

Andrea Almenningen Solstad
Elin Linnerud Onstad

Comparison of Measured and Predicted Buffeting Response of the Hålogaland Bridge Using a Probabilistic Description of the Wind Field

Master's thesis in Civil and Environmental Engineering
Supervisor: Ole Andre Øiseth
Co-supervisor: Aksel Fenerci, Øyvind Wiig Petersen
June 2022

Andrea Almenningen Solstad
Elin Linnerud Onstad

Comparison of Measured and Predicted Buffeting Response of the Hålogaland Bridge Using a Probabilistic Description of the Wind Field

Master's thesis in Civil and Environmental Engineering
Supervisor: Ole Andre Øiseth
Co-supervisor: Aksel Fenerci, Øyvind Wiig Petersen
June 2022

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



MASTER THESIS 2022

SUBJECT AREA: Structural dynamics	DATE: June 9, 2022	NO. OF PAGES: 93 + 33
--------------------------------------	-----------------------	--------------------------

TITLE:

Comparison of Measured and Predicted Buffeting Response of the Hålogaland Bridge Using a Probabilistic Description of the Wind Field

Sammenlikning av målt og predikert buffeting-respons av Hålogalandsbrua ved bruk av en probabilistisk beskrivelse av vindfeltet

BY:

Andrea Almenningen Solstad
Elin Linnerud Onstad



SUMMARY:

Long-span suspension bridges are prone to wind-induced response due to the slenderness of these structures. Therefore, it is important to be able to predict the response due to wind. However, this is made difficult due to the variability of the wind. The aim of this thesis was therefore to assess how well the buffeting theory and wind field models were able to predict the wind-induced response of the Hålogaland Bridge.

In order to take the variability of the wind into account, measurement data from the monitoring system on the Hålogaland Bridge were processed and analysed. A covariance-driven stochastic subspace identification was performed. This was done in order to assess the accuracy of the numerical model of the bridge used in the response calculations. Information about the aerodynamic behaviour of the bridge, required to model the wind load, was obtained through wind tunnel tests. The buffeting response of the bridge was then predicted, using two different probabilistic approaches to obtain the turbulence parameters needed to model the wind load. First, the parameters were determined as specific percentiles from their probability distributions. Next, a probabilistic model was used to simulate the turbulence standard deviations. Finally, the predicted and measured response were compared.

The numerical model was found to be satisfactory for use in the response predictions. Furthermore, the analysis of the wind field revealed a large variability of the wind characteristics. In the probabilistic model, the turbulence standard deviations were modelled as correlated and lognormally distributed random variables, dependent on the mean wind speed and direction. The model managed to represent the true turbulence standard deviations, evident by the good agreement between the simulated and the measured parameters.

It was observed that the variability of the wind propagated into the measured response. The lateral and vertical response components were well predicted, while the torsional response was overestimated. However, the good agreement for the lateral response was unexpected. Furthermore, the probabilistic model of the turbulence standard deviations was able to translate the variability of the wind into the predicted response. It was found that the turbulence parameters greatly affected the predictions. Additionally, variables such as modal parameters and aerodynamic derivatives had a great impact.

Using a probabilistic approach, the buffeting theory and wind field models were mostly able to predict the response of the Hålogaland Bridge. However, determining accurate values of all variables was found to be of utmost importance in order to obtain satisfactory predictions.

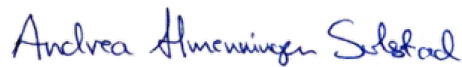
RESPONSIBLE TEACHER: Ole Andre Øiseth
SUPERVISOR(S): Ole Andre Øiseth, Aksel Fenerci, Øyvind Wiig Petersen
CARRIED OUT AT: Department of Structural Engineering, NTNU

Preface

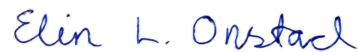
This master thesis in structural dynamics constitutes as the final submission of the MSc program in Civil and Environmental Engineering at the Norwegian University of Science and Technology (NTNU). It is written at the Department of Structural Engineering, during the spring semester of 2022. The topic of the thesis was suggested by our supervisor Ole Andre Øiseth.

We wish to thank our supervisor Ole Andre Øiseth for help and guidance throughout the semester. We would also like to express our sincere thanks to Aksel Fenerci, who have been an invaluable source of knowledge when it comes to everything related to wind. We also want to thank Øyvind Wiig Petersen for helpful advice regarding data processing, as well as providing us with an Abaqus model of the Hålogaland Bridge. Finally, we would like to thank Knut Andreas Kvåle, who has contributed with his extensive knowledge of Cov-SSI.

Trondheim, 2022-06-09



Andrea Almenningen Solstad



Elin Linnerud Onstad

Abstract

Long-span suspension bridges are prone to wind-induced response due to the slenderness of these structures. Therefore, it is important to be able to predict the response due to wind. However, this is made difficult due to the variability of the wind. The aim of this thesis was therefore to assess how well the buffeting theory and wind field models were able to predict the wind-induced response of the Hålogaland Bridge.

In order to take the variability of the wind into account, measurement data from the monitoring system on the Hålogaland Bridge were processed and analysed. A covariance-driven stochastic subspace identification was performed. This was done in order to assess the accuracy of the numerical model of the bridge used in the response calculations. Information about the aerodynamic behaviour of the bridge, required to model the wind load, was obtained through wind tunnel tests. The buffeting response of the bridge was then predicted, using two different probabilistic approaches to obtain the turbulence parameters needed to model the wind load. First, the parameters were determined as specific percentiles from their probability distributions. Next, a probabilistic model was used to simulate the turbulence standard deviations. Finally, the predicted and measured response were compared.

The numerical model was found to be satisfactory for use in the response predictions. Furthermore, the analysis of the wind field revealed a large variability of the wind characteristics. In the probabilistic model, the turbulence standard deviations were modelled as correlated and lognormally distributed random variables, dependent on the mean wind speed and direction. The model managed to represent the true turbulence standard deviations, evident by the good agreement between the simulated and the measured parameters.

It was observed that the variability of the wind propagated into the measured response. The lateral and vertical response components were well predicted, while the torsional response was overestimated. However, the good agreement for the lateral response was unexpected. Furthermore, the probabilistic model of the turbulence standard deviations was able to translate the variability of the wind into the predicted response. It was found that the turbulence parameters greatly affected the predictions. Additionally, variables such as modal parameters and aerodynamic derivatives had a great impact.

Using a probabilistic approach, the buffeting theory and wind field models were mostly able to predict the response of the Hålogaland Bridge. However, determining accurate values of all variables was found to be of utmost importance in order to obtain satisfactory predictions.

Sammendrag

Hengebruer med lange spenn er svært utsatt for vindindusert respons, på grunn av slankheten til disse konstruksjonene. Derfor er det viktig å kunne predikere responsen forårsaket av vind. Variabiliteten til vinden gjør det imidlertid vanskelig å gjøre nettopp dette. Formålet med denne oppgaven var derfor å vurdere hvor godt *buffeting*-teorien og vindfeltmodeller var i stand til å predikere den vindinduserte responsen til Hålogalandsbrua.

For å ta variabiliteten til vinden med i betraktning, ble data fra målesystemet på Hålogalandsbrua prosessert og analysert. En *covariance-driven stochastic subspace identification* ble utført. Dette ble gjort for å evaluere nøyaktigheten til den numeriske modellen av brua, som ble brukt i responsberegningene. Informasjon om den aerodynamiske oppførselen til brua, som var nødvendig for å modellere vindlasten, ble funnet ved hjelp av vindtunneltester. *Buffeting*-responsen til brua ble deretter predikert. For å bestemme turbulensparameterne som krevdes for å modellere vindlasten, ble to ulike sannsynlighetsbaserte metoder brukt. I den første metoden ble parameterne bestemt ved å bruke spesifikke persentiler fra sannsynlighetsfordelingene deres. I den andre metoden ble en probabilistisk modell brukt til å simulere turbulensstandardavvikene. Til slutt ble den predikerte og den målte responsen sammenlignet.

Den numeriske modellen viste seg å være nøyaktig nok til å kunne brukes i responsprediksjonene. Videre viste analysen av vindfeltet at det var stor variasjon i vindens egenskaper. I den probabilistiske modellen ble turbulensstandardavvikene modellert som korrelerte og log-normalfordelte stokastiske variabler, avhengige av den gjennomsnittlige vindhastigheten og -retningen. Modellen klarte å representere de virkelige turbulensstandardavvikene, tydeliggjort av den gode overensstemmelsen mellom de simulerte og de målte parameterne.

Det ble observert at variabiliteten til vinden forplantet seg inn i den målte responsen. Den laterale og vertikale responsen ble godt predikert, mens torsjonsresponsen ble overestimert. Imidlertid var den gode overensstemmelsen for den laterale responsen uventet. Videre klarte den probabilistiske modellen av turbulensstandardavvikene å overføre variabiliteten til vinden inn i den predikerte responsen. Det viste seg at turbulensparameterne påvirket prediksjonene i stor grad. I tillegg hadde variabler, som modale parametere og aerodynamiske deriverte, stor påvirkning.

Ved å bruke en sannsynlighetsbasert metode klarte *buffeting*-teorien og vindfeltmodellene stort sett å predikere responsen til Hålogalandsbrua. Imidlertid viste det seg å være svært viktig å estimere nøyaktige verdier av alle variabler for å oppnå gode responsprediksjoner.

Contents

Preface	i
Abstract	ii
Sammendrag	iii
1 Introduction	1
1.1 Background	1
1.2 Short Literature Review	2
1.3 Problem Formulation and Objectives	4
1.4 Limitations	5
1.5 Structure of the Report	5
2 Theory	7
2.1 Data Processing	7
2.1.1 Correlation Functions and Spectral Density Functions	7
2.1.2 Welch Procedure	8
2.1.3 Filtering and Downsampling	9
2.2 Probability Theory	10
2.3 Modal Analysis	12
2.4 Covariance-Driven Stochastic Subspace Identification	13
2.4.1 Challenges Related to Cov-SSI	17
2.5 Wind	18
2.5.1 Buffeting Load	18
2.5.2 Wind Turbulence Spectra	21
2.6 Buffeting Response	23
3 Methodology	27
3.1 The Hålogaland Bridge and the Monitoring System	27
3.2 Abaqus Model of the Bridge	30
3.3 Data Processing	30
3.4 Covariance-Driven Stochastic Subspace Identification	33
3.5 Wind Tunnel Test	35
3.6 Extraction of Data from the Monitoring System	36
3.7 Probabilistic Wind Field Model	38
3.8 Calculation of Buffeting Response	40
4 Results	43
4.1 Modal Analysis	43
4.1.1 Results from Abaqus	43
4.1.2 Results from Cov-SSI	45

4.1.3 Comparison of Modal Parameters from Abaqus and Cov-SSI	47
4.2 Wind Tunnel Test	48
4.3 Wind Field	50
4.3.1 Wind Field Characteristics	50
4.3.2 Wind Turbulence Spectra	53
4.3.3 Probabilistic Wind Field Model	55
4.4 Response of the Bridge	60
4.4.1 Measured Response	60
4.4.2 Comparison of Measured and Predicted Response	63
5 Discussion	67
5.1 Modal Analysis	67
5.1.1 Comparison of Modal Parameters from Abaqus and Cov-SSI	67
5.1.2 Reasons for Discrepancies	69
5.2 Wind Tunnel Test	71
5.3 Wind Field	71
5.3.1 Wind Field Characteristics	71
5.3.2 Wind Turbulence Spectra	73
5.3.3 Probabilistic Wind Field Model	75
5.4 Response of the Bridge	77
5.4.1 Measured Response	77
5.4.2 Comparison of Measured and Predicted Response	79
5.4.3 Influence of the Turbulence Parameters	81
5.4.4 Contribution of Wind Forces on Cables and Hangers	81
5.4.5 Uncertainty of the Structural Damping Ratio	82
5.4.6 Uncertainty of the Load Coefficients and the Aerodynamic Derivatives	83
5.4.7 Uncertainty of the Numerical Model	85
5.4.8 Uncertainty of the Probabilistic Wind Field Model	85
6 Conclusion and Further Work	87
6.1 Conclusion	87
6.2 Recommendations for Further Work	88
Bibliography	90
A Python Scripts	94
A.1 Data Processing	
A.2 Probabilistic Wind Field Model	
A.3 Buffeting Response	
A.4 Operational Modal Analysis	

List of Symbols

Latin

Symbol	Explanation
\mathbf{A}, \mathbf{A}_c	discrete and continuous state matrix
A_{mn}	cross sectional admittance function
$A_{u,w}$	spectral parameters
$A_1^* - A_6^*$	aerodynamic derivatives associated with the motion in torsion
\mathbf{a}	vector of Fourier amplitudes
a	Fourier amplitude
\mathbf{B}_c	continuous input matrix
\mathbf{B}_q	buffeting dynamic load coefficient matrix
B	width of bridge deck
\mathbf{C}	damping matrix
\mathbf{C}_a	measured acceleration location matrix
\mathbf{C}_{ae}	aerodynamic damping matrix
\mathbf{C}, \mathbf{C}_c	discrete and continuous output matrix
\mathbf{C}_d	measured displacement location matrix
\mathbf{C}_v	measured velocity location matrix
$\tilde{\mathbf{C}}$	modal damping matrix
C_{nm}	normalised cross-spectrum of turbulence
$C_{D,L,M}$	load coefficients
$\tilde{C}_{D,L,M}$	load coefficients
$C'_{D,L,M}$	load coefficients
\tilde{C}_{aei}	modal aerodynamic damping for mode i
\tilde{C}_i	modal damping for mode i
\mathcal{C}_i	controllability matrix
\mathbf{D}_c	direct transmission matrix
D	depth of bridge deck
f	frequency in Hz
f_A	Nyquist frequency
\mathbf{H}_i	block-Hankel matrix with i rows and columns
\tilde{H}_i	modal frequency response function for mode i
$H_1^* - H_6^*$	aerodynamic derivatives associated with the across-wind motion
\mathbf{I}	identity matrix
I_u, I_w	along-wind and vertical turbulence intensity
i	number of blockcrows
\mathbf{K}	stiffness matrix
\mathbf{K}_{ae}	aerodynamic stiffness matrix
$\tilde{\mathbf{K}}$	modal stiffness matrix

Symbol	Explanation
K	reduced frequency
$K_{u,w}$	decay coefficients
\tilde{K}_{aei}	modal aerodynamic stiffness for mode i
\tilde{K}_i	modal stiffness for mode i
L_{exp}	length of structure exposed to wind
${}^xL_u, {}^xL_w$	integral length scales in along-wind and vertical direction
\mathbf{M}	mass matrix
$\tilde{\mathbf{M}}$	modal mass matrix
\tilde{M}_i	modal mass for mode i
N	order of Butterworth filter
N_{mod}	number of modes
n_d	number of segments for Welch procedure
\mathcal{O}_i	observability matrix
$P_1^* - P_6^*$	aerodynamic derivatives associated with the along-wind motion
\mathbf{p}	load vector
$\tilde{\mathbf{p}}$	modal load vector
\tilde{Q}_{aei}	modal motion induced load for mode i
\tilde{Q}_i	modal turbulence load for mode i
\mathbf{q}	turbulence load vector
\mathbf{q}_{ae}	motion induced load vector
\mathbf{q}_{tot}	total wind load vector
$\tilde{\mathbf{q}}$	static wind load vector
\mathbf{R}_i	output correlation matrix at time lag i
R_{xx}	auto-correlation function of x
R_{xy}	cross-correlation function between x and y
\mathbf{r}	displacement vector
\mathbf{S}	cross-spectral density matrix
S	auto- or cross-spectral density
\mathbf{s}	state vector
s	stabilisation level
T	period of a time series
t	time
\mathbf{U}_1	matrix containing left singular vectors corresponding to non-zero singular values
U	instantaneous horizontal wind velocity
u	turbulence component in along-wind direction
\mathbf{V}_1	matrix containing right singular vectors corresponding to non-zero singular values
V	mean wind velocity
V_{rel}	relative wind velocity
\hat{V}	reduced velocity
\mathbf{v}	vector containing wind turbulence components

Symbol	Explanation
\mathbf{v}_k	vector of measurement noise at time instance k
$\mathbf{W}_{1,2}$	weighting matrices
\mathbf{w}_k	vector of process noise at time instance k
w	turbulence component in vertical direction
\mathbf{y}_l	vector of measured outputs
z	height above ground

Greek

Symbol	Explanation
α	angle of flow incidence
α	mean wind direction
Δt	sampling interval
Δx	distance between two points
$\boldsymbol{\eta}$	generalised coordinate vector
η_i	generalised coordinate of mode i
$\hat{\lambda}_r, \lambda_r$	discrete and continuous system poles
$\tilde{\mu}, \tilde{\sigma}$	lognormal distribution parameters
ξ	damping ratio
ρ	air density
ρ_{ij}	correlation coefficient of two random variables
$\boldsymbol{\Sigma}_1$	matrix containing non-zero singular values
σ_x	standard deviation of x
σ_x^2	variance of x
σ_{xy}^2	covariance of x and y
τ	time lag
$\boldsymbol{\Phi}$	mode shape matrix
$\boldsymbol{\phi}_i$	mode shape of mode i
$\boldsymbol{\phi}_r$	continuous mode shape
$\boldsymbol{\psi}_r$	discrete eigenvector
ω	frequency in rad/s
ω_c	cut-off frequency
ω_i	natural frequency of mode i

List of Abbreviations

Abbreviation	Explanation
AD	Aerodynamic Derivative
Cov-SSI	Covariance-driven Stochastic Subspace Identification
DOF	Degree Of Freedom
E	East
FFT	Fast Fourier Transform
MDOF	Multi-Degree Of Freedom
NPRA	Norwegian Public Roads Administration
ODB	Output Database
OMA	Operational Modal Analysis
PDF	Probability Density Function
PSD	Power Spectral Density
RMS	Root Mean Square
SVD	Singular Value Decomposition
SW	Southwest

Chapter 1

Introduction

1.1 Background

Today, traffic demands make it necessary to build increasingly longer suspension bridges. In order to assure good design of the bridges, it is important to acquire more extensive knowledge of the behaviour of these types of structures. Due to the slenderness of these bridges, they are particularly prone to large wind-induced response. In addition, the wind load is one of the most significant loads acting on a suspension bridge. Thus, prediction of the wind-induced response is a crucial part of the wind resistant design. The buffeting theory is commonly used for this purpose. As the wind is a stochastic process, the wind load needs to be modelled in order to predict the buffeting response. Previous studies have shown that the current models for wind fields, wind loads and wind-induced response sometimes fail to predict every aspect of the wind and the wind effects (Macdonald, 2003)(Fenerci and Øiseth, 2017). One probable reason for discrepancies is that the models do not fully account for the random nature of the wind. The models typically utilise the spectral densities of the turbulence components in order to predict the wind field and the wind-induced response. However, since the wind is a stochastic process, it is not possible to predict the exact spectral densities of the turbulence components using any mathematical expression. Several expressions have been proposed to describe the spectra (Kaimal et al., 1972)(Kármán, 1948). However, they are all based on deterministic parameters of the turbulence components, with the mean wind speed being the only design parameter. Thus, the variability of the turbulence characteristics, due to the random nature of the wind, is not represented.

In recent years, full-scale monitoring of long-span bridges has made it possible to study wind fields and wind-induced response of bridges more thoroughly. Previous papers presenting long-term monitoring studies show a large randomness in the wind characteristics (Bastos et al., 2018)(Fenerci et al., 2017). Furthermore, recent studies show that buffeting response analyses of long-span cable supported bridges are greatly affected by this variability, and that this should be considered in the design process (Fenerci et al., 2017)(Fenerci and Øiseth, 2017). In order to

take the variability into account, a probabilistic approach to the problem is one method to improve the prediction of the wind field, and thus improve the buffeting response prediction. This can be achieved by utilising the probability distributions of the turbulence parameters from the measurements. In this way, the spectral densities of the turbulence components can be better approximated, which results in more accurate predictions of the buffeting response. In a complete probabilistic model of the wind field, the dependence of the parameters on other wind characteristics, such as the mean wind speed and direction, also needs to be considered. Additionally, the correlation structure of the parameters needs to be determined. Such a probabilistic model is proposed by [Fenerci and Øiseth \(2018a\)](#), for the wind field along the Hardanger Bridge.

In this thesis, the wind field and the wind-induced response of the Hålogaland Bridge is studied for the first time. The Hålogaland Bridge is a suspension bridge located in northern Norway. The bridge was opened to traffic in December 2018 and is, with its span of 1145 m, currently the second longest suspension bridge in Norway. Due to the desire for an increased understanding of the dynamic behaviour of long-span suspension bridges, a full-scale monitoring system was installed in 2021. The system was not completely operational until February 2022, which results in only two months of data being available for use in this thesis. A thorough description of the monitoring system will be given later. In this thesis, the measured wind data from the monitoring system are utilised to improve the response predictions of the Hålogaland Bridge. The acceleration response of the bridge is predicted, using buffeting theory. Two different probabilistic approaches are utilised to obtain the turbulence parameters needed for the turbulence spectra. First, specific percentile values from the probability distributions of the turbulence parameters are used in the response calculations. Later, a probabilistic model of the wind field is established, like for the Hardanger Bridge ([Fenerci and Øiseth, 2018a](#)). In this way, simulated turbulence parameters from the probabilistic model can be utilised in the response predictions. The accuracy of the predicted acceleration response of the Hålogaland Bridge is then evaluated, by comparing it to the response obtained by the monitoring system at the site.

1.2 Short Literature Review

The number of studies comparing predicted buffeting response with measured response from full-scale measurements is relatively limited. In a study performed by [Macdonald \(2003\)](#), quasi-steady theory was used to predict the buffeting response of the Second Severn Crossing, by the use of full-scale monitoring data. A satisfactory prediction of the vertical and torsional buffet-

ing response was achieved when the turbulence intensity and total damping was overestimated. However, these response components were underestimated if values of the turbulence intensity and damping from the measurements were used. On the other hand, the predicted lateral buffeting response was overestimated, presumably due to underestimation of the aerodynamic damping from the quasi-steady theory. However, the monitoring system only consisted of an anemometer and three accelerometers at one location on the bridge girder, and so the predictions did not account for correlation of the wind along the bridge span.

Furthermore, [Cheynet et al. \(2016\)](#) did a study on the Lysefjord Bridge in Norway, where the calculated buffeting response was compared to the measured response, using measurement data from one single day. The predicted response matched the measured response reasonably well, although the predictions showed a tendency to underestimate the lateral and vertical response. The study found that two of the main reasons for these discrepancies might be the topography at the site and the non-stationarity of the wind.

Another study, performed by [Fenerci and Øiseth \(2017\)](#), did a comparison between the measured response and the predicted buffeting response of the Hardanger Bridge. Long-term monitoring data from the bridge were used to study the wind characteristics along the bridge and to improve the response predictions. Several different alternatives of the turbulence parameters, from the design basis and from the measurements, were utilised in the response predictions. In this study it was found that the use of parameters from the design basis resulted in an underestimation of the predicted response. However, by utilising the probability distributions of the parameters from the measurements to account for the variability of the wind field, the predicted response matched the measured response to a higher degree. The predicted lateral response was generally underestimated, presumably due to the wind forces on the cables and hangers being neglected. It was also found that using conditional probability distributions for the turbulence intensities, which accounted for the dependence on the mean wind speed, gave more accurate predictions. Furthermore, it was found that the turbulence parameters were dependent on each other. Therefore, it was suggested that better predictions could be obtained by accounting for the interaction between the turbulence parameters, by using joint probability distributions or conditional probability distributions.

As previously described, [Fenerci and Øiseth \(2018a\)](#) utilised measurement data from the Hardanger Bridge monitoring system, in order to form a probabilistic model of the wind field along the bridge. The purpose of the study was to take into account the variability and the correlation of the turbulence parameters, giving a more accurate prediction of the wind field.

The turbulence parameters were considered as correlated lognormally distributed random variables, depending on the mean wind speed and direction. Simulations of the wind field from the probabilistic model were compared with measurement data, which in general showed good agreement.

1.3 Problem Formulation and Objectives

As demonstrated above, methods for predicting the wind-induced response of long-span suspension bridges and methodologies for wind field modelling for such bridges are still in development. It is found that, especially, accounting for the stochastic nature of the wind, can significantly improve the predictions. This master thesis will contribute to this work by studying the Hålogaland Bridge.

The main focus of this thesis is to compare the predicted buffeting response of the Hålogaland Bridge with the measured response from the monitoring system. This is done in order to assess how well the buffeting theory and the wind field models are able to predict the true response of the bridge. Acceleration and wind data from the monitoring system are therefore retrieved and processed. The wind data are used to take the variability of the wind field into account, thereby improving the buffeting response predictions. A numerical model of the bridge is utilised in order to calculate the predicted buffeting response. To assess how well this model is able to represent the true dynamic characteristics of the bridge, an operational modal analysis is performed, using the method covariance-driven stochastic subspace identification (Cov-SSI). Furthermore, wind tunnel tests are performed in order to obtain knowledge about the aerodynamic behaviour of the bridge girder, which is required to model the wind load acting on the bridge. To summarise, the main objectives of this thesis are:

- Perform a system identification of the Hålogaland Bridge, using Cov-SSI, in order to assess the accuracy of the numerical model
- Analyse the wind field at the Hålogaland Bridge
- Establish a probabilistic model of the wind field at the Hålogaland Bridge
- Calculate and compare the predicted buffeting response of the Hålogaland Bridge with the measured response from the monitoring system

1.4 Limitations

The main limitations of this thesis are:

- The amount of measurement data available for the analyses is limited to only two months of data, which may introduce inaccuracies. For example will extreme values not be well represented in the data.
- The stationarity of the data is not investigated properly. As many of the procedures in this thesis are based on an assumption of stationarity, the presence of non-stationary data might introduce errors, particularly in the response predictions.
- The contribution to the predicted buffeting response caused by wind forces acting on the cables and hangers is not included.
- Only two of the six turbulence parameters necessary to describe the wind field along the bridge are included in the probabilistic model of the wind field.

1.5 Structure of the Report

Chapter 2 presents the theory used in this thesis. First, relevant theory related to data processing is presented, followed by an introduction to basic probability theory. Then, theory necessary to perform a modal analysis is presented, as well as a brief introduction to Cov-SSI. Finally, theory related to the buffeting load and calculation of the buffeting response is presented.

Chapter 3 describes the methods utilised in order to get the results. First, an introduction of the Hålogaland Bridge and the monitoring system is given, followed by a brief description of the numerical model of the bridge. Then, the methods used to perform the data processing and Cov-SSI are explained. A short description of the wind tunnel tests and the extraction of the results from these tests is also given. Lastly, the process of extracting the necessary information from the measurement data, the probabilistic wind field modelling and the calculation of the buffeting response is described.

Chapter 4 presents all the results obtained in this thesis. This includes the results from the numerical model and Cov-SSI, the wind tunnel tests, the wind field analysis and the measured and predicted response of the bridge.

Chapter 5 is a discussion of all the results obtained in this thesis.

Chapter 6 presents the final conclusions of the thesis, as well as suggestions for further work.

Chapter 2

Theory

2.1 Data Processing

As the monitoring system on the Hålogaland Bridge is relatively new, only raw data are available. In order to use the data as a tool to predict and compare the buffeting response with the measured response, the data must be processed. Therefore, this chapter gives an introduction to several important aspects regarding processing of data.

2.1.1 Correlation Functions and Spectral Density Functions

Correlation functions and spectral density functions play an important role in the processing of data from random processes. Given two time series, $x_k(t)$ and $y_k(t)$, assumed to be from stationary random processes, the auto-correlation function of $x_k(t)$ and the cross-correlation function between $x_k(t)$ and $y_k(t)$ are defined as ([Rainieri and Fabbrocino, 2014](#))

$$R_{xx}(\tau) = E[x_k(t) x_k(t + \tau)] \quad (2.1)$$

$$R_{xy}(\tau) = E[x_k(t) y_k(t + \tau)] \quad (2.2)$$

where $E[\cdot]$ denotes the expected value operation, t is the time and τ is the time lag. The assumption of stationarity implies that the statistical properties of the time series are independent of time ([Newland, 2005](#)). If the time series have zero means, the covariance functions are equal to the correlation functions ([Rainieri and Fabbrocino, 2014](#)).

Furthermore, the two-sided spectral density functions are the Fourier transforms of the correlation functions, given that the time series are processed so that the mean values are zero, ([Bendat and Piersol, 2011](#)). The spectral density functions, also called power spectral densities (PSDs), are widely used to analyse stochastic processes such as wind, waves and earthquakes. The PSDs are useful due to the fact that they contain information about the frequency content of a signal. This can be used to obtain statistical properties of the signal, such as variance and

covariance, in the frequency domain. The one-sided auto- and cross-spectral density functions are defined as (Rainieri and Fabbrocino, 2014)

$$S_{xx}(f) = 2 \lim_{T \rightarrow \infty} \frac{1}{T} E \left[|X_k(f, T)|^2 \right] \quad (2.3)$$

$$S_{xy}(f) = 2 \lim_{T \rightarrow \infty} \frac{1}{T} E [X_k^*(f, T) Y_k(f, T)] \quad (2.4)$$

where f is the frequency in the range ($0 < f < +\infty$). $X_k(f, T)$ and $Y_k(f, T)$ are the Fourier transforms of the time series $x_k(t)$ and $y_k(t)$, and $X_k^*(f, T)$ is the complex conjugate of $X_k(f, T)$. The time series have a finite period, T , and are assumed to be stationary within this period.

The spectral density functions can be used to obtain the variance, σ_x^2 , and covariance, σ_{xy}^2 , of the time series through the relations (Bendat and Piersol, 2011)

$$\sigma_x^2 \approx \int_0^\infty S_{xx}(f) df \quad (2.5)$$

$$\sigma_{xy}^2 \approx \int_0^\infty S_{xy}(f) df \quad (2.6)$$

In order to use these relations, it is required that the time series, $x(t)$ and $y(t)$, are from ergodic Gaussian random processes with zero means. A stationary process is said to be ergodic if the ensemble averages are equal to the time averages (Newland, 2005).

2.1.2 Welch Procedure

There are several ways of estimating the PSDs, defined in Equation (2.3) and Equation (2.4), from time series of sample records. One common method is the one developed by Welch (1967), based on a fast Fourier transform (FFT) of the signal. The signal is first divided into n_d overlapping segments. An appropriate time window is then applied to each segment, in order to suppress leakage. Leakage is a phenomenon that is introduced when the signals contain frequencies where the period does not coincide with the record length (Bendat and Piersol, 2011), which is the case for signals of finite length. Leakage results in power at frequencies at the ends of the segment to leak into nearby frequencies, causing the spectral density to be distorted. This problem is avoided by the use of a time window, which tapers the data such that the data at the beginning and the end of the signal are weighted less than the data in the middle. However, tapering the segments causes errors, which could be rectified by increasing the length of the

segments. For finite signals, this could be done by reducing the number of segments. However, the Welch procedure is based on averaging over the segments. Thus, this will lead to a larger variability in the spectral estimates. Instead, the signal is divided into overlapping segments, which still increases the length of the segments, while maintaining an acceptable number of divisions. Finally, the FFT of each segment is computed and the PSDs are obtained by averaging over the segments.

2.1.3 Filtering and Downsampling

Time series of measured data contain discrete values. In order to obtain the necessary information from the time series, it is important to determine an appropriate sampling frequency. If the sampling frequency is too high, the values will be redundant due to correlation and cause increased computation. On the other hand, a low sampling frequency can lead to high frequency content being perceived as low frequency content, a phenomenon called aliasing. The highest frequency of the signal that can be reproduced is called the Nyquist frequency, and is given as (Bendat and Piersol, 2011)

$$f_A = \frac{1}{2\Delta t} \quad (2.7)$$

where Δt is the sampling interval. In other words, the Nyquist frequency is half of the sampling frequency of the signal. If the signal contains frequency content higher than the Nyquist frequency, aliasing will occur.

For a signal that has a higher sampling frequency than what is relevant for the analysis, it could be convenient to downsample the signal. By doing this, both unwanted noise and high computational time is avoided. One common way of downsampling the signal is to create a new signal by taking for instance every tenth sample and discarding the rest, creating a new signal with a lower sampling frequency (Bendat and Piersol, 2011) (Tan and Jiang, 2013) (Rao and Swamy, 2018).

When a signal has been downsampled, the Nyquist frequency is also changed, causing issues with aliasing to occur. To avoid this problem, it is necessary to apply a low-pass filter prior to the downsampling. A low-pass filter will remove frequency content above a specified cut-off frequency from the signal, keeping only the frequency content of interest (Rainieri and Fabbro-cino, 2014). A frequently used filter is the Butterworth low-pass filter, which is described by the magnitude response function, $H(i\omega)$, in the frequency domain (Tseng and Lee, 2017)

$$|H(i\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \quad (2.8)$$

Here, i is the complex number, N is the order of the filter, ω is the sampling frequency and ω_c is the cut-off frequency. The order of the filter determines how abrupt the frequencies are cut off. Furthermore, by modifying the Butterworth low-pass filter, a high-pass filter can be obtained. Such a filter removes the frequency content below a cut-off frequency. High-pass filters are often used to remove spurious trends from data (Rainieri and Fabbrocino, 2014).

2.2 Probability Theory

A huge part of this thesis concerns the analysis of the wind field at the Hålogaland Bridge. Since wind is a stochastic process, a probabilistic framework is useful for describing the parameters related to the wind field. The probability density function (PDF) is commonly used, as it expresses the distribution of the values of a random variable (Newland, 2005). In this thesis, the relevant random variables appear to fit a lognormal probability distribution, which is closely connected to the normal probability distribution. Taking the natural logarithm of a lognormally distributed variable, yields a normally distributed variable (Thomopoulos, 2018). The lognormal PDF can be written as

$$f(y|\tilde{\mu}, \tilde{\sigma}) = \frac{1}{y\tilde{\sigma}\sqrt{2\pi}} \exp\left(\frac{-(\ln(y) - \tilde{\mu})^2}{2\tilde{\sigma}^2}\right) \quad (2.9)$$

where $\tilde{\mu}$ and $\tilde{\sigma}$ are the mean value and standard deviation of the normally distributed variable, x , obtained when taking the natural logarithm of the lognormally distributed variable, y .

As a stochastic process consists of several random variables, it is important to consider the correlation between the variables. Correlation is the statistical relationship between two random variables. In other words, the correlation expresses how the values of the random variables vary together. It is common to express the correlation of random variables using either the covariance or the correlation coefficient, which are defined in Equations (2.10) and (2.11), respectively (Newland, 2005).

$$cov(Z_i, Z_j) = E[(Z_i - E[Z_i])(Z_j - E[Z_j])] \quad (2.10)$$

$$\rho_{ij} = \frac{\text{cov}(Z_i, Z_j)}{\sqrt{\sigma_{Z_i}^2 \cdot \sigma_{Z_j}^2}} \quad (2.11)$$

with $\sigma_{Z_{i,j}}^2$ being the variance of the random variables $Z_{i,j}$.

In this thesis, it is desired to generate correlated lognormally distributed random variables, related to the wind field at the Hålogaland Bridge. This is achieved with knowledge of the probability distributions and the correlation structure of the random variables. The easiest method for sampling a set of multivariate correlated random variables is by utilising diagonalisation of the covariance matrix, in order to sample the random variables separately. For normally distributed random variables this introduces no errors. However, for all other distributions this diagonalisation causes a distortion of the original distribution (Žerovnik et al., 2012). In order to avoid this effect, the relationship between the normal and the lognormal probability distribution is utilised. The method presented by Žerovnik et al. (2012) is based on sampling normally distributed random variables and then transforming these to the desired distribution, which in this case is the lognormal distribution.

In order to do this, only the mean values and the covariance matrix of the normally distributed random variables are needed. The covariance matrix of the vector \mathbf{X} , containing the normally distributed variables, can be expressed as a function of the covariance matrix of the vector \mathbf{Y} , containing the lognormally distributed variables, using the relations (Žerovnik et al., 2012)

$$\text{cov}(X_i, X_j) = \ln \left(\frac{\text{cov}(Y_i, Y_j)}{m_i \cdot m_j} + 1 \right) \quad , \quad m_{i,j} = \exp \left(\tilde{\mu}_{i,j} + \frac{\tilde{\sigma}_{i,j}^2}{2} \right) \quad (2.12)$$

where $X_{i,j}$ and $Y_{i,j}$ are the elements of the vectors \mathbf{X} and \mathbf{Y} , while $\tilde{\mu}_{i,j}$ and $\tilde{\sigma}_{i,j}$ are the mean values and the standard deviations of $X_{i,j}$.

Furthermore, Equation (2.12) can be rewritten as an expression of the correlation coefficient matrix of \mathbf{Y} and the standard deviations of \mathbf{X} , as

$$\text{cov}(X_i, X_j) = \ln \left[(\rho_y)_{ij} \sqrt{\exp(\tilde{\sigma}_i^2) - 1} \sqrt{\exp(\tilde{\sigma}_j^2) - 1} + 1 \right] \quad (2.13)$$

where $(\rho_y)_{ij}$ are the elements of the correlation coefficient matrix of \mathbf{Y} . The mean values and the covariance matrix of the vector \mathbf{X} can then be utilised to sample normally distributed random variables.

Finally, a set of multivariate correlated lognormally distributed random variables can be ob-

tained by taking the natural exponents of the set of multivariate correlated normally distributed random variables.

2.3 Modal Analysis

The behaviour of a dynamic system is governed by the equation of motion. For multi-degree-of-freedom (MDOF) systems the equilibrium equation is given as (Humar, 2002)

$$\mathbf{M}\ddot{\mathbf{r}}(x, t) + \mathbf{C}\dot{\mathbf{r}}(x, t) + \mathbf{K}\mathbf{r}(x, t) = \mathbf{p}(x, t) \quad (2.14)$$

Here, \mathbf{M} , \mathbf{C} and \mathbf{K} denote the mass, damping and stiffness matrices, respectively. The applied force is represented by the vector \mathbf{p} . The displacement of the structure is described by \mathbf{r} , which contains the degrees of freedom (DOFs).

The modal analysis is based on the assumption that the displacements of the structure, $\mathbf{r}(x, t)$, can be described as the sum of the response from several modes. Each mode is described by its modal parameters: natural frequency, damping ratio and mode shape. The response from each mode is obtained as the product of a spatially dependent part and a time dependent part, called mode shapes and generalised coordinates. The total response of the structure is then obtained as (Strømmen, 2006)

$$\mathbf{r}(x, t) = \sum_{i=1}^{N_{mod}} \boldsymbol{\phi}_i(x) \cdot \eta_i(t) = \boldsymbol{\Phi}(x) \cdot \boldsymbol{\eta}(t) \quad , \quad \boldsymbol{\Phi}(x) = [\boldsymbol{\phi}_1 \quad \dots \quad \boldsymbol{\phi}_i \quad \dots \quad \boldsymbol{\phi}_{N_{mod}}] \quad (2.15)$$

where $\boldsymbol{\Phi}(x)$ is the mode shape matrix, which contains the N_{mod} mode shapes $\boldsymbol{\phi}_i(x)$ corresponding to the N_{mod} generalised coordinates $\eta_i(t)$. Here, the mode shapes are regarded as continuous functions.

The natural frequencies and mode shapes of the system is found by solving the eigenvalue problem (Strømmen, 2006)

$$(\mathbf{K} - \omega_i^2 \mathbf{M})\boldsymbol{\phi}_i = \mathbf{0} \quad (2.16)$$

where ω_i is the natural frequency corresponding to the mode shape $\boldsymbol{\phi}_i$. Here, the damping of the system is neglected.

Inserting the expression for \mathbf{r} (Equation (2.15)) into Equation (2.14), pre-multiplying with

Φ^T and integrating both sides over the total length of the structure, results in the uncoupled equation of motion for the MDOF system (Strømmen, 2006)

$$\tilde{\mathbf{M}}\ddot{\boldsymbol{\eta}}(t) + \tilde{\mathbf{C}}\dot{\boldsymbol{\eta}}(t) + \tilde{\mathbf{K}}\boldsymbol{\eta}(t) = \tilde{\mathbf{p}}(t) \quad (2.17)$$

Here $\tilde{\mathbf{M}}$, $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{K}}$ are the modal mass, damping and stiffness matrices, and $\tilde{\mathbf{p}}$ is the modal load vector. Due to the orthogonal properties of the mode shapes, the modal matrices are diagonal. Therefore, Equation (2.17) corresponds to N_{mod} uncoupled equations of motion, which can be solved separately for each mode.

2.4 Covariance-Driven Stochastic Subspace Identification

The modal parameters of a structure, such as the natural frequencies, damping ratios and mode shapes, can be obtained through measurements. When the estimation of the modal parameters is based on response measurements conducted on a structure in operational conditions, the method is called operational modal analysis (OMA). The method is based on the assumption of a linear and stationary system, measured by sensors that are located such that all modes of interest can be observed (Rainieri and Fabbrocino, 2014). The two first conditions imply that the response of the system is a linear combination of the inputs and that the dynamic characteristics of the structure are invariant with time.

When conducting an operational modal analysis, the loads acting on the structure are not controlled or measured. Therefore, some assumptions about the inputs to the system are required. The structure is assumed to be excited by a broad band of frequencies. If this is not the case, the dynamic properties of both the input and the structure can be observed in the response. Thus, some estimated modes might be related to the loads and not to the structure (Brincker and Kirkegaard, 2010). The broad band excitation is also necessary to ensure that all the modes of interest are excited (Rainieri and Fabbrocino, 2014).

There are several different OMA methods, and in this thesis the method called covariance-driven stochastic subspace identification (Cov-SSI) will be utilised. This is a time domain method, meaning that it is based on the analysis of correlation functions, as opposed to the frequency domain methods, which are based on the spectral density functions (Rainieri and Fabbrocino, 2014). The Cov-SSI will be conducted in Python by the use of KOMA, which is a package for OMA implemented by Kvåle (2022), developed for the research paper by Kvåle et al. (2017). Here, the

essential theory behind Cov-SSI will be briefly explained.

By the use of the Cov-SSI method, a stochastic state-space model of the system is identified from the response measurements (Kvåle et al., 2017). The state-space model is needed to convert the equation of motion, given in Equation (2.14), from a second order differential equation into two first order differential equations (Rainieri and Fabbrocino, 2014). The two new equations are defined as the state equation and the observation equation. The state equation is obtained by first introducing $\mathbf{p}(t) = \bar{\mathbf{B}}\mathbf{u}(t)$ into Equation (2.14), and pre-multiplying by \mathbf{M}^{-1} , giving

$$\ddot{\mathbf{r}}(t) + \mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{r}}(t) + \mathbf{M}^{-1}\mathbf{K}\mathbf{r}(t) = \mathbf{M}^{-1}\bar{\mathbf{B}}\mathbf{u}(t) \quad (2.18)$$

where $\bar{\mathbf{B}}$ is a matrix defining the input locations and \mathbf{u} represents the time variation. Furthermore, the state vector, \mathbf{s} , is defined as

$$\mathbf{s}(t) = \begin{bmatrix} \dot{\mathbf{r}}(t) \\ \mathbf{r}(t) \end{bmatrix} \quad (2.19)$$

By combining Equations (2.18) and (2.19) the state equation is obtained as (Rainieri and Fabbrocino, 2014)

$$\dot{\mathbf{s}}(t) = \mathbf{A}_c \mathbf{s}(t) + \mathbf{B}_c \mathbf{u}(t) \quad (2.20)$$

where

$$\mathbf{A}_c = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{C} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} -\mathbf{M}^{-1}\bar{\mathbf{B}} \\ \mathbf{0} \end{bmatrix} \quad (2.21)$$

\mathbf{A}_c and \mathbf{B}_c are the continuous state matrix and input matrix, respectively, and \mathbf{I} is the identity matrix. Furthermore, the general observation equation, expressing the measured outputs, \mathbf{y}_l , at l locations, is given as (Rainieri and Fabbrocino, 2014)

$$\mathbf{y}_l(t) = \mathbf{C}_a \ddot{\mathbf{r}}(t) + \mathbf{C}_v \dot{\mathbf{r}}(t) + \mathbf{C}_d \mathbf{r}(t) \quad (2.22)$$

where \mathbf{C}_a , \mathbf{C}_v and \mathbf{C}_d are matrices representing the locations for the measured acceleration, velocity and displacement. Hence, Equation (2.22) represents the observable part of the dynamic

response modelled in the state equation.

