



NTNU – Trondheim
Norwegian University of
Science and Technology

Stochastic Analysis of an Offshore Wind Turbine Using a Simplified Dynamic Model

John Magne Hembre

Civil and Environmental Engineering

Submission date: June 2014

Supervisor: Michael Muskulus, BAT

Norwegian University of Science and Technology
Department of Civil and Transport Engineering



Report Title: Stochastic Analysis of an Offshore Wind Turbine Using a Simplified Dynamic Model	Date: 20/12/2013		
	Number of pages (incl. appendices): 144		
	Master Thesis	X	Project Work
Name: John Magne Hembre			
Professor in charge/supervisor: Michael Muskulus			
Other external professional contacts/supervisors: Sebastian Schafhirt			

<p>Abstract:</p> <p>The objective of this thesis is to propose and evaluate a simplified dynamic model of a wind turbine. This is done by assuming that the structure behaves linearly, and that a load model based of the mean and variance of simulated rotor thrust, depending linearly on turbulence, is sufficient to estimate the response.</p> <p>The realistic response of the wind turbine is calculated in time domain using the NREL FAST finite element code, whereas the simplified model response are calculated analytically because of the Gaussian behavior of the load model.</p> <p>It is suspected that more sophisticated simplified load models in the future might not be linear, and the response of the simplified system is therefore also calculated using the relatively new probability density evolution method. This is done to assess the general feasibility of using this method to estimate responses of such models. Probability density evolution is performed numerically using the cell-mapping method, and evaluation of the accuracy and performance of these calculations is also a primary object of the thesis. It is also shown how autoregression could be used to incorporate correlated forcing in the cell-mapping method.</p> <p>The wind turbine considered is the NREL 5-MW Baseline turbine, supported by its reference tower.</p> <p>Results show that the response of the proposed simplified dynamic system is too coarse compared to the simulated response distributions. The model seems to overestimate both the mean value and the variance of the true displacements. Possible improvements of the dynamic system, especially of the simplified load model, is evaluated and discussed.</p> <p>The results also show that first order autoregression successfully implements correlated force realizations into the cell-mapping method, and that both Kaimal simulated turbulence and the proposed simplified thrust are processes that could be estimated to a satisfactory degree using first order autoregression.</p> <p>With correlated forcing incorporated through autoregression, probability density evolution is shown to provide accurate estimates of the model response. The method also shows great promise to be useful when considering future simplified models, provided that one could find ways to keep the number of state-space dimensions to a minimum.</p>

Keywords:

- | |
|---|
| 1. Offshore Wind Turbine, Simplified Dynamic Model |
| 2. Probability Density Evolution, Cell-Mapping |
| 3. Structural Dynamics, Modal Analysis |
| 4. Stochastic Environmental Loading, Autoregression |

(signature)

Acknowledgements

I want to thank my thesis supervisors, Professor Michael Muskulus and PhD Candidate Sebastian Schafhirt, for all advice, discussions, and help.

Companionship and rewarding discussions, both on- and off-topic, with my friends and fellow students at the NTNU Department of Structural Engineering was also very much appreciated when writing this thesis.

Finally, I want to thank Anett and my family for all their support throughout my studies.

Contents

1	Introduction	1
2	Theory	3
2.1	Stochastic Dynamics of a Linear Single-Degree-of-Freedom System	3
2.1.1	System Properties	3
2.1.2	Stochastic Properties of the Response	6
2.1.3	Example of a SDOF System Response to Stationary White Noise Gaussian Loading	9
2.2	Probability Density Evolution	15
2.2.1	Markov Processes and the Fokker-Planck Equation	16
2.2.2	Numerical Solution of the Fokker-Planck Equation using the Cell-to-Cell Mapping Method	17
2.3	Environmental Loading	19
2.3.1	Wind	19
2.3.2	The Blade Element Momentum Method	23
2.3.3	Waves	25
2.4	Autoregressive Modeling	27
2.5	The Finite Element Method in Structural Dynamics	30
2.5.1	Deriving the General Equation of Motion	31
2.5.2	Modal Analysis	33
3	Methodology	35
3.1	The NREL 5-MW Baseline Wind Turbine	35
3.1.1	Reference Tower	35
3.2	TurbSim	37
3.3	FAST	38
3.4	Environmental Load Cases	38
3.5	Development of a Simplified Load Model	39
3.6	The Probability Density Evolution Script	42
3.6.1	Basic PDE Functionality with Uncorrelated Forcing	42
3.6.2	Incorporating Correlated Forcing Through First Order Autoregressive Modeling	43
3.6.3	General Comments on Numerical Cell-to-Cell Mapping	48
3.7	Calculating the Nonlinear First Mode Response Contribution from Wind Loading Along the Tower	50
3.7.1	General Description of the Modal Force	50
3.7.2	Probability Density Function of the Modal Force	51
3.7.3	Constants of the Modal Force and Resulting Probability Density Function for Load Case 2	54

4	Results	57
4.1	Modal Properties of the Reference Tower	57
4.2	Simplified Thrust Force Model	59
4.3	Estimated First Order Autoregression Parameters	70
4.3.1	AR(1) Parameters Describing Turbulence	70
4.3.2	AR(1) Parameters Describing Simplified Thrust Fluctuations	70
4.4	Response of Tower Top to Rotor Thrust	72
4.5	First Mode Response Contribution from Nonlinear Wind Loads Along the Tower for Load Case 2	80
5	Discussion	83
6	Conclusion	89
7	Attachments	93
7.1	Attachment 1: Matlab script used to calculate analytical SDOF response to stochastic excitation	93
7.2	Attachment 2: Matlab script used to evolve probability distributions of structural response through the cell-mapping method	97
7.3	Attachment 3: Matlab script used to simulate wind turbulence through spectral decomposition of the Kaimal spectra	107
7.4	Attachment 4: Matlab script illustrating calculations of total thrust and torque at the hub using reference blade properties and the Blade Element Momentum Method	109
7.5	Attachment 5: Matlab script used to estimate AR(1) parameter values describing turbulence	117
7.6	Attachment 6: Flexibility matrix of the 20 DOF discretized reference tower	121
7.7	Attachment 7: Matlab script used to derive the modal properties of the reference tower	123
7.8	Attachment 8: Example of a TurbSim input file	127
7.9	Attachment 9: Matlab script used to analyze FAST simulation output and estimate parameters in the simplified thrust model	131
7.10	Attachment 10: Plots of the estimated turbulence AR(1) spectra and their corresponding Kaimal spectra	137
7.11	Attachment 11: Plots of the estimated tower top velocity responses for the first mode	141
	References	143

List of Figures

1	Illustration of a dynamic single-degree-of-freedom system.	3
2	Plots of $h(t)$ and $ H(\omega) ^2$ for a SDOF system where $m = 1$ kg, $k = 10$ N/m and $\zeta = 5$ %.	10
3	Plots of the example one-sided load spectrum and response variance spectrum.	11
4	Time series of the displacement and velocity response of the system.	11
5	Normalized histograms showing realizations of $f(t)$, $u(t)$ and $\dot{u}(t)$ during a 1500 second simulation. Plotted with the analytical Gaussian distributions for the random variables $F(t)$, $U(t)$ and $\dot{U}(t)$	12
6	The analytical joint probability density function of $U(t)$ and $\dot{U}(t)$	14
7	Normalized joint histogram over the realization pairs of $u(t)$ and $\dot{u}(t)$ for a 1500 second long Monte Carlo simulation with a time step of $\Delta t = 0.05$ s, resulting in 30000 sample points.	15
8	Joint probability density of $U(t)$ and $\dot{U}(t)$ from section 2.1.3, calculated using probability density evolution.	18
9	Superposition of time dependent and mean wind velocities.	19
10	Logarithmic x-axis plot of the Kaimal spectrum for $L_k = 340.2$ m, $v_{hub} = 16$ m/s and turbulence class B.	21
11	Matlab plot of simulated wind velocity fluctuations over 10 minutes in the main wind direction at hub height, given the specified Kaimal spectrum from figure 10.	22
12	Air velocity components around a rotor blade airfoil.	23
13	Estimated total rotor thrust resultant for the NREL 5-MW Baseline Turbine given a 10 m/s reference wind speed with turbulence class B.	25
14	Example of a Pierson-Moskowitz spectrum with significant wave height of 2.19 meters and a spectral peak period of 6.37 seconds.	26
15	An example of a 200 seconds long time domain simulation of the surface elevation, given the sea state described by the PM-spectrum in figure 14.	27
16	Comparison of the Kaimal spectrum from figure 10 and the best estimate of its first order autoregression spectrum for time step $\Delta t = 0.05$ s.	29
17	AR(1) simulation of the wind turbulence defined by the spectrum in figure 10.	30
18	Illustration of a wind velocity field generated by TurbSim.	37
19	Plot of the class B turbulence intensities for the different reference wind velocities considered.	39
20	Comparison between a FAST simulation of the reference turbine for a 10 m/s reference wind speed in turbulence class B, and a corresponding simplified model based on the squared wind velocity.	40
21	Contour plot of the joint probability density function from figure 6.	44
22	300 second simulated time series, with histogram and analytical PDF, of an AR(1) load process with $\mu_F = 30$ N, $\phi = 0.7$, $\sigma^2 = 12.75$ N ² and $\Delta t = 0.05$ s.	45
23	Joint probability distribution of the previous and current value of the described load fluctuations in this example.	45

24	Log-log plot of the one-sided variance spectrum of the autoregressive load process simulated in figure 22, the constant white noise spectra of the same time step and PDF, and the squared frequency response function of the example structure.	46
25	Contour plot showing the analytical joint PDF of the structural response to the autoregressive load process from figure 22 and 24.	46
26	The joint probability distribution of the displacement and velocity response for the example in section 3.6.2, estimated using the probability density evolution script with correlated excitation in attachment 2.	47
27	Illustration of iteration errors associated with the state space discretization.	48
28	Sketch of the modal force from equation (106) as a function of the random variable V	52
29	Reynolds number at different heights of the tower for $V_{ref} = 4$ m/s and $\alpha = 0.14$	54
30	Histogram of simulated modal drag force realizations for the first mode subjected to load case 2. Plotted with the analytical PDF of the modal force for $r_c > 0.83$ N	55
31	Histogram of simulated modal drag force realizations for the second mode subjected to load case 2. Plotted with the analytical PDF of the modal force for $r_c > 1.89$ N	56
32	Plots of the first three normalized tower mode shapes for transversal displacements.	58
33	Plot showing the fit of the simplified load model with respect to the mean thrust value.	60
34	Plot showing the fit of the simplified load model with respect to the turbulence proportionality constant $\hat{C}_s V_{ref}^\beta$	60
35	Time series and PSD plots comparing FAST simulations to the simplified load model for load cases 1-3.	62
36	Time series and PSD plots comparing FAST simulations to the simplified load model for load cases 4-6.	63
37	Time series and PSD plots comparing FAST simulations to the simplified load model for load cases 7-9.	64
38	Time series and PSD plots comparing FAST simulations to the simplified load model for load cases 10-12.	65
39	Time series and PSD plots comparing FAST simulations to the simplified load model for load cases 13-15.	66
40	Sample PSD plots at low frequencies for load case 2, with marked multiples of the rotor frequency, 1P.	67
41	Sample PSD plots at low frequencies for load case 10, with marked multiples of the rotor frequency, 1P.	67
42	100 second autocovariances of the different thrust force fluctuations for load case 2, $V_{ref} = 4$ m/s.	68
43	100 second autocovariances of the different thrust force fluctuations for load case 4, $V_{ref} = 8$ m/s.	68
44	100 second autocovariances of the different thrust force fluctuations for load case 8, $V_{ref} = 16$ m/s.	69

45	100 second autocovariances of the different thrust force fluctuations for load case 12, $V_{ref} = 24$ m/s.	69
46	Tower top displacement for the 15 load cases estimated from time domain simulations in FAST, and calculated analytically for the first mode using the simplified load model.	72
47	Plots showing the two ratios from table 10.	74
48	Probability distributions of tower top displacement for load case 2.	75
49	Probability distributions of tower top displacement for load case 6.	75
50	Probability distributions of tower top displacement for load case 4.	76
51	Semi-logarithmic plots of the thrust variance spectra near the natural frequency of mode one for load case 4. Plotted with the transfer function of the modal coordinate response.	76
52	Probability distributions of tower top displacement for load case 10.	77
53	Semi-logarithmic plots of the thrust variance spectra near the natural frequency of mode one for load case 10. Plotted with the transfer function of the modal coordinate response.	77
54	Contour plot showing the joint probability density of the displacement and velocity response, estimated using 1800 second FAST simulations.	78
55	Contour plot showing the joint probability density of the displacement and velocity response for the first mode, estimated using the simplified load model.	79
56	Tower top displacement distributions calculated analytically for the second mode using the simplified load model.	80
57	Estimated probability distributions of first mode tower top displacement for wind loads distributed along the tower. The wind loads are described by load case 2.	80
58	Result of a 1200 second Monte Carlo simulation of the tower top response under LC2 wind loads along the tower, for the first mode.	81
59	The tower top response under LC2 wind loads along the tower, estimated for the first mode using white noise probability density evolution.	81
60	AR(1) turbulence spectra and Kaimal spectra for load cases 1 to 4.	138
61	AR(1) turbulence spectra and Kaimal spectra for load cases 5 to 8.	138
62	AR(1) turbulence spectra and Kaimal spectra for load cases 9 to 12.	139
63	AR(1) turbulence spectra and Kaimal spectra for load cases 13 to 15.	139
64	Tower top velocity for the 15 load cases estimated from time domain simulations in FAST, and calculated analytically for the first mode using the simplified load model.	142

List of Tables

1	Some selected properties of the reference wind turbine.	35
2	Discretization and lumped masses of the reference tower.	36
3	The wind velocity load cases considered in this thesis.	38
4	The constants of the modal force as defined in equation (106), caused by wind loads distributed along the tower, for the first two fore-aft tower modes.	55
5	Table over the first three normalized tower mode shapes for transversal displacements.	57
6	Properties of the first three tower modes.	58
7	Estimated simplified load model parameters.	59
8	Table showing estimated AR(1) parameters describing turbulence in the 15 load cases for time steps of $\Delta t = 0.0125$ s and $\Delta t = 0.05$ s.	70
9	Table showing estimated AR(1) parameters describing thrust fluctuations in the 15 load cases for time steps of $\Delta t = 0.0125$ s and $\Delta t = 0.05$ s.	71
10	Table presenting statistical properties of estimated tower top displacements.	73
11	Flexibility matrix of the discretized reference tower.	122

Sammendrag

På grunn av kompleksiteten til den ikke-lineære strukturelle oppførselen og dynamiske belastningen av en offshore vindturbin vil den beste beskrivelsen av responsen gis av stokastiske elementmetodesimuleringer i tidsplanet.

Slike simuleringer er imidlertid beregningsmessig kostbare, spesielt hvis man må utføre mange nok til å estimere utmattingskader. Det er derfor av interesse å studere hvor nøyaktig en forenklet dynamisk modell kan tilnærme denne responsen.

Denne masteroppgaven tar sikte på å foreslå og evaluere en forenklet dynamisk modell av en vindturbin. I denne grunnleggende studien er det interessant å avdekke hvor godt responsen tilnærmes hvis denne modellen gjøres så enkel å utlede og bruke som mulig. De realistiske responsfordelingene av tårnet i hovedvindretningen er estimert for 15 ulike lasttilfeller ved bruk av elementmetodesimulering i tidsplanet. Koden som er benyttet til dette heter FAST og er utviklet av the U.S. Department of Energy's National Renewable Energy Laboratory (NREL). Responsfordelingene til den forenklete dynamiske modellen er funnet ved hjelp av den relativt nyutviklede sannsynlighets-evolusjonsmetoden og, siden den forenklete lastmodellen som vurderes er lineær og Gaussisk, også analytisk ved hjelp av stokastisk dynamikkteori.

Sannsynlighets-evolusjon har flere attraktive egenskaper forbundet med å estimere responsfordelingen til slike forenklete dynamiske modeller med få frihetsgrader, spesielt for potensielle framtidige systemer med ikke-lineære lastmodeller. Den største fordelene med denne metoden er at alle regioner av fordelingen estimeres med samme effektivitet. Dette er nyttig når sannsynlighetsfordelingenes haler er viktige, noe de er ved beregning av utmattingskader.

På grunn av styrkene forbundet med sannsynlighets-evolusjon er det også en vesentlig del av målsettingen til denne avhandlingen å evaluere nøyaktigheten og effektiviteten til denne metoden for de aktuelle dynamiske systemene. Sannsynlighets-evolusjon er utført numerisk ved bruk av cell-mapping metoden, framsatt av Hsu i [1].

For å forbedre denne metoden for problemer som involverer korrelert eksitasjon er det også vist hvordan autoregresjon kan brukes til å innlemme betingede sannsynligheter av lastrealiseringene i cell-mapping metoden.

Vindturbinen som studeres i denne avhandlingen er NRELS 5-MW referanseturbin, med tilhørende referansetårn. [2] Den forenklete lastmodellen, som er modellert med lineær turbulensavhengighet, er tilpasset til simuleringresultater av skyvekraften fra rotoren for de 15 lasttilfellene slik at den gir tilnærmet samme gjennomsnittsverdi og varians.

Ved å sammenlikne elementmetoderesultater og estimerte responsfordelinger ved bruk av den forenklete modellen er det klart at denne modellen bør forbedres for å gi mer nøyaktige estimat av responsen.

Den største forenklingen i den foreslåtte lastmodellen er at belastningens variansspekter antas å være tilnærmet proporsjonalt til turbulensens variansspekter. Resultatene viser at dette medfører at periodiske skyvekraftbidrag fra rotoren neglisjeres i lastmodellen, samtidig som lastvariansen fra disse bidragene ivaretas over alle frekvenser siden modellen er tilpasset den totale realistiske lastvariansen. Dette mistenkes å ligge bak størsteparten av avviket mellom de simulerte og estimerte responsfordelingene, siden det utvilsomt vil påvirke responsvariansen. Variansen av den modellerte responsen kan generelt sies å være for høy.

Det påvises også at de estimerte gjennomsnittsverdiene av forskyvningene generelt virker å være høye sammenliknet med de reelle gjennomsnittsverdiene, noe som kan tilsa at den virkelige ikke-lineære strukturen oppfører seg stivere enn den lineariserte strukturen brukt i den forenklete modellen.

Resultatene viser også at både turbulens simulert ved bruk av Kaimalspekteret, og den forenklete lastprosessen, kan estimeres tilfredsstillende ved bruk av førsteordens autoregresjon. Korrelasjon av de modellerte lastrealiseringene er derfor vellykket inkludert i cell-mapping metoden for dynamiske modeller av denne typen, og sannsynlighets-evolusjonsmetoden er da godt egnet til å nøyaktig estimere responsen.

Dette er den første avhandlingen som er skrevet om dette emnet ved NTNUs Offshore Vindteknologi-gruppe, og selv om en tilfredsstillende forenklet dynamisk modell ikke er fullt utviklet enda har vi lært mye om hvordan man kan fortsette arbeidet mot å etablere en slik modell. Vi har også fått verdifull innsikt i sannsynlighets-evolusjonsmetoden, og hvordan denne kan brukes til å numerisk estimere en vindturbinns globale responsfordeling.

Summary

Given the complexity of both the structure and the dynamic loading situation, time domain simulations using aero-hydro-servo-elastic finite element codes provides the most accurate way to determine the response of an offshore wind turbine.

However, such calculations are computationally expensive, especially when one needs to perform enough simulations to estimate fatigue damage. It is therefore of interest to study how accurate a simplified dynamic model could estimate this response.

The main objective of this thesis is to propose and evaluate a simplified dynamic model of a wind turbine. In this initial study it is interesting to see how well a model as simple as possible would perform in estimating the structural response.

The realistic response distributions are estimated using the U.S. Department of Energy's National Renewable Energy Laboratory (NREL) FAST code to perform Monte Carlo simulations of the turbine in time domain. Response of the simplified dynamic model is estimated with the relatively new probability density evolution method and, because of the Gaussian linearity of the proposed load model, also using random vibrations theory directly.

Probability density evolution possess several strengths associated with estimating the response distributions for such problems, especially when a nonlinear load model might be considered in the future. The main advantage is that all areas of the distribution is treated equally, enabling an efficient estimation of the distribution tails as well. Evaluating the performance and accuracy of this method is therefore also a key objective of the thesis. Probability evolution is performed numerically using the cell-mapping method derived by Hsu in [1]. In order to make this method more accurate for problems involving correlated forcing it is also shown how autoregression could be used to incorporate conditional probabilities of force realizations into the cell-mapping method.

The wind turbine being considered in this thesis is the NREL 5-MW Baseline turbine, supported by an onshore reference tower. [2] The simplified load model in this study, which has a linear dependence on wind turbulence, is fitted to simulated thrust values for 15 load cases in order to obtain a similar mean and variance.

By comparing finite element simulations and estimated response distributions using the simplified model, it is clear that the dynamic model would have to be improved in order to provide more accurate estimates of the response. The largest simplification in the proposed load model is that it does not account for cyclic thrust contributions from the rotor, while the variance contained in spectral peaks at these frequencies are still included in the model. This is suspected to be the main reason for discrepancies between the simulated and estimated response distributions, as it certainly affects the calculated response variance. This variance is in general observed to be too high.

It is also seen that the estimated mean value of the displacement response seems high compared to simulation results, which could imply that the realistic nonlinear structure acts stiffer than the linearized model structure used in this thesis.

The results also show that both Kaimal simulated wind turbulence and the proposed simplified thrust are accurately estimated using first order autoregression.

Correlation of the modeled force realizations are therefore successfully implemented in the cell-mapping method, and probability density evolution is now well suited to provide

good estimates of the model response.

This is the first thesis written on this subject at the NTNU Offshore Wind Turbine Technology group, and even though a satisfactory simplified dynamic model is not fully developed yet, we have indeed learned a great deal on how one could proceed the work towards establishing such a model. We have also gained valuable insight into the probability density evolution method, and how it could be used to numerically estimate the global probability distribution of the structural response.

1 Introduction

International focus on renewable energy has increased over the recent decades, partly motivated by a will to reduce emissions of greenhouse gasses and to shift the total energy production towards a situation more sustainable than the present. In Europe, some nations are also politically motivated to become independent of foreign energy. Higher environmental consciousness among consumers is also a factor that could make renewable energy more economically viable through the use of "Green Certificates".

Offshore wind energy could contribute substantially to the percentage of total renewable energy generated, and it is therefore a market with potential for considerable growth in the years to come.

This focus has naturally also resulted in an increased research activity on offshore wind turbines, like the Offshore Code Comparison Collaboration Continuation (OC4) Project. OC4 is a multinational research project that aims to compare dynamic computer codes used to design offshore turbines and their support structures, with the ultimate goal of making structure designs, and offshore wind energy in general, more cost efficient.

These are capable of nonlinear finite element simulations in time domain considering aerodynamics, hydrodynamics and turbine control system. Because of this, they are in general named aero-hydro-servo-elastic codes.

Offshore wind turbines is subjected to complex dynamic loads that are high compared to the self-weight of the structure. Nonlinearities and coupling effects would affect the response of the structure, and such time domain simulations would undoubtedly provide the most accurate assessment of this response. However, such simulations are computationally expensive, in particular when fatigue loads needs to be obtained. It is therefore of interest to study how good a simplified dynamic model could estimate the structural response.

Such a simplified dynamic model could for example be used in an early design phase to compare different support structures. A more ambitious goal is that a model like this over time should become accurate enough to replace, at least some, time domain simulations entirely. This could for example be accomplished by having a model with parameters that could be calibrated using time domain simulations.

One of the main objectives in this thesis is to establish a simplified dynamic model of a wind turbine and its load situation, and evaluate the calculated response compared to results obtained by time domain simulations using a certified finite element code. The response distribution could for some simple dynamic models be calculated analytically, but in general one would have to compute this distribution by numerical means if the structure model or load model should be described nonlinearly.

Monte Carlo simulation provides a straightforward method of obtaining an estimate of the response distribution. This method is however computationally expensive, especially when fatigue is important and an accurate estimate of the probability content of the tails need to be obtained.

An alternative method, which is subjected to ongoing research, is the probability density evolution method. This is a method that is capable of evolving the entire distribution using instantaneous probabilities. This makes it just as efficient when estimating low probability areas of the distribution. The Monte Carlo Method on the other hand, would

need many sample points to capture low probability response. Probability evolution has therefore proven to be effective compared to Monte Carlo simulation when a global distribution estimated is important. [3] This is also true when comparing probability evolution to improved Monte Carlo methods, such as those described by Robert and Casella in [4].

Like for Monte Carlo simulation, the efficiency of probability evolution could also be dramatically increased by parallel computing using a graphics processing unit.

Because of the advantages associated with this relatively new method, a significant part of the thesis is also dedicated to probability density evolution. This is the first thesis written on this subject at the NTNU Offshore Wind Turbine Technology group, and emphasis have been laid on gaining knowledge about the method and how it behaves for problems relevant to the group. Results obtained using the probability density evolution method is therefore evaluated and compared to those obtained by Monte Carlo simulations or analytical stochastic dynamics theory.

It is also shown how autoregressive modeling could be used to incorporate correlated excitation into probability evolution of the structural response, a subject on which there is no earlier publications as far as the author knows.

The thesis is divided into five main sections. Section 2 cover some of the basic background theory regarding stochastic structural dynamics, stochastic loads, probability density evolution and autoregression. Section 3 presents the methods used and section 4 presents the results obtained. The results are discussed in section 5, and finally conclusions and recommendations for further research are given in section 6. Essential Matlab scripts and other attachments could be found in section 7.

2 Theory

The following theory sections will introduce some of the main principles used in the subsequent stochastic analysis of this thesis. This is done rather to give an overview of the background theory than to provide a complete, in-depth, review of each subject. The reference literature would in many cases be a good source of detailed information for the interested reader.

2.1 Stochastic Dynamics of a Linear Single-Degree-of-Freedom System

The physics behind the response of a single-degree-of-freedom (SDOF) system as illustrated in figure 1 is important. This is because it is the most basic conceivable dynamic system including damping and inertia, and because relatively complex structures could be simplified and represented through generalized coordinates that behave like SDOF systems. The following theory of stochastic dynamics is largely based on literature by Lutes and Sarkani [5], Næss and Moan [6], and Newland. [7]

2.1.1 System Properties

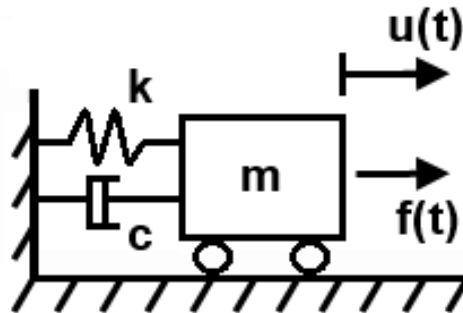


Figure 1: Illustration of a dynamic single-degree-of-freedom system.

The equation of motion for a SDOF system with spring stiffness k , mass m and viscous damping coefficient c , excited by some force $f(t)$, is given as an equilibrium between the internal inertia, damping and elastic force of the system, and the external force applied:

$$m \ddot{u}(t) + c \dot{u}(t) + k u(t) = f(t). \quad (1)$$

In equation (1), $u(t)$ is the time dependent displacement of the mass from the unloaded equilibrium position of the system. The dots denotes the time derivatives of this displacement. The mass velocity and acceleration is then given as $\frac{du}{dt} = \dot{u}(t)$ and $\frac{d^2u}{dt^2} = \ddot{u}(t)$, respectively.

Introducing the damping ratio, $\zeta = c/(2\sqrt{km})$, and the natural angular frequency of the structure, $\omega_0 = \sqrt{k/m}$, equation (1) could be written as:

$$\ddot{u}(t) + 2\zeta\omega_0 \dot{u}(t) + \omega_0^2 u(t) = \frac{f(t)}{m}. \quad (2)$$

The solution of the differential equation above naturally depends on the excitation of the system, and analytical solutions for the special cases where $f(t) = 0$, or where $f(t)$ is some harmonic function, is for example given by Erwin Kreyzig. [8] This thesis deals with arbitrary non-periodic excitation functions. In such cases, the response of the system could be calculated in the frequency domain using the frequency response function, $H(\omega)$, or in the time domain using the impulse response function, $h(t)$.

The frequency response function of the system due to the external force is found by demanding the harmonic displacement $u(t) = \eta e^{i\omega t}$ in the equation of motion, where η is an amplitude. This yields:

$$\begin{aligned} \frac{d^2}{dt^2} (\eta e^{i\omega t}) + 2\zeta\omega_0 \frac{d}{dt} (\eta e^{i\omega t}) + \omega_0^2 \eta e^{i\omega t} &= \frac{f(t)}{m} \\ \Rightarrow ((i\omega)^2 + 2i\zeta\omega_0\omega + \omega_0^2) \eta e^{i\omega t} &= \frac{f(t)}{m} \\ \Rightarrow \eta e^{i\omega t} = H(\omega) f(t) &= \frac{1}{m(\omega_0^2 + 2i\zeta\omega_0\omega - \omega^2)} f(t) \end{aligned}$$

and therefore:

$$H(\omega) = \frac{1}{m(\omega_0^2 + 2i\zeta\omega_0\omega - \omega^2)}. \quad (3)$$

The system response could now be found in the frequency domain by using the Fourier transformation of the force, $F(\omega)$, as shown in equation (4).

$$\begin{aligned} u(t) &= \int_{-\infty}^{\infty} H(\omega) F(\omega) e^{i\omega t} d\omega \\ &= \int_{-\infty}^{\infty} H(\omega) \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \right) e^{i\omega t} d\omega \end{aligned} \quad (4)$$

As mentioned, the system response could also be calculated in the time domain. This is done by using the Duhamel convolution integral shown below.

$$u(t) = \int_{-\infty}^{\infty} f(s) h(t-s) ds \quad (5)$$

Here, $h(t)$ is the response of the system due to a unit impulse load at $t = 0$. The integral in equation (5) could be thought of as a superposition at time t of all impulse responses

up to t , and the impulse response at t . Introducing the substitution $r = t - s$ results in equation (6).

$$u(t) = \int_{-\infty}^{\infty} f(t-r) h(r) dr. \quad (6)$$

For causal systems there is no response prior to the load, and $h(r) = 0$ for $r < 0$ (or $t < s$). The impulse response function $h(t)$ could be found as the Fourier transform of the frequency response function $H(\omega)$, since it is shown, e.g. in [6], that these two functions form a Fourier transform pair.

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{i\omega t} d\omega \quad (7)$$

Another, perhaps more intuitive, method to determine $h(t)$ is by exciting the system with a unit impulse load and solve the resulting equation of motion. This is done through the Dirac delta function, given as:

$$\delta(t) = \begin{cases} \infty & \text{for } t = 0 \\ 0 & \text{for } t \neq 0. \end{cases} \quad (8)$$

Combined with equation (1), this results in equation (9).

$$m \frac{d^2}{dt^2} h(t) + c \frac{d}{dt} h(t) + k h(t) = \delta(t) \quad (9)$$

This is a second order homogeneous ordinary differential equation for $t > 0$ as $\delta(t \neq 0) = 0$. For lightly damped structures where $|\zeta| < 1$ it has the general solution:

$$h(t) = e^{-\zeta\omega_0 t} (A \cos(\omega_d t) + B \sin(\omega_d t)) \quad (10)$$

Where $\omega_d = \omega_0 \sqrt{1 - \zeta^2}$ is the damped natural frequency of the system. A and B are constants that need to be determined from the initial conditions at $t = 0$. From equations (8) and (9) it is clear that one of the $h(t)$ time derivatives has to be infinite at $t = 0$. This is problematic unless it is the double derivative that is infinite. In this case however, if $d^2/dt^2(h(t)) = b\delta(t)$ for small $|t|$, then the two other terms on the right side of equation (9) is finite, $m b \delta(t) = \delta(t)$ and therefore $b = m^{-1}$. By integration, it is found that

$$\left(\frac{d}{dt} h(t) \right)_{0+} = m^{-1} \quad (11)$$

and

$$h(t)_{0+} = 0. \quad (12)$$

The initial conditions from equation (11) and (12) put into the solution for $h(t)$ from (10) yields that $A = 0$ and $B = 1/(\omega_d m)$. The resulting impulse response function is shown in equation (13).

$$h(t) = \frac{e^{-\zeta\omega_0 t}}{\omega_d m} \sin(\omega_d t) \quad \text{for } t \geq 0 \quad (13)$$

The same result could naturally be obtained by using equation (7). According to equation (5), the velocity response at time t could also be found by time differentiation as

$$\dot{u}(t) = \int_{-\infty}^{\infty} f(s) \dot{h}(t-s) ds \quad (14)$$

where $\dot{h}(t-s)$ is found for $t-s \geq 0$ from equation (13).

$$\dot{h}(t-s) = \frac{e^{-\zeta\omega_0(t-s)}}{\omega_d m} [\omega_d \cos(\omega_d(t-s)) - \zeta\omega_0 \sin(\omega_d(t-s))] \quad (15)$$

2.1.2 Stochastic Properties of the Response

$f(t)$ could be described as a realization of a stochastic load process $F(t)$. This forcing causes a displacement response $u(t)$, which could be described as a realization of the stochastic process $U(t)$. It could be shown that for a linear, time invariant structure as the SDOF system described in 2.1.1, the response would be a stationary stochastic process provided that $F(t)$ is stationary. Equation (6) gives the connection between the two processes shown below.

$$U(t) = \int_{-\infty}^{\infty} F(t-s) h(s) ds \quad (16)$$

The mean value of the response process is now given as shown in equation (17).

$$\begin{aligned} E[U(t)] &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N u_j(t) \\ &= \int_0^{\infty} h(s) \left(\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N f_j(t-s) \right) ds \\ &= \int_0^{\infty} h(s) E[F(t-s)] ds \end{aligned} \quad (17)$$

$F(t)$ could often be modeled as a stationary process for short timespans relevant for simulations. In this case the mean of the forcing will be constant in time, $E[F(t-s)] = \mu_F$, and therefore

$$E[U(t)] = \mu_F \int_0^\infty h(s) ds. \quad (18)$$

The velocity response of the structure could also be described as a stochastic process denoted $\dot{U}(t)$. This process would have a mean value of

$$\begin{aligned} E[\dot{U}(t)] &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N \frac{d}{dt} u_j(t) \\ &= \int_0^\infty \dot{h}(s) E[F(t-s)] ds. \\ &= \frac{d}{dt} E[U(t)]. \end{aligned} \quad (19)$$

From the equation above one could easily see that the mean value of $\dot{U}(t)$ would be zero for a stationary and time differentiable process.

A useful quantity in stochastic dynamics is the autocorrelation function, $R_U(\tau)$, which is the correlation a stochastic process have with itself at another time.

$$R_U(\tau) = E[U(t)U(t+\tau)] \quad (20)$$

The autocorrelation function is closely related to the autocovariance function $C_U(\tau)$, shown in equation (21).

$$\begin{aligned} C_U(\tau) &= E[(U(t) - \mu_U)(U(t+\tau) - \mu_U)] \\ &= R_U(\tau) - \mu_U \end{aligned} \quad (21)$$

A stochastic process is called weakly stationary if the mean is constant in time, and if the two functions in (20) and (21) depend on τ alone. The Wiener-Khintchine relations now states that the autocovariance function forms a Fourier transform pair with the power spectral density (PSD) function, or simply the variance spectrum, of the process. The variance spectrum of a stochastic process U is denoted by $S_U(\omega)$ and is given as:

$$S_U(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} C_U(\tau) e^{-i\omega\tau} d\tau \quad (22)$$

while $C_U(\tau)$ could be obtained as the inverse transform:

$$C_U(\tau) = \int_{-\infty}^{\infty} S_U(\omega) e^{i\omega\tau} d\omega. \quad (23)$$

Equation (23) evaluated at $\tau = 0$ illustrates that the variance spectrum could be thought of as a variance density distribution along frequencies:

$$C_U(0) = E [(U(t) - \mu_U)^2] = \sigma_U^2 = \int_{-\infty}^{\infty} S_U(\omega) d\omega. \quad (24)$$

It could be shown that $S_U(\omega)$ is symmetric about $\omega = 0$. This information could be used to avoid integrating over negative frequencies by defining the one-sided variance spectrum as

$$S_U^+(\omega) = \begin{cases} 2 S_U(\omega) & \text{for } \omega \geq 0 \\ 0 & \text{for } \omega < 0. \end{cases} \quad (25)$$

As derived by Næss and Moan [6], the variance spectrum of the load process and the response process are connected as shown in equation (26).

$$\begin{aligned} S_U(\omega) &= H(\omega)^* H(\omega) S_F(\omega) \\ &= |H(\omega)|^2 S_F(\omega) \end{aligned} \quad (26)$$

Combined with equation (24), this results in the following expression for the variance of the stochastic process $U(t)$:

$$\sigma_U^2 = \int_0^{\infty} |H(\omega)|^2 S_F^+(\omega) d\omega. \quad (27)$$

Variances of time differentiated processes could simply be found by differentiating the spectrum from equation (22), obtaining that $S_{\dot{U}}(\omega) = |i\omega|^2 S_U(\omega) = \omega^2 S_U(\omega)$, and therefore

$$\sigma_{\dot{U}}^2 = \int_0^{\infty} \omega^2 |H(\omega)|^2 S_F^+(\omega) d\omega. \quad (28)$$

Similarly, one would obtain that

$$\sigma_{\ddot{U}}^2 = \int_0^{\infty} \omega^4 |H(\omega)|^2 S_F^+(\omega) d\omega. \quad (29)$$

For a SDOF system with the frequency response function derived in 2.1.1, $|H(\omega)|^2$ is found to be:

$$\begin{aligned}
|H(\omega)|^2 &= H(\omega)^* H(\omega) \\
&= \text{Re}[H(\omega)]^2 - i^2 \text{Im}[H(\omega)]^2 \\
&= \text{Re}[H(\omega)]^2 + \text{Im}[H(\omega)]^2 \\
&= \left(\frac{m\omega_0^2 - m\omega^2}{(m\omega_0^2 - m\omega^2)^2 + (2m\zeta\omega_0\omega)^2} \right)^2 + \left(\frac{-2m\zeta\omega_0\omega}{(m\omega_0^2 - m\omega^2)^2 + (2m\zeta\omega_0\omega)^2} \right)^2 \\
&= \frac{1}{m^2 [(\omega_0^2 - \omega^2)^2 + (2\zeta\omega_0\omega)^2]}
\end{aligned} \tag{30}$$

These principles are in the following section used to illustrate the response of a SDOF system subjected to a Gaussian loading process.

2.1.3 Example of a SDOF System Response to Stationary White Noise Gaussian Loading

A SDOF system like the one illustrated in figure 1 could for example have values of $m = 1$ kg, $k = 10$ N/m and $\zeta = 5$ %. This would result in the impulse response function and the squared frequency response functions plotted in figure 2.

In figure 2, one could easily identify the natural frequency of the system by the peak in the plot for $|H(\omega)|^2$ at $\omega_0 = \sqrt{k/m} \approx 3.16$ rad/s ≈ 0.5 Hz. This peak would be higher and more narrow if the structure was less damped, increasing the resonance effect. Similarly, if the structure was more damped, the frequency response peak would be wider and have a smaller amplitude.

Since this is a one degree of freedom system, there is only one natural frequency.

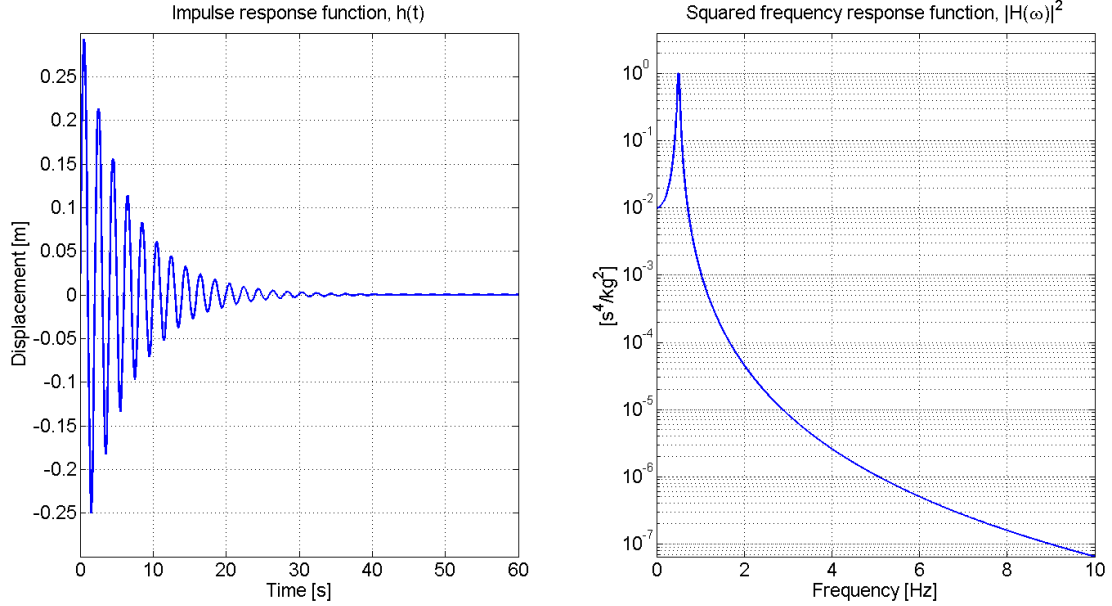


Figure 2: Plots of $h(t)$ and $|H(\omega)|^2$ for a SDOF system where $m = 1$ kg, $k = 10$ N/m and $\zeta = 5$ %.

Now we could assume that this system is excited by a load F that is Gaussian distributed with a mean value of 30 N, and a standard deviation of 5 N, at each time step $\Delta t = 0.05$ s. The Gaussian distribution have the following probability density function for the random variable X :

$$f_X(x) = P[x < X \leq x + dx] = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left[-\frac{1}{2}\left(\frac{x - \mu_X}{\sigma_X}\right)^2\right]. \quad (31)$$

By picking a force from the Gaussian distribution at each time step we effectively assign the same energy content for all frequencies up to the Nyquist frequency, which is the highest detectable frequency for the chosen sampling interval. The Nyquist frequency f_n , is given by the sampling frequency f_s , as $f_n = f_s/2 = 1/(0.05*2)$ Hz = 10 Hz. In this case the load variance spectrum would be a constant, S_0 . Such spectra is called white noise spectra, since they contain the same amount of energy for all frequencies. There is obviously no feasible way of obtaining all frequencies, so an ideal white noise spectrum, which would have contained infinite variance, cannot be simulated.

Even though white noise loading is used in this simple example, the energy content of environmental loads is generally a function of frequency. This will become apparent in later theory sections. The one-sided load variance spectrum and the resulting one-sided response spectrum, calculated by equation (26), is shown in figure 3. Since the area below S_0 up to the Nyquist frequency should equal the variance of the stochastic load process, S_0 is found as $S_0 = \sigma_F^2/f_n = 25\text{N}^2/10\text{ Hz} = 2.5\text{ N}^2/\text{Hz}$.

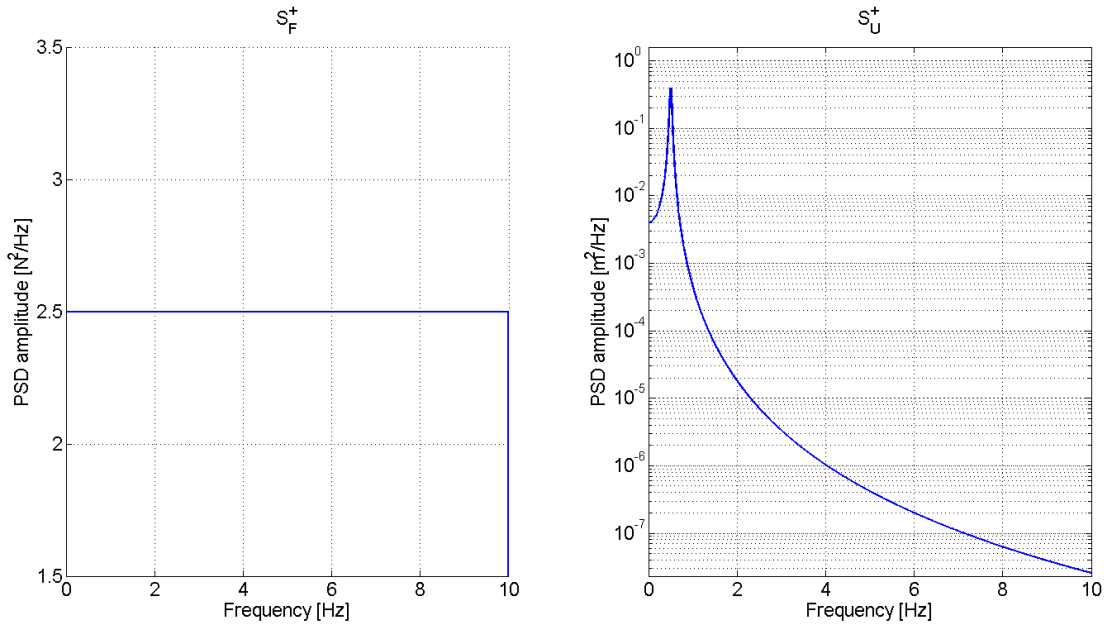


Figure 3: Plots of the example one-sided load spectrum and response variance spectrum.

Using the Duhamel integrals shown in equations (5) and (14), the response realizations could be calculated in the time domain. $u(t)$ and $\dot{u}(t)$, are shown for one such 220 second long simulation in figure 4. Since we are interested in the stationary long-term response of the system, the initial 20 seconds of these time series are removed to exclude initial transient effects.

