NTwGages=0]
   3        NBlGages    - Number of blade nodes that have strain gages for output [0 to 9] (-)
 5,9,13     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 OutList.txt for a listing of available output channels, (-)
"WindVxi  , WindVyi  , WindVzi"                                 - Longitudinal, lateral, and vertical wind speeds
"WaveElev"                                                      - Wave elevation at the platform reference point
"GenPwr   , GenTq"                                              - Electrical generator power and torque
"OoPDefl1 , IPDefl1  , TwstDefl1"                               - Blade 1 out-of-plane and in-plane deflections and tip twist
"BldPitch1"                                                     - Blade 1 pitch angle
"Azimuth"                                                       - Blade 1 azimuth angle
"RotSpeed , GenSpeed"                                           - Low-speed shaft and high-speed shaft speeds
"TTDspFA  , TTDspSS  , TTDspTwst"                               - Tower fore-aft and side-to-side displacements and top twist
"TeetDefl , TeetDefl"                                           - Dummy placeholders for the unavailable fore-aft and side-to-side displacements of the tower base at the monopile attachement location
"PtfmTDxi , PtfmTDyi , PtfmRDxi , PtfmRDyi"                     - Fore-aft, side-to-side, roll, and pitch displacements of the monopile at the mudline
"TeetDefl , TeetDefl"                                           - Dummy placeholders for the unavailable fore-aft and side-to-side displacements of the monopile at 7m below the mudline
"Spn2MLxb1, Spn2MLyb1"                                          - Blade 1 local edgewise and flapwise bending moments at span station 2 (approx. 50% span)
"RootFxc1 , RootFyc1 , RootFzc1"                                - Out-of-plane shear, in-plane shear, and axial forces at the root of blade 1
"RootMxc1 , RootMyc1 , RootMzc1"                                - In-plane bending, out-of-plane bending, and pitching moments at the root of blade 1
"RotTorq  , LSSGagMya, LSSGagMza"                               - Rotor torque and low-speed shaft 0- and 90-bending moments at the main bearing
"YawBrFxp , YawBrFyp , YawBrFzp"                                - Fore-aft shear, side-to-side shear, and vertical forces at the top of the tower (not rotating with nacelle yaw)
"YawBrMxp , YawBrMyp , YawBrMzp"                                - Side-to-side bending, fore-aft bending, and yaw moments at the top of the tower (not rotating with nacelle yaw)
"TwHt3MLxt, TwHt3MLyt"                                          - Local side-to-side and fore-aft bending moments at tower gage 3 (approx. tower base / monopile attachement location)
"TwHt2MLxt, TwHt2MLyt"                                          - Local side-to-side and fore-aft bending moments at tower gage 2 (approx. MSL)
"TwHt1MLxt, TwHt1MLyt"                                          - Local side-to-side and fore-aft bending moments at tower gage 1 (approx. halfway between MSL and mudline)
"TwrBsFxt , TwrBsFyt , TwrBsFzt"                                - Fore-aft shear, side-to-side shear, and vertical forces at the mudline
"TwrBsMxt , TwrBsMyt , TwrBsMzt"                                - Side-to-side bending, fore-aft bending, and yaw moments at the mudline
"TeetDefl , TeetDefl , TeetDefl , TeetDefl"                     - Dummy placeholders for the unavailable local fore-aft shear, side-to-side shear, side-to-side bending moment, and fore-aft bending moment of the
monopile at 7m below the mudline
"PtfmSurge, PtfmSway , PtfmRoll , PtfmPitch"                    - Platform surge, sway, roll, and pitch displacements (for use in calculating in Crunch the tower-top displacements in the inertia frame)
END of FAST input file (the word "END" must appear in the first 3 columns of this last line).
---------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------
---------------------- FAST PLATFORM FILE ------------------------------------
NREL 5.0 MW offshore baseline monopile platform with rigid foundation input properties.
---------------------- FEATURE FLAGS (CONT) ----------------------------------
False       PtfmSgDOF   - Platform horizontal surge translation DOF (flag)
False       PtfmSwDOF   - Platform horizontal sway translation DOF (flag)
False       PtfmHvDOF   - Platform vertical heave translation DOF (flag)
False       PtfmRDOF    - Platform roll tilt rotation DOF (flag)
False       PtfmPDOF    - Platform pitch tilt rotation DOF (flag)
False       PtfmYDOF    - Platform yaw rotation DOF (flag)
---------------------- INITIAL CONDITIONS (CONT) -----------------------------
   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)
   0.0      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 (CONT) --------------------------
  20.0      TwrDraft    - Downward distance from the ground level [onshore] or MSL [offshore] to the tower base platform connection (meters)
  20.0      PtfmCM      - Downward distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
  20.0      PtfmRef     - Downward distance from the ground level [onshore] or MSL [offshore] to the platform reference point (meters)
---------------------- MASS AND INERTIA (CONT) -------------------------------
   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)
   0.0      PtfmYIner   - Platform inertia for yaw rotation about the platform CM (kg m^2)
---------------------- PLATFORM (CONT) ---------------------------------------
   0        PtfmLdMod   - Platform loading model {0: none, 1: user-defined from routine UserPtfmLd} (switch)
---------------------- TOWER (CONT) ------------------------------------------
   1        TwrLdMod    - Tower loading model {0: none, 1: Morison's equation, 2: user-defined from routine UserTwrLd} (switch)
   6.0      TwrDiam     - Tower diameter in Morison's equation (meters) [used only when TwrLdMod=1]
   1.0      TwrCA       - Normalized hydrodynamic added mass   coefficient in Morison's equation (-) [used only when TwrLdMod=1] [determines TwrCM=1+TwrCA]
   1.0      TwrCD       - Normalized hydrodynamic viscous drag coefficient in Morison's equation (-) [used only when TwrLdMod=1]
---------------------- WAVES -------------------------------------------------
1027.0      WtrDens     - Water density (kg/m^3)
  20.0      WtrDpth     - Water depth (meters)
   2        WaveMod     - Incident wave kinematics model {0: none=still water, 1: plane progressive (regular), 2: JONSWAP/Pierson-Moskowitz spectrum (irregular), 3: user-defind spectrum from routine
UserWaveSpctrm (irregular), 4: GH Bladed wave data} (switch)
   3        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]
3630.0      WaveTMax    - Analysis time for incident wave calculations (sec) [unused when WaveMod=0] [determines WaveDOmega=2Pi/WaveTMax in the IFFT]
   0.25     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]
   5.0      WaveHs      - Significant wave height of incident waves (meters) [used only when WaveMod=1 or 2]
  12.0      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 (unquoted string) [used only when WaveMod=2] [use 1.0 for Pierson-Moskowitz]
   180.0      WaveDir     - Incident wave propagation heading direction (degrees) [unused when WaveMod=0 or 4]
123456789   WaveSeed(1) - First  random seed of incident waves [-2147483648 to 2147483647] (-) [unused when WaveMod=0 or 4]
1011121314  WaveSeed(2) - Second random seed of incident waves [-2147483648 to 2147483647] (-) [unused when WaveMod=0 or 4]
"Dummy"     GHWvFile    - Root name of GH Bladed files containing wave data (quoted string) [used only when WaveMod=4]
---------------------- CURRENT -----------------------------------------------
   0        CurrMod     - Current profile model {0: none=no current, 1: standard, 2: user-defined from routine UserCurrent} (switch)
   0.0      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 (unquoted string) [used only when CurrMod=1]
  20.0      CurrNSRef   - Near-surface current reference depth (meters) [used only when CurrMod=1]
   0.0      CurrNSV0    - Near-surface current velocity at still water level (m/s) [used only when CurrMod=1]
   0.0      CurrNSDir   - Near-surface current heading direction (degrees) [used only when CurrMod=1]
   0.0      CurrDIV     - Depth-independent current velocity (m/s) [used only when CurrMod=1]
   0.0      CurrDIDir   - Depth-independent current heading direction (degrees) [used only when CurrMod=1]
---------------------- OUTPUT (CONT) -----------------------------------------
   0        NWaveKin    - Number of points where the wave kinematics can be output [0 to 9] (-)
   0        WaveKinNd   - List of tower nodes that have wave kinematics sensors [1 to TwrNodes] (-) [unused if NWaveKin=0]
```

