NTNU – Trondheim
Norwegian University of
Science and Technology

Damage Identification of an Offshore Wind Turbine Jacket Support Structure

Simon Adrian Brauer

Wind Energy
Submission date: July 2014
Supervisor: Torgeir Moan, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Master's Thesis

Damage Identification of an Offshore Wind Turbine Jacket Support Structure

Simon A. Brauer

27th July 2014

Committee:
Prof. Dr. A. V. Metrikine, Delft University of Technology
Prof. Dr. T. Moan, Norwegian University of Science and Technology
Dr. E. Lourens, Delft University of Technology
Dr. A. Romeijn, Delft University of Technology
C. Keijdener, Delft University of Technology
L. van Pijkeren, Ballast Nedam Beheer

Abstract

Due to the variable and stochastic offshore environment, offshore wind turbines are prone to fatigue. Fatigue damage usually occurs as cracks and is an important indicator of the structure's condition and the remaining life-time. However, the detection of fatigue cracks today is very cumbersome and easier and more straight-forward methods are desirable.

In this thesis, it is investigated whether fatigue damage can be identified or detected based on operational vibrations. The idea is that sensors like accelerometers are placed at the offshore wind turbine, for example at the relatively easily accessible tower, to monitor the structure's response over time. Changes of the dynamic characteristics over time might then be an indicator for structural damage.

To this end, an offshore wind turbine jacket support structure model is developed in Matlab by means of the finite element method. Realistic operational conditions are assumed and loads from wind and waves calculated accordingly. Models to represent a crack in structural members are reviewed and suitable methods are combined and adapted to fit the present model. To account for measurement and model errors, a Kalman filter is implemented that gives an optimal estimate for the structure's response. The response is then calculated for an undamaged support structure model and a support structure model, where at a certain location a crack has been simulated. The severity of the damage is varied and the effects are investigated.

In an eigenvalue analysis it is found that the effect of the damage on the eigenfrequencies is small. Higher local brace modes at the crack location are affected most, however in a small order of magnitude. Based on ambient vibrations due to wind and waves the response of the support structure is investigated. In an attempt to amplify the deviation between the undamaged and the damaged support structure, a forced harmonic vibration is introduced into the system.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

By the end of 2013 a bit more than 6.5 GW of offshore wind capacity has been installed in European waters, of which 1.5 GW were installed in 2013 alone [1]. With a goal of 20 % share of renewable energy in the European Union by 2020 [2], further growth is expected as offshore wind energy plays a major role in many countries' future plans to avoid CO_2 emissions and meet climate targets.

Most offshore wind farms so far have been built in rather shallow water (2–20 m) and mainly with monopile foundations. Recent developments proceed to greater water depths around 40 m, e.g. the Global Tech 1 offshore wind farm in Germany, which is currently under construction. The Beatrice demonstration project in the United Kingdom built a few years ago is an early example of the deployment of offshore wind energy in larger water depth. The combination of larger water depth and the trend to bigger wind turbines demands stiffer support structures. Space frame structures like jacket support structures provide sufficient stiffness and comparably small weight at the same time, so that their use is regarded as more cost-effective under these conditions.

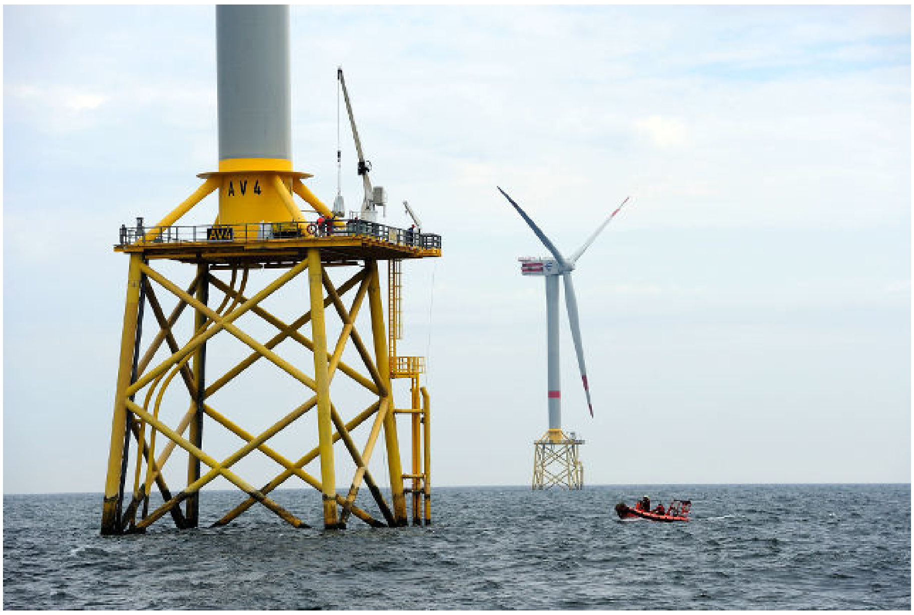

Figure 1.1: Jacket support structures in the Alpha Ventus offshore wind farm [3]

Jacket support structures are composed of tubular slender elements that are welded at intersections. The stochastic nature of wind and waves acting on the turbine and the support structure cause a large number of stress cycles. Moreover, the operation of the wind turbine induces further cyclic loads in the support structure. Stress raising effects make tubular joints the most critical points with respect to fatigue strength in the jacket structure and their design a crucial and challenging task in the planning stage. However, fatigue degradation and thus cracks cannot always be avoided. In order to guarantee the structural integrity and to predict the remaining fatigue life, inspections have to be carried out.

A major focus of the industry, the European Union and the national governments is to reduce the cost of energy in offshore wind. The investment in an offshore wind farm is divided into capital expenditures (CAPEX) and operational expenditures (OPEX). The CAPEX are one-time expenditures and comprise the cost for planning, manufacturing and installing the offshore wind farm, whereas the OPEX are either one-time or recurring expenditures occurring for operating the wind farm, e.g. inspection, maintenance and repair. The goal is to balance the requirements and cost from on the one hand inspection, maintenance and repair and on the other hand from design point of view. The designer would like the structure to be light and slender and thus cheaper, whereas from an operational point of view the design should possibly be robust and heavier in order to save money on inspections and repairs during the operational life.

1.2 State of the Art in Operation and Maintenance

As operation and maintenance comprises roughly 25 % of the life-time cost of an offshore wind farm, ways to reduce the OPEX are searched for. Operation and maintenance planning and activities are framed by statutory requirements, the available inspection techniques and the associated costs. Operations and maintenance of an offshore wind turbine support structure consists of many different scopes. Here, the focus is narrowed down to the detection of cracks in the support structure. Innovative methods that are able to detect damage at the structure with low effort will help to reduce cost.

1.2.1 Maintenance Principles

In general, different maintenance principles can be distinguished: Corrective maintenance, preventive maintenance and predictive maintenance (see Figure 1.2).

Corrective or reactive maintenance is also known as repair. It is performed when a failure or damage occurs that needs to be rectified. The maintenance is executed on demand (whenever a failure occurs) or batch-wise. Often repair is considered more expensive than preventing a failure. Preventive maintenance describes the maintenance activities that are performed to prevent a failure from happening. It comprises routine inspections and replacements of fragile components. A more sophisticated maintenance type is predictive maintenance. As the name indicates it refers to predicting necessary maintenance activities, thus minimising unscheduled maintenance, which is very expensive, and optimising scheduled maintenance by knowledge about the condition

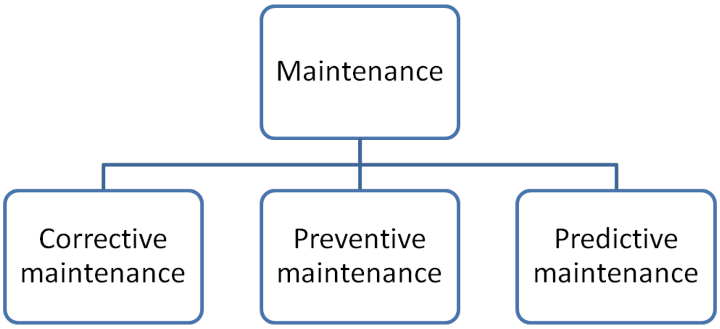

Figure 1.2: Breakdown of maintenance principles

of the asset. Predictive maintenance requires some sort of condition monitoring. For an offshore wind turbine this is preferably done remotely by means of sensors or other measurement devices that monitor certain operational parameters. For example, accelerometers can be used to measure accelerations as is already done nowadays in many offshore wind turbine towers.

1.2.2 Inspection Methods for Fatigue Cracks

Inspection for fatigue cracks, especially under water, is a difficult and expensive task. Figure 1.3 shows a snapshot taken by a remotely operated vehicle (ROV).