By combining Equations (2.18) and (2.22), the observation equation can be expressed as a function of the state and the input, as (Rainieri and Fabbrocino, 2014)

$$\mathbf{y}_l(t) = \mathbf{C}_c \mathbf{s}(t) + \mathbf{D}_c \mathbf{u}(t) \quad (2.23)$$

where \mathbf{C}_c and \mathbf{D}_c are the continuous output matrix and the direct transmission matrix, which are functions of the output and input location matrices and the system matrices.

Equations (2.20) and (2.23) represent the deterministic continuous-time state-space model. In an operational modal analysis, the inputs to the system are unmeasured. Therefore, it is modelled as process noise, \mathbf{w}_k , and measurement noise, \mathbf{v}_k . The model also needs to be converted to discrete time, resulting in the discrete-time stochastic state-space model (Rainieri and Fabbrocino, 2014)

$$\begin{aligned} \mathbf{s}_{k+1} &= \mathbf{A} \mathbf{s}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C} \mathbf{s}_k + \mathbf{v}_k \end{aligned} \quad (2.24)$$

where \mathbf{A} and \mathbf{C} are the discrete state and output matrices and k denotes the discrete time instance.

The state matrix contains the dynamic properties of the system, through its eigenvalues and the observable parts of the eigenvectors. Thus, the matrices \mathbf{A} and \mathbf{C} need to be obtained to find the modal parameters of the system. For this purpose, the correlation functions, \mathbf{R}_k , of the outputs of the system are estimated, as (Hermans and Auweraer, 1999)

$$\mathbf{R}_k = E[\mathbf{y}_{k+n} \mathbf{y}_n^T] \quad (2.25)$$

Here, k is the number of time lags. Further, the correlation matrices for different time lags are collected in a block-Hankel matrix, given by (Hermans and Auweraer, 1999)

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \dots & \mathbf{R}_i \\ \mathbf{R}_2 & \mathbf{R}_3 & \dots & \mathbf{R}_{i+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_i & \mathbf{R}_{i+1} & \dots & \mathbf{R}_{2i-1} \end{bmatrix} \quad (2.26)$$

where i corresponds to the number of rows and columns in the block-Hankel matrix, and is

called the number of blockrows.

The block-Hankel matrix can be decomposed as (Hermans and Auweraer, 1999)

$$\mathbf{H}_i = \mathcal{O}_i \mathcal{C}_i \quad (2.27)$$

where \mathcal{O}_i and \mathcal{C}_i are the observability matrix and the controllability matrix, defined as

$$\mathcal{O}_i = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{i-1} \end{bmatrix}, \quad \mathcal{C}_i = \begin{bmatrix} \mathbf{G} & \mathbf{AG} & \dots & \mathbf{A}^{i-1} \mathbf{G} \end{bmatrix}; \quad \mathbf{G} = E[\mathbf{s}_{k+n} \mathbf{y}_n^T] \quad (2.28)$$

Furthermore, the block-Hankel matrix is pre- and post-multiplied with two weighting matrices, \mathbf{W}_1 and \mathbf{W}_2 , to enhance the identification of modes with low excitation. Then, a singular value decomposition (SVD) is conducted, resulting in the relation (Hermans and Auweraer, 1999)

$$\mathbf{W}_1 \mathbf{H}_i \mathbf{W}_2^T = \mathbf{W}_1 \mathcal{O}_i \mathcal{C}_i \mathbf{W}_2^T = \mathbf{U}_1 \boldsymbol{\Sigma}_1 \mathbf{V}_1^T \quad (2.29)$$

where $\boldsymbol{\Sigma}_1$ is the matrix containing the non-zero singular values, and \mathbf{U}_1 and \mathbf{V}_1 are matrices containing the corresponding left and right singular vectors. The details behind the SVD and the weighting matrices will not be explained further, but a more thorough explanation can be found in Rainieri and Fabbrocino (2014) and Hermans and Auweraer (1999).

From Equation (2.29) it can be shown that

$$\mathcal{O}_i = \mathbf{W}_1^{-1} \mathbf{U}_1 \boldsymbol{\Sigma}_1^{1/2} \quad (2.30)$$

Finally, from the definition of the observability matrix, given in Equation (2.28), it can be seen that \mathbf{C} equals the first l rows of \mathcal{O}_i , while \mathbf{A} can be found from (Hermans and Auweraer, 1999)

$$\mathcal{O}_i^\dagger = \mathcal{O}_i^\downarrow \mathbf{A} \quad (2.31)$$

where \mathcal{O}_i^\dagger and \mathcal{O}_i^\downarrow are found by discarding the first and last l rows of \mathcal{O}_i .

When the output matrix and the state matrix have been estimated, the modal parameters of the system can be obtained through an eigenvalue decomposition of \mathbf{A} , resulting in the discrete system poles, $\hat{\lambda}_r$, and eigenvectors, $\boldsymbol{\psi}_r$. Then, the continuous system poles, λ_r , and the corresponding mode shapes, $\boldsymbol{\phi}_r$, at the sensor locations, are found as (Hermans and Auweraer, 1999)

$$\begin{aligned}\lambda_r &= \frac{1}{\Delta t} \ln \hat{\lambda}_r \\ \boldsymbol{\phi}_r &= \mathbf{C} \boldsymbol{\psi}_r\end{aligned}\tag{2.32}$$

where Δt is the sampling period. As damping is not neglected, λ_r and $\boldsymbol{\phi}_r$ are complex.

Then, assuming that the system is under-critically damped, the following relations can be utilised to obtain the natural frequencies, ω_r , and damping ratios, ξ_r (Kvåle et al., 2016)

$$\lambda_r = -\xi_r \omega_r \pm \sqrt{1 - \xi_r^2} \omega_r i \quad \implies \quad \omega_r = |\lambda_r|, \quad \xi_r = -\frac{\Re(\lambda_r)}{|\lambda_r|}\tag{2.33}$$

where $\Re()$ implies the real part of a complex number.

2.4.1 Challenges Related to Cov-SSI

Due to noise and modelling inaccuracies, the number of non-zero singular values obtained from the SVD of the weighted block-Hankel matrix, given in Equation (2.29), is larger than the actual order of the system (Rainieri and Fabbrocino, 2014) (Kvåle et al., 2017). Thus, the order of the system needs to be defined manually, according to the number of non-zero singular values and vectors to include in the computation of the observability matrix. This is a complicated task, since the actual order of the system is unknown and the choice greatly affects the eigenvalues and eigenvectors obtained from the eigenvalue decomposition. If the order of the system is set too low, weakly excited modes may not be identified, while a large value will cause spurious modes to appear in addition to the physical ones (Rainieri and Fabbrocino, 2014). Therefore, the order of the system is overestimated to ensure that all physical modes are identified, and then the spurious modes are eliminated.

The approach to identify the spurious modes starts with the estimation of the modal parameters for several system orders. Then, specified criteria are utilised to obtain the stable poles. This is done by comparing all the poles obtained for each system order with the poles of the s preceding orders, where s is a defined stabilisation level. The poles that fulfil specified sta-

bilisation criteria by comparison with the corresponding poles from all the s preceding orders are defined as stable. The determination of the corresponding poles from the different system orders are based on the maximisation of a chosen modal indicator. Furthermore, the stabilisation criteria are based on the deviance of modal parameters obtained for different values of the system order. A more thorough description of this approach can be found in [Kvåle et al. \(2017\)](#).

After the determination of stable poles, a stabilisation diagram is made, by plotting the system orders against the corresponding natural frequencies, for the stable poles. The physical poles can be detected as straight sequences of stable poles, while the spurious poles are typically more scattered ([Rainieri and Fabbrocino, 2014](#)).

Furthermore, the selection of the number of blockrows, i , and the stabilisation level, s , will influence the results from the Cov-SSI analysis. If the value of i is chosen too low, the low frequency content of the data will not be identified, while a high value will result in more spurious modes ([Kvåle et al., 2017](#)). In addition, the stabilisation level can be adjusted to eliminate the spurious modes, while still keeping the physical ones.

2.5 Wind

2.5.1 Buffeting Load

For long-span bridges the most significant loads are the loads due to wind. In general, it is assumed that the wind can be expressed as the sum of a mean component and a fluctuating component ([Tamura and Kareem, 2013](#)). The wind velocity is split into three orthogonal components, with U describing the wind in the along-wind direction, while v and w are the velocity fluctuations in the across wind horizontal and vertical directions, respectively. The wind in the along-wind direction is split into a mean value and a fluctuating part, $U = V + u(x, t)$. V is commonly known as the mean wind velocity, while u , v and w are known as the turbulence components ([Strømmen, 2006](#)). The amount of turbulence in a wind field is usually described using the turbulence intensities, I_u and I_w , which are defined as the ratio between the standard deviations of the turbulence components, σ_u and σ_w , and the mean wind velocity ([Tamura and Kareem, 2013](#))

$$I_u = \frac{\sigma_u}{V} \qquad I_w = \frac{\sigma_w}{V} \qquad (2.34)$$

The buffeting theory is used to describe the loads due to the wind fluctuations as well as the motion induced forces. The theory is derived under the assumption of a line-like structure, surrounded by a stationary and homogeneous wind field, where the mean wind is perpendicular to the x-axis of the structure (Strømmen, 2006). Thus, the across wind horizontal turbulence component, v , is not of interest. A stationary wind field implies that the statistical properties are independent of time (Newland, 2005), whereas homogeneity implies that the statistical properties are independent of the location in the wind field. A basic assumption in the buffeting theory is that the wind loads can be calculated from the velocity pressure and the flow incidence dependent drag, lift and moment coefficients, C_D , C_L and C_M . These load coefficients can be obtained from wind tunnel tests of a section model of the relevant structure. The forces acting on the bridge deck are by definition given as (Strømmen, 2006)

$$\begin{bmatrix} q_D(x, t) \\ q_L(x, t) \\ q_M(x, t) \end{bmatrix} = \frac{1}{2} \rho V_{rel}^2 \begin{bmatrix} D \cdot C_D(\alpha) \\ B \cdot C_L(\alpha) \\ B^2 \cdot C_M(\alpha) \end{bmatrix} \quad (2.35)$$

Here, q_D , q_L and q_M are the drag, lift and moment forces in the wind flow axes. B and D are the width and the depth of the deck, ρ is the density of air, V_{rel} is the instantaneous relative wind velocity and α is the angle of flow incidence. It is further assumed that the displacements and rotations are small, and that the turbulence components are much smaller than the mean wind velocity. Thus, the relative wind velocity and the angle of flow incidence can be approximated as (Strømmen, 2006)

$$\begin{aligned} V_{rel}^2 &\approx V^2 + 2Vu - 2V\dot{r}_y \\ \alpha &\approx \bar{r}_\theta + r_\theta + \frac{w}{V} - \frac{\dot{r}_z}{V} \end{aligned} \quad (2.36)$$

with \bar{r}_i and r_i being the static and dynamic part of the displacements of the cross section of the bridge in the lateral (y), vertical (z) or rotational (θ) direction, and \dot{r}_i being the velocity of the cross section.

The load coefficients in reality have a nonlinear variation. However, they are linearly approximated as functions of the coefficients \bar{C}_D , \bar{C}_L , \bar{C}_M , C'_D , C'_L and C'_M . Finally, the forces are transformed into the structural axes, which results in the expression for the total wind load, \mathbf{q}_{tot} , given in Equation (2.37). A more thorough derivation can be found in Strømmen (2006).

$$\mathbf{q}_{tot}(x, t) = \begin{bmatrix} \bar{q}_y(x) \\ \bar{q}_z(x) \\ \bar{q}_\theta(x) \end{bmatrix} + \begin{bmatrix} q_y(x, t) \\ q_z(x, t) \\ q_\theta(x, t) \end{bmatrix} = \bar{\mathbf{q}} + \underbrace{\mathbf{B}_q \cdot \mathbf{v}}_{\mathbf{q}} + \underbrace{\mathbf{C}_{ae} \cdot \dot{\mathbf{r}} + \mathbf{K}_{ae} \cdot \mathbf{r}}_{\mathbf{q}_{ae}} \quad (2.37)$$

The total wind load contains a horizontal, vertical and torsional component, indicated by y , z and θ . As can be seen, it is divided into a static part, $\bar{\mathbf{q}}$, and a dynamic part, $\mathbf{q} + \mathbf{q}_{ae}$. The static part is considered trivial and will not be included in the response calculations, described in Section 2.6. \mathbf{q} is the load due to the turbulence of the wind, $\mathbf{v} = [u \ w]^T$, while \mathbf{q}_{ae} is the motion induced load, associated with the motion of the structure, \mathbf{r} and $\dot{\mathbf{r}}$. The matrix \mathbf{B}_q can be defined as in Equation (2.38), as a function of the load coefficients and the cross sectional admittance functions, A_{mn} (Strømmen, 2006). The cross sectional admittance functions are functions of the frequency and can be determined from section model wind tunnel tests.

$$\mathbf{B}_q(\omega, x) = \frac{\rho V B}{2} \begin{bmatrix} 2(D/B)\bar{C}_D A_{yu} & ((D/B)C'_D - \bar{C}_L) A_{yw} \\ 2\bar{C}_L A_{zu} & (C'_L + (D/B)\bar{C}_D) A_{zw} \\ 2B\bar{C}_M A_{\theta u} & BC'_M A_{\theta w} \end{bmatrix} A_{mn}(\omega) \quad \begin{cases} m = y, z, \theta \\ n = u, w \end{cases} \quad (2.38)$$

The motion induced load, \mathbf{q}_{ae} , also called the self-excited forces, are caused by the interaction between the structure and the wind. These forces can be represented by the use of non-dimensional parameters called aerodynamic derivatives (ADs). The theoretical expressions for the self-excited forces as functions of the ADs were first established in the study of aeronautics. However, studies conducted by Scanlan and Tomko (1971) concluded that this theory can be applied also in bridge engineering. Originally, Scanlan and Tomko (1971) only investigated the vertical and torsional motion, but by including the lateral motion the expressions for the forces are given by

$$\begin{aligned} q_y^{ae} &= \frac{1}{2} \rho V^2 B \left(KP_1^* \frac{\dot{r}_y}{V} + KP_2^* \frac{B\dot{r}_\theta}{V} + K^2 P_3^* r_\theta + K^2 P_4^* \frac{r_y}{B} + KP_5^* \frac{\dot{r}_z}{V} + K^2 P_6^* \frac{r_z}{B} \right) \\ q_z^{ae} &= \frac{1}{2} \rho V^2 B \left(KH_1^* \frac{\dot{r}_z}{V} + KH_2^* \frac{B\dot{r}_\theta}{V} + K^2 H_3^* r_\theta + K^2 H_4^* \frac{r_z}{B} + KH_5^* \frac{\dot{r}_y}{V} + K^2 H_6^* \frac{r_y}{B} \right) \\ q_\theta^{ae} &= \frac{1}{2} \rho V^2 B^2 \left(KA_1^* \frac{\dot{r}_z}{V} + KA_2^* \frac{B\dot{r}_\theta}{V} + K^2 A_3^* r_\theta + K^2 A_4^* \frac{r_z}{B} + KA_5^* \frac{\dot{r}_y}{V} + K^2 A_6^* \frac{r_y}{B} \right) \end{aligned} \quad (2.39)$$

Here q_y^{ae} , q_z^{ae} and q_θ^{ae} are the motion induced forces and r_y , r_z and r_θ are the displacements of the structure for lateral, vertical and torsional movement, respectively. Furthermore, P_i^* , H_i^* and A_i^* are the ADs and $K = (\omega B)/V$ is the reduced frequency.

The ADs can be obtained through wind tunnel tests and are functions of the reduced velocity, \hat{V} , which is the inverse of the reduced frequency. That is, the ADs depend on the frequency of motion, ω , the type of cross section, and the mean wind velocity. Thus, the self-excited forces depend on both the motion of the structure and the wind field. Equation (2.39) is commonly written in matrix notation, resulting in the expression for $\mathbf{q}_{ae} = [q_y^{ae} \quad q_z^{ae} \quad q_\theta^{ae}]^T$ as given in Equation (2.37), where \mathbf{C}_{ae} and \mathbf{K}_{ae} are the aerodynamic damping and stiffness matrices, given as

$$\mathbf{C}_{ae} = \frac{\rho B^2}{2} \cdot \omega \begin{bmatrix} P_1^* & P_5^* & BP_2^* \\ H_5^* & H_1^* & BH_2^* \\ BA_5^* & BA_1^* & B^2 A_2^* \end{bmatrix}, \quad \mathbf{K}_{ae} = \frac{\rho B^2}{2} \cdot \omega^2 \begin{bmatrix} P_4^* & P_6^* & BP_3^* \\ H_6^* & H_4^* & BH_3^* \\ BA_6^* & BA_4^* & B^2 A_3^* \end{bmatrix} \quad (2.40)$$

2.5.2 Wind Turbulence Spectra

Due to the stochastic nature of the wind, the buffeting load used to calculate the buffeting response of a structure can not be precisely determined. Therefore, the load is described by the use of turbulence spectra, which contain information about the frequency content of the wind. Several models have been suggested to describe the turbulence spectra. One of the models for the one-point auto-spectra, $S_{u,w}$, of the turbulence components is the one proposed by [Kaimal et al. \(1972\)](#), which is of the form:

$$\frac{S_{u,w} f}{\sigma_{u,w}^2} = \frac{A_{u,w} f z}{(1 + 1.5 A_{u,w} f z)^{5/3}}, \quad f_z = \frac{f z}{V(z)} \quad (2.41)$$

where f is the frequency, $\sigma_{u,w}$ are the standard deviations of the turbulence components, $A_{u,w}$ are the non-dimensional spectral parameters, z is the height above ground and V is the mean wind speed.

In Norway, the Norwegian Public Roads Administration (NPRA) has published a handbook named N400 ([Vegdirektoratet, 2015](#)), which regulates the design of bridges. In N400 the one-point auto-spectra of the turbulence components are defined based on the Kaimal spectra, giv-

ing the expression

$$\frac{S_{u,w}f}{\sigma_{u,w}^2} = \frac{A_{u,w}\hat{f}_{u,w}}{(1 + 1.5A_{u,w}\hat{f}_{u,w})^{5/3}}, \quad \hat{f}_{u,w} = \frac{f^x L_{u,w}(z)}{V(z)} \quad (2.42)$$

where ${}^xL_{u,w}(z)$ are the integral length scales. N400 gives the following recommendations for the values of the spectral parameters: $A_u = 6.8$ and $A_w = 9.4$. The expressions for the integral length scales of u and w are given in N400 as (Vegdirektoratet, 2015)

$${}^xL_u = \begin{cases} L_1(z/z_1)^{0.3} & z > z_{min} \\ L_1(z_{min}/z_1)^{0.3} & z \leq z_{min} \end{cases} \quad (2.43)$$

$${}^xL_w = \frac{1}{12} {}^xL_u \quad (2.44)$$

where the expression for xL_w in Equation (2.44) is based on the assumption of approximately homogeneous wind flow conditions. Furthermore $L_1 = 100$ m is a reference length scale, $z_1 = 10$ m is a reference height and z_{min} is determined according to NS-EN 1991-1-4:2005+NA:2009, Table NA.4.1 (Standard Norge, 2005).

In order to include information about the spatial variation of the turbulence components, the normalised cross-spectra are used. In N400 the formulation by Dyrbye and Hansen (1997) is utilised to define the normalised cross-spectrum, C_{nm} , as

$$C_{nm}(f, \Delta x) = \frac{S_{nm}(f)}{\sqrt{S_n(f)S_m(f)}} \quad \begin{cases} n \in \{u, w\} \\ m \in \{u, w\} \end{cases} \quad (2.45)$$

where $S_{nm}(f)$ is the cross-spectral density, $S_{n,m}(f)$ denote the auto-spectral densities at two separate points and Δx is the horizontal or vertical distance between the two points.

Furthermore, N400 adopts the widely used expression proposed by Davenport (1961) for the normalised cross-spectrum, C , which is written as

$$C(f, \Delta x) = \exp\left(-K_i \frac{f\Delta x}{V}\right) \quad i \in \{u, w, uw\} \quad (2.46)$$

where K_i is the decay coefficient. Recommended values from N400 are $K_u = 10.0$ and $K_w = 6.5$.

Inserting Equation (2.46) into Equation (2.45) yields the expression for the cross-spectrum of the turbulence components, as

$$S_{nm}(f, \Delta x) = \sqrt{S_n(f)S_m(f)} \cdot \exp\left(-K_i \frac{f \Delta x}{V}\right) \quad (2.47)$$

2.6 Buffeting Response

The procedure for calculating the buffeting response of a structure is based on the theory described in Sections 2.3 and 2.5. There exist several methods for this purpose, with different levels of detail. In this thesis the single mode three component response calculation, described by Strømmen (2006), is utilised.

In this method a basic assumption is that the natural frequencies are well spaced out on the frequency axis. If the structure is considered as continuous, line-like and two-dimensional, the displacements can be described by a horizontal, vertical and torsional component. Thus, each mode shape contains three components, $\phi_{yi}(x)$, $\phi_{zi}(x)$ and $\phi_{\theta i}(x)$, in these directions respectively. As described in Section 2.3, the introduction of generalised coordinates leads to the uncoupled equation of motion. Therefore, the response from each mode, $\mathbf{r}_i(x, t)$, can be found separately, as

$$\mathbf{r}_i(x, t) = \begin{bmatrix} r_y(x, t) \\ r_z(x, t) \\ r_\theta(x, t) \end{bmatrix}_i = \begin{bmatrix} \phi_y(x) \\ \phi_z(x) \\ \phi_\theta(x) \end{bmatrix}_i \cdot \eta_i(t) = \boldsymbol{\phi}_i(x) \cdot \eta_i(t) \quad (2.48)$$

where r_{yi} , r_{zi} and $r_{\theta i}$ are the response components in the horizontal, vertical and torsional direction, respectively.

The equation of motion for each mode can be written as

$$\tilde{M}_i \cdot \ddot{\eta}_i(t) + \tilde{C}_i \cdot \dot{\eta}_i(t) + \tilde{K}_i \cdot \eta_i(t) = \tilde{Q}_i(t) + \tilde{Q}_{aei}(t, \eta_i, \dot{\eta}_i, \ddot{\eta}_i) \quad (2.49)$$

Here, \tilde{M}_i , \tilde{C}_i and \tilde{K}_i are the modal mass, damping and stiffness related to each mode, expressed as

$$\begin{aligned}
\tilde{M}_i &= \int_L (\boldsymbol{\phi}_i^T \cdot \mathbf{M}_0 \cdot \boldsymbol{\phi}_i) dx \\
\tilde{C}_i &= 2 \cdot \tilde{M}_i \cdot \omega_i \cdot \xi_i \\
\tilde{K}_i &= \omega_i^2 \cdot \tilde{M}_i
\end{aligned} \tag{2.50}$$

where $\mathbf{M}_0 = \text{diag} [m_y(x) \quad m_z(x) \quad m_\theta(x)]$ is the mass matrix containing the distributed masses in the horizontal, vertical and torsional direction and ξ_i is the damping ratio for each mode. Furthermore, \tilde{Q}_i and \tilde{Q}_{aei} constitute the total modal load acting on the structure for each mode, expressing the flow induced (\mathbf{q}) and the motion induced (\mathbf{q}_{ae}) part of the load, respectively, as

$$\begin{aligned}
\tilde{Q}_i(t) &= \int_{L_{exp}} \boldsymbol{\phi}_i^T \mathbf{q} dx = \int_{L_{exp}} \boldsymbol{\phi}_i^T \cdot \mathbf{B}_q \cdot \mathbf{v} dx \\
\tilde{Q}_{aei}(t, \eta_i, \dot{\eta}_i, \ddot{\eta}_i) &= \int_{L_{exp}} \boldsymbol{\phi}_i^T \mathbf{q}_{ae} dx = \tilde{C}_{aei} \cdot \dot{\eta}_i + \tilde{K}_{aei} \cdot \eta_i
\end{aligned} \tag{2.51}$$

where \tilde{C}_{aei} and \tilde{K}_{aei} are the modal aerodynamic damping and stiffness related to each mode, given in Equation (2.52), and L_{exp} is the part of the structure exposed to wind.

$$\tilde{C}_{aei} = \int_{L_{exp}} \boldsymbol{\phi}_i^T \cdot \mathbf{C}_{ae} \cdot \boldsymbol{\phi}_i dx \quad , \quad \tilde{K}_{aei} = \int_{L_{exp}} \boldsymbol{\phi}_i^T \cdot \mathbf{K}_{ae} \cdot \boldsymbol{\phi}_i dx \tag{2.52}$$

As the buffeting response is calculated for each mode separately, motion induced coupling between the modes needs to be avoided. Therefore, \mathbf{C}_{ae} and \mathbf{K}_{ae} have to be diagonal.

Furthermore, the equation of motion is converted into the frequency domain, by taking the Fourier transform on each side of Equation (2.49). This leads to the following equation

$$(-\tilde{M}_i \omega^2 + \tilde{C}_i i \omega + \tilde{K}_i) \cdot a_{\eta_i}(\omega) = a_{\tilde{Q}_i}(\omega) + a_{\tilde{Q}_{aei}}(\omega, \eta_i, \dot{\eta}_i, \ddot{\eta}_i) \tag{2.53}$$

where $a_{\eta_i}(\omega)$, $a_{\tilde{Q}_i}(\omega)$ and $a_{\tilde{Q}_{aei}}(\omega)$ are the Fourier amplitudes of $\eta_i(t)$, $\tilde{Q}_i(t)$ and $\tilde{Q}_{aei}(t)$, respectively. The i in the expression $\tilde{C}_{aei} i \omega$ is the imaginary unit, $\sqrt{-1}$. The Fourier amplitudes of $\tilde{Q}_i(t)$ and $\tilde{Q}_{aei}(t)$ are given as

$$a_{\tilde{Q}_i}(\omega) = \int_{L_{exp}} \boldsymbol{\phi}_i^T \cdot \mathbf{B}_q \cdot \mathbf{a}_v dx \quad , \quad a_{\tilde{Q}_{aei}}(\omega) = (\tilde{C}_{aei} i \omega + \tilde{K}_{aei}) a_{\eta_i}(\omega) \tag{2.54}$$

where \mathbf{a}_v is the Fourier amplitude of $\mathbf{V} = [u \quad w]^T$. In Equations (2.53) and (2.54), the relations $a_{\dot{\eta}_i} = i \omega a_{\eta_i}$ and $a_{\ddot{\eta}_i} = (i \omega)^2 a_{\eta_i}$, between the Fourier amplitudes of the response, η_i , and its derivatives, $\dot{\eta}_i$ and $\ddot{\eta}_i$, are utilised.

By collecting all terms depending on $a_{\eta_i}(\omega)$ in Equation (2.53) on the left side, the expression for the Fourier amplitude of the modal response from mode i is obtained as

$$a_{\eta_i}(\omega) = \tilde{H}_i(\omega) \cdot a_{\tilde{Q}_i}(\omega) \quad (2.55)$$

where the modal frequency response function, $\tilde{H}_i(\omega)$, is introduced. The expression for $\tilde{H}_i(\omega)$ is given as

$$\tilde{H}_i(\omega) = [-\tilde{M}_i \cdot \omega^2 + (\tilde{C}_i - \tilde{C}_{aei}) \cdot i\omega + (\tilde{K}_i - \tilde{K}_{aei})]^{-1} \quad (2.56)$$

Furthermore, the one-sided auto-spectrum of the modal response, $S_{\eta_i}(\omega)$, can be obtained as

$$S_{\eta_i}(\omega) = \lim_{T \rightarrow \infty} \frac{1}{\pi T} E \left[|a_{\eta_i}(\omega)|^2 \right] = |\tilde{H}_i(\omega)|^2 \cdot \lim_{T \rightarrow \infty} \frac{1}{\pi T} E \left[|a_{\tilde{Q}_i}(\omega)|^2 \right] = |\tilde{H}_i(\omega)|^2 \cdot S_{\tilde{Q}_i}(\omega) \quad (2.57)$$

whereas the one-sided auto-spectrum of the modal load, $S_{\tilde{Q}_i}(\omega)$, can be found as

$$\begin{aligned} S_{\tilde{Q}_i}(\omega) &= \lim_{T \rightarrow \infty} \frac{1}{\pi T} E \left[|a_{\tilde{Q}_i}(\omega)|^2 \right] \\ &= \int_{L_{exp}} \int_{L_{exp}} \boldsymbol{\phi}_i^T(x_1) \cdot \mathbf{B}_q \cdot \lim_{T \rightarrow \infty} \frac{1}{\pi T} E \left[|\mathbf{a}_V \cdot \mathbf{a}_V^T| \right] \cdot \mathbf{B}_q^T \cdot \boldsymbol{\phi}_i(x_2) dx_1 dx_2 \\ &= \int_{L_{exp}} \int_{L_{exp}} \boldsymbol{\phi}_i^T(x_1) \cdot \mathbf{B}_q \cdot \mathbf{S}_V \cdot \mathbf{B}_q^T \cdot \boldsymbol{\phi}_i(x_2) dx_1 dx_2 \end{aligned} \quad (2.58)$$

Here, $\mathbf{S}_V(\Delta x, \omega)$ is the cross-spectral density matrix of the turbulence. The cross-spectra between u and w are often neglected, as they have little influence on the response (Øiseth et al., 2013). Thus, the expression for \mathbf{S}_V can be written as

$$\mathbf{S}_V(\Delta x, \omega) = \begin{bmatrix} S_{uu}(\Delta x, \omega) & 0 \\ 0 & S_{ww}(\Delta x, \omega) \end{bmatrix} \quad (2.59)$$

where the expressions for the cross-spectra $S_{uu,ww}$ are given in Equation (2.47). By assuming a homogeneous wind field, these spectra can be simplified to (Fenerci and Øiseth, 2018b)

$$S_{uu,ww}(\Delta x, \omega) = S_{u,w}(\omega) \cdot C_{uu,ww}(\Delta x, \omega) \quad (2.60)$$

Next, the one-sided auto-spectra of the three displacement components for mode i can be obtained at a given position, x_r , as

$$\mathbf{S}_{r_i}(x_r, \omega) = \begin{bmatrix} S_{r_y r_y} \\ S_{r_z r_z} \\ S_{r_\theta r_\theta} \end{bmatrix}_i = \begin{bmatrix} \phi_y^2(x_r) \\ \phi_z^2(x_r) \\ \phi_\theta^2(x_r) \end{bmatrix}_i \cdot S_{\eta_i}(\omega) \quad (2.61)$$

where $S_{r_y r_y}$, $S_{r_z r_z}$ and $S_{r_\theta r_\theta}$ are the auto-spectra for the horizontal, vertical and torsional displacement response for mode i .

Furthermore, the spectra for the acceleration response can be obtained as

$$\mathbf{S}_{\ddot{r}_i}(x_r, \omega) = \omega^4 \cdot \mathbf{S}_{r_i}(x_r, \omega) \quad (2.62)$$

Finally, the acceleration response of the three displacement components, represented by the variances, can be obtained by integration of the auto-spectra and summation of the contributions from each mode. Thus,

$$\begin{bmatrix} \sigma_{\ddot{r}_y \ddot{r}_y}^2(x_r) \\ \sigma_{\ddot{r}_z \ddot{r}_z}^2(x_r) \\ \sigma_{\ddot{r}_\theta \ddot{r}_\theta}^2(x_r) \end{bmatrix} = \sum_{i=1}^{N_{mod}} \begin{bmatrix} \sigma_{\ddot{r}_y \ddot{r}_y}^2(x_r) \\ \sigma_{\ddot{r}_z \ddot{r}_z}^2(x_r) \\ \sigma_{\ddot{r}_\theta \ddot{r}_\theta}^2(x_r) \end{bmatrix}_i = \sum_{i=1}^{N_{mod}} \int_0^\infty \mathbf{S}_{\ddot{r}_i}(x_r, \omega) d\omega \quad (2.63)$$

where $\sigma_{\ddot{r}_y \ddot{r}_y}^2(x_r)$, $\sigma_{\ddot{r}_z \ddot{r}_z}^2(x_r)$ and $\sigma_{\ddot{r}_\theta \ddot{r}_\theta}^2(x_r)$ are the total variances of the horizontal, vertical and torsional acceleration response at position x_r . By taking the square root of these terms, the root mean square (RMS) acceleration responses are obtained.

Chapter 3

Methodology

3.1 The Hålogaland Bridge and the Monitoring System

The Hålogaland Bridge is a suspension bridge located near Narvik in northern Norway. The bridge crosses the Rombak fjord, which is a branch of the Ofotfjord. In Figures 3.1 and 3.2 the bridge is pictured and the location of the bridge is shown. The Hålogaland Bridge is the second longest suspension bridge in Norway, with its main span of 1145 m. The cross section of the bridge girder is a steel box, which is 18.6 m wide and 3 m high. The girder is carried by two main cables and 110 hangers. Furthermore, the main cables are supported by two towers reaching up to approximately 180 m above sea level.



Figure 3.1: Picture of the Hålogaland Bridge. Courtesy of NTNU.

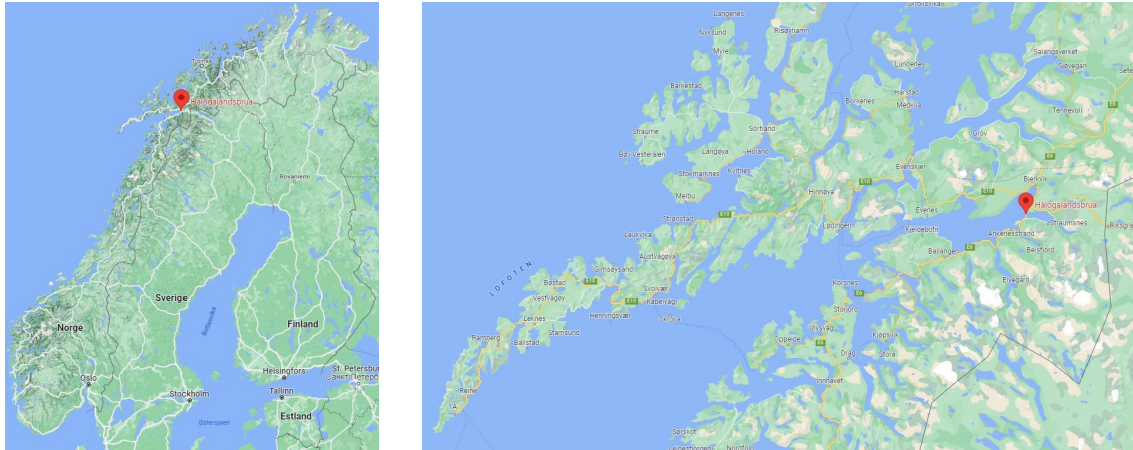


Figure 3.2: The location of the Hålogaland Bridge. The bridge location is marked with a red pin. (Map data ©2022 Google, www.google.com/maps)

The slenderness of the Hålogaland Bridge, due to the combination of a remarkably long span and a relatively small cross section, leads to particularly low natural frequencies. Furthermore, the bridge is exposed to wind loading, which is dominated by frequencies in the same frequency range. Thus, the bridge is particularly prone to wind-induced response. In order to gain more knowledge about how the wind affects the behaviour of long-span suspension bridges, a monitoring system has been installed on the Hålogaland Bridge.

The monitoring system is extensive, and is able to measure acceleration, temperature, strain and wind. As the main objective of this thesis is to compare the predicted buffeting response with the measured response, only the acceleration data and the wind data are studied. In total there are 22 triaxial accelerometers located in various positions on the bridge deck, while there are 10 anemometers located on the hangers. The monitoring system consists of 11 logger boxes: 2 in the bridge towers and 9 along the bridge span. Each logger is connected to an accelerometer pair consisting of one accelerometer on each side of the bridge deck, in addition to a varying number of anemometers. All sensors are connected to the nearest logger box. An illustration of the bridge with the monitoring system is given in Figure 3.3 (Petersen et al., 2021). Data from 8 accelerometer pairs and 5 anemometers along the bridge are utilised in this thesis. Figure 3.4 illustrates the sensors that are utilised, and the x-coordinates of the sensors along the bridge can be found in Tables 3.1 and 3.2. Inside the logger boxes, a NI CompactRIO controller is used for data acquisition. This main hardware unit is programmed by the LabVIEW software to sample and filter the measurement data from the sensors. In addition, a timestamp with accurate time from a Trimble Bullet GPS antenna is added, before the data is saved locally on hard drives.

The CompactRIOs are connected to the internet, and transfers the data to a server with regular intervals. A more thorough description of the monitoring system on the Hålogaland Bridge can be found in [Petersen et al. \(2021\)](#).

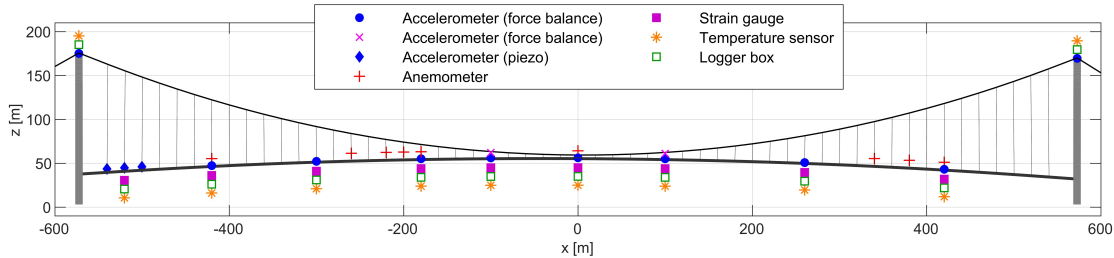


Figure 3.3: Illustration of the monitoring system on the Hålogaland Bridge ([Petersen et al., 2021](#)).

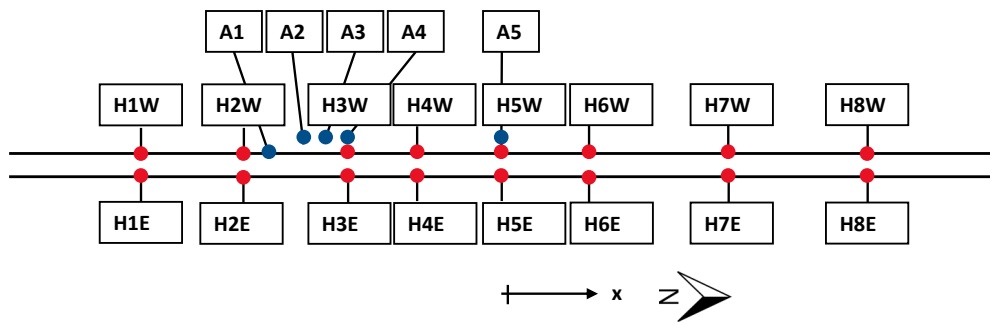


Figure 3.4: Top view of the sensors on the Hålogaland Bridge that are utilised in this thesis. The red and blue dots illustrate accelerometers and anemometers, respectively.

Table 3.1: x-coordinates of the accelerometers on the Hålogaland Bridge that are utilised in this thesis. The coordinate system has its origin at the midspan of the bridge, with the positive x-direction pointing to the north end of the bridge.

Name	H1E/ H1W	H2E/ H2W	H3E/ H3W	H4E/ H4W	H5E/ H5W	H6E/ H6W	H7E/ H7W	H8E/ H8W
x-coordinate [m]	-420	-300	-180	-100	0	100	260	420

Table 3.2: x-coordinates of the anemometers on the Hålogaland Bridge that are utilised in this thesis. The coordinate system has its origin at the midspan of the bridge, with the positive x-direction pointing to the north end of the bridge.

Name	A1	A2	A3	A4	A5
x-coordinate [m]	-260	-220	-200	-180	0

3.2 Abaqus Model of the Bridge

A finite element model of the Hålogaland Bridge, modelled in [Abaqus \(2019\)](#), has been provided by Øyvind Wiig Petersen. This model is used to obtain the modal parameters of the bridge, which are needed to predict the buffeting response. Here, only a brief description of the model is given.

The main parts of the bridge are the two towers, the two main cables, the hangers and the girder. Figure 3.5 shows the Abaqus model of the bridge. The geometry of the model is based on as-built geometry. The main parts of the bridge are modelled with beam elements. Thus, the three-dimensional parts are modelled as one-dimensional. This is assumed to be an adequate approximation, as the cross sections of all parts of the bridge are small compared to the global dimensions along the beam axes. Both B31 and B32 elements are used, which are 2-node linear and 3-node quadratic beam elements in space. All degrees of freedom are active for these elements, and a lumped mass formulation is used for the dynamic calculations ([Abaqus, 2019](#)).

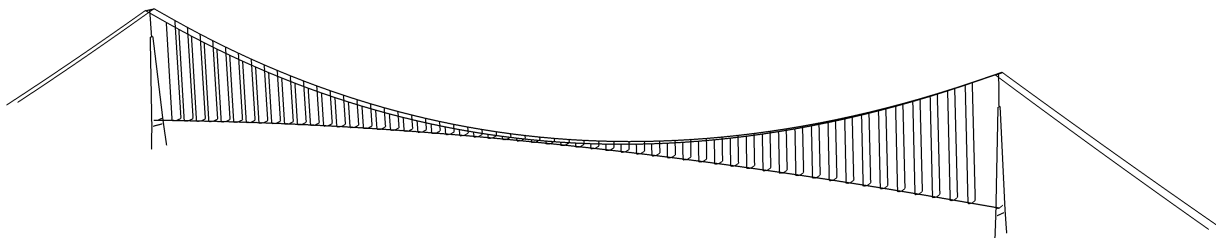


Figure 3.5: Abaqus model of the Hålogaland Bridge.

The bridge girder, which is the essential part of the model for the buffeting response calculations, is modelled with 286 elements of type B32. Consequently, each element is approximately 4 m long. A modal analysis step is performed to extract the modal parameters of the bridge. The Lanczos eigensolver is chosen for this purpose, which means that Abaqus solves the eigenvalue problem represented by Equation (2.16) ([Abaqus, 2019](#)).

3.3 Data Processing

The data from the monitoring system on the Hålogaland Bridge are stored on a server as Tdms-files, each containing 8 hours of measurements from one logger box. A Tdms-file is structured using a hierarchical system with three levels. The highest level is the file itself. The file contains

groups, and the groups contain channels, where the latter act as arrays. A Tdms-file can be read using the Python package *npTDMS* (Reeve, 2021). For a Tdms-file from a logger box on the Hålogaland Bridge, all information regarding the accelerations is gathered in one group, whereas the wind data are organised into one group per anemometer. All groups contain a timestamp channel, which gives information about the time the data are sampled. The timestamps are given in Unix time with nanosecond precision, meaning that the time is given as nanoseconds elapsed since 00:00:00 UTC on 1 January 1970. The acceleration data in each group are separated into channels for data in x-, y-, and z-direction from each accelerometer in the pair. In addition to the raw data, the group also contains information about the coordinates of the accelerometers, as well as the orientation of the accelerometers in relation to the global coordinate system of the bridge. This is used to verify that the local coordinate system of each accelerometer is consistent with the global one. As previously mentioned, the wind data are separated into one group per anemometer. Each group has one channel for the direction of the horizontal wind velocity, one channel for the magnitude of the horizontal wind velocity and one channel for the vertical wind velocity.