```
------------------------------------------------------------------------------
---------------------- FAST PLATFORM FILE ------------------------------------
NREL 5.0 MW offshore baseline monopile platform with rigid foundation input properties.
---------------------- FEATURE FLAGS (CONT) ----------------------------------
False       PtfmSgDOF   - Platform horizontal surge translation DOF (flag)
False       PtfmSwDOF   - Platform horizontal sway translation DOF (flag)
False       PtfmHvDOF   - Platform vertical heave translation DOF (flag)
False       PtfmRDOF    - Platform roll tilt rotation DOF (flag)
False       PtfmPDOF    - Platform pitch tilt rotation DOF (flag)
False       PtfmYDOF    - Platform yaw rotation DOF (flag)
---------------------- INITIAL CONDITIONS (CONT) -----------------------------
   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)
   0.0      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 (CONT) --------------------------
  20.0      TwrDraft    - Downward distance from the ground level [onshore] or MSL [offshore] to the tower base platform connection (meters)
  20.0      PtfmCM      - Downward distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
  20.0      PtfmRef     - Downward distance from the ground level [onshore] or MSL [offshore] to the platform reference point (meters)
---------------------- MASS AND INERTIA (CONT) -------------------------------
   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)
   0.0      PtfmYIner   - Platform inertia for yaw rotation about the platform CM (kg m^2)
---------------------- PLATFORM (CONT) ---------------------------------------
   0        PtfmLdMod   - Platform loading model {0: none, 1: user-defined from routine UserPtfmLd} (switch)
---------------------- TOWER (CONT) ------------------------------------------
   1        TwrLdMod    - Tower loading model {0: none, 1: Morison's equation, 2: user-defined from routine UserTwrLd} (switch)
   6.0      TwrDiam     - Tower diameter in Morison's equation (meters) [used only when TwrLdMod=1]
   1.0      TwrCA       - Normalized hydrodynamic added mass   coefficient in Morison's equation (-) [used only when TwrLdMod=1] [determines TwrCM=1+TwrCA]
   1.0      TwrCD       - Normalized hydrodynamic viscous drag coefficient in Morison's equation (-) [used only when TwrLdMod=1]
---------------------- WAVES -------------------------------------------------
1027.0      WtrDens     - Water density (kg/m^3)
  20.0      WtrDpth     - Water depth (meters)
   2        WaveMod     - Incident wave kinematics model {0: none=still water, 1: plane progressive (regular), 2: JONSWAP/Pierson-Moskowitz spectrum (irregular), 3: user-defind spectrum from routine
UserWaveSpctrm (irregular), 4: GH Bladed wave data} (switch)
   3        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]
3630.0      WaveTMax    - Analysis time for incident wave calculations (sec) [unused when WaveMod=0] [determines WaveDOmega=2Pi/WaveTMax in the IFFT]
   0.25     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]
   5.0      WaveHs      - Significant wave height of incident waves (meters) [used only when WaveMod=1 or 2]
  12.0      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 (unquoted string) [used only when WaveMod=2] [use 1.0 for Pierson-Moskowitz]
   180.0      WaveDir     - Incident wave propagation heading direction (degrees) [unused when WaveMod=0 or 4]
123456789   WaveSeed(1) - First  random seed of incident waves [-2147483648 to 2147483647] (-) [unused when WaveMod=0 or 4]
1011121314  WaveSeed(2) - Second random seed of incident waves [-2147483648 to 2147483647] (-) [unused when WaveMod=0 or 4]
"Dummy"     GHWvFile    - Root name of GH Bladed files containing wave data (quoted string) [used only when WaveMod=4]
---------------------- CURRENT -----------------------------------------------
   0        CurrMod     - Current profile model {0: none=no current, 1: standard, 2: user-defined from routine UserCurrent} (switch)
   0.0      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 (unquoted string) [used only when CurrMod=1]
  20.0      CurrNSRef   - Near-surface current reference depth (meters) [used only when CurrMod=1]
   0.0      CurrNSV0    - Near-surface current velocity at still water level (m/s) [used only when CurrMod=1]
   0.0      CurrNSDir   - Near-surface current heading direction (degrees) [used only when CurrMod=1]
   0.0      CurrDIV     - Depth-independent current velocity (m/s) [used only when CurrMod=1]
   0.0      CurrDIDir   - Depth-independent current heading direction (degrees) [used only when CurrMod=1]
---------------------- OUTPUT (CONT) -----------------------------------------
   0        NWaveKin    - Number of points where the wave kinematics can be output [0 to 9] (-)
   0        WaveKinNd   - List of tower nodes that have wave kinematics sensors [1 to TwrNodes] (-) [unused if NWaveKin=0]
```