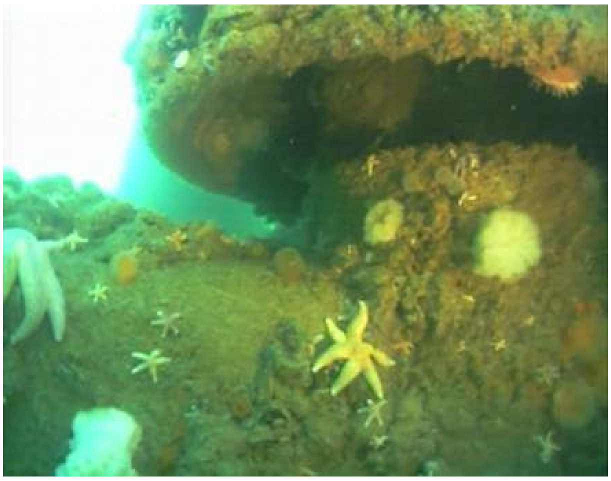

Figure 1.3: Snapshot of a boat-landing flange by an remote operating vehicle [4]

It is obvious that the detection of fatigue cracks in this environment is not easily accomplishable. The parts of the structure to inspect are usually covered with marine growth and marine life. Visibility in water may be bad. Thus, a visual inspection (by eye in case of a diver or by camera in case of a ROV) is limited to qualitative statements about e.g. gross damage of the structure, coating protection, sacrificial anodes or similar. For the detection of cracks, non-destructive techniques (NDT) are commonly employed, where the tested parts are not harmed. Different techniques with different accuracy level and complexity exist. Whereas they might be suitable to detect very small cracks, they all have the common disadvantage of being time consuming and expensive. For the inspection it is necessary to employ a diving team, to access the structure, which may lie remotely offshore, clean the part of the structure to be tested and perform the testing in adverse conditions. Since the testing itself is a very local process, the procedure has to be repeated for other areas of the structure or even for other structures.

Figure 1.4: Diver performing non-destructive testing [5]

1.3 Objective

Today's methods to detect damage in offshore wind turbine support structures could be classified as preventive maintenance and are very cumbersome and inefficient. A cheaper and reliable method is desired. Recent trends in operations and maintenance show a shift from preventive or even corrective maintenance to predictive maintenance.

Remote monitoring is relatively cheap and easy to perform because the structure does not need to be accessed and thus a lot of time and effort can be saved. Sensors that transmit their data to a remote onshore location are attached to the structure at different places. The data is then used to draw conclusions on the structure's condition. The main benefit would be that the structure is only visited when it is necessary, thereby reducing cost.

This work's main objective is to investigate whether this remote monitoring principle is viable for the detection of damage or cracks in an complex offshore wind turbine jacket support structure, especially underwater. To this end, two two-dimensional simplified offshore wind turbine jacket support structure are modelled. A crack is simulated in one of the support structures. Accelerometers are placed at the wind turbine tower at different heights to measure the accelerations of the structure due to ambient excitation by wind and waves. The dynamic responses of the undamaged and the damaged support structure are compared for visible differences that allow to detect damage in the support structure.

1.4 Reading Guide

After this introduction, the support structure model is presented and explained in Chapter 2. The UpWind jacket is taken as reference and simplified to a two-dimensional model. The representation of the wind turbine and the soil-structure interaction are defined.

In Chapter 3 concepts to simulated cracks in beams are reviewed and suitable methods are combined to give a proper crack model.

Chapter 4 includes the description of the ambient environment. Aerodynamic and hydrodynamic loads are defined and the implementation in the model is shown.

In Chapter 5 state-space representation is introduced to solve the system and to incorporate the Kalman filter.

In Chapter 6 the results of the investigations are presented. An eigenvalue analysis, ambient vibration testing and forced vibration testing is performed in order to identify changes in the modal characteristics due to the damage.

2 Support Structure Modelling

2.1 Introduction

To investigate the dynamic behaviour of an offshore wind turbine with a jacket support structure in the damaged and undamaged condition, a model for the calculation has to be developed. As reference, the jacket support structure developed in the UpWind project is chosen. For this work the reference jacket is simplified to a two-dimensional offshore wind turbine jacket support structure. Only the primary steel structure is considered, secondary steel appurtenances and other effects like marine growth are neglected. The support structure is modelled by means of the finite element method, where the braces and legs of the jacket structure and the turbine tower are modelled by plane-frame elements.

The offshore wind turbine itself, thus the rotor-nacelle assembly, is modelled by a circular, stationary disc and a cuboid on top of the turbine tower. The size and the masses are taken from the NREL 5 MW turbine. The interaction of the wind flow with the moving rotor that occurs in reality, is thus neglected. Dynamic excitation characteristics of the turbine such as the 1P and 3P excitation[1] or multiples of them are thus not taken into account. For damage identification these higher-order excitation frequencies could be beneficial because they could excite higher local modes, where identification might be easier. However, including those effects would go beyond the scope of this thesis.

The soil-structure interaction is represented by additional stiffness matrices at the soil-structure interface.

2.2 Support Structure

2.2.1 Introduction

For the purpose of this investigation a simplified two-dimensional jacket support structure is developed. The term support structure refers here to the jacket support structure and the wind turbine tower. The UpWind reference jacket [6] is chosen as guideline regarding layout and geometrical dimensions. In the following simplifying assumptions allowing to transform the three-dimensional jacket support structure to two dimensions are presented.

[1] 1P and 3P excitation refer to frequency ranges that are excited by the rotation of the rotor and its interaction with the wind.

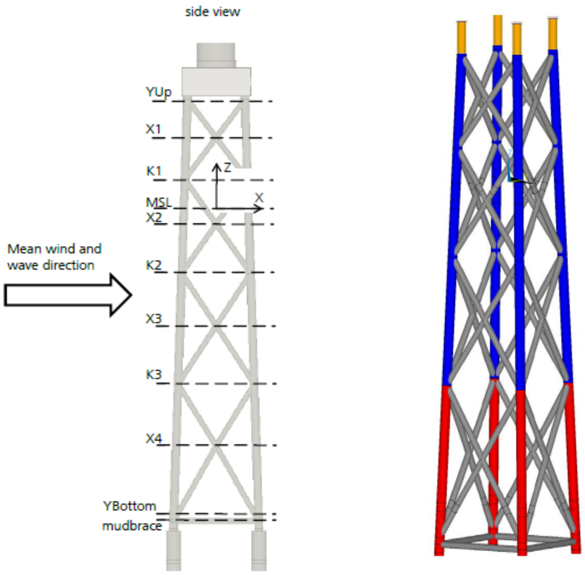

Figure 2.1: Side view (left) and isometric projection (right) of the UpWind jacket support structure [6]

The UpWind reference jacket is shown in Figure 2.1. The jacket support structure is designed for a water depth of 50 m, has a height of about 62 m from seabed to transition piece, a top width of 8 m and a bottom width of 12 m. The structure is composed of four legs connected by four bays with X-bracings made of tubular steel members. A cuboid concrete transition piece connects the jacket substructure with a 70 m high tubular steel tower with the wind turbine on top.

The support structure is founded in the soil by tubular steel piles with a length of 45 m, a diameter of 2.082 m and a wall thickness of 0.06 m.

In Appendix B the node coordinates, the element list and the member's properties are listed. The node coordinates are taken from the UpWind reference jacket, however only from one plane parallel to the mean wind and wave direction.

2.2.2 Finite Element Model

Introduction

The dynamic behaviour of the offshore wind jacket support structure shall be investigated and compared to a damaged support structure. To this end, the real structure has to be described mathematically.

Both beams and bars are one-dimensional structures that are much longer in one direction than the other two. Their behaviour can be described by the displacements (and hence velocities and accelerations) in time and along the longitudinal space coordinate only. Continuous theory takes into account the distributed material properties of real structures. In order to express one-dimensional continuous structures mathematically, partial differential equations are employed. In theory, the partial differential equations describe the behaviour of the structure at infinitely many points, which is why they could also be considered as systems with infinite number of degrees of freedom, as opposed to discrete systems with a finite number of degrees of freedom. Continuous systems are superior to discrete systems because the shape of the elastic system and thus the deformations and stresses are taken into account. However, for complex and large models continuous theory does not provide straight-forward solutions.

Fortunately, the finite element method is a good approximation to the closed-form solutions that the continuous theory provides. The finite element method uses a discretisation of the structure into single (finite) elements to reduce the complexity of the problem and the number of degrees of freedom from infinite to finite. The elements' continuous properties and external forces are lumped into the elements' nodes and the governing equations are established on element level. Subsequently, the element equations are merged to a global equation for the whole structure.

Mathematically, the finite element method dissolves the higher order partial differential equations used in continuous theory into second order ordinary differential equations in matrix form, the solution of which is straight-forward.

Plane Frame Element

The support structure's elements are represented by so-called plane frame elements. Two dimensional plane frame elements consist of one node with three degrees of freedom at each end. These degrees of freedom are translation in axial and transversal direction and rotation. For every element, element matrices are established. As the plane frame element is a combination of the bar and the beam element, these elements' matrices are presented first.