In order to process the data, the *npTDMS* package is used to read the data into Python. The first step of the processing is to time synchronise the data. Even though the logger boxes are supposed to start recording wind and acceleration at the same time, it is registered time lags of up to several minutes between different accelerometers and anemometers. In order to ensure that the wind and acceleration data from the considered loggers have the same time axes, the data are time synchronised. This is achieved by generating a "master time vector" that starts at the last recorded initial timestamp and ends at the earliest recorded final timestamp of all the considered accelerometers and anemometers. The data are then synchronised, by interpolating the data from the initial, sensor-specific time vector to the master time vector. Acceleration and wind data are interpolated to the same master time vector. As the wind data are sampled at 32 Hz and the acceleration data are sampled at 64 Hz, the acceleration data are first low-pass filtered and downsampled to 32 Hz. A Butterworth filter of order 10 and with a cut-off frequency of half the new Nyquist frequency is used for the low-pass filtering. All low-pass filtering of the data is done in this way throughout the thesis. Further processing is done separately for the acceleration data and the wind data, due to different requirements. The main steps of the routines are explained below, but for details see Appendix A.1.

The data from the accelerometers are given in volt, and has to be converted to m/s^2 . This is done by dividing the data by a conversion factor given in the Tdms-file and then multiplying

with the gravity constant $g = 9.82 \text{ m/s}^2$, due to the unit of the conversion factor. The acceleration data are once again low-pass filtered and downsampled to a final sampling frequency of 2 Hz, since the frequencies of interest are in the range 0-1 Hz. As a final step, the data are divided into intervals of a desired time period.

The anemometers are more susceptible to errors than the accelerometers, and therefore the wind data have to be cleaned more thoroughly. The Python codes developed to clean the data can be found in Appendix A.1. Some of these are based on codes written by Knut Andreas Kvåle and Aksel Fenerci in Matlab, and then translated to Python by the authors of this thesis. The first step is to remove all error values from the data. Each sample from the anemometers comes with a status code, which contains information about the quality of the sample. Samples with an error code are removed from the time series. In order to maintain the sampling frequency, the time series are then filled out using linear interpolation. The time series are also cleaned for values that exceed six times the standard deviation, in order to avoid anomalies. Finally, since the directional data contain spikes due to the circular nature of the data, they have to be corrected so that they only contain values between 0 and 360 degrees.

Similar to the acceleration data, the wind data are low-pass filtered and downsampled to 2 Hz and divided into intervals of a specified length. For each interval, the mean wind speed and mean wind direction is calculated. Then, the instantaneous horizontal wind is decomposed into an along-wind turbulence component and an across wind turbulence component using the transformation

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} U \cdot \cos(\phi) - V \\ U \cdot \sin(\phi) \\ w \end{bmatrix} \quad (3.1)$$

Here, U is the instantaneous horizontal wind velocity, ϕ is the difference between the mean wind direction (α) and the instantaneous direction, V is the mean wind speed and u , v , and w are the turbulence components. The decomposition of the instantaneous horizontal wind is illustrated in Figure 3.6. w is equivalent to the vertical wind velocity. A high-pass filter is then applied to the turbulence data in each interval, using a Butterworth filter of order 2 and with a cut-off frequency of 1/300 Hz. This is done in order to remove spurious trends from the data.

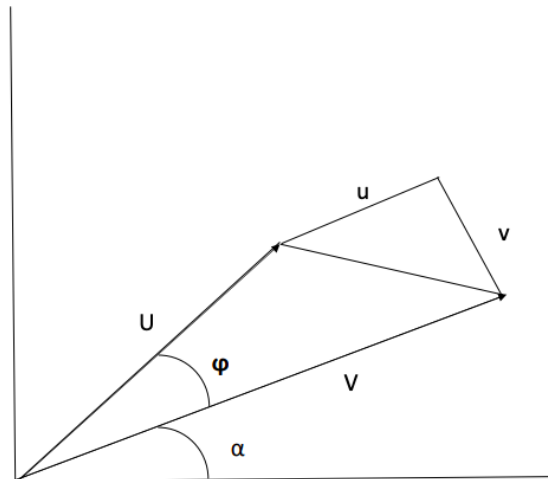


Figure 3.6: Illustration of the decomposition of instantaneous horizontal wind, U , into turbulence components, u and v .

3.4 Covariance-Driven Stochastic Subspace Identification

The natural frequencies, damping ratios and mode shapes of the Hålogaland Bridge are estimated with covariance-driven stochastic subspace identification (Cov-SSI). The theory behind the method is explained in Section 2.4. All calculations are conducted in Python, and are based on the functions included in the KOMA package, developed by Kvåle (2022). The results from Cov-SSI are used to verify the modal parameters from the Abaqus model.

Acceleration data from the accelerometers H1E/H1W-H8E/H8W (see Figure 3.4) are used as the input to conduct the Cov-SSI. The Tdms-files from the data loggers connected to these accelerometers are used to extract time synchronised and processed x-, y- and z-acceleration data, divided in intervals of 30 minutes. The same 30 minute interval of acceleration data from all accelerometers are then utilised for the Cov-SSI.

The scripts for conducting Cov-SSI can be found in Appendix A.4. It is difficult to know in advance what to choose as the optimal values for the number of blockrows (i), the stabilisation level (s) and the stabilisation criteria. Therefore, different options have to be investigated. The procedure starts by calculating all the complex poles and eigenvectors from the measurement data for all the chosen orders of the system, with the chosen number of blockrows. No weighting of the block-Hankel matrix is performed. Next, the stable poles and corresponding eigenvectors are found from all poles, based on the chosen stabilisation level and stabilisation criteria. The

natural frequencies of the poles are used as the modal indicator. Furthermore, the natural frequencies and damping ratios for the stable poles are found from Equation (2.33). A stabilisation plot is constructed, by plotting the system orders against the corresponding natural frequencies for the stable poles.

This procedure is repeated for acceleration data from several different Tdms-files and several different 30-minute intervals. One specific 30 minute interval, from 4th of February 2022 starting at 00:00:00, is then chosen to extract the modal parameters of the Hålogaland Bridge. The reason is that the stabilisation plot from this interval includes the evident poles from all the investigated plots. The mean wind speed corresponding to the chosen interval is 18 m/s. The stabilisation plot for this interval is constructed for different values of the parameters i and s . In the end, these values are chosen as $i = 24$ and $s = 6$. This provides a clear stabilisation plot with little scatter, while still including all the evident poles. The calculations are conducted for the system orders $n = 2, 4, \dots, 250$, since no new stable poles are observed for higher system orders. The stabilisation criteria used to determine the stable poles are chosen as the default criteria from KOMA.

Finally, this stabilisation plot is utilised to determine the physical modes, from the straight sequences of stable poles. The natural frequencies and corresponding damping ratios are calculated from the stable poles. To obtain the mode shapes of the bridge girder in horizontal, vertical and torsional direction, the following matrix is used to transform the horizontal and vertical mode shape values on each side of the bridge into horizontal, vertical and torsional values at the midspan.

$$\begin{bmatrix} y \\ z \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{-1}{B} & \frac{1}{B} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ z_1 \\ z_2 \end{bmatrix} \quad (3.2)$$

Here, B is the width of the bridge deck, y , z and θ are the new mode shape values at the midspan of the bridge, and $y_{1,2}$ and $z_{1,2}$ are the initial mode shape values on each side of the bridge girder.

Due to the damping of the bridge, the components of the mode shapes are complex numbers, which leads to a phase shift between them. Thus, the maximum amplitude of the components does not occur at the same time (Kvåle et al., 2017). Therefore, a function from KOMA is utilised

to obtain the mode shapes where the absolute values of the real parts of all components are maximised. Furthermore, the mode shapes are plotted as the real parts of the components.

3.5 Wind Tunnel Test

In order to obtain the aerodynamic derivatives and the load coefficients required to model the wind load acting on the Hålogaland Bridge, wind tunnel tests of a scaled section model of the bridge girder are performed. The tests are carried out at NTNU, using a wind tunnel with a test section of approximately $2 \times 3 \text{ m}^2$ and a length of 11 m. The scale of the section model is 1:70, and includes railings in order to reflect the real behaviour of the bridge accurately. The wind tunnel tests are performed as forced vibration tests. Thus, the section model is set in various motions and exposed to different wind velocities, while the aerodynamic forces acting on the model are measured. The tests are performed for winds approaching the bridge from both the east and the west side. The data from the wind tunnel tests are then processed by Ole Andre Øiseth, which results in files containing the necessary data to obtain the load coefficients and the expressions for the ADs. The data for winds from both east and west are investigated, but as it appears that the data are very similar for the two directions, only the westerly wind data are utilised further in the calculations.

The file used to obtain the expressions for the aerodynamic derivatives contains values for the ADs and the corresponding reduced velocities. As the buffeting response of the Hålogaland Bridge is calculated for each mode separately, only expressions for the ADs on the diagonal of the aerodynamic damping and stiffness matrices given in Equation (2.40) need to be obtained. Since the ADs are used to model the self-excited forces given by Equation (2.39), P_1^* , H_1^* and A_2^* are first multiplied by the reduced frequency, K , while P_4^* , H_4^* and A_3^* are multiplied by K^2 . Then, these terms are plotted against the reduced velocities, and appropriate curve fits are obtained. The fitted curves are only valid for the range of reduced velocities from the experimental data. Outside this range, the curves are estimated as constant values, corresponding to the start and end values of the curves. To determine the final expressions for the ADs, the obtained expressions for KP_1^* , KH_1^* and KA_2^* are divided by K , while the expressions for $K^2P_4^*$, $K^2H_4^*$ and $K^2A_3^*$ are divided by K^2 .

The file used to obtain the load coefficients, \bar{C}_D , \bar{C}_L , \bar{C}_M , C'_D , C'_L and C'_M , contains values for the load coefficients, C_D , C_L and C_M , and the corresponding angles of flow incidence. First, expressions for C_D , C_L and C_M as functions of the angle of flow incidence are determined through

appropriate curve fits of the data from the file. Then, \bar{C}_D , \bar{C}_L and \bar{C}_M are obtained as the function values for the mean value of the angles of incidence. C'_D , C'_L and C'_M are obtained as the derivatives of the functions, evaluated at the mean value of the angles of incidence. The mean value of the angles of incidence is assumed to be zero.

3.6 Extraction of Data from the Monitoring System

The buffeting response of the Hålogaland Bridge is calculated at the midspan. Wind and acceleration data from a total of 292 Tdms-files from two logger boxes are utilised to extract necessary data for the response calculations. Acceleration data are extracted from accelerometers H5E and H5W, while wind data are extracted from anemometers A1-A5 (see Figure 3.4). These sensors are used because they are located at or near the midspan of the bridge. The data are processed as described in Section 3.3 and divided into intervals of 10 minutes. Due to errors in Tdms-files from 2021, as well as only having files from May and June 2021 where the wind speed is mostly relatively low, only files from February and March 2022 are considered in this thesis. A period of 10 minutes for the intervals is used since it is long enough to represent the wind in a satisfactory manner, while it is short enough to generally satisfy the requirement of stationarity (Tamura and Kareem, 2013). Furthermore, due to the fact that low wind speeds are often associated with high non-stationarity, only recordings with a mean wind speed above 3 m/s are used for further analysis.

Since the main focus of the thesis is to compare the predicted buffeting response with the measured response, acceleration data are extracted from the monitoring system. As mentioned, acceleration data are extracted from the accelerometers at the midspan: H5E and H5W. The predicted buffeting response is calculated as RMS acceleration response in the horizontal, vertical and torsional direction. Therefore, the acceleration data from the measurements are transformed into a lateral, vertical and torsional component, using the transformation matrix given in Equation (3.2). Then, the standard deviations of the three response components are calculated for each 10 minute interval.

Wind data from anemometer A5 are used to extract mean wind speeds, mean wind directions and turbulence components. The wind data are further utilised to improve the buffeting response predictions. In the predictions, the wind load is modelled by the use of the auto-spectra and the normalised cross-spectra of the turbulence components. In N400, these spectra are defined as in Equations (2.42) and (2.46), with suggestions for the values of the spectral parameters,

$A_{u,w}$, and the decay coefficients, $K_{u,w}$. N400 refers to NS-EN 1991-1-4:2005+NA:2009 (Standard Norge, 2005) for the calculation of the turbulence intensities, which are used to calculate the standard deviations of the turbulence components after Equation (2.34). However, since the turbulence parameters given in N400 are not site-specific, they do not reflect the actual wind field at the location of the Hålogaland Bridge. Hence, the turbulence parameters from N400 are not utilised in the response calculations. Better approximations of these values are estimated by utilising the measurement data from the monitoring system. Therefore, the turbulence intensities for each 10 minute interval is calculated according to Equation (2.34). These values are calculated using data from anemometer A5. In addition, more appropriate values of $A_{u,w}$ and $K_{u,w}$, adjusted for each 10-min interval, are calculated.

Wind data from anemometer A5 are used to approximate $A_{u,w}$. For each 10 minute interval from the wind recordings, the auto-spectral densities of the along-wind and the vertical turbulence components are estimated. Welch procedure is used for the estimation, with 8 segments and an overlap of 50%. The values of $A_{u,w}$ are then estimated for each 10 minute interval, through a non-linear least squares fit between the auto-spectra from the measurements and the auto-spectra using the Kaimal model as given in Equation (2.41).

For the estimation of $K_{u,w}$, it is chosen to use data from anemometers A1-A4. Thus, three anemometer pairs are used, with separation distances of 20, 40 and 80 m. These anemometers are chosen because of the relatively small separation distances, to overcome weaknesses of Davenport's expression for large separation distances (Simiu and Scanlan, 1996). The normalised cross-spectra, given in Equation (2.45), are estimated from the wind data for the three anemometer pairs, for each 10 minute interval. Welch procedure is used to estimate the spectral densities, with 8 segments and an overlap of 50%. The values of $K_{u,w}$ are then estimated for each 10 minute interval, through a non-linear least squares fit between the normalised cross-spectra from the measurements and Davenport's expression for the normalised cross-spectra, given in Equation (2.46). When the spatial correlation of the turbulence components is not apparent, the expression is unable to represent the normalised cross-spectra satisfactory. Thus, values of $K_{u,w}$ above 20 are deemed non-coherent and are consequently ignored. Python scripts used to obtain the fitted values of $A_{u,w}$ and $K_{u,w}$ are found in Appendix A.3, where also the script for extracting all the relevant wind- and acceleration data from all the files can be found.

As the wind varies with time, the estimates of the turbulence parameters, $I_{u,w}$, $A_{u,w}$ and $K_{u,w}$, also show significant variation between each interval. Therefore, the probability distributions of the turbulence parameters are estimated in order to reflect the variability of the wind field. Log-

normal probability distributions are fitted to the data, by calculating the lognormal distribution parameters, $\tilde{\mu}$ and $\tilde{\sigma}$, as the mean and standard deviation of the natural logarithm of the data, as explained in Section 2.2. The turbulence intensities are divided into three segments, due to a dependence on the mean wind speed, and a lognormal distribution is fitted to each segment. Later, the probability distributions will be utilised to choose values of the turbulence parameters for the buffeting response predictions. Since the framework for the buffeting theory is mainly applicable for wind that comes perpendicular to the bridge x-axis, samples of the turbulence parameters which correspond to wind with mean direction deviating more than 20 degrees from the two perpendicular directions (90 and 270 degrees), are ignored when fitting the probability distributions. Instances where values of $A_{u,w}$ and $K_{u,w}$ coincide with the boundaries of the considered interval for the least squares fitting are discarded when establishing the probability distributions. The reason is that these values most likely should be outside the interval, and the probability distributions will be erroneously represented by including these instances.

3.7 Probabilistic Wind Field Model

As established, the wind field at the Hålogaland Bridge is modelled by the use of the auto-spectra and the normalised cross-spectra given by Equations (2.41) and (2.46). Thus, six turbulence parameters are needed: $\sigma_{u,w}$, $A_{u,w}$ and $K_{u,w}$. As stated, these parameters vary with time, and in the previous section this was considered by establishing the probability distributions of each parameter. In a second approach, a probabilistic model of the wind field is established by further investigation of the probability distributions, the correlation between the parameters and the dependence of the parameters on the mean wind speed and direction. Simulations from this model will be utilised in the buffeting response predictions.

In this thesis, only the standard deviations of the turbulence components, $\sigma_{u,w}$, are investigated and simulated, due to time limitations. Moreover, these parameters have shown to be the most influential parameters for the wind field. To begin with, $\sigma_{u,w}$ -values are calculated from the extracted $I_{u,w}$ -values by the use of Equation (2.34). These values are utilised to establish a probabilistic model of the turbulence standard deviations. The mean wind speeds and the standard deviations of the turbulence components are first divided into variables from easterly and westerly winds, as the winds from these directions show clear differences. The easterly and westerly winds are defined as the winds with mean directions (α) $0^\circ < \alpha \leq 180^\circ$ and $180^\circ < \alpha \leq 360^\circ$, respectively. Here, 0° and 180° are defined as the north and the south direction, respectively. All

further calculations in this section are conducted for the easterly and westerly winds separately.

The dependence of the mean wind speed on $\sigma_{u,w}$ is investigated by plotting $\sigma_{u,w}$ against V . These plots show that there is a linear dependence between the mean wind speed and the turbulence standard deviations. Due to this dependence, the probability distributions of $\sigma_{u,w}$ need to be determined conditional to the mean wind speed. Therefore, the turbulence standard deviations are divided into intervals based on the corresponding mean wind speed. The easterly and westerly wind data are divided into intervals with approximately 200 and 150 values in each interval, respectively. Then, the lognormal distribution parameters, $\tilde{\mu}$ and $\tilde{\sigma}$, are obtained for each interval and plotted against the corresponding mean wind speed. In addition, the correlation coefficients of σ_u and σ_w are computed for each interval, to investigate if there is any dependence on the mean wind speed.

At a mean wind speed of approximately 11 m/s, the $\tilde{\mu}$ -parameters seem to stabilise (Figures 4.24 and 4.25). Thus, expressions for these parameters as functions of the mean wind speed are established through appropriate curve fits of the $\tilde{\mu}$ -values above 11 m/s. As the $\tilde{\sigma}$ -parameters do not seem to stabilise at any mean wind speed, these parameters are modelled as constant values by calculating the parameters from all data in the mean wind speed range where the curve fits are obtained for $\tilde{\mu}$. Furthermore, the plot of the correlation coefficients of σ_u and σ_w against the mean wind speed shows that these values remain relatively constant. Thus, the correlation coefficients are also modelled as constant values for the data in the same wind speed range.

The lognormal distribution parameters and correlation coefficients for $\sigma_{u,w}$ are utilised to generate correlated lognormally distributed random samples of $\sigma_{u,w}$. To begin with, functions from the *NumPy* library in Python are utilised to obtain random samples from a normal distribution. As explained in Section 2.2, only the mean value and the covariance matrix of the normally distributed random variables are needed. The mean value corresponds to the parameter $\tilde{\mu}$ of the lognormal distribution, while the covariance matrix is obtained from the parameter $\tilde{\sigma}$ of the lognormal distribution and the correlation coefficients of the lognormally distributed random variables, by the use of Equation (2.13). The correlated lognormally distributed random variables are then obtained by taking the natural exponent of the generated samples.

In this way, correlated lognormally distributed random samples of $\sigma_{u,w}$ from the probabilistic model can be generated for each 10 minute wind recording from the Hålogaland Bridge, according to the mean wind speed and direction. The values are only simulated for the recordings with mean wind speed in the range that the lognormal distribution parameters and the corre-

lation coefficients are modelled for, i.e. wind recordings with mean wind speed larger than 11 m/s.

The script used to obtain the expressions for the lognormal distribution parameters and the correlation coefficients for the probabilistic model of the turbulence standard deviations can be found in Appendix A.2. There, the script used to simulate the turbulence standard deviations can also be found.

3.8 Calculation of Buffeting Response

The buffeting response of the Hålogaland Bridge is predicted based on a single mode three component response calculation, as described in Section 2.6. All calculations are conducted in Python, by the use of a function which takes the mean wind speed, the turbulence parameters, $A_{u,w}$, $K_{u,w}$, $I_{u,w}$, and the location on the bridge as the input. The function returns the RMS acceleration response in the lateral, vertical and torsional direction. The Python function can be found in Appendix A.3. This function is utilised to calculate the buffeting response of the bridge at the midspan.

Natural frequencies, mode shapes and modal masses for the response calculations are obtained from the Abaqus model of the bridge. The modal parameters are extracted from the model by the use of a Python script, provided by Øyvind Wiig Petersen. This script takes the ODB (output database) file from the Abaqus model as input and gives several text files with the parameters as output. Information from these text files are then imported to Python as arrays. The structural damping ratio is assumed to be 0.5% for all modes, based on suggestions from N400 (Vegdirektoratet, 2015). The modal damping and stiffness for each mode is calculated based on Equation (2.50). As the wind is dominated by frequencies in the range 0-1 Hz, the modes with natural frequencies up to 1 Hz are included in the response calculations. Thus, 56 modes are included.

The auto-spectral density of the modal load, given by Equation (2.58), is computed for each mode. The load coefficients in \mathbf{B}_q are found from the wind tunnel tests, as explained in Section 3.5, and the cross sectional admittance functions are set to unity. Furthermore, the cross-spectral density matrix of the turbulence is computed as given in Equation (2.59), where the cross-spectra are calculated from Equation (2.60). The auto-spectra and the normalised cross-spectra of the turbulence components from Equations (2.41) and (2.46) are converted to functions of the frequency in rad/s.

The response calculations are performed for several different cases of the cross-spectral density matrix of the turbulence. The cases are based on different alternatives of the turbulence parameters, $A_{u,w}$, $K_{u,w}$ and $I_{u,w}$. Some of the cases are included to investigate the influence of the different turbulence parameters on the response predictions. The parameters are taken as different percentile values from their probability distributions, and for one case the simulations of $\sigma_{u,w}$ are also utilised. The following alternatives for the parameters are tested:

- Case 1: $A_{u,w}$, $K_{u,w}$ and $I_{u,w}$ as the 50th percentile values.
- Case 2: $A_{u,w}$ as the 50th percentile values, $K_{u,w}$ as the 50th percentile values and $I_{u,w}$ as the 95th percentile values.
- Case 3: $A_{u,w}$ as the 50th percentile values, $K_{u,w}$ as the 5th percentile values and $I_{u,w}$ as the 50th percentile values.
- Case 4: $A_{u,w}$ as the 5th percentile values, $K_{u,w}$ as the 50th percentile values and $I_{u,w}$ as the 50th percentile values.
- Case 5: $A_{u,w}$ as the 50th percentile values, $K_{u,w}$ as the 50th percentile values and $I_{u,w}$ calculated from the simulated $\sigma_{u,w}$ -values (see Section 3.7).

The percentile values for case 1-4 are calculated from the probability distributions that are obtained as explained in Section 3.6. Thus, only wind data with mean wind speeds larger than 3 m/s and that comes approximately perpendicular to the bridge x-axis are considered. Also, since the turbulence intensities are divided into three segments based on the mean wind speeds, three different values for these percentiles are regarded in each case, depending on the mean wind speed. For the percentile values of $A_{u,w}$ and $K_{u,w}$ for case 5, only wind data with mean wind speeds larger than 11 m/s are considered, since the probabilistic model utilised to simulate the $\sigma_{u,w}$ -values only apply to this wind speed range. To be consistent, the percentiles of $A_{u,w}$ for this particular case are calculated from probability distributions corresponding to easterly and westerly winds separately. For the percentiles of $K_{u,w}$ there is no distinction between easterly and westerly winds, as the information needed to do this is not obtained. Furthermore, for case 1-4 the response is calculated for mean wind speeds in the range 0-25 m/s, while for case 5 the response is calculated for the mean wind speeds used to simulate the $\sigma_{u,w}$ -values.

The auto-spectral density of the modal load is computed for frequencies in the range 0-1 Hz. As the calculation of the spectrum is computational demanding, it is first computed for only 50

frequency values. Then, the frequency axis for the spectrum is refined, to assure an accurate response calculation. This is done by linear interpolation of the spectrum computed for the initial frequency values, to a new frequency axis with 1000 points.

Furthermore, the aerodynamic damping and stiffness matrices are computed, as given in Equation (2.40), for the refined frequency axis. However, only the diagonal terms are included. The expressions for the aerodynamic derivatives, P_1^* , H_1^* , A_2^* , P_4^* , H_4^* and A_3^* , are found from the wind tunnel tests, as explained in Section 3.5.

Next, the modal aerodynamic damping and stiffness for each mode are calculated, from Equation (2.52). Thus, the modal frequency response function, given in Equation (2.56), can be calculated for each mode. The auto-spectra of the horizontal, vertical and torsional acceleration response for each mode are then calculated, through Equations (2.57), (2.61) and (2.62). Finally, the total variances of the horizontal, vertical and torsional acceleration response from all modes are found from Equation (2.63), and the root mean squares are calculated.

Other constants needed for the calculation of the response are given in Table 3.3.

Table 3.3: Constants needed for the response calculations: air density (ρ), width and height of the bridge cross section (B , D) and approximate height of anemometer A5 (z).

Constant	ρ [kg/m ³]	B [m]	D [m]	z [m]
Value	1.25	18.6	3	54.2

Chapter 4

Results

4.1 Modal Analysis

In this section, the modal parameters of the Hålogaland Bridge, found from the Abaqus model and the covariance-driven stochastic subspace identification (Cov-SSI), are presented. Only modes with natural frequencies below 1 Hz are considered from Cov-SSI. The corresponding modes are also found from the Abaqus model. The presented mode shapes from both Abaqus and Cov-SSI represent the mode shapes of the bridge girder, and does not include the bridge towers, the cables or the hangers.

Only mode shapes that can be correctly identified by the monitoring system are presented in Section 4.1.2, and the corresponding mode shapes from the Abaqus model are presented in Section 4.1.1. However, the natural frequencies and damping ratios from all the modes that are identified through Cov-SSI are presented in Section 4.1.3, along with the natural frequencies from the corresponding modes from Abaqus.

4.1.1 Results from Abaqus

The horizontal, vertical and torsional mode shapes and natural frequencies from Abaqus are shown in Figures 4.1, 4.2 and 4.3, respectively.

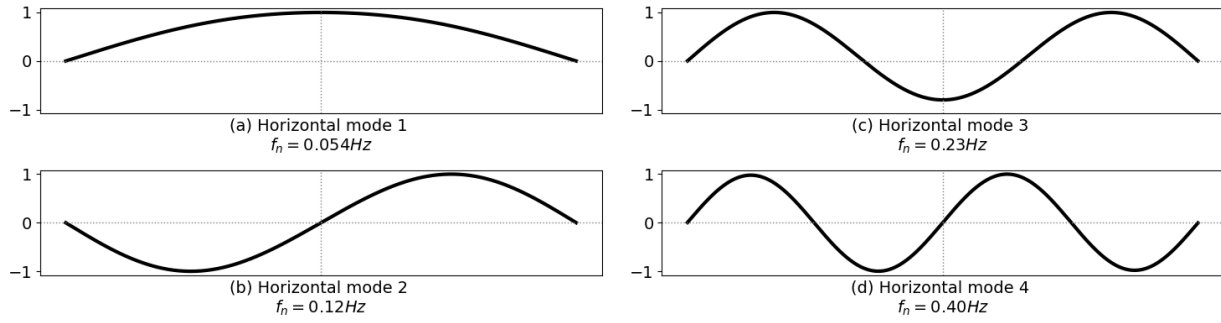


Figure 4.1: Mode shapes and corresponding natural frequencies of horizontal modes from Abaqus.

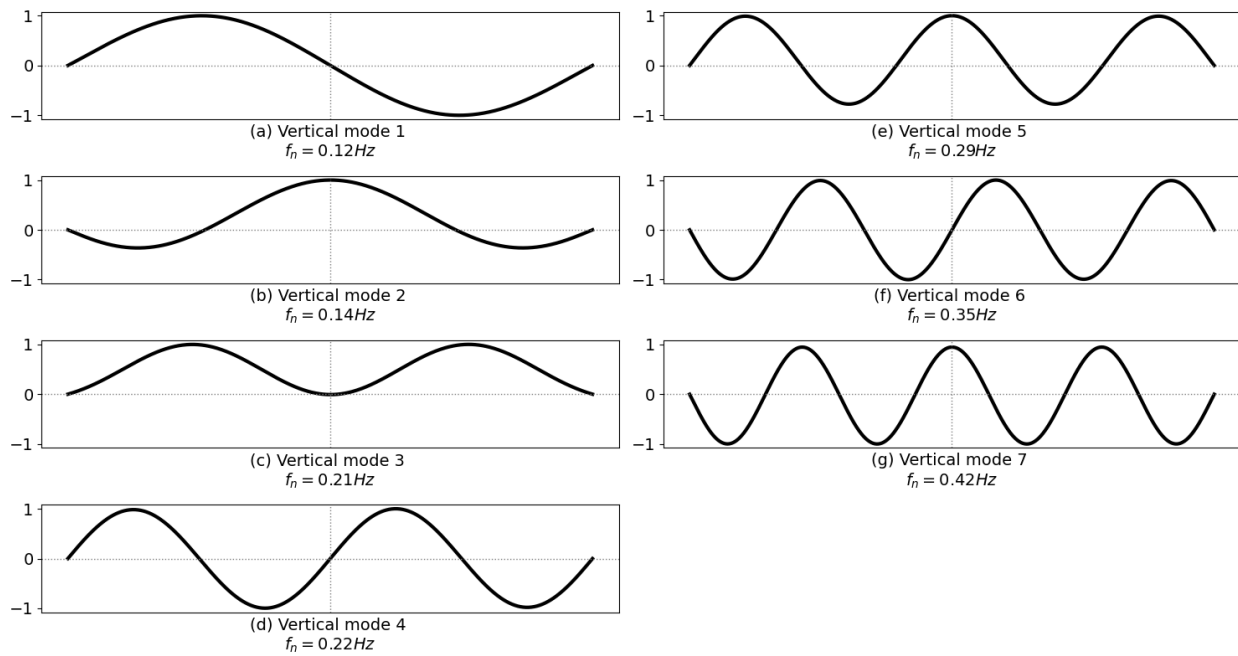


Figure 4.2: Mode shapes and corresponding natural frequencies of vertical modes from Abaqus.

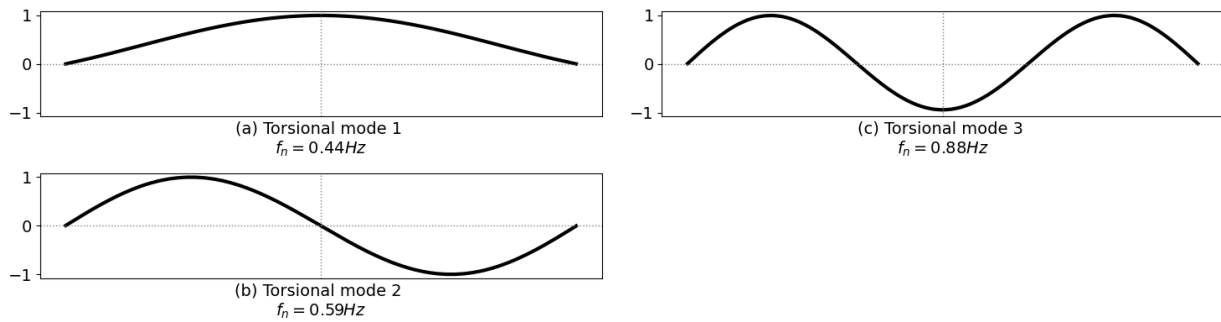


Figure 4.3: Mode shapes and corresponding natural frequencies of torsional modes from Abaqus.

4.1.2 Results from Cov-SSI

The stabilisation plot that is utilised to determine the physical modes from Cov-SSI is presented in Figure 4.4, along with the auto-spectra of the horizontal, vertical and torsional acceleration response at the midspan of the bridge. In the stabilisation plot, the poles that are assumed to originate from physical modes are marked with a grey, dotted line.

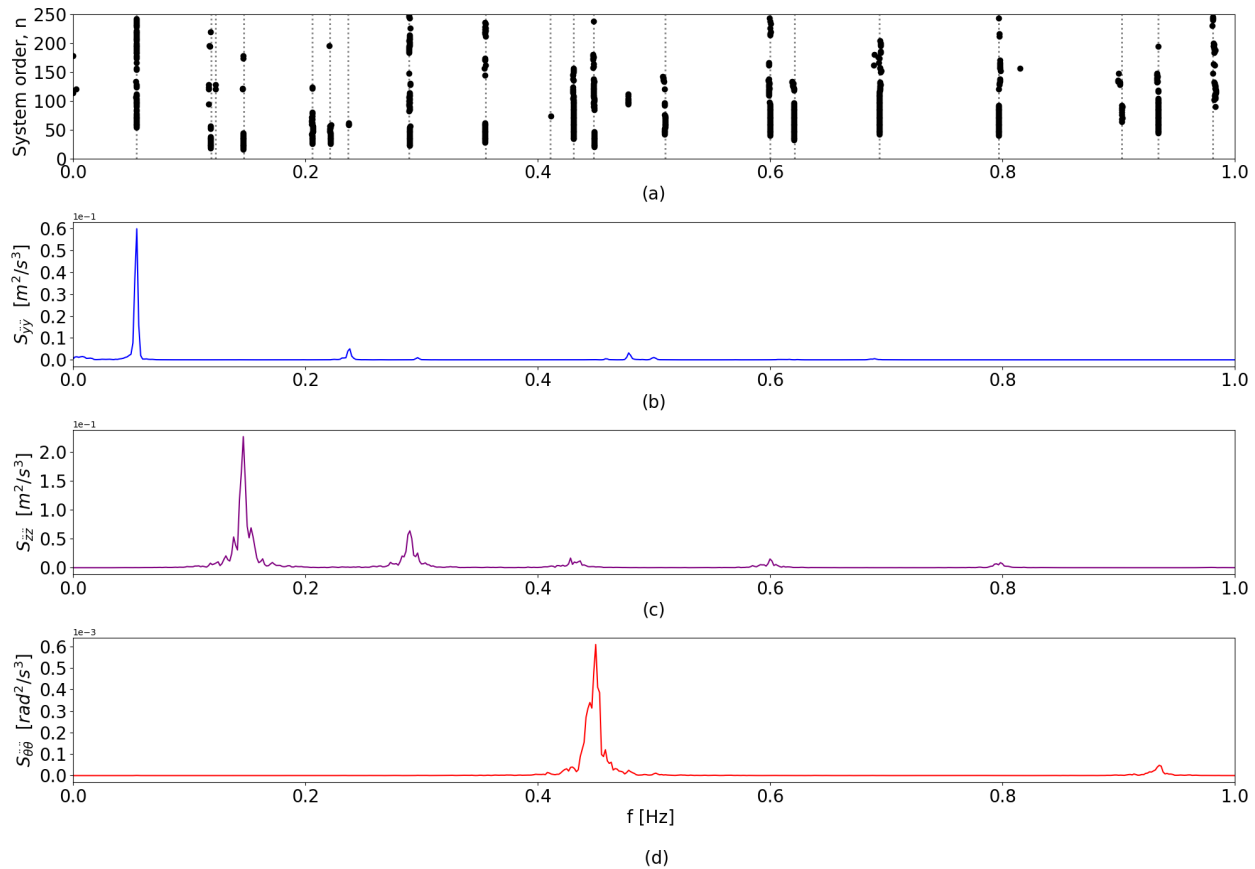


Figure 4.4: Results from the acceleration recording that was chosen as the basis for Cov-SSI. (a) Stabilisation plot. Poles assumed to be from physical modes are marked with a grey, dotted line. (b),(c),(d) Auto-spectra of the acceleration response at the midspan, for the horizontal, vertical and torsional component, respectively.

The estimated horizontal, vertical and torsional mode shapes, natural frequencies and damping ratios found from Cov-SSI are shown in Figures 4.5, 4.6 and 4.7, respectively. The mode shapes are assumed to have zero values at the bridge tower locations.

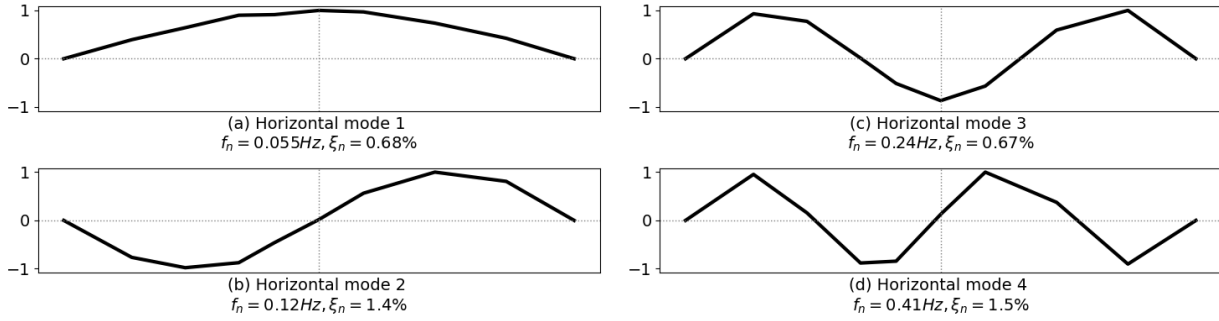


Figure 4.5: Estimated mode shapes and corresponding natural frequencies and damping ratios of horizontal modes from Cov-SSI.

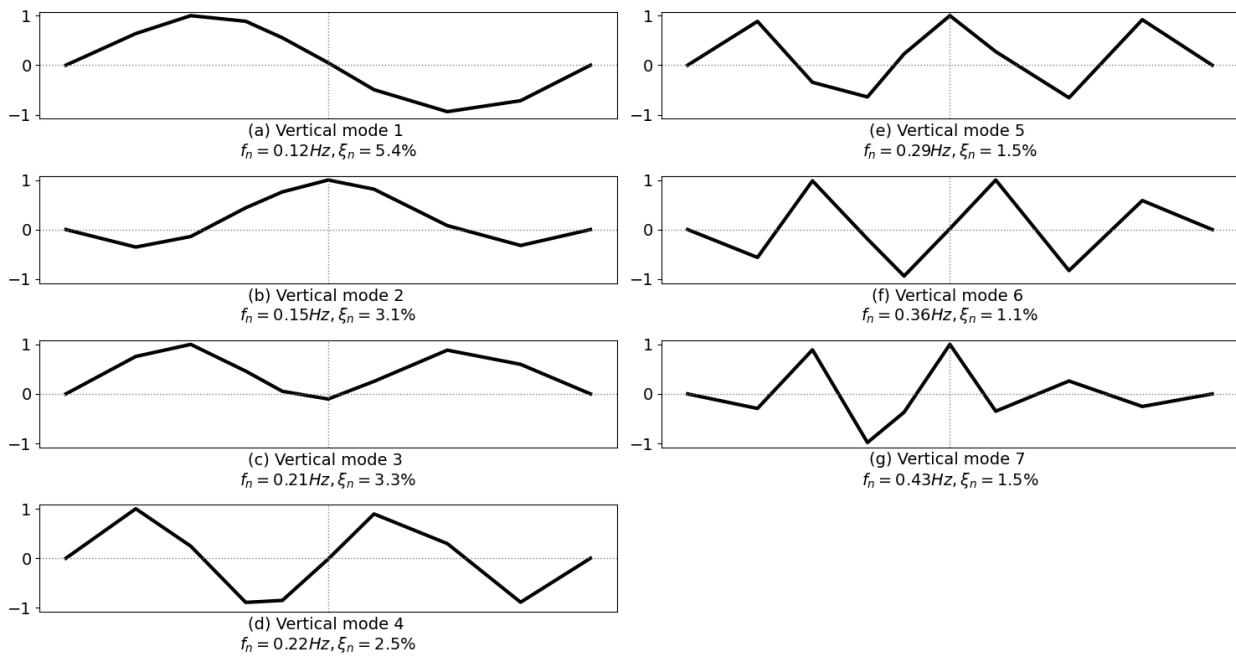


Figure 4.6: Estimated mode shapes and corresponding natural frequencies and damping ratios of vertical modes from Cov-SSI.

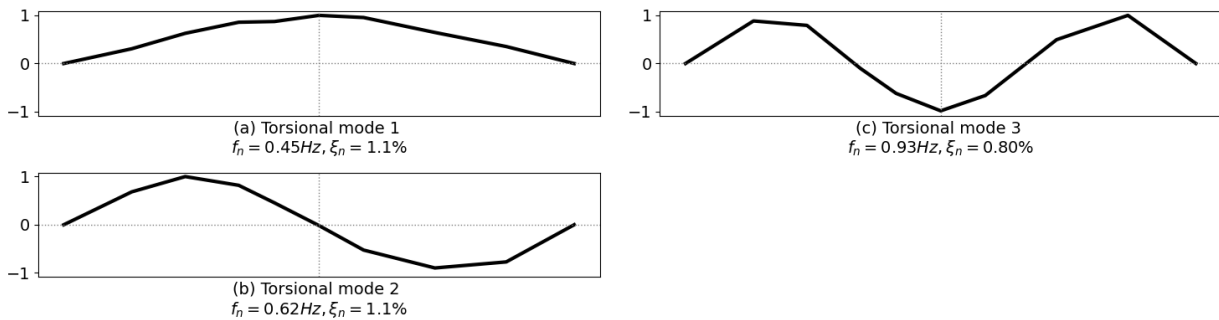


Figure 4.7: Estimated mode shapes and corresponding natural frequencies and damping ratios of torsional modes from Cov-SSI.

4.1.3 Comparison of Modal Parameters from Abaqus and Cov-SSI

The natural frequencies and damping ratios of all the horizontal, vertical and torsional modes that are identified with Cov-SSI are shown in Tables 4.1, 4.2 and 4.3, respectively. The natural frequencies of the corresponding modes from Abaqus are shown in the same tables. Two of the horizontal modes from Abaqus are not identified with Cov-SSI. For comparison, the relative differences between the natural frequencies found with Cov-SSI and Abaqus are also given in these tables. It should be noted that the relative differences are calculated by including three significant figures for the values of the natural frequencies, as opposed to the values given in the tables. Furthermore, the natural frequencies from the two methods are plotted against each other in Figure 4.8.

Table 4.1: Modal parameters for the horizontal modes: natural frequencies from Abaqus (f_{Abaqus}) and Cov-SSI (f_{OMA}), relative differences between the natural frequencies from the two methods (Δf_n) and damping ratios from Cov-SSI (ξ_{OMA}).

Mode number	f_{Abaqus} [Hz]	f_{OMA} [Hz]	$\Delta f_n = \frac{f_{OMA} - f_{Abaqus}}{f_{Abaqus}} \cdot 100$ [%]	ξ_{OMA} [%]
1	0.054	0.055	1.1	0.68
2	0.12	0.12	2.5	1.4
3	0.23	0.24	3.9	0.67
4	0.40	0.41	3.5	1.5
5	0.59	-	-	-
6	0.89	-	-	-

Table 4.2: Modal parameters for the vertical modes: natural frequencies from Abaqus (f_{Abaqus}) and Cov-SSI (f_{OMA}), relative differences between the natural frequencies from the two methods (Δf_n) and damping ratios from Cov-SSI (ξ_{OMA}).