```
NREL 5.0 MW offshore baseline aerodynamic input properties; Compatible with AeroDyn v12.58.
SI          SysUnits    - System of units used for input and output [must be SI for FAST] (unquoted string)
BEDDOES     StallMod    - Dynamic stall included [BEDDOES or STEADY] (unquoted string)
USE_CM      UseCm       - Use aerodynamic pitching moment model? [USE_CM or NO_CM] (unquoted string)
EQUIL       InfModel    - Inflow model [DYNIN or EQUIL] (unquoted string)
SWIRL       IndModel    - Induction-factor model [NONE or WAKE or SWIRL] (unquoted string)
   0.005    AToler      - Induction-factor tolerance (convergence criteria) (-)
PRANDtl     TLModel     - Tip-loss model (EQUIL only) [PRANDtl, GTECH, or NONE] (unquoted string)
PRANDtl     HLModel     - Hub-loss model (EQUIL only) [PRANdtl or NONE] (unquoted string)
"WindData\90m_12mps.wnd"                                WindFile    - Name of file containing wind data (quoted string)
  90.0      HH          - Wind reference (hub) height [TowerHt+Twr2Shft+OverHang*SIN(ShftTilt)] (m)
   0.0      TwrShad     - Tower-shadow velocity deficit (-)
9999.9      ShadHWid    - Tower-shadow half width (m)
9999.9      T_Shad_Refpt - Tower-shadow reference point (m)
   1.225    AirDens     - Air density (kg/m^3)
   1.464E-5 KinVisc     - Kinematic air viscosity [CURRENTLY IGNORED] (m^2/sec)
   0.02479  DTAero      - Time interval for aerodynamic calculations (sec)
   8        NumFoil     - Number of airfoil files (-)
"AeroData\Cylinder1.dat"                        FoilNm      - Names of the airfoil files [NumFoil lines] (quoted strings)
"AeroData\Cylinder2.dat"
"AeroData\DU40_A17.dat"
"AeroData\DU35_A17.dat"
"AeroData\DU30_A17.dat"
"AeroData\DU25_A17.dat"
"AeroData\DU21_A17.dat"
"AeroData\NACA64_A17.dat"
  17        BldNodes    - Number of blade nodes used for analysis (-)
RNodes   AeroTwst  DRNodes  Chord  NFoil  PrnElm
 2.8667  13.308    2.7333   3.542  1      NOPRINT
 5.6000  13.308    2.7333   3.854  1      NOPRINT
 8.3333  13.308    2.7333   4.167  2      NOPRINT
11.7500  13.308    4.1000   4.557  3      NOPRINT
15.8500  11.480    4.1000   4.652  4      NOPRINT
19.9500  10.162    4.1000   4.458  4      NOPRINT
24.0500   9.011    4.1000   4.249  5      NOPRINT
28.1500   7.795    4.1000   4.007  6      NOPRINT
32.2500   6.544    4.1000   3.748  6      NOPRINT
36.3500   5.361    4.1000   3.502  7      NOPRINT
40.4500   4.188    4.1000   3.256  7      NOPRINT
44.5500   3.125    4.1000   3.010  8      NOPRINT
48.6500   2.319    4.1000   2.764  8      NOPRINT
52.7500   1.526    4.1000   2.518  8      NOPRINT
56.1667   0.863    2.7333   2.313  8      NOPRINT
58.9000   0.370    2.7333   2.086  8      NOPRINT
61.6333   0.106    2.7333   1.419  8      NOPRINT
```

Appendix B

TurbSim

The input file used in TurbSim to generate full-field wind is presented here.

```
TurbSim Input File. Valid for TurbSim v1.06.00, 21-Sep-2012

---------Runtime Options-----------------------------------
511347            RandSeed1       - First random seed  (-2147483648 to 2147483647)
RANLUX      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)
False       WrADFF          - Output full-field time-series data in TurbSim/AeroDyn form? (Generates Rootname.bts)
True        WrBLFF          - Output full-field time-series data in BLADED/AeroDyn form?  (Generates RootName.wnd)
False       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-----------------------
31          NumGrid_Z       - Vertical grid-point matrix dimension
31          NumGrid_Y       - Horizontal grid-point matrix dimension
0.05         TimeStep        - Time step [seconds]
630.0       AnalysisTime    - Length of analysis time series [seconds] (program will add time if necessary: AnalysisTime = MAX(AnalysisTime, UsableTime+GridWidth/MeanHHWS) )
630.0       UsableTime      - Usable length of output time series [seconds] (program will add GridWidth/MeanHHWS seconds)
90.0        HubHt           - Hub height [m] (should be > 0.5*GridHeight)
145.00      GridHeight      - Grid height [m]
145.00      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"=Kaimal, "IECVKM"=von Karman, "GP_LLJ", "NWTCUP", "SMOOTH", "WF_UPW", "WF_07D", "WF_14D", "TIDAL", or "NONE")
"1-ED3"     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]
PL          WindProfileType - Wind profile type ("JET";"LOG"=logarithmic;"PL"=power law;"H2L"=Log law for TIDAL spectral model;"IEC"=PL on rotor disk, LOG elsewhere; or "default")
90          RefHt           - Height of the reference wind speed [m]
12          URef            - Mean (total) wind speed at the reference height [m/s] (or "default" for JET wind profile)
default     ZJetMax         - Jet height [m] (used only for JET wind profile, valid 70-490 m)
default     PLExp           - Power law exponent [-] (or "default")
default     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 (or "default")
default     PC_UV           - Hub mean u'v' Reynolds stress (or "default")
default     PC_VW           - Hub mean v'w' Reynolds stress (or "default")
default     IncDec1         - u-component coherence parameters (e.g. "10.0  0.3e-3" in quotes) (or "default")
default     IncDec2         - v-component coherence parameters (e.g. "10.0  0.3e-3" in quotes) (or "default")
default     IncDec3         - w-component coherence parameters (e.g. "10.0  0.3e-3" in quotes) (or "default")
default     CohExp          - Coherence exponent (or "default")