The element mass and stiffness matrix for a bar element are given in equations (2.1) and (2.2) and for a beam element in equations (2.3) and (2.4) [7]:

$$\mathbf{M}_{\text{bar}} = \frac{1}{6}\rho A L \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{2.1}$$

$$\mathbf{K}_{\text{bar}} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{2.2}$$

$$\mathbf{M}_{\text{beam}} = \frac{\rho A L}{420} \begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix} \tag{2.3}$$

$$\mathbf{K}_{\text{beam}} = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{2.4}$$

The derivation of these element matrices are described in Appendix A.

The beam and the bar element are combined to obtain the plane frame element that provides us with the characteristics of both the beam and the bar. To this end, the equation of motion is extended to six degrees of freedom as shown in equations (2.5) to (2.7), so that the mass and the stiffness matrix become 6 by 6 matrices, as every element has six degrees of freedom. The corresponding spots in the matrices are filled with the beam and the bar matrix entries and the rest is filled with zeros.

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.5}$$

$$\mathbf{u} = \begin{bmatrix} u_1 & w_1 & \theta_1 & u_2 & w_2 & \theta_2 \end{bmatrix}^{\mathrm{T}} \tag{2.6}$$

$$\ddot{\mathbf{u}} = \begin{bmatrix} \ddot{u}_1 & \ddot{w}_1 & \ddot{\theta}_1 & \ddot{u}_2 & \ddot{w}_2 & \ddot{\theta}_2 \end{bmatrix}^{\mathrm{T}} \tag{2.7}$$

The resulting mass matrix for the plane frame element is then obtained:

$$\mathbf{M} = \frac{\rho A L}{420} \begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2 \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2 \end{bmatrix} \tag{2.8}$$

Equivalently, the stiffness matrix is:

$$\mathbf{K} = \frac{EI}{L^3} \begin{bmatrix} \alpha & 0 & 0 & -\alpha & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ -\alpha & 0 & 0 & \alpha & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} \tag{2.9}$$

where

$$\alpha = \frac{AL^2}{I} \tag{2.10}$$

Transformation from Local to Global Coordinate System

Every element has its own local coordinate system, in which its element mass and stiffness matrices as well as the nodal force vector are defined. In a multi-member structure like the jacket support structure, members with different orientations occur. In order to assemble the element matrices to global system matrices, the element matrices first have to be transformed to the global coordinate system. In the following the global coordinate system is denoted with the coordinates x, z, θ and the local coordinate system with the coordinates $\bar{x}, \bar{z}, \bar{\theta}$.

The displacements of a two node (i,j) plane frame element in the local coordinate system can be expressed in the global coordinate system as follows

$$\bar{u}_i = u_i \cos(\varphi) + w_i \sin(\varphi) \tag{2.11}$$

$$\bar{w}_i = -u_i \sin(\varphi) + w_i \cos(\varphi) \tag{2.12}$$

$$\bar{u}_j = u_j \cos(\varphi) + w_j \sin(\varphi) \tag{2.13}$$

$$\bar{w}_j = -u_j \sin(\varphi) + w_j \cos(\varphi) \tag{2.14}$$

In matrix form

$$\bar{\mathbf{u}} = \begin{bmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ w_i \\ \theta_i \\ u_j \\ w_j \\ \theta_j \end{bmatrix} = \mathbf{Tu} \tag{2.15}$$

where $c = \cos(\varphi)$ and $s = \sin(\varphi)$.

$\mathbf{T}$ is called the displacement-transformation matrix. The nodal forces are transformed in exactly the same manner:

$$\bar{\mathbf{f}} = \mathbf{Tf} \tag{2.16}$$

In the global coordinate system, the equation of motion reads

$$\mathbf{M\ddot{u}} + \mathbf{Ku} = \mathbf{f} \tag{2.17}$$

The matrices in the local coordinate system are denoted with a bar on top:

$$\bar{\mathbf{M}}\ddot{\bar{\mathbf{u}}} + \bar{\mathbf{K}}\bar{\mathbf{u}} = \bar{\mathbf{f}} \tag{2.18}$$

Applying equations (2.15) and (2.16) yields

$$\bar{\mathbf{M}}\mathbf{T\ddot{u}} + \bar{\mathbf{K}}\mathbf{Tu} = \mathbf{Tf} \tag{2.19}$$

$$\mathbf{T}^{-1}\bar{\mathbf{M}}\mathbf{T\ddot{u}} + \mathbf{T}^{-1}\bar{\mathbf{K}}\mathbf{Tu} = \mathbf{f} \tag{2.20}$$

Since $\mathbf{T}$ is square and orthogonal, its inverse is equal to its transform.

$$\mathbf{T}^{-1} = \mathbf{T}^{\mathrm{T}} \tag{2.21}$$

Equation (2.20) becomes then:

$$\mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\mathbf{T\ddot{u}} + \mathbf{T}^{\mathrm{T}}\bar{\mathbf{K}}\mathbf{Tu} = \mathbf{f} \tag{2.22}$$

Comparing equation (2.22) with the equation of motion in (2.17), the following global element matrix definitons are obtained:

$$\mathbf{M} = \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\mathbf{T} \tag{2.23}$$

$$\mathbf{K} = \mathbf{T}^{\mathrm{T}}\bar{\mathbf{K}}\mathbf{T} \tag{2.24}$$

$\mathbf{M}$ is the element mass matrix in global coordinates and $\mathbf{K}$ is the element stiffness matrix in global coordinates.

The force vector in global coordinates is determined by equation (2.16):

$$\mathbf{f} = \mathbf{T}^{\mathrm{T}}\bar{\mathbf{f}} \tag{2.25}$$

Assembly of System Matrices

So far only the element level has been considered. To investigate the structure's behaviour, the elements have to be assembled to constitute the whole structure. Mathematically this means that the element equations of motion are merged together, so that they form one global equation of motion.

The assembly is the merging of the element's nodes. Two important mechanical rules have to fulfilled:

- Compatibility of displacements
- Force equilibrium

The first rule requires that the displacements of different members meeting in one node have to be the same, i.e. they should move as one entity. The second rule states that all internal forces in a node imposed by attached members have to be balanced by external forces. The assembly process is thus governed by the following relationship:

$$\mathbf{f} = \mathbf{f}^{(1)} + \mathbf{f}^{(2)} + \ldots + \mathbf{f}^{(n)} \tag{2.26}$$

The element equations are given as

$$\begin{aligned} \mathbf{f}^{(1)} &= \mathbf{M}^{(1)}\ddot{\mathbf{u}} + \mathbf{K}^{(1)}\mathbf{u} \\ \mathbf{f}^{(2)} &= \mathbf{M}^{(2)}\ddot{\mathbf{u}} + \mathbf{K}^{(2)}\mathbf{u} \\ \mathbf{f}^{(n)} &= \mathbf{M}^{(n)}\ddot{\mathbf{u}} + \mathbf{K}^{(n)}\mathbf{u} \end{aligned} \tag{2.27}$$

Using equation (2.26) and equations (2.27) yields

$$\mathbf{f} = \left(\mathbf{M}^{(1)} + \mathbf{M}^{(2)} + \ldots + \mathbf{M}^{(n)}\right)\ddot{\mathbf{u}} + \left(\mathbf{K}^{(1)} + \mathbf{K}^{(2)} + \ldots + \mathbf{K}^{(n)}\right)\mathbf{u} \tag{2.28}$$

$$\mathbf{f} = \mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} \tag{2.29}$$

The merging process turns out to be a simple matrix addition. Equation (2.29) is the global equation of motion.

The global stiffness matrix has to be quadratic, symmetric and singular after assembly and before application of boundary conditions. Hence, a good check is to determine whether the stiffness matrix is singular. A quadratic matrix is singular if its determinant is zero. If this check is done in Matlab with the code written for this report, sometimes values much larger than zero are obtained. Since numerical errors are assumed as reason for this, the system stiffness matrix is also assembled using Mathematica. The determinant is then calculated using symbolic variables. This way the determinant of the system stiffness matrix is in fact zero.

Damping

Damping is the dissipation of kinetic energy for vibrating systems. It is present in every real structure. A straight-forward way of accounting for damping is based on the master mass and stiffness matrices, where it is assumed that the damping matrix **C** is proportional to the mass matrix **M** and stiffness matrix **K**:

$$\mathbf{C} = \alpha_1 \mathbf{M} + \alpha_2 \mathbf{K} \tag{2.30}$$

This way of accounting for damping in the structure is called Rayleigh damping. In the model, damping is not of big importance because it is usually small and here only forced vibrations are considered. The coefficients α_1 and α_2 are thus chosen as small values, so that the response is not affected much.