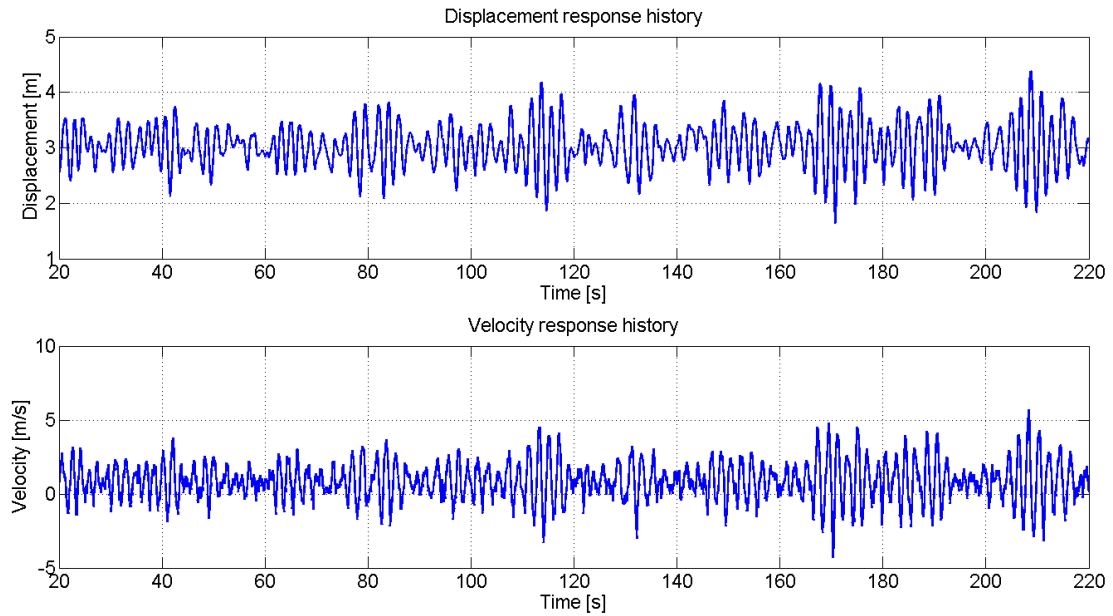


Figure 4: Time series of the displacement and velocity response of the system.

It is derived, e.g. by Lutes and Sarkani in [5], that any linear system excited by a Gaussian load process F , will have a Gaussian response process $U(t)$. The derivative of the response, $\dot{U}(t)$, will also be Gaussian distributed since it is a limit of a linear combination of the Gaussian processes $U(t)$ and $U(t + \Delta t)$. The mean values and variances of the two processes $U(t)$ and $\dot{U}(t)$ are determined by the equations provided in section 2.1.2. Using Matlab to calculate these values numerically, I found that:

$$\mu_U = 3.000 \text{ m}$$

$$\mu_{\dot{U}} = 0.7520 \frac{\text{m}}{\text{s}}$$

$$\sigma_U^2 = 0.1976 \text{ m}^2$$

$$\sigma_{\dot{U}}^2 = 1.9701 \frac{\text{m}^2}{\text{s}^2}$$

The mean value of the response equals the static solution $u = f/k = 3 \text{ m}$ corresponding to the mean load, as could be expected for this linear system. Normalized histograms of the realizations $f(t)$, $u(t)$ and $\dot{u}(t)$ during a 1500 second long simulation are plotted with the respective underlying Gaussian distributions for the stochastic processes $F(t)$, $U(t)$ and $\dot{U}(t)$ in figure 5.

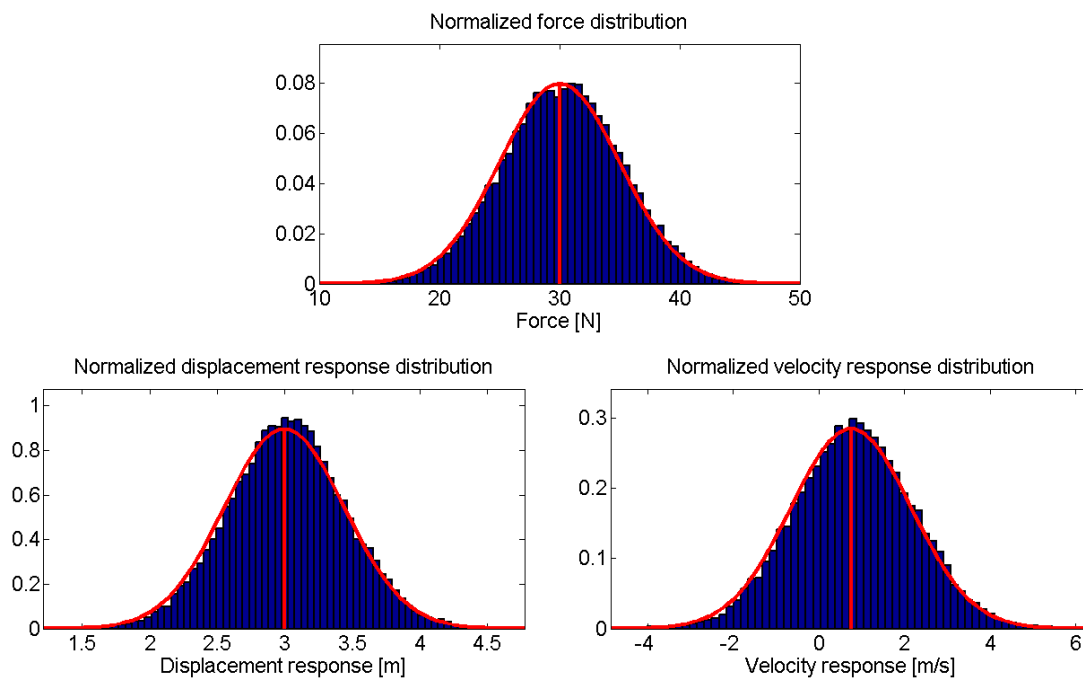


Figure 5: Normalized histograms showing realizations of $f(t)$, $u(t)$ and $\dot{u}(t)$ during a 1500 second simulation. The red plots display the analytical Gaussian distributions for the random variables $F(t)$, $U(t)$ and $\dot{U}(t)$, with their respective mean values.

One should be aware that if $F(t)$ is supposed to model a continuous force on the structure, which is often the case, then the result $\mu_{\dot{U}} = 0.7520$ m/s is somewhat erroneous. This is an effect of the discrete nature of the force process. By equation (19) it is apparent that $\mu_{\dot{U}} = 0$ m/s for a stationary and time differentiable process. Using my script to calculate $\mu_{\dot{U}}$ and run the simulation for smaller time steps it could indeed be found that $\mu_{\dot{U}} \rightarrow 0$ m/s as $\Delta t \rightarrow 0$ s.

If $U(t)$ is a stationary and differentiable Gaussian process, then $\dot{U}(t)$ is jointly Gaussian distributed with $U(t)$ [6], that is: [9]

$$\begin{aligned} f_{U\dot{U}}(u, \dot{u}) &= P[u < U \leq u + du \cap \dot{u} < \dot{U} \leq \dot{u} + d\dot{u}] \\ &= \frac{1}{\sqrt{2\pi}\sigma_U\sigma_{\dot{U}}\sqrt{1-\rho_{U\dot{U}}^2}} e^{\vartheta} \end{aligned} \quad (32)$$

where

$$\vartheta = -\frac{1}{2(1-\rho_{U\dot{U}}^2)} \left[\left(\frac{u - \mu_U}{\sigma_U} \right)^2 - 2\rho_{U\dot{U}} \left(\frac{u - \mu_U}{\sigma_U} \right) \left(\frac{\dot{u} - \mu_{\dot{U}}}{\sigma_{\dot{U}}} \right) + \left(\frac{\dot{u} - \mu_{\dot{U}}}{\sigma_{\dot{U}}} \right)^2 \right].$$

In equation (32), $\rho_{U\dot{U}}$ is the correlation coefficient between $U(t)$ and $\dot{U}(t)$, measuring their linear dependence as a value between -1 and 1.

$$\rho_{U\dot{U}} = \frac{Cov[U, \dot{U}]}{\sqrt{Var[U] \cdot Var[\dot{U}]}} = \frac{C_{U\dot{U}}(\tau = 0)}{\sigma_U \cdot \sigma_{\dot{U}}} \quad (33)$$

According to equation (23), $C_{U\dot{U}}(\tau)$ could be found as:

$$\begin{aligned} C_{U\dot{U}}(\tau) &= \int_{-\infty}^{\infty} S_{U\dot{U}}(\omega) e^{i\omega\tau} d\omega \\ &= \int_{-\infty}^{\infty} i\omega S_U(\omega) e^{i\omega\tau} d\omega. \end{aligned} \quad (34)$$

Evaluated at $\tau = 0$, the integral above results in $C_{U\dot{U}} = 0$ since $\omega S_U(\omega)$ is an odd function of ω . Using the result that U and \dot{U} are uncorrelated processes, their joint probability distribution function from equation (32) simplifies to:

$$f_{U\dot{U}}(u, \dot{u}) = f_U(u) \cdot f_{\dot{U}}(\dot{u}) = \frac{1}{\sqrt{2\pi}\sigma_U\sigma_{\dot{U}}} \exp \left[-\frac{1}{2} \left(\left(\frac{u - \mu_U}{\sigma_U} \right)^2 + \left(\frac{\dot{u} - \mu_{\dot{U}}}{\sigma_{\dot{U}}} \right)^2 \right) \right]. \quad (35)$$

This joint probability density function of $U(t)$ and $\dot{U}(t)$ is plotted in figure 6 for the previously calculated mean values and standard deviations. A normalized joint histogram

over the realization pairs of $u(t)$ and $\dot{u}(t)$ for the 1500 second long simulation is shown in figure 7 for comparison.

Because of the Gaussian loading process and the linear dynamic system, we are in this neat case able to find the *complete* probability information of the response through exclusive use of the first two moments of the response process, the expected value and the variance. The methods used to determine these two moments of the response are however, as shown in 2.1.2, not in any way limited to the description of Gaussian processes.

The Matlab script used to perform the calculations in this section could be found in attachment 1.

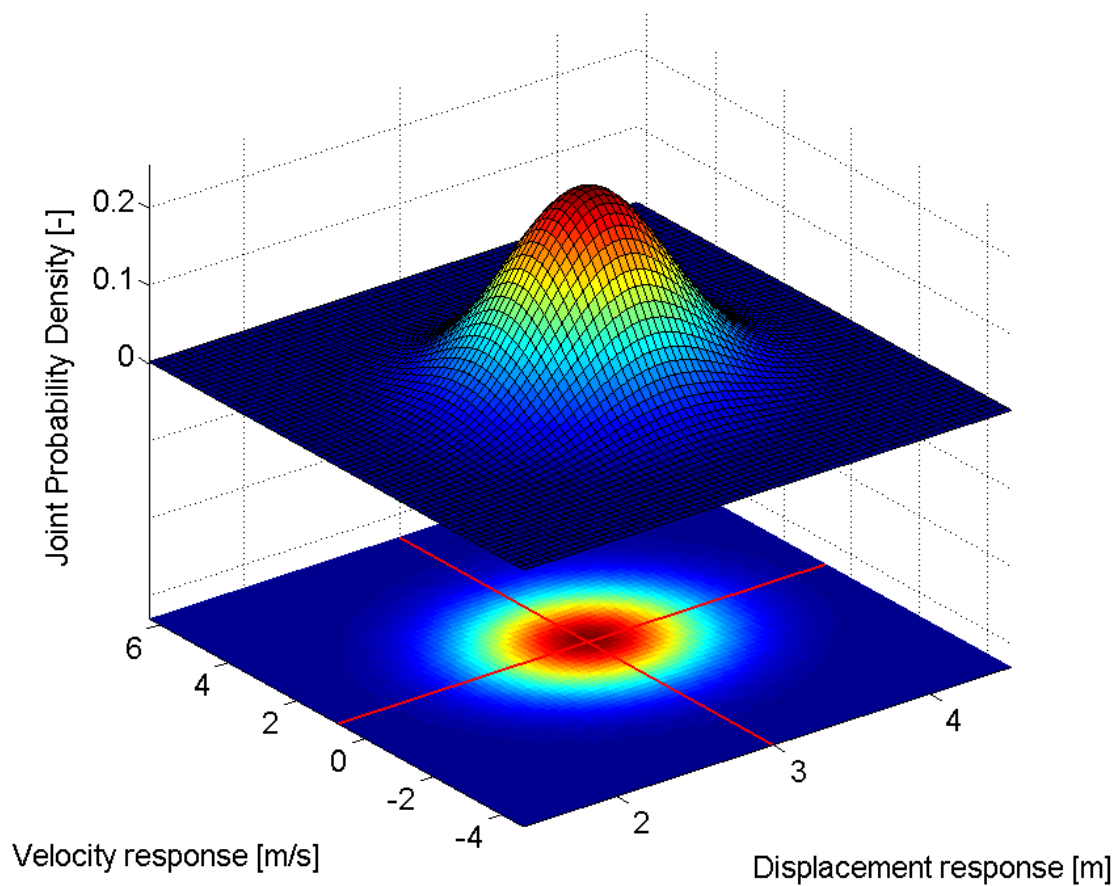


Figure 6: The analytical joint probability density function of $U(t)$ and $\dot{U}(t)$. The red lines mark the mean values μ_U and $\mu_{\dot{U}}$.

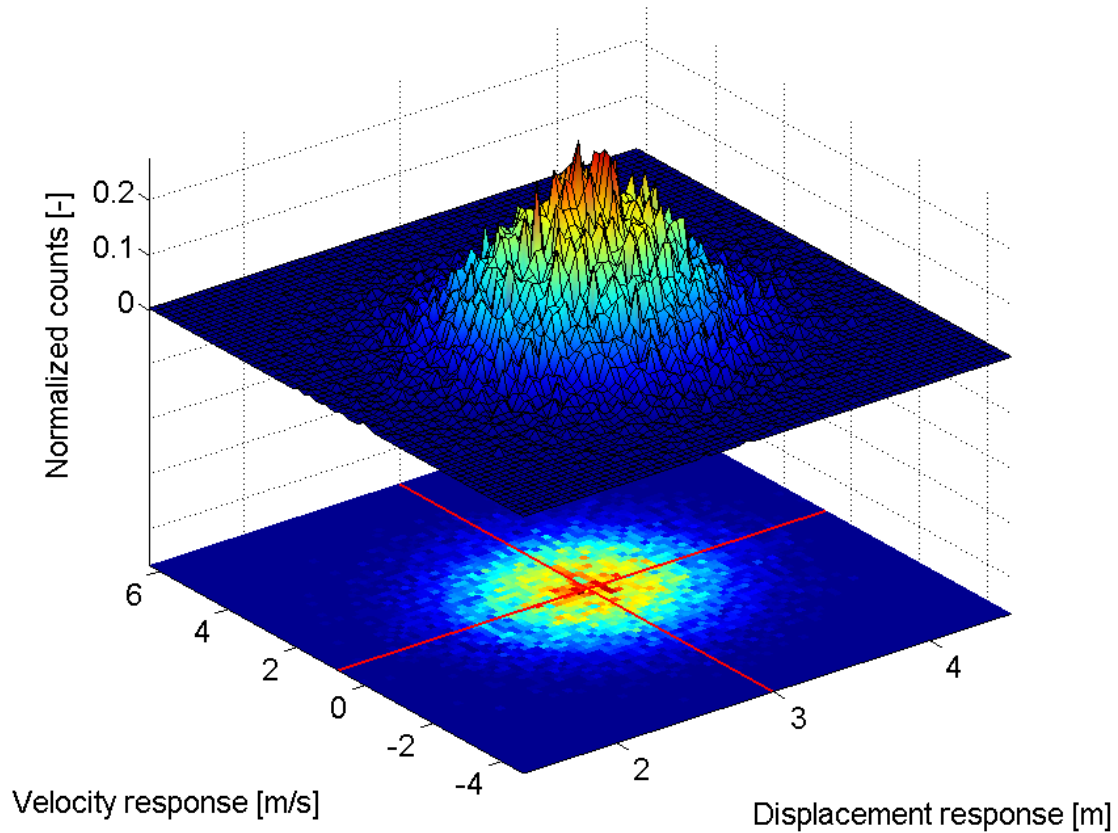


Figure 7: Normalized joint histogram over the realization pairs of $u(t)$ and $\dot{u}(t)$ for a 1500 second long Monte Carlo simulation with a time step of $\Delta t = 0.05$ s, resulting in 30000 sample points. The red lines mark the mean values μ_U and $\mu_{\dot{U}}$.

2.2 Probability Density Evolution

While Monte Carlo Simulation (MCS) of the system provides a straightforward method of estimating the relevant probabilities of the response, this method is in general computationally expensive. This is especially true if an accurate estimate of the probability content in the distribution tails are important, which it often is for structures where fatigue is thought to be a considerable design driver.

One alternative method of obtaining the probability density of the response is to evolve the probability through the use of a partial differential equation describing the both the drift of the system due to deterministic behavior, and the diffusion of the system due to the randomness involved. This enables an instantaneous evolution estimation of every probability in the state space, given a description of the initial state. The probability content of the response could now be found accurately through an iteration process that has proved to be effective compared to MCS of the same system. [3]

Probability evolution methods are subject to substantial ongoing research, especially re-

garding their extension to non-linear multi degree of freedom systems, and to situations with multiple sources of randomness. [10] The analyses in this thesis will solely focus on randomness in the excitation of a system, while the structural system itself will be modeled as deterministic and linear.

The following theory on probability density evolution is largely based on Hsu [1], Sun and Luo [11], Li and Chen [10], and Wijker [12].

2.2.1 Markov Processes and the Fokker-Planck Equation

A Markov process could be described as a process where the next state exclusively depends on the current state. This implies that one could predict the next state of the process just as accurately knowing the current state as one could knowing the entire history of the process. This property could be expressed mathematically in terms of the conditional probability f_c of a Markov process X :

$$\begin{aligned} & f_c(x_n, t_n | x_{n-1}, t_{n-1}) \\ &= P[(x_n < X \leq x_n + dx, t_n) | (x_{n-1} < X \leq x_{n-1} + dx, t_{n-1})] \quad (36) \\ &= f_c(x_n, t_n | x_1, t_1, x_2, t_2, \dots, x_{n-1}, t_{n-1}). \end{aligned}$$

Given such a N-dimensional process \mathbf{X} , it is derived e.g. in [12], that the stochastic differential equation governing the time evolution of the probability density in state space is:

$$\frac{\partial}{\partial t} f_c(\mathbf{x}, t | \mathbf{x}_0, t_0) = - \sum_{i=1}^N \frac{\partial}{\partial x_i} [A_i(\mathbf{x}) f_c] + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2}{\partial x_i \partial x_j} [B_{ij}(\mathbf{x}) f_c] \quad (37)$$

when neglecting higher-order migration terms. In this equation, \mathbf{x}_0 contains the initial state of the system at time t_0 , $A_i(\mathbf{x})$ is the drift coefficient and $B_{ij}(\mathbf{x})$ is the diffusion coefficient of the system. Equation (37) is known as the Fokker-Planck equation. It is in general subjected to the initial condition shown in equation (38).

$$f_c(\mathbf{x}, t | \mathbf{x}_0, t_0) = \delta(\mathbf{x} - \mathbf{x}_0) \quad (38)$$

If the conditional probability density function in equation (37) is independent of time, then $f_c(\mathbf{x}, t | \mathbf{x}_0, t_0) = f(\mathbf{x})$ and the Fokker-Planck equation simplifies to its steady-state form:

$$- \sum_{i=1}^N \frac{\partial}{\partial x_i} [A_i(\mathbf{x}) f(\mathbf{x})] + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2}{\partial x_i \partial x_j} [B_{ij}(\mathbf{x}) f(\mathbf{x})] = 0. \quad (39)$$

2.2.2 Numerical Solution of the Fokker-Planck Equation using the Cell-to-Cell Mapping Method

One way of solving the Fokker-Planck equation numerically is to evaluate its path integral, which is given by: [11]

$$f(\mathbf{x}_{i+1}, t_{i+1}) = \int_{R^n} f(\mathbf{x}_{i+1}, t_{i+1} | \mathbf{x}_i, t_i) f(\mathbf{x}_i, t_i) d\mathbf{x}_i \quad (40)$$

where R^n denotes the state space. Time could now be discretized setting $\Delta t = t_{i+1} - t_i$ for all $i \geq 0$, and $t_i = i\Delta t$. Equation (40) now becomes

$$f(\mathbf{x}_{i+1}, (i+1)\Delta t) = \int_{R^n} f(\mathbf{x}_{i+1}, (i+1)\Delta t | \mathbf{x}_i, i\Delta t) f(\mathbf{x}_i, i\Delta t) d\mathbf{x}_i. \quad (41)$$

The state space R^n is also discretized into m cells, $\Delta c_1, \Delta c_2, \dots, \Delta c_m$ with center points c_1, c_2, \dots, c_m , in order to numerically solve the integral. Equation (41) could now be written as a discrete Markov chain:

$$\mathbf{p}(k+1) = \mathbf{P}(k)\mathbf{p}(k) \quad (42)$$

where $\mathbf{p}(k) = [p_i(k)]$ is the probability of finding the system in cell i at time $k\Delta t$, and $\mathbf{P}(k) = [P_{ij}(k)]$ is the probability of finding the system in cell i at time $(k+1)\Delta t$ when the system is in cell j at time $k\Delta t$ with a probability of one.

$$\begin{aligned} p_i(k) &= \int_{\Delta c_i} f(\mathbf{x}, k\Delta t) d\mathbf{x} \\ &\approx \Delta c_i f(c_i, k\Delta t) \end{aligned} \quad (43)$$

$$\begin{aligned} P_{ij}(k) &= \int_{\Delta c_i} f(\mathbf{x}, (k+1)\Delta t | c_j, k\Delta t) d\mathbf{x} \\ &\approx \Delta c_i f(\Delta c_i, (k+1)\Delta t | c_j, k\Delta t) \end{aligned} \quad (44)$$

In equation (43) and (44) the approximation of constant probability within cells requires sufficiently small cell volumes, Δc . The transition matrix \mathbf{P} could now be estimated using a transition time interval that is small enough. Iterations of the Markov chain from equation (42) would now converge towards the joint probability density of \mathbf{x} as $k \rightarrow \infty$. This is in principle the cell-to-cell mapping method derived by Hsu. [1]

Figure 8 shows the joint probability density of $U(t)$ and $\dot{U}(t)$ from section 2.1.3, calculated using the probability density evolution Matlab script in attachment 2. This script is elaborated in section 3.6 along with a general discussion of the cell-mapping method.

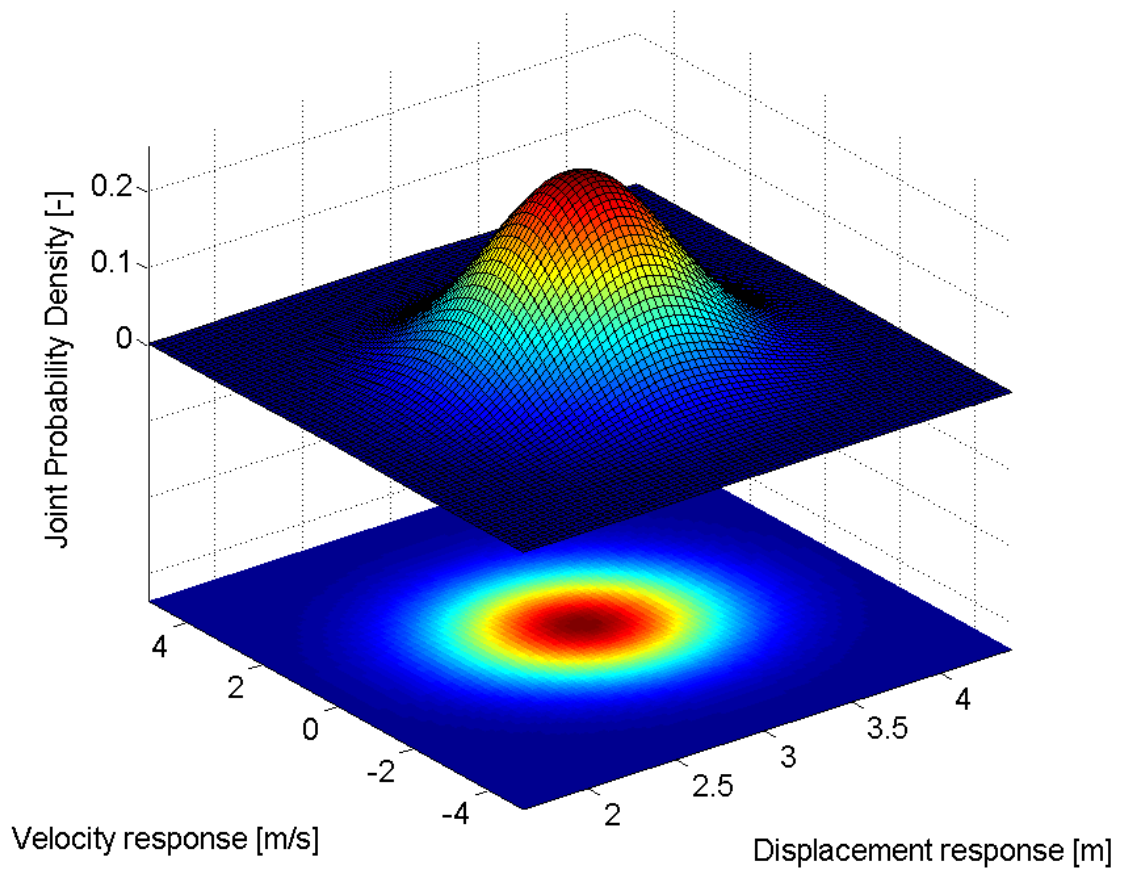


Figure 8: Joint probability density of $U(t)$ and $\dot{U}(t)$ from section 2.1.3, calculated using probability density evolution.

2.3 Environmental Loading

Offshore wind turbines are subjected to highly dynamic, irregular external loads as they are excited by wind and waves. Such environmental loading processes shows considerable correlation. The following sections discusses how these could be simulated through the concept of variance spectra introduced in 2.1.2.

2.3.1 Wind

The wind velocity field acting on a structure in the main wind direction is commonly described as the sum of a mean wind velocity as a function of height z , and a fluctuating wind velocity dependent on both height and time. This is illustrated in figure 9.

$$V_{tot}(z, t) = \bar{V}(z) + v(z, t) \quad (45)$$

The mean wind velocity could be described by the Normal Wind Profile (NWP), defined in equation (46). [13]

$$\bar{V}(z) = \bar{V}(z_{hub}) \cdot \left(\frac{z}{z_{hub}} \right)^\alpha \quad (46)$$

The exponent α is determined by the surface roughness. Germanischer Lloyd recommends using $\alpha=0.14$ for sites located offshore. This is a low value compared to onshore sites, where α typically lies around 0.2 due to higher surface roughness. [13]

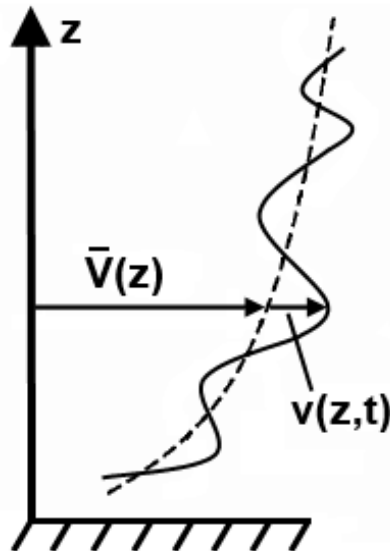


Figure 9: Superposition of time dependent and mean wind velocities. [14]

The fluctuating contribution, named turbulence, is best described stochastically as a process with a mean value of 0 m/s and a standard deviation of σ_v m/s. The standard deviation of the turbulence is defined in equation (47) as the coefficient of variation I of the wind speed at hub height multiplied by v_{hub} , which is the short term mean value of the hub height wind speed. v_{hub} is typically measured over a period of 10 minutes. I is commonly called the turbulence intensity.

$$\sigma_v = I \cdot v_{hub} \quad (47)$$

This standard deviation could also be described through the Normal Turbulence Model (NTM) as defined by the International Electrotechnical Commission (IEC) in equation (48). [13]

$$\sigma_v = I_{ref} \cdot \left(0.75 \cdot v_{hub} + 5.6 \frac{\text{m}}{\text{s}} \right) \quad (48)$$

Here, I_{ref} is a turbulence intensity specified by IEC for three different turbulence classes. The velocity variance spectra of the turbulent wind field as a function of frequency, $S_v(f)$, is now defined by:

$$\int_0^{\infty} S_v(f) \, df = \sigma_v^2 \quad (49)$$

One model for these spectra is called the Kaimal model, defined by IEC 61400 3rd edition [15] as

$$S_v(f) = \frac{4\sigma^2 (L_k/\bar{v}_{hub})}{(1 + 6fL_k/\bar{v}_{hub})^{5/3}} \quad (50)$$

In equation (50), L_k is an integral scale parameter given in IEC 61400 3rd edition [15] for the main wind direction as

$$\begin{aligned} L_k &= 8.10 \cdot \Lambda_U \\ &= 8.10 \cdot 0.7 \cdot \min\{60 \text{ m}, z_{hub}\} \end{aligned} \quad (51)$$

For the model used in this project $z_{hub}=90.55$ m yields $L_k=340.2$ m. If one assumes $v_{hub}=16$ m/s and turbulence class B, then $I_{ref}=0.14$. Equation (48) now results in $\sigma_v=2.464$ m/s. The Kaimal spectrum for this case is plotted in figure 10.

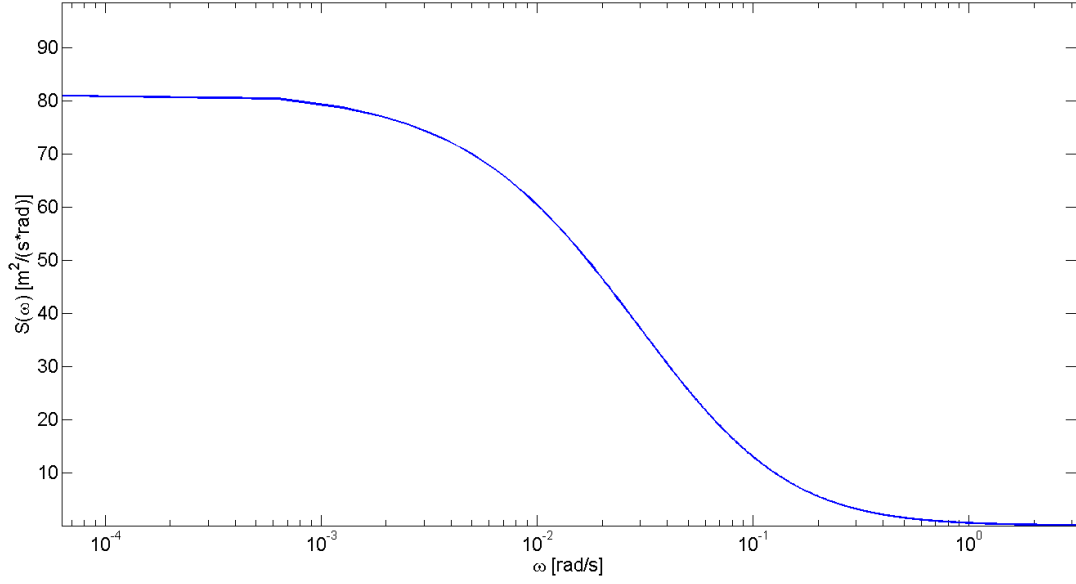


Figure 10: Logarithmic x-axis plot of the Kaimal spectrum for $L_k=340.2$ m, $v_{hub}=16$ m/s and turbulence class B. It could be shown that the square root of the area under this curve equals $\sigma_v = 2.464$ m/s, as defined by equation (49).

To generate a wind velocity field in the time domain, $v(z, t)$, one would have to perform a simulation based on spectral decomposition, as described by e.g. Strømmen. [14] The Fourier series of the wind velocity field at a single point in space could be written as

$$\begin{aligned}
 v(t) &\approx \sum_{k=0}^N x_k \\
 &= Re \left(\sum_{k=0}^N c_k \cdot e^{i(\omega_k t + \psi_k)} \right) \\
 &= \sum_{k=0}^N c_k \cdot \cos(\omega_k t + \psi_k)
 \end{aligned} \tag{52}$$

where

$$c_k = \sqrt{2\Delta\omega_k \cdot S(\omega_k)} \tag{53}$$

and

$$\omega_k = 2\pi f_k. \tag{54}$$

ψ_k is a random phase angle for each k described by the uniform probability distribution between $\psi_k=0$ and $\psi_k=2\pi$. When transforming from the time domain to the frequency domain one lose information about the placement of the velocities along the time axis.

Simulation is therefore the only way of going back. The simulated turbulence would however have the same variance, mean value and correlation as the original fluctuating part of the wind field, and these stochastic properties are in principle what is needed to simulate response. One such simulation based on the Kaimal spectrum from figure 10 is shown as an example in figure 11.

The Matlab script used to generate spectral decomposition simulations of turbulence in this thesis, could be found in attachment 3. For the finite element time domain simulations, stochastic wind is generated using software described in section 3.2.

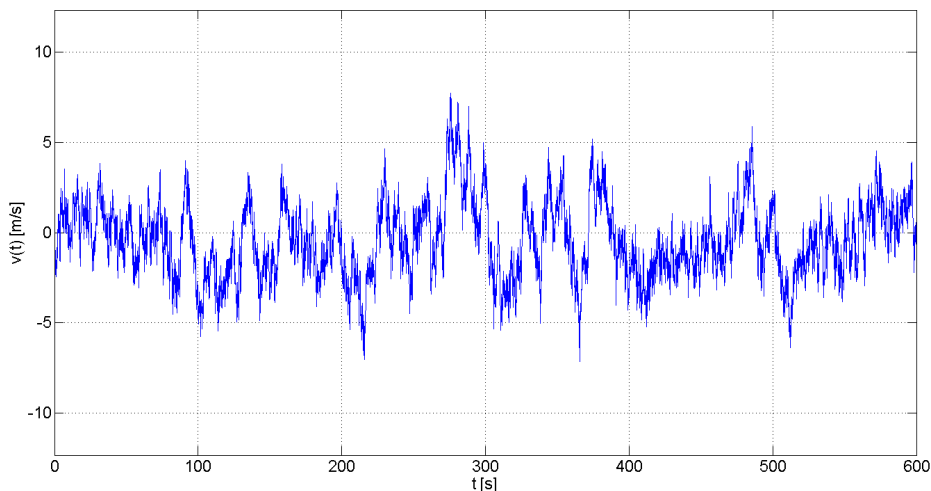


Figure 11: Matlab plot of simulated wind velocity fluctuations over 10 minutes in the main wind direction at hub height, given the specified Kaimal spectrum from figure 10.

Turbulence also leads to fluctuating wind velocity components along the two axes perpendicular to the main wind direction. These components have their own turbulence intensity and are calculated in the same manner as described above.

When a complete description of the wind velocity field have been established, the resulting aerodynamic forces that act on the structure could be calculated using the dimensionless aerodynamic drag coefficient, C_D . Neglecting the relative velocity of the structure, the incremental force from aerodynamic drag would be given as:

$$dF(z, t) = \frac{1}{2} \rho C_D dA [\bar{V}(z) + v(z, t)]^2. \quad (55)$$

In equation (55), dA is the cross sectional area of the section and ρ is the mass density of the air.

2.3.2 The Blade Element Momentum Method

The true airflow near the rotor is immensely complicated due to the relative movement of the blades which also affects the turbulent flow directions. The Blade Element Momentum (BEM) method is considered one of the best simplified approaches to estimate the resulting loads acting on the rotor.

This model is based on the idea that the area swept by the rotor could be divided into infinitesimal thick annular ring elements, and that angular momentum must be conserved between the rotor and the fluid within each ring. One assumes that there is no fluid interaction between the annular rings, and that the wind velocity is constant over each element. Provided the configuration and aerodynamic coefficients of the blades, airfoil theory could now be used on these individual blade elements to obtain local lift and drag forces.

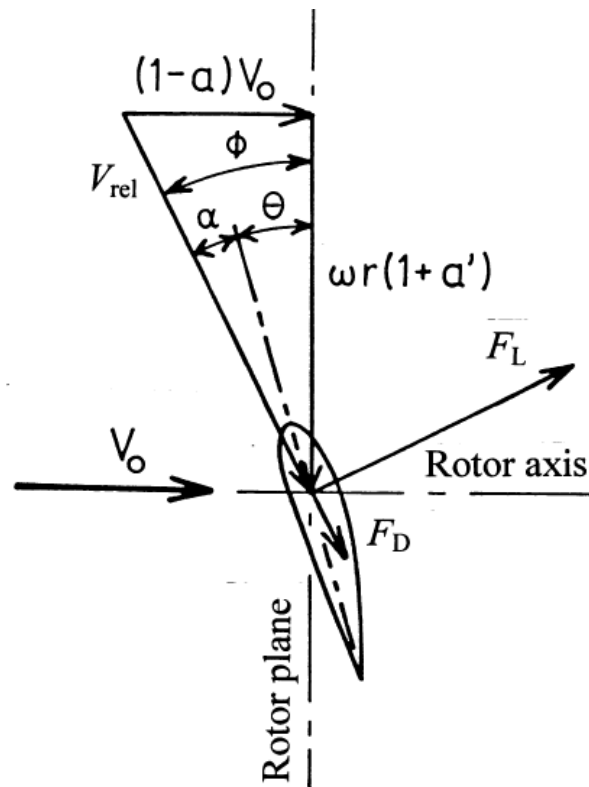


Figure 12: Velocity components around rotor blade airfoil. [16]

Figure 12 illustrates some of the terminology used in the following brief summary of the concept. A complete description of the BEM method and its background theory is for example given by Det Norske Veritas in [16], or by Manwell et.al in [17].

From momentum theory it is known that the contribution to total thrust and torque from a blade section could be found as

$$dT = \rho V_0^2 4a(1 - a) \pi r dr \quad (56)$$

and

$$dQ = \rho V_0 4a'(1 - a) \pi r^3 \omega dr, \quad (57)$$

respectively. In equation (56) and (57) ρ is the mass density of the air, V_0 is the incoming wind velocity, r is the radius from the hub to the blade section centroid and dr is the section length. a and a' is the axial and angular induction factors, given as:

$$a = 1 - \frac{V_1}{V_0} \quad (58)$$

and

$$a' = \frac{V_w}{2\omega r}. \quad (59)$$

V_1 is the axial wind velocity after the rotor, V_w is the tangential wind velocity of the wake after the rotor and ω is the angular velocity of the rotor. The contribution to total thrust and torque from a blade section could also be derived from blade element theory, as shown in equation (60) and (61).

$$dT = \frac{1}{2} B \rho V_{rel}^2 (C_l \cos\phi + C_d \sin\phi) c dr \quad (60)$$

$$dQ = \frac{1}{2} B \rho V_{rel}^2 (C_l \sin\phi - C_d \cos\phi) c dr \quad (61)$$

In these equations, ϕ is the angle of the relative wind velocity, $V_{rel} = V_0(1 - a)/\sin\phi$. The angle ϕ is the sum of the local pitch angle θ of the blade and the local inflow angle α . B is simply the number of blades considered, and c is the chord length of the blade section.

In general, the induction factors a and a' is unknown, and one has to iterate to find these for every blade section at every time step. This could be done by guessing values for a and a' , updating the relative angle and then updating the lift and drag coefficients C_l and C_d , which are functions of the local angle of attack. The induction factors could now be updated by using relationships given by equation (56), (57), (60) and (61).

After obtaining a convergent pair of induction factors for a given blade section, the relevant contribution to the total thrust or torque could be determined. These contributions are projected into the rotor plane and its normal direction and summed up to represent the total rotor torque and thrust at each time step.

Figure 13 shows a 60 second estimation of the total thrust resultant obtained using

the BEM method Matlab script in attachment 4. The wind load for this case is given by a Kaimal simulation with reference wind speed 10 m/s and a standard deviation of 1.834 m/s. The rotor considered is the NREL 5-MW Baseline Turbine described in section 2.3. The mass density of the air is set to 1.25 kg/m³.

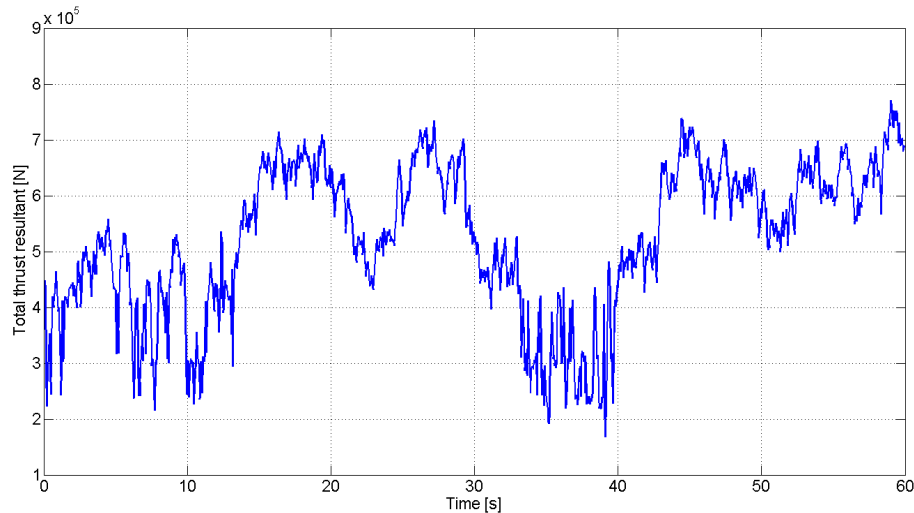


Figure 13: Estimated total rotor thrust resultant for the NREL 5-MW Baseline Turbine given a 10 m/s reference wind speed with turbulence class B.

It should be stressed that the script in attachment 4 is solely intended to serve as a simplified illustration of the BEM method. Highly relevant considerations such as the wind turbine control system, the flexible motion of the blades and fluctuations in the angular velocity of the rotor are all not included in this script. For the subsequent time domain analyses of this thesis, the certified code described in section 3.3 will be used to obtain simulated rotor thrust values.

2.3.3 Waves

Even though wave forces are not included in the simplified load model presented in this thesis, they would be important when considering more sophisticated models. Because of this, the inclusion of wave loads is also important when discussing the future usefulness of probability evolution for problems of this kind.

In order to mimic realistic sea waves one has to take into account that real sea waves are a superposition of many waves with different properties regarding amplitude, length and period. This is done by describing some statistical processes assumed stationary over the short-term simulations. Such processes are called irregular sea states, and their properties are modeled using the spectral density function of the surface elevation. [18] This method of describing a stationary stochastic process is basically analog to the description of wind turbulence in 2.3.1.

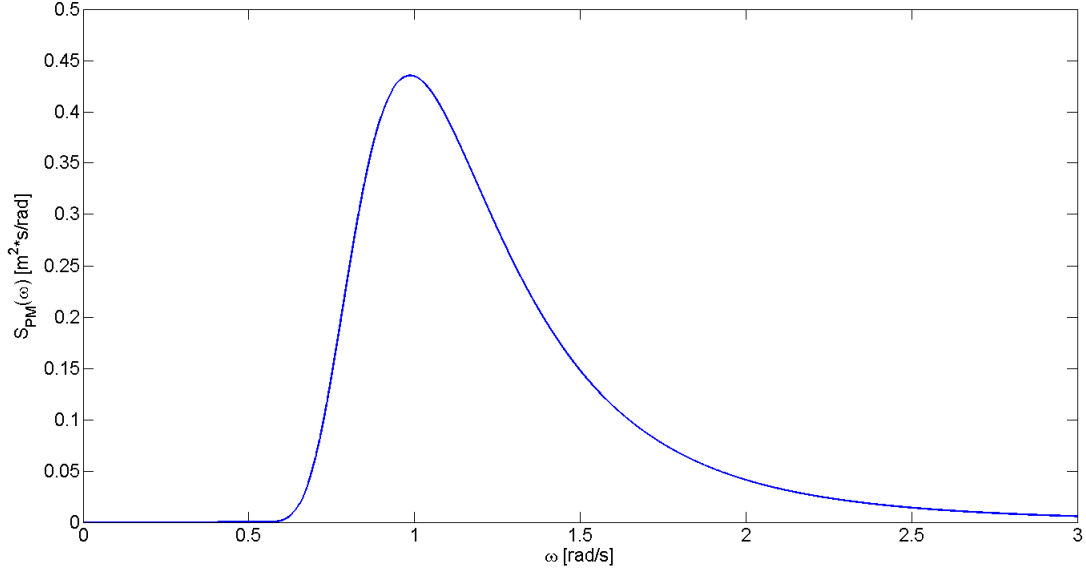


Figure 14: Example of a Pierson-Moskowitz spectrum with significant wave height of 2.19 meters and a spectral peak period of 6.37 seconds.

One potential wave energy spectrum is the Pierson-Moskowitz spectrum (PM-spectrum), defined in equation (62).

$$S_{PM}(\omega) = \frac{5}{16} H_s^2 \omega_p^2 \omega^{-5} e^{-\frac{5}{4} \left(\frac{\omega}{\omega_p}\right)^{-4}} \quad (62)$$

In equation (62), H_s and ω_p describes the properties of the specific sea state. H_s is the significant wave height, defined as the mean value of the highest 1/3 of the observed wave heights, measured from trough to crest, in a sample. [18] ω_p is the angular spectral peak frequency, which could be defined by the spectral peak period, T_p , as $\omega_p = 2\pi/T_p$. Figure 14 shows the PM-Spectrum for $H_s=2.19$ meters and $T_p=6.37$ seconds.

The established sea state spectrum could now be discretized and the contributions from each frequency in every time step could be superimposed to give the surface elevation, $\eta(t)$, as in equation (63). As with the turbulence simulation, a uniformly distributed random phase shift angle between 0 and 2π is introduced for each frequency.

$$\chi(t) = \sum_{k=0}^N \sqrt{2\Delta\omega_k \cdot S(\omega_k)} \cdot \cos(\omega_k t + \psi_k) \quad (63)$$

One such time domain simulation of the Pierson-Moskowitz spectrum from figure 14 is shown in figure 15.

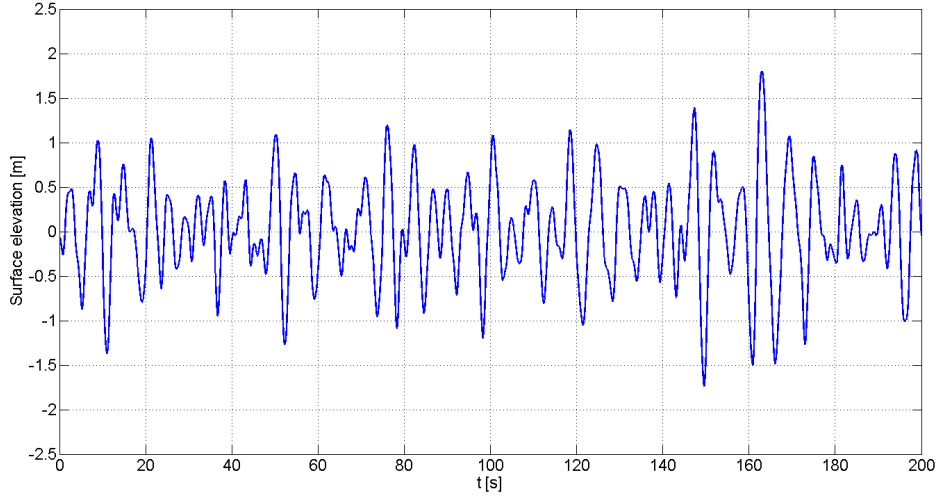


Figure 15: An example of a 200 seconds long time domain simulation of the surface elevation, given the sea state described by the PM-spectrum in figure 14.