--------Coherent Turbulence Scaling Parameters-------------------
"C:\DesignCodes\TurbSim\Test\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 wave 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.)
10.0        CTStartTime     - Minimum start time for coherent structures in RootName.cts [seconds]

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

Appendix C

BModes

Input files to BModes are included in this chapter, namely *CS_Monopile.bmi*, two first pages, and *CS_Monopile_tower_sec.dat*, last page.

```
======================   BModes v3.00 Main Input File  ==================
NREL 5MW Tower

--------- General parameters -----------------------------------------------------------
true      Echo        Echo input file contents to *.echo file if true.
2         beam_type   1: blade, 2: tower (-)
0.        romg:       rotor speed, automatically set to zero for tower modal analysis (rpm)
1.        romg_mult:  rotor speed muliplicative factor (-)
87.6      radius:     rotor tip radius measured along coned blade axis, OR tower height above ground level [onshore] or MSL [offshore](m)
0.        hub_rad:    hub radius measured along coned blade axis OR tower rigid-base height (m)
0.        precone:    built-in precone angle, automatically set to zero for a tower (deg)
0.        bl_thp:     blade pitch setting, automatically set to zero for a tower (deg)
3         hub_conn:   hub-to-blade or tower-base boundary condition [1: cantilevered; 2: free-free; 3: only axial and torsion constraints] (-)
20        modepr:     number of modes to be printed (-)
t         TabDelim    (true: tab-delimited output tables; false: space-delimited tables)
f         mid_node_tw  (true: output twist at mid-node of elements; false: no mid-node outputs)

---------- Blade-tip or tower-top mass properties ------------------------------------------
3.500003109E+005   tip_mass    blade-tip or tower-top mass (kg)
-0.4137754432      cm_loc      tip-mass c.m. offset from the tower axis measured along x-tower axis (m)
1.9669893542       cm_axial    tip-mass c.m. offset tower tip measures axially along the z axis (m)
4.370E7            ixx_tip     blade lag mass moment of inertia about the tip-section x reference axis (kg-m^2)
2.353E7            iyy_tip     blade flap mass moment of inertia about the tip-section y reference axis (kg-m^2)
2.542E7            izz_tip     torsion mass moment of inertia about the tip-section z reference axis (kg-m^2)
0.                 ixy_tip     cross product of inertia about x and y reference axes(kg-m^2)
1.169E6            izx_tip     cross product of inertia about z and x reference axes(kg-m^2)
0.                 iyz_tip     cross product of inertia about y and z reference axes(kg-m^2)

--------- Distributed-property identifiers ------------------------------------------------
1         id_mat:     material_type [1: isotropic; non-isotropic composites option not yet available]
'CS_monopile_tower_secs.dat'   : sec_props_file   name of beam section properties file (-)

Property scaling factors.............................
1.0       sec_mass_mult:   mass density multiplier (-)
1.0       flp_iner_mult:   blade flap or tower f-a inertia multiplier (-)
1.0       lag_iner_mult:   blade lag or tower s-s inertia multiplier (-)
1.0       flp_stff_mult:   blade flap or tower f-a bending stiffness multiplier (-)
1.0       edge_stff_mult:  blade lag or tower s-s bending stiffness multiplier (-)
1.0       tor_stff_mult:   torsion stiffness multiplier (-)
1.0       axial_stff_mult: axial stiffness multiplier (-)
1.0       cg_offst_mult:   cg offset multiplier (-)
1.0       sc_offst_mult:   shear center multiplier (-)
1.0       tc_offst_mult:   tension center multiplier (-)

--------- Finite element discretization --------------------------------------------------
61        nselt:     no of blade or tower elements (-)
Distance of element boundary nodes from blade or flexible-tower root (normalized wrt blade or tower length), el_loc()
0         0.003481894        0.010445682        0.017409471        0.024373259        0.031337047        0.038300836        0.045264624        0.052228412        0.059192201        0.066155989        0.073119777
          0.080083565        0.087047354        0.094011142        0.10097493         0.107938719        0.114902507        0.121866295        0.128830084        0.135793872         0.13990 0.149721448
0.156685237        0.163649025        0.170612813        0.177576602        0.18454039         0.191504178        0.198467967        0.205431755        0.212395543        0.219359331        0.22632312
0.233286908        0.240250696        0.247214485        0.250696379        0.320334262         0.37971  0.424791072       0.45961  0.486635 0.51366  0.54068  0.5677    0.594715 0.62173  0.64875  0.67577  0.70279
          0.72981  0.75683  0.78385  0.81087  0.83789  0.864905 0.89192  0.91894  0.94596  0.97298  1.0

--------- Properties of tower support subsystem (read only if beam_type is 2) ------------
1          tow_support: : aditional tower support [0: no additional support; 1: floating-platform or monopile with or without tension wires] (-)
20.0       draft        : depth of tower base from the ground or the MSL (mean sea level) (m)
0.0        cm_pform     : distance of platform c.m. below the MSL (m)
0.0        mass_pform   : platform mass (kg)
Platform mass inertia 3X3 matrix (i_matrix_pform):
0.   0.   0.
0.   0.   0.
0.   0.   0.
20.0       ref_msl    : distance of platform reference point below the MSL (m)
Platform-reference-point-referred hydrodynamic 6X6 matrix (hydro_M):
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.
Platform-reference-point-referred hydrodynamic 6X6 stiffness matrix (hydro_K):
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.
Mooring-system 6X6 stiffness matrix (mooring_K):
  2574800000.0             0.0     0.0              0.0    -22532500000.0      0.0
           0.0     2574800000.0     0.0      22532500000.0              0.0      0.0
           0.0             0.0     0.0              0.0              0.0      0.0
           0.0    22532500000.0     0.0    262912300000.0              0.0      0.0
-22532500000.0             0.0     0.0              0.0    262912300000.0      0.0
           0.0             0.0     0.0              0.0              0.0      0.0

Distributed (hydrodynamic) added-mass per unit length along a flexible portion of the tower length:
0          n_secs_m_distr: number of sections at which added mass per unit length is specified (-)
0.  0.    : z_distr_m [row array of size n_added_m_pts; section locations wrt the flexible tower base over which distributed mass is specified] (m)
0.  0.    : distr_m [row array of size n_added_m_pts; added distributed masses per unit length] (kg/m)

Distributed elastic stiffness per unit length along a flexible portion of the tower length:
0          n_secs_k_distr: number of points at which distributed stiffness per unit length is specified (-)
0        1       2       3       4       5       6       7       8       9       10      11      12      13      14      15      16      17      18      19      20      21      22
         23      24      25      26      27      28      29      30      31      32      33      34      35      36  : z_distr_k [row array of size n_added_m_pts; section locations wrt the
flexible tower base over which distributed stiffness is specified] (m)
595318000.0      1165208000       1129400000       1095553000       1059931000       1024493000       989209000 953643000 918718000 883287000 847803000 812541000 777187000 741870000 706616000 671440000
636229000 600957000 565919000 530470000 495081000 459574000 385327000 305479000 280059000 254125000 227500000 200112000 171927000 143115000 114173000 80184000  52237000  35561000  20912000  9000000   1156000   : distr_k [row
array of size n_added_m_pts; distributed stiffness per unit length] (N/m^2)

Tension wires data
0          n_attachments: no of wire-attachment locations on tower [0: no tension wires] (-)
3 3        n_wires:       no of wires attached at each location (must be 3 or higher) (-)
6 9        node_attach:   node numbers of attacments location (node number must be more than 1 and less than nselt+2) (-)
0.e0 0.e0 wire_stfness:  wire spring constant in each set (see users' manual) (N/m)
0. 0.      th_wire:       angle of tension wires (wrt the horizontal ground plane) at each attachment point (deg)

END of Main Input File Data ********************************************************************
***************************************************************************************************
```