2.2.3 Transition Piece

The transition piece is a concrete block with dimensions 9.6 × 8 × 4 m and a mass of 666 t. In the model the transition piece is represented by plane frame elements by an equivalent area moment of inertia and cross-section with the following formulae:

$$I_{\mathrm{TP}} = \frac{b_{\mathrm{TP}} h_{\mathrm{TP}}^3}{12} \tag{2.31}$$

$$A_{\mathrm{TP}} = \frac{m_{\mathrm{TP}}}{\rho_{\mathrm{c}} l_{\mathrm{TP}}} \tag{2.32}$$

2.2.4 Implementation

The finite element model is developed in Matlab. Routines for the different steps are developed and programmed. The programme takes the support structure's properties as input and determines the response of the structure at every degree of freedom. The input consists of a node coordinate list, an element list, element properties and material properties. Possible outputs variables are displacements, velocities and accelerations. The programme code can be found in Appendix ABC. (figure programme?)

2.3 Representation of the Soil in the Model

2.3.1 Stiffness of the Soil-Pile System

For simplicity it is assumed that the soil at the fictitious location of the jacket structure consists solely of dense sand as can be encountered more often in the North Sea. Important soil parameters are listed in Table 2.1.

The foundation of the jacket support structure consists of four (in case of the two-dimensional jacket structure two) long, slender piles that are embedded in the soil and ensure the load transmission into the sea bed. Different models to represent the soil-structure interaction for offshore wind turbine support structures have been investigated by Zaaijer [8]. He recommends the use of a stiffness matrix, that is added to the two nodes that are located at the soil-structure interface to represent the soil-structure interaction.

Parameter	Value
E_s	$80 \cdot 10^6\,\mathrm{N/m^2}$
G_s	$3.0769 \cdot 10^7\,\mathrm{N/m^2}$
ν_s	0.3

Table 2.1: Soil properties

The stiffness matrix is of the following form:

$$\mathbf{K}_\mathrm{s}\mathbf{u} = \begin{bmatrix} k_x & 0 & k_{x\theta} \\ 0 & k_z & 0 \\ k_{x\theta} & 0 & k_\theta \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \end{bmatrix} \tag{2.33}$$

The stiffness matrix consists of three global degrees of freedom, i.e. horizontal motion, vertical motion and rotation. The stiffness values on the diagonal of the matrix can be seen as adding springs in the respective degree of freedom at the nodes. The two non-diagonal term $k_{x\theta}$ stand for the coupling between the horizontal stiffness and the rotational stiffness. No interaction between the two slender foundation piles in the soil is assumed.

Horizontal and Rotational Soil-Pile Stiffness

Carter [9] analysed laterally loaded flexible shafts in rock subjected to lateral loads and moments. He states two closed-form expressions that provide accurate approximations for the horizontal displacement u and rotation θ of the shaft at the level of the soil surface.

$$u = 0.50\left(\frac{H}{G^*L}\right)\left(\frac{E_e}{G^*}\right)^{-1/7} + 1.08\left(\frac{M}{G^*L^2}\right)\left(\frac{E_e}{G^*}\right)^{-3/7} \tag{2.34}$$

$$\theta = 1.08\left(\frac{H}{G^*L^2}\right)\left(\frac{E_e}{G^*}\right)^{-3/7} + 6.40\left(\frac{M}{G^*L^3}\right)\left(\frac{E_e}{G^*}\right)^{-5/7} \tag{2.35}$$

H and M are the horizontal force and moment respectively. G^* is the equivalent shear modulus of the soil and is defined as

$$G^* = G_\mathrm{s}\left(1 + \frac{3\nu_\mathrm{s}}{4}\right) \tag{2.36}$$

For an isotropic soil the soil's shear modulus G_s is given by

$$G_\mathrm{s} = \frac{E_\mathrm{s}}{2(1+\nu_\mathrm{s})} \tag{2.37}$$

E_e is the effective Young's modulus of the elastic shaft embedded in the soil and given by

$$E_e = \frac{EI}{\pi D^4/64} \tag{2.38}$$

where E and I are the Young's modulus and the area moment of inertia in bending of the shaft. According to the UpWind reference jacket support structure the piles are assumed to be made of steel and have a diameter D of 2.082 m and a wall thickness t of 0.06 m. The pile's embedment depth L is 45 m.

In contrast to the paper, in this report expressions for piles in soil and not in rock are searched for. However, the equations (2.34) and (2.35) cover also Young's moduli and shear moduli that are characteristic for sandy soils, as can be seen from the range of applicability stated by Carter. He verified that the modulus ratio E_e/E_s can lie between 1 and 10^6 and that the slenderness ratio L/D has to be larger than 1. Soil and pile properties used in this report fall within the range of applicability.

From equations (2.34) and (2.35) it is straight-forward to determine the stiffness constants for the soil-pile interaction. The two equations can be written in matrix form as

$$\mathbf{u} = \mathbf{C}_{\text{coeff}}\mathbf{f} \tag{2.39}$$

Transforming this equation into the well-known stiffness relation yields the stiffness matrix for the horizontal and rotational terms. The stiffness is obtained as the inverse of the coefficient matrix.

$$\mathbf{K} = \mathbf{C}_{\text{coeff}}^{-1} \tag{2.40}$$

The inverse of the coefficient matrix $\mathbf{C}_{\text{coeff}}$ is

$$\mathbf{K} = \mathbf{C}_{\text{coeff}}^{-1} = \begin{bmatrix} 0.50\left(\frac{1}{G^*L}\right)\left(\frac{E_e}{G^*}\right)^{-1/7} & 1.08\left(\frac{1}{G^*L^2}\right)\left(\frac{E_e}{G^*}\right)^{-3/7} \\ 1.08\left(\frac{1}{G^*L^2}\right)\left(\frac{E_e}{G^*}\right)^{-3/7} & 6.40\left(\frac{1}{G^*L^3}\right)\left(\frac{E_e}{G^*}\right)^{-5/7} \end{bmatrix}^{-1} \tag{2.41}$$

The upper left and the lower right diagonal entries stand for the horizontal and rotational stiffness term, respectively. With the soil properties stated previously, the horizontal and rotational stiffness constants become

$$k_x = 6.7821 \cdot 10^8 \,\frac{\text{N}}{\text{m}^2} \tag{2.42}$$

$$k_\theta = 1.3061 \cdot 10^{10} \,\frac{\text{N}}{\text{m}^2} \tag{2.43}$$

$$k_{x\theta} = -1.7969 \cdot 10^9 \,\frac{\text{N}}{\text{m}^2} \tag{2.44}$$

Because of different coordinate system definitions here and in Carter's paper, the negative value of $k_{x\theta}$ has to be taken.

Vertical Soil-Pile Stiffness

Randolph [10] investigated the deformation characteristics of vertically loaded piles. By means of an analytical approach, he developed an approximate closed-form solution for the settlement of a pile under a given load, which can be used here to derive a stiffness value for the vertical soil-structure interaction.

Randolph gives the load settlement ratio for a pile-soil system with the following formula:

$$\frac{P_\mathrm{t}}{G_\mathrm{s}Rw_\mathrm{t}} = \frac{P_\mathrm{b}}{G_\mathrm{s}Rw_\mathrm{b}} + \frac{P_\mathrm{s}}{G_\mathrm{s}Rw_\mathrm{s}} = \frac{4}{\eta(1-\nu_\mathrm{s})} + \frac{2\pi L}{\zeta R} \tag{2.45}$$

The total deformation of the pile has been divided into a part due to the pile's base and a part due to the pile's shaft deformation. ζ is defined as

$$\zeta = \ln\left(\frac{R_\mathrm{m}}{R}\right) \tag{2.46}$$

R_m defines a magical radius where the effect of the shear stress becomes negligible and is given as

$$R_\mathrm{m} = 2.5L(1-\nu_\mathrm{s}) \tag{2.47}$$

Rewriting equation (2.45) to obtain the vertical stiffness term yields

$$k_z = \frac{P_\mathrm{t}}{w_\mathrm{t}} = \left(\frac{4}{\eta(1-\nu_\mathrm{s})} + \frac{2\pi L}{\zeta R}\right) G_\mathrm{s}R \tag{2.48}$$

Here, the same definitions as before apply for G_s, ν_s and L. R is simply the radius of the foundation pile and η a depth factor that allows for the interaction of the soil above and below the pile base and whose value is 1. Inserting the coefficients and properties, the vertical stiffness term is obtained:

$$k_z = 2.1940 \cdot 10^9\,\frac{\mathrm{N}}{\mathrm{m}^2} \tag{2.49}$$

2.3.2 Soil Damping

To account for the damping effect of the soil-pile system, a damping matrix similar to the soil-pile stiffness matrix is set up.

$$\mathbf{C}_\mathrm{s}\mathbf{u} = \begin{bmatrix} c_x & 0 & c_{x\theta} \\ 0 & c_z & 0 \\ c_{x\theta} & 0 & c_\theta \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \end{bmatrix} \tag{2.50}$$