After having described the stochastic nature of the surface elevation χ , the kinematics of the water particles could be estimated as a function of depth using the Wheeler stretching method and wave theory. This is explained in detail by e.g. DNV. [18]

Structures that are slender compared to the wave length, λ , is called hydrodynamically transparent. This implies that the structure only affects the flow locally. For structures with D less than one fifth of the wave length, where D is the projection of the cross section obstructing the flow, Morison's equation is used to derive the wave induced loads normal to the structure. This formula describe such loads as a sum of the forces caused by drag of the flow relative to the structure movement, and the inertia forces of the displaced water volume. [18]

$$dF(z, t) = \rho A \dot{v} + \rho C_A A (\dot{v} - \ddot{r}) + \frac{1}{2} \rho C_D D (v - \dot{r}) |v - \dot{r}|. \quad (64)$$

In equation (64), dots symbolize time derivation and r is the normal displacement of the structure. v is here the normal particle velocity and must obviously not be confused with the wind turbulence velocity, v , from section 2.3.1. A is the cross sectional area, D is the projected diameter, and ρ is the mass density of the fluid. The dimensionless values C_D and C_A is the drag coefficient and the added mass coefficient, respectively.

2.4 Autoregressive Modeling

The white noise example in 2.1.3 illustrated structure response to a loading process where a load value was picked randomly from a Gaussian distribution. Effectively, this would imply that the value of the excitation at one point in time are independent of the values of the excitation in the near past. As seen in 2.3, environmental processes would generally not display such chaotic behavior, but rather contain some degree of correlation, or frequency dependence.

A time series of the process in question could be simulated as shown in 2.3, this would be an acceptable approach when performing a Monte Carlo simulation of the system response. When employing the probability density evolution method however, one would need represent the behavior of the stochastic process through the use of random variables. This could be achieved through autoregressive modeling. The following derivations on autoregression theory is based on Brockwell and Davis. [19]

The process providing the variable X_t is autoregressive of order p , AR(p), if it is weakly stationary and could be written on the following form for every step t :

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + Z_t. \quad (65)$$

In equation (65), ϕ_1 to ϕ_p are constants with absolute values below 1, and Z_t is a Gaussian distributed random variable with a mean value of zero and a standard deviation σ , picked as white noise.

For the sake of introducing a minimum of additional parameters into the probability density evolution, the environmental processes is in the following PDE analyses modeled as first order autoregressive. A variable determined by a first order autoregressive model, AR(1), will only depend on the closest preceding value of the variable as shown below.

$$X_t = \phi X_{t-1} + Z_t \quad (66)$$

Multiplying both sides of equation (66) with $X_{t-\tau}$ and taking the expectations leads to an expression for the autocovariance function of the process:

$$\begin{aligned} C_X(\tau) &= E[X_t X_{t-\tau}] \\ &= E[\phi X_{t-1} X_{t-\tau}] + E[Z_t X_{t-\tau}] \\ &= \phi C_X(\tau - 1) + 0 \\ &= \phi^\tau C_X(0). \end{aligned} \quad (67)$$

In equation (67),

$$C_X(0) = \sigma^2 / (1 - \phi^2) \quad (68)$$

since

$$\begin{aligned} C_X(0) &= E[X_t X_t] \\ &= E[(\phi X_{t-1} + Z_t) (\phi X_{t-1} + Z_t)] \\ &= \phi^2 C_X(0) + \sigma^2. \end{aligned} \quad (69)$$

From equation (67) one could see that first order autoregression would model a weakly stationary process, with constant mean and variance, for $\phi < 1$. The variance spectrum

of this process could now be found from the discrete version of equation (22), as shown below. [19]

$$\begin{aligned}
S_{X,AR(1)}^+(\omega) &= \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} e^{-i\tau\omega} C_X(\tau) \\
&= \frac{\sigma^2\Delta t}{\pi(1-\phi^2)} \left(1 + \frac{\phi e^{i\omega\Delta t}}{1-\phi e^{i\omega\Delta t}} + \frac{\phi e^{-i\omega\Delta t}}{1-\phi e^{-i\omega\Delta t}} \right) \\
&= \frac{\sigma^2\Delta t}{\pi} (1 - 2\phi \cos(\omega\Delta t) + \phi^2)^{-1}
\end{aligned} \tag{70}$$

Above, Δt is the time step between points to be modeled by the spectrum. Given a specific time series, the parameter ϕ and the variance of the white noise contribution, σ , could for example be estimated using the Yule-Walker equations. [19]

However, if the variance spectrum of the process is known, then the unknown model parameters could be estimated directly using the relationships provided in equations (70), (68) and (24). This could be done numerically by curve fitting, for example using the least squared error method.

Figure 16 shows the example Kaimal wind turbulence spectrum from figure 10, and its estimated first order autoregression spectrum with parameters $\phi = 0.9966$ and $\sigma = 0.203$ m/s, for a time step of $\Delta t = 0.05$ s. The Matlab script used to estimate these parameters could be found in attachment 5.

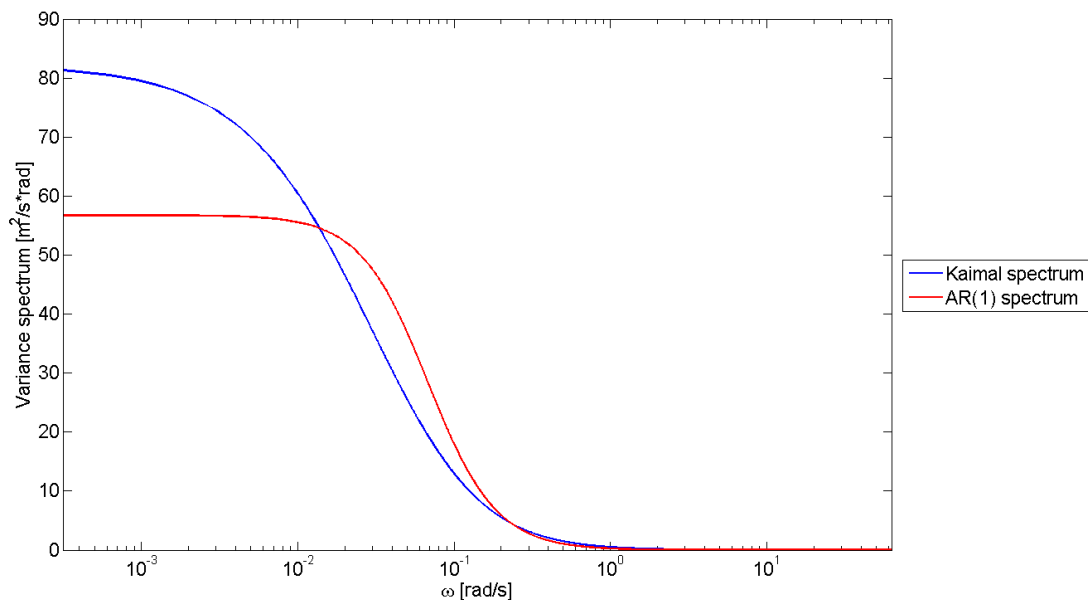


Figure 16: Comparison of the Kaimal spectrum from figure 10 and the best estimate of its first order autoregression spectrum for time step $\Delta t = 0.05$ s.

The resulting AR(1) model is

$$X_t = 0.9966 \cdot X_{t-1} + Z_t \quad (71)$$

where Z_t is a Gaussian distributed white noise random variable with $\mu = 0$ m/s and $\sigma = 0.203$ m/s.

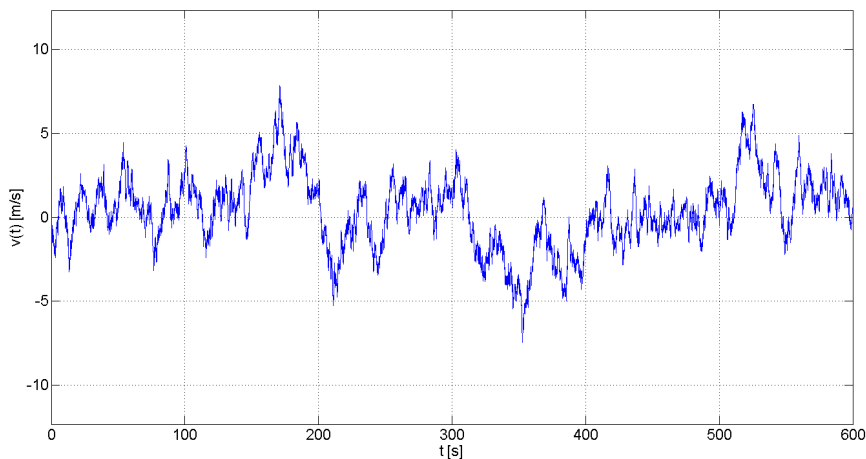


Figure 17: AR(1) simulation of the wind turbulence defined by the spectrum in figure 10.

A time series of the process defined in (71) could now easily be simulated back by specifying the initial value X_0 . X_0 could for example be picked from a Gaussian distribution with the same variance as the total process, $C_X(0)$. This is illustrated over ten minutes in figure 17. While it is worth noting that a higher order autoregressive model would provide a more accurate estimation of the correlation in the underlying process, one could see by comparing figure 17 and 11 that a first order autoregression model would describe the actual turbulence behaviour much better than a white noise signal with the same mean and variance.

2.5 The Finite Element Method in Structural Dynamics

The results of the simplified stochastic analyses are going to be compared to analyses performed using finite element software. Problems in structural dynamics lead to partial differential equations of motion because of their dependency on both time and spatial position of the distributed mass points. The finite element method (FEM) is based on the idea of discretizing the spatial distribution of the mass into elements connected at nodes with defined degrees of freedom. The system could now only vary in space along these degrees of freedom.

When using the finite element method to consider dynamic problems, the displacements along the degrees of freedom is time dependent. After having discretized the structure, it is possible to describe the motion of the system by a set of coupled second-order ordinary differential equations. This results in a problem that could be written in matrix form

and solved numerically if the specific boundary values are known. It is shown by Cook et al. in [20] that the solution of this equation set, under the correct circumstances, would converge to the true solution of the boundary value problem as the number of elements are increased.

2.5.1 Deriving the General Equation of Motion

The principle of virtual work could be used, as shown by Cook et al. in [20], to derive a general form of the equation of motion for a FE discretized structural system.

The principle of virtual work states that work done by external loads must equal the sum of work absorbed by inertial, dissipative, and internal elastic forces for any virtual displacement. A virtual displacement, δu , is any imagined small motion that satisfies essential boundary conditions and displacement continuity between elements.

This equilibrium of work is expressed for a single element of volume V and surface area S as shown in equation (72).

$$\int_V \{\delta \mathbf{u}\}^T \mathbf{F}_V dV + \int_S \{\delta \mathbf{u}\}^T \mathbf{F}_S dS + \sum_{i=1}^n \{\delta \mathbf{u}\}_i^T \mathbf{p}_i = \int_V [\{\delta \mathbf{u}\}^T \rho \ddot{\mathbf{u}} + \{\delta \mathbf{u}\}^T c \dot{\mathbf{u}} + \{\delta \boldsymbol{\epsilon}\}^T \boldsymbol{\sigma}] dV \quad (72)$$

\mathbf{F}_V and \mathbf{F}_S is the prescribed body forces and surface tractions, while \mathbf{p}_i is the prescribed concentrated loads acting in the n corresponding virtual displacements $\{\delta \mathbf{u}\}_i$. ρ is the mass density and c is a viscous damping parameter. $\{\delta \boldsymbol{\epsilon}\}$ represents the strains associated with the virtual displacements $\{\delta \mathbf{u}\}$.

Finite element discretization of the continuous structure lead to the relationships below. The generalized coordinates \mathbf{q} are functions of time, while the shape functions contained in the shape function matrix \mathbf{N} depend on spatial position.

$$\begin{aligned} \mathbf{u} &= \mathbf{N}\mathbf{q} \\ \dot{\mathbf{u}} &= \mathbf{N}\dot{\mathbf{q}} \\ \ddot{\mathbf{u}} &= \mathbf{N}\ddot{\mathbf{q}} \\ \boldsymbol{\epsilon} &= \frac{d}{dx} \mathbf{N}\mathbf{q} = \mathbf{B}\mathbf{q} \end{aligned} \quad (73)$$

Substituting these relationships into the principle of virtual work, and assuming that the concentrated loads \mathbf{p}_i act directly in nodes, would now provide:

$$\begin{aligned} \{\delta \mathbf{q}\}^T & \left(\int_V \rho \mathbf{N}^T \mathbf{N} dV \ddot{\mathbf{q}} + \int_V c \mathbf{N}^T \mathbf{N} dV \dot{\mathbf{q}} + \int_V \mathbf{B}^T \boldsymbol{\sigma} dV \right) \\ & - \{\delta \mathbf{q}\}^T \left(\int_V \mathbf{N}^T \mathbf{F}_V dV + \int_S \mathbf{N}^T \mathbf{F}_S dS + \sum_{i=1}^n \mathbf{p}_i \right) = 0 \end{aligned} \quad (74)$$

In equation (74),

$$\begin{aligned}\int_V \rho \mathbf{N}^T \mathbf{N} dV &= \mathbf{m} \\ \int_V c \mathbf{N}^T \mathbf{N} dV &= \mathbf{c}\end{aligned}\tag{75}$$

is the consistent element mass and damping matrices, respectively. These are called consistent because they are based on the same interpolation functions as the stiffness matrix, and because they are a direct result from finite element discretization. It could be shown that both would be symmetric and positive definite. Nonconsistent forms of these matrices are also used, for example the lumped mass matrix applied in the modal analysis of this thesis.

The other terms in equation (74) could be rearranged into two vectors. \mathbf{r}^{int} describes internal forces (or moments) applied to the element by nodes to resist internal element stresses. \mathbf{r}^{ext} describes external forces (or moments) applied to nodes as a result of external loads acting on the element. These vectors are presented in equation (76).

$$\begin{aligned}\mathbf{r}^{\text{int}} &= \int_V \mathbf{B}^T \boldsymbol{\sigma} dV \\ \mathbf{r}^{\text{ext}} &= \int_V \mathbf{N}^T \mathbf{F}_V dV + \int_S \mathbf{N}^T \mathbf{F}_S dS + \sum_{i=1}^n \mathbf{p}_i\end{aligned}\tag{76}$$

These definitions are true in general. For a linearly elastic material the internal force vector simplifies to $\mathbf{r}^{\text{int}} = \mathbf{k} \mathbf{q}$, where \mathbf{k} is the element stiffness matrix, and the equation of motion becomes

$$\mathbf{m} \ddot{\mathbf{q}} + \mathbf{c} \dot{\mathbf{q}} + \mathbf{k} \mathbf{q} = \mathbf{r}^{\text{ext}}.\tag{77}$$

By using the relationship between the element degrees of freedom and the global degrees of freedom, the global form of the equation of motion is found as:

$$\mathbf{M} \ddot{\mathbf{Q}} + \mathbf{C} \dot{\mathbf{Q}} + \mathbf{K} \mathbf{Q} = \mathbf{R}^{\text{ext}} = \mathbf{F}.\tag{78}$$

In equation (78), \mathbf{M} , \mathbf{C} and \mathbf{K} is the global mass, damping and stiffness matrix of the system, respectively. \mathbf{F} represents the load projection into each degree of freedom. If the global generalized coordinates \mathbf{Q} describes the displacements of interest directly, the equation of motion could be equivalently written as

$$\mathbf{M} \ddot{\mathbf{u}}(t) + \mathbf{C} \dot{\mathbf{u}}(t) + \mathbf{K} \mathbf{u}(t) = \mathbf{F}(t).\tag{79}$$

The structural equation of motion is presented as matrices and vectors in equation (80). Here, n is the total number of degrees of freedom used to discretize the system, and $u_i(t)$ represent the displacement along each of them.

$$\begin{aligned}
& \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nn} \end{bmatrix} \begin{bmatrix} \ddot{u}_1(t) \\ \ddot{u}_2(t) \\ \vdots \\ \ddot{u}_n(t) \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{bmatrix} \begin{bmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \\ \vdots \\ \dot{u}_n(t) \end{bmatrix} \\
& + \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_n(t) \end{bmatrix} = \begin{bmatrix} F_1(t) \\ F_2(t) \\ \vdots \\ F_n(t) \end{bmatrix}
\end{aligned} \tag{80}$$

Equation (79) is discretized in space, but the displacements and forcing are continuous in time. This is therefore a system of coupled second-order ordinary differential equations in time. One could now further discretize time and use direct numerical time integration algorithms such as the Newmark Method, e.g. described in [13], to establish the structure response history.

2.5.2 Modal Analysis

The modal form of the dynamic system is found assuming that the displacements $u(x, t)$ could be written as a superposition of products between $\phi(x)$, accounting for spatial dependence, and $\eta(t)$ accounting for time dependence. Written in matrix form:

$$\mathbf{u}(x, t) = \mathbf{\Phi}(x) \cdot \boldsymbol{\eta}(t). \tag{81}$$

$\mathbf{\Phi}(x)$ is the modal matrix of the system (not to be confused with the autoregression parameter ϕ) and $\boldsymbol{\eta}(t)$ is a vector of generalized modal coordinates. The modal matrix consist of the eigenvectors of the system:

$$\mathbf{\Phi}(x) = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_n]. \tag{82}$$

Both eigenvectors and eigenvalues are determined by solving the eigenvalue problem derived for the undamped and unloaded structure. The equation of motion is in this case given as:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{0}. \tag{83}$$

Substituting for assumed harmonic oscillations, $\mathbf{u}(t) = \boldsymbol{\phi} \cdot e^{i\omega t}$, would now result in

$$(\mathbf{K} - \omega_i^2 \mathbf{M}) \boldsymbol{\phi}_i e^{i\omega_i t} = \mathbf{0}. \tag{84}$$

$e^{i\omega_i t}$ could be disregarded as it is not zero in general for non-trivial solutions, leading to the structural eigenvalue problem presented in equation (85).

$$(\mathbf{K} - \omega_i^2 \mathbf{M}) \phi_i = \mathbf{0} \quad (85)$$

Solving the eigenvalue problem presented in equation (85) yields the eigenvalues ω_i^2 and eigenvectors ϕ_i of the system. Having obtained these one could substitute u in the equation of motion and premultiply by Φ^T :

$$\Phi^T \mathbf{M} \Phi \cdot \ddot{\eta} + \Phi^T \mathbf{C} \Phi \cdot \dot{\eta} + \Phi^T \mathbf{K} \Phi \cdot \eta = \Phi^T \mathbf{F}. \quad (86)$$

Equation (86) could be rewritten to the form of the modal equation of motion:

$$\widetilde{\mathbf{M}} \cdot \ddot{\eta} + \widetilde{\mathbf{C}} \cdot \dot{\eta} + \widetilde{\mathbf{K}} \cdot \eta = \Phi^T \mathbf{F}. \quad (87)$$

Where $\widetilde{\mathbf{M}}$, $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{K}}$ is the diagonal modal mass, damping and stiffness matrix of the system, respectively. The modal matrix equation of motion is decoupled because the eigenvectors of the structural eigenvalue problem $(\mathbf{K} - \omega_i^2 \mathbf{M})\phi_i = \mathbf{0}$ are orthogonal. It could be proven that $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{K}}$ are diagonal matrices by premultiplying equation (85) for the i th solution with ϕ_j^T to obtain

$$\phi_j^T \mathbf{K} \phi_i = \omega_i^2 \phi_j^T \mathbf{M} \phi_i. \quad (88)$$

The similar operation could naturally be performed for the j th solution:

$$\phi_i^T \mathbf{K} \phi_j = \omega_j^2 \phi_i^T \mathbf{M} \phi_j. \quad (89)$$

Transposing equation (88) would yield

$$(\phi_j^T \mathbf{K} \phi_i)^T = (\omega_i^2 \phi_j^T \mathbf{M} \phi_i)^T, \quad (90)$$

and, since \mathbf{M} and \mathbf{K} are symmetric matrices:

$$\phi_i^T \mathbf{K} \phi_j = \omega_i^2 \phi_i^T \mathbf{M} \phi_j. \quad (91)$$

Subtracting equation (91) from equation (89) would result in (92).

$$(\omega_i^2 - \omega_j^2) \phi_i^T \mathbf{M} \phi_j = 0 \quad (92)$$

Equation (92) concludes the proof, showing that $\phi_i^T \mathbf{M} \phi_j$ must equal zero when $i \neq j$. The same result could now be found for the modal stiffness matrix by substituting this result into equation (91).

The decoupling of the modal equation of motion causes the generalized coordinates of each mode to act as a single degree of freedom system, which is a key property of modal analysis. The contributions of each mode could then be superimposed as shown in equation (81) to obtain the real displacement u .

3 Methodology

This section aims to present and discuss the methods used in the thesis.

3.1 The NREL 5-MW Baseline Wind Turbine

The wind turbine considered in this thesis is the U.S. Department of Energy’s National Renewable Energy Laboratory (NREL) 5-MW baseline turbine, as defined by Jonkman et al. [2] This numerical turbine model provides standardized and accurate data for the properties of a realistically sized offshore wind turbine, and it is therefore also used in the OC4 project. Some key properties of the baseline turbine is shown in table 1.

Table 1: Some selected properties of the reference wind turbine. [2]

Rating	5 MW
Rotor orientation, Configuration	Upwind, 3 blades
Control	Variable speed, Collective pitch
Drivetrain	High speed, Multiple-Stage gearbox
Rotor, Hub diameter	126 m, 3 m
Hub height	90 m
Cut-in, Rated, Cut-out wind speed	3 m/s, 11.4 m/s, 25 m/s
Cut-in, Rated rotor speed	6.9 rpm, 12.1 rpm
Rated tip speed	80 m/s
Overhang, Shaft tilt, Precone	5 m, 5°, 2.5°
Rotor mass	110 000 kg
Nacelle mass	240 000 kg
Tower mass	347 460 kg
Overall center of mass	(-0.2 m, 0.0 m, 64.0 m)

3.1.1 Reference Tower

Validation of the thrust load model and comparison between results obtained by probability density evolution and time domain simulations are the main objectives of the thesis. Because of this, the site-dependent offshore support structure and wave loads are not included in the subsequent analyses.

The tower considered in this thesis is the onshore reference tower described in [2]. This tower is modeled as clamped at the base. It has an outer base diameter of 6 m and a base thickness of 0.0351 mm. Both the diameter and thickness tapers linearly to top values of 3.87 m and 0.0247 mm, respectively. The material used is steel with a Young’s modulus of 210 GPa and a shear modulus of 80.8 GPa. To account for secondary steel and paint, the mass density of the steel is raised from 7850 kg/m³ to 8500 kg/m³. The tower has a specified structural damping ratio of 1 % for all modes.

I have divided the tower into 20 nodes and 20 elements in order to obtain estimates of the first eigenmodes and frequencies. Each node has one transversal degree of freedom.

The lumped mass technique, e.g. covered by Chopra in [21] or Murtagh et al. in [22], is used to obtain a diagonal mass matrix for the tower. The entries of the mass matrix are shown in the last column of table 2, since $\mathbf{M} = \text{Diag}[M_{ii}]$ where $M_{ii} = m_i$.

The lumped masses are determined by the mass of the corresponding tower section with height dz_i . The lumped mass acting in the degree of freedom on the top of the tower, DOF 20, also includes the mass of the rotor and nacelle from table 1. In table 2, Height_{*i*} refers to the height of the DOF measured from the base of the tower, while HtFract_{*i*} refers to the percentage of the total height of the tower.

Table 2: Discretization and lumped masses of the reference tower.

DOF #, <i>i</i>	Height _{<i>i</i>} [m]	HtFract _{<i>i</i>} [-]	dz _{<i>i</i>} [m]	Thickness _{<i>i</i>} [m]	Outer diameter _{<i>i</i>} [m]	Volume _{<i>i</i>} [m ³]	m _{<i>i</i>} [kg]
1	4,38	0,05	6,57	0,03458	5,89350	4,216663	35841,6
2	8,76	0,10	4,38	0,03406	5,78700	2,696303	22918,6
3	13,14	0,15	4,38	0,03354	5,68050	2,606227	22152,9
4	17,52	0,20	4,38	0,03302	5,57400	2,517668	21400,2
5	21,90	0,25	4,38	0,03250	5,46750	2,430626	20660,3
6	26,28	0,30	4,38	0,03198	5,36100	2,345101	19933,4
7	30,66	0,35	4,38	0,03146	5,25450	2,261092	19219,3
8	35,04	0,40	4,38	0,03094	5,14800	2,178599	18518,1
9	39,42	0,45	4,38	0,03042	5,04150	2,097624	17829,8
10	43,80	0,50	4,38	0,02990	4,93500	2,018165	17154,4
11	48,18	0,55	4,38	0,02938	4,82850	1,940222	16491,9
12	52,56	0,60	4,38	0,02886	4,72200	1,863797	15842,3
13	56,94	0,65	4,38	0,02834	4,61550	1,788888	15205,5
14	61,32	0,70	4,38	0,02782	4,50900	1,715495	14581,7
15	65,70	0,75	4,38	0,02730	4,40250	1,643619	13970,8
16	70,08	0,80	4,38	0,02678	4,29600	1,573260	13372,7
17	74,46	0,85	4,38	0,02626	4,18950	1,504418	12787,5
18	78,84	0,90	4,38	0,02574	4,08300	1,437092	12215,3
19	83,22	0,95	4,38	0,02522	3,97650	1,371282	11655,9
20	87,60	1,00	2,19	0,02470	3,87000	0,661437	355622,2

In order to obtain the modes of vibration for the tower one would also need to know the stiffness matrix of the structure. In this thesis the stiffness matrix \mathbf{K} is obtained through the flexibility matrix \mathbf{f} of the structure. The flexibility matrix of the tower have been obtained using the unit load method, i.e. introducing a unit load at each degree of freedom and measuring the corresponding displacement in all degrees of freedom. This was done using a tower model with 40 elements and nodes in the FEM frame analysis program "Focus Konstruksjon 2014" developed by Focus Software. [23]

The resulting flexibility matrix are presented in attachment 6. The stiffness matrix of the tower could now be found as the inverse of the flexibility matrix, $\mathbf{K} = \mathbf{f}^{-1}$. Knowing both the mass and stiffness matrix, the undamped and unloaded equation of motion becomes

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = 0. \quad (93)$$

Assuming harmonic response, $\mathbf{u}(t) = \boldsymbol{\phi}e^{i\omega t}$, and using the knowledge that $e^{i\omega t}$ is not zero in general, equation (93) simplifies to

$$(\mathbf{K} - \omega^2\mathbf{M})\boldsymbol{\phi} = 0. \quad (94)$$

Equation (94) is the well known eigenvalue problem in structural dynamics, the solution of which results in eigenvalues equal to the squared natural frequencies of the tower and eigenvectors giving the corresponding modes of vibration, $\boldsymbol{\phi}$. These computations are performed using the Matlab script in attachment 7, and the results are presented in section 4.1.

3.2 TurbSim

TurbSim is a software developed by NREL to generate numerical, turbulent wind fields based on the theory presented in 2.3.1. TurbSim generates time domain simulations of three dimensional wind velocity vectors at points on a two dimensional grid which is fixed vertically in space. [24] This is illustrated in figure 18.

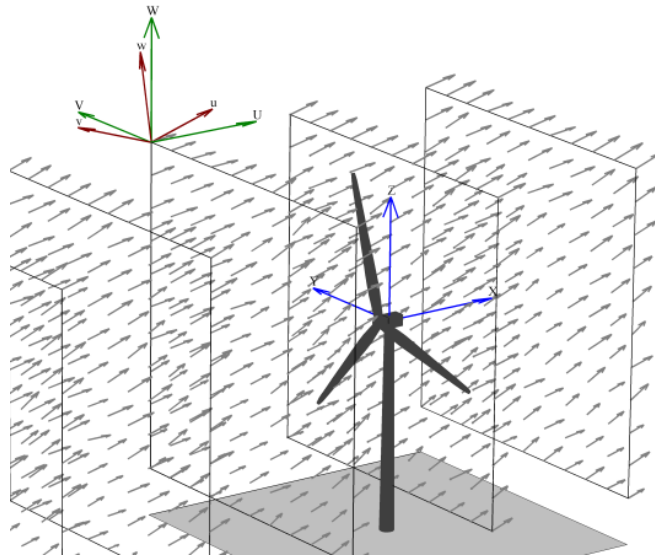


Figure 18: Illustration of a wind velocity field generated by TurbSim. [24]

In this thesis TurbSim have been used to generate the wind fields used in time domain simulations of the wind turbine. TurbSim reads an input file containing the necessary information to simulate the wind velocity field. An example of such an input file, for a 12 m/s reference wind speed and turbulence class B, is shown in attachment 8.

3.3 FAST

FAST, an acronym for Fatigue, Aerodynamics, Structures and Turbulence, is an aeroelastic finite element simulation code developed by NREL capable of simulating an entire wind turbine in time domain. [25] FAST is certified by Germanischer Lloyd for the calculation of onshore wind turbine loads for design and certification purposes. Wind velocity simulations from TurbSim are used as input for the environmental loading in the FAST time domain simulations. FAST input files describing the reference turbine from section 3.1, and its control system, are also provided by NREL.

In this thesis, FAST time domain simulations have been used to obtain time series for both rotor thrust forces and tower top response. These simulations consider flexible motion of the tower and blades with aerodynamic damping, the control system of the turbine and periodic gravity loads.

3.4 Environmental Load Cases

The 15 wind velocity situations considered in this thesis are taken from IEC 61400 3rd edition [15], using turbulence class B. For this turbulence class, the Normal Turbulence Model shown in equation (48) provides the standard deviation of the fluctuating wind velocity using $I_{ref} = 0.14$.

Table 3: The wind velocity load cases considered in this thesis.

Load Case	V_{ref} [m/s]	σ_v [m/s]	Turbulence Intensity [%]
1	2	0,994	49,7
2	4	1,204	30,1
3	6	1,414	23,6
4	8	1,624	20,3
5	10	1,834	18,3
6	12	2,044	17,0
7	14	2,254	16,1
8	16	2,464	15,4
9	18	2,674	14,9
10	20	2,884	14,4
11	22	3,094	14,1
12	24	3,304	13,8
13	26	3,514	13,5
14	28	3,724	13,3
15	30	3,934	13,1

These 15 load cases are presented in table 3, and the turbulence intensity is shown as a function of the reference wind velocity in figure 19. The AR(1) model parameters for these turbulence cases have been estimated using the Matlab script in attachment 5. The results of these calculations are presented in section 4.3.

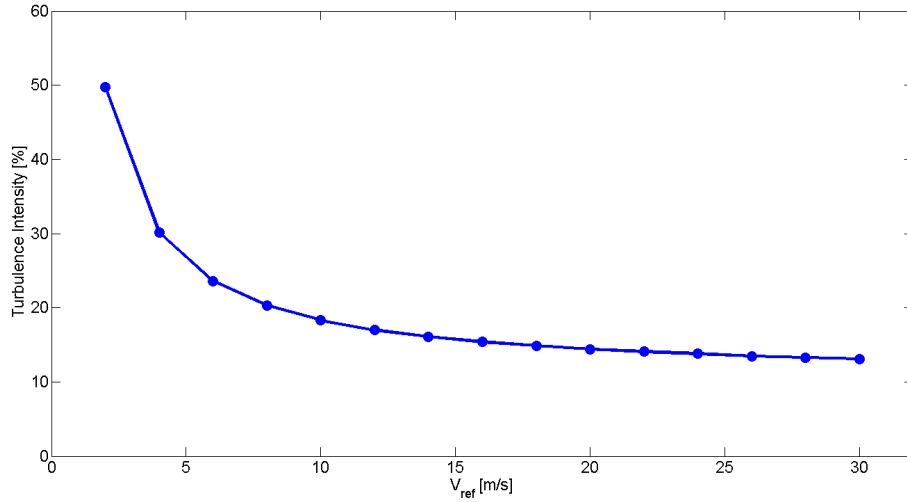


Figure 19: Plot of the class B turbulence intensities for the different reference wind velocities considered.

3.5 Development of a Simplified Load Model

To obtain the response distribution of the system through the probability evolution method one would, as it is made apparent in section 2.2, need to know the probability distribution of the exciting force. The real loading situation of an offshore wind turbine is governed by complex environmental processes, nonlinear coupling effects and cyclic loads from rotational components. It is therefore necessary to develop a simplified load model for which it is feasible to determine the probability density.

Since only the response in the main wind direction is considered, relevant contributions to the load situation of an offshore wind turbine would be the total thrust force from wind loads acting on the rotor, wind loads distributed along the tower above water and wave loads distributed along the sub-structure under water. The two latter load contributions could be calculated using equation (55) and (64), respectively.

The accurate estimation of the former load contribution is far more complex, as it depends on rotational wake of the wind, interaction between wind and the flexible motion of the blades and the control system of the turbine. The total thrust from wind loads on the rotor would naturally also be the most significant load for the tower top displacement of a wind turbine, and its simplified description would therefore be the main objective in the current load model development.

In general, aerodynamic drag force on an object is described using a dimensionless drag coefficient and equation (55), where the drag force is proportional to the wind velocity squared. For a rotor, it is the relative wind that determines the sectional contributions to lift and drag along the airfoil axes. This could be seen in figure 12. As it is mentioned earlier in this thesis, the load resultants on the rotor also depends on the turbine control system and interrelation effects between the blades and the local wind velocities. It is therefore not obvious that there exists a good model of the rotor thrust force on a form where the thrust is proportional to the *incoming* wind velocity squared.

For example, when using the attached BEM Matlab script to compare force contributions dT from different blade sections, one would see that the largest contribution to the total thrust originates from lift forces in the outer parts of the rotor blades. In these blade regions, the angular velocity of the rotor ω generates a very high tangential velocity, ωr . Since the angular velocity of the rotor is not very sensitive to fluctuations in the wind velocity, one could assume that wind turbulence would cause less variance in the total thrust force of the rotor than it would do for a stationary object with the same area.

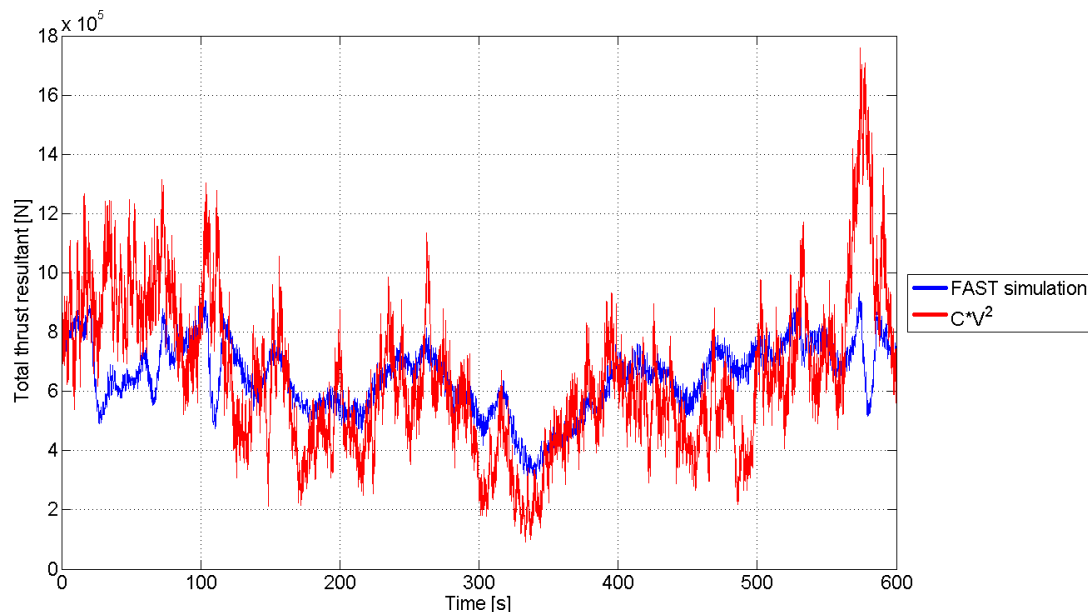


Figure 20: Comparison between a FAST simulation of the reference turbine for a 10 m/s reference wind speed in turbulence class B, and a corresponding simplified model based on the squared wind velocity.

Even though this is clearly not the only factor influencing the total loading in the main wind direction, figure 20 does indeed illustrate that the total thrust calculated using FAST displays less erratic behavior than a simplified model CV^2 , where C is a total drag coefficient determined such that the mean value of the two simulations match each other. Note that since the mass density of the air and the area are implemented in C , this simplified drag coefficient would have the dimension kg/m. All things considered, there is no reason to think that there should exist one *constant* simplified drag coefficient for the turbine either, since the behavior of the total thrust resultant are very much dependent on the control system. [2]

An appropriate load model describing the total thrust resultant acting in the mean wind direction at hub height, over certain V_{ref} intervals, are in this thesis assumed to exist on the form:

$$T(V_{ref}, v(t)) = \bar{T}(V_{ref}) + \hat{t}(V_{ref}, v(t)) \quad (95)$$

where

$$\bar{T}(V_{ref}) = \bar{C}_s V_{ref}^\alpha \quad (96)$$

and

$$\hat{t}(V_{ref}, v(t)) = \hat{C}_s V_{ref}^\beta v(t). \quad (97)$$

The subscript s is short for "simplified". The two simplified coefficients and the exponents α and β are determined by curve fitting using results from FAST simulations and the Matlab script in attachment 9. As it could be seen from the equations above, the total thrust is now given by a term that describes the time invariant mean thrust and a term that also varies with the reference wind velocity, but depends linearly on the turbulence, $v(t)$.

The linear dependence on turbulence is certainly not obvious at this point, but it is later seen in section 4.2 that this model describes the mean and variance of the fluctuations in the rotor thrust well. It is also an advantage to avoid having a random variable with fractional exponents.

This result could also be justified, somewhat heuristically, by assuming that the load model could be written on a form where the thrust resultant is proportional to some exponential expression of the total wind velocity, $f(v) = (V_{ref} + v(t))^q$. A Taylor series expansion around $v(t) = 0$ would now produce the series

$$\begin{aligned} f(v) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} (v - 0)^n \\ &= V_{ref}^q + q V_{ref}^{q-1} v + \frac{1}{2} q(q-1) V_{ref}^{q-2} v^2 + O(v^3). \end{aligned} \quad (98)$$

Assuming that $V_{ref} \gg v(t)$ and neglecting higher order terms of $v(t)$ in equation (98) would leave the following estimation of $f(v)$ for values of $v(t)$ close to zero:

$$f(v) \approx V_{ref}^q + q V_{ref}^{q-1} v. \quad (99)$$

In the proposed load model, \bar{C}_s would now have the unit $(\text{kg m}^{4-\alpha})/(\text{m}^3 \text{s}^{2-\alpha})$ and \hat{C}_s would have the unit $(\text{kg m}^{3-\beta})/(\text{m}^3 \text{s}^{1-\beta})$. The resulting models for the environmental load cases considered in this thesis are presented in section 4.2.

3.6 The Probability Density Evolution Script

The probability density evolution method is central in this thesis. The following sections elaborates, based on the theory presented in section 2.2, how the Matlab script in attachment 2 works. This is done by first considering a case of basic white noise forcing, and then explaining how autoregressive theory is used to incorporate correlated excitation.

3.6.1 Basic PDE Functionality with Uncorrelated Forcing

For white noise forcing, the probability density function of the force alone provides enough information to estimate the joint probability density of the response in a displacement-velocity state space.

In this case, the force, displacement and velocity intervals are divided into cells with assigned cell index numbers and respective widths of dF , dX and $d\dot{X}$. The displacement and velocity cells constitute a discretized state space domain where each two dimensional cell maps into a new cell through an estimate of the response for small time steps. This new cell is called the image cell of the original cell.

For displacements, the image cell estimate is deterministic when knowing the original cell. That is, for each cell in the state space domain there is one image cell index for displacement. In the PDE script this cell is found by using the modified Euler method as shown in equation (100).

$$X_2 \approx X_1 + \Delta t \left(\dot{X}_1 + \frac{d\dot{X}}{2} \right) \quad (100)$$

In this equation, X_1 and X_2 represents the original and the new displacement. Velocity response also depends on the force working at that time, and since the force is a random variable one must estimate the velocity response for each force cell combined with each displacement-velocity state cell. Since X_2 now is estimated, the implicit Euler method could be used with the SDOF equation of motion (1) to find the image cell for each state and load cell:

$$\dot{X}_2 \approx \dot{X}_1 + \Delta t \left(\frac{F_1 - C \frac{X_2 - X_1}{\Delta t} - K X_2}{M} \right). \quad (101)$$

In equation (101), \dot{X}_1 and \dot{X}_2 represents the original and the new velocity state, while F_1 is the force acting in the original state. The PDE script uses these two equations to develop a two dimensional state map for displacements X , where each entry is the displacement image cell index number, and a three dimensional state map for velocities \dot{X} , where each entry is the velocity image cell index number.

Transition between cells is now completely deterministic, and the state maps could be used to evolve the probability density as illustrated in section 2.2. For each displacement, velocity and force cell index number, the image cell index number are determined for both displacement and velocity. The contribution to the image cell probability content is now given by the probability of being in the original state cell, determined by the previous

iteration, multiplied with the probability of the current force realization.

An initial probability density description of the response space is required to get the iterations going, this could for example be a deterministic initial condition where one starts at a given cell with a probability of one. For increased efficiency, a sensible selection of distributed points or a good guess of the response distribution could also be used as an initial condition.

Convergence towards the exact response probability distribution is obtained as the number of iterations increase and the total diffusion of probability goes towards a state of equilibrium, where the probability content assigned to each cell is the same as the probability content leaving the cell for each iteration. As mentioned in section 2.2, the relationship between the discretization of the mapping domain and the time step used is crucial for both efficiency and accuracy of the PDE method.

Figure 8 shows a converged joint probability density result for the white noise loading example from section 2.1.3.

3.6.2 Incorporating Correlated Forcing Through First Order Autoregressive Modeling

Knowing the probability density function of the environmental loading is in general not enough to establish a good estimate the structure response, since this response also depends on the frequency content of the force.

Section 2.4 presents how a stochastic process could be modeled by describing the next value as a deterministic weighted sum of previous values and a random Gaussian noise term. This thesis would only consider first order autoregression, i.e. that the next value of the force depends exclusively on the previous value. In this case, the relationship between two neighboring force realizations need to be accounted for when evolving the probability density of the response. In the attached script, this is done by letting F describe the previous force realization and incorporating it into the state of system. The weighting factor ϕ and the new random variable describing the autoregression noise term, Z , would now provide a mapping between a given force state and the next.

In this case, the total state space is four dimensional, including the previous and the current force state, displacement and velocity. The mapping state space is now three dimensional, including the current force state, displacement and velocity. The projected probability volume of each cell into the displacement-velocity plane would therefore yield the estimated joint probability density of these structural responses.

Obviously, adding a dimension to the system state domain does not come numerically cheap, resulting in considerably reduced efficiency compared to the iterations described in 2.4.

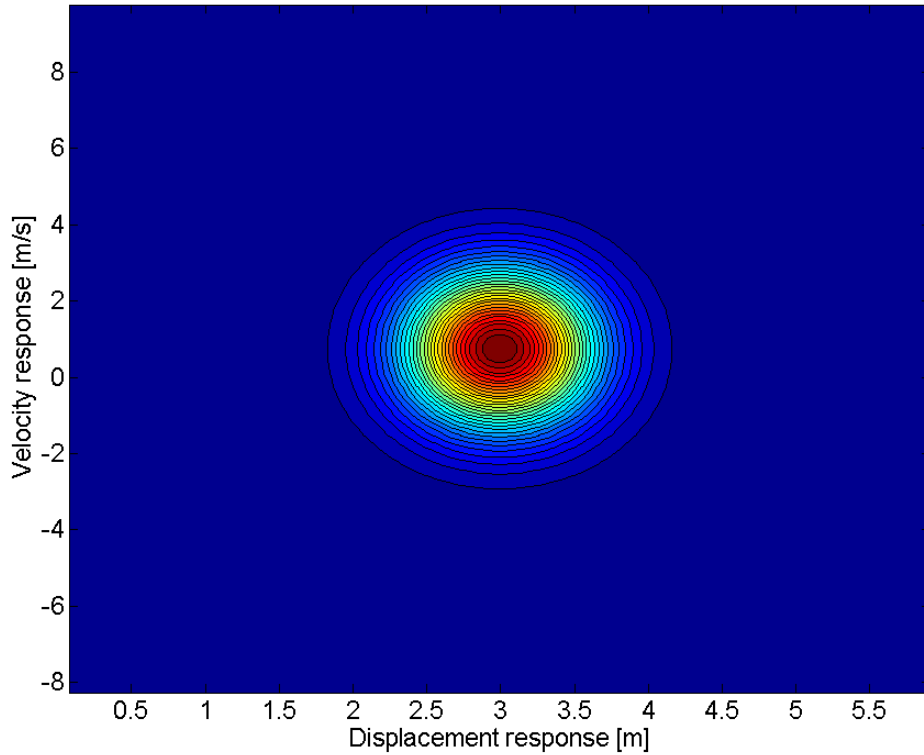


Figure 21: Contour plot of the joint probability density function from figure 6.

Figure 21 is a contour plot showing the analytical solution of the structural response given the white noise example from section 2.1.3.

Given the exact same SDOF structure, we could now assume that it is excited at each time step $\Delta t = 0.05$ s by some force with a mean value of 30 N and fluctuations described by a first order autoregression model with parameters $\phi = 0.7$ and $\sigma^2 = 12.75 \text{ N}^2$. It could be shown, as illustrated for a 300 second simulation in figure 22, that this would produce a long term probability density function of the force that is Gaussian with mean value 30 N and a standard deviation of 5 N.

The PDF of the force is identical to the one used in 2.1.3, but now the realizations of the force are no longer uncorrelated. The correlation between the previous and the current force fluctuation value could be seen in figure 23. The joint PDF in figure 23 is found as the product of the previous force fluctuation PDF and the conditional PDF of the current force fluctuation given the previous one. The former is the total PDF of the force fluctuations, while the latter PDF is given by the autoregressive noise term.