Tower section properties
13 n_secs: number of tower sections at which properties are specified (-)

sec_loc	str_tw	tw_iner	mass_den	flp_iner	edge_iner	flp_stff	edge_stff	tor_stff	axial_stff	cg_offst	sc_offst	tc_offst
(-)	(deg)	(deg)	(kg/m)	(kg-m)	(kg-m)	(Nm^2)	(Nm^2)	(Nm^2)	(N)	(m)	(m)	(m)
0.00000	0.0	0.0	9517.14	4.2E-3	4.2E-3	1037.13E9	1037.13E9	798.09E15	235.12E15	0.0	0.0	0.0
0.27881	0.0	0.0	9517.14	4.2E-3	4.2E-3	1037.13E9	1037.13E9	798.09E15	235.12E15	0.0	0.0	0.0
0.27882	0.0	0.0	4306.51	1.9E-3	1.9E-3	474.49E9	474.49E9	365.13E15	106.39E15	0.0	0.0	0.0
0.35094	0.0	0.0	4030.44	1.7E-3	1.7E-3	413.08E9	413.08E9	317.87E15	99.57E15	0.0	0.0	0.0
0.42306	0.0	0.0	3763.45	1.4E-3	1.4E-3	357.83E9	357.83E9	275.35E15	92.97E15	0.0	0.0	0.0
0.49517	0.0	0.0	3505.52	1.2E-3	1.2E-3	308.30E9	308.30E9	237.24E15	86.60E15	0.0	0.0	0.0
0.56729	0.0	0.0	3256.66	1.1E-3	1.1E-3	264.08E9	264.08E9	203.22E15	80.45E15	0.0	0.0	0.0
0.63941	0.0	0.0	3016.86	9.1E-4	9.1E-4	224.80E9	224.80E9	172.98E15	74.53E15	0.0	0.0	0.0
0.71153	0.0	0.0	2786.13	7.7E-4	7.7E-4	190.06E9	190.06E9	146.25E15	68.83E15	0.0	0.0	0.0
0.78365	0.0	0.0	2564.46	6.5E-4	6.5E-4	159.49E9	159.49E9	122.73E15	63.35E15	0.0	0.0	0.0
0.85576	0.0	0.0	2351.87	5.4E-4	5.4E-4	132.77E9	132.77E9	102.16E15	58.10E15	0.0	0.0	0.0
0.92788	0.0	0.0	2148.34	4.4E-4	4.4E-4	109.54E9	109.54E9	84.29E15	53.07E15	0.0	0.0	0.0
1.00000	0.0	0.0	1953.87	3.6E-4	3.6E-4	89.49E9	89.49E9	68.86E15	48.27E15	0.0	0.0	0.0

**Note: If the above data represents TOWER properties, the following are overwritten:
  str_tw is set to zero
  tw_iner is set to zero
  cg_offst is set to zero
  sc_offst is set to zero
  tc_offst is set to zero
  edge_iner is set equal to flp_iner
  edge_stff is set equal to flp_stff

Appendix D

Fortran Code for Coupled Linear Springs

In this chapter the subroutine for the coupled linear springs representation of the foundation stiffness is presented. It was supplied by NREL.