The damping value estimation is based on the consideration of a single degree of freedom system. The natural frequency ω_n is then given by

$$\omega_\mathrm{n} = \sqrt{\frac{k}{m}} \tag{2.51}$$

The stiffness k is given by the soil's stiffness determined in the previous section and the mass is determined based on the knowledge of the first natural frequency of the overall system as follows

$$m = \frac{k}{\omega_\mathrm{n}^2} \tag{2.52}$$

Assuming the damping ratio to be $\xi = 0.02$, the damping coefficients can be determined:

$$\xi = \frac{c}{2\sqrt{km}} \tag{2.53}$$

$$c = 2\xi\sqrt{km} \tag{2.54}$$

This procedure is repeated for every coefficient, so that the soil damping coefficients are finally obtained as

$$c_x = 8.8800 \cdot 10^7 \, \frac{\mathrm{N}}{\mathrm{m}^2} \tag{2.55}$$

$$c_{x\theta} = 2.3527 \cdot 10^8 \, \frac{\mathrm{N}}{\mathrm{m}^2} \tag{2.56}$$

$$c_\theta = 1.7101 \cdot 10^9 \, \frac{\mathrm{N}}{\mathrm{m}^2} \tag{2.57}$$

$$c_z = 2.8727 \cdot 10^8 \, \frac{\mathrm{N}}{\mathrm{m}^2} \tag{2.58}$$

2.4 Offshore Wind Turbine

The wind turbine used for the investigation is the NREL offshore 5 MW baseline wind turbine that has been developed for research purposes and is also used in the UpWind project [11]. It is a conventional, three-bladed, horizontal upwind variable-speed turbine. Important parameters for the further investigation are listed in Table 2.2.

Parameter	Symbol	Value
Rotor diameter	D_R	126 m
Top mass	m_RNA	350 t
Rotor mass	m_R	110 t
Nacelle mass	m_NA	240 t
Nacelle height	h_NA	3.5 m
Nacelle length	l_NA	14 m
Rated wind speed	V_r	11.4 m/s

Table 2.2: Important parameters of the NREL turbine

The wind turbine, or the rotor-nacelle assembly, is represented by a circular disc for the rotor and a cuboid for the nacelle on top of the turbine tower with the respective masses given in Table 2.2. The procedure to introduce the turbine into the matrix equation of motion is similar to that for the soil springs. Neglecting interaction or coupling effects, the equation for the turbine can be written as an additional mass matrix on the top node:

$$\mathbf{M\ddot{u}} = \begin{bmatrix} m_{\mathrm{RNA}} & 0 & 0 \\ 0 & m_{\mathrm{RNA}} & 0 \\ 0 & 0 & I_{\mathrm{RNA}} \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \ddot{w} \\ \ddot{\theta} \end{bmatrix} \tag{2.59}$$

where I_{RNA} is determined by

$$I_{\mathrm{RNA}} = I_{\mathrm{R}} + I_{\mathrm{NA}} = \frac{m_{\mathrm{R}} R_{\mathrm{R}}^2}{4} + \frac{1}{12} m_{\mathrm{NA}} \left(l_{\mathrm{NA}}^2 + h_{\mathrm{NA}}^2 \right) \tag{2.60}$$

For the translational degrees of freedom the mass of the rotor-nacelle assembly is added to the diagonal term of the matrix. For the rotational degree of freedom, the mass moment of inertia is subdivided into the mass moment of inertia for the rotor and the nacelle first. The rotor is assumed to be a circular disc with the same mass as the rotor of the NREL turbine. The nacelle is assumed as a cuboid with the mass of the NREL nacelle. The values for the calculation are specified in Table 2.2. Most values have been taken from the NREL turbine [11], however, nacelle dimensions are not defined there except for the height, so that a value has been assumed for the length of the nacelle.

3 Crack Modelling for Beam Elements

3.1 Introduction

Jacket support structures are composed of tubular slender elements that are welded at intersections. The stochastic nature of wind and waves acting on the turbine and the support structure cause a large number of stress cycles. Moreover, the operation of the wind turbine induces further cyclic loads in the support structure. Stress raising effects make tubular joints the most critical points with respect to fatigue strength in the jacket structure. Fatigue cracks are thus most likely to develop from these tubular joints. Two effects make welded tubular joints especially prone to fatigue initiation. On the one hand, the geometric layout of tubular joints leads to increased stresses in the vicinity of the joints. These additional stresses are also called geometrical stress or hot spot stress. On the other hand, the local weld geometry influences the stress level. At the weld toe, the most fatigue critical area, unavoidable irregularities like notches occur and induce very localised stress concentrations. Together, these two effects make tubular joints in jacket support structures the most prone area for fatigue crack initiation and growth.

Tubular joint K2 in the right leg is chosen (cf. Figure 2.1) as the location where a crack is introduced to the model. The location is chosen, on the on hand, because the joint K2 lies below sea level and is thus not easily accessible and, on the other hand, because it lies close to the sea surface, where wave excitation is larger than in lower areas.

In this chapter, different ways to model structural damage or cracks in beams are investigated in the literature review. Subsequently, a method is selected and the implementation is described.

3.2 Literature Review

The effects of cracks or local defects on the dynamic response of structural members have been investigated for the past decades. Different concepts have been developed. The early investigations considered mainly one-dimensional single members like beams or bars. The crack was then modelled as a local flexibility causing local variations in the stiffness of the member, often described as 'softening' effect. The flexibility of the crack area is a highly local phenomenon and has only influence on the crack's immediate neighbourhood. However, the overall dynamic behaviour is affected to a considerable degree, which attracted the attention of the researchers.

First approaches to the problem modelled the crack by an equivalent reduced section or local bending moment, with magnitudes estimated by experimentation. Later, the local flexibility was quantified by relating it to the crack's stress intensity factor (SIF) based on fracture mechanics methods. In this way, the magnitude of the local flexibility was related to the crack's geometry. Subsequently, this approach was used extensively for the dynamic investigation of cracked one-dimensional structures. Most works in this area deal with beams subject to various boundary conditions. Parameters that are investigated are especially the natural frequencies and the mode shapes. Okamura et al. [12] studied the stability of cracked columns. Adams et al. [13] were among the first to represent the crack by an equivalent linear spring to investigate axial vibrations of cracked bars. Chondros et al. [14] used experiments to find a relation between the crack depth and the rotational spring constant. For damage detection, they suggested periodic measurements of the first three natural frequencies to compare the rate of change to natural frequencies of the structure when it was first erected. The aforementioned 'softening' effect of the crack leads to a reduction of the eigenfrequencies that can possibly be measured. Rizos et al. [15] related the measured vibration modes to the crack location and depth. Ostachowicz et. al. [16] studied the effect of two cracks on the dynamic behaviour of a cantilever beam. Narkis [17] developed a method that only requires the first two natural frequencies to identify a crack in a beam.

The aforementioned investigations that model the crack by means of a rotational spring assume that only one degree of freedom, namely in-plane rotation, is affected by the crack. Other researchers, cf. e.g. Liu et al. [18], developed the concept of a local flexibility matrix with a maximum size of 6×6, which is equivalent to inserting a joint at the crack location and representing that joint by local flexibility matrix. Such a matrix defines the relationship between the displacements and the forces at the crack location and gives a flexibility constant for every considered degree of freedom on its diagonal. Additional non-diagonal terms indicate coupling between the different modes of vibration.

An important step was made when Christides and Barr [19] developed a continuous cracked beam theory, which was further developed by Chondros et al. [20]. They developed a crack disturbance function, that distributes the effect of the flexibility along the beam, turning away from the concept of local flexibility to 'continuous' flexibility introduced by the crack.

Dimarogonas [21] has performed an extensive review of the early publications until 1996 on vibrations of cracked structures to which the reader is referred for further information.

Most publications mentioned here deal with solid rectangular or circular cross-sections. However, many realistic civil engineering structures like bridges and offshore wind turbine jacket support structures comprise hollow section members. Liu et al. [18] were among the first to examine cracked hollow section beams by means of coupled response measurements, where the crack is simulated by a local flexibility matrix. Furthermore, Zheng et al. [22] derived general integral formulae for the equivalent flexibility coefficient of a crack for hollow section beams with different cross-sections.

The concepts presented so far have a significant restriction because they only apply to simple single element structures. For the investigation of an offshore wind turbine jacket support structure, the concepts are not applicable. Today, it would be possible to study the whole support structure by a detailed finite element model that includes modelling of the crack itself by very small three-dimensional finite elements to represent the sharp crack tip. However, this requires extensive modelling and computation time. For identification problems, a simple and straight-forward way to simulate the crack is needed because of the possibility to shift the crack location in the model or change the crack depth. In general, the computational model has to be capable of performing fast and extensive modifications of the crack.

To this end, cracked beam finite elements have been developed. They allow the study of more complex structures as well as easy modifications because the crack itself is not modelled but represented by one of the local flexibility methods presented before. The cracked beam finite elements can be easily inserted into a finite element model. Gounaris et al. [23] and Krawczuk et al. [24] formulated cracked beam finite elements based on static interpolation functions. Skrinar [25] presented a dynamic cracked beam finite element, where so-called dynamic interpolation functions were used, which reduces the size of the problem as a minimum number of finite elements are needed to model structural members.