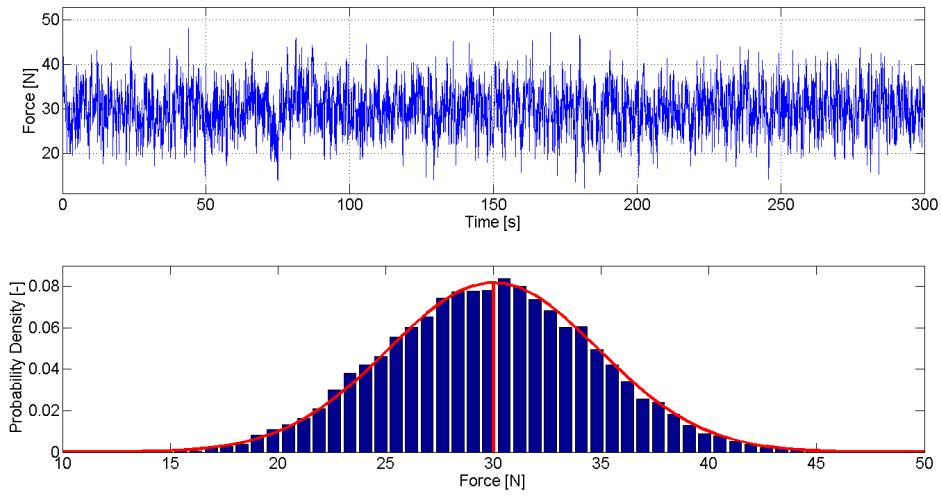


Figure 22: 300 second simulated time series, with histogram and analytical PDF, of an AR(1) load process with $\mu_F = 30$ N, $\phi = 0.7$, $\sigma^2 = 12.75$ N² and $\Delta t = 0.05$ s.

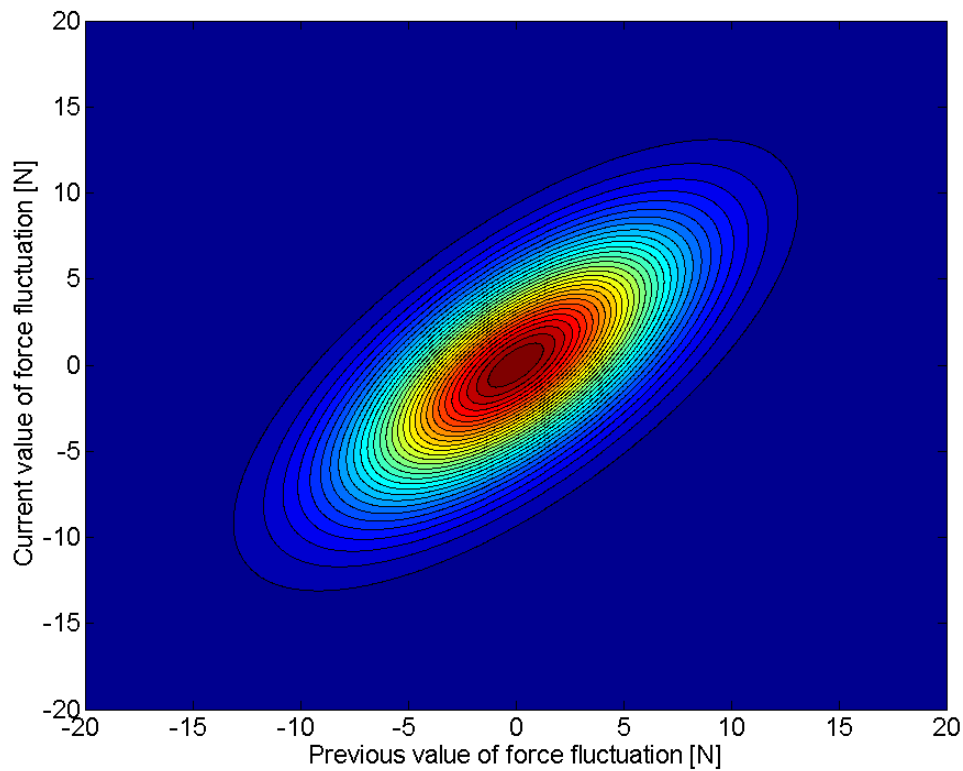


Figure 23: Joint probability distribution of the previous and current value of the described load fluctuations in this example.

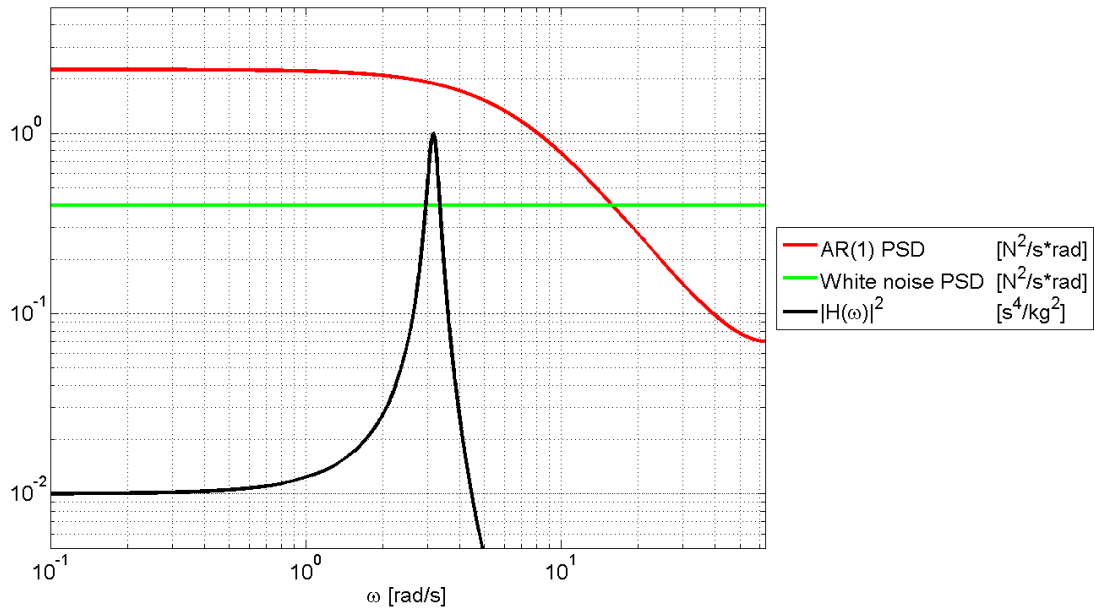


Figure 24: Log-log plot of the one-sided variance spectrum of the autoregressive load process simulated in figure 22, the constant white noise spectra of the same time step and PDF, and the squared frequency response function of the example structure.

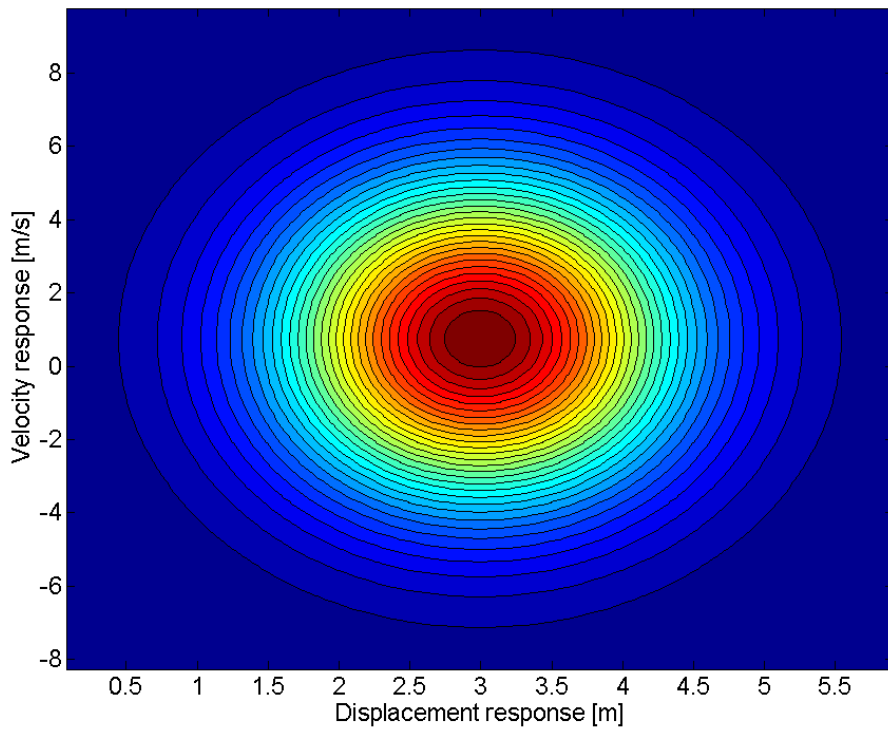


Figure 25: Contour plot showing the analytical joint PDF of the structural response to the autoregressive load process from figure 22 and 24.

The definition of a joint probability of two correlated random variables A and B is shown in equation (102).

$$P(A, B) = P(B) \cdot P(A|B) \quad (102)$$

The variance spectrum of the load process, calculated using equation (70), are presented in figure 24 with the squared frequency response function of the structure. From this figure, one could see that the load process has a higher energy content near the natural frequency of the structure compared to the constant white noise spectrum from figure 3. The joint probability density of the displacement and velocity response under this load process should therefore be wider than it is under white noise excitation. This is also shown to be the case in figure 25, where the analytical joint probability density of the response are calculated using theory from section 2.1.2.

The PDE script could now also be used to estimate the response to this load process, for example by choosing the interval discretizations $dX = 0.05$ m, $d\dot{X} = 0.05$ m/s, $dF = 0.5$ N and $dZ = 0.5$ N. For these input values, a contour plot of the estimated response is presented in 26. This figure shows good agreement with the analytical solution in figure 25. The noise in the estimated distribution is mainly caused by the relatively coarse force discretization in this example, showing the importance of domain discretization on the accuracy of PDE results.

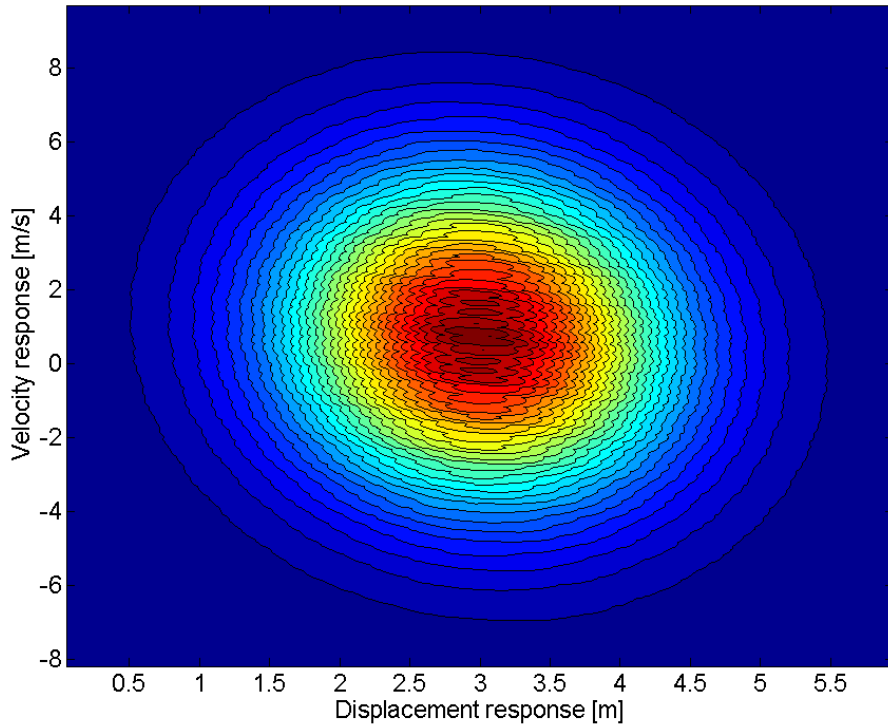


Figure 26: The joint probability distribution of the displacement and velocity response for the example in section 3.6.2, estimated using the probability density evolution script with correlated excitation in attachment 2.

3.6.3 General Comments on Numerical Cell-to-Cell Mapping

While they are two different numerical methods, there is also some similarities between cell-to-cell mapping and classical Monte Carlo simulation.

MCS involves calculating the response for each time step in a continuous domain and assigning probability to regions of this domain based on the number of occurrences within the region. This implies that the number of time steps, or sampling points, would need to be large in order to accurately estimate low probability regions of the response distribution.

The cell-to-cell numerical probability evolution method is basically an iteration scheme where each iteration could be considered as a MCS from one time step to another for each discretized cell. Each cell is represented by a single point used as an initial state for each iteration. The responses to these Monte Carlo simulations is then arranged into target cells in a state space domain grid, and probabilities for the next iteration could be found. The subdivision of each cell to be represented by a single central point would lead to discretization errors, as illustrated in figure 27 for a two dimensional state space described by x_1 and x_2 .

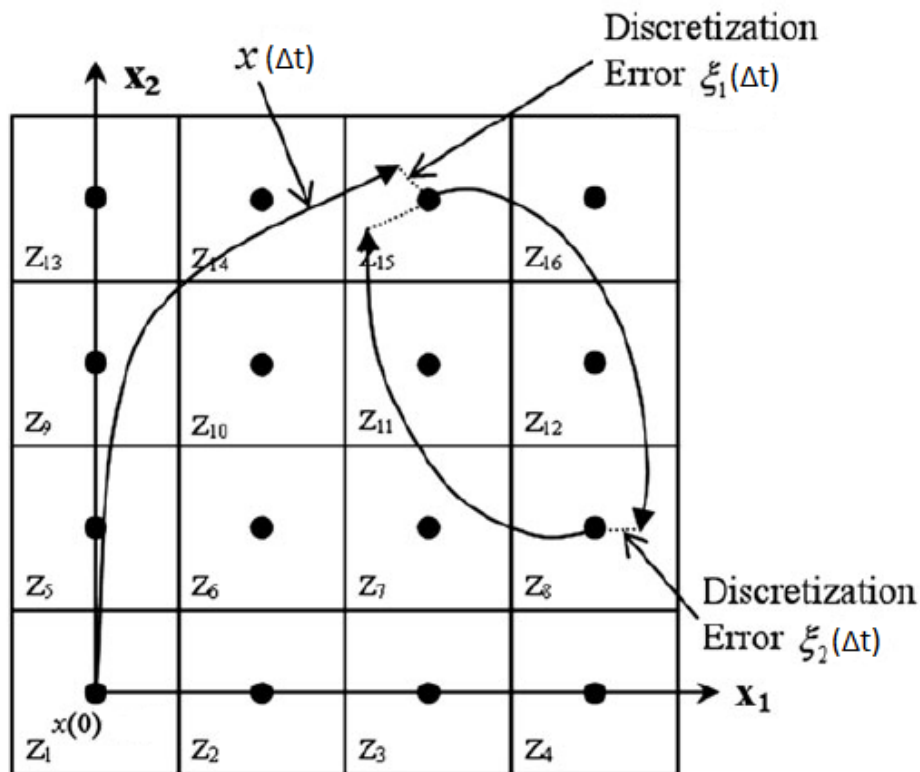


Figure 27: Illustration of iteration errors associated with the state space discretization. [11]

Because of these errors, it is clear that cell-to-cell mapping requires a fine discretization of the state space for accurate probability estimation. This would result in a large number of cells, as opposed to a large number of sampling points. A tradeoff between accuracy

and efficiency would often have to be made, at least for multidimensional state spaces, as the number of cells increase exponentially as the state space is refined. A discretization of N cells in each of the n dimensions would result in a total of N^n cells.

As explained in section 3.6.2, incorporating the previous force realization into the state space would result in four state space dimensions and a total number of cells that would experience quartic growth with increasing refinement in each dimension. Even using parallel computing, a state space with more than five dimensions would probably push the boundaries of what is realistic to evolve with the computational power available today. [11] This would pose a challenge when using cell-to-cell mapping to estimate the response of systems with multiple degrees of freedom, or when using higher order autoregression models to simulate correlated excitation.

It should be noted that improved cell-to-cell mapping methods which relieves the required domain discretization to some extent does exist. For example, the Generalized Cell Mapping method proposed by Hsu randomly assigns several initial starting points within each cell for each iteration. [1] Another method, proposed by Levitas et al. in [26], uses a hypersurface system constraint to reduce the cell space dimension to $n - 1$. Such PDE methods is however beyond the scope of this thesis.

Another factor that is naturally affecting accuracy of the cell-to-cell method is the time integration scheme used in the iterations. This integration scheme would have to be evaluated with respect to the behavior (stiffness) of the differential equation and the time step used to avoid energy drift from leading to a divergent system.

For the relevant time steps and structure considered in this thesis, using the explicit Euler method to estimate velocities caused substantial energy drift in early versions of the PDE script. As mentioned in section 3.6.1, the modified Euler method is used for displacements and the implicit Euler method is used for velocities in the final version of the script.

More accurate energy conserving methods, like those described by Hairer et al. in [27], could of course also be used to estimate the response in probability evolution algorithms. The reward of implementing a more computationally expensive time integration scheme should obviously be weighed against any loss of performance.

It is clear that a small time step is necessary to obtain accurate estimates of the response for each iteration. For the cell-mapping method used in this thesis, the number of required cells is closely related to the time step used. This is because the time step should not become too small, depending on the structural properties of the structure and the applied force distribution, compared to the domain covered by each cell. Probability content for cells in such a system could end up mapping into itself for any probable force, and therefore never evolving beyond the initial cells.

3.7 Calculating the Nonlinear First Mode Response Contribution from Wind Loading Along the Tower

This section is motivated by the fact that, as could be seen in section 4, a simplified model with linear dependence on turbulence is found to describe the rotor thrust accurately enough with respect to mean and variance. This rather unexpected result renders probability evolution unnecessary for Gaussian distributed turbulence, as the thrust would be Gaussian as well, and an analytic solution of the response could be found using theory from section 2.1. However, a more complete and realistic load model would surely not be linear in general for an offshore wind turbine, this could both be true for the rotor thrust and distributed loads along the structure. In particular, both wave loads acting on the sub-structure and wind loads acting on the tower would both need to be modeled as nonlinear. As could be seen from equation (64) and (55), wave loads would be nonlinear with respect to water particle velocity, and drag forces on the tower are nonlinear with respect to wind turbulence velocity.

Especially wave loads could be assumed to be non-negligible with regards to the total displacement of the wind turbine, at least for bottom fixed sub-structures at relatively deep waters, such as jackets. The contribution of wind loading distributed along the tower was not included in the simple load model acting on the onshore reference turbine in this thesis, as rotor thrust would dominate the forcing in the main wind direction.

This section therefore aims to illustrate the usefulness of probability density evolution associated with nonlinear processes. This is done by calculating the contribution to the structural response from wind acting along the tower alone for load case 2, as an example. This section also aims to justify the neglect of tower wind loads in the simplified load model by illustrating that this contribution is small compared to the rotor thrust, at least for the load case considered.

3.7.1 General Description of the Modal Force

In section 3.1.1, the tower was discretized into 20 degrees of freedom. The distributed force from wind loads could then also be discretized and represented by the 20 sectional contributions in the force vector $\mathbf{F} = [F_1, F_2, \dots, F_i, \dots, F_{19}, F_{20}]^T$.

In \mathbf{F} , the i th sectional force is found by combining equation (46) and (55) to F_i as shown in equation (103).

$$F_i = \frac{1}{2} \rho C_{D,i} d_i dz_i \left[V_{ref} \left(\frac{z_i}{z_{ref}} \right)^\alpha + v(t) \right]^2 \quad (103)$$

In equation (103), d_i and dz_i is the sectional diameter and height. The spatial dependence of the turbulence is neglected, and the turbulence at reference height is used for all heights. While this is a simplified assumption, it is certainly a conservative one. In a realistic situation the standard deviation of the fluctuations would vary with height, and spatial correlation of the wind would make it highly implausible that the turbulence along the tower should act in the same direction at all times.

Expanding equation (103) would lead to

$$F_i = A_i v(t)^2 + B_i v(t) + C_i \quad (104)$$

where

$$\begin{aligned} A_i &= \frac{1}{2} \rho C_{D,i} d_i dz_i \\ B_i &= \rho C_{D,i} d_i dz_i V_{ref} \left(\frac{z_i}{z_{ref}} \right)^\alpha \\ C_i &= \frac{1}{2} \rho C_{D,i} d_i dz_i V_{ref}^2 \left(\frac{z_i}{z_{ref}} \right)^{2\alpha}. \end{aligned} \quad (105)$$

As shown in equation (106), the modal force R_j of a given mode j could now be found using the relevant eigenvector.

$$\begin{aligned} R_j(v) &= \phi_j^T \mathbf{F} \\ &= v^2 \sum_{i=1}^{20} \phi_{j,i} A_{j,i} + v \sum_{i=1}^{20} \phi_{j,i} B_{j,i} + \sum_{i=1}^{20} \phi_{j,i} C_{j,i} \\ &= \hat{A}_j v(t)^2 + \hat{B}_j v(t) + \hat{C}_j \end{aligned} \quad (106)$$

The modal sums in equation (106) provide three constants of the modal force, \hat{A} , \hat{B} and \hat{C} . These constants depend on the sectional properties, mean wind velocities and the mode shape in question. It could be seen that the modal force depends nonlinearly on turbulence, even though one could argue that this nonlinear term would be negligible for load situations with small turbulence intensities, where $V_{ref} \gg \sigma_v$.

3.7.2 Probability Density Function of the Modal Force

Because of the squared turbulence term in equation (106), the modal force would not be Gaussian distributed even if the turbulence is. In the following, v is treated as a realization of the random variable V representing the wind turbulence velocity, and r is treated as a realization of the random variable R representing the modal force. Equation (106) now provide a mapping between these two random variables, $R = g(V)$. This mapping could be used with the principle of probability density conservation to obtain the PDF of the modal force. Figure 28 shows a generic plot of equation (106) in the random space represented by V and R .

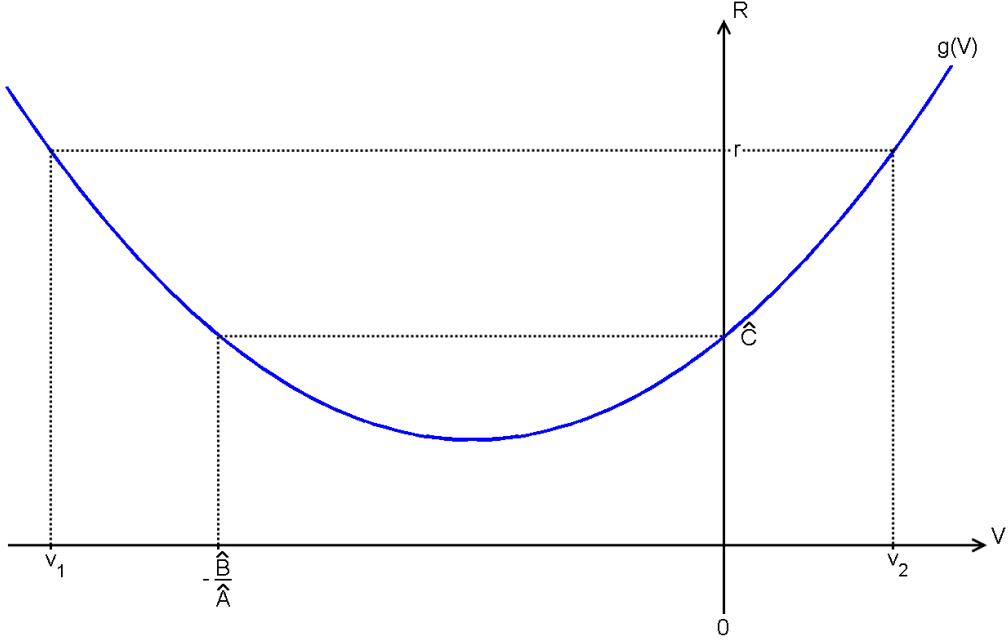


Figure 28: Sketch of the modal force from equation (106) as a function of the random variable V .

Using the principle of probability density conservation, it becomes clear from figure 28 that the cumulative probability density function (CDF) of the modal force could be written as:

$$\begin{aligned}
 F_R(r) &= P[R \leq r] \\
 &= P[v_1 < V < v_2] \\
 &= P[V < v_2] - P[V < v_1] \\
 &= F_V[v_2] - F_V[v_1].
 \end{aligned} \tag{107}$$

The PDF of R is now found as the derivative of the CDF of R , as shown in equation (108).

$$\begin{aligned}
 f_R(r) &= \frac{d}{dr} F_R(r) \\
 &= \sum_{i=1}^2 f_V(v_i) \left| \frac{d}{dr} v_i \right|
 \end{aligned} \tag{108}$$

In equation (108), v_i is the two real roots of $g(v_i) = r$.

$$\left. \begin{array}{l} v_2(r) \\ v_1(r) \end{array} \right\} = \frac{-\hat{B} \pm \sqrt{\hat{B}^2 - 4\hat{A}(\hat{C} - r)}}{2\hat{A}} \quad (109)$$

The Jacobian of the random variable mapping is found as

$$\begin{aligned} \frac{dv_i}{dr} &= \left(\frac{dr}{dv_i} \right)^{-1} \\ &= \left(\frac{d}{dv_i} (\hat{A}v(t)^2 + \hat{B}v(t) + \hat{C}) \right)^{-1} \\ &= \frac{1}{\hat{B} + 2\hat{A}v_i}. \end{aligned} \quad (110)$$

Given the results above, the PDF of the modal force as a function of modal force would be given by equation (111) for Gaussian distributed turbulence.

$$f_R(r) = \frac{1}{\sqrt{2\pi}\sigma_v\sqrt{\hat{B}^2 - 4\hat{A}(\hat{C} - r)}} \left(e^{-\frac{1}{2}\left(\frac{-\hat{B} + \sqrt{\hat{B}^2 - 4\hat{A}(\hat{C} - r)}}{2\hat{A}\sigma_v}\right)^2} + e^{-\frac{1}{2}\left(\frac{-\hat{B} - \sqrt{\hat{B}^2 - 4\hat{A}(\hat{C} - r)}}{2\hat{A}\sigma_v}\right)^2} \right) \quad (111)$$

The response PDF of the structure under this PDF of the modal force, which is clearly not Gaussian, could not be solved analytically. However, the PDF could easily be implemented, like any other, in the probability density evolution method in order to obtain a numeric estimate of the response PDF for this mode.

It is worth noting that equation (103) does not preserve the sign of the total wind velocity, since $V_{tot}^2 \neq V_{tot} \cdot |V_{tot}|$. As a result of this, equation (111) is only valid for positive values of the modal force larger than the critical value

$$r_c = \hat{C} - \frac{\hat{B}^2}{4\hat{A}}, \quad (112)$$

which is the value of the modal force leading to complex roots in equation (109). This modal force occurs at the corresponding critical turbulence velocity value, v_c :

$$v_c = -\frac{\hat{B}}{2\hat{A}}. \quad (113)$$

3.7.3 Constants of the Modal Force and Resulting Probability Density Function for Load Case 2

As previously mentioned, the nonlinear behavior of the drag force increase with the turbulence intensity. Because of this, load case 2 with $V_{ref} = 4$ m/s and $\sigma_v = 1.204$ m/s was chosen to serve as an example load case to demonstrate the PDE functionality for nonlinear processes.

The representative vertical position, height and diameter of the sections are presented in table 2. The mass density of the air is set to $\rho = 1.25$ kg/m³, $\alpha = 0.14$ and $z_{ref} = 90$ m. The drag coefficient for each section depends on the Reynolds number of the airflow:

$$Re_i = \frac{V_{tot,i} d_i \rho}{\mu_{air}}. \quad (114)$$

The dynamic viscosity of air is set to $\mu_{air} = 1.81 \cdot 10^{-5}$ N s/m². Using the mean velocity at each section to calculate Re_i would result in the plot shown in figure 29.

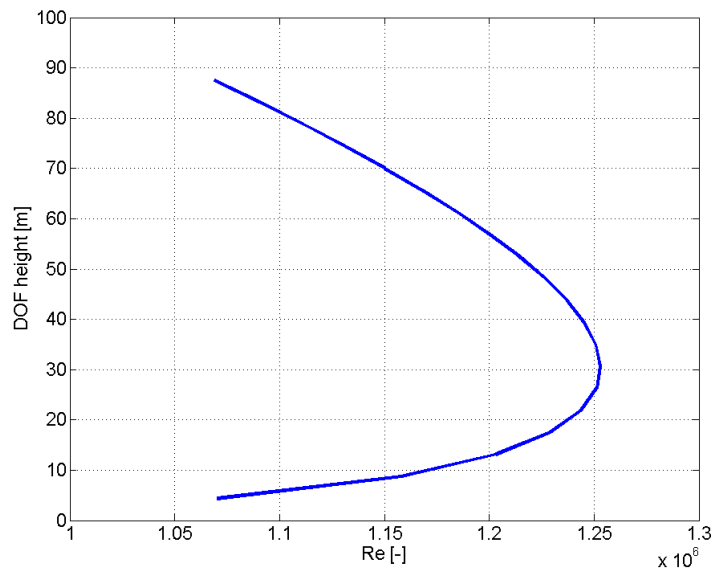


Figure 29: Reynolds number at different heights of the tower for $V_{ref} = 4$ m/s and $\alpha = 0.14$.

For the Reynolds numbers in this figure the drag coefficient C_D could be approximated as a constant, $C_{D,i} \approx C_D = 0.3$. This value could for example be determined using the C_D diagram of a two dimensional smooth cylinder from [28].

Given these sectional properties and the mode shapes presented in section 4.1, the constants of the modal force because of wind loads distributed along the tower are summed up to the values presented in table 4, for the first two tower modes.

Table 4: The constants of the modal force as defined in equation (106), caused by wind loads distributed along the tower, for the first two fore-aft tower modes.

	\hat{A} [kg/m]	B [kg/s]	\hat{C} [N]
Mode 1	25,0108	190,2907	362,7748
Mode 2	41,9390	305,6178	558,6578

Figure 30 and 31 show histograms of simulated modal drag force realizations for mode 1 and 2 respectively. These figures also show the analytical PDF of the modal forces from equation (111), plotted in red for values larger than $r_c = 0.83$ N for mode 1 and $r_c = 1.89$ N for mode 2. Modal force values smaller than r_c will be disregarded in the following analyses. In practice, this would mean neglecting small negative values of the modal force. Such values represent about 0.5 % of the total probability content for both modes.

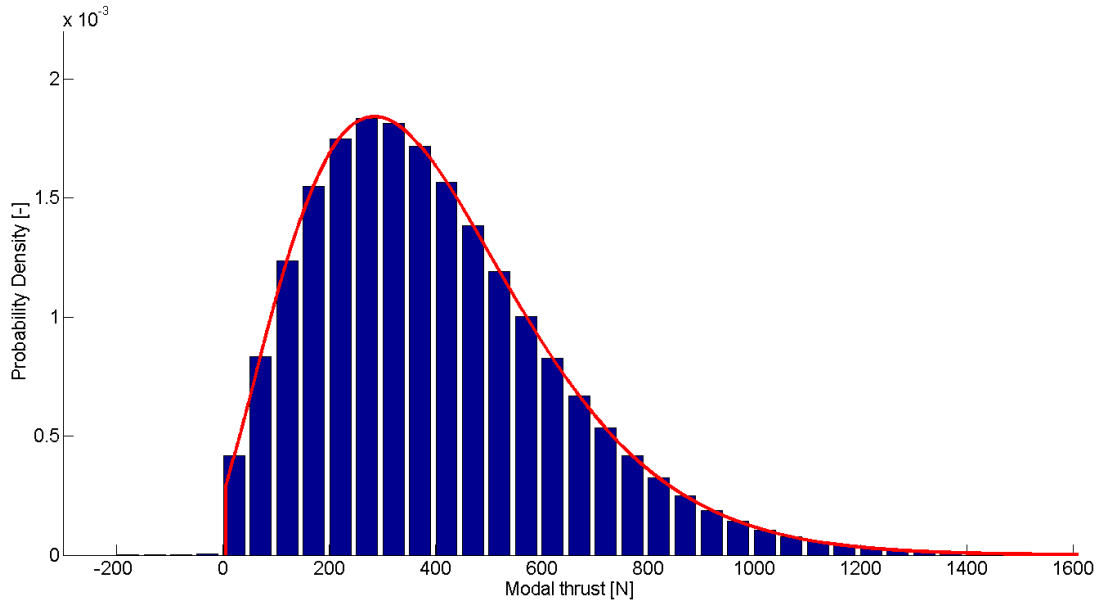


Figure 30: Histogram of simulated modal drag force realizations for the first mode subjected to load case 2. The analytical PDF of the modal force for $r_c > 0.83$ N is plotted in red.

As could be observed in 30 and 31, both the mean value and the variation of the modal force is larger for the second mode. This is not an unexpected result when considering the mode shapes presented under 4.1. Using results from section 4.1 and 4.2, these modal forces could be compared to the modal forces caused by rotor thrust for the same load case. The mean value of the modal force because of wind loads along the tower in mode 1 is 400 N, which is about 0.17 % of the mean modal force caused by the rotor thrust. For mode 2 the mean modal force is 620 N, or 1.52 % of the modal force caused by rotor thrust for this mode and load case. Based on these results, my neglect of wind loads acting along the tower in the simplified load model seems justifiable for this load case.

The results of the PDE response estimation for this nonlinear load situation is presented for the first mode in section 4.5. The second mode is not considered, as the small increase in modal force would still not make this mode influential on the tower top displacement response. It is shown in section 4.1 that the modal stiffness increase by about 2800 % from the first mode to the second, while the modal amplitude decrease from 100 % to about 17 % for the tower top.

The response probability is only evolved for white noise forcing, as the AR(1) implementation described earlier assumes that the noise term is independent of the previous force realization value, i.e. that $Z_i = Z_{i-1} = Z$. This is elaborated in sections 5 and 6. While it may be feasible to develop an autoregressive model where the noise term Z is a function of the previous force realization, this is beyond the scope of this thesis.

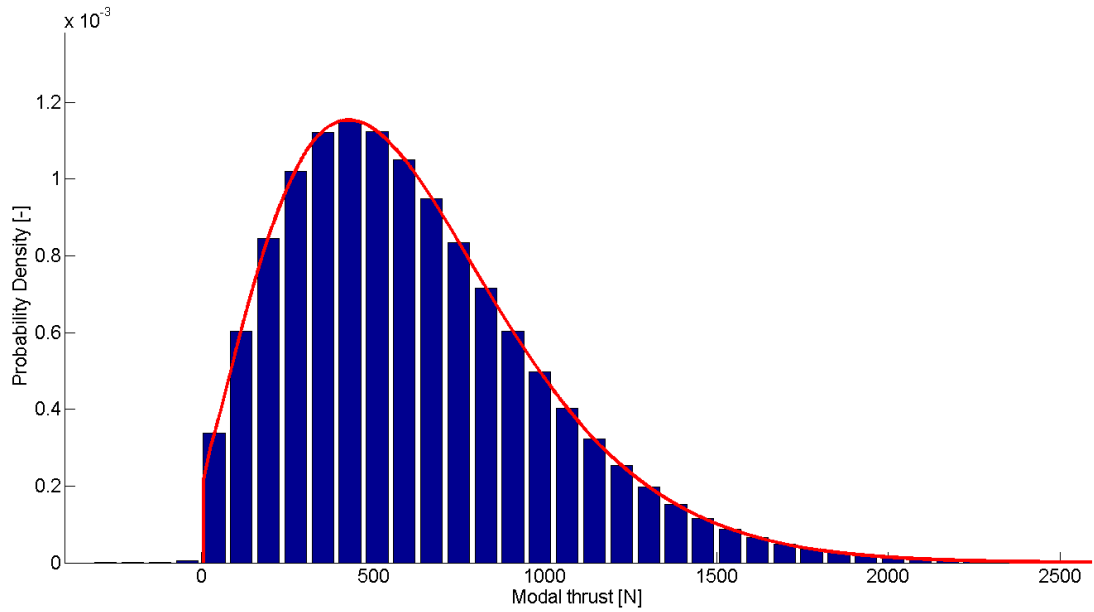


Figure 31: Histogram of simulated modal drag force realizations for the second mode subjected to load case 2. The analytical PDF of the modal force for $r_c > 1.89$ N is plotted in red.

4 Results

This section presents representative results obtained in the thesis. In some cases, the reader is referred to attachments for additional results.

4.1 Modal Properties of the Reference Tower

The modal results are constricted to the three first modes of the tower structure. This is because the structural response is mainly determined by the contribution from the first modes. In fact, for a flexible tower structure in bending, more than 90 % of the total energy dissipation normally occurs due to the fundamental mode. [22] Considerably lower accuracy could also be expected in the estimation of higher modes because of the tower discretization. Table 5 shows calculated normalized values for the three first mode shapes of the tower, obtained as described in section 3.1.1. The results from table 5 are plotted in figure 32.

Table 5: Table over the first three normalized tower mode shapes for transversal displacements.

DOF #, i	Height _i [m]	ϕ_1 [-]	ϕ_2 [-]	ϕ_3 [-]
1	4,38	0,002581	0,017106	-0,047480
2	8,76	0,010346	0,065261	-0,169198
3	13,14	0,023368	0,140087	-0,336621
4	17,52	0,041684	0,236862	-0,519303
5	21,90	0,065354	0,349236	-0,684589
6	26,28	0,094374	0,471154	-0,801382
7	30,66	0,128763	0,595504	-0,847173
8	35,04	0,168505	0,714944	-0,804511
9	39,42	0,213546	0,822188	-0,668746
10	43,80	0,263793	0,910117	-0,445564
11	48,18	0,319145	0,971278	-0,153362
12	52,56	0,379437	1,000000	0,171820
13	56,94	0,444461	0,990773	0,492172
14	61,32	0,513950	0,940215	0,765899
15	65,70	0,587578	0,846294	0,945728
16	70,08	0,664946	0,709208	1,000000
17	74,46	0,745528	0,532250	0,910783
18	78,84	0,828743	0,320790	0,678511
19	83,22	0,913861	0,083182	0,326928
20	87,60	1,000000	-0,168547	-0,091946

Table 6: Properties of the first three tower modes.

Mode #	1	2	3
Natural frequency, f_{Nat} [Hz]	0,3361	3,0723	9,1793
Modal mass [kg]	405574,9	142243,3	128105,5
Modal stiffness [N/m]	1809033,4	53005758,2	426131788,5
Modal damping [kg/s]	17131,2	54917,1	147769,8

Table 6 presents the natural frequencies of the relevant modes. It also shows the corresponding modal mass, stiffness and damping, as defined by equations (86) and (87).

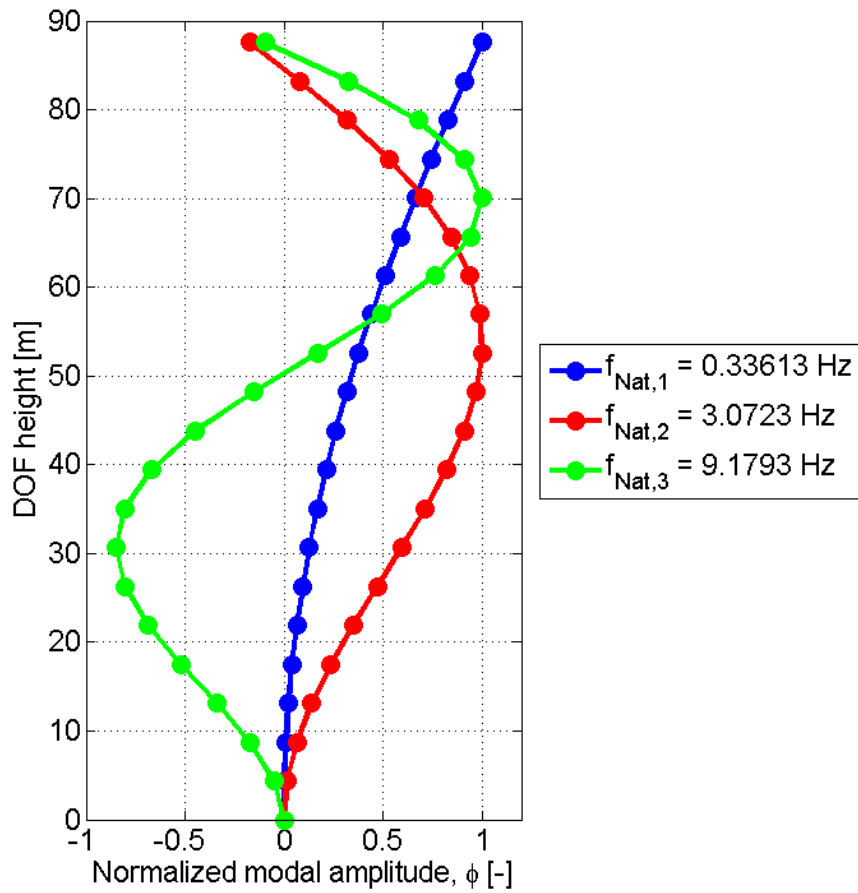


Figure 32: Plots of the first three normalized tower mode shapes for transversal displacements.

4.2 Simplified Thrust Force Model

Using 660 second time series simulated in FAST for all 15 load cases, where the initial 60 seconds are discarded to exclude transient effects, the parameters of the simplified rotor thrust force model proposed in section 3.5 have been estimated using the Matlab script in attachment 9. The proposed simplified load model describing the rotor thrust, T , is repeated below for convenience.

$$T(V_{ref}, v(t)) = \bar{T}(V_{ref}) + \hat{t}(V_{ref}, v(t))$$

where

$$\bar{T}(V_{ref}) = \bar{C}_s V_{ref}^\alpha$$

and

$$\hat{t}(V_{ref}, v(t)) = \hat{C}_s V_{ref}^\beta v(t)$$

The fitted parameters are presented in table 7 for two intervals of reference wind velocities. Gravity loads, movement of the structure and rotor blades is included in the FAST simulations, but the tower shadow effect is not. The time step used is $\Delta t = 0.0125$ s.

Table 7: Estimated simplified load model parameters.

Model Interval	2 m/s < V _{ref} < 11 m/s	11 m/s < V _{ref} < 30 m/s
α	1,2	-0,85
β	1,0	-1,35
C_s	40430 kg/m ^{0,25} s ^{0,8}	5515000 kg m ^{1,85} /s ^{2,85}
C_s	6007 kg/s	1682000 kg m ^{1,35} /s ^{2,35}

Figure 33 and 34 respectively show the fit of the simplified load model to the mean thrust, \bar{T} , and the turbulence proportionality constant, $\hat{C}_s V_{ref}^\beta$, estimated from the FAST simulations.

The former is fitted to the mean thrust from simulations using the least squared error method. The latter is estimated as a value that yields a model variance of the force fluctuations as close as possible to the simulated sample variance, if the same wind velocity time series are used. It is worth noting that the proposed load model assumes that the thrust fluctuations goes to zero when the mean wind velocity goes to zero. This does not seem to be entirely accurate, resulting in a relatively coarse variance fit for reference wind speeds below the cut-in velocity of 4 m/s.

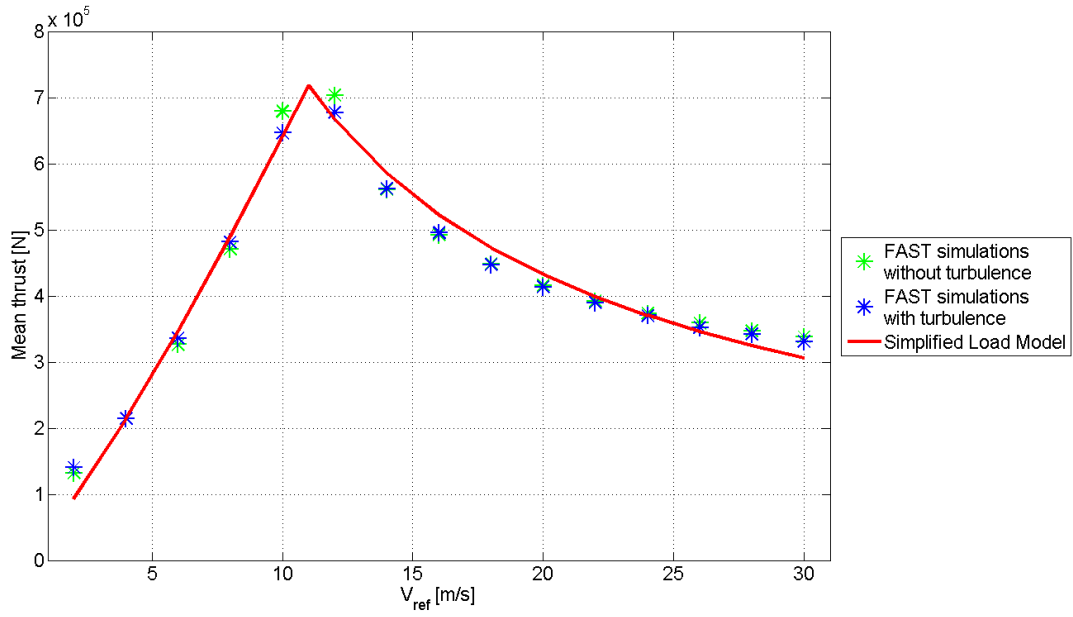


Figure 33: Plot showing the fit of the simplified load model with respect to the mean thrust value.

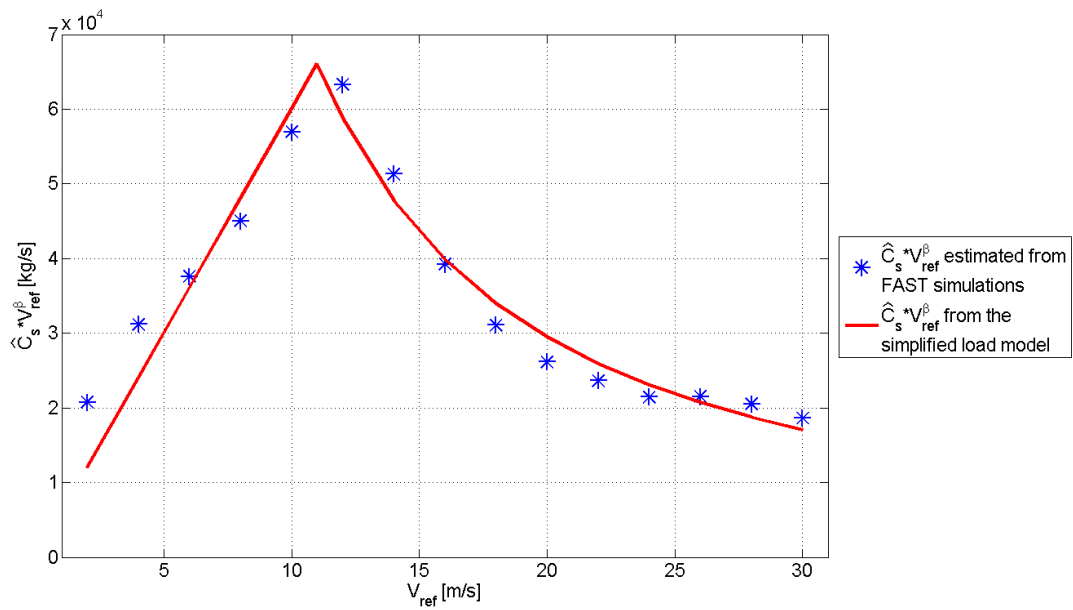


Figure 34: Plot showing the fit of the simplified load model with respect to the turbulence proportionality constant $\hat{C}_s V_{ref}^\beta$.