Constant damping or added mass may optionally be included by adding damping coefficients to the *Damp*- or *PtfmAM*-matrix, respectively.

```
!========================================================================
!========================================================================
SUBROUTINE UserPtfmLd ( X, XD, ZTime, DirRoot, PtfmAM, PtfmFt )

! This routine implements the discrete coupled-springs foundation
!   model for the NREL 5MW Offshore Baseline Wind Turbine.  Since the
!   model is based on a simple linear stiffness matrix, the template
!   UserPtfmLd() routine from the FAST archive is used; the only
!   change being the specification of the stiffness matrix.  The
!   spring coefficients were derived by Patrik Passon of the
!   University of Stuttgart in Germany and are documented in:
!"OC3-Derivation and Description of the Soil-Pile-Interaction Models.pdf"
!"OC3-Soil-Pile_Interaction_Model.xls"
```

```
! The order of indices in all arrays passed to and from this routine
!   is as follows:
!      1 = Platform surge / xi-component of platform translation
!           (internal DOF index = DOF_Sg)
!      2 = Platform sway  / yi-component of platform translation
!           (internal DOF index = DOF_Sw)
!      3 = Platform heave / zi-component of platform translation
!           (internal DOF index = DOF_Hv)
!      4 = Platform roll  / xi-component of platform rotation
!           (internal DOF index = DOF_R )
!      5 = Platform pitch / yi-component of platform rotation
!           (internal DOF index = DOF_P )
!      6 = Platform yaw   / zi-component of platform rotation
!           (internal DOF index = DOF_Y )

! This routine was written by J. Jonkman of NREL/NWTC on August 25,
!   2006 for use in the IEA Annex XXIII OC3 studies.
! This routine was updated by J. Jonkman of NREL/NWTC on August 16,
!   2007.  I corrected the stiffness matrix based on my e-mail to
!   Jose Azcona Armendariz of CENER dated 4/4/2007.  That is, I
!   switched the signs of elements (2,4) and (4,2) based on the error
!   found by James Nichols of Garrad Hassan.

USE                             Precision

IMPLICIT                        NONE

   ! Passed Variables:

REAL(ReKi), INTENT(OUT)      :: PtfmAM (6,6)
REAL(ReKi), INTENT(OUT)      :: PtfmFt   (6)
REAL(ReKi), INTENT(IN )      :: X        (6)
REAL(ReKi), INTENT(IN )      :: XD       (6)
```

```
REAL(ReKi), INTENT(IN )      :: ZTime

CHARACTER(1024), INTENT(IN ) :: DirRoot

   ! Local Variables:

REAL(ReKi)                   :: Damp   (6,6)
REAL(ReKi)                   :: Stff   (6,6)

INTEGER(4)                   :: I
INTEGER(4)                   :: J

Damp(1,:)=(/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
Damp(2,:)=(/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
Damp(3,:)=(/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
Damp(4,:)=(/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
Damp(5,:)=(/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
Damp(6,:)=(/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)

Stff(1,:)=(/  2574800000.0,           0.0,0.0,
         &             0.0,-22532500000.0,0.0 /)

Stff(2,:)=(/           0.0,  2574800000.0,0.0,
         &   22532500000.0,           0.0,0.0 /)

Stff(3,:)=(/           0.0,           0.0,0.0,
         &             0.0,           0.0,0.0 /)

Stff(4,:)=(/           0.0,22532500000.0,0.0,
         &  262912300000.0,            0.0,0.0 /)

Stff(5,:)=(/-22532500000.0,            0.0,0.0,
         &             0.0,262912300000.0,0.0 /)
```

```
Stff(6,:)=(/           0.0,            0.0,0.0,
         &             0.0,            0.0,0.0 /)

PtfmAM(1,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(2,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(3,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(4,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(5,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(6,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)

PtfmFt(1)   = 0.0
PtfmFt(2)   = 0.0
PtfmFt(3)   = 0.0
PtfmFt(4)   = 0.0
PtfmFt(5)   = 0.0
PtfmFt(6)   = 0.0

DO J = 1,6
   DO I = 1,6
      PtfmFt(I) = PtfmFt(I) - Damp(I,J)*XD(J) - Stff(I,J)*X(J)
   ENDDO
ENDDO

RETURN
END SUBROUTINE UserPtfmLd
!========================================================================
```

Appendix E

Fortran Code for Noncoupled Linear Springs with Damping

In this chapter the subroutine for the noncoupled linear springs representation of the foundation stiffness is presented. It is based on the template for coupled linear springs that was supplied by NREL.

Constant damping is added, and the option to included added mass may is also available.

```
!==========================================================================
SUBROUTINE UserPtfmLd ( X, XD, ZTime, DirRoot, PtfmAM, PtfmFt )

   !      1 = Platform surge / xi-component of platform translation
   !      2 = Platform sway  / yi-component of platform translation
   !      3 = Platform heave / zi-component of platform translation
   !      4 = Platform roll  / xi-component of platform rotation
   !      5 = Platform pitch / yi-component of platform rotation
   !      6 = Platform yaw   / zi-component of platform rotation

USE                             Precision
```

```
IMPLICIT                        NONE

   ! Passed Variables:

REAL(ReKi), INTENT(OUT)      :: PtfmAM (6,6)
REAL(ReKi), INTENT(OUT)      :: PtfmFt   (6)
REAL(ReKi), INTENT(IN )      :: X        (6)
REAL(ReKi), INTENT(IN )      :: XD       (6)
REAL(ReKi), INTENT(IN )      :: ZTime

CHARACTER(1024), INTENT(IN ) :: DirRoot

   ! Local Variables:

REAL(ReKi)                   :: Damp   (6,6)
REAL(ReKi)                   :: Stff   (6,6)

INTEGER(4)                   :: I
INTEGER(4)                   :: J

Damp  (1,:) = (/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp  (2,:) = (/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp  (3,:) = (/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp  (4,:) = (/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp  (5,:) = (/ 0.0, 0.0, 0.0, 0.0, 3134519070.0, 0.0 /)
Damp  (6,:) = (/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)

Stff(1,:)=(/ 2574800000.0,          0.0, 0.0,
         &            0.0,          0.0, 0.0 /)

Stff(2,:)=(/          0.0, 2574800000.0, 0.0,
         &            0.0,          0.0, 0.0 /)

Stff(3,:)=(/          0.0,          0.0, 0.0,
```

```
         &            0.0,          0.0, 0.0 /)

Stff(4,:)=(/          0.0,          0.0, 0.0,
         & 262912300000.0,            0.0, 0.0 /)

Stff(5,:)=(/          0.0,          0.0, 0.0,
         &            0.0, 262912300000.0, 0.0 /)

Stff(6,:)=(/          0.0,          0.0, 0.0,
         &            0.0,            0.0, 0.0 /)

PtfmAM(1,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(2,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(3,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(4,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(5,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
PtfmAM(6,:) = (/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)

PtfmFt(1)   = 0.0
PtfmFt(2)   = 0.0
PtfmFt(3)   = 0.0
PtfmFt(4)   = 0.0
PtfmFt(5)   = 0.0
PtfmFt(6)   = 0.0

DO J = 1,6
   DO I = 1,6
      PtfmFt(I) = PtfmFt(I) - Damp(I,J)*XD(J) - Stff(I,J)*X(J)
   ENDDO
ENDDO



RETURN
```

```
END SUBROUTINE UserPtfmLd


!=======================================================================
```

Appendix F

Fortran Code for Nonlinear Foundation with Damping

In this appendix, the Fortran code used for the implementation of a nonlinear foundation is presented. Throughout the presentation of the code, for simplification, some of the code is left out. This is marked by ":". The complete original code is obtained when downloading the FAST archive from the NREL websites.