For the dynamic analysis in this thesis, a cracked beam finite element formulation is chosen, where the crack is modelled by a local flexibility (cf. Section 3.3). Subsequently, a relation between the crack geometry and the local flexibility is determined (cf. Section 3.4).

3.3 Cracked Beam Finite Element

The crack is modelled by means of a rotational spring, i.e. a local flexibility. Thus, only the bending stiffness of the beam is reduced due to the crack. It is assumed that the beam stays continuous in horizontal and vertical direction. To introduce the rotational spring in the model of the jacket support structure, a new finite element is needed. Krawczuk et al. [24] have developed a cracked beam finite element that consists of three parts: an elastic beam of length $L/2$, a rotational spring with stiffness K_r and no geometrical length and an elastic beam of length $L/2$. The finite element has four degrees of freedom consisting of one translational and one rotational degree of freedom per node. The cracked beam finite element is shown in Figure 3.1. Since the crack, represented by the rotational spring, introduces a discontinuity of the slopes of the left and right beam segment, different shape functions for the left and the right beam sections have to be employed.

$$w_1(x) = a_1 + a_2 x + a_3 x^2 + a_4 x^3 \tag{3.1}$$

$$w_2(x) = a_5 + a_6 x + a_7 x^2 + a_8 x^3 \tag{3.2}$$

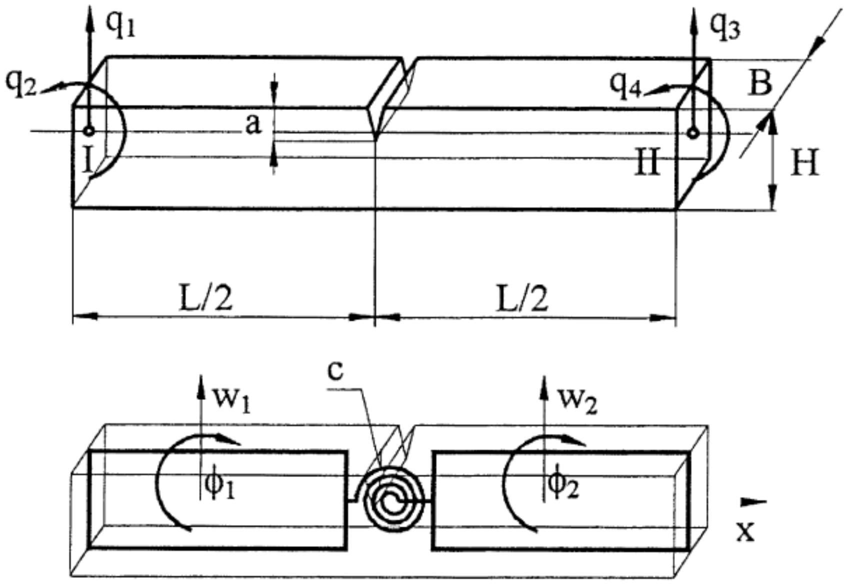

Figure 3.1: Cracked beam finite element [24]

In matrix form, equations (3.1) and (3.2) can be written as

$$w_1(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{3.3}$$

$$w_2(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} \tag{3.4}$$

The corresponding derivatives are determined to be

$$w_1'(x) = \phi_1(x) = a_2 + 2a_3x + 3a_4x^2 \tag{3.5}$$
$$w_2'(x) = \phi_2(x) = a_6 + 2a_7x + 3a_8x^2 \tag{3.6}$$
$$w_1''(x) = 2a_3 + 6a_4x \tag{3.7}$$
$$w_2''(x) = 2a_7 + 6a_8x \tag{3.8}$$
$$w_1'''(x) = 6a_4 \tag{3.9}$$
$$w_2'''(x) = 6a_8 \tag{3.10}$$

The boundary conditions are the unknown translations and rotations at both ends of the element:

$$w_1(0) = q_1 \tag{3.11}$$
$$\phi_1(0) = q_2 \tag{3.12}$$
$$w_2(L) = q_3 \tag{3.13}$$
$$\phi_2(L) = q_4 \tag{3.14}$$

The interface conditions are

- the equality of displacements:

$$w_1(L/2) = w_2(L/2) \tag{3.15}$$

- the equality of rotations taking into account the increase of rotation due to the rotational spring:

$$\phi_1(L/2) = \phi_2(L/2) - \frac{EI}{K_\mathrm{r}} w_2''(L/2) \tag{3.16}$$

- the equality of bending moments:

$$w_1''(L/2) = w_2''(L/2) \tag{3.17}$$

- and the equality of shear forces:

$$w_1'''(L/2) = w_2'''(L/2) \tag{3.18}$$

In the following, the factor EI/K_r in equation (3.16) is replaced by a crack coefficient K introduced by Krawczuk et al. [24].

With these eight equations the constants a_1 to a_8 can be determined. The following equation system is obtained:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & L & L^2 & L^3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2L & 3L^3 \\ 1 & \frac{L}{2} & \frac{L^2}{4} & \frac{L^3}{8} & -1 & -\frac{L}{2} & -\frac{L^2}{4} & -\frac{L^3}{8} \\ 0 & 1 & L & \frac{3}{4}L^2 & 0 & -1 & -L+2K & -\frac{3}{4}L^2 + 3LK \\ 0 & 0 & 2 & 3L & 0 & 0 & -2 & -3L \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & -6 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.19}$$

The equation system is solved and the coefficients of q_1, q_2, q_3 and q_4 are collected in a matrix $\mathbf{A}$, so that

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{L^2} & -\frac{3K+4L}{2L(K+L)} & \frac{3}{L^2} & -\frac{3K+2L}{2L(K+L)} \\ \frac{2}{L^3} & \frac{1}{L^2} & -\frac{2}{L^3} & \frac{1}{L^2} \\ 1 & \frac{KL}{2(K+L)} & 0 & -\frac{KL}{2(K+L)} \\ 0 & \frac{L}{K+L} & 0 & \frac{K}{K+L} \\ -\frac{3}{L^2} & -\frac{3K+4L}{2L(K+L)} & \frac{3}{L^2} & -\frac{3K+2L}{2L(K+L)} \\ \frac{2}{L^3} & \frac{1}{L^2} & -\frac{2}{L^3} & \frac{1}{L^2} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} \tag{3.20}$$

which is equivalent to

$$\mathbf{A}\mathbf{u} = \mathbf{a} \tag{3.21}$$

$\mathbf{u}$ contains the four nodal degrees of freedom of the finite element. Inserting equation (3.21) into (3.3) and (3.4) and having in mind that

$$w(x) = \mathbf{N}\mathbf{u} \tag{3.22}$$

the shape function matrices $\mathbf{N}_1$ and $\mathbf{N}_2$ are obtained as

$$\mathbf{N}_1 = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{L^2} & -\frac{3K+4L}{2L(K+L)} & \frac{3}{L^2} & -\frac{3K+2L}{2L(K+L)} \\ \frac{2}{L^3} & \frac{1}{L^2} & -\frac{2}{L^3} & \frac{1}{L^2} \end{bmatrix} \tag{3.23}$$

$$\mathbf{N}_2 = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} 1 & \frac{KL}{2(K+L)} & 0 & -\frac{KL}{2(K+L)} \\ 0 & \frac{L}{K+L} & 0 & \frac{K}{K+L} \\ -\frac{3}{L^2} & -\frac{3K+4L}{2L(K+L)} & \frac{3}{L^2} & -\frac{3K+2L}{2L(K+L)} \\ \frac{2}{L^3} & \frac{1}{L^2} & -\frac{2}{L^3} & \frac{1}{L^2} \end{bmatrix} \tag{3.24}$$

The strain-displacement matrix is

$$\mathbf{B}_i = \frac{\mathrm{d}^2}{\mathrm{d}x^2}\mathbf{N}_i \qquad \text{with} \quad i = 1, 2 \tag{3.25}$$

The stiffness matrix of the cracked beam finite element is given by the following formula that is derived in Appendix A.2:

$$\mathbf{K} = \int_0^L EI\mathbf{B}^{\mathrm{T}}\mathbf{B}\,\mathrm{d}x \tag{3.26}$$

Adapted to the case with discontinuous shape functions the formula reads

$$\mathbf{K} = \int_0^{L/2} EI\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1 \, \mathrm{d}x + \int_{L/2}^{L} EI\mathbf{B}_2^{\mathrm{T}}\mathbf{B}_2 \, \mathrm{d}x \tag{3.27}$$

The stiffness matrix is then determined to be

$$\mathbf{K} = EI \begin{bmatrix} \frac{12}{L^3} & \frac{6}{L^2} & -\frac{12}{L^3} & \frac{6}{L^2} \\ \frac{6}{L^2} & \frac{3K^2+6KL+4L^2}{L(K+L)^2} & -\frac{6}{L^2} & \frac{3K^2+6KL+2L^2}{L(K+L)^2} \\ -\frac{12}{L^3} & -\frac{6}{L^2} & \frac{12}{L^3} & -\frac{6}{L^2} \\ \frac{6}{L^2} & \frac{3K^2+6KL+2L^2}{L(K+L)^2} & -\frac{6}{L^2} & \frac{3K^2+6KL+4L^2}{L(K+L)^2} \end{bmatrix} \tag{3.28}$$

When the crack coefficient K is equal to zero, the stiffness matrix for the beam element with a crack reduces to the stiffness matrix of a normal beam element (cf. equation (2.4)).