Figures 35 to 39 presents the time series of the FAST thrust simulations and the respective simplified thrust simulations obtained using the same wind velocity samples. The same figures also show the power spectral densities of these samples, and the analytical thrust fluctuation PSD for the simplified model with turbulence described as an AR(1) process.

In equation (115) and (116), the analytical autocovariance function of the thrust fluctuations, calculated from the simplified model using first order autoregressive turbulence with known autocovariance $C_v(\tau)$ and variance spectrum $S_v(\omega)$, is denoted $C_T(\tau)$. The corresponding variance spectrum is denoted $S_T(\omega)$.

The analytical autocovariance function of the rotor thrust fluctuations could now be calculated as:

$$\begin{aligned}
C_T(\tau) &= E[T(t)T(t + \tau)] \\
&= E[(\hat{C}_s V_{ref}^\beta v(t)) (\hat{C}_s V_{ref}^\beta v(t + \tau))] \\
&= (\hat{C}_s V_{ref}^\beta)^2 E[v(t)v(t + \tau)] \\
&= (\hat{C}_s V_{ref}^\beta)^2 C_v(\tau).
\end{aligned} \tag{115}$$

$S_T(\omega)$ from figures 35 to 41 could be found as shown in equation (116) by using $C_T(\tau)$ from equation (115) and the definition of a variance spectrum from equation (22).

$$\begin{aligned}
S_T(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} C_T(\tau) e^{-i\omega\tau} d\tau \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} (\hat{C}_s V_{ref}^\beta)^2 C_v(\tau) e^{-i\omega\tau} d\tau \\
&= (\hat{C}_s V_{ref}^\beta)^2 S_v(\omega)
\end{aligned} \tag{116}$$

Figure 40 and 41 show PSD sample estimates for load case 2 and 10 at low frequencies. In these figures, multiples of the rotor frequency is marked at $3P = 0.37$ Hz and $6P = 0.73$ Hz for load case 2, and $3P = 0.61$ Hz and $6P = 1.21$ Hz for load case 10.

Figures 42 to 45 presents 100 second autocovariance plots of the thrust force fluctuations for the four load cases 2, 4, 8 and 12. These figures show that the correlation in the thrust behaves quite differently for different load cases, something that is also supported by the plotted variance spectra.

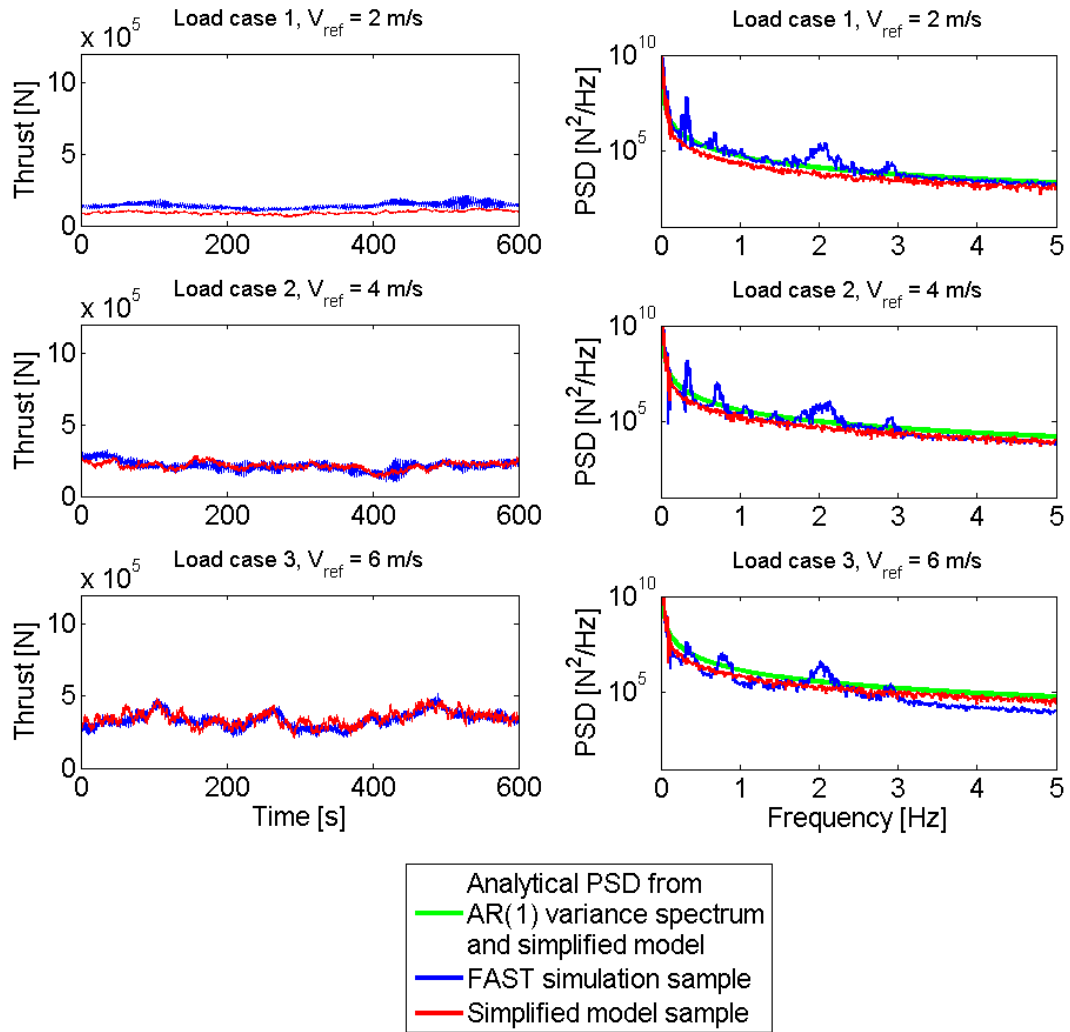


Figure 35: Figure showing, for load cases 1-3, time series of the rotor thrust for FAST simulations and simplified model simulations using the same wind velocity sample to the left, and the respective power spectral densities of these simulations to the right. In the variance spectra plots, analytical PSDs of the thrust fluctuations calculated using the AR(1) thrust spectra defined by the parameters presented in section 4.3, is also included.

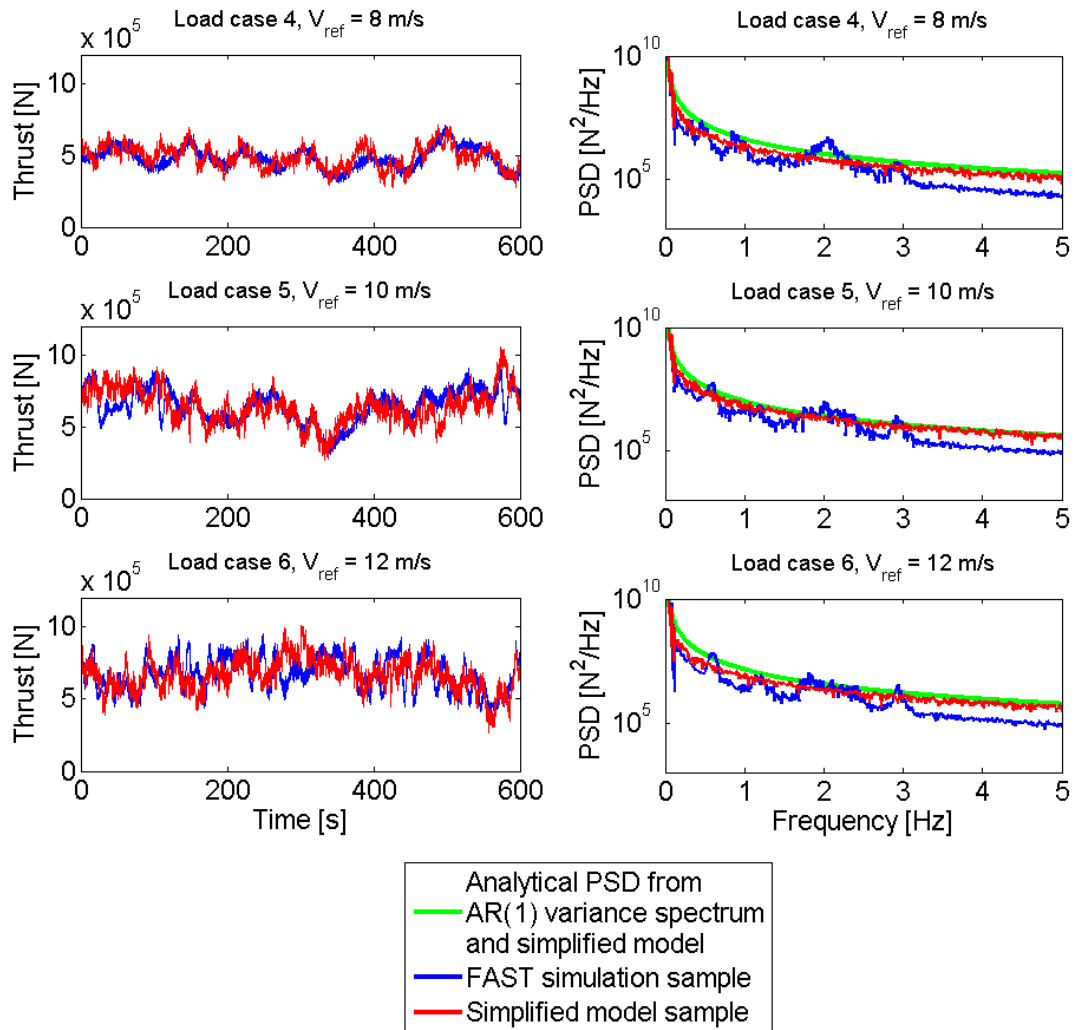


Figure 36: Figure showing, for load cases 4-6, time series of the rotor thrust for FAST simulations and simplified model simulations using the same wind velocity sample to the left, and the respective power spectral densities of these simulations to the right. In the variance spectra plots, analytical PSDs of the thrust fluctuations calculated using the AR(1) thrust spectra defined by the parameters presented in section 4.3, is also included.

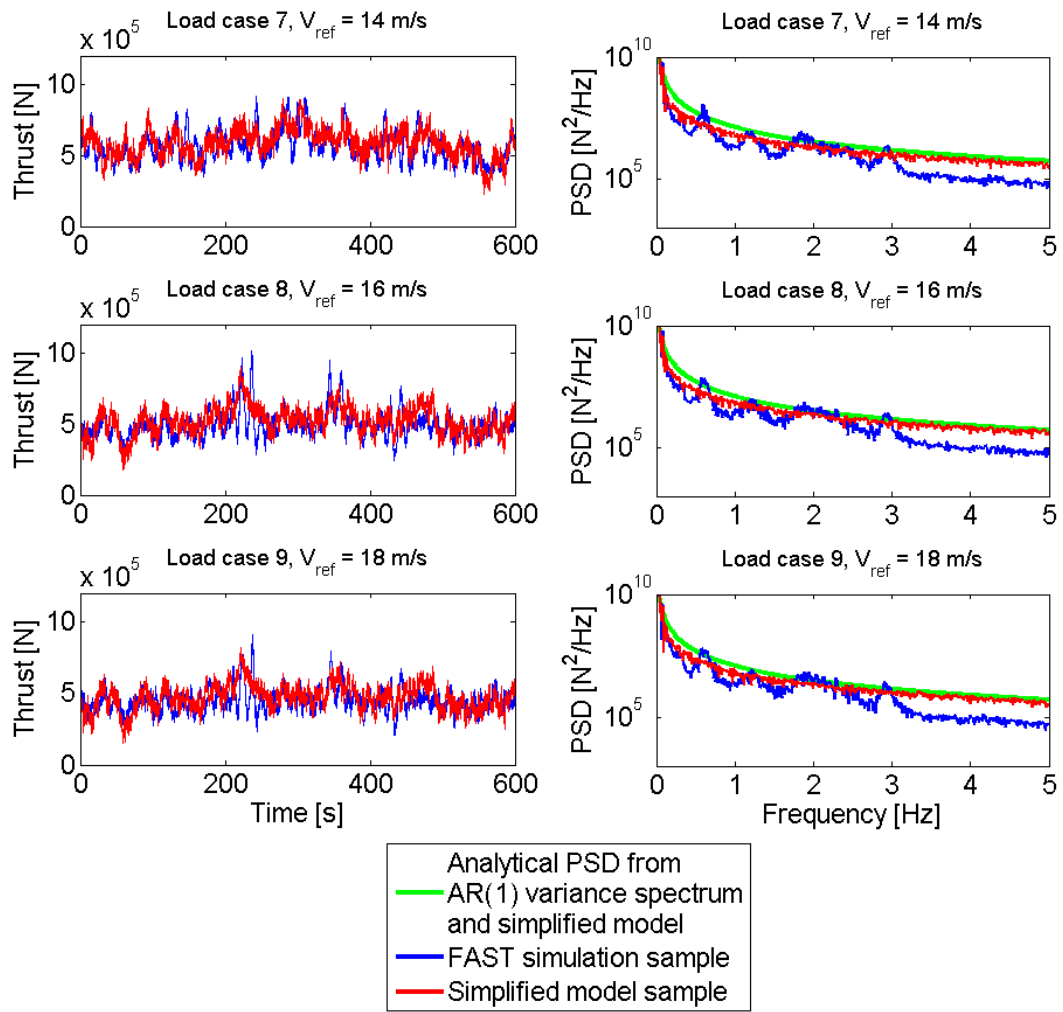


Figure 37: Figure showing, for load cases 7-9, time series of the rotor thrust for FAST simulations and simplified model simulations using the same wind velocity sample to the left, and the respective power spectral densities of these simulations to the right. In the variance spectra plots, analytical PSDs of the thrust fluctuations calculated using the AR(1) thrust spectra defined by the parameters presented in section 4.3, is also included.

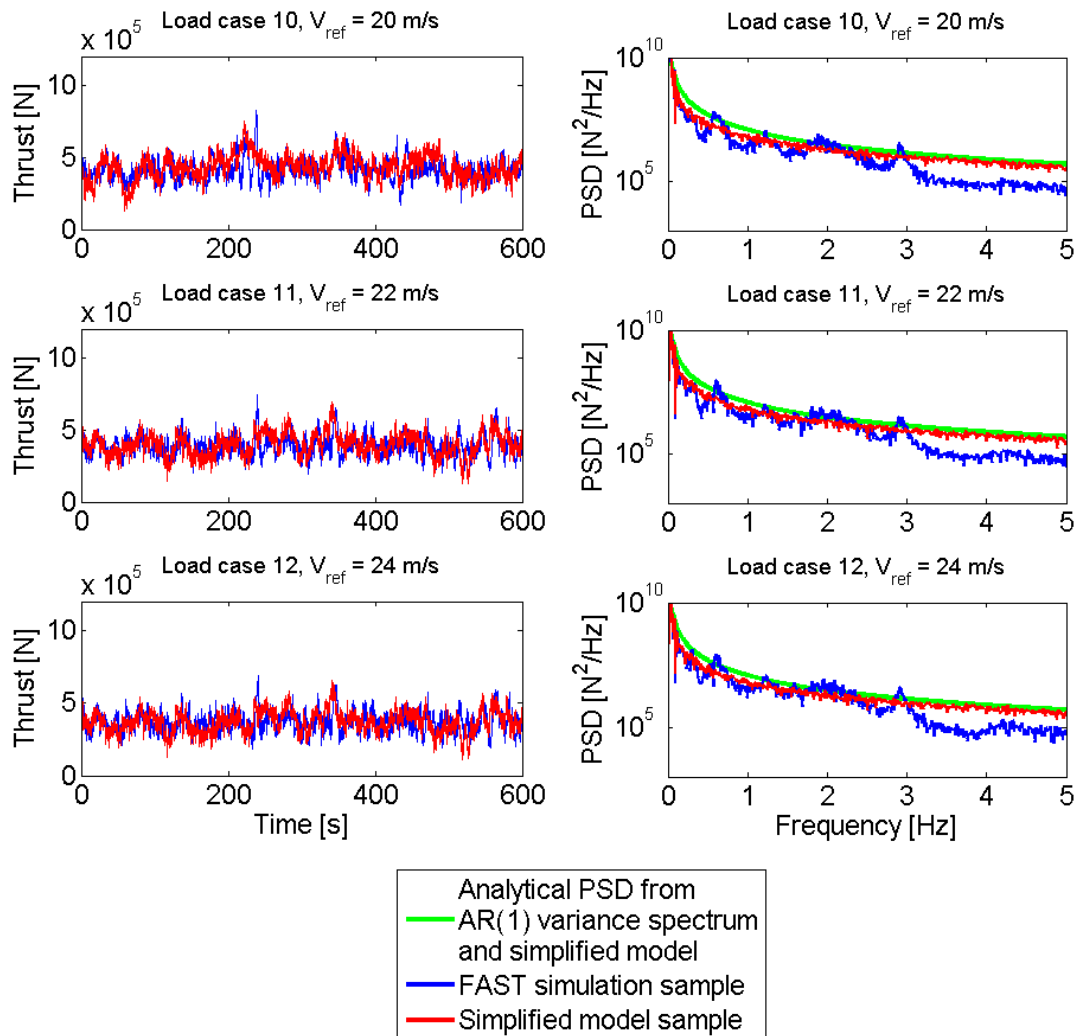


Figure 38: Figure showing, for load cases 10-12, time series of the rotor thrust for FAST simulations and simplified model simulations using the same wind velocity sample to the left, and the respective power spectral densities of these simulations to the right. In the variance spectra plots, analytical PSDs of the thrust fluctuations calculated using the AR(1) thrust spectra defined by the parameters presented in section 4.3, is also included.

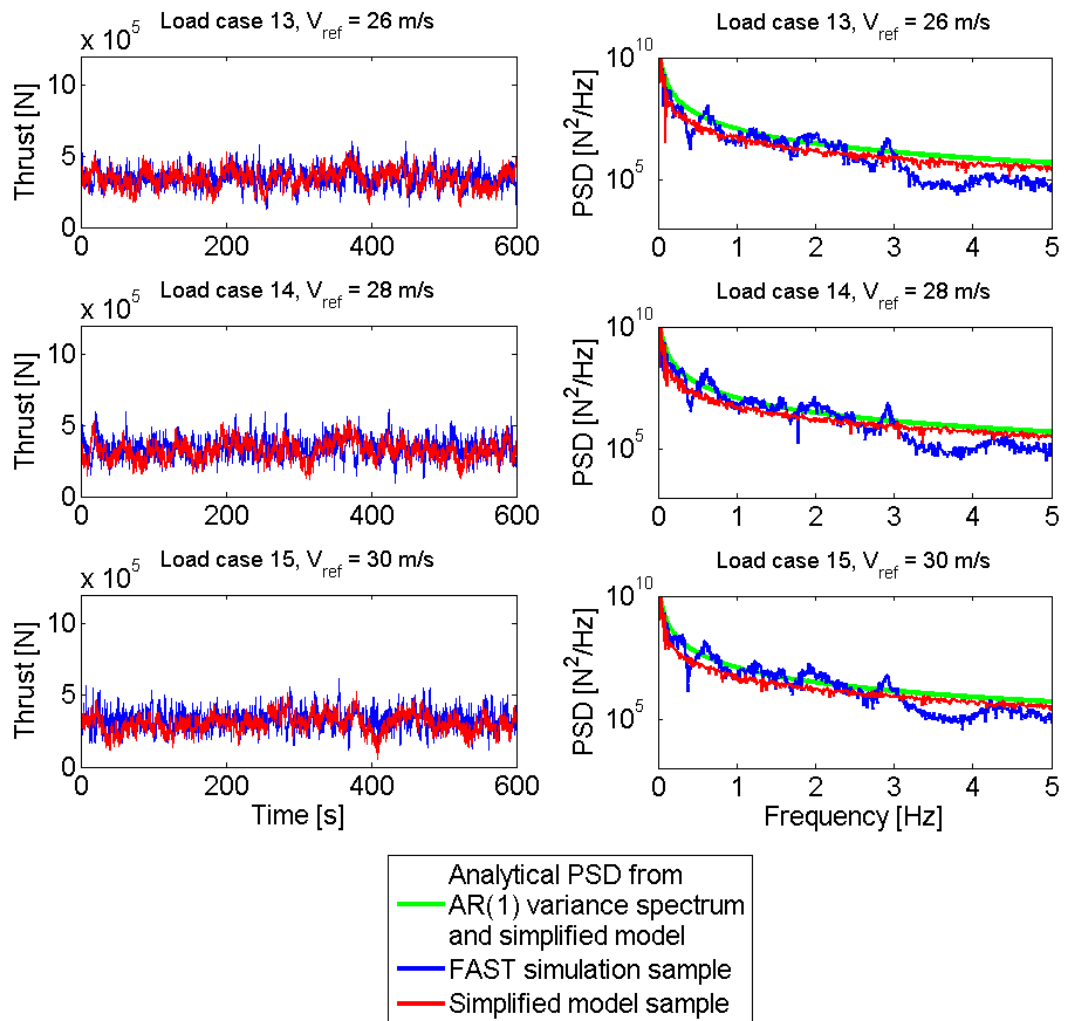


Figure 39: Figure showing, for load cases 13-15, time series of the rotor thrust for FAST simulations and simplified model simulations using the same wind velocity sample to the left, and the respective power spectral densities of these simulations to the right. In the variance spectra plots, analytical PSDs of the thrust fluctuations calculated using the AR(1) thrust spectra defined by the parameters presented in section 4.3, is also included.

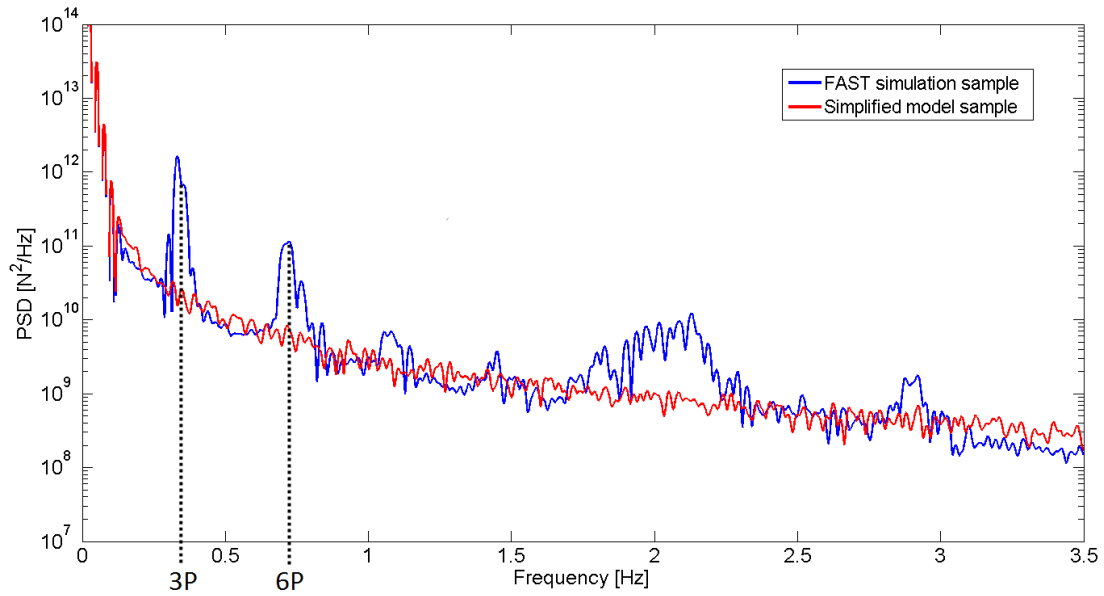


Figure 40: Sample PSD plots at low frequencies for load case 2, with marked multiples of the rotor frequency, 1P.

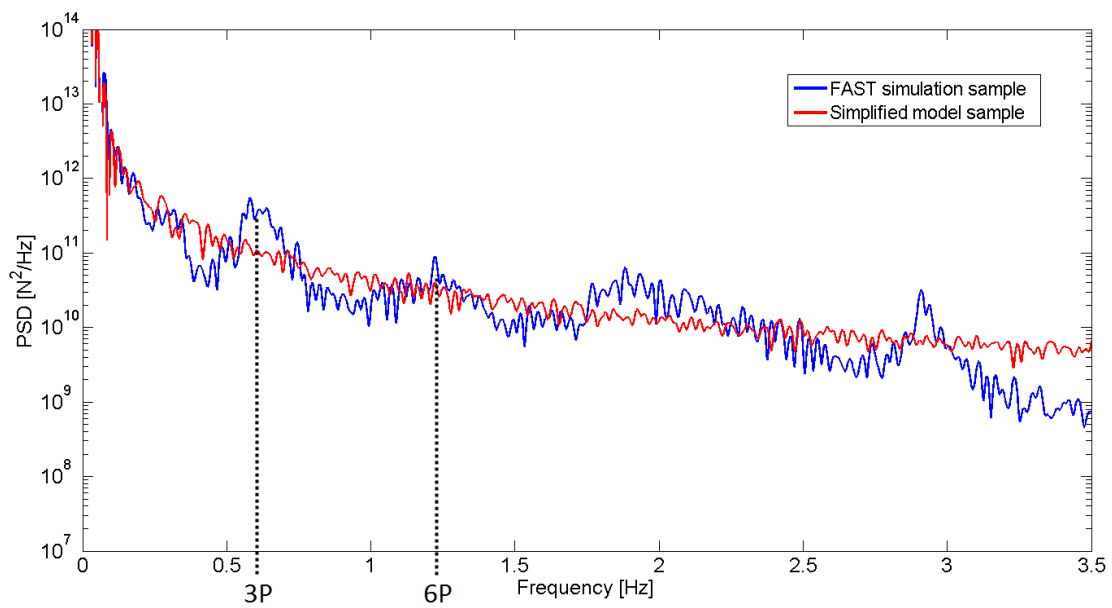


Figure 41: Sample PSD plots at low frequencies for load case 10, with marked multiples of the rotor frequency, 1P.

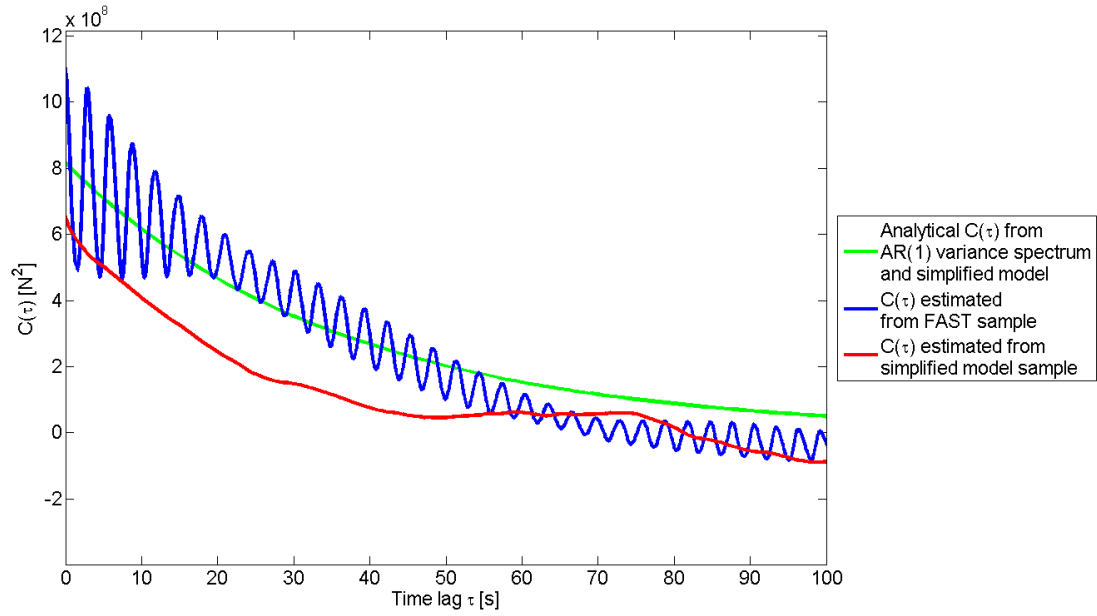


Figure 42: 100 second autocovariances of the different thrust force fluctuations for load case 2, $V_{ref} = 4$ m/s.

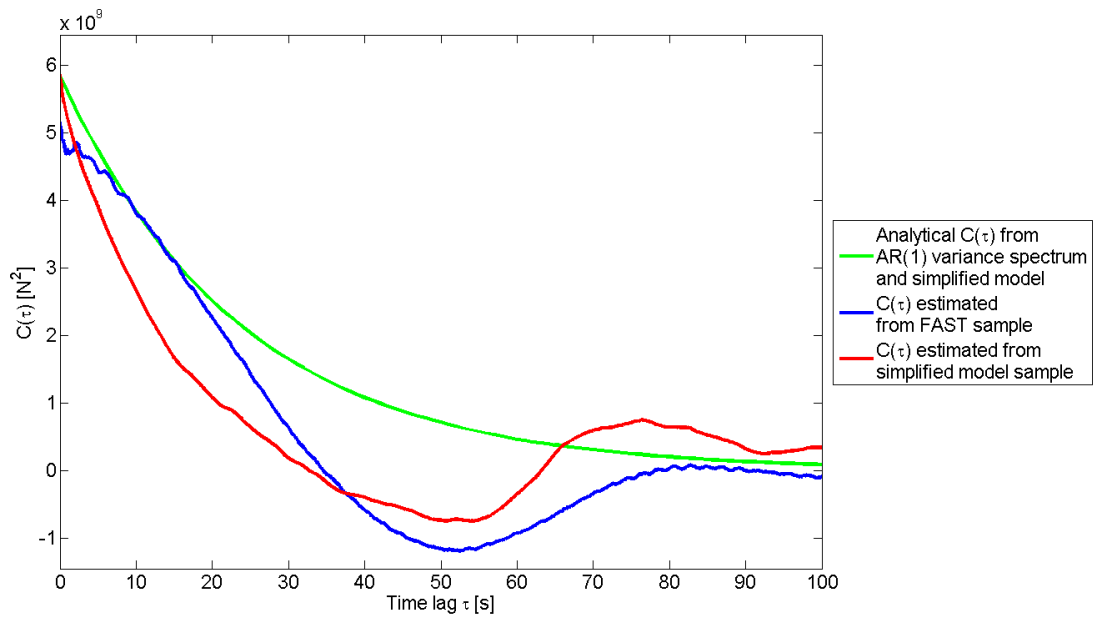


Figure 43: 100 second autocovariances of the different thrust force fluctuations for load case 4, $V_{ref} = 8$ m/s.

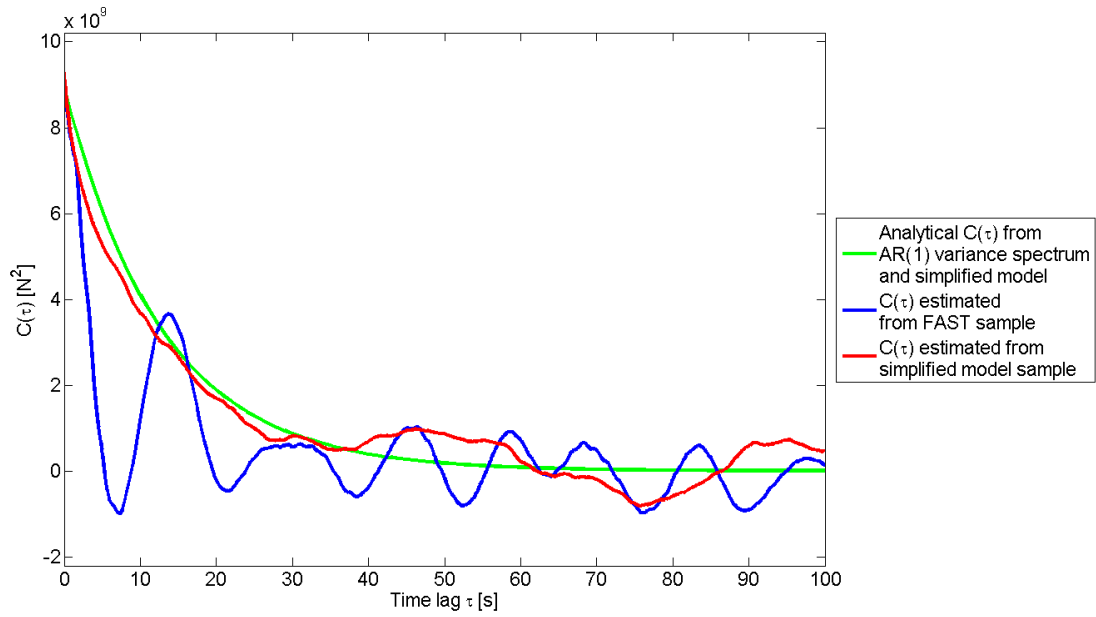


Figure 44: 100 second autocovariances of the different thrust force fluctuations for load case 8, $V_{ref} = 16$ m/s.

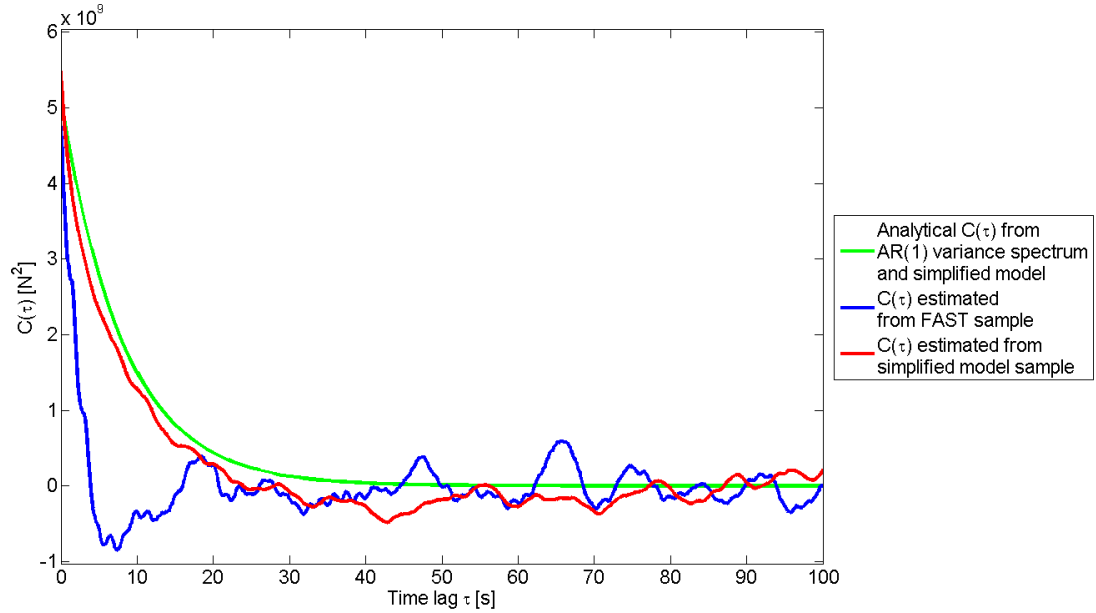


Figure 45: 100 second autocovariances of the different thrust force fluctuations for load case 12, $V_{ref} = 24$ m/s.

4.3 Estimated First Order Autoregression Parameters

4.3.1 AR(1) Parameters Describing Turbulence

In this thesis I utilize autoregression in order to incorporate frequency dependent forcing into the probability density evolution method. Table 8 presents the estimated first order autoregression parameters describing turbulence for sampling time steps of $\Delta t = 0.0125$ s and $\Delta t = 0.05$ s. This is done for all 15 load cases defined in section 3.4.

The parameters have been estimated by numerically curve fitting the different AR(1) spectra to Kaimal spectra using the Matlab script from attachment 5. The columns named σ_{tot} shows the total standard deviations of the autoregressive models found by integrating over the AR spectra.

Plots of the estimated AR(1) spectra and their respective underlying Kaimal spectra are shown in attachment 10, figure 60 to 63.

Table 8: Table showing estimated AR(1) parameters describing turbulence in the 15 load cases for time steps of $\Delta t = 0.0125$ s and $\Delta t = 0.05$ s.

Load Case	$\Delta t = 0.0125$ s			$\Delta t = 0.05$ s		
	AR(1) ϕ_v [-]	AR(1) σ_v [m/s]	σ_{tot} [m/s]	AR(1) ϕ_v [-]	AR(1) σ_v [m/s]	σ_{tot} [m/s]
1	0,99973	0,02310	0,994	0,99892	0,04616	0,994
2	0,99965	0,03181	1,204	0,99861	0,06355	1,204
3	0,99957	0,04170	1,414	0,99826	0,08333	1,414
4	0,99947	0,05277	1,624	0,99789	0,10542	1,624
5	0,99937	0,06499	1,834	0,99749	0,12981	1,834
6	0,99927	0,07835	2,044	0,99707	0,15649	2,044
7	0,99915	0,09286	2,254	0,99661	0,18549	2,254
8	0,99903	0,10856	2,464	0,99612	0,21679	2,464
9	0,99890	0,12539	2,674	0,99561	0,25043	2,674
10	0,99876	0,14346	2,884	0,99506	0,28643	2,884
11	0,99862	0,16272	3,094	0,99448	0,32476	3,094
12	0,99846	0,18317	3,304	0,99386	0,36554	3,304
13	0,99830	0,20487	3,514	0,99321	0,40874	3,514
14	0,99813	0,22782	3,724	0,99253	0,45442	3,724
15	0,99795	0,25202	3,934	0,99181	0,50261	3,934

4.3.2 AR(1) Parameters Describing Simplified Thrust Fluctuations

Given the autoregression parameters of the wind turbulence, $v(t)$, the corresponding parameters of the thrust fluctuations described by the simplified model could be easily derived since the rotor thrust depends linearly on turbulence.

Let ϕ_v and σ_v be the AR(1) properties for turbulence given in table 8. Wind turbulence would now be estimated as shown in equation (117).

$$v_t = \phi_v v_{t-1} + Z_{v,t} \quad (117)$$

In equation (117), $Z_{v,t}$ is white noise Gaussian distributed with mean 0 m/s and standard deviation σ_v . Using equation (97) the AR(1) estimated thrust fluctuations are found to be described by

$$\begin{aligned} \hat{t}_t &= \hat{C}_s V_{ref}^\beta v_t \\ &= \hat{C}_s V_{ref}^\beta (\phi_v v_{t-1} + Z_{v,t}) \\ &= \phi_v \hat{t}_{t-1} + \hat{C}_s V_{ref}^\beta Z_{v,t} \\ &= \phi_{\hat{t}} \hat{t}_{t-1} + Z_{\hat{t},t}. \end{aligned} \quad (118)$$

This leads to the result that $\phi_{\hat{t}} = \phi_v$, and that $Z_{\hat{t},t}$ is white noise Gaussian distributed with mean 0 N and standard deviation $\sigma_{\hat{t}} = \hat{C}_s V_{ref}^\beta \sigma_v$. Values of these parameters for time steps of $\Delta t = 0.0125$ s and $\Delta t = 0.05$ s are shown in table 9.

The total standard deviation of the AR(1) described thrust fluctuation process is also calculated using equation (68) as $\sigma_{tot,AR} = \sqrt{\sigma_{\hat{t}}^2 / (1 - \phi_{\hat{t}}^2)}$, and presented together with the sample standard deviation from the FAST simulations, $\sqrt{S_{2tot,FAST}}$.

Table 9: Table showing estimated AR(1) parameters describing thrust fluctuations in the 15 load cases for time steps of $\Delta t = 0.0125$ s and $\Delta t = 0.05$ s.

	$\Delta t = 0.0125$ s	$\Delta t = 0.05$ s			
Load Case	AR(1) $\sigma_{\hat{t}}$ [N]	AR(1) $\sigma_{\hat{t}}$ [N]	$\sigma_{tot,AR}$ [N]	$\sqrt{S_{2tot,FAST}}$ [N]	$\sigma_{tot,AR} / \sqrt{S_{2tot,FAST}}$ [%]
1	277,5	554,6	11935,6	19312,2	161,8
2	764,3	1527,0	28970,9	33230,1	114,7
3	1503,0	3003,4	50934,2	51012,1	100,1
4	2535,9	5066,1	78026,8	71836,4	92,1
5	3903,9	7797,7	110125,2	115013,8	104,4
6	4602,3	9192,2	120167,6	116632,8	97,1
7	4429,8	8848,6	107554,2	105059,6	97,7
8	4324,5	8635,8	98128,3	95071,6	96,9
9	4260,6	8509,3	90912,7	81785,2	90,0
10	4228,3	8442,2	85037,9	73849,1	86,8
11	4216,9	8416,2	80210,9	70570,5	88,0
12	4220,8	8423,2	76128,0	69060,1	90,7
13	4237,3	8453,9	72668,8	65604,6	90,3
14	4263,4	8503,9	69703,9	71330,1	102,3
15	4296,8	8569,3	67093,0	68169,6	101,6

4.4 Response of Tower Top to Rotor Thrust

This section presents results obtained for the tower top response to thrust from the wind turbine rotor. Finite element MCS results are presented as estimated PDFs using kernel smoothing on 1800 second response time series with a time step of 0.0125 seconds, resulting in 144000 sample points. The estimated response of the structure have also been calculated for the two first modes, using theory presented in section 2.1. In these calculations, the analytical Kaimal spectra have been used to describe wind velocity turbulence, while the thrust force are described using the simplified load model from section 4.2. The modal structure properties used are those presented in 4.1.

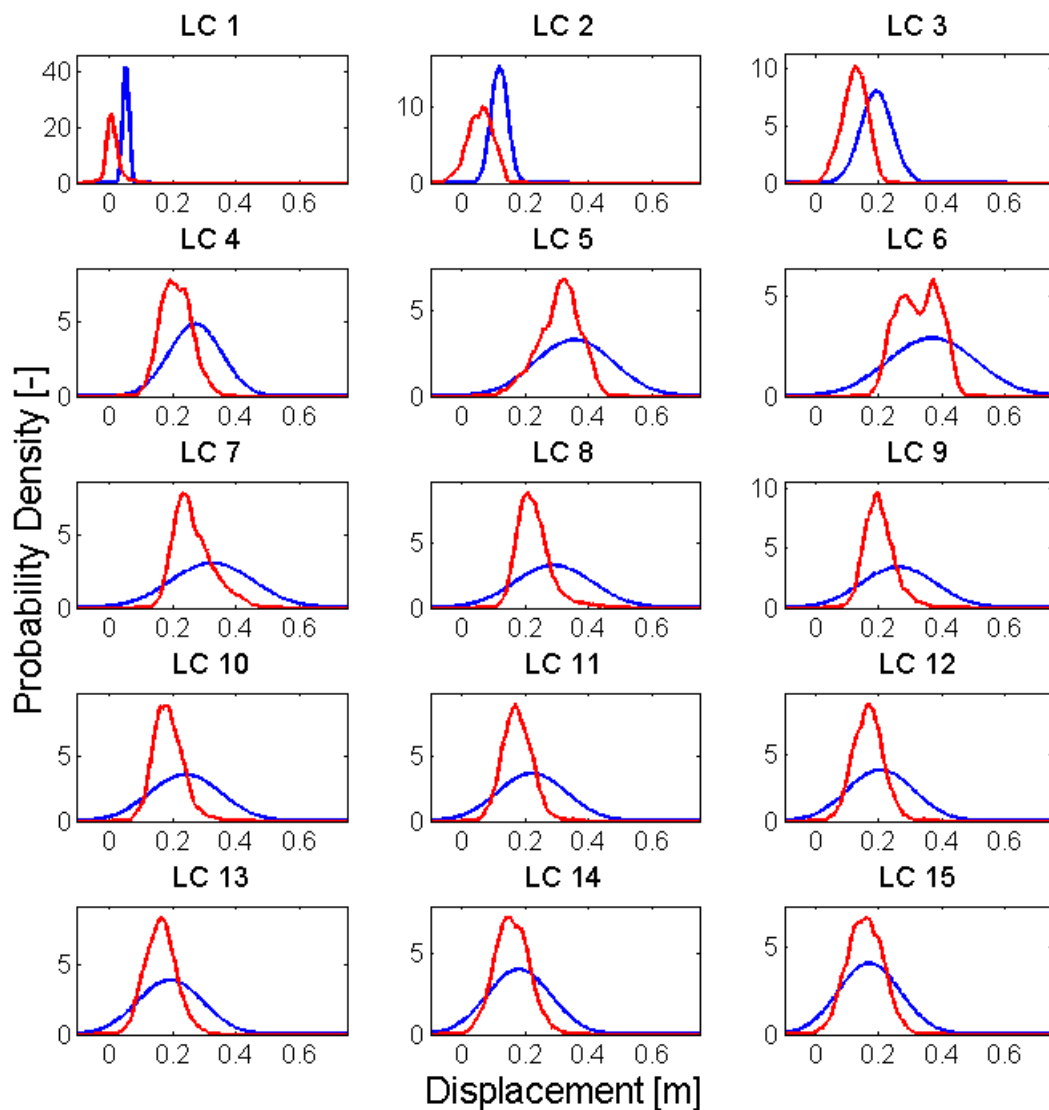


Figure 46: Tower top displacement for the 15 load cases estimated from time domain simulations in FAST (shown in red), and calculated analytically for the first mode using the simplified load model (shown in blue).

Figure 46 and 64 respectively show the total tower top displacement and velocity response from FAST, and the first mode responses estimated using the simplified load model. Figure 64 could be found in attachment 11.

Table 10 show statistical properties of the estimated displacement responses. S_1 is the sample mean of the 1800 second FAST simulations, while $\sqrt{S_2}$ is the sample standard deviation of the same simulations. μ_{u1} is the analytical mean value of the first mode response, calculated using the simplified load model:

$$\mu_{u1} = \phi_{1,20} \cdot \frac{\bar{C}_s V_{ref}^\alpha}{\tilde{K}_1}. \quad (119)$$

σ_{u1} is the analytical standard deviation of the first mode response to excitation by the simplified load model:

$$\begin{aligned} \sigma_{u1} &= \sqrt{\phi_{1,20}^2 \cdot \int_0^\infty |H(\omega)|^2 S_T^+(\omega) d\omega} \\ &= \phi_{1,20} \cdot \hat{C}_s V_{ref}^\beta \sqrt{\int_0^\infty |H(\omega)|^2 S_v^+(\omega) d\omega}. \end{aligned} \quad (120)$$

Figure 47 is a plot of the ratio between S_1 and μ_{u1} , and $\sqrt{S_2}$ and σ_{u1} , for all 15 load cases.

Table 10: Table presenting statistical properties of estimated tower top displacements.