F.1 Subroutine UserPtfmLd

Here, the actual subroutine containing 20 parallel springs is presented. It was based on the template for coupled springs applied by NREL, but fundamental changes have been made. In this version, damping is also included. If the user would want to omit the damping term of the foundation, deleting the terms that includes *Damp* is sufficient. The code is as follows:

```
SUBROUTINE UserPtfmLd ( X, XD, ZTime, DirRoot, PtfmFt1, PtfmFt2, ...
...,  PtfmFt20, PtfmFt, PtfmFtC1, PtfmFtC2, ... , PtfmFtC20, QTC, QDTC )
! PtfmFt3 to PtfmFt19 and PtfmFtC3 to PtfmFtC19 should be included above
```

```
   ! The order of indices in all arrays passed to and from this routine
   !   is as follows:
   !      1 = Platform surge
   !      2 = Platform sway
   !      3 = Platform heave
   !      4 = Platform roll
   !      5 = Platform pitch
   !      6 = Platform yaw

USE                             Precision

IMPLICIT                        NONE

   ! Passed Variables:

REAL(ReKi), INTENT(OUT)      :: PtfmFt     (6)
REAL(ReKi), INTENT(OUT)      :: PtfmFt1    (6)
:
: ! PtfmFt2 to PtfmFt19 should be placed here
:
REAL(ReKi), INTENT(OUT)      :: PtfmFt20   (6)
REAL(ReKi), INTENT(IN )      :: PtfmFtC1   (6)
:
: ! PtfmFtC2 to PtfmFtC19 should be placed here
:
REAL(ReKi), INTENT(IN )      :: PtfmFtC20  (6)
REAL(ReKi), INTENT(IN )      :: QTC        (6)
REAL(ReKi), INTENT(IN )      :: QDTC       (6)
REAL(ReKi), INTENT(IN )      :: XD         (6)
REAL(ReKi), INTENT(IN )      :: X          (6)
REAL(ReKi), INTENT(IN )      :: ZTime

CHARACTER(1024), INTENT(IN ) :: DirRoot
```

```
   ! Local Variables:

REAL(ReKi)                    :: Damp       (6,6)
REAL(ReKi)                    :: Stff1      (6,6)
:
: ! Stff2 to Stff19 should be placed here
:
REAL(ReKi)                    :: Stff20     (6,6)
REAL(ReKi)                    :: FYield1      (6) = 0.0
:
: ! FYield2 to Fyield19 should be placed here
:
REAL(ReKi)                    :: FYield20     (6) = 0.0
REAL(ReKi)                    :: DeltaX       (6)
REAL(ReKi)                    :: Acc          (6)
REAL(ReKi)                    :: TUnload      (6)
REAL(ReKi)                    :: DUnload      (6)
REAL(ReKi)                    :: SignVel      (6)

INTEGER(4)                    :: A
:
: !Integers B to E
:
INTEGER(4)                    :: F
INTEGER(4)                    :: I
:
: !Integers J to V
:
INTEGER(4)                    :: W
INTEGER(4)                    :: y
INTEGER(4)                    :: Z

DO L= 1, 6
    DeltaX (L) = X(L) - QTC (L)
```

```
    SignVel(L) = XD(L) * QDTC (L)
ENDDO

!Assume linear variation of velocity
DO K=1,6
    Acc(K)=(XD(K)-QDTC(K))/0.0025
    IF (Acc(K)==0) THEN
        DUnload(K)=DeltaX(K)
    ELSE
        TUnload(K)=ABS((XD(K)))/(ABS(Acc(K)))
        DUnload(K)=0.5*Acc(K)*(TUnload(K)*TUnload(K))
    ENDIF
ENDDO

FYield1   (1) =  3554000.0
FYield1   (2) =  3554000.0
FYield1   (4) =  1779230.0
FYield1   (5) =  1779230.0
:
: ! FYield2 to FYield 19 should be placed here
:
FYield20  (1) = 17725000.0
FYield20  (2) = 17725000.0
FYield20  (4) =  8025000.0
FYield20  (5) =  8025000.0

Stff1(1,:)=(/355400000.0,        0.0,0.0,          0.0,          0.0,0.0 /)
Stff1(2,:)=(/        0.0,355400000.0,0.0,          0.0,          0.0,0.0 /)
Stff1(3,:)=(/        0.0,        0.0,0.0,          0.0,          0.0,0.0 /)
Stff1(4,:)=(         0.0,        0.0,0.0,35584600000.0,          0.0,0.0 /)
Stff1(5,:)=(/        0.0,        0.0,0.0,          0.0,35584600000.0,0.0 /)
Stff1(6,:)=(/        0.0,        0.0,0.0,          0.0,          0.0,0.0 /)
:
: ! Stff2 to Stff19 should be placed here
```

```
:
Stff20(1,:)=(/1772500.0,       0.0, 0.0,        0.0,            0.0, 0.0 /)
Stff20(2,:)=(/      0.0, 1772500.0, 0.0,        0.0,            0.0, 0.0 /)
Stff20(3,:)=(/      0.0,       0.0, 0.0,        0.0,            0.0, 0.0 /)
Stff20(4,:)=(/      0.0,       0.0, 0.0, 80250000.0,            0.0, 0.0 /)
Stff20(5,:)=(/      0.0,       0.0, 0.0,        0.0,     80250000.0, 0.0 /)
Stff20(6,:)=(/      0.0,       0.0, 0.0,        0.0,            0.0, 0.0 /)

Damp(1,:)=(/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp(2,:)=(/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp(3,:)=(/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp(4,:)=(/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)
Damp(5,:)=(/ 0.0, 0.0, 0.0, 0.0, 3134519070.0, 0.0 /)
Damp(6,:)=(/ 0.0, 0.0, 0.0, 0.0,          0.0, 0.0 /)

!Plastic Perfect Elastic Springs
!Spring nr 1
DO I = 1,6
    IF (PtfmFtC1 (I) >= 0.0) THEN
        IF (PtfmFtC1 (I) == FYield1 (I)) THEN
            IF (DeltaX (I) <= 0.0) THEN
                IF (SignVel (I) >= 0.0) THEN
                    PtfmFt1 (I) = PtfmFtC1 (I)
                ELSE
                    PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DUnload (I)
                ENDIF
            ELSE !DeltaX > zero.
                IF (XD (I) == 0.0) THEN
                    PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DeltaX (I)
                ELSE
                    PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DUnload (I)
                ENDIF
            ENDIF
        ELSE !on loading curve
```

```
            PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DeltaX (I)
        ENDIF
        IF (PtfmFt1 (I) > FYield1(I)) THEN
            PtfmFt1 (I) = FYield1(I)
        ENDIF
        IF (PtfmFt1 (I) < -FYield1(I)) THEN
            PtfmFt1 (I) = -FYield1(I)
        ENDIF
    ELSE !PtfmFtC1 < 0.0
        IF (PtfmFtC1 (I) == -FYield1 (I)) THEN
            IF (DeltaX (I) >= 0.0) THEN
                IF (SignVel (I) >= 0.0) THEN
                    PtfmFt1 (I) = PtfmFtC1 (I)
                ELSE
                    PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DUnload (I)
                ENDIF
            ELSE !DeltaX > zero.
                IF (XD (I) == 0.0) THEN
                    PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DeltaX (I)
                    ELSE
                    PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DUnload (I)
                ENDIF
            ENDIF
        ELSE
            PtfmFt1 (I) = PtfmFtC1 (I) - Stff1 (I,I) * DeltaX (I)
        ENDIF
        IF (PtfmFt1 (I) < -FYield1(I)) THEN
            PtfmFt1 (I) = -FYield1(I)
        ENDIF
        IF (PtfmFt1 (I) > FYield1(I)) THEN
            PtfmFt1 (I) = FYield1(I)
        ENDIF
    ENDIF
ENDDO
```

```
!Spring nr 2
:
: ! Repeat procedures for spring nr 2-20. Apply integers specified above.
:

DO M= 1, 6
    PtfmFt(M)=-Damp(M,M)*XD(M)+PtfmFt1(M)+...+PtfmFt20(M)
    !Include PtfmFt2 to PtfmFt19
ENDDO

RETURN
END SUBROUTINE UserPtfmLd
```