Mode number	f_{Abaqus} [Hz]	f_{OMA} [Hz]	$\Delta f_n = \frac{f_{OMA} - f_{Abaqus}}{f_{Abaqus}} \cdot 100$ [%]	ξ_{OMA} [%]
1	0.12	0.12	2.6	5.4
2	0.14	0.15	2.1	3.1
3	0.21	0.21	0.0	3.3
4	0.22	0.22	1.4	2.5
5	0.29	0.29	1.0	1.5
6	0.35	0.36	2.0	1.1
7	0.42	0.43	1.7	1.5
8	0.50	0.51	2.2	1.3
9	0.59	0.60	1.5	1.0
10	0.69	0.69	0.6	0.90
11	0.79	0.80	1.0	0.66
12	0.90	0.90	0.0	1.3
13	1.0	0.98	-3.8	0.52

Table 4.3: Modal parameters for the torsional modes: natural frequencies from Abaqus (f_{Abaqus}) and Cov-SSI (f_{OMA}), relative differences between the natural frequencies from the two methods (Δf_n) and damping ratios from Cov-SSI (ξ_{OMA}).

Mode number	f_{Abaqus} [Hz]	f_{OMA} [Hz]	$\Delta f_n = \frac{f_{OMA} - f_{Abaqus}}{f_{Abaqus}} \cdot 100$ [%]	ξ_{OMA} [%]
1	0.44	0.45	2.3	1.1
2	0.59	0.62	6.2	1.1
3	0.88	0.93	5.8	0.80

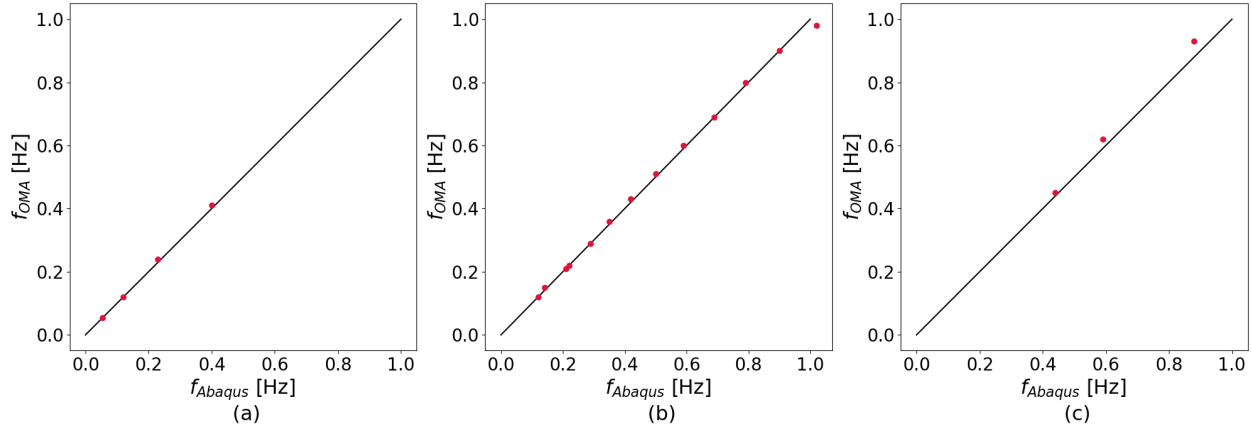


Figure 4.8: Natural frequencies from Cov-SSI plotted against the ones from the Abaqus analysis. A 45° line is also plotted, to illustrate perfect correlation between the natural frequencies. (a),(b),(c) Plots of horizontal, vertical and torsional natural frequencies, respectively.

4.2 Wind Tunnel Test

In this section, the results from the wind tunnel tests are presented. Figure 4.9 shows the plots used to determine the expressions for the ADs as functions of the reduced velocity, \hat{V} . The plots illustrate both the data from the wind tunnel tests and the curve fits to the data. All the fitted curves are 2nd degree polynomials, which are only valid in the range of reduced velocities between 1.35 and 17. Outside this range, the curves are estimated as constant values, corresponding to the start and end values of the curves. Table 4.4 presents the final expressions for the ADs, which are obtained from the fitted curves in Figure 4.9.

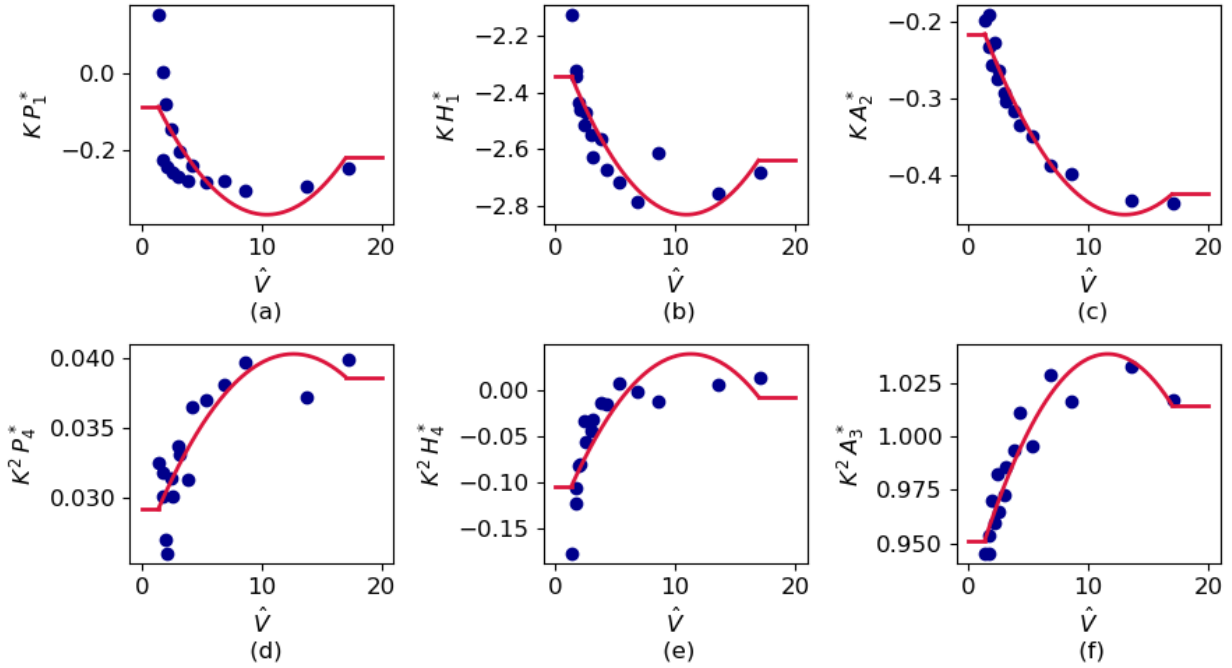


Figure 4.9: Plots used to determine the expressions for the ADs. (a),(b),(c) ADs related to damping, multiplied by reduced frequency, K , against reduced velocity, \hat{V} . (d),(e),(f) ADs related to stiffness, multiplied by K^2 , against \hat{V} . The blue dots illustrate the values from the wind tunnel tests, while the red curves show the curve fits.

Table 4.4: Expressions for the ADs as functions of the reduced velocity, \hat{V} .

AD	Expression		
	$\hat{V} \leq 1.35$	$1.35 < \hat{V} < 17$	$\hat{V} \geq 17$
P_1^*	$-0.088 / K$	$(0.0034\hat{V}^2 - 0.071\hat{V} + 0.0015) / K$	$-0.22 / K$
H_1^*	$-2.3 / K$	$(0.0053\hat{V}^2 - 0.12\hat{V} - 2.2) / K$	$-2.6 / K$
A_2^*	$-0.22 / K$	$(0.0017\hat{V}^2 - 0.045\hat{V} - 0.16) / K$	$-0.43 / K$
P_4^*	$0.029 / K^2$	$(-0.000087\hat{V}^2 + 0.0022\hat{V} + 0.026) / K^2$	$0.039 / K^2$
H_4^*	$-0.10 / K^2$	$(-0.0014\hat{V}^2 + 0.033\hat{V} - 0.15) / K^2$	$-0.0084 / K^2$
A_3^*	$0.95 / K^2$	$(-0.00083\hat{V}^2 + 0.019\hat{V} + 0.93) / K^2$	$1.0 / K^2$

Furthermore, Figure 4.10 shows the plots of the load coefficients, C_D , C_L and C_M , against the angle of flow incidence, α . Similar as for the ADs, the plots illustrate both the data from the wind tunnel tests and the curve fits to the data. The fitted curves are 2^{nd} degree polynomials. Table 4.5 presents the final values for the load coefficients \bar{C}_D , \bar{C}_L , \bar{C}_M , C'_D , C'_L and C'_M , obtained from the fitted curves in Figure 4.10.

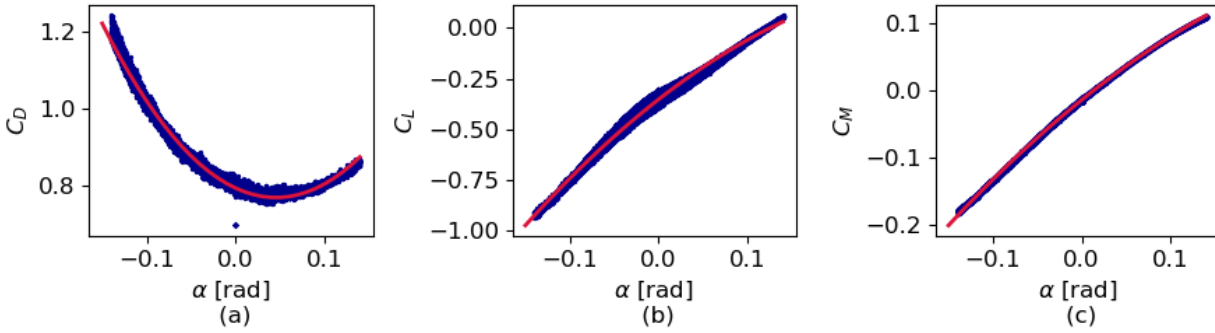


Figure 4.10: Plots of load coefficients against the angle of flow incidence, α . (a) C_D , (b) C_L , (c) C_M . The blue dots illustrate the values from the wind tunnel tests, while the red curves show the curve fits.

Table 4.5: Values for the load coefficients.

\bar{C}_D	\bar{C}_L	\bar{C}_M	C'_D	C'_L	C'_M
0.79	-0.35	-0.015	-1.1	3.4	1.1

4.3 Wind Field

In this section, the main results from the analysis of the wind field at the Hålogaland Bridge are presented. Section 4.3.1 presents wind characteristics such as mean wind speed and turbulence intensities obtained from the measurements. Section 4.3.2 concerns results related to the turbulence spectra and the turbulence parameters. Finally, in Section 4.3.3, results from the probabilistic modelling of the wind field are presented.

4.3.1 Wind Field Characteristics

The mean wind speed from all 10 minute recordings from the Hålogaland Bridge are presented in Figure 4.11. In Figure 4.11(a), the mean wind speed is presented using a wind rose plot, illustrating the relation between the magnitude and the direction of the wind. In addition, Figure 4.11(b) presents a histogram of the mean wind speed, showing the distribution of the recorded mean wind speeds.

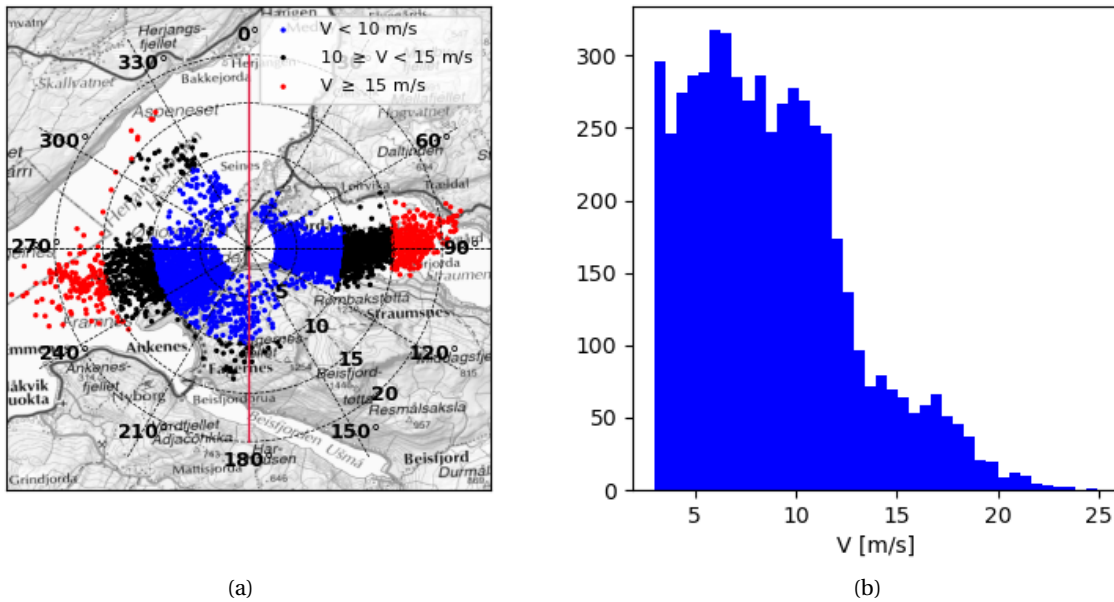


Figure 4.11: Mean wind speeds from all 10 min recordings presented as (a) a wind rose plot and (b) a histogram.

Furthermore, the turbulence intensities are presented in Figure 4.12 using wind rose plots. Additionally, the turbulence intensities are depicted using scatter plots in Figure 4.13, showing how the turbulence intensities vary with the mean wind speed.

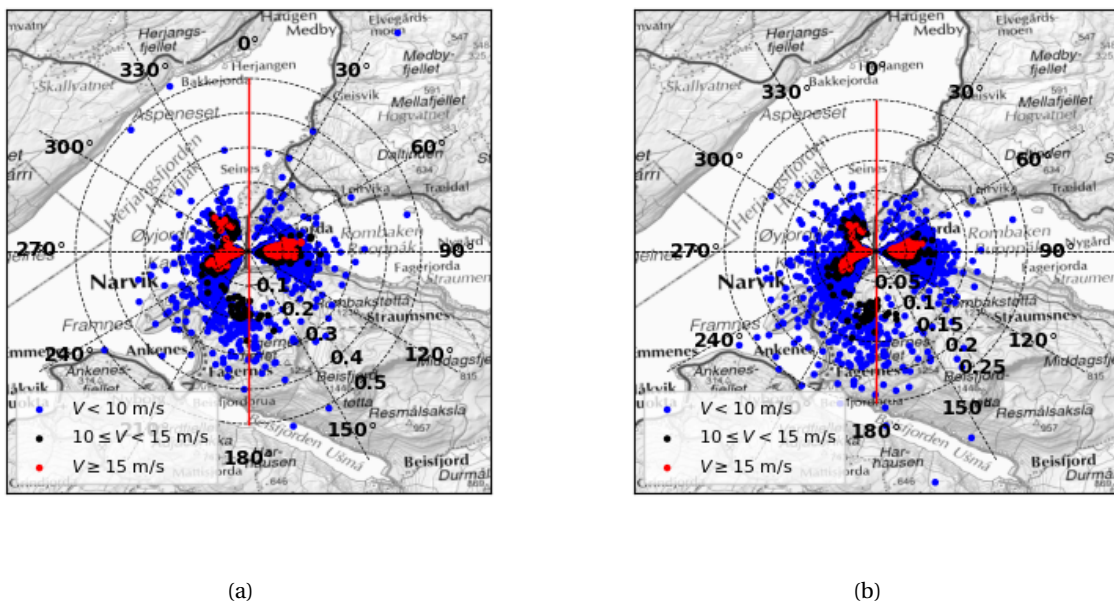


Figure 4.12: Wind rose plots of turbulence intensities: (a) along-wind turbulence intensity and (b) vertical turbulence intensity.

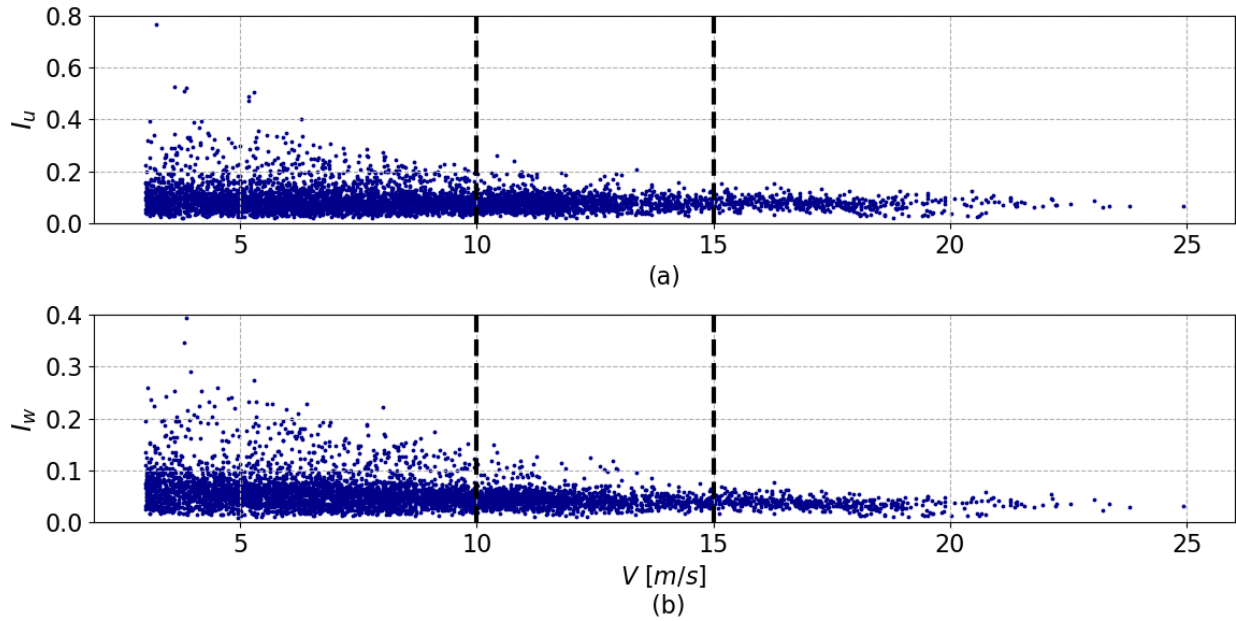


Figure 4.13: Turbulence intensities against mean wind speed: (a) along-wind turbulence intensity and (b) vertical turbulence intensity. The dashed lines illustrate the division of the turbulence intensities into three segments.

Finally, Figure 4.14 presents plots to illustrate how selected wind field characteristics vary along the bridge girder. Only wind data from one single 10 minute recording were utilised for these calculations. The recording was chosen to display the general trend that could be observed from investigation of several recordings.

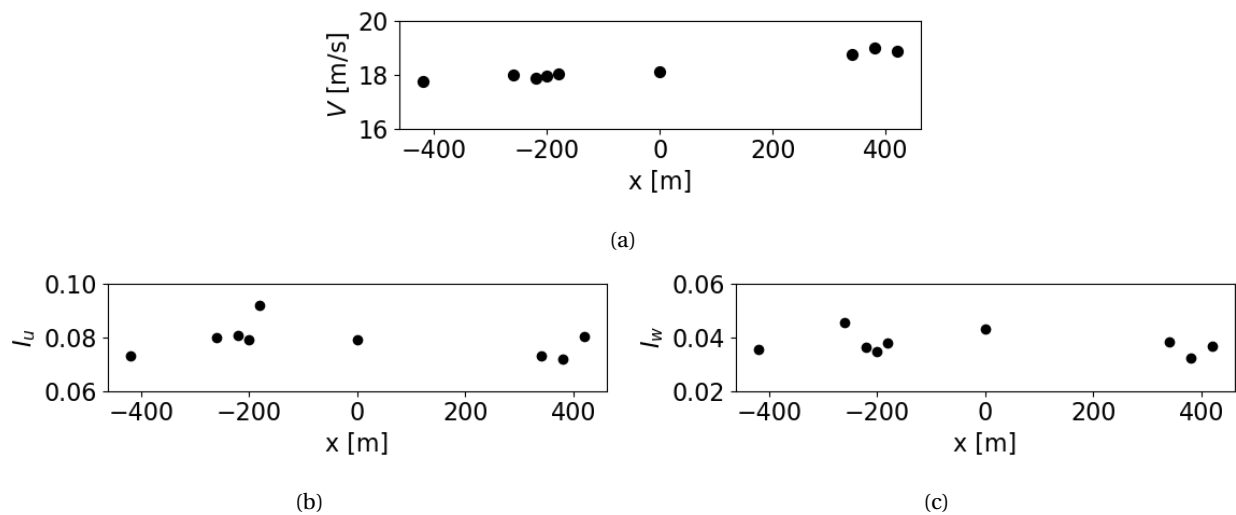


Figure 4.14: Wind characteristics along the bridge girder of the Hålogaland Bridge: (a) mean wind speed, (b) along-wind turbulence intensity and (c) vertical turbulence intensity.

4.3.2 Wind Turbulence Spectra

Figures 4.15 and 4.16 show the estimated auto-spectra and normalised cross-spectra of the turbulence components for two arbitrary recordings. It is chosen to showcase recordings with relatively high mean wind speeds, in order to avoid non-stationary behaviour of the wind.

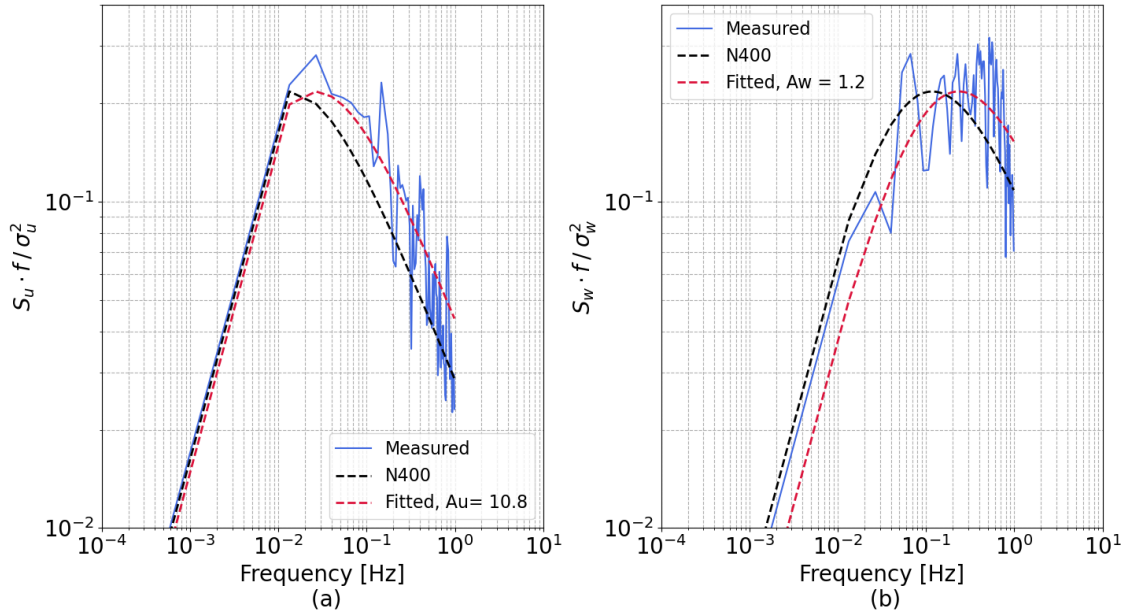


Figure 4.15: Estimated auto-spectra of turbulence components for a 10 minute recording ($V = 15\text{ m/s}$): (a) along-wind turbulence and (b) vertical turbulence.

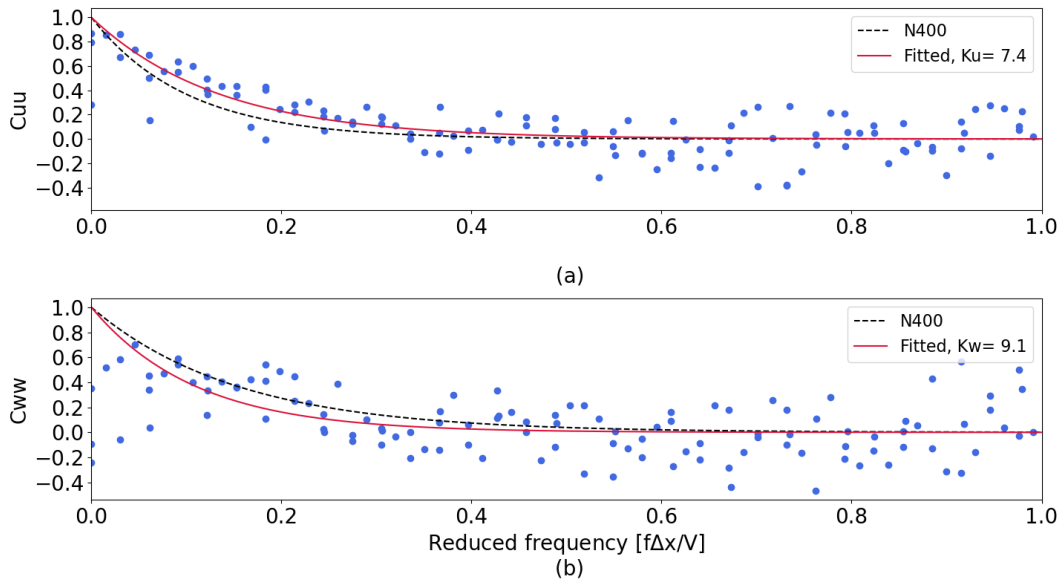


Figure 4.16: Estimated normalised cross-spectra of turbulence components for a 10 minute recording ($V = 17\text{ m/s}$): (a) along-wind turbulence and (b) vertical turbulence. The blue dots illustrate the spectra from the measurements.

Furthermore, the histograms and the fitted probability distributions of the turbulence parameters, obtained as explained in Section 3.6, are presented in Figures 4.17-4.20. As previously explained, only data from 10 minute recordings where the mean wind comes approximately perpendicular to the bridge x-axis are included in these probability distributions. The lognormal distribution parameters, μ and σ , are indicated in the figures.

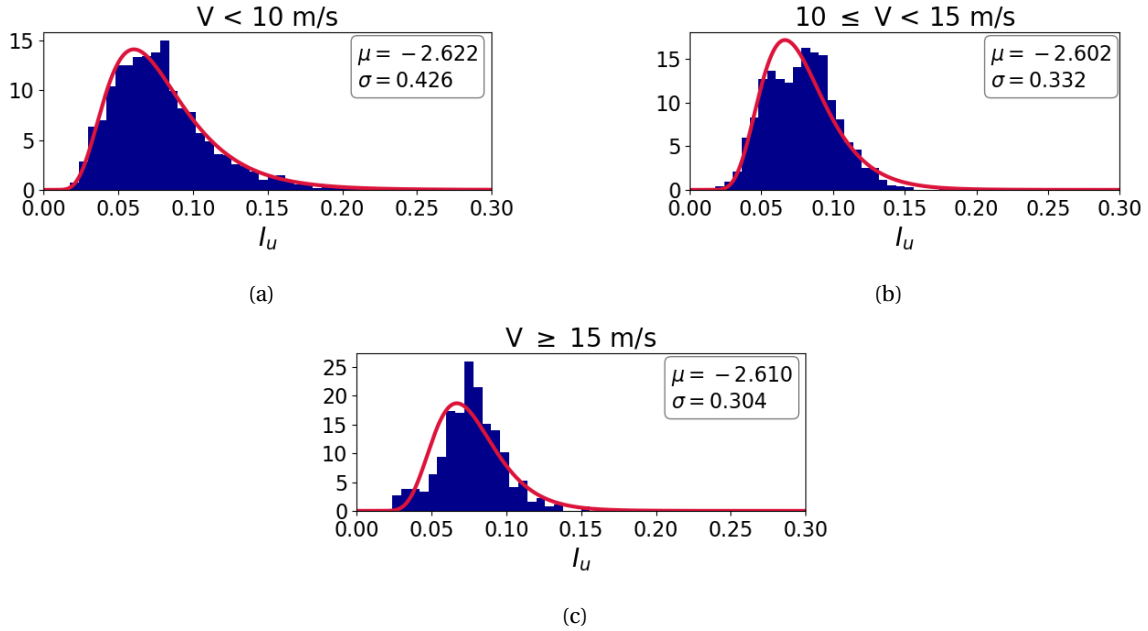


Figure 4.17: Probability distributions of along-wind turbulence intensity.

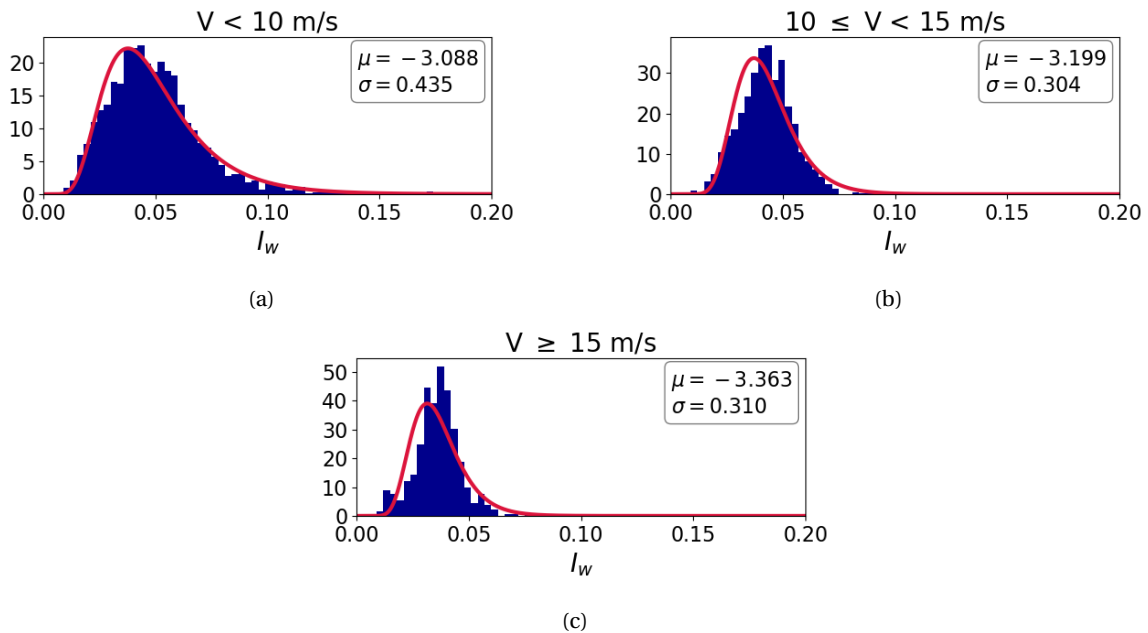


Figure 4.18: Probability distributions of vertical turbulence intensity.

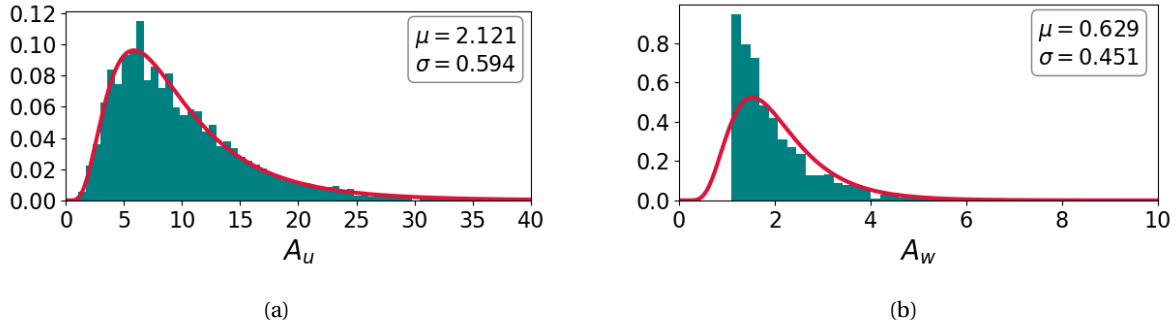


Figure 4.19: Probability distributions of the spectral parameters: (a) u component and (b) w component.

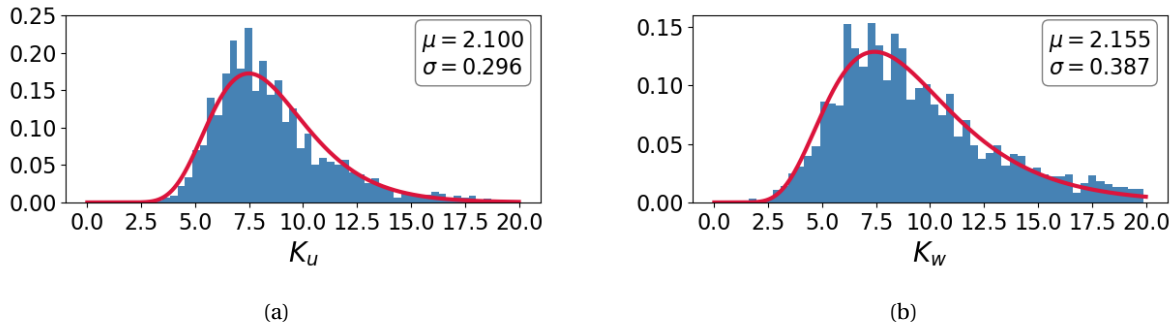


Figure 4.20: Probability distributions of the decay coefficients: (a) u component and (b) w component.

4.3.3 Probabilistic Wind Field Model

In this section, the results from the probabilistic wind field modelling are presented. As previously explained, only the turbulence standard deviations, σ_u and σ_w , are included in the model.

Figure 4.21 shows the scatter plot matrices of the turbulence standard deviations for easterly and the westerly winds. The plots only include data from recordings with mean wind speed larger than 11 m/s, as the probabilistic model only applies to this mean wind speed range. This figure is included to illustrate the probability distributions for the parameters as well as the correlation between them. The histograms for the parameters are shown on the diagonals, along with the fitted lognormal probability distributions. The off-diagonal plots illustrate the correlation between the parameters, along with linear regression fits of the data, to clearly illustrate the correlation.

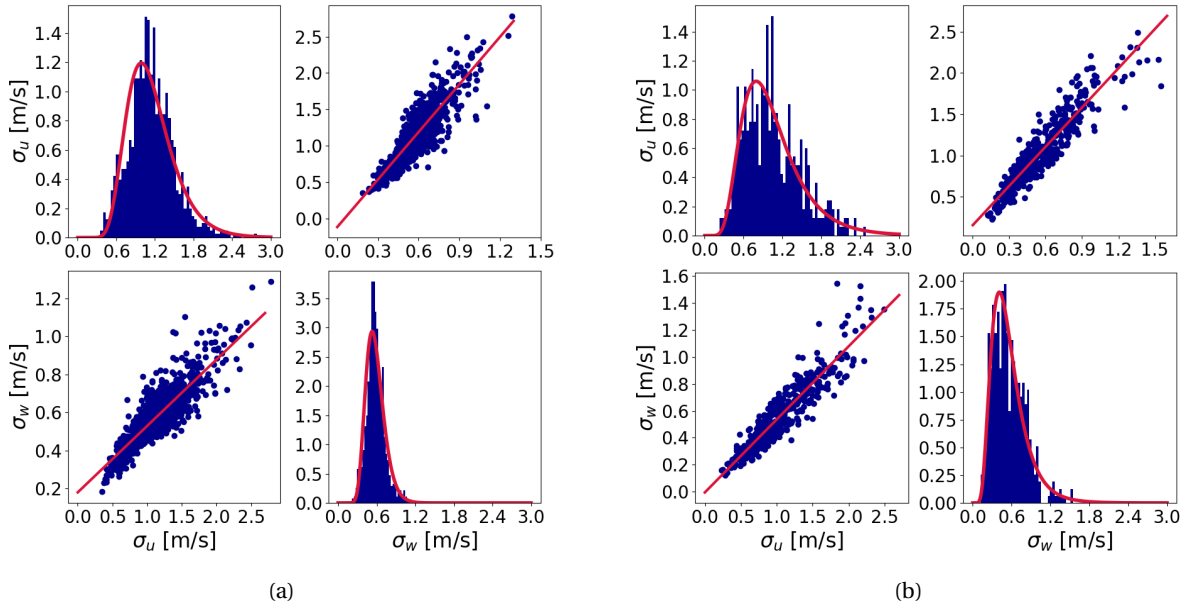


Figure 4.21: Scatter plot matrices of the turbulence standard deviations: (a) easterly winds and (b) westerly winds. The plots on the diagonals illustrate the histograms of $\sigma_{u,w}$, while the off-diagonal plots illustrate the correlation between σ_u and σ_w .

Furthermore, Figures 4.22 and 4.23 show the turbulence standard deviations plotted against the mean wind speed, for the easterly and westerly winds, respectively. The linear regression fits of the data are also shown in the figures, to clearly illustrate the linear dependence between the turbulence standard deviations and the mean wind speed.

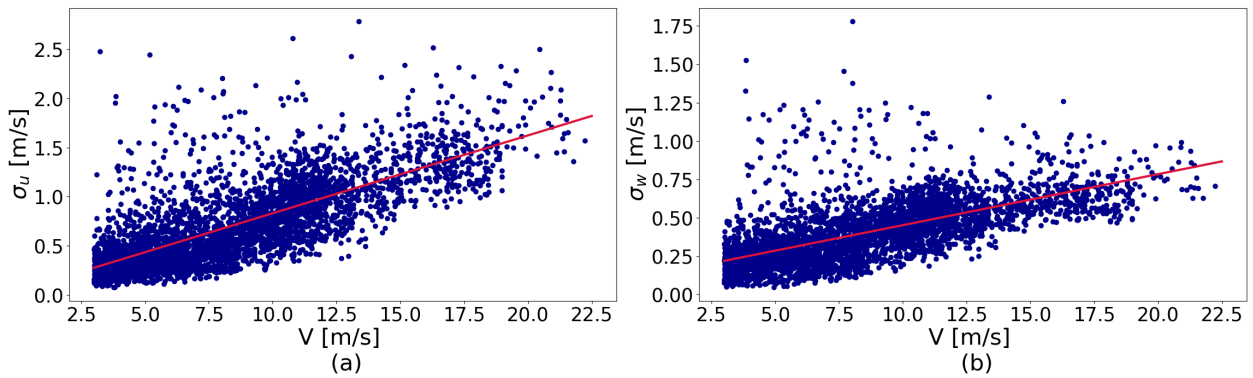


Figure 4.22: Turbulence standard deviations against mean wind speed for easterly winds: (a) σ_u and (b) σ_w . The straight lines illustrate the linear regression fits of the data.

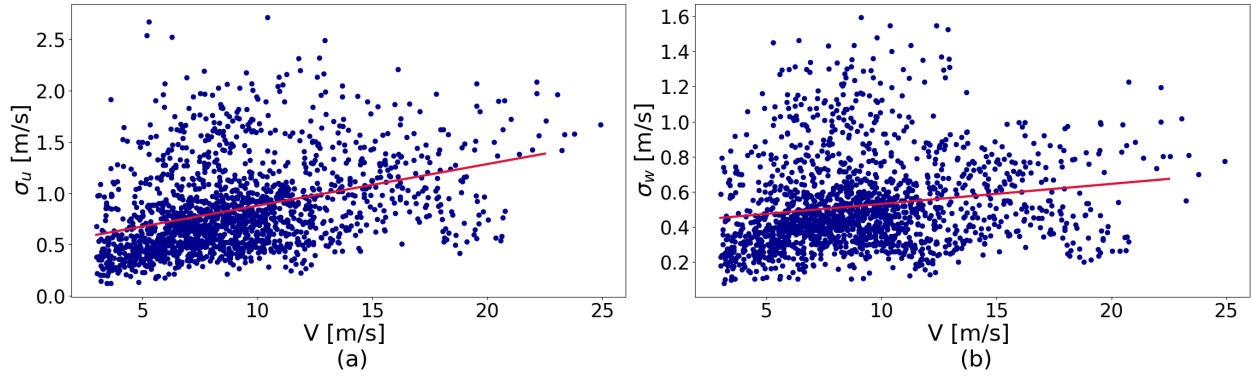


Figure 4.23: Turbulence standard deviations against mean wind speed for westerly winds: (a) σ_u and (b) σ_w . The straight lines illustrate the linear regression fits of the data.

Figures 4.24 and 4.25 show the plots of the lognormal distribution parameters of the turbulence standard deviations against the mean wind speed for easterly and westerly winds, respectively. Figure 4.26 shows the plot of the correlation coefficients of the turbulence standard deviations against mean wind speed, for both easterly and westerly winds. The plots illustrate the parameters calculated for each mean wind speed interval, in addition to the curves that are used for the probabilistic model of the parameters. As can be seen from the figures, the $\tilde{\mu}$ -parameters are modelled as linear curves, while $\tilde{\sigma}$ and $\rho_{\sigma_u\sigma_w}$ are modelled as constant values, as explained in Section 3.7.

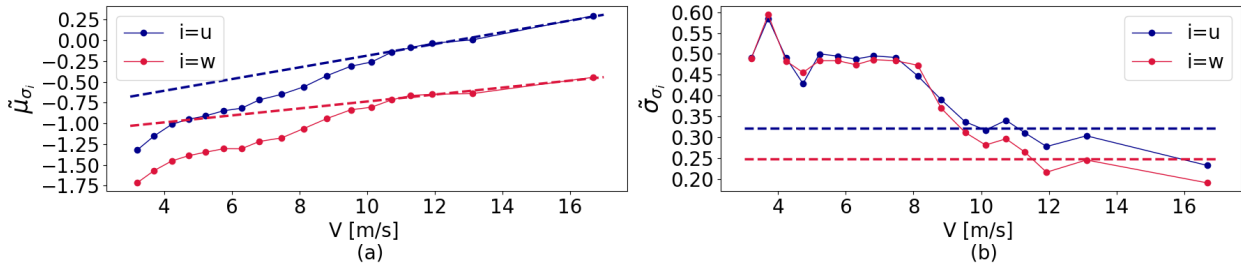


Figure 4.24: Lognormal distribution parameters against mean wind speed, for the turbulence standard deviations of easterly winds: (a) $\tilde{\mu}$ and (b) $\tilde{\sigma}$. The dotted curves show the calculated parameters for each mean wind speed interval, while the dashed lines show the curves used for the probabilistic model of $\sigma_{u,w}$.

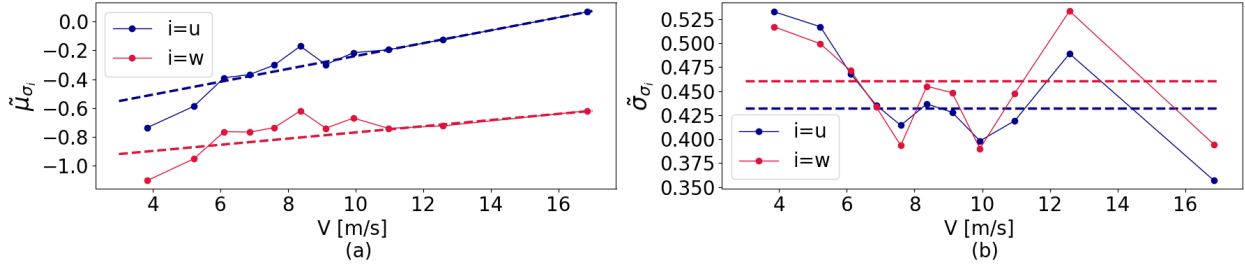


Figure 4.25: Lognormal distribution parameters against mean wind speed, for the turbulence standard deviations of westerly winds: (a) $\tilde{\mu}$ and (b) $\tilde{\sigma}$. The dotted curves show the calculated parameters for each mean wind speed interval, while the dashed lines show the curves used for the probabilistic model of $\sigma_{u,w}$.