Load Case	S_1 [m]	$\sqrt{S_2}$ [m]	μ_{u1} [m]	σ_{u1} [m]	S_1/μ_{u1} [%]	$\sqrt{S_2}/\sigma_{u1}$ [%]
1	0,0102	0,0223	0,0513	0,0095	19,8	236,2
2	0,0572	0,0398	0,1180	0,0261	48,5	152,8
3	0,1225	0,0394	0,1919	0,0500	63,8	78,8
4	0,2103	0,0475	0,2710	0,0818	77,6	58,0
5	0,3072	0,0646	0,3542	0,1218	86,7	53,1
6	0,3257	0,0642	0,3688	0,1387	88,3	46,3
7	0,2647	0,0614	0,3235	0,1291	81,8	47,6
8	0,2268	0,0544	0,2888	0,1218	78,5	44,7
9	0,2025	0,0496	0,2613	0,1162	77,5	42,7
10	0,1859	0,0475	0,2389	0,1116	77,8	42,5
11	0,1746	0,0466	0,2203	0,1079	79,2	43,2
12	0,1666	0,0478	0,2046	0,1047	81,4	45,6
13	0,1613	0,0488	0,1911	0,1020	84,4	47,8
14	0,1576	0,0544	0,1795	0,0997	87,8	54,5
15	0,1555	0,0567	0,1693	0,0976	91,9	58,1

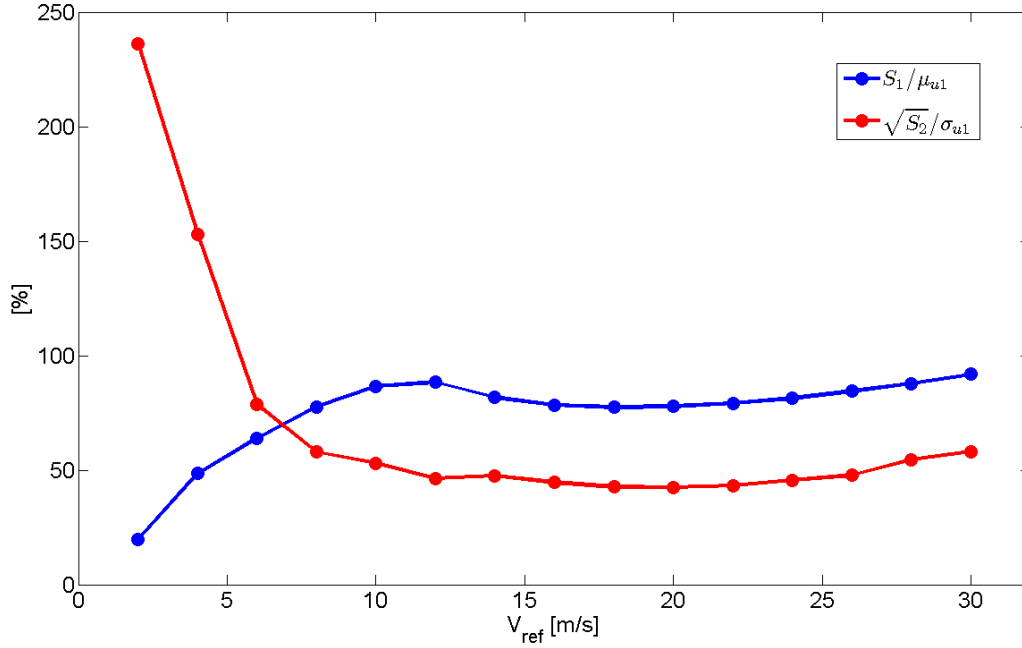


Figure 47: Plots showing the two ratios from table 10.

Figures 48, 49, 50 and 52 respectively show enlarged plots of the estimated displacement distributions for load case 2, 6, 4 and 10. These figures include plots of response distributions found using probability density evolution with white noise excitation. Figure 50 and 52 also present distributions found using the probability density evolution with the previous force realization incorporated using first order autoregression.

Figure 51 and 53 show plots of the analytical model force spectrum, the corresponding fitted AR(1) spectrum and the constant white noise spectrum near the natural frequency of the first mode for load case 4 and 10. These spectra are plotted with the squared frequency response function of the first mode using a logarithmic y-axis.

Figure 54 and 55 presents joint probability density plots of the tower top displacement and velocity response. Figure 54 is the MCS results from FAST simulations, while the results in figure 55 is found analytically for the first mode using the simplified load model and Kaimal turbulence spectra directly.

Figure 56 show the estimated displacement response distributions for the second mode, calculated analytically using the simplified load model and Kaimal turbulence spectra.

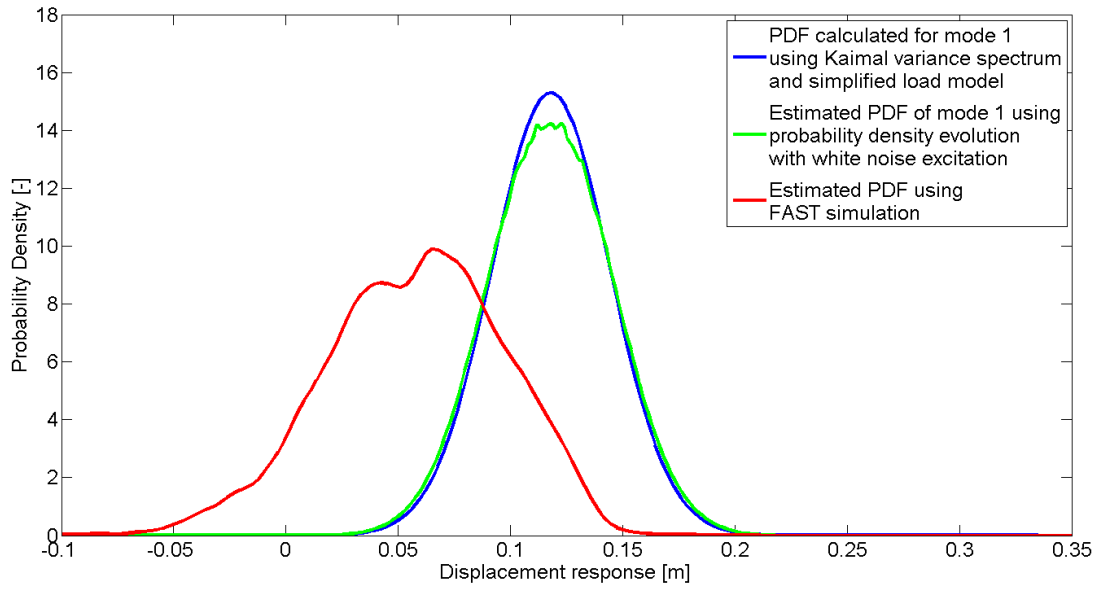


Figure 48: Probability distributions of tower top displacement for load case 2.

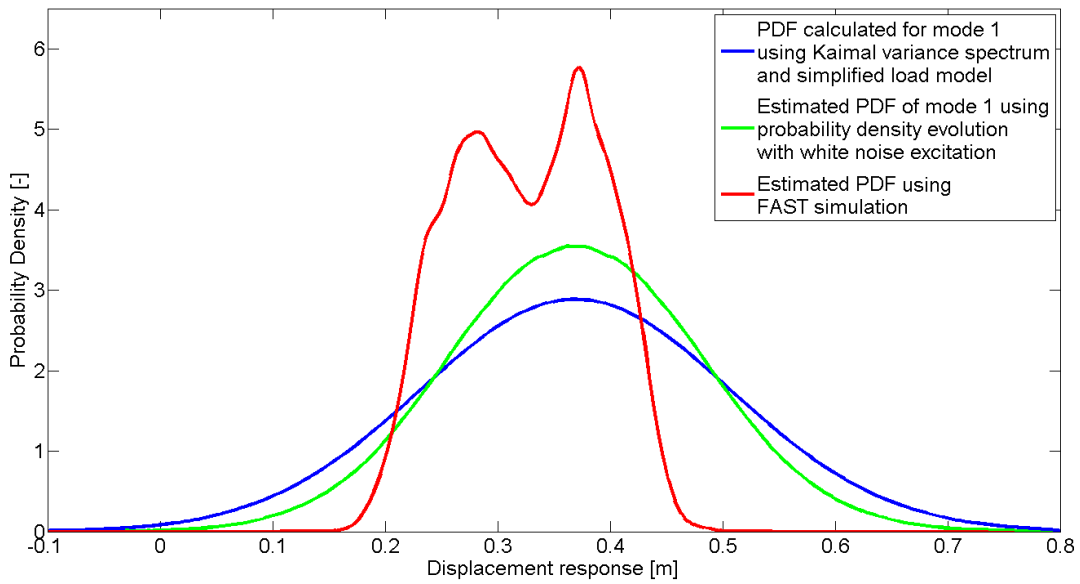


Figure 49: Probability distributions of tower top displacement for load case 6.

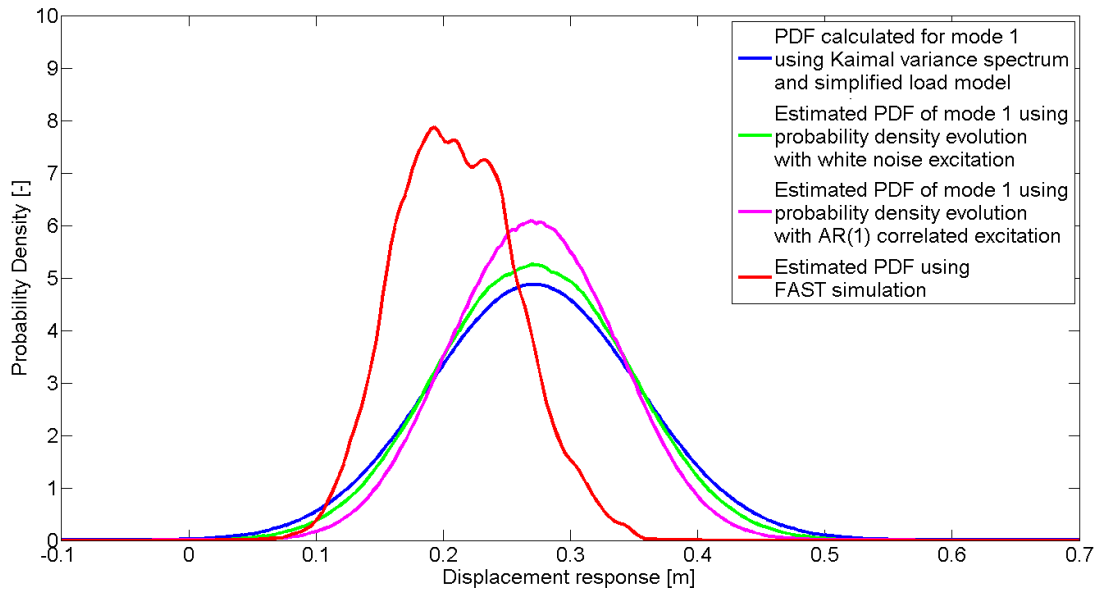


Figure 50: Probability distributions of tower top displacement for load case 4.

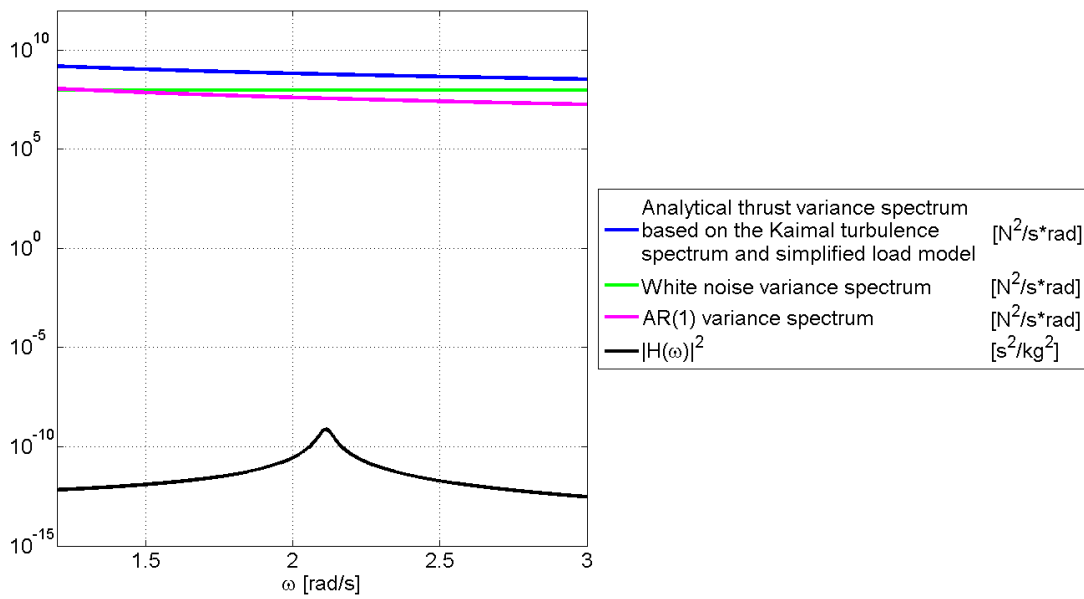


Figure 51: Semi-logarithmic plots of the thrust variance spectra near the natural frequency of mode one for load case 4. Plotted with the transfer function of the modal coordinate response.

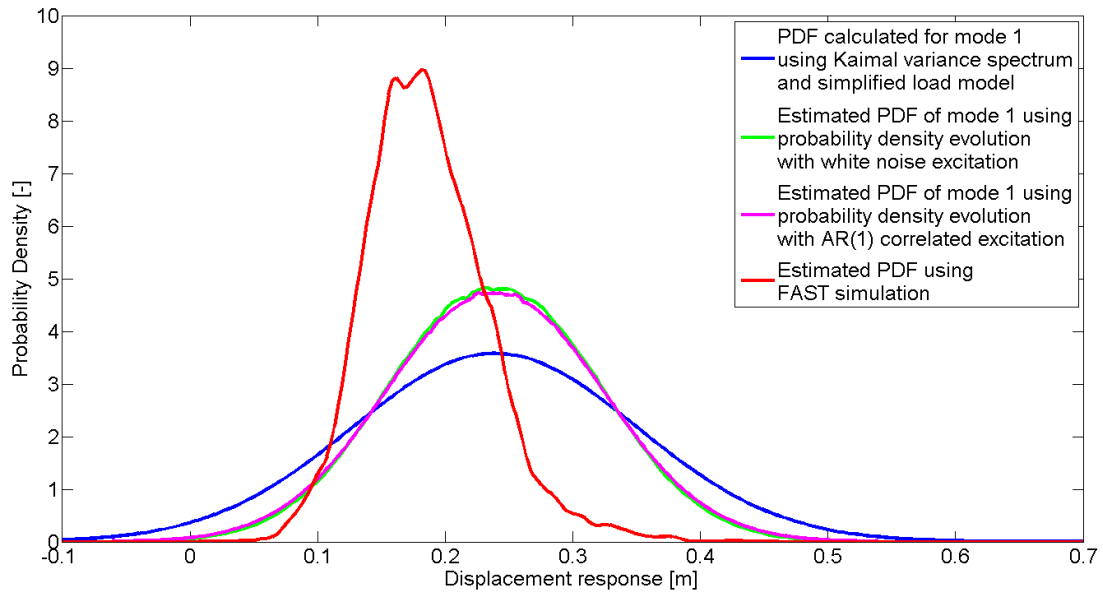


Figure 52: Probability distributions of tower top displacement for load case 10.

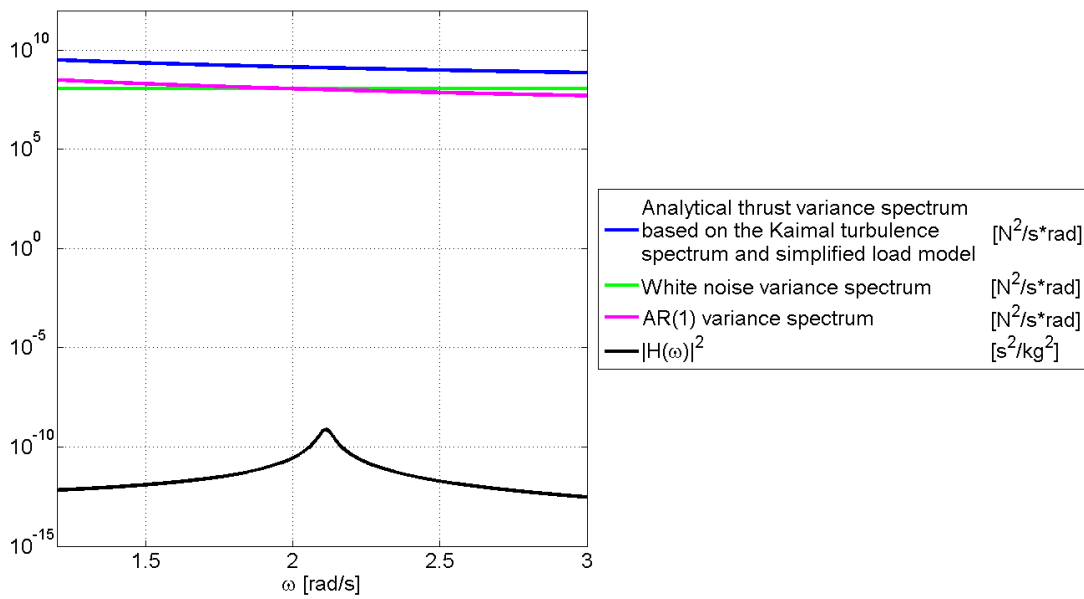


Figure 53: Semi-logarithmic plots of the thrust variance spectra near the natural frequency of mode one for load case 10. Plotted with the transfer function of the modal coordinate response.

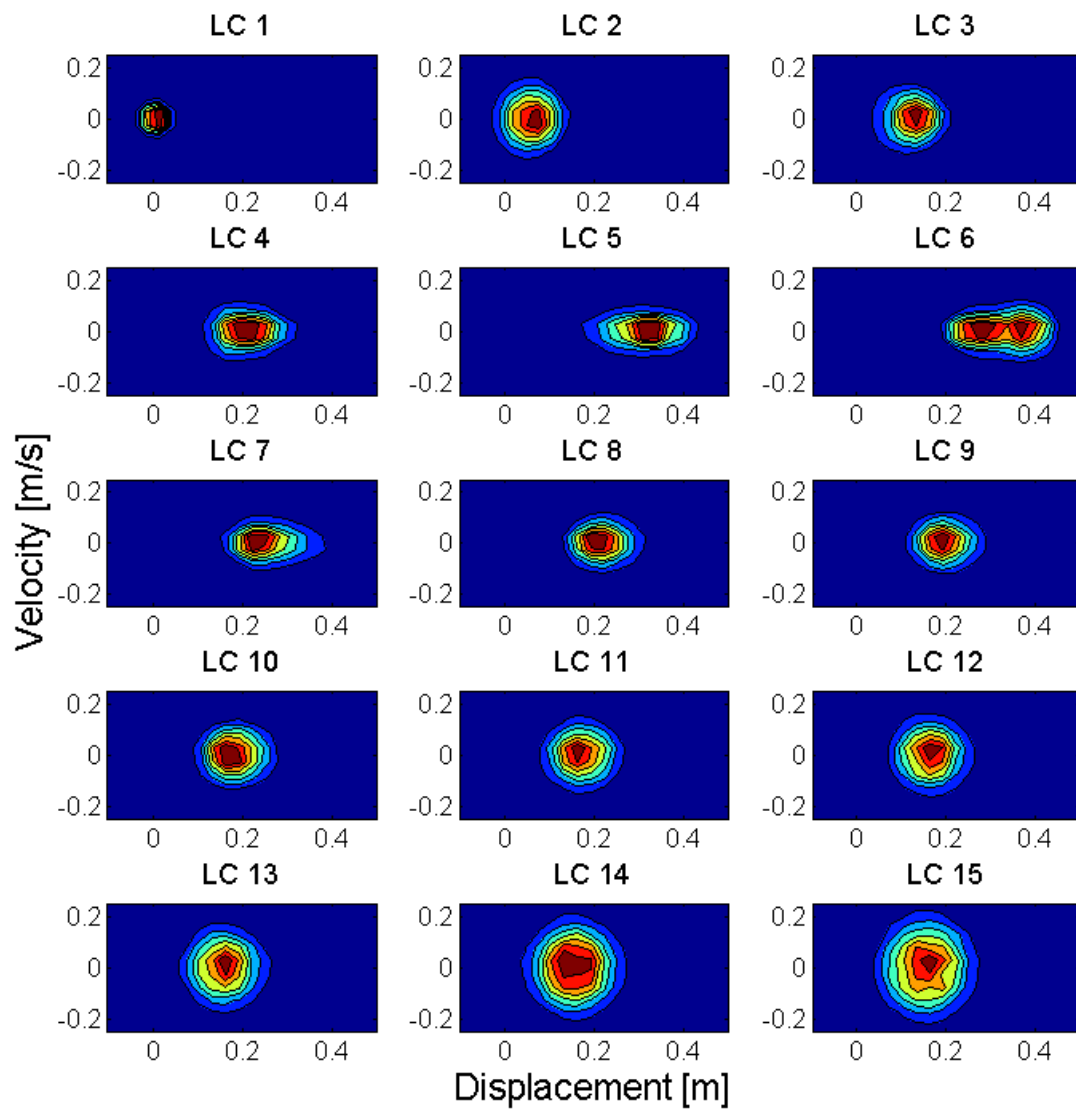


Figure 54: Contour plot showing the joint probability density of the displacement and velocity response, estimated using 1800 second FAST simulations.

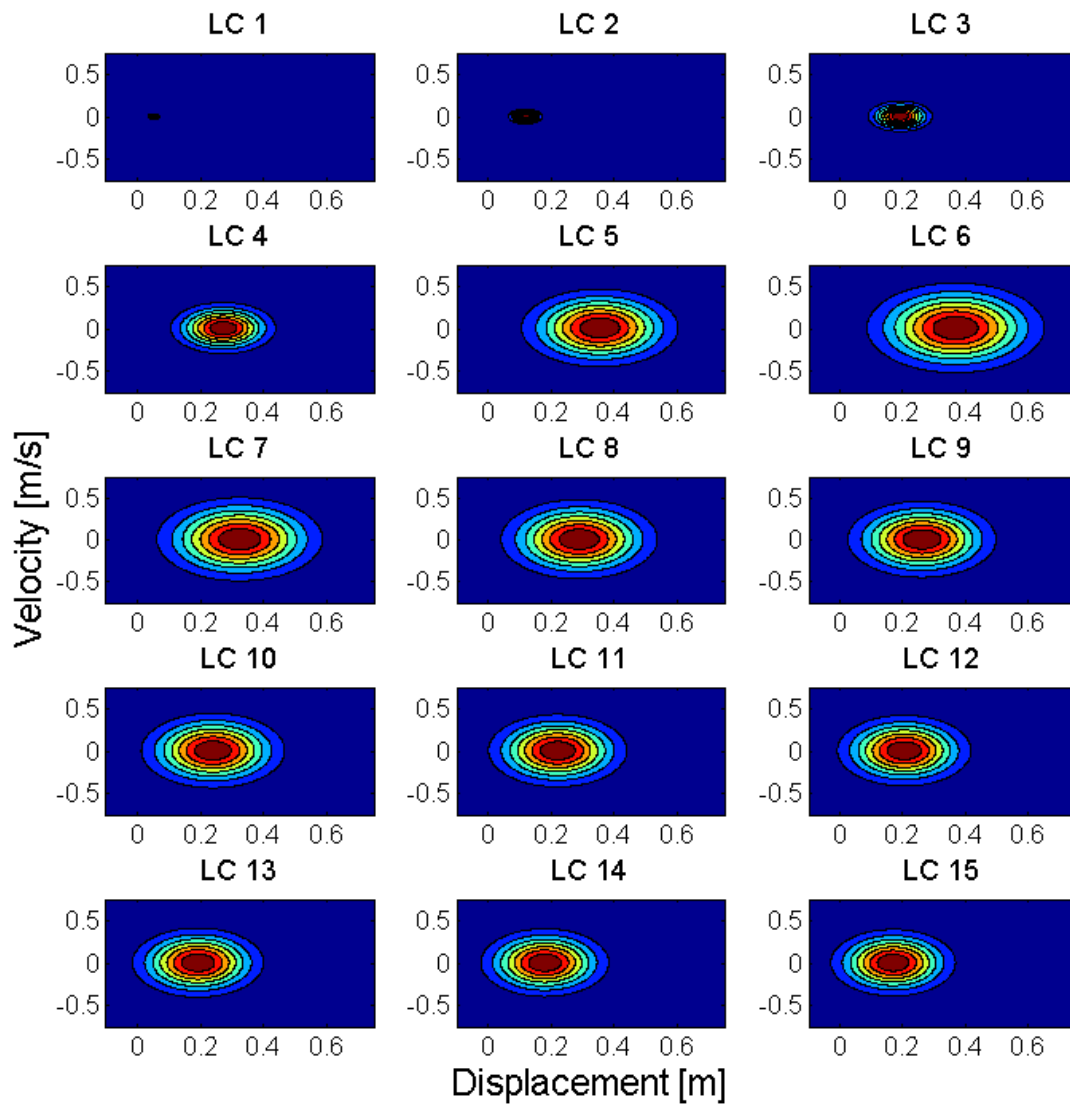


Figure 55: Contour plot showing the joint probability density of the displacement and velocity response for the first mode, estimated using the simplified load model.

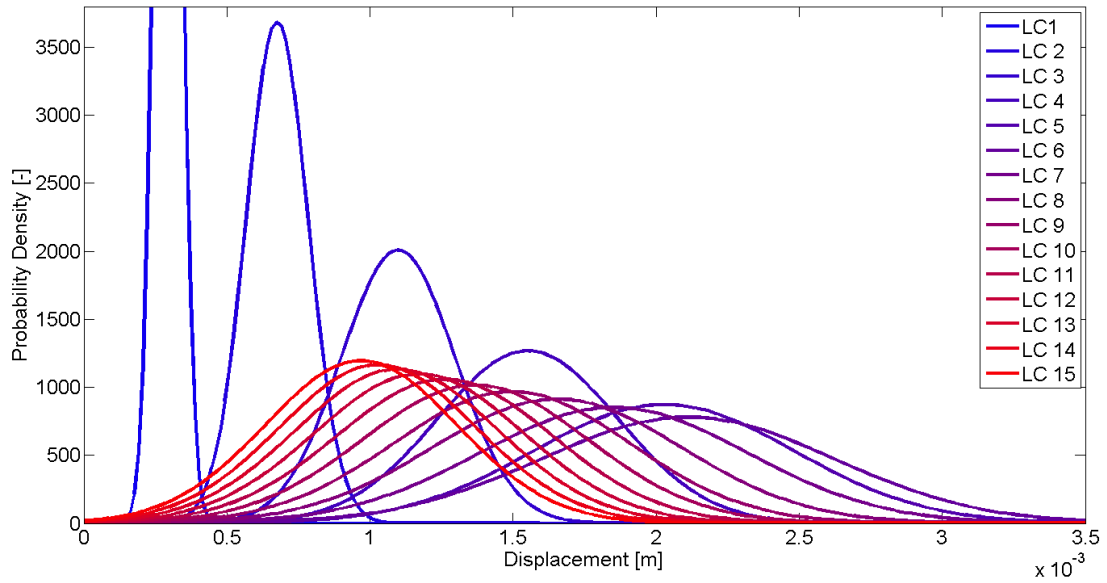


Figure 56: Tower top displacement distributions calculated analytically for the second mode using the simplified load model.

4.5 First Mode Response Contribution from Nonlinear Wind Loads Along the Tower for Load Case 2

The tower top response to wind loads acting along the tower for load case 2 has been estimated for the first fore-aft mode using both 1200 second Monte Carlo simulations and probability density evolution.

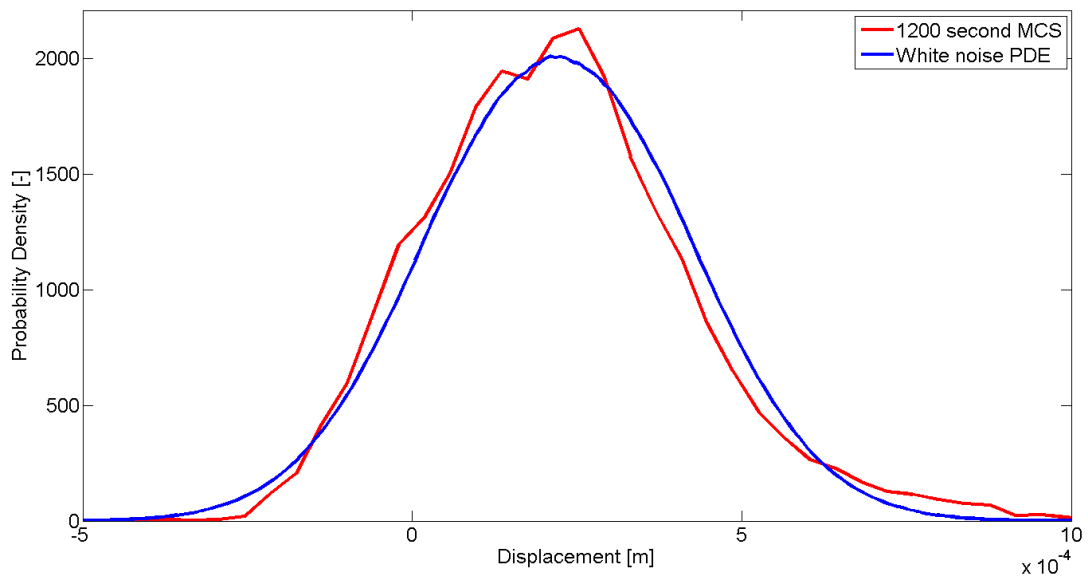


Figure 57: Estimated probability distributions of first mode tower top displacement for wind loads distributed along the tower. The wind loads are described by load case 2.

The time step used in Monte Carlo simulations is 0.0125 seconds, resulting in 96000 sampling points. Figure 57 presents the estimated displacement response distributions. A contour plot of the estimated joint displacement-velocity response using Monte Carlo simulation is shown in figure 58. The turbulence time series used in this simulation was generated by spectral decomposition of the Kaimal spectrum for load case 2. The corresponding PDE result for white noise excitation is presented in figure 59. A time step of 0.04 seconds was used in the PDE iterations. The domain discretizations were chosen as $dX = 7.5 \cdot 10^{-6}$ m, $d\dot{X} = 1.5 \cdot 10^{-5}$ m/s and $dF = 8$ N.

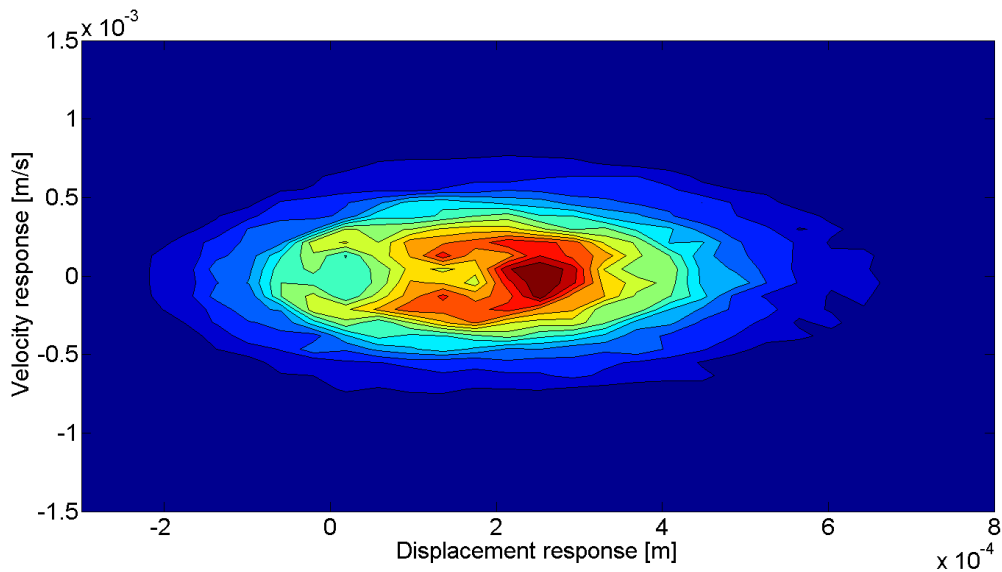


Figure 58: Result of a 1200 second Monte Carlo simulation of the tower top response under LC2 wind loads along the tower, for the first mode.

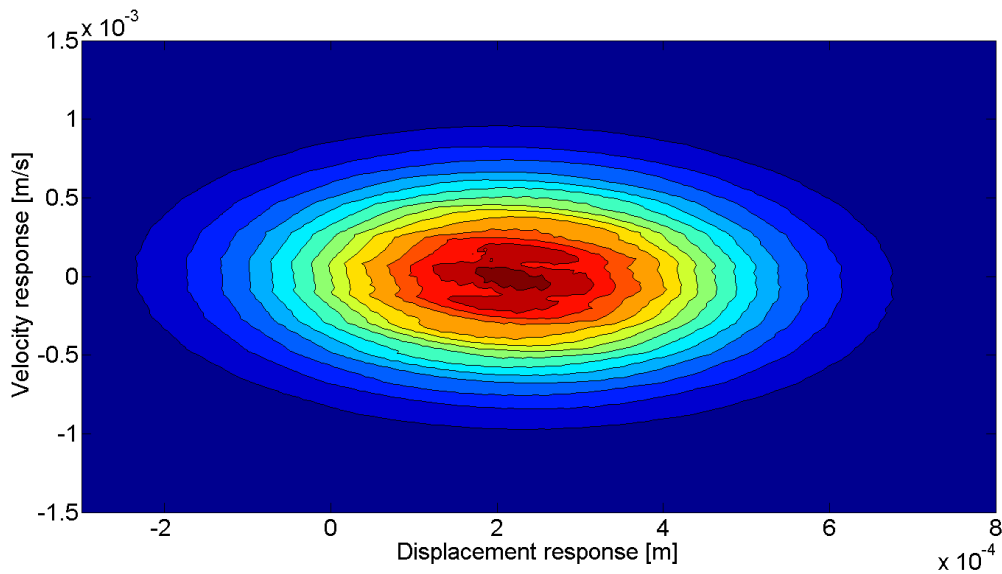


Figure 59: The tower top response under LC2 wind loads along the tower, estimated for the first mode using white noise probability density evolution.

5 Discussion

In this section, the results presented in section 4 are evaluated and discussed.

Modal Properties

In [2], Jonkman et al. states that the full-system natural frequencies for the two first fore-aft tower modes are found to be 0.3240 Hz and 2.9003 Hz, respectively. This corresponds well with the modal results presented in section 4.1, using a tower divided into 20 degrees of freedom. A higher accuracy is observed for the first mode, as could be expected because of the discretization.

The modal stiffness for the first mode could also be approximated by applying a point load to the top of the tower and treating it as a cantilever. Using the flexibility matrix in table 11, this modal property could be approximated as the inverse of the entry in the 20th row of the 20th column. This results in an estimated modal stiffness of 1807200 N/m, a value deviating only 1.8 kN/m from the one presented in table 6. The modal properties of the structure seems sensible for the early modes considered in this thesis.

It should be borne in mind that the real structure might not act linear, due to effects influencing the observed structural properties not accounted for. Examples of this could be gyroscopic forces from rotating machinery and effects of the turbine control system. The assumption of a structure that behaves linearly must therefore also be viewed as a model of the realistic situation.

One could suspect, based on the mode shapes presented in table 5 and figure 32, that the first mode would dominate the response of the tower top because of the small mode shape amplitudes at this point for the higher modes. This low modal amplitude is an effect of the relatively large point mass at the top of the tower representing the nacelle and rotor masses. This suspicion is strengthened by the fact that higher eigenfrequencies than the first would lie far from the main portion of the thrust fluctuation energy content, which is shown to occur at low frequencies in figure 35 to 41.

The Simplified Load Model and its Autoregressive Description

The load model is split into two intervals, with one set of model parameters between reference wind velocities of 2 m/s and 11 m/s, and one set between 11 m/s and 30 m/s. The rotor thrust force is obviously heavily influenced by the turbine control system. The sudden change in behavior around the rated wind speed of 11.4 m/s, which could be observed in figure 33 and 34, is caused by stall development as a turbulent wake forms above the airfoil because of separation of the boundary layer from the surface. Stalling of the blades is used to reduce the power from the wind turbine at high wind velocities, resulting in a close to constant power production in this range.

Figure 33 corresponds well with the thrust values for mean wind velocities presented by Jonkman et al. in [2]. From figure 33 and 34 one could observe that the mean value and variance of the load model thrust follows the behavior of the thrust obtained from FAST results closely for most load cases. This could also be seen by directly comparing time

series of both thrust simulations, plotted for the same wind history over 600 seconds in figure 35 to 39. One exception is for load case 1, where the variance of the simplified thrust is relatively low and the mean value relatively high. The overall good description of mean and variance is obviously an expected result, since the load model parameters were estimated as those giving the best fit of the model to these two statistical properties of the FAST simulations. Even though the simplified load model do provide a good estimate of the realistic mean value and variance of the rotor thrust, this does not necessarily imply that the proposed load model is accurate enough to provide a good structural response estimate.

Particularly, in order to test a plausible load model as simple to use and derive as possible, no measures were taken to match the variance spectra of the load model with the sample variance spectra of rotor thrust simulations. It is therefore an implicit assumption in the simplified load model that the frequency energy content of the rotor thrust could be reasonably described as proportional to the frequency energy content of the wind turbulence. To assess the validity of this assumption, both power spectral densities and sample autocovariances was presented in section 4.2.

First of all, comparing the PSD plots in figure 35 to 39 one could see that first order autoregression describes the frequency content of the proposed load model well. This is also observed in the autocovariance plots.

The good estimation of an AR(1) load spectrum to the analytical load spectrum, calculated using the simplified load model and the Kaimal turbulence spectrum, provides an excellent fit of the two modeled variances. The load model proposed in this thesis is therefore in general accurately described by first order autoregression. This also leads to a good fit of the AR(1) estimated thrust variance and the sample thrust variance of the FAST simulations, which could be observed in table 9. From this table it could also be observed that the AR(1) parameter ϕ is in general estimated as a value close to one. This supports other findings showing that the modeled stochastic processes is subjected to a considerable degree of correlation, and that white noise would describe the relationship between thrust realizations poorly.

Even though the PSD plots show that first order autoregression seems to describe the behavior of the simplified load model well, they also show that the proposed load model fail to account for distinct peaks in the low frequency region of the simulated thrust spectra. This is by extension also true for the AR(1) spectra since it is an estimate of the simplified load model spectra.

Such periodic contributions to the thrust stems from the rotor, and could usually be identified as multiples of the rotational frequency. These multiples are denoted 1P, 2P and so on, where 1P is the fundamental rotational frequency.

For load case 2, the average rotational velocity of the rotor is about 7.3 rpm, which would result in a 1P of 0.12 Hz, a 3P of 0.37 Hz and a 6P of 0.73 Hz. The two latter rotor frequencies could easily be identified as peaks in the PSD of the thrust for load case 2, presented in figure 40. The 3P rotor frequency, representing the blade passing frequency for a three bladed rotor like the one considered, contains the highest periodic energy content for this load case. This frequency could also be readily observed in the autocorrelation plot of the simulated thrust in figure 42.

When studying the spectral densities for both the turbulence from attachment 10 and the thrust from section 4.2 in light of the modal properties presented in table 6, one has reason to suspect that these cyclic loads indeed would affect the estimated structural response of the turbine.

This is because the greater part of the energy content in the thrust fluctuations described by the simplified load model would lie below the first natural frequency of the structure, 0.336 Hz. In practice, this would provide a close to quasistatic loading situation where inertial effects would only have a small impact on the estimated response. This is not particularly accurate if periodic rotor loads contribute with significant energy content at similar or higher frequencies than that of the first mode.

In addition to missing peaks in energy content at these specific frequencies, the accuracy of the estimated response could also be affected by the fact that the total variance of the thrust fluctuations are preserved when $\hat{C}_s V_{ref}^\beta$ is fitted as described in section 4.2. This basically leads to a simplified load model where energy peaks at relevant rotor frequencies are missing, but the variance contribution contained in those peaks are spread out over all frequencies.

Even though these periodic thrust contributions could be seen in the PSD plots for all load cases, the energy content of the peaks decrease relative to the rest of the spectra as the reference wind velocity and turbulence standard deviation increase. It is also a point that the positioning of the peaks would move further away from the first natural frequency of the structure, towards higher frequencies, until the turbine reaches the rated rotational speed of the rotor.

For load case 10 the rotational velocity of the rotor is 12.1 rpm, which is the rated speed. This results in a 3P frequency of 0.61 Hz and a 6P frequency of 1.21 Hz. In figure 41, both peaks are now lower compared to the general trend of the PSD than for load case 2. Because of these effects, cyclic rotor loads could be suspected to affect the response somewhat more at lower wind velocities.

Estimated Response Distributions and Evaluation of the Probability Density Evolution Method

As suspected from evaluating the system eigenvectors, the fore-aft tower top response seems to be dominated by the first mode. Comparing figure 46 and 56, the displacements for the second mode are roughly estimated to be about 0.5 % of the corresponding first mode displacements. Provided that the dynamic model describes the structure and excitation accurately, the response of the first mode should therefore provide a reasonable estimate of the total response. Unfortunately, the accuracy of the proposed simplified dynamic model seems somewhat debatable.

Considering figure 46 and table 10 one could see that use of the proposed simplified load model probably overestimates the true mean value of the displacement response of the tower top for all load cases. The analytically solved model response distributions also show that the response variance appears to be overestimated for all load cases except the two first.

As could be seen from figure 47, the ratios between the model values of mean and variance and the simulated values are fairly constant from load case 4 to 15. In this range of reference wind velocities the average value of S_1/μ_{u1} is about 83 %, while $\sqrt{S_2}/\sigma_{u1}$ is

about 49 %. A slight increasing trend in the fit of both response mean and variance could still be observed for wind velocities above 20 m/s in figure 47.

Discrepancies between simulated response and model response could be caused by either the model structure or the model thrust, since both of these are simplified. Most probably, they are caused by a combination of both.

Since all the mean displacements seem to be overestimated, one could argue that the modal stiffness obtained for the linearized model structure seem to result in a system that behaves too soft compared to the real turbine including its nonlinear effects.

It is obvious that the control system of the turbine, by definition, would affect the response to some degree. This is perhaps most visually apparent in figure 46 for load case 6 (presented as an enlarged plot in figure 49), which has a reference wind velocity of 12 m/s, just above the rated wind velocity of 11.4 m/s. This is a wind velocity region where the control system would affect the thrust significantly as stalling of the blades are initiated, causing two peaks to appear in the FAST simulated Monte Carlo estimate of the distribution. The control system could also be responsible for at least some of the skewness observed in a couple of the simulated response distributions.

While the linearization of the structure could account for some discrepancies, the load model is also likely to affect the estimated response since it neglects cyclic rotor contributions to thrust. This could partially explain the higher response variance in load case 1 and 2, as these reference wind velocities produce 3P rotor frequencies of about 0.30 Hz and 0.37 Hz respectively, close to the fundamental natural frequency of the tower at 0.34 Hz. As previously mentioned, the estimated response for load case 1 and 2 is also influenced by a relatively low thrust fluctuation variance from the model.

As the reference wind velocity increase beyond 4 m/s one could see from figure 47 that the variance given by the response to the load model becomes larger than the simulated response variance. One possible explanation for this is that the 3P frequency peak moves away from the fundamental natural frequency of the system and therefore makes a smaller impact on the simulated response. This is now combined with the effect of an artificially high energy content distributed over all frequencies in the load model, originating from the neglected periodic thrust contributions.

An observation supporting this hypothesis is that the displacement response variance calculated for the first mode using the simplified load model seems to approach the true variance of the displacements for high wind velocities, where the energy content in the rotor frequency peaks is less compared to the general spectral density.

From the first mode model response distributions obtained by PDE in figure 48, 49, 50 and 52, one could see that probability evolution gives a reasonable estimate of the analytical first mode response distribution. This is even true for PDE using white noise excitation in some cases. All PDE response estimates in these figures are found using a iteration time step of 0.05 seconds.

One noteworthy thing to learn from these distributions is the apparent importance of the spectral magnitude of the load process for frequencies close to the natural frequency of the structure. The spectra in this region is plotted with the squared transfer function of the structure for load case 4 in figure 51, and for load case 10 in figure 53.

For load case 4 the white noise force spectra happens to lie closer than the approximated AR(1) spectra to the theoretical model force spectra near the natural frequency of the first mode. As could be observed in figure 50, PDE with white noise excitation would

in this case actually yield a better estimate of the model response than correlated PDE using the relevant fitted AR(1) parameters. For load case 10, figure 53 reveals that the white noise spectra and the AR(1) spectra have approximately similar spectral densities near the fundamental natural frequency of the structure. This produces close to identical displacement response distributions using probability density evolution with and without incorporated force correlation.

In both of these load cases, the white noise and fitted AR(1) spectra have a smaller magnitude than the real spectrum of the simplified thrust near the natural frequency. This produces estimated response distributions with a smaller variance than the theoretical distribution obtained using equation (27)

Even though white noise excitation happens to give relatively accurate response for this particular time step, dynamic model and load cases, this is in no way thought to be true in general. For example, by using half the time step in iterations, the Nyquist frequency doubles and the constant white noise spectra ends up with half its current magnitude. When considering all frequencies, the autoregressive model would give a much better estimate of the entire thrust model spectra than a constant spectra would. It is therefore not hard to find situations where PDE with white noise excitation results in a much poorer estimation of the structural response than PDE with AR(1) correlated excitation, something that the example in section 3.6.2 clearly illustrates.

The transfer function in section 3.6.2 also has a broader peak at the natural frequency, since this system is more damped. This makes the gradient of the excitation spectrum even more important. This gradient could obviously not be described using a constant spectrum.

Two important conclusions could be made from these PDE results. First of all, it could be seen that if the structure is modeled as a single degree of freedom system with low damping, then a carefully selected iteration time step could result in a fairly accurate displacement response distribution using PDE with white noise excitation. This could be useful in some situations since the inclusion of the previous force realization as an additional dimension in the state space is computationally expensive. However, one should be aware that changing the time step is closely related to the discretization of the state space, accuracy and efficiency of the method. One should also note that by doing this, one would lose some information about the velocity and acceleration of the response process.

Second of all, first order autoregression seems well suited to represent the correlated forcing for the load model proposed in this thesis. However, one should note that a close fit of the autoregression spectra for frequencies close to the natural frequency might actually be more valuable than a fit that is as good as possible in a global frequency perspective. The latter approach is used to obtain estimates of the autoregression spectra used in this thesis.

As shown in figure 64 from attachment 11, the estimated velocity responses is in general less accurate than the estimated displacements. This is as expected for velocity response, because of the time differentiation affecting numerical accuracy.

Response to Nonlinear Wind Loads Along the Tower for Load Case 2

Section 4.5 was included to demonstrate the strength of the probability density evolution method associated with nonlinear excitation. As could be observed by the results, the white noise evolution of the structural response to any forcing probability density function is trivial as long as this function is known.