Since plane frame elements are considered, the stiffness matrix for the beam has to be extended to a 6 × 6 matrix, as has been done in Section 2.2.2.

A similar approach could have been taken for the mass matrix as well. However, Krawczuk et al. [24] concluded that the mass matrix of a non-cracked beam element can be taken as the results are not influenced. They also provided numerical examples that confirmed the form of the stiffness matrix.

The Mathematica code for the derivation of the stiffness matrix of the cracked beam finite element can be found in Appendix C.1.

3.4 Determination of Spring Constant

Zheng et al. [22] present a derivation for the flexibility coefficient of cracked hollow section beams. Fracture mechanics theory is employed to describe the size and the influence of the crack on the stiffness or flexibility of the beam. The flexibility coefficient is, among others, dependent on the depth of the crack.

The derivation is divided into two parts to account for the abrupt change of cross-section when the crack reaches the inner hollow section of the beam. Case 1 covers a shallow open crack whose penetration depth is contained in the top solid sectional region. Case 2 covers deeper penetration in which the crack advances to and within the middle hollow section.

During the whole derivation it is assumed that the crack stays always open and that the beam acts in its linear-elastic range.

3.4.1 Case 1

Figure 3.2 shows the circular hollow cross-section of the beam with a sectional crack. The crack depth is denoted a, the external diameter D_e and the internal diameter D_i.

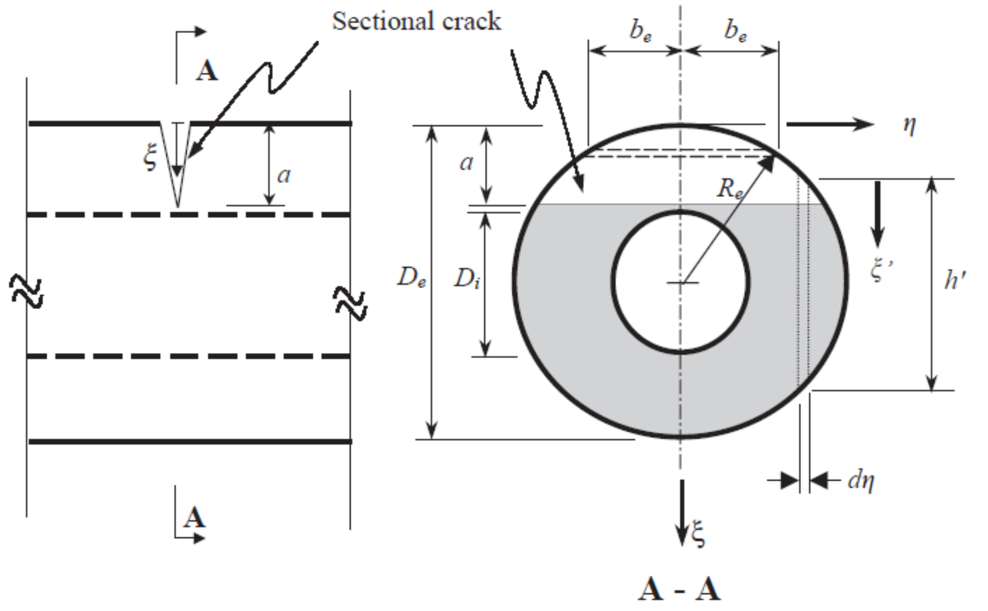

Figure 3.2: Crack in a circular hollow section beam – Case 1 [22]

ξ and η are location variables. This case is applicable when $0 \leq a \leq (D_e - D_i)/2$. The additional strain energy due to the crack can be expressed by a local flexibility coefficient C, which by definition is

$$C = \frac{\partial \theta}{\partial M} \tag{3.29}$$

The relative rotation θ, that is caused by the presence of the crack, can be derived by Castigliano's theorem in the linear elastic range [26]. Castigliano's theorem states that the partial derivative of the strain energy with respect to an applied force or moment is equal to the displacement or rotation of the force or moment along its line of action for linearly elastic structures. In equation form for this case:

$$u_i = \frac{\partial U}{\partial F_i} \tag{3.30}$$

where u_i is the relative displacement or rotation of the cracked section and F_i is a force or moment acting at the same location. The strain energy U can be expressed as follows:

$$U = \int_{A_c} J \, \mathrm{d}A \tag{3.31}$$

J is the strain energy release rate, which can be related to the stress intensity factor [21]

$$J = \left[\left(\sum_{i=1}^{6} K_{\mathrm{I}i} \right)^2 + \left(\sum_{i=1}^{6} K_{\mathrm{II}i} \right)^2 + m \left(\sum_{i=1}^{6} K_{\mathrm{III}i} \right)^2 \right] / E \tag{3.32}$$

The crack in this investigation is an fracture mode I crack[1] and influences only one degree of freedom, namely the rotation in the plane of the beam. Using equations (3.30) and (3.31) the relative rotation is then

$$\theta = \frac{\partial}{\partial M} \int_{A_c} J \,\mathrm{d}A \tag{3.33}$$

Together with equation (3.32) we get

$$\theta = \frac{\partial}{\partial M} \left[\int_0^a \int_{-b_e}^{b_e} \frac{K_I^2}{E} \,\mathrm{d}\eta \,\mathrm{d}\xi \right] \tag{3.34}$$

where b_e is defined as

$$b_e = \sqrt{\frac{D_e^2}{4} - \left(\frac{D_e}{2} - \xi \right)^2} \tag{3.35}$$

Considering a small vertical sectional strip with depth h', width $d\eta$ and ξ', a local depth variable measured from the top of the vertical strip, the stress intensity factor K_I is

$$K_I = \frac{Mh'}{2I} \sqrt{\pi \xi'} F' \tag{3.36}$$

where the area moment of inertia I is defined as

$$I = \pi \frac{D_e^4 - D_i^4}{64} \tag{3.37}$$

and F' is a function of the local relative position:

$$F' = \frac{\sqrt{\frac{2}{\pi x'} \tan\left(\frac{\pi x'}{2}\right)} \left[0.923 + 0.199 \left(1 - \sin\left(\frac{\pi x'}{2}\right)\right)^4 \right]}{\cos\left(\frac{\pi x'}{2}\right)} \tag{3.38}$$

The local relative position x' is given by

$$x' = \frac{\xi'}{h'} = \frac{2x + \sqrt{1 - 4y^2} - 1}{2\sqrt{1 - 4y^2}} \tag{3.39}$$

where $x = \xi / D_e$ and $y = \eta / D_e$. Finally, the flexibility coefficient is obtained when equation (3.36) together with equations (3.37) to (3.39) is inserted into (3.34), which subsequently is applied in the definition equation for the flexibility coefficient (3.29):

$$C = \frac{1024}{E D_e^3 \pi (1 - \gamma^4)^2} \int_0^{a/D_e} \int_{-\sqrt{x - x^2}}^{\sqrt{x - x^2}} (1 - 4y^2)(2x + \sqrt{1 - 4y^2} - 1) F'^2 \,\mathrm{d}y \,\mathrm{d}x \tag{3.40}$$

where $\gamma = D_i / D_e$.

[1] There are three standard crack modes in which a crack propagates. Mode I is called the opening mode, where tensile stresses act normal to the plane of the crack, mode II is called the shearing mode and mode III is called tearing mode. The latter two are only of minor importance.

3.4.2 Case 2

The procedure for the second case is very similar to the first case. Figure 3.3 shows the situation when the crack has propagated into the middle hollow section of the beam cross-section. This case is applicable when $(D_e - D_i)/2 \leq a \leq (D_e + D_i)/2$. The formula for the relative rotation θ is now adapted to the new situation, where the cracked cross-section consists of three parts: First, the solid sectional part with a depth of $t = (D_e - D_i)/2$, second the part left of the hollow middle section from depth t to the crack depth a and third the part right of the hollow middle section from depth t to the crack depth a. These three parts can be found in equation (3.41).