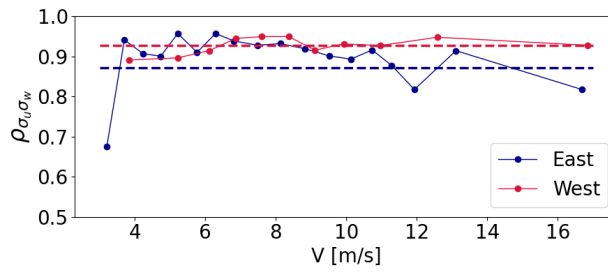


Figure 4.26: Correlation coefficients of the turbulence standard deviations against mean wind speed, for both east-erly and westerly winds. The dotted curves show the calculated correlation coefficients for each mean wind speed interval, while the dashed lines show the curves used for the probabilistic model of $\sigma_{u,w}$.

Furthermore, the final expressions for the lognormal distribution parameters and correlation coefficients that are utilised to simulate the turbulence standard deviations are presented in Tables 4.6 and 4.7, respectively. The $\tilde{\mu}$ -parameters are functions of the mean wind speed, V , while $\tilde{\sigma}$ and $\rho_{\sigma_u\sigma_w}$ are constants.

Table 4.6: Lognormal distribution parameters for the probabilistic model of the turbulence standard deviations.

		σ_u	σ_w
East	$\tilde{\mu}$	$-0.889 + 0.0705V$	$-1.155 + 0.0419V$
	$\tilde{\sigma}$	0.3211	0.2481
West	$\tilde{\mu}$	$-0.685 + 0.0446V$	$-0.982 + 0.0214V$
	$\tilde{\sigma}$	0.4324	0.4604

Table 4.7: Correlation coefficient matrix for the probabilistic model of the turbulence standard deviations.

		σ_u	σ_w
East	σ_u	1	0.8713
	σ_w	0.8713	1
West	σ_u	1	0.9268
	σ_w	0.9268	1

In order to illustrate how well the probabilistic model of the turbulence standard deviations is able to reproduce the measured turbulence standard deviations, several plots of the measured and simulated parameters are presented. Figures 4.27 and 4.28 show scatter plots of the measured and simulated turbulence standard deviations against the mean wind speed, for easterly and westerly winds, respectively.

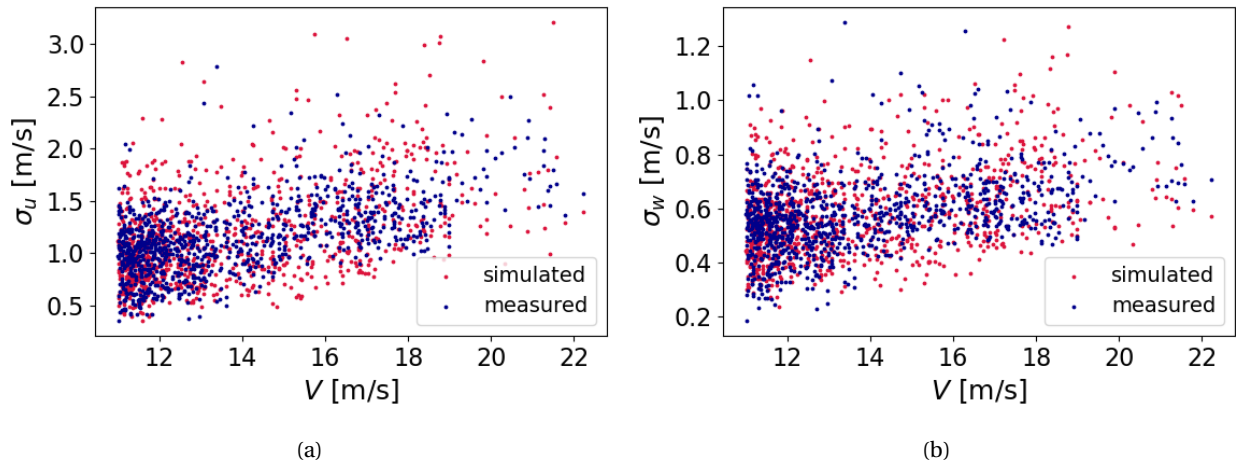


Figure 4.27: Scatter plots of measured and simulated turbulence standard deviations against mean wind speed, for easterly winds: (a) σ_u and (b) σ_w .

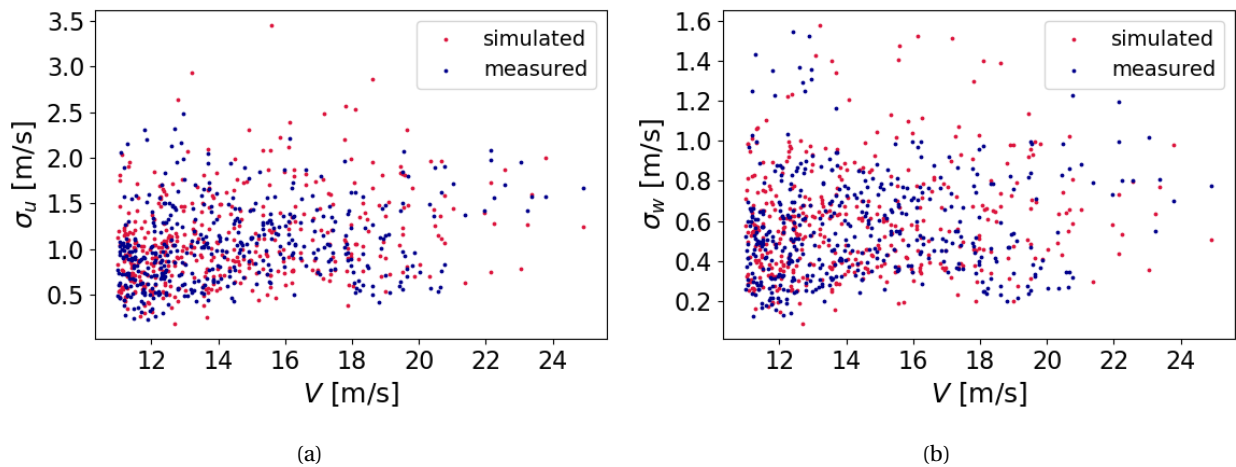


Figure 4.28: Scatter plots of measured and simulated turbulence standard deviations against mean wind speed, for westerly winds: (a) σ_u and (b) σ_w .

Furthermore, Figure 4.29 shows scatter plots of vertical turbulence standard deviation plotted against along-wind turbulence standard deviation, from both measurements and simulations.

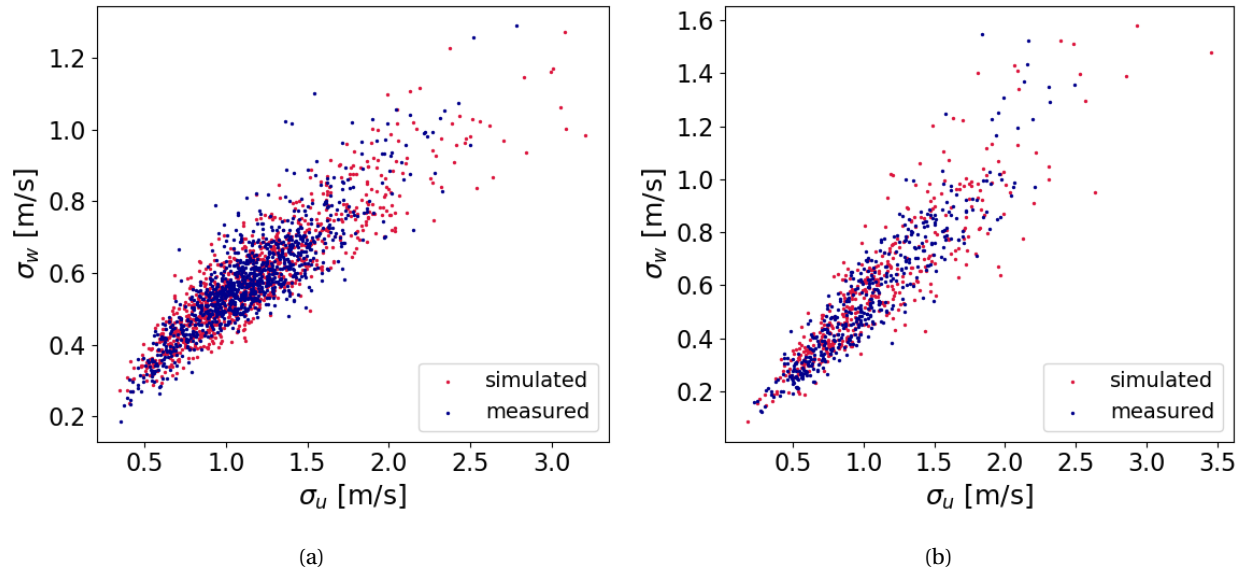


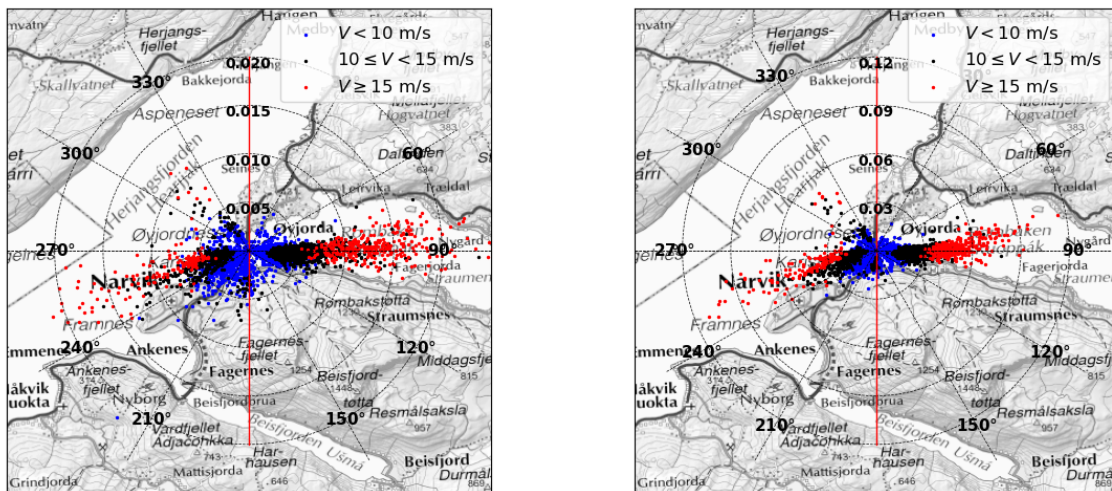
Figure 4.29: Scatter plots of σ_w against σ_u , from both measurements and simulations: (a) easterly winds and (b) westerly winds.

4.4 Response of the Bridge

In this section, the root mean square (RMS) acceleration response of the Hålogaland Bridge is illustrated in different ways. Section 4.4.1 presents plots illustrating how the wind field is influencing the measured response. Furthermore, in Section 4.4.2, the results from the buffeting response predictions are presented together with the measured response.

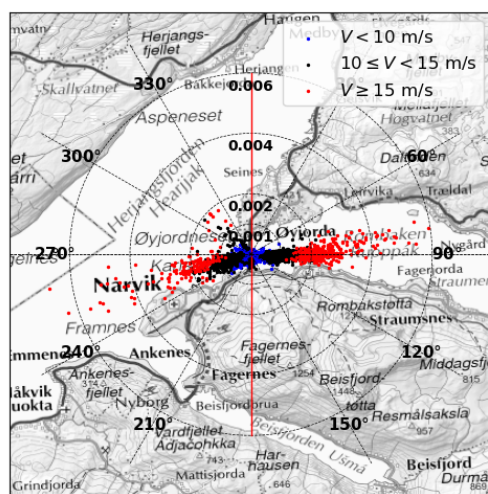
4.4.1 Measured Response

The measured RMS acceleration response is presented in Figure 4.30 using wind rose plots, showcasing the relation between the magnitude of the response and the mean wind direction.



(a)

(b)



(c)

Figure 4.30: Wind rose plots of RMS acceleration response from measurements: (a) lateral component [m/s^2], (b) vertical component [m/s^2] and (c) torsional component [rad/s^2].

Additionally, in Figure 4.31 the measured RMS acceleration response is plotted against the mean wind speed, colour coded according to the turbulence intensity.

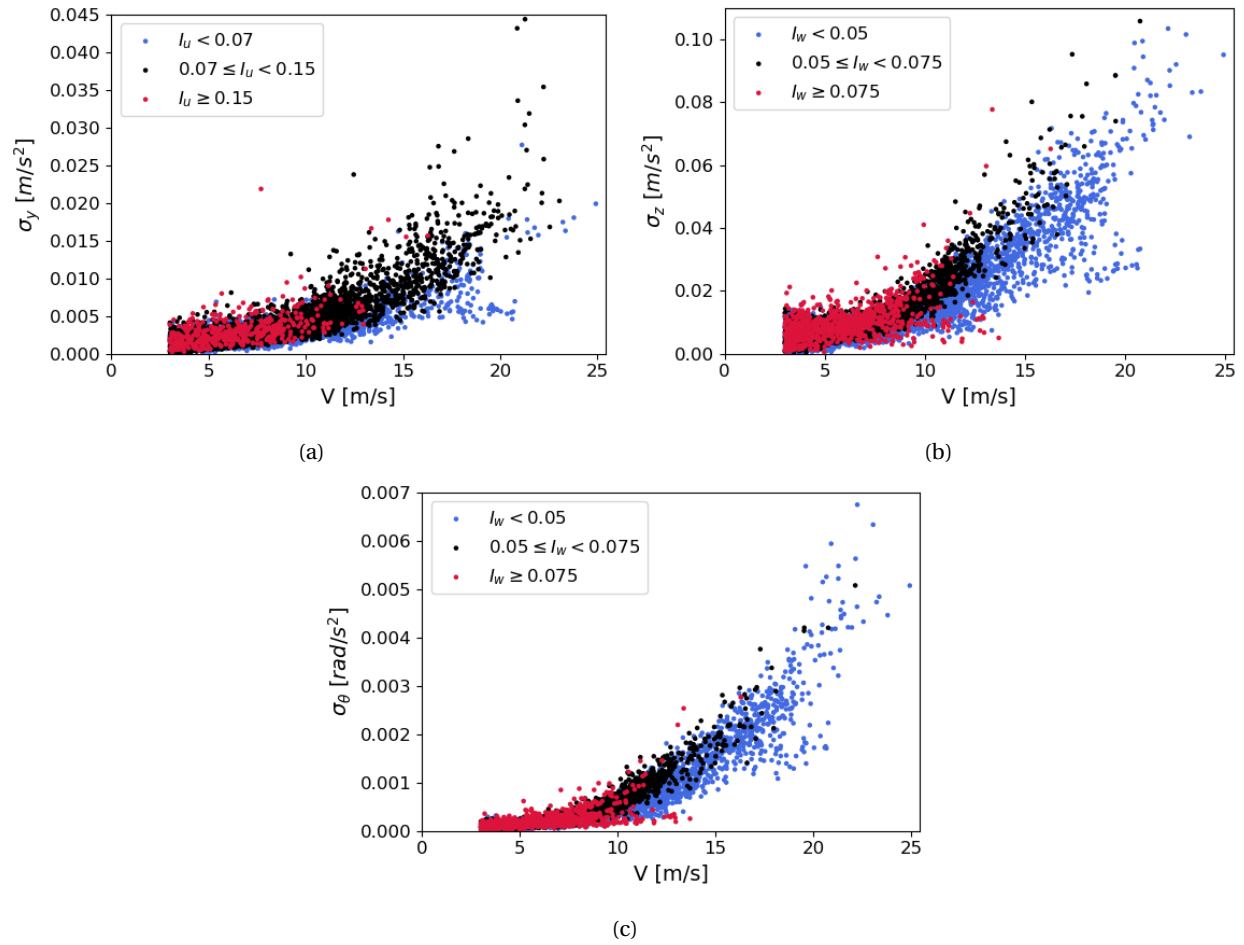


Figure 4.31: RMS acceleration response from measurements against mean wind speed: (a) lateral component, (b) vertical component and (c) torsional component. Colour coding according to turbulence intensity.

Lastly, in Figure 4.32 the measured RMS acceleration response is plotted against the mean wind speed, colour coded according to the mean wind direction.

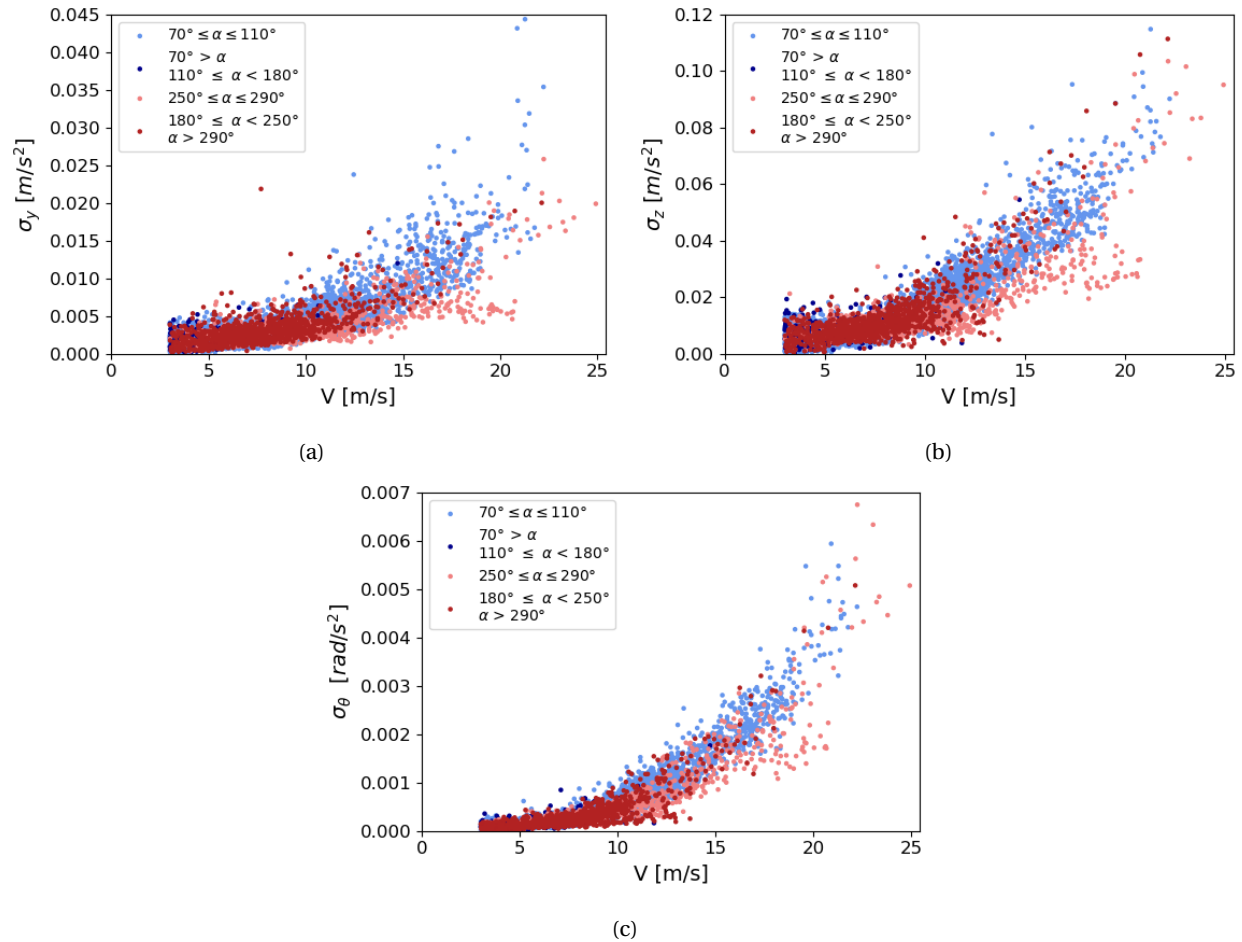


Figure 4.32: RMS acceleration response from measurements against mean wind speed: (a) lateral component, (b) vertical component and (c) torsional component. Colour coding according to mean wind direction.

4.4.2 Comparison of Measured and Predicted Response

In this section, plots of the RMS acceleration response, from both predictions and measurements, against the mean wind speed are presented. The predicted buffeting response is calculated for the five cases presented in Section 3.8. The values of the turbulence parameters, $A_{u,w}$, $K_{u,w}$ and $I_{u,w}$, corresponding to each case are presented in Table 4.8. Figure 4.33 displays the predicted buffeting response for case 1-4. As mentioned, the turbulence parameters used to calculate the predicted response for these cases are obtained by only considering perpendicular wind. Therefore, only measured response from wind deviating less than 20 degrees from the two perpendicular directions (90 and 270 degrees) are plotted in Figure 4.33.

Table 4.8: Turbulence parameters for the auto-spectra and the normalised cross-spectra of turbulence, used to calculate the predicted buffeting response.

Case	A_u	A_w	K_u	K_w	I_u^{**}			I_w^{**}		
					1	2	3	1	2	3
Case 1	8.3	1.9	8.2	8.6	0.0726	0.0741	0.0735	0.0456	0.0408	0.0346
Case 2	8.3	1.9	8.2	8.6	0.146	0.128	0.121	0.0932	0.0673	0.0576
Case 3	8.3	1.9	5	4.6	0.0726	0.0741	0.0735	0.0456	0.0408	0.0346
Case 4	3.1	0.89	8.2	8.6	0.0726	0.0741	0.0735	0.0456	0.0408	0.0346
Case 5*	E: 13.9 W: 9.4	E: 1.7 W: 1.8	7.8 7.8	8.9 8.9	S	S	S	S	S	S

*E: value for easterly winds, W: value for westerly winds, S: value calculated from simulated $\sigma_{u,w}$.

** 1: $V < 10$ m/s, 2: $10 \leq V < 15$ m/s, 3: $V \geq 15$ m/s

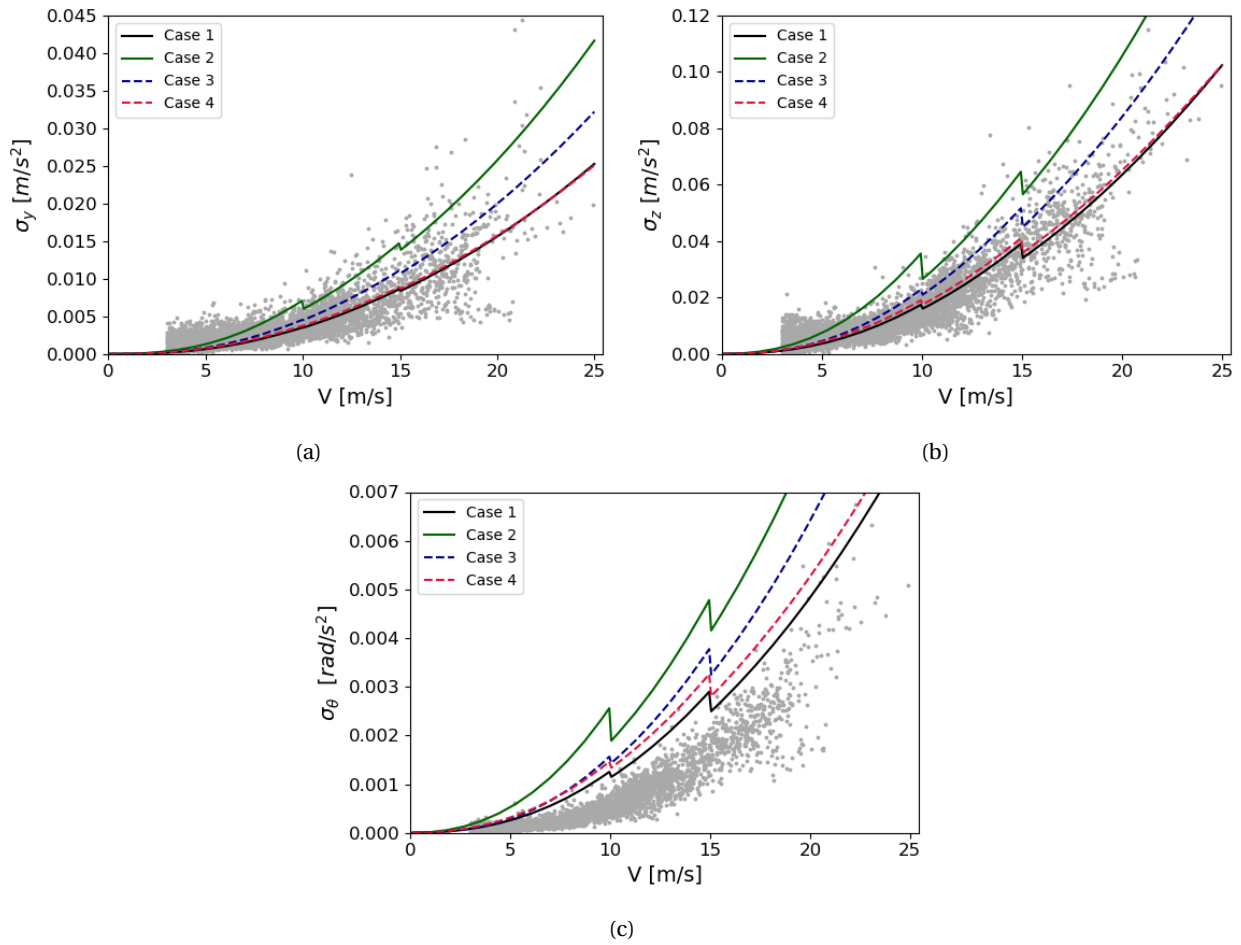


Figure 4.33: RMS acceleration response from predictions and measurements against mean wind speed: (a) lateral component, (b) vertical component and (c) torsional component. The grey dots illustrate the measured response, while the continuous lines illustrate the different cases of predicted response.

Furthermore, Figures 4.34 and 4.35 display the predicted buffeting response for case 5, for easterly and westerly winds, respectively. The response is only plotted for mean wind speeds above 11 m/s, as the probabilistic wind field model only applies to this mean wind speed range.

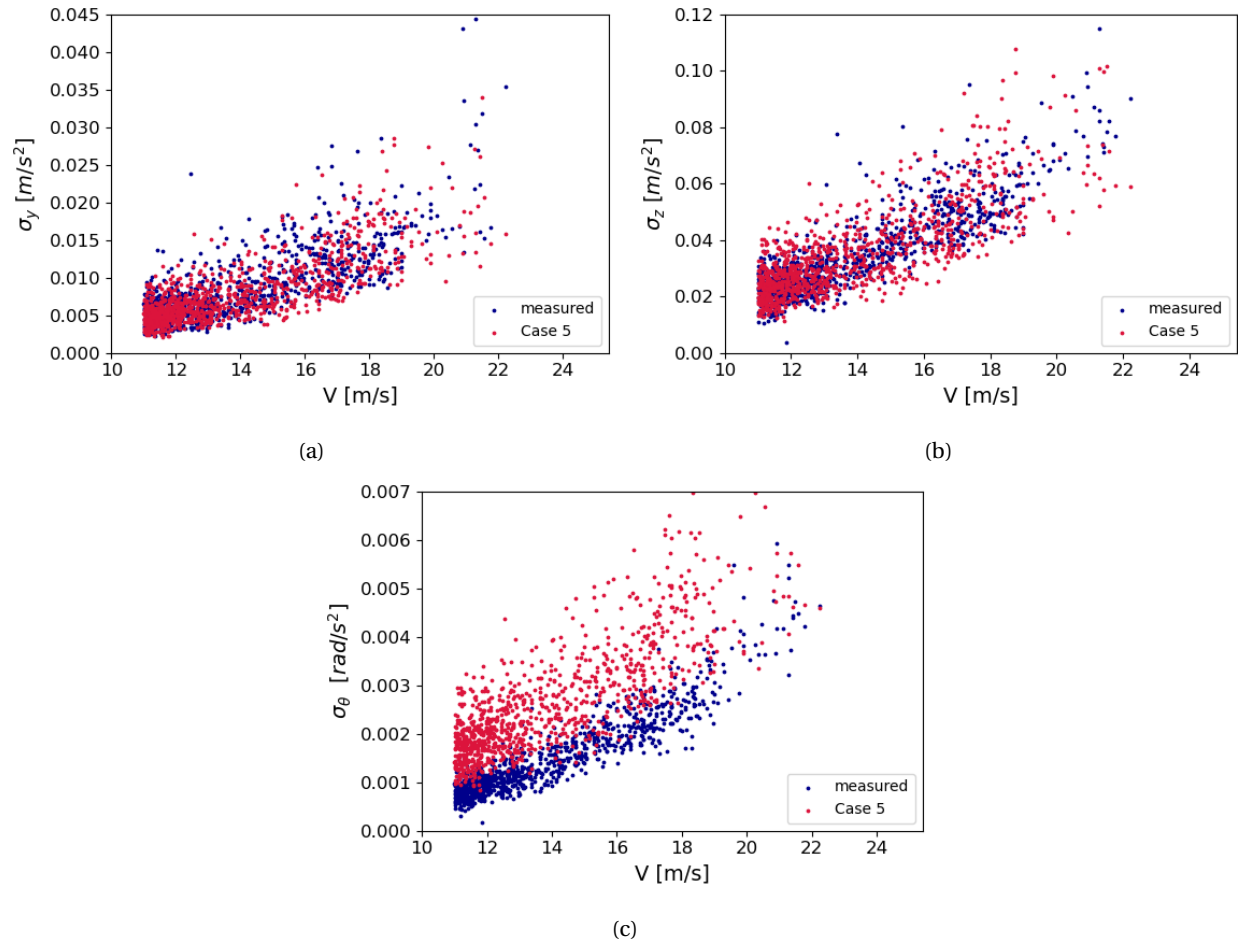


Figure 4.34: Scatter plots of RMS acceleration response from predictions and measurements against mean wind speed for easterly winds: (a) lateral component, (b) vertical component and (c) torsional component.

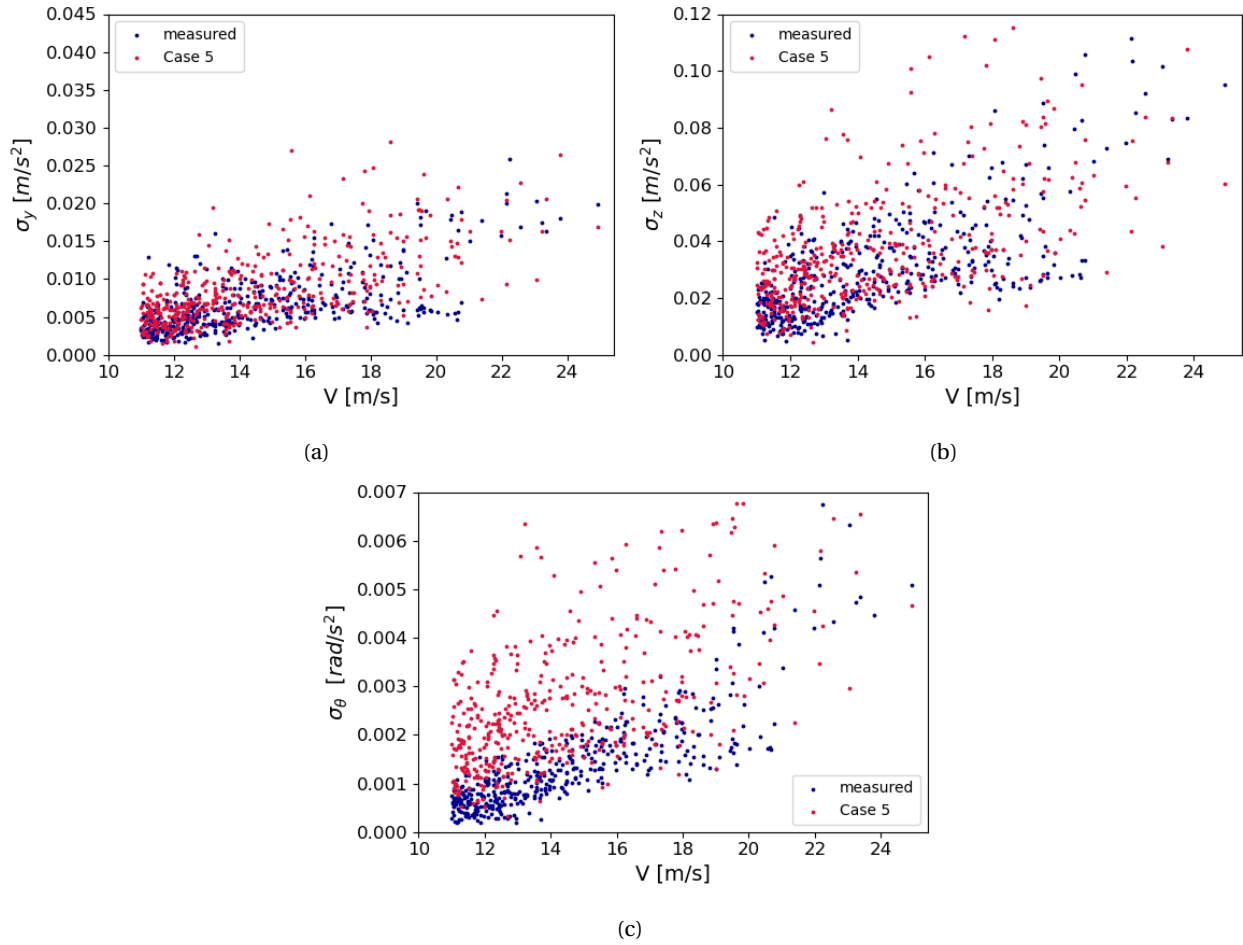


Figure 4.35: Scatter plots of RMS acceleration response from predictions and measurements against mean wind speed for westerly winds: (a) lateral component, (b) vertical component and (c) torsional component.

Chapter 5

Discussion

5.1 Modal Analysis

In order to verify the accuracy of the Abaqus model, the modal parameters of the Hålogaland Bridge obtained from Abaqus and Cov-SSI are compared. All the modal parameters are presented in Section 4.1.

5.1.1 Comparison of Modal Parameters from Abaqus and Cov-SSI

As previously mentioned, all identified modes with natural frequencies up to 1 Hz are considered from Cov-SSI. The reason is that the buffeting response calculations only include modes up to 1 Hz from Abaqus. The stabilisation plot that is utilised to identify the physical modes, displayed in Figure 4.4, shows clear straight sequences of stable poles, with little scatter. Nearly all poles in the plot are identified as physical modes. The auto-spectral densities of the horizontal, vertical and torsional acceleration response at the midspan of the bridge are also shown in the figure. The peaks of the auto-spectra correspond to the straight sequences of stable poles from the stabilisation plots, and corroborate the classification of the horizontal, vertical and torsional modes.

To compare the modal parameters obtained from Abaqus and Cov-SSI, it is necessary to assure correct mode pairing between the modes obtained from the two methods. Thus, both the natural frequencies and the mode shapes corresponding to each mode need to be regarded. The modes that are considered as corresponding modes between the two methods have a similar mode shape and natural frequency. Only the overall shapes, including the symmetry of the mode shapes and the number of half-sines, are considered in the comparison. The identified modes from Cov-SSI include 4 horizontal, 13 vertical and 3 torsional modes. All of these modes are also found from the modal analysis in Abaqus. A visual inspection of the mode shapes from Abaqus, shown in Figures 4.1, 4.2 and 4.3, and the mode shapes from Cov-SSI, shown in Figures 4.5, 4.6 and 4.7, shows that the shapes largely agree.

As explained earlier, not all identified modes are presented in Sections 4.1.1 and 4.1.2. Only mode shapes that can be correctly identified by the monitoring system are presented. The reason is that eight accelerometers on the bridge are utilised for the Cov-SSI: one at the midspan of the bridge, four on the left side and three on the right side. Thus, the mode shapes found with Cov-SSI will include a maximum of six or seven half-sines, for asymmetric or symmetric mode shapes, respectively. Mode shapes with more half-sines than this will be wrongfully represented. This is the case for 6 of the vertical modes that are identified from Cov-SSI. These are not shown in Section 4.1.2. However, the corresponding modes were found from the Abaqus model. Although these mode shapes from Cov-SSI can not be correctly represented, the similarity in the symmetry of the mode shapes and the values of the natural frequencies with the modes from Abaqus, justify the mode pairing. The natural frequencies of these modes from Cov-SSI and Abaqus are shown in Table 4.2, along with the natural frequencies from the correctly identified modes.

Furthermore, the natural frequencies of all identified modes from Cov-SSI and the corresponding modes from Abaqus are presented in Tables 4.1, 4.2 and 4.3. The relative differences between the natural frequencies obtained from the two methods, which are also shown in these tables, demonstrate that the natural frequencies are very similar. The largest relative difference is 6.2%, however most of the relative differences are below 3%. Figure 4.8 also illustrates the similarity of the natural frequencies obtained from the two methods. In addition, it can be seen that all natural frequencies obtained from Cov-SSI are larger than the corresponding values from Abaqus, except from the natural frequency of vertical mode number 13. This indicates that there is a systematic error present.

As can be seen from Table 4.1, horizontal mode number 5 and 6 were only found from the modal analysis in Abaqus, not from Cov-SSI. There are also several other modes found from the Abaqus analysis, which were not identified from Cov-SSI. However, these are modes related to the hangers or other parts of the bridge than the girder. Only modes related to the girder can be identified from Cov-SSI, since only sensors on the bridge girder are utilised. All modes from Abaqus with natural frequencies up to 1 Hz are included in the buffeting response calculations, including the modes that are mainly related to other parts than the bridge girder. However, since the response is calculated for the bridge girder, these modes have little contribution to the total response.

Furthermore, the damping ratios of all the modes that are identified from Cov-SSI are presented in Tables 4.1, 4.2 and 4.3. The damping ratios are low, as expected. Yet, they are all larger

than the structural damping ratio assumed in the buffeting response calculations, which is 0.5% for all modes. However, the damping ratios calculated from Cov-SSI include contributions from both structural and aerodynamic damping of the bridge. Therefore, the estimated damping ratios depend on the mean wind speed of the chosen acceleration interval. As the mean wind speed corresponding to the acceleration interval used to perform the Cov-SSI is 18 m/s, there is likely a significant contribution to the damping ratios from the aerodynamic damping. It is also worth mentioning that the damping ratios corresponding to the three last horizontal modes showed great variation, resulting in highly unreliable estimates. Furthermore, the tables show that the bridge is clearly under-critically damped, which is a necessary assumption for utilising the formulas to calculate the natural frequencies and damping ratios, given in Equation (2.33).

5.1.2 Reasons for Discrepancies

To summarise, all modes found with Cov-SSI are also obtained from the Abaqus analysis, but two of the horizontal modes from Abaqus were not identified with Cov-SSI. Although the Cov-SSI is performed to verify the Abaqus model, there are several sources of error related to the method, and several assumptions that need to be fulfilled. Firstly, after investigating the stabilisation plots from several different 30 minute intervals of acceleration recordings from the Hålogaland Bridge, only one interval is chosen to perform the final identification of the modes. The different intervals reveal very similar stabilisation plots. The same straight sequences of stable poles are found in most of the plots. The stabilisation plot from the interval that is chosen in the end includes the evident poles from all the investigated plots. However, the two last horizontal modes from Abaqus can not be identified from this recording. Since the loads on the bridge are not controlled when conducting the OMA, it is assumed that the bridge is excited by a broad band of frequencies. If this is not the case, not all modes of interest might be excited. Thus, the reason for the two unidentified modes from Cov-SSI might be that they were weakly excited during the chosen recording. By further investigation and mode identification from several other acceleration recordings, these modes might also be identified. However, as operational modal analysis is not a large part of this thesis, further investigations are not conducted.

Furthermore, if the assumption of broad band excitation is wrong and the loads on the bridge include dominant frequency components, these might be present in the stabilisation plot along with the actual modes of the bridge. This might be the case for the poles that are present in the chosen stabilisation plot, but are not identified as physical modes. Some of the reasons why these poles are not identified as structural modes of the Hålogaland Bridge, are that the mode

shapes from the poles look strange, the poles are not found from other recordings or no modes with natural frequencies close to these values are obtained from Abaqus.

It should also be mentioned that the modal parameters from Cov-SSI are found by simply picking one of the poles in the straight sequences of stable poles that are identified as physical modes. The natural frequencies and mode shapes corresponding to each of the poles within the straight sequences are almost identical. However, the damping ratios vary more. Therefore, these values are more uncertain, and the choice of pole influences these values to a greater extent. Furthermore, the choice of which acceleration interval to use as the basis for the identification of the modes from Cov-SSI might have a small influence on the modal parameters. However, the difference of the modal parameters obtained from different recordings should be small, as the system is assumed to be stationary.

The choices made when modelling the bridge in Abaqus will obviously have a great impact on the accuracy of the modal parameters obtained from the model. As the model is not made by the authors of this thesis, the influence of the modelling choices has not been investigated. The modal analysis in Abaqus is based on solving the eigenvalue problem given in Equation (2.16). This means that the damping of the bridge is neglected. Furthermore, this equation represents the eigenvalue problem for still-air conditions. Thus, the self-excited forces, which will be present because the bridge is exposed to wind loading, are not considered. As explained in Section 2.5.1, the self-excited forces depend on the motion of the bridge. Therefore, these forces will influence the eigenvalue problem. The fact that Abaqus provides still-air vibration modes, might cause small differences with the modal parameters obtained from Cov-SSI, as the Hålogaland Bridge is exposed to wind loading. However, a study conducted by Øiseth et al. (2015) indicates that errors of the natural frequencies, caused by neglecting the in-wind conditions, are limited. In the study, the in-wind natural frequencies of the Hardanger Bridge were predicted, by the use of a finite element model of the bridge in combination with the aerodynamic derivatives of the bridge cross section. The study showed that the natural frequencies barely changed for mean wind speeds that are relevant in this thesis.

From the comparison of the modal parameters obtained from the Abaqus model and Cov-SSI, it is concluded that the Abaqus model is able to represent the dynamic behaviour of the Hålogaland Bridge in a satisfactory way. The model includes all vibration modes that were identified from the measurements on the bridge, and both the mode shapes and natural frequencies largely agree with the ones obtained from Cov-SSI.

5.2 Wind Tunnel Test

The wind tunnel tests are performed to obtain expressions for the ADs and the load coefficients, used to model the buffeting load. Observing Figure 4.9, it can be seen that the polynomials fit reasonably well for all the ADs, although several discrepancies are present. It is worth noting that the values of KP_1^* from the test indicate two distinct trends for low reduced velocities, which is the result of different mean wind speeds. The difference in the values between the two trends for the low reduced velocities results in the polynomial not fitting one of the trends, meaning that it does not manage to reflect all aspects of the aerodynamic behaviour that the AD is supposed to describe. Furthermore, the polynomial of $K^2H_4^*$ differs greatly from the measured data for reduced velocities above 6, with the values seemingly following two different trends before and after a certain reduced velocity. This causes the polynomial to be ill-fitting in this range of reduced velocities. The polynomials of KH_1^* and $K^2P_4^*$ also show large deviations from some of the measured data points. As a final comment regarding the ADs, it is worth mentioning that the curves used to determine the expressions for the ADs are estimated as constant values outside the range of measured data points, in order to avoid unrealistic values. As there in reality is no information about the ADs in these ranges, an uncertainty is introduced into the buffeting response calculations.

The load coefficients shown in Figure 4.10 all show a clear and distinct trend. C_D shows overall more scatter than C_L and C_M . C_L has some scatter around zero degrees, which is the angle of flow incidence used in the calculations, meaning that the uncertainty of the coefficient is at its maximum at this point. However, the fitted polynomials match well for all the load coefficients.