This code replaces the template for UserPtfmLd that already exists in UserSubs_forBladedDLL.f90. Keep in mind that UserSubs_forBladedDLL.f90 is a modified version of UserSubs.f90 thast is provided in the FAST archive when FAST is downloaded. The modification must be done by the user itself, and includes features that are necessary to model the 5MW NREL. This procedure is described by Jonkman [13].

F.2 Modification of Modules

The modules that were modified are named RtHndSid and Platform. These modules are already contained in the original FAST_Mods.f90-file. Some extra variables were introduced, and they are marked with "*!ADDED VARIABLE*". No variables were removed.

```
!=======================================================================
MODULE Platform

   ! This MODULE stores input variables for platform loading.

USE                             Precision

REAL(ReKi), ALLOCATABLE      :: LAnchxi  (:)
:
```

```
REAL(ReKi)                    :: PtfmDraft
REAL(ReKi)                    :: PtfmFt   (6) = 0.0
REAL(ReKi)                    :: PtfmFt1  (6) = 0.0 !ADDED VARIABLE
:
: ! PtfmFt2 to PtfmFt19 should be placed here
:
REAL(ReKi)                    :: PtfmFt20 (6) = 0.0 !ADDED VARIABLE
:
:
CHARACTER(1024)               :: WAMITFile

END MODULE Platform
!=====================================================================
:
!=====================================================================
MODULE RtHndSid

   ! This MODULE stores variables used in RtHS.

USE                              Precision

REAL(ReKi)                    :: AngAccEBt(3)
:
REAL(ReKi)                    :: PMXHydro (6,3)
REAL(ReKi)                    :: PtfmFtC1 (6)    = 0.0 !ADDED VARIABLE
:
: ! PtfmFtC2 to PtfmFtC19 should be placed here
:
REAL(ReKi)                    :: PtfmFtC20(6)    = 0.0 !ADDED VARIABLE
REAL(ReKi), ALLOCATABLE       :: QDT      (:)
REAL(ReKi)                    :: QDTC     (6)    = 0.0 !ADDED VARIABLE
REAL(ReKi), ALLOCATABLE       :: QD2T     (:)
REAL(ReKi), ALLOCATABLE       :: QD2TC    (:)
REAL(ReKi), ALLOCATABLE       :: OgnlGeAzRo(:)
```

```
REAL(ReKi), ALLOCATABLE      :: QT       (:)
REAL(ReKi)                   :: QTC      (6)    = 0.0 !ADDED VARIABLE
:
REAL(ReKi)                   :: TeetAngVel

END MODULE RtHndSid
!=======================================================================
```

F.3 Changes in Solver

Some small, but necessary, code was added to the original version of the subroutine Solver. Solver is contained in the FAST.f90-file. Changes are marked with "*!ADDED*".

```
!=======================================================================
SUBROUTINE Solver

   ! Solver solves the equations of motion by marching in time using a
   !   predictor-corrector scheme.  Fourth order Runge-Kutta is used to
   !   get the first 4 points from the initial degrees of freedom and
   !   velocities.

USE                             DOFs
USE                             RtHndSid
USE                             SimCont
USE                             TurbCont
USE                             Platform     ! ADDED

IMPLICIT                        NONE

   ! Local variables:

REAL(ReKi), ALLOCATABLE      :: ZK1      (:)
:
REAL(ReKi), ALLOCATABLE      :: ZK4D     (:)
```

```
INTEGER(4)                  :: I
INTEGER(4)                  :: Sttus

IF ( Step < 3 )  THEN
:
: ! Runge-Kutta integration
:
ELSE
:
: ! Adams-Bashforth predictor and Adams-Moulton corrector integration
:
ENDIF

QT  = Q (:,IC(NMX))
QDT = QD(:,IC(NMX))

CALL RtHS

QD2(:,IC(NMX)) = QD2T

! ADDED:
PtfmFtC1  = PtfmFt1
:
: ! PtfmFtC2 = PtfmFt2 to PtfmFtC19 = PtfmFt19 should be included here
:
PtfmFtC20 = PtfmFt20

! ADDED:
QTC = QT (1:6)
QDTC = QDT (1:6)

IC(1) = IC(1) + 1
IF ( IC(1) > NMX )  IC(1) = IC(1) - NMX
```

```
DO
:
: ! Update IC() index so IC(1) is the location of current Q values.
:
ENDDO
:
:
:

RETURN
END SUBROUTINE Solver
!=======================================================================
```