Probability evolution with white noise forcing provides a reasonable estimate of the tower top response. The magnitude of this response is very small compared to the estimated rotor thrust response, and the neglect of tower wind loads in the simplified load model therefore seems justifiable for this load case.

In order to obtain an even more accurate response distribution, the challenge would in this case be to model the correlation of the force, or the conditional probability of having the current force realizations given the previous one. This correlation would, in contrast to the AR(1) modeled correlation of the simplified thrust earlier in the thesis, not be independent of the value of the previous force realization.

One way to see this is by trying to write the modal force of mode j as a first order autoregressive process like it is shown in equation (121). In this equation, turbulence is still assumed to be accurately modeled as a first order autoregressive model also.

$$\begin{aligned}
 R_j(t) &= \hat{A}_j v(t)^2 + \hat{B}_j v(t) + \hat{C}_j \\
 &= \hat{A}_j (\phi_v v_{t-1} + Z_{v,t})^2 + \hat{B}_j (\phi_v v_{t-1} + Z_{v,t}) + \hat{C}_j \\
 &= \hat{A}_j \phi_v^2 v_{t-1}^2 + 2\hat{A}_j \phi_v v_{t-1} Z_{v,t} + \hat{A}_j Z_{v,t}^2 + \hat{B}_j \phi_v v_{t-1} + \hat{B}_j Z_{v,t} + \hat{C}_j
 \end{aligned} \tag{121}$$

As opposed to the simplified thrust fluctuations in equation (118), the modal force in equation (121) could not be written on the form of an AR(1) model. All terms involving $Z_{v,t}$ in this equation would contribute to the variance of the current modal force realization, and the cross term $2\hat{A}_j \phi_v v_{t-1} Z_{v,t}$ would make this variance dependent on the last realization.

6 Conclusion

The linearized structure properties derived in 4.1 corresponds well with results presented by Jonkman et al. in [2]. Both eigenvalues and eigenvectors of the model structure suggests that the first mode would contribute much more to the total response of the tower top than other modes. This is because of the relatively large mass in this point representing the nacelle and rotor, and because the natural frequencies of higher modes are found far from the low frequency main part of the modeled thrust spectra.

In sections 3.5 and 4.2, a simplified load model was proposed to describe the rotor thrust force of the 5-MW reference turbine. In this initial study it was desirable to assess the accuracy of a dynamic model that was simple to derive and use.

The proposed load model conserves the mean value and total variance of the simulated rotor thrust force, but is built on the assumption that the thrust force spectra could be estimated as one that is proportional to the turbulence spectra. This approximation does not seem to be good enough, as pronounced spectral peaks from cyclic rotor loads are not accounted for. In fact, cyclic rotor loads might contribute even more to the thrust force as the tower shadow effect is not considered in this thesis.

This is suspected to affect the accuracy of the estimated response negatively in two ways. First of all, it would lead to a falsely low response variance if a non-negligible multiple of the rotor frequency happens to be close to the fundamental natural frequency of the structure, which indeed is the case for low wind velocities. For the dynamic model described in this thesis the majority of the modeled thrust fluctuation energy content lies below the first natural frequency, leading to a system that in practice behaves quasistatically.

It would also lead to a falsely high energy content at other frequencies since the energy in these peaks are still accounted for by fitting the model thrust variance to the simulated thrust variance. These suspicions are supported by the variances observed in estimated response distributions for different load cases.

In this thesis the probability density evolution method is successfully used to estimate the response distribution of single degree of freedom systems. It is also shown how correlated excitation could be successfully implemented into the cell-mapping method, provided that the stochastic process could be reasonably modeled using autoregression. The results, both for the example in section 3.6.2 and the structure response to the simplified load model in section 4.4, show that incorporating the previous force realization into the state space certainly has the potential to provide improved response estimates for such dynamic systems. However, one should be aware that a good fit of the autoregressive spectra for frequencies near the natural frequency of the structure in many cases would result in a better estimate of the response than an autoregressive spectra that provides the best possible global fit.

If the modeled damping is small and the structure is modeled with a single degree of freedom, then PDE with white noise excitation could also yield a good estimate of the displacement response. However, adjusting the constant white noise spectra to match the model force spectra near the natural frequency would in principle imply changing the iteration time step. This could be problematic for several reasons. One also loses information about the velocity and acceleration response of the structure.

It is shown that first order autoregression provides a satisfactory estimate of the Kaimal turbulence spectra, and also of the proposed thrust fluctuation spectra because of the linear relationship between the two. Using autoregression to accurately account for the periodic correlation observed in rotor thrust simulations would however require a higher order model.

It is also observed that nonlinear load processes, which would be relevant for example when considering wave induced forces acting on a site-specific offshore sub-structure, might be less accurately described by first order autoregression than the current proposed load process.

As mentioned in the introduction, this is the first thesis written on PDE at the NTNU Offshore Wind Turbine Technology group. We have learned a lot, and the method shows great potential to be used in future projects involving simplified dynamic descriptions of wind turbines.

Recommendations and Comments for Future Studies

This thesis provides a starting point for further development of an accurate and reliable simplified dynamic model describing an offshore wind turbine. There are still many unresolved issues and possible improvements to be examined in future studies.

Improving the load model would be a natural way to continue these studies. In this case, the accurate response estimation of an onshore wind turbine should be a first priority. This would essentially involve considering cyclic thrust contributions from the rotor.

This could be done by deriving a simplified load spectra model that includes the most prominent peaks due to rotor thrust contributions. However, this would definitely pose a challenge when PDE should be used to estimate the response distribution. First order autoregression would in this case not be able to capture the necessary force correlation, and using a higher order autoregression model would increase the state space exponentially resulting in considerable loss of computational performance.

One possible solution is to use a simplified spectrum like the one used in this thesis, but to filter out variance from the peaks, for example by utilizing rotational sampling theory presented by Connell in [29]. By doing this, the periodic thrust contributions from the rotor could possibly be approximated as deterministic harmonic forces and therefore be kept separate from the probability evolution of the response due to wind. In this case the displacements due to cyclic rotor thrust would somehow have to be combined with the evolved displacements.

When a good dynamic model of an onshore turbine is established, the load model could be expanded to include wave loads acting on offshore sub-structures by using theory from section 2.3.3.

This would involve introducing the water particle velocity as another random variable, expanding the state space with at least one dimension. Additional dimensions might be necessary depending on how correlation of the wave loads are modeled. From the discussion of nonlinear load models one could conclude that first order autoregression might not directly describe nonlinear load processes as nicely as it describes the proposed load process in this thesis. Parallel computing would certainly be unavoidable if the state space is to be expanded with even more dimensions.

Even if an accurate model giving the approximate response distribution of an offshore wind turbine is eventually obtained, this is not a particularly useful result in its own right. The ultimate goal is to estimate the fatigue damage of various structural parts of the tower and sub-structure. The response distributions could readily be converted to stress distributions using mechanics theory. However, in order to obtain the necessary stress *range* distributions one would also have to gain knowledge about the total number of stress peaks.

For a narrow banded process x , this number of peaks could be estimated using the expected number of a -upcrossings per unit of time, $\nu_x^+(a)$. That is, the expected number of times a process crosses a given threshold a , heading upwards, per unit of time. This number is found, as shown in equation (122), as the integral over all positive velocities \dot{x} of the product between the positive velocity and the joint probability of having that velocity at the displacement threshold a . [6]

$$\nu_x^+(a) = \int_0^\infty \dot{x} f_{x\dot{x}}(a, \dot{x}) d\dot{x} \quad (122)$$

For a broad banded process the number of a -upcrossings per unit of time does not necessarily equal the number of peaks per unit of time, and this could possibly pose a challenge when estimating fatigue damage exclusively using the joint response distribution.

7 Attachments

7.1 Attachment 1:

Matlab script used to perform calculations of analytical SDOF response to stochastic excitation in section 2.1.3.

```

%
% John Magne Hembre 2014
%
% Script written to find and analyze response of a single degree of
% freedom system subjected to stationary stochastic delta-correlated
% (white noise) gaussian forcing.
%
clc
clear
close all
tic
%
% INPUT:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Forcing properties:
mu_f=0;
% Mean value of gaussian force F [N]
var_f=25;
% Variance of gaussian force F [N^2]

% Timespan properties:
reclength=620;
% Record length [s]
timestep=0.05;
% Timestep [s]
initial=20;
% Initial time period to be disregarded in the response calculation [s]
% (to obtain stationary signal without start-up singularity)
freqresolution=100000;
% Frequency resolution, deltafreq=fn/freqresolution

% SDOF Structure properties:
M=1;
% System mass [kg]
zeta=0.05;
% System damping [-]
K=3;
% System stiffness [N/m]

% Visualization properties:
histres=75;
% 3D histogram resolution

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SD_f=sqrt(var_f);
% Standard deviation of force signal [N]
n=reclength/timestep;
% Number of timesteps
forcevec=normrnd(mu_f,sqrt(var_f),n,1);
% Vector of gaussian distributed forces
timevec=timestep:timestep:reclength;
% Vector of time values
if initial~=0
initialsteps=initial/timestep;
% Number of initial timesteps to be disregarded
end

fs=1/timestep;
% The sampling frequency [Hz]
fn=fs/2;
% The Nyquist frequency [Hz]

```

```

% Determining S0:
S0=var_f/fn;
S0_omega=S0/(2*pi);

frequencies=0:fn/freqresolution:fn;
% Vector of the frequencies considered, in Hz.
omegafrequencies=frequencies*2*pi;
% Vector of the angular frequencies considered, in rad/s.

% Defining the Gaussian Probability Distribution:
gauss=@(mu,sigma,y) (1/(sqrt(2*pi)*sigma))*exp(-0.5*((y-mu)/sigma).^2);

% Defining the Joint Gaussian Probability Distribution, rho=0 for
% uncorrelated variables:
jointgauss=@(mu_1,sigma_1,y_1,mu_2,sigma_2,y_2) ((1/(sqrt(2*pi)*sigma_1))*...
    exp(-0.5*((y_1-mu_1)/sigma_1).^2))*...
    ((1/(sqrt(2*pi)*sigma_2))*exp(-0.5*((y_2-mu_2)/sigma_2).^2));

y_f=mu_f-(4*sqrt(var_f)):(2*4*sqrt(var_f))/1000:mu_f+(4*sqrt(var_f));

omega0=sqrt(K/M);
% Angular natural frequency of the SDOF system [rad/s]
omegad=omega0*sqrt(1-zeta^2);
% Damped angular natural frequency of the SDOF system [rad/s]
natfreq=omega0/(2*pi);
% Natural frequency of the SDOF system [Hz]

% Defining the SDOF system impulse response function:
hx=@(r) (exp(-zeta*omega0*r)/(omegad*M))*sin(omegad*r);

% Defining the SDOF system impulse velocity response function:
hxdot=@(rr) (exp(-zeta*omega0*rr)/(omegad*M))*((omegad*cos(omegad*rr))-...
    (zeta*omega0*sin(omegad*rr)));

% Defining the squared SDOF system frequency response function:
Hfxsquarded=@(omega) abs(1/(M^2*((omega0^2-
omega^2)^2+(2*zeta*omega0*omega)^2)));

% Generating vector of the squared transfer function to plot,
% and calculating sigma_x=S0*int(|H(omega)|^2):
Hfxsquardedvec=zeros(length(frequencies),1);
Sx=zeros(length(frequencies),1);
var_x=0;
var_xdot=0;
deltaomegafreq=(omegafrequencies(2)-omegafrequencies(1));
for i=1:length(omegafrequencies)
    Hfxsquardedvec(i)=Hfxsquarded(omegafrequencies(i));
    Sx(i)=Hfxsquardedvec(i)*(S0_omega);
    % Finding Sx=|Hfx|^2*Sf, where Sf=S0 (constant), times 2*pi
    % since S0(omega)=S0(f)/(2*pi)
    var_x=var_x+(Sx(i)*deltaomegafreq);
    var_xdot=var_xdot+(Sx(i)*(omegafrequencies(i)^2)*deltaomegafreq);
end

x_total=zeros(n,1);
xdot_total=zeros(n,1);
hx_vec=zeros(n,1);
hxdot_vec=zeros(n,1);
hx_static=0;
hxdot_static=0;
for t=1:n
    x_impulse=0;
    xdot_impulse=0;
    for s=1:t

```

```

        x_impulse=x_impulse+forcevec(s)*hx((t-1)*timestep-(s-
1)*timestep)*timestep;
        % Displacement response at t due to impulse at s
        xdot_impulse=xdot_impulse+(forcevec(s)*...
            hxdot((t-1)*timestep-(s-1)*timestep)*timestep);
        % Velocity response at t due to impulse at s
    end
    x_total(t)=x_impulse;
    % Total displacement response at t
    xdot_total(t)=xdot_impulse;
    % Total velocity response at t
    hx_static=hx_static+hx((t-1)*timestep)*timestep;
    % Used to calculate displacement response mean
    hxdot_static=hxdot_static+hxdot((t-1)*timestep)*timestep;
    % Used to calculate velocity response mean
    hx_vec(t)=hx((t-1)*timestep);
    % Used to plot impulse response function
    hxdot_vec(t)=hxdot((t-1)*timestep);
    % Used to plot h_dot
end

hx_static_real=0;
hxdot_static_real=0;
for t=1:n*1000
    hx_static_real=hx_static_real+hx(t*(timestep/1000))*(timestep/1000);
    % Used to calculate (the real) displacement response mean

hxdot_static_real=hxdot_static_real+hxdot(t*(timestep/1000))*(timestep/1000);
    % Used to calculate (the real) velocity response mean
end

if initial~=0
x=x_total(initialsteps:n);
xdot=xdot_total(initialsteps:n);
timevec_resp=timevec(initialsteps:n);
else
x=x_total;
xdot=xdot_total;
timevec_resp=timevec;
end
% Cuts away unwanted initial part

% Displacement response mean:
mu_x=mu_f*hx_static
mu_x_real=mu_f*hx_static_real
mu_x_fromsignal=mean(x)

% Velocity response mean:
mu_xdot=mu_f*hxdot_static
mu_xdot_real=mu_f*hxdot_static_real % Should be 0
mu_xdot_fromsignal=mean(xdot)

toc

```


7.2 Attachment 2:

Matlab script used to evolve probability distributions of structural response through the cell-mapping method.

```

%
% John Magne Hembre and Michael Muskulus, 2014
%
% Script written to find estimate the Joint Probability Density Function
% of the response of a SDOF system using the Probability Density Evolution.
% The excitation could either be described by the current force
% realization, F, picked as white noise, or by the current and
% the previous force realization, F and Z. In the latter case the
% conditional probability of Z is described using first order
% autoregression.
%
clc
close all
%
% INPUT:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Time properties:
timestep=0.05;
% Time step [s]

% Iteration properties:
itmax=300000;
% Maximum number of iterations
L2_max=10^(-90);
% Convergence criterion; L2 difference
plotfreq=2;
% Plot every plotfreq iteration

% Initial condition, t=0:
initstate_disp=0;      % [m]
initstate_velo=0;     % [m/s]
initstate_force=0;    % [N]
initialcond=2;        % 1 = Use spike at initstate
                     % 2 = Use predefined "initpdf" from workspace
                     % 3 = Other experimental initial states

% Domain properties:
X_min=-0.3;           % [m]
X_max=0.3;            % [m]
Xdot_min=-0.6;       % [m/s]
Xdot_max=0.6;        % [m/s]
F_min=-300000;        % [N]          (used if forcemode=1 or 2)
F_max=300000;         % [N]          (used if forcemode=1 or 2)
Z_min=-30000;         % [N]          (used if forcemode=2)
Z_max=30000;          % [N]          (used if forcemode=2)

% Resolution properties:
delta_X=(X_max-X_min)/200;      % [m]
delta_Xdot=(Xdot_max-Xdot_min)/200; % [m/s]
delta_F=(F_max-F_min)/100;      % [N]
delta_Z=(Z_max-Z_min)/30;       % [N]

delta_u=0.1;                    % [m/s]
% delta_Z_u=0.1;                % [m/s]

% SDOF Structure properties:

M=405574.9;
% System mass [kg]
K=1809033.4;
% System stiffness [N/m]
zeta=0.01;
%zeta=1/(2*sqrt(K*M)); % [To define zeta from C; zeta=C/(2*sqrt(K*M))]

```

```

% System damping [-]

% Force properties:
V_ref=20;
% Mean value of wind velocity V [m/s]
sd_u=2.884;
% Standard deviation of gaussian distribution for turbulence u (mean=0)

forcemode=2;          % Mode 1: Explicit white noise force with a known
                    %           PDF given force
                    %           [PDF_thrust(F)]
                    %
                    % Mode 2: Explicit force with a known PDF given
                    %           force, modeled as an AR(1) process with
                    %           known parameters phi and noise variance
                    %           [PDF_thrust(F)]

if V_ref<11
    turbconst=6007*V_ref^(1);
elseif V_ref>=11
    turbconst=1682000*V_ref^(-1.35);
end

if forcemode==2

    phi_force=0.99506;
    sigma_noise=8442.2;
%   noisevar_force=9;
%   sigma_noise=sqrt(noisevar_force);

    % Noise properties:
    n_Z=(Z_max-Z_min)/delta_Z;
    Z=linspace(Z_min,Z_max,n_Z);
    pdfZ=(1/(sqrt(2*pi)*sigma_noise))*exp(-0.5.*(Z./sigma_noise).^2);
    figure(50)
    plot(Z,pdfZ)
    axis([Z_min Z_max 0 1.1*max(pdfZ)])
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

C=zeta*2*sqrt(K*M);
% Calculate C [kg/s] from zeta [-]

omega0=sqrt(K/M);
% Angular natural frequency of the SDOF system [rad/s]

n_X=(X_max-X_min)/delta_X;
n_Xdot=(Xdot_max-Xdot_min)/delta_Xdot;
n_F=(F_max-F_min)/delta_F;           % (used if forcemode = 1 or 3)
n_u=(u_max-u_min)/delta_u;         % (used if forcemode = 2)
% Number of steps in the different domains

% Generating vectors:
X=linspace(X_min,X_max,n_X);
Xdot=linspace(Xdot_min,Xdot_max,n_Xdot);
F=linspace(F_min,F_max,n_F);       % (used if forcemode = 1 or 3)
u=linspace(u_min,u_max,n_u);       % (used if forcemode = 2)

% Finding cell ID-number of initial state cells:
for cellindex_x=1:(n_X-1)
    if (initstate_disp>=X(cellindex_x))&&(initstate_disp<X(cellindex_x+1))
        d1=abs(initstate_disp-X(cellindex_x));
        d2=abs(initstate_disp-X(cellindex_x+1));
    end
end

```

```

        if d1<d2
            initcell_disp=cellindex_x;
        else
            initcell_disp=cellindex_x+1;
        end
        break
    end
end
for cellindex_x=1:(n_Xdot-1)
    if (initstate_velo>=Xdot(cellindex_x))&&...
        (initstate_velo<Xdot(cellindex_x+1))
        d1=abs(initstate_velo-Xdot(cellindex_x));
        d2=abs(initstate_velo-Xdot(cellindex_x+1));
        if d1<d2
            initcell_velo=cellindex_x;
        else
            initcell_velo=cellindex_x+1;
        end
        break
    end
end
for cellindex_f=1:(n_F-1)
    if (initstate_force>=F(cellindex_f))&&...
        (initstate_force<F(cellindex_f+1))
        d1=abs(initstate_force-F(cellindex_f));
        d2=abs(initstate_force-F(cellindex_f+1));
        if d1<d2
            initcell_force=cellindex_f;
        else
            initcell_force=cellindex_f+1;
        end
        break
    end
end
end

if forcemode==1
    pdf=zeros(n_X,n_Xdot);
    pdf2=zeros(n_X,n_Xdot);
elseif forcemode==2
    pdf=zeros(n_X,n_Xdot,n_F);
    pdf2=zeros(n_X,n_Xdot,n_F);
end

disp('Setting up initial state and force properties...')
tic
if initialcond==1
    if forcemode==1
        % If spike: setting the probability of the defined initial state to 1
        pdf(initcell_disp,initcell_velo)=1;
    elseif forcemode==2
        % If spike: setting the probability of the defined initial state to 1
        pdf(initcell_disp,initcell_velo,initcell_force)=1;
    end
end

elseif initialcond==2
    pdf=initpdf;
    % Use initial pdf from workspace if predefined

elseif initialcond==3

```

```

% Used to create other initial states
num=3;
dist=1;

probcells=1/((num*8)+1);
pdf(initcell_disp,initcell_velo,initcell_force)=probcells;
for i=1:num
pdf(initcell_disp+dist*i,initcell_velo+dist*i,initcell_force+dist*i)...
    =probcells;
pdf(initcell_disp+dist*i,initcell_velo+dist*i,initcell_force-dist*i)...
    =probcells;
pdf(initcell_disp+dist*i,initcell_velo-dist*i,initcell_force-dist*i)...
    =probcells;
pdf(initcell_disp+dist*i,initcell_velo-dist*i,initcell_force+dist*i)...
    =probcells;
pdf(initcell_disp-dist*i,initcell_velo-dist*i,initcell_force-dist*i)...
    =probcells;
pdf(initcell_disp-dist*i,initcell_velo+dist*i,initcell_force-dist*i)...
    =probcells;
pdf(initcell_disp-dist*i,initcell_velo+dist*i,initcell_force+dist*i)...
    =probcells;
pdf(initcell_disp-dist*i,initcell_velo-dist*i,initcell_force+dist*i)...
    =probcells;
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Force PDF properties:
if forcemode==1 % If explicit white noise force PDF...
    % Define Force PDF:

    sigma_F=sqrt((turbconst^2)*(sd_u^2)); % [N]
    mu_F=0; % [N]

    pdfvec_prevthrust=(1/(sqrt(2*pi)*sigma_F))*...
        exp(-0.5*((F-mu_F)./sigma_F).^2);
    % Vector containing previous thrust values (Gaussian distribution)

    pdfF=pdfvec_prevthrust;

%     mode=1; % Mode considered
%     if mode==1;
%         modalA=25.0108; % Mode 1, LC 2
%         modalB=190.2907; % Mode 1, LC 2
%         modalC=362.7748; % Mode 1, LC 2
%     elseif mode==2;
%         modalA=41.9390; % Mode 2, LC 2
%         modalB=305.6178; % Mode 2, LC 2
%         %modalC=558.6578; % Mode 2, LC 2
%     end

%     modalthrust_PDF=@(sigma_v,A,B,C,Tval)...
%     (1./(sqrt(2*pi)*sigma_v.*sqrt((B^2)-4*A*(C-Tval))))*...
%     (exp(-0.5*((-B+sqrt((B^2)-4*A*(C-Tval)))./(2*sigma_v*A)).^2)+...
%     exp(-0.5*((-B-sqrt((B^2)-4*A*(C-Tval)))./(2*sigma_v*A)).^2));
%     % PDF of nonlinear modal force caused by wind distributed along tower
%
%     pdfF=modalthrust_PDF(sd_u,modalA,modalB,modalC,F);
%     % Vector containing previous thrust values
%     % (nonlinear PDF of modal force caused by wind distributed along tower)

area=trapz(F,pdfF);
pdfF=pdfF./area;
% Normalized pdf of the force

```

```

figure(2)
plot(F,pdfF)

elseif forcemode==2 % If considering frequency dependent force with
                    % known PDF(F)

    sigma_F=sqrt((turbconst^2)*(sd_u^2)); % [N]
    mu_F=0; % [N]

    pdfvec_prevthrust=(1/(sqrt(2*pi)*sigma_F))*...
        exp(-0.5.*((F-mu_F)./sigma_F).^2);
    % Vector containing previous thrust values (Gaussian distribution)

    figure(2)
    plot(F,pdfvec_prevthrust)

%     mode=1; % Mode considered
%     if mode==1;
%         modalA=25.0108; % Mode 1, LC 2
%         modalB=190.2907; % Mode 1, LC 2
%         modalC=362.7748; % Mode 1, LC 2
%     elseif mode==2;
%         modalA=41.9390; % Mode 2, LC 2
%         modalB=305.6178; % Mode 2, LC 2
%         %modalC=558.6578; % Mode 2, LC 2
%     end
%
%     modalthrust_PDF=@(sigma_v,A,B,C,Tval)...
%     (1./(sqrt(2*pi)*sigma_v.*sqrt((B^2)-4*A*(C-Tval))))*...
%     (exp(-0.5.*((-B+sqrt((B^2)-4*A*(C-Tval)))./(2*sigma_v*A)).^2)+...
%     exp(-0.5.*((-B-sqrt((B^2)-4*A*(C-Tval)))./(2*sigma_v*A)).^2));
%     % PDF of nonlinear modal force caused by wind distributed along tower
%
%     pdfvec_prevthrust=modalthrust_PDF(sd_u,modalA,modalB,modalC,F);
%     % Vector containing previous thrust values
%     % (nonlinear PDF of modal force caused by wind distributed along tower)

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp(['That took ', num2str(toc), ' s'])
fprintf('\n')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

statemap_F=zeros(n_F,1); % Empty vector
statemap_X=zeros(n_X,n_Xdot); % Empty 2D matrix
statemap_Xdot=zeros(n_X,n_Xdot,n_F); % Empty 3D matrix

disp('Calculating transition probabilities...')
tic

if forcemode==1||forcemode==3
    n_forcing=n_F;
    forcevec=F;
elseif forcemode==2
    n_forcing=n_u;
    forcevec=currs_T;
end

if forcemode==2||forcemode==3

```

```

for i=1:n_F
    for j=1:n_Z
        f1=F(i);
        f2=f1*phi_force+Z(j);
        % Finding resulting cell ID-number for statemap_F:
        if f2<=F_min
            statemap_F(i,j)=1;
        elseif f2>=F_max
            statemap_F(i,j)=n_F;
        else
            for cellindex_f=1:(n_F-1)
                if (f2>=F(cellindex_f)) && (f2<F(cellindex_f+1))
                    d1=abs(f2-F(cellindex_f));
                    d2=abs(f2-F(cellindex_f+1));
                    if d1<d2
                        statemap_F(i,j)=cellindex_f;
                    else
                        statemap_F(i,j)=cellindex_f+1;
                    end
                    break
                end
            end
        end
    end
end
end

for i=1:n_X
    x1=X(i);

    for j=1:n_Xdot
        xdot1=Xdot(j);
        x2=x1+timestep*(xdot1+(delta_Xdot/2)); % Modified Euler
        %x2=x1+timestep*(xdot1); % Explicit Euler

        % Finding resulting cell ID-number for statemap_X:
        if x2<=X_min
            statemap_X(i,j)=1;
        elseif x2>=X_max
            statemap_X(i,j)=n_X;
        else
            for cellindex_x=1:(n_X-1)
                if (x2>=X(cellindex_x)) && (x2<X(cellindex_x+1))
                    d1=abs(x2-X(cellindex_x));
                    d2=abs(x2-X(cellindex_x+1));
                    if d1<d2
                        statemap_X(i,j)=cellindex_x;
                    else
                        statemap_X(i,j)=cellindex_x+1;
                    end
                    break
                end
            end
        end
    end

    for k=1:n_forcing
        f=forcevec(k);
        xdot2=xdot1+timestep*((f-C*((x2-x1)/timestep)-K*x2)/M);
        % Implicit Euler

        % Finding resulting cell ID-number for statemap_Xdot:
        if xdot2<=Xdot_min
            statemap_Xdot(i,j,k)=1;
        elseif xdot2>=Xdot_max

```

```

        statemap_Xdot(i,j,k)=n_Xdot;
    else
        for cellindex_xdot=1:(n_Xdot-1)
            if (xdot2>=Xdot(cellindex_xdot))&&...
                (xdot2<Xdot(cellindex_xdot+1))
                d1=abs(xdot2-Xdot(cellindex_xdot));
                d2=abs(xdot2-Xdot(cellindex_xdot+1));
                if d1<d2
                    statemap_Xdot(i,j,k)=cellindex_xdot;
                else
                    statemap_Xdot(i,j,k)=cellindex_xdot+1;
                end
                break
            end
        end
    end
end

end
end
end

disp(['That took ', num2str(toc), ' s'])
fprintf('\n')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if forcemode==1 % If explicit white noise force PDF...

disp('Iterating...')
tic
for o=1:itmax
    area_pdf=sum(sum(pdf))*delta_X*delta_Xdot; %
    pdf=pdf/area_pdf; % Normalize

    sum_frame=sum(pdf(1,:))+sum(pdf(n_X,:))+sum(pdf(:,1))+...
        sum(pdf(:,n_Xdot));
    sum_picture=sum(sum(pdf(2:(n_X-1),2:(n_Xdot-1))));
    if sum_frame>0.001*sum_picture
        fprintf('\n')
        disp({'##### Error: #####'})
        disp({'PDF out of bounds'})
        disp({'#####'})
        fprintf('\n')
        break
    end

    L2=sqrt(mean2((pdf-pdf2).^2));
    if L2<L2_max
        disp(['Obtained L2 difference = ', num2str(L2)])
        break
    end
    disp(['Iteration ', num2str(o), ', L2-difference = ', num2str(L2)])
    if o==itmax
        disp('Convergence not obtained before maximum number of iterations')
    end
    pdf2=pdf;
    pdf=zeros(n_X,n_Xdot);
    for i=1:n_X
        % For each X...
        for j=1:n_Xdot
            % For each Xdot...

```



```

        i2=statemap_X(i,j);           % Resulting X-cell ID
        for k=1:n_F
            % For each F...
            j2=statemap_Xdot(i,j,k); % Resulting Xdot-cell ID
            pdf(i2,j2)=pdf(i2,j2)+(pdf2(i,j)*pdfF(k));
        end
    end
end
end
if rem(o,plotfreq)==0
figure(1000+o)
area_pdf=sum(sum(pdf))*delta_X*delta_Xdot;
pdfplot=pdf/area_pdf;           % Normalize
pdfplot=pdfplot';             % Transpose to get axes right for surf
subplot(1,2,1)
surf(X,Xdot,pdfplot,'EdgeColor','none')
xlabel('Displacement [m]')
ylabel('Velocity [m/s]')
view(0,90);
subplot(1,2,2)
contourf(X,Xdot,pdfplot,40)
end

end
iteration_time=toc;
fprintf('\n')
disp(['The ',num2str(o),' iterations took ',...
    num2str(iteration_time), ' s,'])
disp(['giving an average of ',...
    num2str(iteration_time/o),' s per iteration'])
fprintf('\n')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif forcemode==2
% If considering frequency dependent wind force...

disp('Iterating...')
tic
for o=1:itmax
    volume_pdf=sum(sum(sum(pdf)))*delta_X*delta_Xdot*delta_F; %
    pdf=pdf./volume_pdf; % Normalize

    L2=sqrt(mean2((pdf-pdf2).^2));
    if L2<L2_max
        disp(['Obtained L2 difference = ', num2str(L2)])
        break
    end
    disp(['Iteration ',num2str(o),' , L2-difference = ',num2str(L2)])
    if o==itmax
        disp('Convergence not obtained before maximum number of iterations')
    end
    pdf2=pdf;
    pdf=zeros(n_X,n_Xdot,n_F);
    for i=1:n_X
        % For each X...
        for j=1:n_Xdot
            % For each Xdot...
            i2=statemap_X(i,j);           % Resulting X-cell ID
            for k=1:n_forcing
                % For each F...
                for m=1:n_Z
                    k2=statemap_F(k,m); % Resulting F-cell ID
                    j2=statemap_Xdot(i,j,k2); % Resulting Xdot-cell ID
                    pdf(i2,j2,k2)=pdf(i2,j2,k2)+(pdf2(i,j,k)*pdfZ(m));
                end
            end
        end
    end
end

```

```

        end
    end
end
end
if rem(o,plotfreq)==0
pdf2D=sum(pdf,3);
figure(1000+o)
area_pdf=sum(sum(pdf2D))*delta_X*delta_Xdot;
pdfplot2D=pdf2D/area_pdf;           % Normalize
pdfplot2D=pdfplot2D';             % Transpose to get axes right for surf
subplot(1,2,1)
surf(X,Xdot,pdfplot2D,'EdgeColor','none')
xlabel('Displacement [m]')
ylabel('Velocity [m/s]')
view(0,90);
subplot(1,2,2)
contourf(X,Xdot,pdfplot2D,40)
end

end
iteration_time=toc;
fprintf('\n')
disp(['The ',num2str(o),' iterations took ',...
      num2str(iteration_time), ' s,'])
disp(['giving an average of ',...
      num2str(iteration_time/o),' s per iteration'])
fprintf('\n')

end

pdf2D=sum(pdf,3);
figure(5)
area_pdf=sum(sum(pdf2D))*delta_X*delta_Xdot;
pdf2D=pdf2D./area_pdf;           % Normalize
pdf2D=pdf2D';             % Transpose to get axes right for surf
surf(X,Xdot,pdf2D)
xlabel('Displacement [m]')
ylabel('Velocity [m/s]')

```

7.3 Attachment 3:

Matlab function used to simulate wind turbulence through spectral decomposition of the Kaimal spectra.

```

%
% John Magne Hembre 2014
%
% Function that simulates turbulence through spectral decomposition of the
% Kaimal wind spectrum.
%
% vis = visualization on/off (1/0)

function v = Kaimalsim(reclength,timestep,v_hub,sigma,hubhght,vis)

clc

maxfreq=1/(2*timestep); % The nyquist frequency [Hz]
N=maxfreq*5000; % Number of points in spectrum
df=maxfreq/N; % Frequency increments [Hz]
freqs=0:df:maxfreq;

if hubhght<60
    L_k=8.10*0.7*hubhght;
else
    L_k=8.10*0.7*60;
end

S_f=(4*sigma^2*(L_k/v_hub))./(1+(6.*freqs*(L_k/v_hub)).^(5/3));

sigmaest_spec=sqrt(sum(df.*S_f))

maxomega=maxfreq*2*pi;
domega=maxomega/N;
omegas=(2*pi).*freqs;

S_omega=S_f./(2*pi);

tmax=reclength; % Wind velocity record length [s]
time=0:timestep:tmax-timestep; % Vector over time steps

v=zeros(1,length(time));

for k=1:length(omegas)
    omega_k=omegas(k);
    S_k=S_omega(k);
    rand_k=rand;
    for step_num=1:length(time)
        v(step_num)=v(step_num)+sqrt(2*S_k*domega)*...
            cos(omega_k*time(step_num)+2*pi*rand_k);
    end
end

v=v+v_hub;
v=v';

if vis==1
figure(1000)
plot(time,v)
grid on
axis([0 tmax -1.2*min(v) 1.2*max(v)])
xhandle=xlabel('t [s]');
yhandle=ylabel('v(t) [m/s]');
set(gca,'fontsize',14)
set(xhandle,'FontSize',14)
set(yhandle,'FontSize',14)

end
end

```

7.4 Attachment 4:

Matlab function estimating total thrust and torque at the hub using blade properties from [2] and the Blade Element Momentum Method.

```

% Function BEM:
%
% John Magne Hembre 2014
%
% Function written to calculate the total thrust and torque force acting
% at the rotor center of the NREL 5-MW Reference Wind Turbine (Jonkman
% et al. 2009). The result is two vectors with the same length as the input
% vector U containing the total calculated thrust and torque in every
% time step. The Blade Element Momentum theory used herein is based on
% the book "Wind Energy Explained" (Manwell et al. 2002).
%
% The function takes the following input:
%
% U = a time series of the wind velocity at hub height. [m/s]
%
% Time = the corresponding time vector for the wind velocity [s]
%
% B = number of blades on rotor.
%
% Omega = angular velocity of the rotor [rad/s]. Omega is assumed
%         constant for the calculation of a and a' (The axial and angular
%         induction factors) (Could be found in Jonkman et al. 2009)
%
% rho = mass density of the air considered (standard is 1.25). [kg/m^3]
%
% maxdev = The maximal deviation required between iteration rounds for the
%           values of a and a'. (e.g. 0.00001)
%
function [total_thrust, total_torque,...
        mom_total_thrust, mom_total_torque] = BEM(U,Time,B,Omega,rho,maxdev)

clc
close all
tic

r=[2.8667;5.6000;8.3333;11.7500;
   15.8500;19.9500;24.0500;28.1500;
   32.2500;36.3500;40.4500;44.5500;
   48.6500;52.7500;56.1667;58.9000;61.6333];
% Values of blade section locations [m]

dr=[2.7333;2.7333;2.7333;4.1000;
    4.1000;4.1000;4.1000;4.1000;
    4.1000;4.1000;4.1000;4.1000;
    4.1000;4.1000;2.7333;2.7333;2.7333];
% Values of blade section lengths [m]

twist=[13.308;13.308;13.308;13.308;
       11.480;10.162;9.011;7.795;
       6.544;5.361;4.188;3.125;
       2.319;1.526;0.863;0.370;0.106].*(pi/180);
% Values of blade section twist angle [rad]

chord=[3.542;3.854;4.167;4.557;
       4.652;4.458;4.249;4.007;
       3.748;3.502;3.256;3.010;
       2.764;2.518;2.313;2.086;1.419];
% Values of blade section chord lengths [m]

foilnumbers=[1;1;2;3;
             4;4;5;6;
             6;7;7;8;
             8;8;8;8;8];

```

```

% Determines what airfoil table to use:
% 1: Cylinder 1 (Special case, does not use external file)
% 2: Cylinder 2 (Special case, does not use external file)
% 3: DU40_A17
% 4: DU35_A17
% 5: DU30_A17
% 6: DU25_A17
% 7: DU21_A17
% 8: NACA64_A17

fid=fopen('DU21_A17.dat','r');
DU21_A17_temp=textscan(fid,'%f');
fclose(fid);
DU21_A17_temp=DU21_A17_temp{1};
DU21_A17=zeros((length(DU21_A17_temp)/4),4);
for i=1:4:length(DU21_A17_temp)
    DU21_A17(((i+3)/4),1)=DU21_A17_temp(i,1);
    DU21_A17(((i+3)/4),2)=DU21_A17_temp(i+1,1);
    DU21_A17(((i+3)/4),3)=DU21_A17_temp(i+2,1);
    DU21_A17(((i+3)/4),4)=DU21_A17_temp(i+3,1);
end
clear('DU21_A17_temp')
% Generate a matrix containing coefficients Cl, Cd and Cm (column 2, 3
% and 4) for different angles of attack (column 1) for airfoil DU21_A17.

fid=fopen('DU25_A17.dat','r');
DU25_A17_temp=textscan(fid,'%f');
fclose(fid);
DU25_A17_temp=DU25_A17_temp{1};
DU25_A17=zeros((length(DU25_A17_temp)/4),4);
for i=1:4:length(DU25_A17_temp)
    DU25_A17(((i+3)/4),1)=DU25_A17_temp(i,1);
    DU25_A17(((i+3)/4),2)=DU25_A17_temp(i+1,1);
    DU25_A17(((i+3)/4),3)=DU25_A17_temp(i+2,1);
    DU25_A17(((i+3)/4),4)=DU25_A17_temp(i+3,1);
end
clear('DU25_A17_temp')
% Generate a matrix containing coefficients Cl, Cd and Cm (column 2, 3
% and 4) for different angles of attack (column 1) for airfoil DU25_A17.

fid=fopen('DU30_A17.dat','r');
DU30_A17_temp=textscan(fid,'%f');
fclose(fid);
DU30_A17_temp=DU30_A17_temp{1};
DU30_A17=zeros((length(DU30_A17_temp)/4),4);
for i=1:4:length(DU30_A17_temp)
    DU30_A17(((i+3)/4),1)=DU30_A17_temp(i,1);
    DU30_A17(((i+3)/4),2)=DU30_A17_temp(i+1,1);
    DU30_A17(((i+3)/4),3)=DU30_A17_temp(i+2,1);
    DU30_A17(((i+3)/4),4)=DU30_A17_temp(i+3,1);
end
clear('DU30_A17_temp')
% Generate a matrix containing coefficients Cl, Cd and Cm (column 2, 3
% and 4) for different angles of attack (column 1) for airfoil DU30_A17.

fid=fopen('DU35_A17.dat','r');
DU35_A17_temp=textscan(fid,'%f');
fclose(fid);
DU35_A17_temp=DU35_A17_temp{1};
DU35_A17=zeros((length(DU35_A17_temp)/4),4);
for i=1:4:length(DU35_A17_temp)
    DU35_A17(((i+3)/4),1)=DU35_A17_temp(i,1);
    DU35_A17(((i+3)/4),2)=DU35_A17_temp(i+1,1);
    DU35_A17(((i+3)/4),3)=DU35_A17_temp(i+2,1);

```

```

    DU35_A17(((i+3)/4),4)=DU35_A17_temp(i+3,1);
end
clear('DU35_A17_temp')
% Generate a matrix containing coefficients Cl, Cd and Cm (column 2, 3
% and 4) for different angles of attack (column 1) for airfoil DU35_A17.

fid=fopen('DU40_A17.dat','r');
DU40_A17_temp=textscan(fid,'%f');
fclose(fid);
DU40_A17_temp=DU40_A17_temp{1};
DU40_A17=zeros((length(DU40_A17_temp)/4),4);
for i=1:4:length(DU40_A17_temp)
    DU40_A17(((i+3)/4),1)=DU40_A17_temp(i,1);
    DU40_A17(((i+3)/4),2)=DU40_A17_temp(i+1,1);
    DU40_A17(((i+3)/4),3)=DU40_A17_temp(i+2,1);
    DU40_A17(((i+3)/4),4)=DU40_A17_temp(i+3,1);
end
clear('DU40_A17_temp')
% Generate a matrix containing coefficients Cl, Cd and Cm (column 2, 3
% and 4) for different angles of attack (column 1) for airfoil DU40_A17.

fid=fopen('NACA64_A17.dat','r');
NACA64_A17_temp=textscan(fid,'%f');
fclose(fid);
NACA64_A17_temp=NACA64_A17_temp{1};
NACA64_A17=zeros((length(NACA64_A17_temp)/4),4);
for i=1:4:length(NACA64_A17_temp)
    NACA64_A17(((i+3)/4),1)=NACA64_A17_temp(i,1);
    NACA64_A17(((i+3)/4),2)=NACA64_A17_temp(i+1,1);
    NACA64_A17(((i+3)/4),3)=NACA64_A17_temp(i+2,1);
    NACA64_A17(((i+3)/4),4)=NACA64_A17_temp(i+3,1);
end
clear('NACA64_A17_temp')
% Generate a matrix containing coefficients Cl, Cd and Cm (column 2, 3
% and 4) for different angles of attack (column 1) for airfoil NACA64_A17.

foilarray{1,1}=0;
foilarray{2,1}=0;
foilarray{3,1}=DU40_A17;
foilarray{4,1}=DU35_A17;
foilarray{5,1}=DU30_A17;
foilarray{6,1}=DU25_A17;
foilarray{7,1}=DU21_A17;
foilarray{8,1}=NACA64_A17;
clear('DU21_A17','DU25_A17','DU30_A17','DU35_A17','DU40_A17','NACA64_A17')
% Generate an array containing the airfoils above. Their order corresponds
% to the order given in the vector foilnumbers.

n=length(U);
% Number of time steps in the vector U

total_thrust=zeros(n,1);
total_torque=zeros(n,1);
mom_total_thrust=zeros(n,1);
mom_total_torque=zeros(n,1);
theta_mat=zeros(n,17);
a_mat=zeros(n,17);
for i=1:n
% For all time steps in U...

    Uval=U(i);

    accumulatedthrust=0;
    accumulatedtorque=0;

```



```

mom_accumulatedthrust=0;
mom_accumulatedtorque=0;
for j=1:17
% For all blade sections...

    sigma_mark=(B*chord(j))/(2*pi*r(j));
    % Local solidity
    lambda_r=(Omega*r(j))/Uval;
    % Local speed ratio
    airfoil=foilnumbers(j);
    % Determines relevant airfoil number

    if airfoil==1
        a=0;
        a_mark=0;
        theta=atan((1-a)/((1+a_mark)*lambda_r));
        % Calculating the angle of the relative wind [rad]
        Cl=0;
        Cd=0.5;
    elseif airfoil==2
        a=0;
        a_mark=0;
        theta=atan((1-a)/((1+a_mark)*lambda_r));
        % Calculating the angle of the relative wind [rad]
        Cl=0;
        Cd=0.35;

    else

        counter=1;
        % Iteration counter set to zero
        a_mark_guess=0;
        a_guess=0;
        % Establish first guesses; a=0 and a'=0
        flag=0;
        while flag==0
            % Iterate to find a and a' for section...

            if counter>150
                disp('Warning: No convergence obtained before 500 iterations')
                a_new=1/3;
                a_mark_new=(1-3*a_new)/(4*a-1);
                break
                %return
            end

            theta=atan((1-a_guess)/((1+a_mark_guess)*lambda_r));
            % Calculating the angle of the relative wind [rad]
            alpha=theta-twist(j);
            % Calculating the angle of attack [rad]
            alphadeg=alpha*(180/pi);
            % Calculating the angle of attack to use in table [deg]

            theta_mat(i,j)=theta; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            aoacounter=1;
            % Angle of attack counter
            aoaflag=0;
            while aoaflag==0;
                angle=foilarray{airfoil,1}(aoacounter,1);
                nextangle=foilarray{airfoil,1}(aoacounter+1,1);
                if alphadeg==angle
                    % If exact match for angle in table...
                    Cl=foilarray{airfoil,1}(aoacounter,2);

```

```

        Cd=foilarray{airfoil,1}(aoacounter,3);
        aoaflag=1;
    elseif alphadeg==nextangle
    % If exact match for next angle in table...
        Cl=foilarray{airfoil,1}(aoacounter+1,2);
        Cd=foilarray{airfoil,1}(aoacounter+1,3);
        aoaflag=1;
    elseif (angle<alphadeg && nextangle>alphadeg) ||...
        (angle>alphadeg && nextangle<alphadeg)
    % If angle of attack inbetween angle and next angle in table,
    % then interpolate...
        Cl_1=foilarray{airfoil,1}(aoacounter,2);
        Cl_2=foilarray{airfoil,1}(aoacounter+1,2);
        Cd_1=foilarray{airfoil,1}(aoacounter,3);
        Cd_2=foilarray{airfoil,1}(aoacounter+1,3);
        Cl=((alphadeg-angle)/(nextangle-angle))*(Cl_2-Cl_1)+Cl_1;
        Cd=((alphadeg-angle)/(nextangle-angle))*(Cd_2-Cd_1)+Cd_1;
        aoaflag=1;
    elseif aoacounter>size(foilarray{airfoil,1},1)
        disp('Error: Angle of attack not recognized')
        return
    end
    aoacounter=aoacounter+1;
    % Update angle of attack counter
end

a_new=1/(1+((4*sin(theta)^2)/(sigma_mark*Cl*cos(theta))));
a_mark_new=1/(((4*cos(theta))/(sigma_mark*Cl))-1);

%convergencearray_a{i,1}(j,counter)=a_new-a_guess;
%convergencearray_a_mark{i,1}(j,counter)=a_mark_new-a_mark_guess;
% Arrays that could be used to check convergence values of a and a'

if (abs(a_mark_new-a_mark_guess)<maxdev) &&...
    (abs(a_new-a_guess)<maxdev)
    flag=1;
end
% Check convergence

a_guess=a_new;
a_mark_guess=a_mark_new;
% Update guesses
counter=counter+1;
% Update induction factor counter

end

a=a_new;
a_mark=a_mark_new;

a_mat(i,j)=a;

end

sectionthrust=sigma_mark*pi*rho*...
    (((Uval^2)*((1-a)^2)/(sin(theta)^2))*...
    ((Cl*cos(theta)+(Cd*sin(theta)))*r(j)*dr(j));
% Calculating thrust contribution from sections using blade
% element theory [N]

sectiontorque=sigma_mark*pi*rho*...
    (((Uval^2)*((1-a)^2)/(sin(theta)^2))*...
    ((Cl*sin(theta)+(Cd*cos(theta)))*(r(j)^2)*dr(j));
% Calculating torque contribution from sections using blade

```

```

% element theory [Nm]

accumulatedthrust=accumulatedthrust+sectionthrust;
accumulatedtorque=accumulatedtorque+sectiontorque;
% Accumulating thrust and torque for all 17 sections for each
% time step using blade element theory

mom_sectionthrust=rho*(Uval^2)*4*a*(1-a)*pi*r(j)*dr(j);
% Calculating thrust contribution from sections using
% momentum theory [N]
mom_sectiontorque=4*a_mark*(1-a)*rho*Uval*pi*(r(j)^3)*Omega*dr(j);
% Calculating torque contribution from sections using
% momentum theory [Nm]

mom_accumulatedthrust=mom_accumulatedthrust+mom_sectionthrust;
mom_accumulatedtorque=mom_accumulatedtorque+mom_sectiontorque;
% Accumulating thrust and torque for all 17 sections for each
% time step using momentum theory

end

total_thrust(i,1)=accumulatedthrust;
total_torque(i,1)=accumulatedtorque;

mom_total_thrust(i,1)=mom_accumulatedthrust;
mom_total_torque(i,1)=mom_accumulatedtorque;

end
toc
end

```


7.5 Attachment 5:

Matlab script used to estimate the turbulence AR(1) parameter values for the load cases defined in section 3.4.

```

%
% John Magne Hembre 2014
%
% Script written to determine AR(1) parameters for the turbulence in the
% 15 load cases.
%

clc
close all

% INPUT:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

reclength=3000;      % [s]
timestep=0.0125;    % [s]
hubhght=90;         % [m]
v_hub=[2,4,6,8,10,12,14,16,18,20,22,24,26,28,30]';          % [m/s]
sigma=[0.994,1.204,1.414,1.624,1.834,2.044,2.254,2.464,2.674...
       2.884,3.094,3.304,3.514,3.724,3.934]';              % [m/s]

N=4000000;          % Number of points in spectra

if hubhght<60
    L_k=8.10*0.7*hubhght;
else
    L_k=8.10*0.7*60;
end

searchmin=0.997;    % Search from phi value...
searchmax=1;        % Search to phi value...
searchresolution=0.000001; % Increment of phi to find best fit

histres=30;         % Histogram resolution

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Estimating AR(1) parameters:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
time=linspace(0,reclength,reclength/timestep);
maxfreq=1/(2*timestep); % The nyquist frequency [Hz]
freqs=linspace(0,maxfreq,N);
dfreq=abs(freqs(2)-freqs(1));
maxomega=maxfreq*2*pi;
omegas=linspace(0,maxomega,N);
domega=abs(omegas(2)-omegas(1));

S_f=zeros(15,1);
var_fromspec=zeros(15,1);
for i=1:15
    for j=1:N
        % Variance spectrum given by hertz:
        S_f(i,j)=(4*sigma(i)^2*(L_k/v_hub(i))./...
            (1+(6.*freqs(j)*(L_k/v_hub(i))))).^5/3);
        % Calculating variances from spectrum:
        var_fromspec(i,1)=var_fromspec(i,1)+S_f(i,j)*dfreq;
    end
end

% Variance spectrum given by angular frequency:
S_omega=S_f./(2*pi);

% Iterating to estimate phi from the spectrum above:
phinum=(searchmax-searchmin)/searchresolution;
% Number of values of phi to try
phivals=linspace(searchmin+searchresolution,...