5.3 Wind Field

5.3.1 Wind Field Characteristics

It is important to analyse the wind characteristics at the Hålogaland Bridge, as the wind field greatly affects the wind-induced response. Starting from Figure 4.11(a) it can be seen that the easterly winds are mostly perpendicular to the bridge, while the westerly winds show more scatter. This can be explained by the bridge being located in the innermost part of the Ofotfjord, where it branches off into two smaller fjords called the Rombak fjord and the Herjangs-

fjord, which stretches eastwards and northwards respectively. The Hålogaland Bridge lies at the mouth of the Rombak fjord, and the relatively steep mountainsides surrounding the fjord explain the predominance of perpendicular wind from the east (E). To the west, due to the connection with the Ofotfjord, the bridge is mainly exposed to wind coming from southwest (SW). However, the figure indicates that a significant amount of wind comes from two additional directions, namely northwest and south. As the buffeting theory is developed for wind approaching perpendicular to the bridge x-axis, the high amount of skew winds present at the site may cause discrepancies in the buffeting response predictions. The discarding of all recordings with mean wind below 3 m/s is clearly visible as the empty ring around the bridge, with the lowest mean wind speed used in the thesis being 3.1 m/s. The highest mean wind speed is around 25 m/s, while most of the recordings have a mean wind of 5-10 m/s, which is also seen in Figure 4.11(b). It should be noted that even though the amount of high mean wind speeds from the east is larger than from the west, the highest mean wind speed comes from the latter direction.

Figure 4.12 shows how the along-wind and vertical turbulence intensities are related to the mean wind speed and direction. It can be seen that high turbulence intensities generally correspond to low mean wind speeds. The plots show that there is a large scatter of the turbulence intensities for low mean wind speeds, which is most likely caused by non-stationary behaviour. Additionally, as the turbulence standard deviations are divided by the mean wind speed to calculate the turbulence intensities, similar standard deviations result in a larger scatter for low mean wind speeds than for higher. It is noticeable that high mean wind speeds result in less scatter of the turbulence intensities, and that the turbulence intensities in general are lower than for low mean wind speeds. Even if most of the high turbulence intensities occur for the lowest mean wind speeds, it can be worth noticing the cluster of fairly high turbulence intensities for winds coming from the south with mean wind speeds between 10-15 m/s. This may be attributed to the mountains south of the bridge causing more turbulence.

The large amount of scatter for low mean wind speeds can also be seen in Figure 4.13. In addition, a linear dependence between the turbulence intensities and the mean wind speed is noticeable in this figure, particularly for mean wind speeds below 15 m/s. It can be seen that the majority of the turbulence intensities lie in the range 0.02-0.2 and 0.01-0.1 for the along-wind and vertical components, respectively.

As previously explained, the length of the time series is chosen to satisfy the assumption of stationarity. In cases where the wind speed levels are sufficiently high, examination of the time series indicates that this assumption is correct. However, when the wind speed is low, several

time series show signs of non-stationarity, in the form of rapid changes in the wind direction. The discarding of all time series with mean wind speeds below 3 m/s is done to exclude effects due to non-stationarity. Yet, the histogram of the mean wind speeds, shown in Figure 4.11(b), clearly illustrates that a significant amount of the remaining recordings have mean wind speeds barely above 3 m/s. This means that a significant part of the included wind data might exhibit non-stationary behaviour. There are methods to evaluate the stationarity of the data, however no such tests are conducted in this thesis. For further analyses in this thesis, it is therefore important to consider that one of the base assumptions of the buffeting theory may be unsatisfied, and might cause discrepancies between the measured and the predicted acceleration response.

Another assumption is that of a homogeneous wind field. The mean wind speed and the turbulence intensities along the bridge girder are studied for several 10 minute recordings, to investigate how the wind characteristics vary along the span. Figure 4.14 shows plots of these wind characteristics along the bridge for one 10 minute recording, and illustrates the general trend observed in all the investigated recordings. Looking at the mean wind speed (Figure 4.14(a)), it can be seen that it tends to be lower at the south end of the bridge and increases towards the north end, with the largest relative difference being around 6.5% for the specific recording. When it comes to the turbulence intensities (Figures 4.14(b) and 4.14(c)), it is observable that they vary most for the anemometers that are closely spaced, while there in general seems to be no specific trend along the bridge girder. The largest relative differences for the turbulence intensities are much larger than those for the mean wind speed, with a 21.5% and a 29.3% relative difference for I_u and I_w , respectively. This reveals that in addition to a temporal variation, there is also a spatial variation of the turbulence intensities. These findings indicate that the wind field along the bridge is non-homogeneous, although it is difficult to determine the extent of the non-homogeneity. This is also in accordance with the expectation, which is based on the long span of the Hålogaland Bridge combined with a surrounding terrain with a certain amount of irregularity in the topography.

5.3.2 Wind Turbulence Spectra

In Figure 4.15, the estimated auto-spectra of the along-wind and vertical turbulence for a 10 minute recording with a mean wind speed of 15 m/s are depicted. Both the spectra from N400 with the proposed spectral parameters and the Kaimal spectra using the fitted spectral parameters are shown on top of the estimated spectra from measurements. It is observable that the fitted Kaimal spectra overall represents the turbulence to a greater degree than the N400 spec-

tra, especially in the 0.1-1 Hz frequency range. The normalised cross-spectra (Figure 4.16) for an arbitrary 10 minute recording with a mean wind speed of 17 m/s indicate that the coherence of the along-wind turbulence is somewhat higher than what is given by N400, while it is slightly lower for the vertical turbulence.

The probability distributions of the turbulence parameters are shown in Figures 4.17-4.20. The turbulence intensities are divided into three segments, containing values corresponding to mean wind speeds below 10 m/s, between 10 and 15 m/s or larger than 15 m/s. This is done to improve the buffeting response predictions for case 1-4, since the turbulence intensities show a linear dependence on the mean wind speed, as stated earlier. In this way, the percentile values of the turbulence intensities corresponding to each mean wind speed interval will be more accurate. However, the amount of samples in each segment will be smaller, which can cause issues when trying to represent the data using a lognormal probability distribution. Looking at the figures, it can be seen that the lognormal probability distributions fit the histograms of the turbulence parameters to a varying degree. For the turbulence intensities (Figures 4.17 and 4.18), a lognormal distribution seems to represent the two segments for mean wind speeds below 10 m/s and between 10-15 m/s. For the third segment, where the sample size is smallest, the agreement between the fitted lognormal distributions and the histograms are less reasonable. For the spectral parameters (Figure 4.19), it is observed that A_u is well represented by a lognormal distribution, while A_w is poorly represented. This is due to the lower boundary of the considered interval for the least squares fitting of the spectral parameters being set to 1.0. In the case of A_w , this value seems to be too high, causing the histogram to be distorted. Thus, the histogram would probably fit better to a lognormal distribution if lower values had been included in the considered interval for the least squares fitting. Lastly, the decay coefficients (Figure 4.20) are well represented by lognormal distributions.

Overall, the findings from the analysis of the wind field clearly imply that there is a large variability of the wind characteristics at the Hålogaland Bridge. This is in accordance with previous long-term monitoring studies of bridges (Bastos et al., 2018)(Fenerci et al., 2017)(Fenerci and Øiseth, 2017). Furthermore, similar to the latter study, the turbulence parameters are found to be represented by lognormal distributions.

5.3.3 Probabilistic Wind Field Model

In order to account for the variability of the wind field at the Hålogaland Bridge, a probabilistic model of the wind field has been established. The probabilistic model only includes the turbulence standard deviations. The plots in Figure 4.21 show that the turbulence standard deviations for both easterly and westerly winds can be well represented by lognormal probability distributions. Furthermore, it is evident from the figure that the standard deviations of the turbulence components are highly correlated for both easterly and westerly winds, as the scatter between the turbulence standard deviations shows a clear linear trend. In addition, the values for the correlation between the turbulence standard deviations, given in Table 4.7, are very high. Figures 4.22 and 4.23 show that there is a linear dependence between $\sigma_{u,w}$ and V . This dependence is especially clear for the easterly winds. For the westerly winds, the dependence is not as clear. However, the linear regression fits of the data highlight the linear trends also for these winds. The fact that there is much less data for the westerly winds might explain why the linear trend is not as conspicuous for these winds.

Due to the dependence of the mean wind speed on the turbulence standard deviations, the probability distributions of σ_u and σ_w are determined conditional to the mean wind speed. Figures 4.24, 4.25 and 4.26 show the plots used to obtain the expressions for the lognormal distribution parameters and the correlation coefficients for the probabilistic model. The $\tilde{\mu}$ -parameters are stabilising at a mean wind speed of approximately 11 m/s, where they start to show a linear variation. Thus, these parameters are modelled with linear curve fits to the data points in the stabilised range. As can be seen from the figures, the curves match the data points with mean wind speed larger than 11 m/s very well, but do not fit at all for the other data points. The $\tilde{\sigma}$ -parameters do not stabilise at any mean wind speed for either of the turbulence standard deviations for either direction, and are therefore modelled as constants. It is evident from the figures that these parameters are in reality not constant, and that the model curves do not fit the calculated values. The reason for approximating the $\tilde{\sigma}$ -parameters as constants is to avoid unrealistically low values for high mean wind speeds, as a curve fit to the data would result in decreasing curves. With this approximation, the $\tilde{\sigma}$ -parameters are overestimated for high mean wind speeds. The plot of the correlation coefficients only show slight variations with the mean wind speed, except for the correlation coefficient from the first mean wind speed interval for the easterly winds. Thus, these parameters are also modelled as constants. The constant values for $\tilde{\sigma}$ and $\rho_{\sigma_u\sigma_w}$ are obtained by considering only data with mean wind speed above 11 m/s, as the

model curves for the $\tilde{\mu}$ -parameters are only valid in this mean wind speed range.

Since the expressions for the lognormal distribution parameters and the correlation coefficients are obtained by only utilising data with mean wind speed larger than 11 m/s, the probabilistic model of $\sigma_{u,w}$ only applies to this mean wind speed range. It is worth noting that the amount of data used to obtain the probabilistic model is rather limited. Consequently, the number of values available to estimate the probability distributions of $\sigma_{u,w}$ for each mean wind speed interval is small. Investigations of the histograms for each interval reveal that some of them do not fit very well to a lognormal probability distribution. Thus, the lognormal parameters calculated for these intervals are not particularly accurate. Furthermore, it can be seen from Figures 4.24, 4.25 and 4.26 that there are very few data points in the high wind speed range. For the easterly and westerly winds there are only four and three points corresponding to a mean wind speed above 11 m/s, respectively. Thus, the observation of a linear trend of the $\tilde{\mu}$ -parameters for mean wind speeds larger than 11 m/s is based on very few points, and a small wind speed range. The largest mean wind speed represented in the plots is approximately 17 m/s. It is therefore uncertain how well the calculated expressions for the lognormal parameters and the correlation coefficients fit for higher mean wind speeds.

The performance of the probabilistic model of the turbulence standard deviations is assessed by comparing measured and simulated values of σ_u and σ_w . Figures 4.27, 4.28 and 4.29 show that the probabilistic model seems to represent the measured data reasonably well. As mentioned, due to the limited amount of data for mean wind speeds above 11 m/s, there were concerns that the plots shown in Figures 4.24, 4.25 and 4.26 did not manage to capture the behaviour of the lognormal parameters and that the estimated expressions would be highly inaccurate. This was especially a concern for mean wind speeds above 17 m/s, where there are no data points. Looking at Figures 4.27 and 4.28, the simulated turbulence standard deviations appear to represent the measured ones also for mean wind speeds larger than 17 m/s reasonably well. However it is difficult to assess the accuracy of the simulated turbulence parameters in the highest mean wind speed range, as there are very few samples from the measurements to compare with.

Additionally, the simulated turbulence parameters appear to reflect the linear dependence of the turbulence standard deviations on the mean wind speed. This indicates that the calculated expressions for $\tilde{\mu}$ manage to reflect the dependence in an accurate way. This is most evident in Figure 4.27, since the dependence of the turbulence standard deviations on the mean wind speed is most apparent for the easterly winds. It can be seen both in Figures 4.27 and 4.28 that

the scatter of the simulated σ_u and σ_w is at times smaller or larger than the scatter of the measured turbulence parameters. However, for the most part the scatter is largest for the simulated turbulence parameters. For the easterly winds this is particularly noticeable in the higher mean wind speed range, while for the westerly winds it is most apparent in the mid mean wind speed range (14-20 m/s). This is probably due to the $\tilde{\sigma}$ -parameters being modelled as constants, even though in reality they are not constant at all. Figures 4.24 and 4.25 show that the $\tilde{\sigma}$ -parameters for the easterly winds have a decreasing trend, while the parameters for the westerly winds show no apparent trend. For the easterly winds the constant $\tilde{\sigma}$ causes the variance of σ_u and σ_w to be larger than the measured for mean wind speeds larger than 11 m/s. For the westerly winds the variance is underestimated in the mean wind speed range of approximately 11-14 m/s, and overestimated for mean wind speeds above 14 m/s.

Furthermore, Figure 4.29 shows that the probabilistic model of the turbulence standard deviations overall is able to represent the relationship between σ_u and σ_w in a good way. The measured values of σ_u and σ_w clearly indicate a linear relationship between the two, with an increasing amount of scatter for higher values, which can also be observed for the simulated parameters. However, for high values of the turbulence standard deviations, the probabilistic model seems less able to represent the correlation between the two components.

5.4 Response of the Bridge

The final part of the discussion pertains what is arguably the main objective of this thesis, which is to assess to what degree the buffeting theory and wind field models are able to predict the true wind-induced response of the Hålogaland Bridge. Before comparing the predicted response with the measured response, the nature of the measured response as well as the influence of the wind characteristics on the response is examined. This also provides more insight into possible reasons for discrepancies between the predicted and the measured response.

5.4.1 Measured Response

Observing the wind rose plots of the RMS acceleration response in Figure 4.30, it is easy to detect that the largest response corresponds to high mean wind speeds from the two main directions, SW and E. It is noticeable that large values of the acceleration response mainly correspond to winds from these two directions for the vertical and torsional components, while for the lateral component, relatively large response is also recorded for skew winds. Furthermore, Figure

4.31, where the RMS acceleration response is plotted against the mean wind speed, indicates that high turbulence intensities are mainly related to low response for all components. For the vertical and torsional acceleration response, it is mainly low vertical turbulence intensity that correspond to large response. The lateral acceleration response on the other hand, seems less affected by the turbulence intensity. For mean wind speeds in the range 17-21 m/s, the acceleration response in all directions suddenly show a larger amount of scatter, which a variability in the turbulence intensity cannot be the cause of. This is particularly visible for the vertical acceleration response, where it can be seen that the wide scatter only contains response corresponding to low turbulence intensity. It is also interesting to note the response for low mean wind speeds for the lateral and vertical components, which shows a wide scatter (Figures 4.31(a) and 4.31(b)). In the study by Cheynet et al., 2016, it was found that a non-stationary flow often coincided with a wide scatter of the response. This could be an explanation for the mentioned phenomenon, since some of these samples are likely to exhibit non-stationary behaviour, as they correspond to wind recordings with low mean wind speeds.

The influence of the wind direction on the acceleration response is visualised in Figure 4.32, which shows a similar trend for the response in all three directions. For low mean wind speeds, the response is similar for both perpendicular and skew winds, from east and west. When the wind speed is higher, the easterly winds give higher response than the westerly winds. This trend is particularly evident in the mean wind speed range of 17-21 m/s, where the westerly winds show a significantly lower response than the easterly winds. This indicates that the sudden wide scatter in the response, mentioned in the previous paragraph, relates to the direction of the wind. Since the response seems to be affected by the wind direction, possible causes for this is further examined.

As mentioned in Section 3.5, the ADs and load coefficients of the bridge are nearly identical for winds approaching from east and west. This implies that whether the wind comes from east or west, the aerodynamic behaviour of the bridge is the same. Hence, it is unlikely that the aerodynamic behaviour of the bridge is the reason for the different acceleration response for winds from the two directions. However, a possible explanation could be that the westerly winds in general are more skew than the easterly, due to the main wind direction being southwest, as seen in Figure 4.11(a). A closer investigation of the wind indicates that the westerly winds in the mean wind speed range of 17-21 m/s are less perpendicular than the easterly. For mean wind speeds above 21 m/s, it appears that the easterly and westerly winds deviate to an equal degree from the 90°-direction and the 270°-direction, respectively. This could explain why the trend is

particularly evident in this specific wind speed range.

5.4.2 Comparison of Measured and Predicted Response

Case 1-4

As previously discussed, it is clear that there are huge variations of the wind field characteristics at the Hålogaland Bridge, mostly due to the random nature of the wind. Furthermore, the plots of the measured acceleration response of the Hålogaland Bridge, shown in Section 4.4.1, illustrate how this variability propagates into the wind-induced response of the bridge, in the same way as could be observed for the Hardanger Bridge in [Fenerci et al. \(2017\)](#). Moreover, this variability leads to difficulties in predicting the buffeting response. Looking at Figure 4.33 it can be observed that the predicted buffeting response of the Hålogaland Bridge for case 1-4 in general represents the measured response in a good way for the lateral and vertical directions, while the torsional response is in general overestimated. It is clear that the different cases, corresponding to different percentile values of the turbulence parameters from the measurements, result in entirely different response predictions. In addition, the case providing the most suitable response curve for design purposes is not the same for the three directions. Therefore, it is challenging to determine one specific case as the most suitable one. It is worth noting that the measured response plots may have looked somewhat different if more measurement data had been included in the analysis. With only two months of data available, there is for instance a limited amount of data corresponding to high mean wind speeds. Consequently, the shape of the measured response for high mean wind speeds is not very conspicuous, and there are few points to compare with the prediction curves.

For the lateral and vertical response predictions, the four cases show similar trends. Case 1 and 4 yield very similar response curves. For mean wind speeds up to 19 m/s, the two response curves lie approximately in the middle of the scatter from the measured response, whereas for higher wind speeds, the curves follow the lowest measured response. The response curve corresponding to case 2 results in the highest predicted response for both directions. The curve almost perfectly follows the highest measured response for the vertical direction, but shows a slight underestimation compared to the highest measured response for the lateral direction. Moreover, case 2 provides the most suitable design curves for these two directions. The response curve corresponding to case 3 lies in the middle of the curves from case 1 and 2. As discussed previously, it is noticeable that the response for the lateral and vertical directions for low mean

wind speeds show a large scatter. The predicted response curves do not represent the measured response in a good way for mean wind speeds in this range, especially for the vertical response. The response is underestimated, and also the gradient of the response curve is much larger than for the measured response, which is almost constant in the range of 3-5 m/s. The poor response prediction is assumed to be due to the fact that the assumption of stationarity is not satisfied for this wind speed range.

For the torsional direction on the other hand, it can be observed from Figure 4.33(c) that the response curve corresponding to case 1 almost perfectly follows the highest measured response, and is thus the most suitable design curve. The other three curves all overestimate the response to a certain degree. Case 4 gives a slightly higher response than case 1, case 2 drastically overestimates the response, and the curve corresponding to case 3 lies in the middle of case 2 and 4. These results differ from the findings of Macdonald (2003), which show an underestimation of the torsional buffeting response of the Second Severn Crossing when values of the turbulence intensity and total damping from the measurements are used. Fenerci and Øiseth (2017) also reported an underestimation of the torsional response, in this case for the Hardanger Bridge, using turbulence parameters determined from wind measurements.

Case 5

The final part of the comparison pertains the results from case 5, shown in Figures 4.34 and 4.35, where the probabilistic model of the wind field at the Hålogaland Bridge is used to predict the buffeting response. For both easterly and westerly winds the overall agreement between the predicted and the measured response for the lateral and vertical directions are satisfactory. For the easterly winds, the lateral and vertical acceleration response are consistently slightly underestimated. The predictions for the westerly winds are more inaccurate than for the easterly winds, with the predicted lateral and vertical response being more scattered than the measured. In addition, these two response components are overestimated for mean wind speeds ranging between 11-20 m/s. However, as previously mentioned, the measured response from the westerly winds are strangely scattered, and it is therefore not surprising that the response predictions do not manage to reflect the response in its entirety.

In addition, it should be noted that the predicted and measured response for case 5 originate from both perpendicular and skew winds. As the buffeting theory is derived under the assumption that the mean wind comes perpendicular to the bridge x-axis, the response predictions for the skew winds might be inaccurate. Since the westerly winds include a significant amount of

skew winds, this might explain why the predictions for the westerly winds are most inaccurate.

Furthermore, the predicted torsional response is severely overestimated and overly scattered for both easterly and westerly winds. Thus, the general trends observed for the buffeting response predictions corresponding to case 5 are the same as the ones for case 1-4. That is to say, the predicted lateral and vertical acceleration response show good agreement with the measured response, while the predicted torsional response is severely overestimated.

5.4.3 Influence of the Turbulence Parameters

As shown in Figures 4.15 and 4.16, the fitted turbulence spectra used to model the wind load, generally represent the measured turbulence spectra reasonably well. However, considering the variation of the spectra with time, it is impossible to determine one case for the turbulence spectra that can represent the whole wind field. As discussed, different alternatives of the turbulence parameters result in very different response curves, as can be observed in Figure 4.33. Therefore, the buffeting response is predicted for case 2-4, to illustrate how the different turbulence parameters, $A_{u,w}$, $K_{u,w}$ and $I_{u,w}$, influence the response predictions. For case 1, the 50th percentile values of all three parameters are used, and for the other three cases only one parameter is varied at the time, as explained in Section 3.8. It is clear, by comparison of case 1 with the other three cases, that the turbulence intensities are the most influential parameters for the predicted buffeting response, while the decay coefficients also show a significant influence. Comparison of case 1 and 4 makes it clear that a change in the values of $A_{u,w}$ barely affects the predicted response, especially for the lateral and vertical acceleration response. Thus, a good estimate of the turbulence intensities and the decay coefficients is crucial for an accurate response prediction, while the estimate of $A_{u,w}$ is not as important. It can be observed from Figure 4.33 that the conditional probability distributions of the turbulence intensities improve the response predictions, by preventing them from being unrealistically high for high mean wind speeds. Again, it should be stressed that the limited amount of measurement data causes the probability distributions, and thus the percentile values, of the turbulence parameters to be less accurate.

5.4.4 Contribution of Wind Forces on Cables and Hangers

Earlier it was stated that the response predictions reflect the measured response reasonably well for the lateral component. However, it should be mentioned that the good agreement is unexpected. The reason is that the wind forces acting on the cables and hangers are neglected in the

response calculations, even though they are expected to have a significant contribution to the lateral response. The reason for neglecting these forces is the lack of information about the wind characteristics at the cable level, causing difficulties in accurately describing the loading. As a consequence, it is presumed that the response predictions will show clear signs of underestimation, particularly for the lateral component. This phenomenon was observed in the study by [Fenerci and Øiseth \(2017\)](#), where the buffeting response of the Hardanger Bridge was predicted. The study examined this further and found that the lateral response of the bridge increased by approximately 25%, when modifying the drag coefficient for the bridge deck to include the cable forces. It should be noted that this estimate is conservative and gives an upper bound for the increase, due to the calculation assuming that the wind forces at the cables and on the bridge girder are perfectly correlated.

A similar investigation is not performed for the Hålogaland Bridge in this thesis, but it is probable that the increase in the lateral response would be of similar magnitude, as the bridges have similar lengths and cross sections. However, the cable geometry of the two bridges is slightly different. The distance between the two main cables of the Hålogaland Bridge is not constant along the girder, as it is largest at the midspan while it decreases towards the bridge towers. When the cables are sufficiently close, one cable might prevent the other one from being fully exposed to the wind. This might cause the total wind force on the cables to be smaller for the Hålogaland Bridge than for the Hardanger Bridge. However, since this is only the case for parts of the span, the significance of this is not assumed to be particularly large and it can not fully explain the unexpectedly high response prediction. Furthermore, it is worth noting that the lateral response is both under- and overestimated in previous studies. Similarly to the study performed by [Fenerci and Øiseth \(2017\)](#), [Cheynet et al. \(2016\)](#) also reported an underestimation of the lateral response. This was thought to be due to the local topography and non-stationarity of the wind. [Macdonald \(2003\)](#) on the other hand, reported an overestimation of the lateral response, presumably due to underestimation of the AD related to lateral damping, which was obtained from quasi-steady theory.

5.4.5 Uncertainty of the Structural Damping Ratio

A possible explanation for the unexpectedly high predicted lateral response and the severe overestimation of the predicted torsional response, is that the structural damping ratio used in the calculations is inaccurate for the lateral and torsional modes. N400 suggests a structural damping ratio between 0.5-0.8% for dynamic response calculations for steel structures ([Vegdirek-](#)

toratet, 2015). In this thesis, the lowest ratio has been chosen for all modes. In order to gain knowledge about the influence of the structural damping ratio on the predicted response, the calculations are repeated using different values for the damping. When a structural damping ratio of 0.3% is utilised, the response predictions increase with 13, 5, and 9% for a mean wind speed of 25 m/s for the lateral, vertical, and torsional direction, respectively. Increasing the structural damping ratio to 1% results in a decrease of 18, 10 and 15% for the three directions. It is clear that the lateral response is most sensitive to the structural damping, however the torsional response is also notably affected. Therefore, it would be interesting to try to estimate the actual structural damping ratio of the bridge.

The Cov-SSI analysis reported in this thesis is performed for an acceleration recording corresponding to a mean wind speed of 18 m/s. Thus, the total damping ratios that are given in Tables 4.1-4.3 will include a significant contribution from the aerodynamic damping. Therefore, several Cov-SSI analyses are performed using acceleration recordings with mean wind speeds around 4 m/s, where it can be assumed that the aerodynamic damping is small compared to the structural damping. From these analyses, it can generally be seen that the total damping ratios corresponding to the vertical and torsional modes lie at or slightly above 0.5%. Assuming that there is an insignificant amount of aerodynamic damping at this wind speed, the structural damping ratio used in the buffeting response predictions agrees quite well with the actual damping ratio for these modes. This further implies that the discrepancies related to the torsional response are not caused by the structural damping ratio. Furthermore, if the aerodynamic damping still contributes to the total damping, this would imply that the structural damping ratio used is too high. Using a lower structural damping ratio would have a negative impact on the agreement between the predicted and the measured response, particularly for the torsional component, since a lower damping ratio would result in an even higher response. For the horizontal modes however, the total damping ratios are found to be considerably larger than 0.5%. This indicates that the structural damping ratio for the horizontal modes are underestimated in the response calculations, leading to an overestimation of the response.

5.4.6 Uncertainty of the Load Coefficients and the Aerodynamic Derivatives

As defined in Section 2.5.1, the buffeting load caused by the wind turbulence is described using the load coefficients \bar{C}_D , \bar{C}_L , \bar{C}_M , C'_D , C'_L , and C'_M . These coefficients are dependent on the angle of flow incidence. When calculating the predicted buffeting response, the angle of incidence is set to zero and the coefficients are determined accordingly. As seen in Figure 4.10, the values

of the coefficients show great variation for different angles. Therefore, if the assumption of the angle of flow incidence is wrong, this could have an impact on the load coefficients, and thus also on the buffeting load and the predicted response. In addition, the coefficients are obtained through wind tunnel tests with a scaled section model, and a certain amount of uncertainty must be expected for these types of tests. The cross sectional admittance functions are also used to describe the load due to the turbulence. For simplicity, these are set to unity in this thesis, which is presumed to yield conservative response predictions (Macdonald, 2003).

The other contribution to the buffeting load is the motion induced load, which is represented by the use of the ADs. As previously discussed, the obtained expressions for the ADs are assumed to be reasonably accurate. However, the wind tunnel tests only gave experimental data points in a limited range of reduced velocities. The accuracy of the ADs outside this range is therefore uncertain. It is well known that the aerodynamic damping may have a great influence on the predicted buffeting response. Repeating the response calculations while neglecting the aerodynamic damping results in a 32, 125 and 63% increase of the lateral, vertical and torsional response, for a mean wind speed of 25 m/s. From this, it is seen that the aerodynamic damping has a significant influence on the predicted response, and any uncertainties in the ADs will propagate into the response calculations. By calculating the reduced velocity corresponding to the natural frequencies of all the modes included in the buffeting response calculations, it is found that only four modes correspond to reduced velocities within the range of experimental data points. Thus, for most of the ADs corresponding to natural frequencies, the accuracy is uncertain. Especially for the torsional response, this may cause inaccuracies in the calculations, as none of the torsional modes have natural frequencies that will give a reduced velocity within the range of experimental data points.

Another consideration regarding the motion induced load, is that the aerodynamic damping and stiffness matrices are considered as diagonal in the buffeting response calculations. Thus, any motion induced coupling between the modes is neglected. In a study performed by Øiseth et al. (2010), the buffeting response of the Hardanger Bridge was predicted both by considering and neglecting the motion induced coupling between the modes. The results from the study indicate that the two approaches yield very similar response, as long as the mean wind speed is lower than half of a certain stability limit. For the Hålogaland Bridge this limit is found to be 68.1 m/s (Kvamstad, 2011). As the largest mean wind speed considered in this thesis is well below half of this limit, mode coupling should not be an issue.

5.4.7 Uncertainty of the Numerical Model

The natural frequencies and mode shapes used in the buffeting response calculations are obtained from an Abaqus model of the Hålogaland Bridge. By comparison of the modal parameters obtained from the Cov-SSI analysis and the Abaqus analysis, it was concluded that the numerical model did manage to represent the dynamic behaviour of the bridge in a satisfactory manner. The natural frequencies obtained from the two methods are very similar, however it is observed that the torsional frequencies differs the most. Furthermore, the torsional frequencies from Abaqus are lower than those from Cov-SSI. This could indicate that the torsional stiffness of the bridge in the Abaqus model is too low, which might lead to an overestimation of the response. Additionally, when comparing the mode shapes from the two methods, only the overall shapes are considered. Thus, the exact accuracy of the mode shapes used in the response calculations is uncertain. However, it is unlikely that it is of great importance.

5.4.8 Uncertainty of the Probabilistic Wind Field Model

The probabilistic wind field model obtained in this thesis only include the turbulence standard deviations. However, a complete probabilistic model would comprise all the six turbulence parameters needed for the auto-spectra and the normalised cross-spectra of the turbulence. Such a probabilistic wind field model was proposed by [Fenerci and Øiseth \(2018a\)](#) for the wind field along the Hardanger Bridge, and the model successfully managed to represent the site-specific variability. As previously discussed, the decay coefficients have a significant influence on the predicted buffeting response, while the spectral parameters are not as important. Thus, extending the wind field model to at least include the decay coefficients, would probably improve the response predictions for case 5.

Nevertheless, the turbulence intensities are found to be the most influential parameters for the predicted buffeting response. For case 5, these parameters are calculated from the simulated turbulence standard deviations, which show good agreement with the ones from the measurements. This indicates that the probabilistic wind field model is not the cause of the overestimated torsional response and the unexpectedly high lateral response, illustrated in [Figures 4.34 and 4.35](#). The considerations regarding the wind forces on the cables and hangers, the structural and aerodynamic damping, the load coefficients and the Abaqus model are probably the cause of these discrepancies. However, the misrepresentation of the scatter of the predicted response for case 5 is believed to be caused by inaccuracies of the probabilistic wind field model. More

specifically, the approximation of the $\tilde{\sigma}$ -parameters as constants due to the lack of stabilisation is believed to be the main reason for this misrepresentation. This indicates that in order for a probabilistic wind field model to perform optimally, the distribution parameters need to stabilise. However, inaccurate $\tilde{\sigma}$ -parameters due to the limited amount of data could also be a part of the explanation. More measurement data would likely eliminate some of these inaccuracies.

Chapter 6

Conclusion and Further Work

6.1 Conclusion

The numerical model of the Hålogaland Bridge included all the vibration modes that were identified through Cov-SSI, and the comparison of the modal parameters from Abaqus and Cov-SSI showed good agreement. It was therefore concluded that the numerical model was able to represent the dynamic behaviour of the bridge in a satisfactory way.

From the wind data it could be observed that there were two main wind directions, namely the east and the southwest direction. Further analysis of the wind field revealed a large variability in the mean wind speed, mean wind direction and turbulence parameters. A thorough investigation of the latter parameters showed that they could be represented by lognormal probability distributions. In order to better reflect the variability of the wind field into the response predictions, a probabilistic model of the wind field at the Hålogaland Bridge was established. The model only included the turbulence standard deviations, as they were found to be the most influential turbulence parameters for the response predictions. The turbulence standard deviations were treated as correlated and lognormally distributed random variables. A dependence of the parameters on the mean wind speed was observed. In addition, there were clear differences between the winds approaching the bridge from east and west. Thus, the probability distributions of the parameters were established conditional on the mean wind speed and direction. The limited amount of data available, especially for high mean wind speeds, affected the accuracy of the probabilistic model. However, comparison of the turbulence standard deviations from the measurements and simulations from the model showed good agreement. It could be seen that the model managed to reflect both the linear dependence of the parameters on the mean wind speed and the correlation between the parameters. Therefore, it was concluded that the model was able to represent the measured turbulence standard deviations well.

A large variability of the measured acceleration response of the Hålogaland Bridge could be observed, making it difficult to predict the buffeting response. The turbulence parameters utilised in the response calculations greatly affected the predictions. It could be seen that while

the turbulence intensities were most influential, the decay coefficients also showed a considerable influence. The lateral and vertical response predictions generally showed good agreement with the measured response, while the predicted torsional response was severely overestimated, and for case 5 also overly scattered. However, the good agreement between the predicted and measured lateral response was unexpected. It was presumed that this response component would be underestimated, as the wind forces acting on the cables and hangers were neglected in the calculations. Furthermore, the structural and aerodynamic damping greatly affected the response predictions. Damping estimates from Cov-SSI suggested that the structural damping ratio used for the lateral modes was too low, which could explain the unexpectedly high predicted lateral response. In addition, the ADs used to describe the aerodynamic damping were found to be uncertain for all frequencies corresponding to the torsional modes. Moreover, comparison of the natural frequencies from the numerical model and Cov-SSI indicated that the torsional stiffness of the bridge in the Abaqus model was too low. These findings might explain the overestimation of the torsional response.

Thus, a final conclusion is that the buffeting theory and wind field models for the most part are able to predict the wind-induced response of the Hålogaland Bridge, when a probabilistic framework is used to reflect the variability of the wind field. However, the accuracy of all variables utilised in the response calculations is crucial in order to obtain satisfactory predictions. That is, good estimates of variables such as the turbulence parameters, the modal parameters, the load coefficients and the ADs should be emphasised during buffeting response analyses of bridges. Additionally, the results suggest that a probabilistic wind field model will be able to translate the variability of the wind into the response predictions, thereby predicting the response in its entirety.

6.2 Recommendations for Further Work

There are several aspects regarding the wind and the buffeting response predictions that have not been investigated in this thesis. Further research of these topics could improve the buffeting response predictions. A recurring issue in this thesis has been the limited amount of data from the monitoring system, which introduced inaccuracies into the response predictions. The main problem was that the limited amount of data caused the probabilistic representation of the turbulence parameters to be less accurate. Furthermore, the amount of measurement data available for comparison has been limited, particularly for high mean wind speeds. Thus, it has

been difficult to determine the accuracy of the predictions for mean wind speeds at this level. As the turbulence parameters greatly affected the predicted buffeting response, it would be of interest to repeat the analysis of the wind field and the response predictions using long-term data. This might lead to more accurate predictions of the buffeting response. In addition, only the turbulence standard deviations were included in the probabilistic model of the wind field at the Hålogaland Bridge. It would therefore be interesting to obtain a complete probabilistic model, including all the six turbulence parameters needed to describe the wind field. By comparison of the predicted response using the two probabilistic models, it could be observed how extensive the model should be in order to achieve a satisfactory representation of the variability of the response.

As previously mentioned, the stationarity of the data has not been properly investigated through stationarity tests, and for low mean wind speeds the data showed signs of non-stationarity. Additionally, a high amount of skew winds were present in the wind field. The effect of non-stationary wind fluctuations and skew winds on the response have not been examined thoroughly. Investigating these effects might lead to further knowledge of the wind-induced response of bridges.

Furthermore, the wind forces on the cables and hangers were neglected in this thesis. It would be interesting to investigate how these forces contribute to the response, particularly in the lateral direction. The buffeting load caused by the wind turbulence may be more accurately described by including the cross sectional admittance functions and by calculating the actual angle of flow incidence. In this thesis it has also been assumed that mode coupling does not occur for the considered mean wind speeds. This should be confirmed by calculating the response using multimode buffeting theory and examining the difference in the predicted response. Finally, the validity of using a structural damping ratio of 0.5% for all modes should be further examined.

Bibliography

- Abaqus (2019). Abaqus version 2019 User's Manuals. *Dassault Systèmes*.
- Bastos, F. E. Caetano, Á. Cunha, X. Cespedes, and O. Flamand (2018). Characterisation of the wind properties in the Grande Ravine viaduct. *Journal of Wind Engineering and Industrial Aerodynamics*, 173, 112–131. ISSN: 01676105. DOI: [10.1016/j.jweia.2017.12.012](https://doi.org/10.1016/j.jweia.2017.12.012).
- Bendat, J. S. and A. G. Piersol (2011). Random Data: analysis and measurement procedures. 4th ed. Hoboken, NJ, USA: John Wiley & Sons, Inc. ISBN: 9781118032428. DOI: [10.1002/9781118032428](https://doi.org/10.1002/9781118032428).
- Brincker, R. and P. H. Kirkegaard (2010). Special issue on Operational Modal Analysis. *Mechanical Systems and Signal Processing*, 24(5), 1209–1212. ISSN: 08883270. DOI: [10.1016/j.ymsp.2010.03.005](https://doi.org/10.1016/j.ymsp.2010.03.005).
- Cheyne, E., J. B. Jakobsen, and J. Snæbjörnsson (2016). Buffeting response of a suspension bridge in complex terrain. *Engineering Structures*, 128, 474–487. ISSN: 01410296. DOI: [10.1016/j.engstruct.2016.09.060](https://doi.org/10.1016/j.engstruct.2016.09.060).
- Davenport, A. G. (1961). The spectrum of horizontal gustiness near the ground in high winds. *Quarterly Journal of the Royal Meteorological Society*, 87(372), 194–211. ISSN: 00359009. DOI: [10.1002/qj.49708737208](https://doi.org/10.1002/qj.49708737208).
- Dyrbye, C. and S. O. Hansen (1997). Wind loads on structures. Chichester: Wiley. ISBN: 0471956511.
- Fenerci, A. and O. Øiseth (2017). Measured Buffeting Response of a Long-Span Suspension Bridge Compared with Numerical Predictions Based on Design Wind Spectra. *Journal of Structural Engineering*, 143(9), 04017131. ISSN: 0733-9445. DOI: [10.1061/\(ASCE\)ST.1943-541X.0001873](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001873).
- Fenerci, A. and O. Øiseth (2018a). Site-specific data-driven probabilistic wind field modeling for the wind-induced response prediction of cable-supported bridges. *Journal of Wind Engineering and Industrial Aerodynamics*, 181, 161–179. ISSN: 01676105. DOI: [10.1016/j.jweia.2018.09.002](https://doi.org/10.1016/j.jweia.2018.09.002).
- Fenerci, A. and O. Øiseth (2018b). Strong wind characteristics and dynamic response of a long-span suspension bridge during a storm. *Journal of Wind Engineering and Industrial Aerodynamics*, 172, 116–138. ISSN: 01676105. DOI: [10.1016/j.jweia.2017.10.030](https://doi.org/10.1016/j.jweia.2017.10.030).

- Fenerci, A., O. Øiseth, and A. Rønnquist (2017). Long-term monitoring of wind field characteristics and dynamic response of a long-span suspension bridge in complex terrain. *Engineering Structures*, 147, 269–284. ISSN: 01410296. DOI: [10.1016/j.engstruct.2017.05.070](https://doi.org/10.1016/j.engstruct.2017.05.070).
- Hermans, L. and H. van der Auweraer (1999). Modal testing and analysis of structures under operational conditions: industrial applications. *Mechanical Systems and Signal Processing*, 13(2), 193–216. ISSN: 08883270. DOI: [10.1006/mssp.1998.1211](https://doi.org/10.1006/mssp.1998.1211).
- Humar, J. L. (2002). Dynamics of structures. 2nd ed. Lisse: Balkema. ISBN: 9058092453.
- Kaimal, J. C., J. C. Wyngaard, Y. Izumi, and O. R. Coté (1972). Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, 98(417), 563–589. ISSN: 00359009. DOI: [10.1002/qj.49709841707](https://doi.org/10.1002/qj.49709841707).
- Kármán, T. von (1948). Progress in the Statistical Theory of Turbulence. *Proceedings of the National Academy of Sciences*, 34(11), 530–539. ISSN: 0027-8424. DOI: [10.1073/pnas.34.11.530](https://doi.org/10.1073/pnas.34.11.530).
- Kvåle, K. A. (2022). knutankv/koma: Minor fixes (v.1.0.7). Zenodo. DOI: [10.5281/ZENODO.5881841](https://doi.org/10.5281/ZENODO.5881841). URL: <https://zenodo.org/record/5881841>.
- Kvåle, K. A., O. Øiseth, and A. Rønnquist (2017). Operational modal analysis of an end-supported pontoon bridge. *Engineering Structures*, 148, 410–423. ISSN: 01410296. DOI: [10.1016/j.engstruct.2017.06.069](https://doi.org/10.1016/j.engstruct.2017.06.069).
- Kvåle, K. A., R. Sigbjörnsson, and O. Øiseth (2016). Modelling the stochastic dynamic behaviour of a pontoon bridge: A case study. *Computers & Structures*, 165, 123–135. ISSN: 00457949. DOI: [10.1016/j.compstruc.2015.12.009](https://doi.org/10.1016/j.compstruc.2015.12.009).
- Kvamstad, T. H. (2011). Assessment of the flutter stability limit of the Hålogaland Bridge using a probabilistic approach [Master's thesis]. The Norwegian University of Science and Technology. URL: <http://hdl.handle.net/11250/236896>.
- Macdonald, J. H. (2003). Evaluation of buffeting predictions of a cable-stayed bridge from full-scale measurements. *Journal of Wind Engineering and Industrial Aerodynamics*, 91(12-15), 1465–1483. ISSN: 01676105. DOI: [10.1016/j.jweia.2003.09.009](https://doi.org/10.1016/j.jweia.2003.09.009).
- Newland, D. E. (2005). An Introduction to Random Vibrations, Spectral & Wavelet Analysis. 3rd ed. Mineola, N.Y: Dover. ISBN: 0486442748.

- Øiseth, O., A. Rönnquist, K. A. Kvåle, and R. Sigbjörnsson (2015). Monitoring Wind Velocities and Dynamic Response of the Hardanger Bridge. URL: https://doi.org/10.1007/978-3-319-15248-6_13.
- Øiseth, O., A. Rönnquist, and R. Sigbjörnsson (2010). Simplified prediction of wind-induced response and stability limit of slender long-span suspension bridges, based on modified quasi-steady theory: A case study. *Journal of Wind Engineering and Industrial Aerodynamics*, 98(12), 730–741. ISSN: 01676105. DOI: [10.1016/j.jweia.2010.06.009](https://doi.org/10.1016/j.jweia.2010.06.009).
- Øiseth, O., A. Rönnquist, and R. Sigbjörnsson (2013). Effects of co-spectral densities of atmospheric turbulence on the dynamic response of cable-supported bridges: A case study. *Journal of Wind Engineering and Industrial Aerodynamics*, 116, 83–93. ISSN: 01676105. DOI: [10.1016/j.jweia.2013.03.001](https://doi.org/10.1016/j.jweia.2013.03.001).
- Petersen, Ø. W., G. T. Frøseth, and O. A. Øiseth (2021). Design and deployment of a monitoring system on a long-span suspension bridge. ISSN: 2564-3738. URL: <https://hdl.handle.net/11250/2982037>.
- Rainieri, C. and G. Fabbrocino (2014). Operational Modal Analysis of Civil Engineering Structures : An Introduction and Guide for Applications. 1st ed. New York, NY: Springer New York : Imprint: Springer. ISBN: 1-4939-0767-0.
- Rao, K. D. and M. N. S. Swamy (2018). Basics of Multirate Digital Signal Processing. *Digital Signal Processing: Theory and Practice*. Singapore: Springer Singapore, pp. 513–573. ISBN: 978-981-10-8081-4. URL: https://doi.org/10.1007/978-981-10-8081-4_8.
- Reeve, A. (2021). npTDMS. URL: <https://nptdms.readthedocs.io/en/stable/>.
- Scanlan, R. H. and J. J. Tomko (1971). Airfoil and Bridge Deck Flutter Derivatives. *Journal of the Engineering Mechanics Division*, 97(6), 1717–1737. ISSN: 0044-7951. DOI: [10.1061/JMCEA3.0001526](https://doi.org/10.1061/JMCEA3.0001526).
- Simiu, E. and R. H. Scanlan (1996). Wind effects on structures: Fundamentals and applications to design. 3rd ed. New York: Wiley.
- Standard Norge (2005). Eurokode 1: Laster på konstruksjoner. Del 1-4: Allmenne laster. Vindlaster. *NS-EN 1991-1-4:2005+NA:2009*.
- Strømmen, E. (2006). Theory of Bridge Aerodynamics. Berlin/Heidelberg: Springer-Verlag. ISBN: 3-540-30603-X. DOI: [10.1007/3-540-30604-8](https://doi.org/10.1007/3-540-30604-8).

- Tamura, Y. and A. Kareem (2013). *Advanced Structural Wind Engineering*. Ed. by Y. Tamura and A. Kareem. Tokyo: Springer Japan. ISBN: 978-4-431-54336-7. DOI: [10.1007/978-4-431-54337-4](https://doi.org/10.1007/978-4-431-54337-4).
- Tan, L. and J. Jiang (2013). Multirate Digital Signal Processing, Oversampling of Analog-to-Digital Conversion, and Undersampling of Bandpass Signals. *Digital Signal Processing*. Elsevier, pp. 555–619. DOI: [10.1016/B978-0-12-415893-1.00012-3](https://doi.org/10.1016/B978-0-12-415893-1.00012-3).
- Thomopoulos, N. T. (2018). Lognormal. *Probability Distributions*. Cham: Springer International Publishing, pp. 135–148. URL: https://doi.org/10.1007/978-3-319-76042-1_9.
- Tseng, C.-C. and S.-L. Lee (2017). Closed-form designs of digital fractional order Butterworth filters using discrete transforms. *Signal Processing*, 137, 80–97. ISSN: 01651684. DOI: [10.1016/j.sigpro.2017.01.015](https://doi.org/10.1016/j.sigpro.2017.01.015).
- Vegdirektoratet (2015). Håndbok N400 - Bruprosjektering.
- Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2), 70–73. ISSN: 0018-9278. DOI: [10.1109/TAU.1967.1161901](https://doi.org/10.1109/TAU.1967.1161901).
- Žerovnik, G., A. Trkov, and I. A. Kodeli (2012). Correlated random sampling for multivariate normal and log-normal distributions. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 690, 75–78. ISSN: 01689002. DOI: [10.1016/j.nima.2012.06.036](https://doi.org/10.1016/j.nima.2012.06.036).

Appendix A

Python Scripts

The appendix contains the most important Python scripts used for data processing, probabilistic wind field modelling, buffeting response calculations and operational modal analysis. Most of the scripts utilise the Python package *NumPy*, which is shortened to *np* in the scripts. In order to shorten the appendix, the lines for importing the package are removed from the scripts in the appendix.

A.1 Data Processing

lowPass_DownSample.py

```

1 def filter_data(s, Wn, N, Fs, ftype):
2     '''
3     Arguments:
4     s          : signal, 1D-array
5     Wn         : cutoff frequency
6     N          : order of the filter
7     Fs         : sampling frequency of signal
8     ftype      : type of filter {'lowpass', 'highpass'}
9     Returns:
10    filtered   : filtered signal
11    -----
12    Takes a signal and applies a filter using Butterworth filter of order N.
13    '''
14    #Normalizes the cutoff frequency, with 1 being the Nyquist frequency
15    normWn = 2*Wn/Fs
16    #Filter coefficients
17    sos = signal.butter(N, normWn, output='sos', btype=ftype)
18    #Forward-backward digital filter
19    filtered = signal.sosfiltfilt(sos, s)
20
21    return filtered
22
23 def lowPassAndDownSample(s, Wn, N, downFreq, Fs, t=[]):
24     '''
25     Arguments:
26     s          : signal, 1D-array
27     Wn         : cutoff frequency, 40 % of the downFreq
28     N          : order of the low pass filter
29     downFreq   : the frequency that the signal is down sampled to
30     Fs         : sampling frequency of signal
31     t          : optional argument, time vector to be downsampled
32     Returns:
33     newSignal  : down-sampled signal
34     -----
35     Takes a signal and applies low pass filter using Butterworth filter, then down samples
36     the signal. If t is given, the function also outputs a resampled time vector
37     '''
38    #Low pass filter
39    filtered = filter_data(s, Wn, N, Fs, 'lowpass')
40    #%%Down sampling by decimation
41    factor = (Fs/downFreq)
42    samples = np.shape(filtered)[0] #Number of samples in original signal
43    n = int(samples/factor) #number of samples in new signal
44

```

```

45     if len(t)!=0:
46         time_out = True
47         t_new = np.zeros(n)
48     else:
49         time_out = False
50     newSignal = np.zeros(n)
51     for i in range(n):
52         newSignal[i] = filtered[i*int(factor)]
53         if time_out:
54             t_new[i] = t[i*int(factor)]
55
56     if time_out:
57         return t_new, newSignal
58     else:
59         return newSignal

```

time_synchronisation.py

```

1 def time_master(tdms_files, wind_nodes):
2     '''
3     Arguments:
4     tdms_files      : list of tdms_file
5     wind_nodes      : list of strings containing the common part of anemometer name
6                       belonging
7                       to considered anode(s). If looking at all nodes, wind_nodes is full
8                       list with
9                       'xx' at indices where there are no anemometers. If looking at one
10                      anode, wind_nodes
11                      only contains the specific anodes
12
13 Returns:
14 t_master          : master time vector given in seconds
15 -----
16 Function takes in a list of tdms-objects, and searches for the one
17 that starts sampling last and the one that ends sampling first, and computes
18 the master time vector from these. The search includes all anemometers.
19 '''
20 start_time = 0 #Initial value for start of master time vector
21 end_time = 1e100 #Initial value for end of master time vector
22
23 #Finding start time and end time
24 for i in range(len(tdms_files)):
25     temp_start = tdms_files[i]['acceleration_data']['timestamp'][0]
26     temp_end = tdms_files[i]['acceleration_data']['timestamp'][-1]
27     if temp_start > start_time:
28         start_time = temp_start
29     if temp_end < end_time:
30         end_time = temp_end
31     anemometers = [s for s in list(tdms_files[i]) if wind_nodes[i] in s]
32     for k in range(len(anemometers)):

```

```

29         temp_start = tdms_files[i][anemometers[k]]['timestamp'][0]
30         temp_end = tdms_files[i][anemometers[k]]['timestamp'][-1]
31         if temp_start > start_time:
32             start_time = temp_start
33         if temp_end < end_time:
34             end_time = temp_end
35
36     t_master = np.arange(np.ceil(start_time*1e-9), np.floor(end_time*1e-9), 1/32)
37     return t_master
38
39 def time_interpolation(data, timestamp, t):
40     '''
41     Arguments:
42     data      : the data to be interpolated
43     timestamp : timestamp corresponding to data, given in nanoseconds
44     t        : master time vector, given in seconds
45     Returns:
46     datn     : time synchronised data
47     -----
48     Function interpolates the data to a master time vector
49     '''
50     t1 = t
51     t = timestamp*1e-9
52     datn = interp1d(t, data)(t1)
53     return datn

```

measurementFunctions.py

```

1 def transform(sensor1, sensor2, B):
2     """
3     Arguments:
4     sensor1 : data from sensor 1 in the sensor pair
5     sensor2 : data from sensor 2 in the sensor pair
6     B      : width of the deck
7     Returns:
8     acc_trans : the transformed data, array with size Nx3, where first column
9                 is y, second column is z and third column is theta
10    -----
11    Function transforms the accelerations from a sensor pair into one component
12    in y, z and theta direction
13    """
14    acc_trans = np.zeros((np.max(np.shape(sensor1)),3))
15    acc_trans[:,0] = (sensor1[:,1] + sensor2[:,1])/2
16    acc_trans[:,1] = (sensor1[:,2] + sensor2[:,2])/2
17    acc_trans[:,2] = (-sensor1[:,2]+sensor2[:,2])/B
18
19    return acc_trans
20
21 def stdAccMeasurements(data_acc):

```



```

22     """
23     Arguments:
24     data_acc : acceleration data , units m/s^2
25     Returns:
26     sigma_y, sigma_z, sigma_theta : standard deviation in y, z and theta direction
27     -----
28     This function is made for both input from one sensor pair, and from all
29     sensor pairs. It takes the acceleration data and calculates the standard deviation
30     """
31     B = 18.6 #width of bridge deck
32
33     data_acc=data_acc-np.mean(data_acc,axis=0) # Removing the mean from the series
34     acc_trans = np.zeros([data_acc.shape[0], int(data_acc.shape[1]/2)])
35     k=0
36     for i in range(0,data_acc.shape[1],6):
37         acc_trans[:,k:k+3] = transform(data_acc[:,i:i+3], data_acc[:,i+3:i+6],B) #
38         #Transforming into one component per sensor pair
39         k +=3
40     #Separating into each component
41     acc_y = acc_trans[:,np.arange(0,acc_trans.shape[1], 3)] #Acceleration in y-direction
42     acc_z = acc_trans[:,np.arange(1,acc_trans.shape[1], 3)] #Acceleration in z-direction
43     acc_theta = acc_trans[:,np.arange(2,acc_trans.shape[1], 3)] #Acceleration in theta-
44     #direction
45     # Computing the standard deviation
46     sigma_y = np.std(acc_y, axis=0)
47     sigma_z= np.std(acc_z, axis=0)
48     sigma_theta = np.std(acc_theta, axis=0)
49
50     return sigma_y,sigma_z, sigma_theta

```

clean_data.py

```

1 def remove_error(data, fs, status):
2     '''
3     Arguments :
4     data      : time series
5     fs       : sampling frequency
6     status   : vector of status codes from the sensor
7     Returns :
8     newData  : cleaned time series
9     error_ratio : amount of error values in time series
10    -----
11    Function returns the clean time series with a time vector. Time
12    is assumed to start from zero. The new series have the same sampling rate as the input
13    Linear interpolation is conducted to preserve the sampling rate
14    NB! Will result in error if the first or last sample is an error value (time
15    interpolation will fail)
16    '''
17
18    #Generate time vector

```

```

17     l = np.max(np.shape(data))
18     t = np.arange(0,l)*1/fs
19     t1 = t #Store initial time vector
20     #Deleting the values that have error
21     error_ind = np.where(np.logical_and(status!='00', status!='0A'))[0]
22     #Apply linear interpolation if there are error values
23     if np.size(error_ind)!=0:
24         data = np.delete(data, error_ind, axis=0)
25         t = np.delete(t, error_ind)
26         newData = np.zeros((l, 3))
27         for i in range(3):
28             if np.size(data[:,i])==0:
29                 newData[:,i] = np.zeros((l, np.shape(data[:,i])[1]))
30                 error_ratio = 1
31             else:
32                 newData[:,i] = interp1d(t, data[:,i])(t1)
33                 error_ratio = np.size(error_ind)/l
34     else:
35         newData = data
36         error_ratio = 0
37
38     return newData, error_ratio
39
40 def remove_std(data, fs, x):
41     '''
42     Arguments:
43     data      : time series
44     x         : discard values greater than x*std in an absolute manner
45     fs        : sampling frequency
46     Returns:
47     datn      : time series
48     stdrig    : bool proclaiming if there were any values exceeding limit
49     -----
50     Function discards values larger than x times the standard deviation in an
51     absolute sense. Returns a clean time series with same sampling rate as input.
52     Linear interpolation is conducted to preserve the sampling rate.
53
54     Function is based on code written by Aksel Fenerci
55     '''
56     #Generate time vector
57     l = np.max(np.shape(data))
58     t = np.arange(0, l)*1/fs
59     t1 = t
60
61     ind = []
62     datn = np.zeros((l, 3))
63     for i in range(3):
64         dummy = np.argwhere(data[:,i] > np.mean(data[:,i]) + x *np.std(data[:,i]))
65         dummy2 = np.argwhere(data[:,i] < np.mean(data[:,i]) - x*np.std(data[:,i]))

```

```

66     ind = np.concatenate((dummy, dummy2))
67     if np.size(ind)!=0:
68         stdtrig = True
69     else:
70         stdtrig = False
71     if stdtrig:
72         ind = np.sort(ind, axis=0)
73         if ind[0] == 0:
74             ind = np.delete(ind, 0)
75             dat = np.delete(data[:,i], ind)
76             t = np.delete(t1, ind)
77             dat[0] = np.mean(data[:,i]) + x *np.std(data[:,i])
78         elif ind[-1] == 1-1:
79             ind = np.delete(ind, -1)
80             dat = np.delete(data[:,i], ind)
81             t = np.delete(t1, ind)
82             dat[-1] = np.mean(data[:,i]) + x *np.std(data[:,i])
83         else:
84             dat = np.delete(data[:,i], ind)
85             t = np.delete(t1, ind)
86             datn[:,i] = interp1d(t, dat)(t1)
87     else:
88         datn[:,i] = data[:,i]
89
90     return datn, stdtrig
91
92 def circArray(data, dataRange):
93     '''
94     Arguments:
95     data      : time series
96     dataRange : range where the value of data is valid
97     Returns:
98     dataCirc : corrected data
99     -----
100    Function translates circular array (typically angles).
101
102    Function based on code written by Knut Andreas Kvaale
103    '''
104    dataCirc = data
105    span = np.max(dataRange) - np.min(dataRange)
106    dataCirc[data>np.max(dataRange)] = dataCirc[data>np.max(dataRange)] - span
107    dataCirc[data<np.min(dataRange)] = span + dataCirc[data<np.min(dataRange)]
108
109    return dataCirc

```

windCharacteristics.py

```

1 def angmean(angles):
2     '''

```

```

3     Arguments :
4     angles : array of angles
5     Returns :
6     mean_angle : mean angle of the array
7     -----
8     Function enestablishes mean of an array of angles
9
10    Function based on code written by Knut Andreas Kvaale
11    '''
12    mean_angle = np.arctan2(np.nanmean(np.sin(angles*np.pi/180)), np.nanmean(np.cos(angles*
13    np.pi/180)))*180/np.pi
14    #Adjust the values to lie between 0 and 360 degrees
15    if mean_angle<0:
16        mean_angle = mean_angle + 360
17    return mean_angle
18
19 def transform_uvw(data, Fs, interval, varargin=[]):
20     '''
21     Arguments :
22     data      : a nx3 matrix, where data[:,0] is direction, data[:,1] is magnitude
23                in horizontal direction and data[:,2] is the vertical velocity
24     Fs       : sampling frequency
25     interval  : time interval in minutes
26     varargin  : list of additional arguments, varargin[0] is the index of which
27                ten-minute interval that is of interest, and varargin[1] is a
28                string defining whether to detrend or not
29
30     Returns :
31     V        : mean wind velocity
32     u        : turbulence component in along-wind direction
33     v        : turbulence component in across-wind direction
34     w        : turbulence component in vertical direction, mean wind included
35     meandir  : mean direction of the wind
36     -----
37     Function transforms wind data to uvw coordinates from the polar coordinates
38
39     Function based on code written by Aksel Fenerci
40     '''
41     detrend_opt = []
42     ind = []
43     if np.size(varargin)!=0:
44         ind = [varargin[0]]
45         if np.size(varargin)>1:
46             detrend_opt = varargin[1]
47     if detrend_opt!='on':
48         detrend_opt = 'off' #default
49     # Finding number of intervals in data
50     l = np.max(np.shape(data))
51     t = np.arange(0,l)*1/Fs
52     ints = t[-1]/60/interval

```

```

51     if ints < 1:
52         ints = 1
53     if ints%1 !=0:
54         ints = int(np.ceil(ints)) - 1
55     l2 = interval*60*Fs*ints + ints #The index of the last element in the last whole
    interval
56
57     data = np.delete(data, np.arange(l2,1, 1), axis=0) #Deleting the excess data that does
    not fit in an interval
58     p = int(l2/ints)
59     pp = p
60     c = 0
61     #Initializing matrices
62     meandir = np.zeros(ints)
63     phi = np.zeros((pp,ints))
64     V_alpha= np.zeros((pp, ints))
65     V_beta = np.zeros((pp, ints))
66     V = np.zeros(ints)
67     u = np.zeros((pp,ints))
68     v = np.zeros((pp, ints))
69     w = np.zeros((pp, ints))
70     if np.any(np.isnan(data)):
71         meandir = np.nan
72         V = np.nan
73         u = np.nan
74         v = np.nan
75         w = np.nan
76     else:
77         if np.size(ind)==0:
78             for i in range(ints):
79                 if ~np.any(np.isnan(np.squeeze(data[c:pp,:]))) :
80                     meandir[i] = angmean(data[c:pp,0])
81                     phi[:,i] = (data[c:pp,0] - meandir[i])*np.pi/180
82                     V_alpha[:,i] = data[c:pp,1]*np.cos(phi[:,i])
83                     V_beta[:,i] = data[c:pp,1]*np.sin(phi[:,i])
84                     V[i] = np.nanmean(V_alpha[:,i])
85                     u[:,i] = V_alpha[:,i] - V[i]
86                     v[:,i] = V_beta[:,i]
87                     if detrend_opt == 'on':
88                         w[:,i] = data[c:pp,2] - np.nanmean(data[c:pp,2])
89                     elif detrend_opt == 'off':
90                         w[:,i] = data[c:pp,2]
91                 else:
92                     V[i] = np.nan
93                     meandir[i] = np.nan
94                     u[:,i] = np.nan*np.arange(0, np.max(np.shape(data[c:pp,1])))
95                     w[:,i] = np.nan*np.arange(0, np.max(np.shape(data[c:pp,1])))
96                     w[:,i] = np.nan*np.arange(0, np.max(np.shape(data[c:pp,1])))
97             c = c + p

```

```

98         pp = pp + p
99     else:         #Transformation for one particular interval
100         c = c + p*(ind)
101         pp = pp + p*(ind)
102         if ~np.any(np.isnan(np.squeeze(data[c:pp,:]))) :
103             meandir = angmean(data[c:pp,0])
104             phi = (data[c:pp,0] - meandir)*np.pi/180
105             V_alpha = data[c:pp,1]*np.cos(phi)
106             V_beta = data[c:pp,1]*np.sin(phi)
107             V = np.nanmean(V_alpha)
108             u = V_alpha - V
109             v = V_beta
110             if detrend_opt == 'on':
111                 w = data[c:pp,2] - np.nanmean(data[c:pp,2])
112             elif detrend_opt == 'off':
113                 w = data[c:pp,2]
114         else:
115             V = np.nan
116             meandir = np.nan
117             u = np.nan*np.arange(0, np.max(np.shape(data[c:pp,1])))
118             v = np.nan*np.arange(0, np.max(np.shape(data[c:pp,1])))
119             w = np.nan*np.arange(0, np.max(np.shape(data[c:pp,1])))
120
121     return V, u, v, w, meandir
122
123 def windProperties(u, w, V):
124     """
125     Arguments :
126     u : turbulence in along wind direction for one anemometer
127     w : turbulence in vertical direction for one anemometer
128     V : mean wind speed
129     Returns:
130     sigma_u : standard deviation of along-wind turbulence
131     sigma_w : standard deviation of vertical turbulence
132     I_u      : turbulence intensity in along wind direction
133     I_w      : turbulence intensity in vertical direction
134     -----
135     Function finds standard deviation and turbulence intensity of the wind. The turbulence
136     given as input is without mean value.
137     """
138     # Standard deviation
139     sigma_u = np.std(u, axis=0)
140     sigma_w = np.std(w, axis=0)
141     # Turbulence intensity
142     I_u = sigma_u/V
143     I_w = sigma_w/V
144     return sigma_u, sigma_w, I_u, I_w

```

get_data.py

```

1 import numpy as np
2 import measurementFunctions as mf
3 import lowPass_DownSample as ld
4 import clean_data as cd
5 import windCharacteristics as wc
6 import time_synchronisation as ts
7
8 def readAcc(anodeName, tdmsFile, interval, t_master, downFreq):
9     """
10    Arguments:
11    anodeName : name of logger box
12    tdmsFile  : one .tdms-file, containing raw data from 8 hour of recordings
13                from one data logger
14    interval  : chosen length of intervals in minutes
15    t_master  : master time vector, for time synchronization
16    downFreq  : frequency that the signal is downsampled to
17    Returns:
18    acc_y     : processed horizontal acceleration data, n_intervals*n_samples
19    acc_z     : processed vertical acceleration data, n_intervals*n_samples
20    acc_theta : processed torsional acceleration data, n_intervals*n_samples
21    -----
22    Function extracts processed acceleration data from one logger box for eight hours of
23    data
24    """
25    # Constants
26    g = 9.82          # Gravity constant
27    B = 18.6         # Width of bridge deck
28    Fs = 64          # Sampling frequency
29    N = 10           # Order of the low pass filter
30    acc_names = ['-1x', '-1y', '-1z', '-2x', '-2y', '-2z']
31    # Retrieving data from file
32    acc_data = tdmsFile['acceleration_data']
33    length = np.max(np.shape(acc_data['timestamp'][:])) # Length of acceleration series
34    # Fixing correct units
35    acc_raw = np.zeros((length, 6))
36    for i in range(6):
37        conversion_factor = float(acc_data[anodeName + acc_names[i]].properties['
38        conversion_factor'])
39        acc_raw[:,i] = acc_data[anodeName + acc_names[i]][:] / conversion_factor * g
40    # Time synchronisation
41    acc_sync = np.zeros((int(length/(Fs/32)), 6))
42    for i in range(6):
43        acc_sync[:,i] = ld.lowPassAndDownSample(acc_raw[:,i] - np.mean(acc_raw[:,i]),
44        0.5*32, N, 32, Fs, False) + np.mean(acc_raw[:,i])
45    stamp = acc_data['timestamp'][:]
46    stamp_sync = ld.lowPassAndDownSample(acc_raw[:,0] - np.mean(acc_raw[:,0]), 0.5*32, N,
47    32, Fs, False, stamp)[0]

```

```

44     acc_sync_fixed = np.zeros((np.shape(t_master)[0], 6))
45     for i in range(6):
46         acc_sync_fixed[:,i] = ts.time_interpolation(acc_sync[:,i], stamp_sync, t_master)
47
48     Fs = 32 # New sampling frequency
49     length = np.shape(acc_sync_fixed)[0] # New length of acceleration series
50     # Lowpass and down sample
51     acc = np.zeros((int(length/(Fs/downFreq)), 6))
52     for i in range(6):
53         acc[:,i] = ld.lowPassAndDownSample(acc_sync_fixed[:,i] - np.mean(acc_sync_fixed[:,i]
54         ), 0.5*downFreq, N, downFreq, Fs, False) + np.mean(acc_sync_fixed[:,i])
55     dt_new = 1 / downFreq
56     t_new = np.arange(0, np.max(np.shape(acc))) * dt_new
57     length_new = np.max(np.shape(acc))
58     # Preparations for division in intervals
59     ints = t_new[-1] / 60 / interval # Number of intervals
60     if ints < 1: # If number of intervals is less than one
61         ints = 1
62     if ints % 1 != 0: # If number of intervals is a decimal number
63         ints = int(np.ceil(ints)) - 1
64     l2 = interval * 60 * downFreq * ints + ints # New length
65     acc = np.delete(acc, np.arange(l2, length_new, 1), axis=0)
66     # Dividing into intervals
67     p = int(l2 / ints)
68     pp = p
69     c = 0
70     # Initializing acceleration matrices
71     acc_y = np.zeros([ints, p])
72     acc_z = np.zeros([ints, p])
73     acc_theta = np.zeros([ints, p])
74     for i in range(ints):
75         acc_trans = mf.transform(acc[c:pp, 0:3], acc[c:pp, 3:6], B)
76         acc_y[i, :] = acc_trans[:, 0]
77         acc_z[i, :] = acc_trans[:, 1]
78         acc_theta[i, :] = acc_trans[:, 2]
79         c = c + p
80         pp = pp + p
81
82     return acc_y, acc_z, acc_theta
83
84 """ Function for extracting processed wind-data from one anemometer
85
86 def readWind(anemometerName, tdmsFile, interval, t_master, downFreq):
87     """
88     Arguments:
89     anemometerName : name of anemometer from the tdmsFile
90     tdmsFile : one .tdms-file, containing raw data from 8 hour of recordings
91                 from one logger box
92     interval : chosen length of intervals in minutes
93     t_master : master time axis, for time synchronization

```



```

92     downFreq : frequency that the signal is down sampled to
93     Returns:
94     u : processed wind data; along wind direction, n_samples*n_intervals
95     w : processed wind data; vertical direction, n_samples*n_intervals
96     -----
97     Function extracts wind data from one anemometer and processes it.
98     """
99     Fs = 32          # Sampling frequency
100    N = 10           # Order of the low pass filter
101    #Retrieving data from file
102    wind = tdmsFile[anemometerName]
103    status = wind['status_code'][:]
104    direction = wind['direction'][:]
105    magnitude = wind['magnitude'][:]
106    vertical = wind['vertical_velocity'][:]
107    # Combining wind data into one big matrix
108    wind_data = np.zeros((np.max(np.shape(direction)), 3))
109    wind_data[:,0] = direction
110    wind_data[:,1] = magnitude
111    wind_data[:,2] = vertical
112    # Removing error values
113    newData, error_ratio = cd.remove_error(wind_data, Fs, status)
114    # Time synchronization
115    stamp = wind['timestamp'][:]
116    data_sync = np.zeros((np.shape(t_master)[0], 3))
117    for i in range(3):
118        data_sync[:,i] = ts.time_interpolation(newData[:,i], stamp, t_master)
119    # Clean data
120    newData, stdtrig = cd.remove_std(data_sync, Fs, 6)
121    # Fixing spike in directional data
122    newData[:,0] = cd.circArray(newData[:,0], [-180 + np.mean(newData[:,0]), 180 + np.mean(
123        newData[:,0])])
124    # Resampling the data
125    if Fs != downFreq:
126        length = np.max(np.shape(newData))
127        data_rs = np.zeros((int(length/(Fs/downFreq)), 3))
128        for i in range(3):
129            data_rs[:,i] = ld.lowPassAndDownSample(newData[:,i] - np.mean(newData[:,i]),
130                0.5*downFreq, N, downFreq, Fs, False) + np.mean(newData[:,i])
131    else:
132        data_rs = newData
133    # Obtaining the turbulence components
134    V, u, v, w, meandir = wc.transform_uvw(np.real(data_rs), downFreq, interval)
135    # Applying highpass filter
136    for i in range(np.min(np.shape(u))):
137        u[:,i] = ld.filter_data(u[:,i] - np.mean(u[:,i]), 1/300, 2, downFreq, 'highpass') +
138            np.mean(u[:,i])
139        w[:,i] = ld.filter_data(w[:,i] - np.mean(w[:,i]), 1/300, 2, downFreq, 'highpass') +
140            np.mean(w[:,i])

```

```

137
138     return u, w, V, meandir
139
140 def getCharacteristics(tdms_file, node_name, anemometer, t_master, downFreq):
141     '''
142     Arguments:
143     filename      : a tdms-file containing 8 hours of data
144     node_name     : a string specifying which node to look at
145     anemometer    : string containing the name of the chosen anemometer
146     t_master      : master time axis, for time synchronisation
147     downFreq     : downFreq : frequency that the signal is down sampled to
148     Returns:
149     V              : mean wind speed
150     sigma_u,w     : standard deviation for turbulence components
151                   : of 10 minutes
152     I_u,w         : turbulence intensity
153     u, w          : turbulence components
154     sigma_y,z,theta : standard deviation of acceleration
155     -----
156     Function goes through one eight-hour file and extracts wind and acceleration
157     characteristics for intervals of 10 minutes. The file that is
158     given as argument should be read in advance.
159     '''
160     # Acceleration
161     acc_y, acc_z, acc_theta = readAcc(node_name, tdms_file, 10, t_master, downFreq)
162     sigma_y = np.std(acc_y, axis=1)
163     sigma_z = np.std(acc_z, axis=1)
164     sigma_theta = np.std(acc_theta, axis=1)
165     # Wind
166     u, w, V, meandir = readWind(anemometer, tdms_file, 10, t_master, downFreq)
167     _, _, I_u, I_w = wc.windProperties(u, w - np.mean(w, axis=0), V)
168     return V, meandir, u, w, I_u, I_w, sigma_y, sigma_z, sigma_theta

```

A.2 Probabilistic Wind Field Model

logNormalDist.py

```

1 def logNormParam(data):
2     '''
3     Arguments:
4     data : the data that the lognormal parameters are fitted to
5     Returns:
6     my, sigma : parameters of the lognormal distribution
7     -----
8     Function for finding the distribution parameters for data with a lognormal distribution
9     '''
10    norm = np.log(data)

```

```

11     my = np.mean(norm)
12     sig = np.std(norm)
13
14     return my, sig

```

windProbApproach.py

```

1  """ Modules
2  import numpy as np
3  import os
4  import logNormalDist as lognorm
5
6  """ Info for loading data
7  nodes_folder = ['\\anode003', '\\anode004', '\\anode005', '\\anode006', '\\anode007', '\\
            anode008', '\\anode009', '\\anode010']
8  save_path = os.getcwd() + '\\Results'
9  """ Retrieve data from csv-files
10 data = []
11 files = []
12 for file in os.listdir(save_path + nodes_folder[4]):
13     if (file.endswith('.csv') and os.path.getsize(save_path + nodes_folder[4] + '\\'+ file)
14         > 0):
15         files.append(file)
16         data.append(np.loadtxt(save_path + nodes_folder[4] + '\\'+ file, delimiter=','))
17 elem = 0
18 for i in range(len(data)):
19     if data[i].ndim == 1:
20         elem += 1
21     else:
22         elem += np.shape(data[i])[0]
23 """ Array of variables
24 V = np.zeros(elem)
25 meandir = np.zeros(elem)
26 Iu = np.zeros(elem)
27 Iw = np.zeros(elem)
28 Au = np.zeros(elem)
29 Aw = np.zeros(elem)
30 c = 0
31 for i in range(len(data)):
32     if data[i].ndim == 1:
33         V[c] = data[i][0]
34         meandir[c] = data[i][1]
35         Iu[c] = data[i][2]
36         Iw[c] = data[i][3]
37         Au[c] = data[i][7]
38         Aw[c] = data[i][8]
39         c += 1
40     else:
41         for k in range(np.shape(data[i])[0]):

```

```

41         V[c] = data[i][k,0]
42         meandir[c] = data[i][k,1]
43         Iu[c] = data[i][k,2]
44         Iw[c] = data[i][k,3]
45         Au[c] = data[i][k,7]
46         Aw[c] = data[i][k,8]
47         c += 1
48     sigma_u = Iu * V
49     sigma_w = Iw * V
50     ### Dividing data into easterly and westerly wind
51     index_east = np.argwhere(np.logical_and(meandir > 0, meandir <= 180))
52     index_west = np.argwhere(np.logical_and(meandir > 180, meandir <= 360))
53
54     V_east = V[index_east]
55     V_west = V[index_west]
56     sigma_u_east = sigma_u[index_east]
57     sigma_u_west = sigma_u[index_west]
58     sigma_w_east = sigma_w[index_east]
59     sigma_w_west = sigma_w[index_west]
60     ### Correlation coefficients matrix - only wind above 11
61     corr_east = np.corrcoef(np.array([sigma_u_east[np.argwhere(V_east[:,0] >= 11)[:,0],0],
62                                     sigma_w_east[np.argwhere(V_east[:,0] >= 11)[:,0],0])))
63     corr_west = np.corrcoef(np.array([sigma_u_west[np.argwhere(V_west[:,0] >= 11)[:,0],0],
64                                     sigma_w_west[np.argwhere(V_west[:,0] >= 11)[:,0],0])))
65     ### Conditional probability parameters and correlation coefficients
66     east_sort = np.argsort(V_east[:,0])
67     V_e = V_east[east_sort,0]
68     sigma_u_e = sigma_u_east[east_sort,0]
69     sigma_w_e = sigma_w_east[east_sort,0]
70
71     west_sort = np.argsort(V_west[:,0])
72     V_w = V_west[west_sort,0]
73     sigma_u_w = sigma_u_west[west_sort,0]
74     sigma_w_w = sigma_w_west[west_sort,0]
75
76     ## Easterly wind
77     V_east_list = []
78     sigma_u_east_list = []
79     sigma_w_east_list = []
80     start = 0
81     end = 200
82     move = end
83     num = np.floor(np.shape(V_e)[0]/end-1)
84     for i in np.arange(int(num)):
85         V_east_list.append(V_e[start:end])
86         sigma_u_east_list.append(sigma_u_e[start:end])
87         sigma_w_east_list.append(sigma_w_e[start:end])
88         start = end
89         end = end + move

```

```

88 V_east_list.append(V_e[start:])
89 sigma_u_east_list.append(sigma_u_e[start:])
90 sigma_w_east_list.append(sigma_w_e[start:])
91 V_east_mean = np.zeros([len(V_east_list)])
92 param_sigma_u_east = np.zeros([len(sigma_u_east_list), 2])
93 param_sigma_w_east = np.zeros([len(sigma_w_east_list), 2])
94 corr_east_list = np.zeros([len(V_east_list)])
95 for i in np.arange(len(sigma_u_east_list)):
96     param_sigma_u_east[i,:] = lognorm.logNormParam(sigma_u_east_list[i])
97     param_sigma_w_east[i,:] = lognorm.logNormParam(sigma_w_east_list[i])
98     V_east_mean[i] = np.mean(V_east_list[i])
99     corr_east_list[i] = np.corrcoef(np.array([sigma_u_east_list[i], sigma_w_east_list[i]]))
    [1,0]
100
101 ## Westerly wind
102 V_west_list = []
103 sigma_u_west_list = []
104 sigma_w_west_list = []
105 start = 0
106 end = 150
107 move = end
108 num = np.floor(np.shape(V_w)[0]/end-1)
109 for i in np.arange(int(num)):
110     V_west_list.append(V_w[start:end])
111     sigma_u_west_list.append(sigma_u_w[start:end])
112     sigma_w_west_list.append(sigma_w_w[start:end])
113     start = end
114     end = end + move
115 V_west_list.append(V_w[start:])
116 sigma_u_west_list.append(sigma_u_w[start:])
117 sigma_w_west_list.append(sigma_w_w[start:])
118 V_west_mean = np.zeros([len(V_west_list)])
119 param_sigma_u_west = np.zeros([len(sigma_u_west_list), 2])
120 param_sigma_w_west = np.zeros([len(sigma_w_west_list), 2])
121 corr_west_list = np.zeros([len(V_west_list)])
122 for i in np.arange(len(sigma_u_west_list)):
123     param_sigma_u_west[i,:] = lognorm.logNormParam(sigma_u_west_list[i])
124     param_sigma_w_west[i,:] = lognorm.logNormParam(sigma_w_west_list[i])
125     V_west_mean[i] = np.mean(V_west_list[i])
126     corr_west_list[i] = np.corrcoef(np.array([sigma_u_west_list[i], sigma_w_west_list[i]]))
    [1,0]

```

simulate_wind_field.py

```

1 import os
2
3 def simulate_wind(V, direction):
4     '''
5     Arguments:

```

```

6     V           : mean wind velocity of a 10 min recording
7     direction   : the direction of the mean wind, 'east' or 'west'
8     Returns:
9     sigma      : sample of standard deviation of along wind and vertical turbulence
10    -----
11    Functions takes in a mean wind velocity, and establishes the conditional distributions
12    of the wind field model. First, a set of multivariate correlated normally distributed
13    random variables are generated, and then the parameters are obtained by taking the
14    natural exponent
15    '''
16    if direction=='east':
17        my_t = [0.07046*V - 0.8886, 0.04191*V - 1.1549]
18        sigma_t = [0.3211, 0.2481]
19        rho = np.array([[1, 0.8713], [0.8713, 1]])
20    else:
21        my_t = [0.04458*V - 0.6850, 0.02141*V - 0.9824]
22        sigma_t = [0.4324, 0.4604]
23        rho = np.array([[1, 0.9268], [0.9268, 1]])
24    cov = np.zeros((len(sigma_t), len(sigma_t)))
25    for i in range(len(sigma_t)):
26        for j in range(len(sigma_t)):
27            cov[i,j] = np.log(rho[i,j]*np.sqrt(np.exp(sigma_t[i]**2)-1)*np.sqrt(np.exp(
28            sigma_t[j]**2) - 1) + 1)
29    # Generating a sample
30    rng = np.random.default_rng()
31    x = rng.multivariate_normal(my_t, cov)
32    sigma = np.exp(x)
33
34    return sigma
35
36    """
37    """
38    """
39    """
40    """
41    """
42    """
43    """
44    """
45    """
46    """
47    """
48    """
49    """
50    """
51    """
52    """
53    """
54    """
55    """
56    """
57    """
58    """
59    """
60    """
61    """
62    """
63    """
64    """
65    """
66    """
67    """
68    """
69    """
70    """
71    """
72    """
73    """
74    """
75    """
76    """
77    """
78    """
79    """
80    """
81    """
82    """
83    """
84    """
85    """
86    """
87    """
88    """
89    """
90    """
91    """
92    """
93    """
94    """
95    """
96    """
97    """
98    """
99    """
100   """
101   """
102   """
103   """
104   """
105   """
106   """
107   """
108   """
109   """
110   """
111   """
112   """
113   """
114   """
115   """
116   """
117   """
118   """
119   """
120   """
121   """
122   """
123   """
124   """
125   """
126   """
127   """
128   """
129   """
130   """
131   """
132   """
133   """
134   """
135   """
136   """
137   """
138   """
139   """
140   """
141   """
142   """
143   """
144   """
145   """
146   """
147   """
148   """
149   """
150   """
151   """
152   """
153   """
154   """
155   """
156   """
157   """
158   """
159   """
160   """
161   """
162   """
163   """
164   """
165   """
166   """
167   """
168   """
169   """
170   """
171   """
172   """
173   """
174   """
175   """
176   """
177   """
178   """
179   """
180   """
181   """
182   """
183   """
184   """
185   """
186   """
187   """
188   """
189   """
190   """
191   """
192   """
193   """
194   """
195   """
196   """
197   """
198   """
199   """
200   """
201   """
202   """
203   """
204   """
205   """
206   """
207   """
208   """
209   """
210   """
211   """
212   """
213   """
214   """
215   """
216   """
217   """
218   """
219   """
220   """
221   """
222   """
223   """
224   """
225   """
226   """
227   """
228   """
229   """
230   """
231   """
232   """
233   """
234   """
235   """
236   """
237   """
238   """
239   """
240   """
241   """
242   """
243   """
244   """
245   """
246   """
247   """
248   """
249   """
250   """
251   """
252   """
253   """
254   """
255   """
256   """
257   """
258   """
259   """
260   """
261   """
262   """
263   """
264   """
265   """
266   """
267   """
268   """
269   """
270   """
271   """
272   """
273   """
274   """
275   """
276   """
277   """
278   """
279   """
280   """
281   """
282   """
283   """
284   """
285   """
286   """
287   """
288   """
289   """
290   """
291   """
292   """
293   """
294   """
295   """
296   """
297   """
298   """
299   """
300   """
301   """
302   """
303   """
304   """
305   """
306   """
307   """
308   """
309   """
310   """
311   """
312   """
313   """
314   """
315   """
316   """
317   """
318   """
319   """
320   """
321   """
322   """
323   """
324   """
325   """
326   """
327   """
328   """
329   """
330   """
331   """
332   """
333   """
334   """
335   """
336   """
337   """
338   """
339   """
340   """
341   """
342   """
343   """
344   """
345   """
346   """
347   """
348   """
349   """
350   """
351   """
352   """
353   """
354   """
355   """
356   """
357   """
358   """
359   """
360   """
361   """
362   """
363   """
364   """
365   """
366   """
367   """
368   """
369   """
370   """
371   """
372   """
373   """
374   """
375   """
376   """
377   """
378   """
379   """
380   """
381   """
382   """
383   """
384   """
385   """
386   """
387   """
388   """
389   """
390   """
391   """
392   """
393   """
394   """
395   """
396   """
397   """
398   """
399   """
400   """
401   """
402   """
403   """
404   """
405   """
406   """
407   """
408   """
409   """
410   """
411   """
412   """
413   """
414   """
415   """
416   """
417   """
418   """
419   """
420   """
421   """
422   """
423   """
424   """
425   """
426   """
427   """
428   """
429   """
430   """
431   """
432   """
433   """
434   """
435   """
436   """
437   """
438   """
439   """
440   """
441   """
442   """
443   """
444   """
445   """
446   """
447   """
448   """
449   """
450   """
451   """
452   """
453   """
454   """
455   """
456   """
457   """
458   """
459   """
460   """
461   """
462   """
463   """
464   """
465   """
466   """
467   """
468   """
469   """
470   """
471   """
472   """
473   """
474   """
475   """
476   """
477   """
478   """
479   """
480   """
481   """
482   """
483   """
484   """
485   """
486   """
487   """
488   """
489   """
490   """
491   """
492   """
493   """
494   """
495   """
496   """
497   """
498   """
499   """
500   """
501   """
502   """
503   """
504   """
505   """
506   """
507   """
508   """
509   """
510   """
511   """
512   """
513   """
514   """
515   """
516   """
517   """
518   """
519   """
520   """
521   """
522   """
523   """
524   """
525   """
526   """
527   """
528   """
529   """
530   """
531   """
532   """
533   """
534   """
535   """
536   """
537   """
538   """
539   """
540   """
541   """
542   """
543   """
544   """
545   """
546   """
547   """
548   """
549   """
550   """
551   """
552   """
553   """
554   """
555   """
556   """
557   """
558   """
559   """
560   """
561   """
562   """
563   """
564   """
565   """
566   """
567   """
568   """
569   """
570   """
571   """
572   """
573   """
574   """
575   """
576   """
577   """
578   """
579   """
580   """
581   """
582   """
583   """
584   """
585   """
586   """
587   """
588   """
589   """
590   """
591   """
592   """
593   """
594   """
595   """
596   """
597   """
598   """
599   """
600   """
601   """
602   """
603   """
604   """
605   """
606   """
607   """
608   """
609   """
610   """
611   """
612   """
613   """
614   """
615   """
616   """
617   """
618   """
619   """
620   """
621   """
622   """
623   """
624   """
625   """
626   """
627   """
628   """
629   """
630   """
631   """
632   """
633   """
634   """
635   """
636   """
637   """
638   """
639   """
640   """
641   """
642   """
643   """
644   """
645   """
646   """
647   """
648   """
649   """
650   """
651   """
652   """
653   """
654   """
655   """
656   """
657   """
658   """
659   """
660   """
661   """
662   """
663   """
664   """
665   """
666   """
667   """
668   """
669   """
670   """
671   """
672   """
673   """
674   """
675   """
676   """
677   """
678   """
679   """
680   """
681   """
682   """
683   """
684   """
685   """
686   """
687   """
688   """
689   """
690   """
691   """
692   """
693   """
694   """
695   """
696   """
697   """
698   """
699   """
700   """
701   """
702   """
703   """
704   """
705   """
706   """
707   """
708   """
709   """
710   """
711   """
712   """
713   """
714   """
715   """
716   """
717   """
718   """
719   """
720   """
721   """
722   """
723   """
724   """
725   """
726   """
727   """
728   """
729   """
730   """
731   """
732   """
733   """
734   """
735   """
736   """
737   """
738   """
739   """
740   """
741   """
742   """
743   """
744   """
745   """
746   """
747   """
748   """
749   """
750   """
751   """
752   """
753   """
754   """
755   """
756   """
757   """
758   """
759   """
760   """
761   """
762   """
763   """
764   """
765   """
766   """
767   """
768   """
769   """
770   """
771   """
772   """
773   """
774   """
775   """
776   """
777   """
778   """
779   """
780   """
781   """
782   """
783   """
784   """
785   """
786   """
787   """
788   """
789   """
790   """
791   """
792   """
793   """
794   """
795   """
796   """
797   """
798   """
799   """
800   """
801   """
802   """
803   """
804   """
805   """
806   """
807   """
808   """
809   """
810   """
811   """
812   """
813   """
814   """
815   """
816   """
817   """
818   """
819   """
820   """
821   """
822   """
823   """
824   """
825   """
826   """
827   """
828   """
829   """
830   """
831   """
832   """
833   """
834   """
835   """
836   """
837   """
838   """
839   """
840   """
841   """
842   """
843   """
844   """
845   """
846   """
847   """
848   """
849   """
850   """
851   """
852   """
853   """
854   """
855   """
856   """
857   """
858   """
859   """
860   """
861   """
862   """
863   """
864   """
865   """
866   """
867   """
868   """
869   """
870   """
871   """
872   """
873   """
874   """
875   """
876   """
877   """
878   """
879   """
880   """
881   """
882   """
883   """
884   """
885   """
886   """
887   """
888   """
889   """
890   """
891   """
892   """
893   """
894   """
895   """
896   """
897   """
898   """
899   """
900   """
901   """
902   """
903   """
904   """
905   """
906   """
907   """
908   """
909   """
910   """
911   """
912   """
913   """
914   """
915   """
916   """
917   """
918   """
919   """
920   """
921   """
922   """
923   """
924   """
925   """
926   """
927   """
928   """
929   """
930   """
931   """
932   """
933   """
934   """
935   """
936   """
937   """
938   """
939   """
940   """
941   """
942   """
943   """
944   """
945   """
946   """
947   """
948   """
949   """
950   """
951   """
952   """
953   """
954   """
955   """
956   """
957   """
958   """
959   """
960   """
961   """
962   """
963   """
964   """
965   """
966   """
967   """
968   """
969   """
970   """
971   """
972   """
973   """
974   """
975   """
976   """
977   """
978   """
979   """
980   """
981   """
982   """
983   """
984   """
985   """
986   """
987   """
988   """
989   """
990   """
991   """
992   """
993   """
994   """
995   """
996   """
997   """
998   """
999   """
1000  """