```

```

searchmax-searchresolution,phinum);

S_bestfit=zeros(15,N);
phi_bestfit=zeros(15,1);
var_bestfit=zeros(15,1);
for i=1:15
disp(['Curve fitting AR(1) spectrum for LC number ',...
num2str(i), '...'])
C_0=sigma(i)^2;
least_sqerror=0;
for j=1:phinum
currentvar=C_0*(1-phivals(j)^2);
S_currentphi=(2*timestep*currentvar)/(2*pi.*(1-2*phivals(j).*...
cos(omegas*timestep)+phivals(j)^2));

sqerror=0;
for k=1:N/5
% (N/5 for efficiency;
% S_omega ~ S_currentphi ~ 0 for higher frequencies)
sqerror=sqerror+(S_omega(k)-S_currentphi(k))^2;
end

if j==1||sqerror<least_sqerror
least_sqerror=sqerror;
S_bestfit(i,:)=S_currentphi;
phi_bestfit(i)=phivals(j);
var_bestfit(i)=currentvar;
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('\n')

% Simulating time series for comparison:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AR_sim=zeros(15,length(time));
kai_sim=zeros(15,length(time));
for i=1:15
disp(['Generating a ', num2str(reclength),...
' s AR(1) simulation of LC number ',...
num2str(i), '...'])

% Simulating a time series using the AR(1) properties found by curve
% fitting the Kaimal spectra:
AR_sim(i,1)=normrnd(0,sigma(i)); % Simulate initial value
for j=2:length(time)
AR_sim(i,j)=phi_bestfit(i)*AR_sim(i,j-1)+...
normrnd(0,sqrt(var_bestfit(i)));
end
end
end

```


7.6 Attachment 6:

Flexibility matrix, f , of the discretized reference tower described in section 3.1.1. [m/N]

Table 11: Flexibility matrix of the discretized reference tower. $[m/N]$

DOF #, i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5,000 E-11	1,200 E-10	1,900 E-10	2,600 E-10	3,300 E-10	4,000 E-10	4,700 E-10	5,400 E-10	6,100 E-10	6,800 E-10	7,500 E-10	8,200 E-10	8,900 E-10	9,600 E-10	1,0300 E-09	1,1000 E-09	1,1700 E-09	1,2400 E-09	1,3100 E-09	1,3800 E-09
2	1,200 E-10	3,800 E-10	6,700 E-10	9,500 E-10	1,2400 E-09	1,5300 E-09	1,8100 E-09	2,1000 E-09	2,3900 E-09	2,6700 E-09	2,9600 E-09	3,2500 E-09	3,5300 E-09	3,8200 E-09	4,1100 E-09	4,3900 E-09	4,6800 E-09	4,9700 E-09	5,2500 E-09	5,5400 E-09
3	1,900 E-10	6,700 E-10	1,3000 E-09	1,9600 E-09	2,6200 E-09	3,2800 E-09	3,9400 E-09	4,6000 E-09	5,2600 E-09	5,9200 E-09	6,5800 E-09	7,2400 E-09	7,9000 E-09	8,5600 E-09	9,2300 E-09	9,8900 E-09	1,0550 E-08	1,1210 E-08	1,1870 E-08	1,2530 E-08
4	2,600 E-10	9,500 E-10	1,9600 E-09	3,1300 E-09	4,3400 E-09	5,5400 E-09	6,7400 E-09	7,9500 E-09	9,1500 E-09	1,0350 E-08	1,1550 E-08	1,2760 E-08	1,3960 E-08	1,5160 E-08	1,6370 E-08	1,7570 E-08	1,8770 E-08	1,9980 E-08	2,1180 E-08	2,2380 E-08
5	3,300 E-10	1,2400 E-09	2,6200 E-09	4,3400 E-09	6,2300 E-09	8,1600 E-09	1,0090 E-08	1,2010 E-08	1,3940 E-08	1,5870 E-08	1,7790 E-08	1,9720 E-08	2,1650 E-08	2,3580 E-08	2,5500 E-08	2,7430 E-08	2,9360 E-08	3,1280 E-08	3,3210 E-08	3,5140 E-08
6	4,000 E-10	1,5300 E-09	3,2800 E-09	5,5400 E-09	8,1600 E-09	1,0970 E-08	1,3820 E-08	1,6660 E-08	1,9510 E-08	2,2360 E-08	2,5200 E-08	2,8050 E-08	3,0900 E-08	3,3740 E-08	3,6590 E-08	3,9440 E-08	4,2280 E-08	4,5130 E-08	4,7980 E-08	5,0820 E-08
7	4,700 E-10	1,8100 E-09	3,9400 E-09	6,7400 E-09	1,0090 E-08	1,3820 E-08	1,7760 E-08	2,1730 E-08	2,5710 E-08	2,9690 E-08	3,3660 E-08	3,7640 E-08	4,1620 E-08	4,5590 E-08	4,9570 E-08	5,3550 E-08	5,7520 E-08	6,1500 E-08	6,5480 E-08	6,9450 E-08
8	5,400 E-10	2,1000 E-09	4,6000 E-09	7,9500 E-09	1,2010 E-08	1,6660 E-08	2,1730 E-08	2,7030 E-08	3,2360 E-08	3,7700 E-08	4,3030 E-08	4,8370 E-08	5,3700 E-08	5,9030 E-08	6,4370 E-08	6,9700 E-08	7,5040 E-08	8,0370 E-08	8,5700 E-08	9,1040 E-08
9	6,100 E-10	2,3900 E-09	5,2600 E-09	9,1500 E-09	1,3940 E-08	1,9510 E-08	2,5710 E-08	3,2360 E-08	3,9260 E-08	4,6200 E-08	5,3130 E-08	6,0070 E-08	6,7010 E-08	7,3950 E-08	8,0890 E-08	8,7820 E-08	9,4760 E-08	1,0170 E-07	1,0864 E-07	1,1558 E-07
10	6,800 E-10	2,6700 E-09	5,9200 E-09	1,0350 E-08	1,5870 E-08	2,2360 E-08	2,9690 E-08	3,7700 E-08	4,6200 E-08	5,4960 E-08	6,3770 E-08	7,2570 E-08	8,1380 E-08	9,0190 E-08	9,9000 E-08	1,0781 E-07	1,1662 E-07	1,2542 E-07	1,3423 E-07	1,4304 E-07
11	7,500 E-10	2,9600 E-09	6,5800 E-09	1,1550 E-08	1,7790 E-08	2,5200 E-08	3,3660 E-08	4,3030 E-08	5,3130 E-08	6,3770 E-08	7,4680 E-08	8,5650 E-08	9,6620 E-08	1,0759 E-07	1,1855 E-07	1,2952 E-07	1,4049 E-07	1,5146 E-07	1,6242 E-07	1,7339 E-07
12	8,200 E-10	3,2500 E-09	7,2400 E-09	1,2760 E-08	1,9720 E-08	2,8050 E-08	3,7640 E-08	4,8370 E-08	6,0070 E-08	7,2570 E-08	8,5650 E-08	9,9040 E-08	1,1248 E-07	1,2592 E-07	1,3936 E-07	1,5280 E-07	1,6624 E-07	1,7968 E-07	1,9312 E-07	2,0656 E-07
13	8,900 E-10	3,5300 E-09	7,9000 E-09	1,3960 E-08	2,1650 E-08	3,0900 E-08	4,1620 E-08	5,3700 E-08	6,7010 E-08	8,1380 E-08	9,6620 E-08	1,1248 E-07	1,2867 E-07	1,4493 E-07	1,6118 E-07	1,7744 E-07	1,9369 E-07	2,0995 E-07	2,2620 E-07	2,4246 E-07
14	9,600 E-10	3,8200 E-09	8,5600 E-09	1,5160 E-08	2,3580 E-08	3,3740 E-08	4,5590 E-08	5,9030 E-08	7,3950 E-08	9,0190 E-08	1,0759 E-07	1,2592 E-07	1,4493 E-07	1,6431 E-07	1,8375 E-07	2,0319 E-07	2,2263 E-07	2,4208 E-07	2,6152 E-07	2,8096 E-07
15	1,0300 E-09	4,1100 E-09	9,2300 E-09	1,6370 E-08	2,5500 E-08	3,6590 E-08	4,9570 E-08	6,4370 E-08	8,0890 E-08	9,9000 E-08	1,1855 E-07	1,3936 E-07	1,6118 E-07	1,8375 E-07	2,0672 E-07	2,2975 E-07	2,5279 E-07	2,7583 E-07	2,9886 E-07	3,2190 E-07
16	1,1000 E-09	4,3900 E-09	9,8900 E-09	1,7570 E-08	2,7430 E-08	3,9440 E-08	5,3550 E-08	6,9700 E-08	8,7820 E-08	1,0781 E-07	1,2952 E-07	1,5280 E-07	1,7744 E-07	2,0319 E-07	2,2975 E-07	2,5675 E-07	2,8383 E-07	3,1091 E-07	3,3799 E-07	3,6507 E-07
17	1,1700 E-09	4,6800 E-09	1,0550 E-08	1,8770 E-08	2,9360 E-08	4,2280 E-08	5,7520 E-08	7,5040 E-08	9,4760 E-08	1,1662 E-07	1,4049 E-07	1,6624 E-07	1,9369 E-07	2,2263 E-07	2,5279 E-07	2,8383 E-07	3,1535 E-07	3,4696 E-07	3,7857 E-07	4,1018 E-07
18	1,2400 E-09	4,9700 E-09	1,1210 E-08	1,9980 E-08	3,1280 E-08	4,5130 E-08	6,1500 E-08	8,0370 E-08	1,0170 E-07	1,2542 E-07	1,5146 E-07	1,7968 E-07	2,0995 E-07	2,4208 E-07	2,7583 E-07	3,1091 E-07	3,4696 E-07	3,8354 E-07	4,2022 E-07	4,5690 E-07
19	1,3100 E-09	5,2500 E-09	1,1870 E-08	2,1180 E-08	3,3210 E-08	4,7980 E-08	6,5480 E-08	8,5700 E-08	1,0864 E-07	1,3423 E-07	1,6242 E-07	1,9312 E-07	2,2620 E-07	2,6152 E-07	2,9886 E-07	3,3799 E-07	3,7857 E-07	4,2022 E-07	4,6246 E-07	5,0480 E-07
20	1,3800 E-09	5,5400 E-09	1,2530 E-08	2,2380 E-08	3,5140 E-08	5,0820 E-08	6,9450 E-08	9,1040 E-08	1,1558 E-07	1,4304 E-07	1,7339 E-07	2,0656 E-07	2,4246 E-07	2,8096 E-07	3,2190 E-07	3,6507 E-07	4,1018 E-07	4,5690 E-07	5,0480 E-07	5,5334 E-07

7.7 Attachment 7:

Matlab script used to derive the modal properties of the reference tower described in section 3.1.1.

```

%
% John Magne Hembre 2014
%
% Script written to estimate mode shapes and natural frequencies of
% the onshore tower structure described by Jonkman in "Definition of a
% 5-MW Reference Wind Turbine for Offshore System Development" (2009)
%
% Input (in workspace):
% - Stiffnes matrix; Kmat (inverse of flexibility matrix)
% - Mass matrix; Mmat
% - Vector containing height of DOF's; HDOF
%
% Input:
zeta=0.01; % Structural damping ratio (all modes)
M=Mmat;
K=Kmat;
%
clc
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Obtaining natural frequencies and mode shapes:

eigenmatrix=M\K; % eigenmatrix = inv(M)*K
% Solving the eigensystem:
[eigenvecs,lambdas]=eig(eigenmatrix);
% Considering the six first eigenvalues:
omegas=zeros(1,6);
modes=zeros(21,6);
sortedeigenvecs=zeros(20,20);
sortedeigenvals=sort(max(lambdas));
for i=1:20
    for j=1:20
        if sortedeigenvals(i)==lambdas(j,j);
            sortedeigenvecs(:,i)=eigenvecs(:,j);
        end
    end
end
maxvals=max(abs(sortedeigenvecs));
for i=1:9
    omegas(1,i)=sqrt(sortedeigenvals(i));
    for j=1:21
        if j==1
            modes(j,i)=0;
        else
            modes(j,i)=sortedeigenvecs(j-1,i)/maxvals(i);
        end
    end
end
end

omegal=omegas(3);
omega2=omegas(4);
omega3=omegas(5);

phi1=modes(:,3);
phi2=modes(:,4)*(-1);
phi3=modes(:,5)*(-1);

figure(1)
plot(phi1,HDOF,'-bo','LineWidth',3,...
'MarkerSize',10,'MarkerFaceColor','b');
hold on
plot(phi2,HDOF,'-ro','LineWidth',3,...
'MarkerSize',10,'MarkerFaceColor','r')
plot(phi3,HDOF,'-mo','LineWidth',3,...

```

```

    'MarkerSize',10,'MarkerFaceColor','m')
axis([-1 1.2 0 90])
xhandle=xlabel('Normalized modal amplitude, \phi [-]');
yhandle=ylabel('DOF height [m]');
legendhandle=legend({'f_{Nat,1} = ',num2str(omega1/(2*pi)),' Hz'},...
    ['f_{Nat,2} = ',num2str(omega2/(2*pi)),' Hz']...
    ['f_{Nat,3} = ',num2str(omega3/(2*pi)),' Hz']},'Location','EastOutside');
grid on
hold off
set(gca,'fontsize',17)
set(xhandle,'FontSize',17)
set(yhandle,'FontSize',17)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Obtaining modal masses and stiffnesses:

phi1=modes(2:21,3);
phi2=modes(2:21,4)*(-1);
phi3=modes(2:21,5)*(-1);

m_mass_1=(phi1.').*M*phi1;
m_mass_2=(phi2.').*M*phi2;
m_mass_3=(phi3.').*M*phi3;

m_stif_1=(phi1.').*K*phi1;
m_stif_2=(phi2.').*K*phi2;
m_stif_3=(phi3.').*K*phi3;

m_damp_1=2*m_mass_1*omega1*zeta;
m_damp_2=2*m_mass_2*omega2*zeta;
m_damp_3=2*m_mass_3*omega3*zeta;

m_stif_ctrl_1=omega1^2*m_mass_1;
m_stif_ctrl_2=omega2^2*m_mass_2;
m_stif_ctrl_3=omega3^2*m_mass_3;

```


7.8 Attachment 8:

Example of a TurbSim input file used to simulate a fluctuating wind field with a reference wind velocity of 12 m/s and turbulence class B.

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

-----Runtime Options-----

2692473	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)
True	WrADFF	- Output full-field time-series data in TurbSim/AeroDyn form? (Generates Rootname.bts)
False	WrBLFF	- Output full-field time-series data in BLADED/AeroDyn form? (Generates RootName.wnd)
True	WrADTWR	- Output tower time-series data? (Generates RootName.twr)
False	WrFMFFF	- 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-----

16	NumGrid_Z	- Vertical grid-point matrix dimension
16	NumGrid_Y	- Horizontal grid-point matrix dimension
0.02500000	TimeStep	- Time step [seconds]
660.00000000	AnalysisTime	- Length of analysis time series [seconds] (program will add time if necessary: AnalysisTime = MAX(AnalysisTime, UsableTime+GridWidth/MeanHHWS))
660.00000000	UsableTime	- Usable length of output time series [seconds] (program will add GridWidth/MeanHHWS seconds)
88.15000000	HubHt	- Hub height [m] (should be > 0.5*GridHeight)
160.00000000	GridHeight	- Grid height [m]
160.00000000	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"))
14.60	IECturbc	- IEC turbulence characteristic ("A", "B", "C" or the turbulence intensity in percent) ("KHTST" 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")
88.15000000	RefHt	- Height of the reference wind speed [m]
12.00000000	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")
0.00	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-----

```

"M:\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          DistSc1      - 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.)
30.0         CTStartTime  - Minimum start time for coherent structures in RootName.cts
                    [seconds]

```

```

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

```


7.9 Attachment 9:

Matlab script used to analyze FAST simulation output and estimate parameters in the simplified thrust model.

```

%
% John Magne Hembre 2014
%
% Script written to analyze output from time domain simulations in FAST
% and estimate parameters for the simplified thrust force model.
%
% Input from workspace used in this script:
% - Thrust matrix "Thrust_classB" from FAST
% - Thrust matrix under mean wind loading "Thrust_meanwind" from FAST
% - Wind matrix "Wind_classB" from FAST
% - Vector phi_bestfit over AR(1) turbulence phi values
% - Vector var_bestfit over AR(1) turbulence noise variances
%
clc
close all

% Wind speeds:
V_ref_vec=[2,4,6,8,10,12,14,16,18,20,22,24,26,28,30];

% Time vector:
timestep=0.0125;
time=linspace(0,600,600/timestep);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mean of thrust:
meanvec=mean(Thrust_classB);
meanthrust_meanwind=mean(Thrust_meanwind);

% Search properties to find best C_simp,1 for section 1:
minsearch_meansec_1=40000;           % Search from...
maxsearch_meansec_1=42000;           % Search to...
N_search_meansec_1=2000;             % Number of points
C_simp1_vec=linspace(minsearch_meansec_1,maxsearch_meansec_1,...
    N_search_meansec_1);

% Search for best fit for section 1:
meanvals1=meanvec(1:5);
Vrefvals1=V_ref_vec(1:5);
Vrefvals1(6)=11;
for i=1:N_search_meansec_1
    sqerror1=0;
    calcmean1=C_simp1_vec(i).*Vrefvals1.^(1.2);
    for j=1:5
        sqerror1=sqerror1+(meanvals1(j)-calcmean1(j))^2;
    end

    if i==1||sqerror1<least_sqerror1
        least_sqerror1=sqerror1;
        C_simp1_bestfit=C_simp1_vec(i);
        calcmean1_bestfit=calcmean1;
    end
end
end

% Search properties to find best C_simp,2 for section 2:
minsearch_meansec_2=5400000;         % Search from...
maxsearch_meansec_2=5500000;         % Search to...
N_search_meansec_2=100000;           % Number of points
C_simp2_vec=linspace(minsearch_meansec_2,maxsearch_meansec_2,...
    N_search_meansec_2);

% Search for best fit for section 2:
meanvals2=meanvec(6:15);
Vrefvals2=V_ref_vec(6:15);
Vrefvals2=[11 Vrefvals2];

```

```

for i=1:N_search_meansec_2
    sqerror2=0;
    calcmean2=C_simp2_vec(i).*Vrefvals2.^(-0.85);
    for j=2:length(calcmean2)
        sqerror2=sqerror2+(meanvals2(j-1)-calcmean2(j))^2;
    end

    if i==1||sqerror2<least_sqerror2
        least_sqerror2=sqerror2;
        C_simp2_bestfit=C_simp2_vec(i);
        calcmean2_bestfit=calcmean2;
    end
end

% Obtaining turbulence time series:
meanwindvec=mean(Wind_classB);
Turb_classB=zeros(48000,15);
for i=1:15
    for j=1:48000
        Turb_classB(j,i)=Wind_classB(j,i)-meanwindvec(i);
    end
end

% Obtaining force fluctuation time series:
Forcefluct_classB=zeros(48000,15);
for i=1:15
    for j=1:48000
        Forcefluct_classB(j,i)=Thrust_classB(j,i)-meanvec(i);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Finding C_simp(Vref) from simulation time series, T(t) = C_simp(Vref)*u:
varvec=var(Forcefluct_classB);
Csearch=linspace(10000,70000,60000);
Cvec=zeros(1,15);
for i=1:15
    for j=1:length(Csearch)
        curr_C=Csearch(j);
        curr_thrust=curr_C.*Turb_classB(:,i);
        curr_var=var(curr_thrust);
        curr_diff=abs(curr_var-varvec(i));

        if j==1||curr_diff<best_diff
            best_diff=curr_diff;
            Cvec(i)=curr_C;
        end
    end
    Cvec(i)
end

% Search properties to find best C_simp_turb for section 1:
minsearch_turbsec_1=5000;           % Search from...
maxsearch_turbsec_1=10000;        % Search to...
N_search_turbsec_1=10000;         % Number of points
C_simp1_turb_vec=linspace(minsearch_turbsec_1,maxsearch_turbsec_1,...
    N_search_turbsec_1);

% Search for best fit for section 1:
Cvals1=Cvec(1:5);
Vrefvals1=V_ref_vec(1:5);
Vrefvals1(6)=11;
for i=1:N_search_turbsec_1
    sqerror1=0;

```

```

    calcval1=C_simp1_turb_vec(i).*Vrefvals1.^(1);
    for j=1:5
        sqerror1=sqerror1+(Cvals1(j)-calcval1(j))^2;
    end

    if i==1||sqerror1<least_sqerror1
        least_sqerror1=sqerror1;
        C_simp1_turb_bestfit=C_simp1_turb_vec(i);
        calcval1_bestfit=calcval1;
    end
end

% Search properties to find best C_simp_turb for section 2:
minsearch_turbsec_2=1500000;           % Search from...
maxsearch_turbsec_2=1800000;         % Search to...
N_search_turbsec_2=300000;           % Number of points
C_simp2_turb_vec=linspace(minsearch_turbsec_2,maxsearch_turbsec_2,...
    N_search_turbsec_2);

% Search for best fit for section 2:
Cvals2=Cvec(6:15);
Vrefvals2=V_ref_vec(6:15);
Vrefvals2=[11 Vrefvals2];
for i=1:N_search_turbsec_2
    sqerror2=0;
    calcval2=C_simp2_turb_vec(i).*Vrefvals2.^(-1.35);
    for j=2:length(calcval2)
        sqerror2=sqerror2+(Cvals2(j-1)-calcval2(j))^2;
    end

    if i==1||sqerror2<least_sqerror2
        least_sqerror2=sqerror2;
        C_simp2_turb_bestfit=C_simp2_turb_vec(i);
        calcval2_bestfit=calcval2;
    end
end

figure(101)
plot(V_ref_vec,Cvec,'b*','LineWidth',1.5,'MarkerSize',15)
hold on
axis([1 31 0 70000])
plot(Vrefvals1,calcval1_bestfit,'LineWidth',3,'Color','r')
plot(Vrefvals2,calcval2_bestfit,'LineWidth',3,'Color','r')
xhandle=xlabel('V_{ref} [m/s]');
yhandle=ylabel('C_s*V_{ref}^\beta [kg/s]');
legendhandle=legend...
    (['C_s*V_{ref}^\beta estimated from' 10 'FAST simulations'],...
    ['C_s*V_{ref}^\beta from the' 10 'simplified load model']},...
    'Location','EastOutside');
grid on
hold off
set(gca,'fontsize',17)
set(xhandle,'FontSize',17)
set(yhandle,'FontSize',17)

% Comparing model and simulated thrust:
C_modelvec=zeros(1,15);
for i=1:5
    for j=1:48000
        total_thrust_model(j,i)=calcmean1_bestfit(i)+...
            calcval1_bestfit(i).*Turb_classB(j,i);

        C_modelvec(i)=calcval1_bestfit(i);
    end
end

```

```

end
for i=1:10
    for j=1:48000
        total_thrust_model(j,i+5)=calcmean2_bestfit(i+1)-...
            calcval2_bestfit(i+1).*Turb_classB(j,i+5);
    end

        C_modelvec(i+5)=calcval2_bestfit(i+1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Frequency analysis:

fn=1/(2*0.0125);
% Nyquist frequency

% Plotting frequency content of force:
num=2^nextpow2(48000);
% Number of points in the DFT (next power of 2 from length of the
% signal).
points_considered = ceil((num+1)/2);
freq=(0:points_considered-1)*2*fn/num;
% Vector of the frequencies considered in DFT, in Hz.
% delta_frequency=(2*fn)/n yields a maximal frequency of fn, since
% points_considered = n/2.
omegafreq=freq*2*pi;
% Vector of the angular frequencies considered in DFT, in rad/s.

PSDmat=zeros(15,points_considered);
PSDmodelmat=zeros(15,points_considered);

for i=1:15
    FFTsignal_T=fft(Thrust_classB(:,i),num);
    % Fast Fourier Transform of the signal.
    FFTsignal_T=FFTsignal_T(1:points_considered);
    % Disregard the upper half of FFTsignal due to symmetry about the
    % Nyquist frequency.
    Mag_signal_T=abs(FFTsignal_T);
    Mag_signal_T=Mag_signal_T*2;
    % Multiply the magnitude of the signal by two since half the signal
    % was disregarded.
    Mag_signal_T=Mag_signal_T.^2;
    % Square the magnitude of the signal to obtain the power spectrum
    Mag_signal_T(1)=Mag_signal_T(1)/2;
    Mag_signal_T(length(Mag_signal_T))=Mag_signal_T(length(Mag_signal_T))/2;
    % Adjust unique endpoints
    Mag_signal_T=Mag_signal_T/48000;
    % Scaling magnitude of the FFT so that it is not a function of the
    % length of the signal.

    FFTsignal_model=fft(total_thrust_model(:,i),num);
    % Fast Fourier Transform of the signal.
    FFTsignal_model=FFTsignal_model(1:points_considered);
    % Disregard the upper half of FFTsignal due to symmetry about the
    % Nyquist frequency.
    Mag_signal_model=abs(FFTsignal_model);
    Mag_signal_model=Mag_signal_model*2;
    % Multiply the magnitude of the signal by two since half the signal
    % was disregarded.
    Mag_signal_model=Mag_signal_model.^2;
    % Square the magnitude of the signal to obtain the power spectrum
    Mag_signal_model(1)=Mag_signal_model(1)/2;
    Mag_signal_model(length(Mag_signal_model))=...
        Mag_signal_model(length(Mag_signal_model))/2;

```

```

% Adjust unique endpoints
Mag_signal_model=Mag_signal_model/48000;
% Scaling magnitude of the FFT so that it is not a function of the
% length of the signal.

% Smoothing signals using a lowpass Butterworth filter:
[b,a]=butter(5,0.1,'low');
% 5th order lowpass Butterworth filter with normalized cut-off
% frequency of 0.06 (for deterministic LC)
filteredPSD_T=filtfilt(b,a,Mag_signal_T);
filteredPSD_model=filtfilt(b,a,Mag_signal_model);

PSDmat(i,:)=filteredPSD_T;
PSDmodelmat(i,:)=filteredPSD_model;
end

% Calculating the analytical variance spectra of the force fluctuations
% using the load model and an AR(1) variance spectra for turbulence:

S_turb_AR=zeros(15,length(freq));
S_forcefluct_AR=zeros(15,length(freq));
for i=1:15
    phi=phi_bestfit(i,1);
    noisevar=var_bestfit(i,1);
    for j=1:length(freq)
        omega=2*pi*freq(j);
        S_turb_AR(i,j)=(noisevar*timestep/pi)*...
            (1/(1-2*phi*cos(omega*timestep)+phi^2));

        S_forcefluct_AR(i,j)=S_turb_AR(i,j)*(C_modelvec(1,i))^2;
    end
end

sigma=[0.994,1.204,1.414,1.624,1.834,2.044,2.254,2.464,2.674...
        2.884,3.094,3.304,3.514,3.724,3.934]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculating sample autocovariances:

taumax=600; % [s]
n=taumax/0.0125; % [-]

FASTsample=Thrust_classB(:,8);
mean_FASTsample=mean(FASTsample);
var_FASTsample=var(FASTsample);
MODsample=total_thrust_model(:,8);
mean_MODsample=mean(MODsample);
var_MODsample=var(MODsample);
C_tau_FASTsample=zeros(1,n);
C_tau_MODsample=zeros(1,n);
for h=1:n
    Sum_FAST=0;
    Sum_MOD=0;
    for t=1:(n-h)
        Sum_FAST=Sum_FAST+(FASTsample(t+h,1)-mean_FASTsample)*...
            (FASTsample(t)-mean_FASTsample);
        Sum_MOD=Sum_MOD+(MODsample(t+h,1)-mean_MODsample)*...
            (MODsample(t)-mean_MODsample);
    end
    C_tau_FASTsample(h)=Sum_FAST/n;
    C_tau_MODsample(h)=Sum_MOD/n;
end
end

```


7.10 Attachment 10:

Plots of the estimated turbulence AR(1) spectra with parameters from table 8, and their corresponding Kaimal spectra.

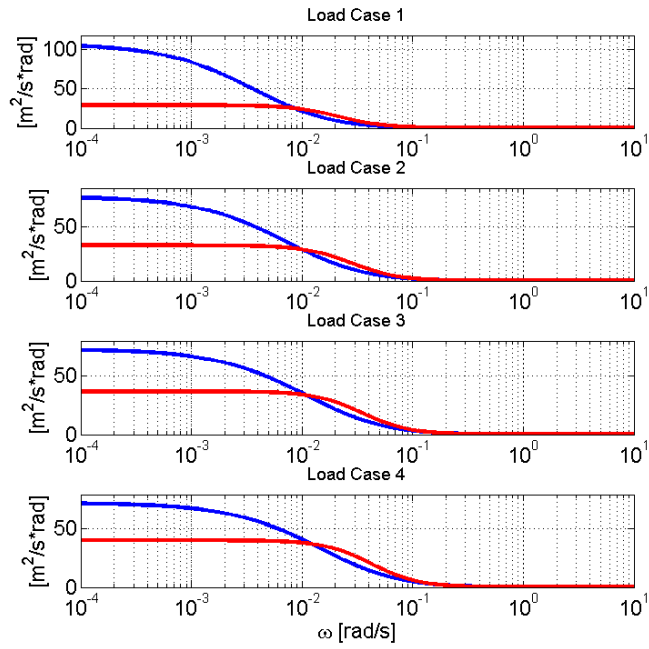


Figure 60: AR(1) wind turbulence spectra for $\Delta t = 0.0125\text{ s}$ and $\Delta t = 0.05\text{ s}$ (in red), and the corresponding Kaimal spectra (in blue) for load cases 1 to 4.

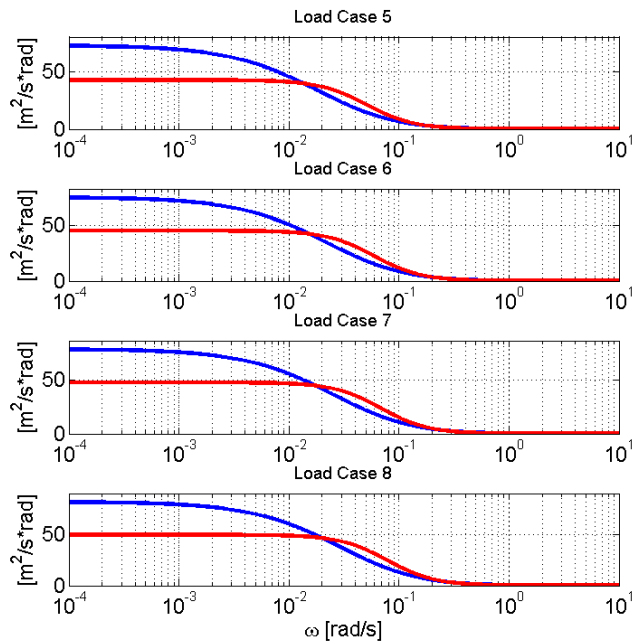


Figure 61: AR(1) wind turbulence spectra for $\Delta t = 0.0125\text{ s}$ and $\Delta t = 0.05\text{ s}$ (in red), and the corresponding Kaimal spectra (in blue) for load cases 5 to 8.

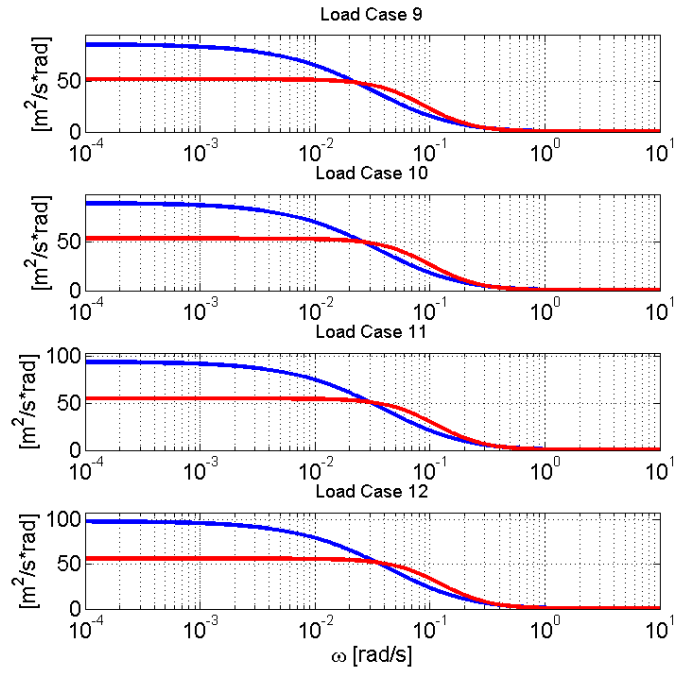


Figure 62: AR(1) wind turbulence spectra for $\Delta t = 0.0125 s$ and $\Delta t = 0.05 s$ (in red), and the corresponding Kaimal spectra (in blue) for load cases 9 to 12.

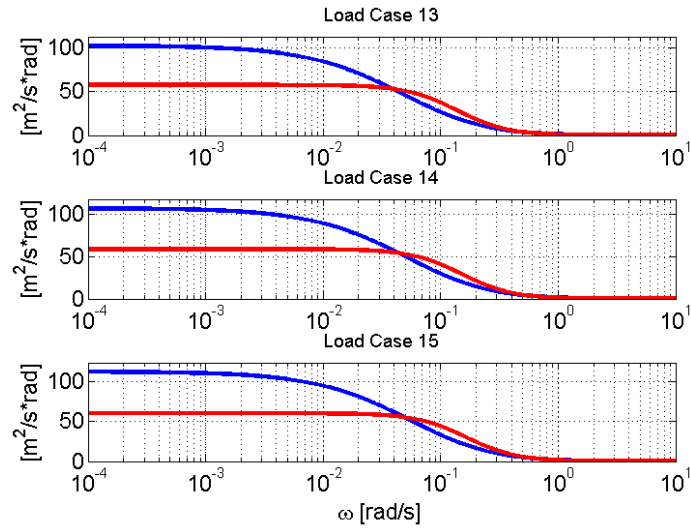


Figure 63: AR(1) wind turbulence spectra for $\Delta t = 0.0125 s$ and $\Delta t = 0.05 s$ (in red), and the corresponding Kaimal spectra (in blue) for load cases 13 to 15.

7.11 Attachment 11:

Tower top velocity response for the 15 load cases estimated from time domain simulations in FAST, and calculated analytically for the first mode using the simplified load model.

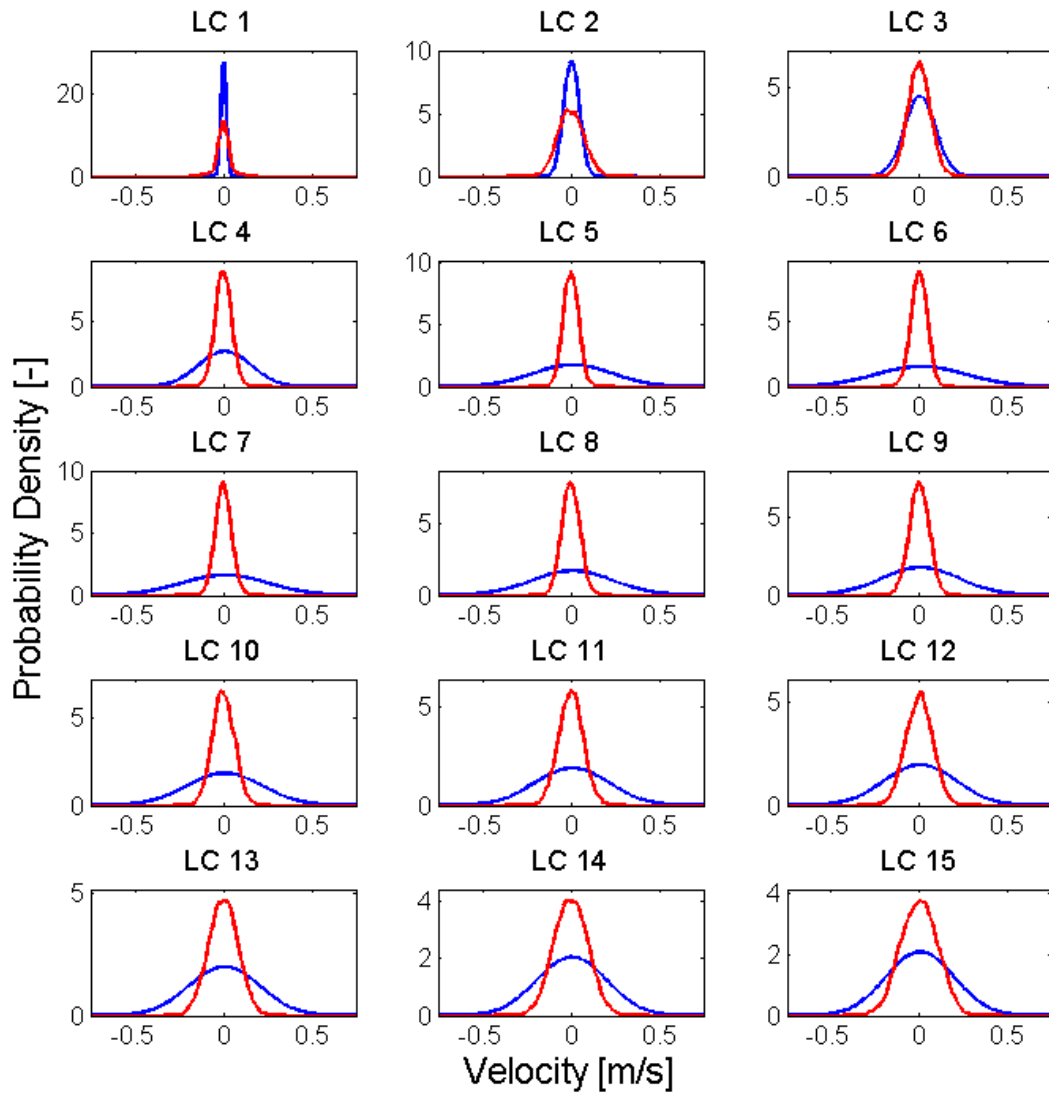


Figure 64: Tower top velocity for the 15 load cases estimated from time domain simulations in FAST (shown in red), and calculated analytically for the first mode using the simplified load model (shown in blue).

References

- [1] Chien-Su Hsu. *Cell-to-Cell Mapping: A Method of Global Analysis for Nonlinear Systems*. Springer, New York, 1987.
- [2] J. Jonkman et al. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical report, National Renewable Energy Laboratory, NREL, February 2009.
- [3] Jie Li and Jianbing Chen. Seismic Reliability Assessment of Nonlinear Stochastic Structures. In *13th World Conference on Earthquake Engineering, Vancouver, B.C. Canada, Paper No. 3015*, 2004.
- [4] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2nd edition, 2004.
- [5] Loren D. Lutes and Shahram Sarkani. *Random Vibrations: Analysis of Structural and Mechanical Systems*. Elsevier, Amsterdam, 2004.
- [6] Arvid Næss and Torgeir Moan. *Stochastic Dynamics of Marine Structures*. Cambridge University Press, New York, 2013.
- [7] David E. Newland. *An Introduction to Random Vibrations, Spectral & Wavelet Analysis*. Dover, Mineola, 3rd edition, 2005.
- [8] Erwin Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, Inc., 9th edition, 2006.
- [9] Joel P. Conte. Reader; Structural Reliability and Risk Analysis. University of California San Diego, 2013.
- [10] Jie Li and Jianbing Chen. *Stochastic Dynamics of Structures*. John Wiley & Sons (Asia), Singapore, 2009.
- [11] Jian-Qiao Sun and Albert C.J. Luo. *Global Analysis of Nonlinear Dynamics*. Springer, New York, 2012.
- [12] Jaap Wijker. *Random Vibrations in Spacecraft Structures Design*. Springer, New York, 2009.
- [13] Cord Böker. *Load Simulation and Local Dynamics of Support Structures for Offshore Wind Turbines*. PhD thesis, Leibniz Universität Hannover, 2009.
- [14] Einar N. Strømmen. *Theory of Bridge Aerodynamics*. Springer, New York, 2nd edition, 2010.
- [15] IEC 61400-1 Ed. 3, Wind Turbines- Part 1: Design Requirements. International Electrotechnical Commission (IEC), 2009.
- [16] Det Norske Veritas and the Wind Energy Department at Risø National Laboratory. *Guidelines for Design of Wind Turbines*. Jydsk Centraltrykkeri, 2nd edition, 2002.
- [17] J.G. McGowan J.F. Manwell and A.L. Rogers. *Wind Energy Explained*. John Wiley & Sons, Ltd., 2002.

- [18] Det Norske Veritas. DNV-RP-C205, Environmental Conditions and Environmental Loads. April 2007.
- [19] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, New York, 2nd edition, 2002.
- [20] Robert. D Cook et al. *Concepts And Applications of Finite Element Analysis*. John Wiley & Sons, Inc., 4th edition, 2002.
- [21] Anil K. Chopra. *Dynamics of Structures, Theory and Applications to Earthquake Engineering*. Prentice-Hall, Inc., 1995.
- [22] B. Basu P.J. Murtagh and B.M. Broderick. Simple models for natural frequencies and mode shapes of towers supporting utilities. *Computers & Structures*, 82:1745–1750, 2004.
- [23] Focus Software. Focus konstruksjon 2014, <http://www.focus.no>, 2014.
- [24] L. Kilcher B.J. Jonkman. TurbSim User’s Guide: Version 1.06.00. Technical report, National Renewable Energy Laboratory, NREL, September 2012.
- [25] Jason M. Jonkman and Marshall L. Buhl Jr. FAST User’s Guide. Technical report, National Renewable Energy Laboratory, NREL, August 2005.
- [26] T. Weller J. Levitas and J. Singer. Poincaré-like simple cell mapping for non-linear dynamical systems. *Journal of Sound and Vibration*, 176:641–662, 1994.
- [27] Christian Lubich Ernst Hairer and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, New York, 2nd edition, 2005.
- [28] Clayton T. Crowe et al. *Engineering Fluid Mechanics*. John Wiley & Sons, Inc., 9th edition, 2010.
- [29] J.R. Connell. *The Spectrum of Wind Speed Fluctuations Encountered by a Rotating Blade of a Wind Energy Conservation System: Observations and Theory*. Pacific Northwest Laboratory, 1981.

Front page image: London Array Limited