```

```

48 elem = 0
49 for i in range(len(data)):
50     if data[i].ndim == 1:
51         elem += 1
52     else:
53         elem += np.shape(data[i])[0]
54 #Array of all V and meandir
55 V = np.zeros(elem)
56 meandir = np.zeros(elem)
57 c = 0
58 for i in range(len(data)):
59     if data[i].ndim == 1:
60         V[c] = data[i][0]
61         meandir[c] = data[i][1]
62         c += 1
63     else:
64         for k in range(np.shape(data[i])[0]):
65             V[c] = data[i][k,0]
66             meandir[c] = data[i][k,1]
67             c += 1
68 ### ----- SIMULATION -----
69 V_lim = 11
70 V_upper = 30
71 V_east = V[np.argwhere(np.logical_and(np.logical_and(V >=V_lim, V <= V_upper), np.
72     logical_and(meandir > 0, meandir <= 180)))]
73 V_west = V[np.argwhere(np.logical_and(np.logical_and(V >=V_lim, V <= V_upper), np.
74     logical_and(meandir > 180, meandir <= 360)))]
75 sigma_east = np.zeros((len(V_east), 2))
76 sigma_west = np.zeros((len(V_west), 2))
77 for i in range(len(V_east)):
78     sigma_east[i,:] = simulate_wind(V_east[i][0], 'east')
79 for i in range(len(V_west)):
80     sigma_west[i,:] = simulate_wind(V_west[i][0], 'west')

```

A.3 Buffeting Response

fitting.py

```

1 def autoSpectraParam(Fs, u, w, v_m, fitting):
2     """
3     Arguments:
4     Fs      : sampling frequency
5     u      : along-wind velocity data from one anemometer, without mean wind
6     w      : vertical velocity data from one anemometer
7     v_m    : mean wind velocity
8     fitting : parameter to decide if the spectral parameters from N400 should
9             be improved by least square fitting
10    Returns:

```

```

11     A_u,w      : spectral parameters for the one-point auto spectra
12     lse_Au,w  : least square error for A_u,w. Set to zero if fitting is not computed.
13     -----
14     Function takes in the turbulence in u- and w- component for a 10 min recording, and
15     finds the spectral parameters that gives best fit of the Kaimal auto spectra (Kaimal et
16     al., 1972) to the spectra from data by using least squares error.
17     """
18     z = 54.2 # height of anemometer
19     # Parameters from wind measurements
20     w = w - np.mean(w) # Subtracting mean value to find zero-mean turbulence component
21     sigma_u = np.std(u)
22     sigma_w = np.std(w)
23
24     if fitting == False:
25         lse_Au = 0
26         lse_Aw = 0
27     else:
28         # Auto-spectra computed with Welch' method
29         Nwelch = 8
30         Nwindow = np.floor(np.max(np.shape(u))/Nwelch) # Length of window
31         f, S_u = sci.csd(u, u, Fs, nperseg = Nwindow)
32         f, S_w = sci.csd(w, w, Fs, nperseg = Nwindow)
33         #Removing start and end values
34         f = np.delete(f, [0,-1])
35         S_u = np.delete(S_u, [0,-1])
36         S_w = np.delete(S_w, [0,-1])
37         # Improvement of A_u,w by least square fitting
38         A_val = np.arange(1,80.1,0.1) # Vector of A_i values
39         lse_Au = np.zeros(np.shape(A_val)[0]) # Vector of least square errors for A_u
40         lse_Aw = np.zeros(np.shape(A_val)[0]) # Vector of least square errors for A_w
41
42         cnt = 0
43         for A_i in A_val:
44             sum_u = 0
45             sum_w = 0
46             for i in range(np.shape(f)[0]):
47                 n_uhat_i = f[i]*z/v_m
48                 S_u_kaimal_i = A_i*n_uhat_i/(1+1.5*A_i*n_uhat_i)**(5/3)
49                 n_what_i = f[i]*z/v_m
50                 S_w_kaimal_i = A_i*n_what_i/(1+1.5*A_i*n_what_i)**(5/3)
51                 sum_u = sum_u + (S_u[i]*f[i]/sigma_u**2 - S_u_kaimal_i)**2
52                 sum_w = sum_w + (S_w[i]*f[i]/sigma_w**2 - S_w_kaimal_i)**2
53             sum_u = np.sqrt(sum_u/np.shape(f)[0])
54             sum_w = np.sqrt(sum_w/np.shape(f)[0])
55             lse_Au[cnt] = sum_u
56             lse_Aw[cnt] = sum_w
57             cnt = cnt + 1
58
59         # Finding optimal value for A_u

```



```

58     lse_Au_min_index = np.argmin(lse_Au)
59     lse_Au = lse_Au[lse_Au_min_index]
60     Au = A_val[lse_Au_min_index]
61     # Finding optimal value for A_w
62     lse_Aw_min_index = np.argmin(lse_Aw)
63     lse_Aw = lse_Aw[lse_Aw_min_index]
64     Aw = A_val[lse_Aw_min_index]
65
66     return Au, Aw, lse_Au, lse_Aw
67
68
69
70 """ Calculation of decay coefficients for the normalised cross-spectra
71 def normCrossSpectraParam(Fs, u, w, dx):
72     """
73     Returns the decay coefficients of the normalised cross-spectra of u and w for one single
74     10-min recording and chosen anemometer pairs.
75     Arguments:
76     Fs      : sampling frequency
77     u, w    : matrices with along-wind (with mean wind) and vertical velocity data
78               from closely spaced anemometers in each column. The first column is
79               the reference anemometer; the normalised cross-spectra are calculated
80               between this anemometer and the others.
81     dx      : vector containing the horizontal distances between the anemometers
82               in the anemometer pairs
83     Returns:
84     K_u,w    : decay coefficients for the normalized cross spectra
85     lse_Ku,w : least square error for K_u,w. Set to zero if fitting is not computed.
86     -----
87     Function returns the fitted decay coefficients of the normalised cross-spectra of u and
88     w for one single 10 min recording and chosen anemometer pairs. The fit is done by using
89     least square error. The expression that is fitted is Davenport's expression(Davenport,
90     1961).
91     """
92     # Parameters from N400
93     Ku = 10.0
94     Kw = 6.5
95     # Parameters needed to estimate Cuu and Cww with Welch method
96     Nwelch = 8
97     Nwindow = np.floor(np.max(np.shape(u))/Nwelch) # Length of window
98     f = sci.csd(u[:,0], u[:,0], Fs, nperseg = Nwindow)[0] # Frequency axis
99     # Estimation of normalised cross-spectra and reduced frequency for each anemometer pair
100    Cuu = np.zeros([np.shape(f)[0], np.shape(u)[1]-1])
101    Cww = np.zeros([np.shape(f)[0], np.shape(u)[1]-1])
102    f_red = np.zeros([np.shape(f)[0], np.shape(u)[1]-1])
103    for i in range(np.shape(u)[1]-1):
104        Suu = sci.csd(u[:,0]-np.mean(u[:,0]), u[:,i+1]-np.mean(u[:,i+1]), Fs, nperseg =
105        Nwindow)[1] # Cross-spectrum
106        Su1 = sci.csd(u[:,0]-np.mean(u[:,0]), u[:,0]-np.mean(u[:,0]), Fs, nperseg = Nwindow)

```

```

102     [1] # Auto-spectrum of u1
        Su2 = sci.csd(u[:,i+1]-np.mean(u[:,i+1]), u[:,i+1]-np.mean(u[:,i+1]), Fs, nperseg =
Nwindow)[1] # Auto-spectrum of u2
103     Cuc[:,i] = Suu / np.sqrt(Su1*Su2)
104
105     Sww = sci.csd(w[:,0]-np.mean(w[:,0]), w[:,i+1]-np.mean(w[:,i+1]), Fs, nperseg =
Nwindow)[1] # Cross spectrum
106     Sw1 = sci.csd(w[:,0]-np.mean(w[:,0]), w[:,0]-np.mean(w[:,0]), Fs, nperseg = Nwindow)
    [1] # Auto-spectrum of w1
107     Sw2 = sci.csd(w[:,i+1]-np.mean(w[:,i+1]), w[:,i+1]-np.mean(w[:,i+1]), Fs, nperseg =
Nwindow)[1] # Auto-spectrum of w2
108     Cww[:,i] = Sww / np.sqrt(Sw1*Sw2)
109
110     f_red[:,i] = f * dx[i]/(np.mean([np.mean(u[:,0]), np.mean(u[:,i+1])])) # Reduced
frequency
111     ### Improvement of K_u,w by least square fitting
112     K_val = np.arange(1,25.1,0.1) # Vector of K_i values
113     lse_Ku = np.zeros(np.shape(K_val)[0]) # Vector of least square errors for K_u
114     lse_Kw = np.zeros(np.shape(K_val)[0]) # Vector of least square errors for K_w
115
116     cnt = 0
117     for K_i in K_val:
118         sum_u = 0
119         sum_w = 0
120         for p in range(np.shape(dx)[0]):
121             for i in range(np.shape(f)[0]):
122                 C_i = np.exp(-K_i*f_red[i, p])
123                 sum_u = sum_u + (Cuc[i, p] - C_i)**2
124                 sum_w = sum_w + (Cww[i, p] - C_i)**2
125             sum_u = np.sqrt(sum_u/(np.shape(f)[0]*np.shape(dx)[0]))
126             sum_w = np.sqrt(sum_w/(np.shape(f)[0]*np.shape(dx)[0]))
127             lse_Ku[cnt] = sum_u
128             lse_Kw[cnt] = sum_w
129             cnt = cnt + 1
130
131     # Finding optimal value for K_u
132     lse_Ku_min_index = np.argmin(lse_Ku)
133     lse_Ku = lse_Ku[lse_Ku_min_index]
134     Ku = K_val[lse_Ku_min_index]
135     # Finding optimal value for K_w
136     lse_Kw_min_index = np.argmin(lse_Kw)
137     lse_Kw = lse_Kw[lse_Kw_min_index]
138     Kw = K_val[lse_Kw_min_index]
139
140     return Ku, Kw, lse_Ku, lse_Kw

```

master_script.py

```

1 %% Import necessary modules and packages

```

```

2 from nptdms import TdmsFile
3 import os
4 import time_synchronisation as ts
5 import get_data as gd
6 import fitting as fit
7 ### Necessary paths
8 path = 'H:'
9 nodes_folder = ['\\anode003', '\\anode004', '\\anode005', '\\anode006', '\\anode007', '\\
    anode008', '\\anode009', '\\anode010']
10 save_path = os.getcwd() + '\\Results'
11 ### INFO
12 acc_nodes = ['A03', 'A04', 'A05', 'A06', 'A07', 'A08', 'A09', 'A10']
13 wind_nodes = ['W03', 'W04', 'W05', 'xx', 'W07', 'xx', 'xx', 'W10']
14 anemometers = ['W03-7-1', 'W04-15-1', 'W05-17-1', 'W05-18-1', 'W05-19-1', 'W05-19-2', 'W07
    -28-1', 'W10-45-1', 'W10-47-1', 'W10-49-1']
15 acc_datasets = ['-1x', '-1y', '-1z', '-2x', '-2y', '-2z']
16 Fs = 2 # The final frequency of the data
17 ### Extracting data from midspan
18 folder = path + nodes_folder[4] # anode007
19 acc_node = acc_nodes[4] # W07
20 anemometer = anemometers[6] # W07-28-1
21
22 filenames = []
23 for file in os.listdir(folder):
24     if (file.endswith('.tdms') and file.startswith('2022')):
25         filenames.append(file)
26
27 for i in range(len(filenames)):
28     try:
29         tdms_file = TdmsFile.read(folder+'\\'+filenames[i])
30     except:
31         print('Error opening the file ' + filenames[i])
32         continue
33     else:
34         print('File number: ' + str(i+1) + ' read')
35         #Computing master time vector
36         t_master = ts.time_master([tdms_file], [wind_nodes[4]])
37         #Extracting data
38         V, meandir, u, w, Iu, Iw, sigma_y, sigma_z, sigma_theta = gd.getCharacteristics(
tdms_file, acc_node, anemometer, t_master, Fs)
39         print('Finished extracting data from file ' + str(i+1))
40         # Deleting intervals with mean wind speed below 3 m/s
41         ind = np.sort(np.argwhere(V < 3))
42         V = np.delete(V, ind); meandir = np.delete(meandir, ind); u = np.delete(u, ind, axis
=1); w = np.delete(w, ind, axis=1)
43         Iu = np.delete(Iu, ind); Iw = np.delete(Iw, ind); sigma_y = np.delete(sigma_y, ind);
sigma_z = np.delete(sigma_z, ind); sigma_theta = np.delete(sigma_theta, ind)
44         # Obtaining the spectral values Au and Aw for the intervals
45         Au = np.zeros(len(V))

```

```

46     Aw = np.zeros(len(V))
47     lse_Au = np.zeros(len(V))
48     lse_Aw = np.zeros(len(V))
49     for k in range(len(V)):
50         print('Calculating A for interval nr. ' + str(k+1))
51         Au[k], Aw[k], lse_Au[k], lse_Aw[k] = fit.autoSpectraParam(Fs, u[:,k], w[:,k], V[
k], True)
52     tdms_file.close()
53     # Saving the extracted data
54     variables = [V, meandir, Iu, Iw, sigma_y, sigma_z, sigma_theta, Au, Aw, lse_Au,
lse_Aw]
55     file_data = np.zeros((len(V), len(variables)))
56     for k in range(np.shape(file_data)[1]):
57         file_data[:,k] = variables[k]
58     turbulence = [u, w]
59     turb_data = np.zeros((np.shape(u)[0], np.shape(u)[1], 2))
60     for k in range(np.shape(turb_data)[2]):
61         turb_data[:,:, k] = turbulence[k]
62
63     np.savetxt(save_path + nodes_folder[4] + '\\\\' + filenames[i][0:20] + '.csv',
file_data, delimiter=',')
64     np.savetxt(save_path + nodes_folder[4] + '\\\\' + 'Turbulence' + '\\\\' + filenames[i
][0:20] + '_u.csv', turb_data[:,:,0], delimiter=',')
65     np.savetxt(save_path + nodes_folder[4] + '\\\\' + 'Turbulence' + '\\\\' + filenames[i
][0:20] + '_w.csv', turb_data[:,:,1], delimiter=',')
66     %% Extract K_u,w from fitting of data from anemometers W05-19-1, W05-18-1, W05-17-1 and W04
-15-1
67     dx = np.array([20, 40, 80])           # Distances between anemometers
68
69     folder1 = path + nodes_folder[1]
70     fileNames_A04 = []
71     for file in os.listdir(folder1):
72         if (file.endswith('.tdms') and file.startswith('2022')):
73             fileNames_A04.append(file)
74
75     folder2 = path + nodes_folder[2]
76     fileNames_A05 = []
77     for file in os.listdir(folder2):
78         if (file.endswith('.tdms') and file.startswith('2022')):
79             fileNames_A05.append(file)
80
81     fileNames_inBoth = list(set(fileNames_A04).intersection(fileNames_A05))
82     fileNames_inBoth.sort()
83
84     for file in fileNames_inBoth: # Loop through all files
85         print('Reading file ' + file)
86         try:
87             cur_file_A04 = TdmsFile.read(path + nodes_folder[1] + '\\\\' + file)
88             cur_file_A05 = TdmsFile.read(path + nodes_folder[2] + '\\\\' + file)

```

```

89     except:
90         print('Error opening the file ' + file)
91         continue
92     else:
93         #Computing master timevector
94         t_master = ts.time_master([cur_file_A04, cur_file_A05], ['W04', 'W05'])
95         # Extracting data from anemometers
96         u19, w19, V19, meandir19 = gd.readWind('W05-19-1', cur_file_A05, 10, t_master, Fs)
97         u18, w18, V18, meandir18 = gd.readWind('W05-18-1', cur_file_A05, 10, t_master, Fs)
98         u17, w17, V17, meandir17 = gd.readWind('W05-17-1', cur_file_A05, 10, t_master, Fs)
99         u15, w15, V15, meandir15 = gd.readWind('W04-15-1', cur_file_A04, 10, t_master, Fs)
100        # Deleting intervals that have mean direction deviating more than 20 degrees from 90
        and 270 and/or mean wind speed below 3 m/s
101        ind = []
102        dum1 = np.argwhere(np.logical_or((np.logical_or(meandir19 < 70, meandir19 > 290)),
        np.logical_and(meandir19 < 250, meandir19 > 110)))
103        dum2 = np.argwhere(np.logical_or((np.logical_or(meandir18 < 70, meandir18 > 290)),
        np.logical_and(meandir18 < 250, meandir18 > 110)))
104        dum3 = np.argwhere(np.logical_or((np.logical_or(meandir17 < 70, meandir17 > 290)),
        np.logical_and(meandir17 < 250, meandir17 > 110)))
105        dum4 = np.argwhere(np.logical_or((np.logical_or(meandir15 < 70, meandir15 > 290)),
        np.logical_and(meandir15 < 250, meandir15 > 110)))
106        dum5 = np.argwhere(np.logical_or(np.logical_or(np.logical_or(V19 < 3, V18 < 3), V17
        < 3), V15 < 3))
107        ind = np.concatenate((np.concatenate((np.concatenate((np.concatenate((dum1, dum2)),
        dum3)), dum4)), dum5))
108        ind = np.sort((list(set(ind.flatten()))))
109        if len(ind)!=0:
110            u19 = np.delete(u19, ind, axis=1); V19 = np.delete(V19, ind); u18 = np.delete(
        u18, ind, axis=1); V18 = np.delete(V18, ind); u17 = np.delete(u17, ind, axis=1); V17 =
        np.delete(V17, ind); u15 = np.delete(u15, ind, axis=1); V15 = np.delete(V15, ind)
111            w19 = np.delete(w19, ind, axis=1); w18 = np.delete(w18, ind, axis=1); w17 = np.
        delete(w17, ind, axis=1); w15 = np.delete(w15, ind, axis=1)
112        # Obtaining Ku and Kw for intervals
113        Ku = np.zeros(np.shape(u19)[1]); lse_Ku = np.zeros(np.shape(u19)[1])
114        Kw = np.zeros(np.shape(u19)[1]); lse_Kw = np.zeros(np.shape(u19)[1])
115        for interval in range(np.shape(u19)[1]):
116            print('Calculating K for interval nr. ' + str(interval+1))
117            u = np.array([u19[:, interval] + V19[interval],
118                        u18[:, interval] + V18[interval],
119                        u17[:, interval] + V17[interval],
120                        u15[:, interval] + V15[interval]]).T
121            w = np.array([w19[:, interval], w18[:, interval], w17[:, interval], w15[:,
        interval]]).T
122            Ku[interval], Kw[interval], lse_Ku[interval], lse_Kw[interval] = fit.
        normCrossSpectraParam(Fs, u, w, dx)
123            cur_file_A04.close()
124            cur_file_A05.close()
125        # Saving the extracted values

```

```

126     variables = [Ku, Kw, lse_Ku, lse_Kw, np.mean(np.array([V19, V18, V17, V15]), axis=0)
127 ]
128     file_data = np.zeros((len(Ku), len(variables)))
129     for k in range(np.shape(file_data)[1]):
130         file_data[:,k] = variables[k]
131     np.savetxt(save_path + '\\K\\' + file[0:20] + '.csv', file_data, delimiter=',')

```

buffeting_response.py

```

1  %% Import necessary modules
2  from scipy.interpolate import interp1d
3  import AbaqusInputToBuffetingResponse as inp
4  from csv import writer
5  import os
6  import numpy as np
7  def RMS(U, I_u, I_w, Au, Aw, Ku, Kw, x_r, spectra_type, filename):
8      """
9      Arguments:
10     U : mean wind velocity
11     I_u,w      : turbulence intensity of along-wind and vertical turbulence component
12     A_u,w      : spectral parameters for the one-point auto spectra
13     K_u,w      : decay coefficients for the normalized cross spectra
14     x_r        : index in x-vector for position along the bridge
15     spectra_type : 'kaimal' / 'N400'
16     filename   : name of the file that the outputs should be saved to
17     Returns:
18     rms_tot : vector of root mean square of acceleration response for y, z and theta
19     ms      : array of mean square of acceleration response for y, z and theta,
20               from each mode
21     -----
22     Function calculates the buffeting response using either a kaimal type of spectra, or the
23     N400 spectra.
24     """
25     ##### Constants / parameters
26     rho = 1.25 # Air density
27     B = 18.6 # Width of bridge cross section
28     D = 3 # Height of bridge cross section
29     # Calculation of integral length scale
30     L1 = 100 # Reference length scale
31     z1 = 10 # Reference height
32     z = 54.2 # height of anemometer
33     xLu = L1 * (z/z1)**0.3
34     xLw = 1/12 * xLu
35     if spectra_type == 'kaimal':
36         xLu = z
37         xLw = z
38     # Retrieving modal parameters from Abaqus
39     nodes, phi, w_n, Mi_t = inp.modalParam(numberOfModes=56)
40     phi = np.transpose(phi, [0,2,1])

```

```

40 #Interpolation of nodal coordinates and mode shapes
41 L = np.max(nodes) - np.min(nodes)
42 x = np.linspace(np.ceil(-L/2), np.floor(L/2), 281) # Coordinates
43 phi = np.array(interp1d(nodes, phi, axis=1)(x))
44
45 ksi = 0.005 # Damping ratio for each mode, set to 0.5%
46 omega = np.logspace(-3, 0, 100)*2*np.pi # Vector of frequencies [rad/s]
47
48 Cd_bar = 0.793; Cl_bar = -0.353; Cm_bar = -0.0149; Cd_prime = -1.0842; Cl_prime = 3.420;
49 Cm_prime = 1.0625 # From wind tunnel test at NTNU
50 Bq = rho*U*B/2 * np.array([[2*(D/B)*Cd_bar, (D/B)*Cd_prime-Cl_bar], [2*Cl_bar, Cl_prime
51 +(D/B)*Cd_bar], [2*B*Cm_bar, B*Cm_prime]])
52
53 sigma_u = I_u * U
54 sigma_w = I_w * U
55
56 ##### Calculations #####
57 # Calculating the spectral density of the generalized load through a double for-loop
58 print('(1) Calculation of spectral density for the generalized loads.')
59 Su = 1/(2*np.pi) * sigma_u**2 * xLu / U * Au / (1 + 1.5 * Au * (omega*xLu/(2*np.pi*U)))
60 ** (5/3) # One-point auto-spectrum for u
61 Sw = 1/(2*np.pi) * sigma_w**2 * xLw / U * Aw / (1 + 1.5 * Aw * (omega*xLw/(2*np.pi*U)))
62 ** (5/3) # One-point auto-spectrum for w
63 S_Q_red = np.zeros([np.shape(omega)[0], np.shape(w_n)[0]])
64 dxdx = np.abs(np.array([x]) - np.array([x]).T)
65 for i in range(np.shape(w_n)[0]):
66     print('    Calculating for mode nr. ' + str(i+1) + ' out of ' + str(np.shape(w_n)
67 [0]))
68     for w in range(np.shape(omega)[0]):
69         Suu = Su[w] * np.exp(-Ku*omega[w]*dxdx/(2*np.pi*U))
70         Sww = Sw[w] * np.exp(-Kw*omega[w]*dxdx/(2*np.pi*U))
71         Int_1 = (np.array([phi[0,:,i]].T @ np.array([phi[0,:,i]]) * Bq[0,0] + np.array
72 ([phi[1,:,i]].T @ np.array([phi[0,:,i]]) * Bq[1,0] + np.array([phi[2,:,i]].T @ np.
73 array([phi[0,:,i]]) * Bq[2,0]) * Suu * Bq[0,0] + \
74 (np.array([phi[0,:,i]].T @ np.array([phi[0,:,i]]) * Bq[0,1] + np.array
75 ([phi[1,:,i]].T @ np.array([phi[0,:,i]]) * Bq[1,1] + np.array([phi[2,:,i]].T @ np.
76 array([phi[0,:,i]]) * Bq[2,1]) * Sww * Bq[0,1]
77         Int_2 = (np.array([phi[0,:,i]].T @ np.array([phi[1,:,i]]) * Bq[0,0] + np.array
78 ([phi[1,:,i]].T @ np.array([phi[1,:,i]]) * Bq[1,0] + np.array([phi[2,:,i]].T @ np.
79 array([phi[1,:,i]]) * Bq[2,0]) * Suu * Bq[1,0] + \
80 (np.array([phi[0,:,i]].T @ np.array([phi[1,:,i]]) * Bq[0,1] + np.array
81 ([phi[1,:,i]].T @ np.array([phi[1,:,i]]) * Bq[1,1] + np.array([phi[2,:,i]].T @ np.
82 array([phi[1,:,i]]) * Bq[2,1]) * Sww * Bq[1,1]
83         Int_3 = (np.array([phi[0,:,i]].T @ np.array([phi[2,:,i]]) * Bq[0,0] + np.array
84 ([phi[1,:,i]].T @ np.array([phi[2,:,i]]) * Bq[1,0] + np.array([phi[2,:,i]].T @ np.
85 array([phi[2,:,i]]) * Bq[2,0]) * Suu * Bq[2,0] + \
86 (np.array([phi[0,:,i]].T @ np.array([phi[2,:,i]]) * Bq[0,1] + np.array
87 ([phi[1,:,i]].T @ np.array([phi[2,:,i]]) * Bq[1,1] + np.array([phi[2,:,i]].T @ np.
88 array([phi[2,:,i]]) * Bq[2,1]) * Sww * Bq[2,1]

```

```

72         Int = Int_1 + Int_2 + Int_3
73         S_Q_red[w,i] = np.trapz(np.trapz(Int,x), x)
74
75
76     ## Refining the w-axis for the generalized loads by interpolation
77     print('(2) Refining the frequency axis')
78     omega_ref = np.linspace(0.001, 1, 1000)*2*np.pi
79     S_Q = interp1d(omega, S_Q_red, axis=0)(omega_ref)
80
81     ## Calculating generalized damping and stiffness
82     print('(3) Calculating generalized damping and stiffness')
83     Ci_t = np.zeros(np.shape(w_n)[0]) # Generalized damping
84     Ki_t = np.zeros(np.shape(w_n)[0]) # Generalized stiffness
85     for i in range(np.shape(w_n)[0]):
86         Ci_t[i] = 2*Mi_t[i]*w_n[i]*ksi
87         Ki_t[i] = w_n[i]**2*Mi_t[i]
88
89     ## Calculating aerodynamic damping and stiffness
90     print('(4) Calculating aerodynamic damping and stiffness')
91     Cae = np.zeros([np.shape(omega_ref)[0], 3, 3]) # Aerodynamic damping
92     Kae = np.zeros([np.shape(omega_ref)[0], 3, 3]) # Aerodynamic stiffness
93     i = 0
94     for w in omega_ref:
95         K = w*B/U # Reduced frequency
96         v_hat = 1 / K
97         # Aerodynamic derivatives
98         if v_hat <= 1.35:
99             v_hat = 1.35
100        elif v_hat >= 17:
101            v_hat = 17
102        P4s = (-8.73239e-05*v_hat**2 + 0.00220737*v_hat + 0.0263768)/K**2
103        H4s = (-0.00143984*v_hat**2 + 0.0325336*v_hat + -0.145406)/K**2
104        A3s = (-0.000826889*v_hat**2 + 0.0192049*v_hat + 0.926908)/K**2
105        P1s = (0.00338829*v_hat**2 + -0.070592*v_hat + 0.00153207)/K
106        H1s = (0.00530354*v_hat**2 + -0.116064*v_hat + -2.19524)/K
107        A2s = (0.0017332*v_hat**2 + -0.0452137*v_hat + -0.157346)/K
108
109        Cae[i, :, :] = (rho*B**2/2)*w * np.array([[P1s, 0, 0], [0, H1s, 0], [0, 0, B**2*A2s
110        ]])
111        Kae[i, :, :] = (rho*B**2/2)*w**2 * np.array([[P4s, 0, 0], [0, H4s, 0], [0, 0, B**2*
112        A3s]])
113        i = i + 1
114
115     ## Calculating the response
116     print('(5) Calculating the response')
117     ms = np.zeros([3, np.shape(w_n)[0]]) # Mean square of acceleration response for y, z and
118     theta
119     # Looping through all frequencies
120     S_rdd = np.zeros([3, np.shape(omega_ref)[0]]) # Matrix with auto-spectra for acc. resp.

```



```

    in y, z, theta
118 # Looping through each mode
119 for i in range(np.shape(w_n)[0]):
120     print('    Calculating for mode nr. ' + str(i+1) + ' out of ' + str(np.shape(w_n)
[0]))
121     j = 0
122     for w in omega_ref:
123         Ci_aet = np.trapz(phi[0,:,i]**2*Cae[j,0,0] + phi[1,:,i]**2*Cae[j,1,1] + phi[2,:,
i]**2*Cae[j,2,2], x) # Generalized aerodynamic damping
124         Ki_aet = np.trapz(phi[0,:,i]**2*Kae[j,0,0] + phi[1,:,i]**2*Kae[j,1,1] + phi[2,:,
i]**2*Kae[j,2,2], x) # Generalized aerodynamic stiffness
125         Hi_t = 1/(-Mi_t[i]*w**2 + (Ci_t[i] - Ci_aet)*1j*w + (Ki_t[i] - Ki_aet)) #
Generalized frequency response function
126         Si_eta = np.abs(Hi_t)**2 * S_Q[j, i] # Auto-spectrum for generalized response
127         Si_r = phi[:, x_r, i]**2 * Si_eta # Auto-spectra for disp. resp. in y, z, theta
128         Si_rdd = w**4 * Si_r # Auto-spectra for acc. resp. in y, z, theta
129         S_rdd[:, j] = Si_rdd
130         j = j + 1
131         ms[0, i] = np.trapz(S_rdd[0, :], omega_ref)
132         ms[1, i] = np.trapz(S_rdd[1, :], omega_ref)
133         ms[2, i] = np.trapz(S_rdd[2, :], omega_ref)
134     rms_tot = np.sqrt(np.array([sum(ms[0]), sum(ms[1]), sum(ms[2])]))
135
136 # Saving results
137 save_path = os.getcwd() + '\\Results' + '\\Predicted'
138 file_data = np.array([U, rms_tot[0], rms_tot[1], rms_tot[2]]).T
139 with open(save_path + filename + '.csv', 'a', newline='') as file:
140     file_write = writer(file)
141     file_write.writerow(file_data)
142     file.close()
143 return rms_tot, ms

```

response_predictions.py

```

1 import buffering_response as br
2 import os
3 from csv import writer
4 ### Predicted buffeting response using semi-probabilistic approach
5 I_u_all = [0.0726, 0.0741, 0.0735]
6 I_w_all = [0.0456, 0.0408, 0.0346]
7 Au = 8.3
8 Aw = 1.9
9 Ku = 8.2
10 Kw = 8.6
11 # RUN the calculation
12 U = np.linspace(0.1, 25, 30) # Vector of mean wind velocities
13 U = np.concatenate((U, np.array[9.95, 10.05, 14.95, 15.05]))
14
15 filename = 'Case 1'

```

```

16
17 for i in range(np.shape(U)[0]):
18     if len(I_u_all) > 1:
19         if U[i] < 10:
20             I_u = I_u_all[0]
21             I_w = I_w_all[0]
22         elif U[i] >= 10 and U[i] < 15:
23             I_u = I_u_all[1]
24             I_w = I_w_all[1]
25         elif U[i] >= 15:
26             I_u = I_u_all[2]
27             I_w = I_w_all[2]
28     print('V = '+str(U[i]))
29     rms_tot, ms = br.RMS(U[i], I_u, I_w, Au, Aw, Ku, Kw, 140, spectra_type='kaimal',
        filename)
30 %% Predicted buffeting response using probabilistic approach
31 # Retrieving the simulated values of sigma_u and sigma_w
32 simulated = os.getcwd() + '\Results' + '\\Predicted\Simulations'
33 V_east = []; std_u_east = []; std_w_east = [];
34 V_west = []; std_u_west = []; std_w_west = [];
35 for file in os.listdir(simulated):
36     if (file.endswith('east.csv')):
37         res = np.loadtxt(simulated + '\\'+ file, delimiter=',')
38         for i in range(np.shape(res)[0]):
39             V_east.append(res[i][0])
40             std_u_east.append(res[i][1])
41             std_w_east.append(res[i][2])
42     if (file.endswith('west.csv')):
43         res = np.loadtxt(simulated + '\\'+ file, delimiter=',')
44         for i in range(np.shape(res)[0]):
45             V_west.append(res[i][0])
46             std_u_west.append(res[i][1])
47             std_w_west.append(res[i][2])
48
49 # Predicted response using probabilistic approach for easterly winds
50 Au = 13.9
51 Aw = 1.7
52 Ku = 7.8
53 Kw = 8.9
54 filename = 'sim_east'
55 for i in range(0, len(V_east)):
56     I_u = std_u_east[i]/V_east[i]
57     I_w = std_w_east[i]/V_east[i]
58     print('    Calculating for V_east nr. ' + str(i+1) + ' out of ' + str(np.shape(V_east)
        [0]))
59     rms_tot, ms = br.RMS(V_east[i], I_u, I_w, Au, Aw, Ku, Kw, 140, spectra_type='kaimal',
        filename)
60
61 # Predicted response using probabilistic approach for westerly winds

```

```

62 Au = 9.4
63 Aw = 1.8
64 Ku = 7.8
65 Kw = 8.9
66 filename = 'sim_west'
67 for i in range(np.shape(V_west)[0]):
68     I_u = std_u_west[i]/V_west[i]
69     I_w = std_w_west[i]/V_west[i]
70     print('    Calculating for V_west nr. ' + str(i+1) + ' out of ' + str(np.shape(V_west)
71           [0]))
71     rms_tot, ms = br.RMS(V_west[i], I_u, I_w, Au, Aw, Ku, Kw, 140, spectra_type='kaimal')

```

A.4 Operational Modal Analysis

OMA_functions.py

```

1 ##### Import modules
2 import koma.oma as oma
3 import koma.plot as oma_plot
4 import koma.modal as modal
5 import scipy.signal as sci
6
7 def modalParamOMA(data, fs, orders, i, s, stabcrit={'freq': 0.05, 'damping': 0.1, 'mac':
8     0.1}, autoSpectrum=False):
9     """
10    Arguments:
11    data          : data matrix, each column containing data from one accelerometer
12    fs            : sampling frequency
13    orders        : array of what orders to perform the Cov-SSI for
14    i             : maximum number of block rows
15    s             : stability level
16    stabcrit      : stabilization criteria
17    autospectrum : parameter to decide whether or not to plot an auto-spectrum in the
18                  stabilization plot
19    Returns:
20    f_n_sort, ksi_sort, phi_sort : natural frequencies, damping ratios and mode
21                                  shapes from stable poles, sorted based on the values of
22                                  the
23                                  natural frequencies
24    -----
25    Function for computing Cov-SSI to obtain modal parameters from stable poles
26    """
27    ## Find all complex poles and eigenvectors for all orders
28    lambd, phi = oma.covssi(data=data, fs=fs, i=i, orders=orders, weighting='none',
29        matrix_type='hankel', algorithm='shift', showinfo=True, balance=True)
30    ## Find stable poles from all poles
31    lambd_stab, phi_stab, orders_stab, idx_stab = oma.find_stable_poles(lambd, phi, orders,
32        s, stabcrit=stabcrit, valid_range={'freq': [0, np.inf], 'damping': [0, np.inf]}),

```

```

    indicator='freq', return_both_conjugates=False)
28  ## Natural frequencies, damping ratios and mode shapes for stable poles, sorted after
    the value of the natural frequencies
29  w_n = np.abs(lambd_stab)
30  f_n = w_n / (2*np.pi)
31  ksi = - np.real(lambd_stab) / np.abs(lambd_stab)
32
33  f_n_sort_index = np.argsort(f_n)
34  f_n_sort = f_n[f_n_sort_index]
35  ksi_sort = ksi[f_n_sort_index]
36  phi_sort = phi_stab[:, f_n_sort_index]
37
38  return f_n_sort, ksi_sort, phi_sort
39
40 def getModeShapeInSpecifiedDirection(phi, i_phi_plot, direction):
41     """
42     Arguments:
43     phi          : mode shape matrix, each column containing mode shape values
44                   for x1, x2, y1, y2, z1, z2 at the sensor locations
45     i_phi_plot   : index of which mode shape to plot
46     direction    : type of mode shape : 'x' / 'y' / 'z' / 'theta'
47     Returns:
48     phi          : normalized mode shape values in specified direction
49     -----
50     Function normalises the mode shape matrix in order to plot the mode shapes in a
    specified direction.
51     """
52     B = 18.6
53     if direction == 'x':
54         norm, _ = modal.normalize_phi(np.real(modal.maxreal((phi[0:8,:]+phi[8:16,:])/2)))
55         phi = norm[:, i_phi_plot]
56     elif direction == 'y':
57         norm, _ = modal.normalize_phi(np.real(modal.maxreal((phi[16:24,:]+phi[24:32,:])/2)))
58         phi = norm[:, i_phi_plot]
59     elif direction == 'z':
60         norm, _ = modal.normalize_phi(np.real(modal.maxreal((phi[32:40,:]+phi[40:48,:])/2)))
61         phi = norm[:, i_phi_plot]
62     else:
63         norm, _ = modal.normalize_phi(np.real(modal.maxreal((-phi[32:40,:]+phi[40:48,:])/B))
64         )
65         phi = norm[:, i_phi_plot]
66     return phi

```

OMA_Halogaland.py

```

1
2 import numpy as np
3 import os

```

```

4 from nptdms import TdmsFile
5 import get_acc_OMA as gd
6 import OMA_functions as OMA
7 import time_synchronisation as ts
8 ### Read corresponding file from all data loggers
9 path = os.getcwd()+'\Data'
10 anodes = ['\\anode003', '\\anode004', '\\anode005', '\\anode006', '\\anode007', '\\anode008',
           '\\anode009', '\\anode010'] # List of all data loggers
11 anode_names = ['A03', 'A04', 'A05', 'A06', 'A07', 'A08', 'A09', 'A10']
12 anemometers = ['W03-7-1', 'W04-15-1', 'W05-19-1', 'W07-28-1', 'W10-49-1'] # List of
           anemometer names for each data logger (the ones closest to accelerometers)
13
14 fileToRead = '2022-02-04-00-00-00Z.tdms'
15 files = []
16 for i in range(np.shape(anodes)[0]):
17     files.append(TdmsFile.read(path + anodes[i] + '\\\ ' + fileToRead))
18     print('File nr. ' + str(i+1) + ' done.')
19 ### Extracting chosen interval of processed data from all accelerometers
20 interval = 0
21 fs = 2
22 t_master = ts.time_master(files, ['W03', 'W04', 'W05', 'xx', 'W07', 'xx', 'xx', 'W10'])
23 # Finding shape to initialize matrices
24 _, acc_1y, _, _, _ = gd.readAccForOMA('A03', files[0], 30, t_master, fs)
25
26 # Saving accelerations from interval nr. interval from all accelerometers
27 acc_x_all = np.zeros([np.shape(acc_1y)[1], np.shape(files)[0]*2])
28 acc_y_all = np.zeros([np.shape(acc_1y)[1], np.shape(files)[0]*2])
29 acc_z_all = np.zeros([np.shape(acc_1y)[1], np.shape(files)[0]*2])
30 for i in range(np.shape(files)[0]):
31     acc = gd.readAccForOMA(anode_names[i], files[i], 30, t_master, fs)
32     acc_x_all[:, i] = acc[0][interval, :]; acc_x_all[:, i+8] = acc[3][interval, :]
33     acc_y_all[:, i] = acc[1][interval, :]; acc_y_all[:, i+8] = acc[4][interval, :]
34     acc_z_all[:, i] = acc[2][interval, :]; acc_z_all[:, i+8] = acc[5][interval, :]
35
36 acc_all = np.concatenate((acc_x_all, acc_y_all, acc_z_all), axis=1) # [1x values; 2x values;
           1y values; 2y values; 1z values; 2z values]
37 ### Cov-SSI - parameters
38 i = 24 # Maximum number of block rows
39 s = 6 # Stability level
40 orders = np.arange(2, 252, 2) # Array of what orders to include
41 stabcrit = {'freq': 0.05, 'damping': 0.1, 'mac': 0.1} # Default
42
43 f_n_sort, ksi_sort, phi_sort, fig = OMA.modalParamOMA(acc_all, fs, orders, i, s, stabcrit=
           stabcrit, autoSpectrum=True)

```