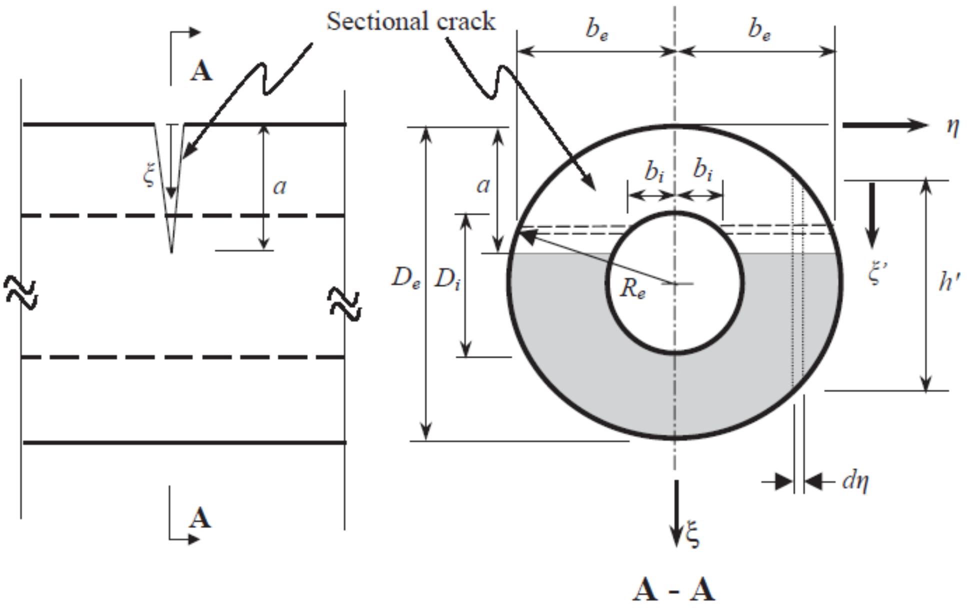

Figure 3.3: Crack in a circular hollow section beam – Case 2 [22]

$$\theta = \frac{\partial}{\partial M}\left[\int_0^t \int_{-b_e}^{b_e} \frac{K_\mathrm{I}^2}{E}\,\mathrm{d}\eta\,\mathrm{d}\xi + \int_t^a \int_{-b_e}^{-b_i} \frac{K_\mathrm{I}^2}{E}\,\mathrm{d}\eta\,\mathrm{d}\xi + \int_t^a \int_{b_i}^{b_e} \frac{K_\mathrm{I}^2}{E}\,\mathrm{d}\eta\,\mathrm{d}\xi\right] \tag{3.41}$$

Substituting equation (3.36) into equation (3.41) and then into the definition equation (3.29) the formula for the local flexibility constant is obtained:

$$\begin{aligned} C = &\frac{1024}{ED_e^3\pi(1-\gamma^4)^2}\left(\int_0^{t/D_e}\int_{-\sqrt{x-x^2}}^{\sqrt{x-x^2}} + \int_{t/D_e}^{a/D_e}\int_{-\sqrt{x-x^2}}^{-\beta} + \int_{t/D_e}^{a/D_e}\int_{\beta}^{\sqrt{x-x^2}}\right) \\ &\cdot \left((1-4y^2)(2x+\sqrt{1-4y^2}-1)F'^2\right)\,\mathrm{d}y\,\mathrm{d}x \end{aligned} \tag{3.42}$$

where $\beta = \sqrt{x - x^2 - (1-\gamma^2)/4}$ and the rest is defined as before in case 1.

3.4.3 Implementation

The rotational spring's constant representing the influence of the crack is denoted K_r and related to the local flexibility constant C as follows

$$K_\mathrm{r} = \frac{1}{C} \tag{3.43}$$

The crack coefficient K in the cracked beam's stiffness matrix is then

$$K = EIC \tag{3.44}$$

where C is the local flexibility of the crack whose integral equation has just been derived.

The cracked beam finite element is inserted into the finite element model at the lower brace of joint K2 in the right jacket leg. The element's length is approximately one meter, so that the crack is located at a distance of half a metre from the centreline of the jacket's leg.

The brace diameter is 0.8 m and the wall thickness is 0.02 m. With this information, the local flexibility for the crack in that brace can be calculated. The result is shown in Figure 3.4, where the crack coefficient K is plotted as a function of the crack depth a.

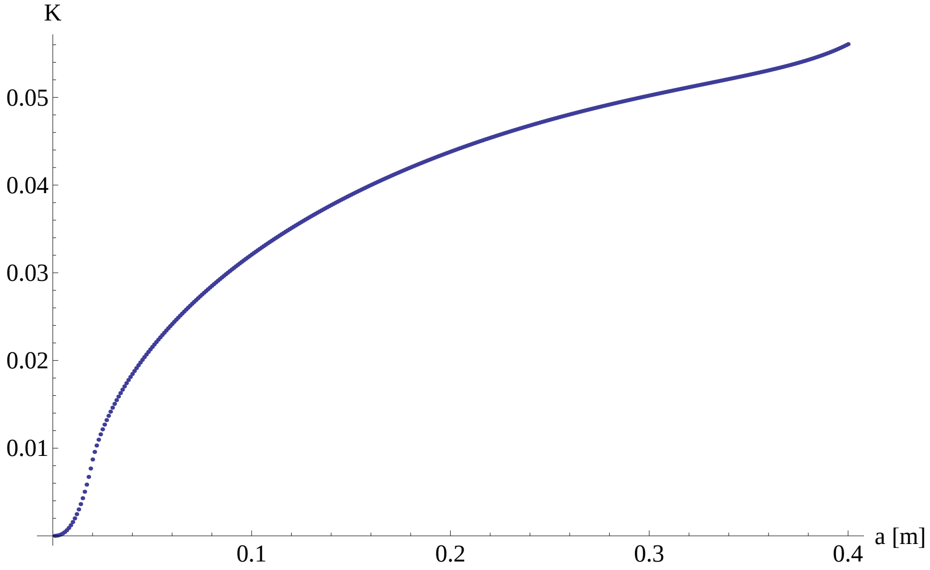

Figure 3.4: Rotational spring coefficient as function of crack depth

There are two different values in the cracked beam finite element's stiffness matrix that are affected by the presence of the crack, here they are denoted $k_{\theta\theta}$ and $k_{\theta z}$. They are defined as:

$$k_{\theta\theta} = EI\frac{3K^2 + 6KL + 4L^2}{L(K+L)^2} \tag{3.45}$$

$$k_{\theta z} = EI\frac{3K^2 + 6KL + 2L^2}{L(K+L)^2} \tag{3.46}$$

For $K = 0$ (no crack), the equations reduce to

$$k_{\theta\theta 0} = 4\frac{EI}{L} \tag{3.47}$$

$$k_{\theta z 0} = 2\frac{EI}{L} \tag{3.48}$$

To estimate the effect of the crack on the stiffness of the beam finite element, equations (3.45) and (3.46) are divided by (3.47) and (3.48). The results are presented in Table 3.1.

Crack depth	Relative crack depth	Crack coefficient	Relative stiffness variation	
a [m]	a/D_e [–]	K [m]	$k_{\theta\theta}/k_{\theta\theta 0}$ [–]	$k_{\theta z}/k_{\theta z 0}$ [–]
0.01 m	0.0125	0.00158	0.999	1.001
0.02 m	0.0250	0.00871	0.996	1.008
0.05 m	0.0625	0.02155	0.991	1.019
0.10 m	0.1250	0.03208	0.986	1.028
0.20 m	0.2500	0.04381	0.981	1.037
0.30 m	0.3750	0.05021	0.979	1.042
0.40 m	0.5000	0.05607	0.977	1.047

Table 3.1: Chosen crack depth and their coefficients

In comparison with the stiffness values of the uncracked beam finite element, it seems that the two diagonal terms become less stiff and that the two non-diagonal terms become stiffer.

4 Loads on the Structure

4.1 Introduction

In the previous two chapters, the structural model and the simulation of the crack in the model have been presented. Now, the loads on the structure are considered. The loading or excitation of an offshore wind turbine jacket support structure consists of mainly two types: aerodynamic and hydrodynamic loading. Aerodynamic loading arise due to the interaction of the airflow with the stationary and moving parts of the wind turbine. As discussed in Chapter 2, the rotating blades of the turbine are neglected. Thus, the aerodynamic loads in this investigation consist of the time-dependent thrust force on the rotor and the time-dependent wind load on the turbine tower. The hydrodynamic loads are caused by the interaction of the water flow with the structure. The water flow is dominated by waves and currents.

The IEC 61400-3 [27] specifies normal external conditions for the wind and wave environment of an offshore wind turbine. Those conditions are used because the investigation is focussed on the dynamic properties of the structure during normal ambient excitation and operation. Thus, no extreme wind or wave events are taken into account.

The wave conditions are described by an irregular sea state with a significant wave height H_s of 6 m and a spectral peak period T_p of 10 s. An unidirectional sea state is assumed, which means that no directional spreading is taken into account. The sea current is taken as 1 m/s and assumed constant over the water depth.

The ambient wind is defined by its mean wind speed V_m at hub height of 12 m/s and its turbulence intensity, which can be assumed to be 12 % in an offshore environment. The chosen mean wind speed lies just above the rated wind speed of the NREL turbine, so that the thrust force on the rotor is highest.

Wind, waves and current are assumed to be co-directional and unidirectional as a two-dimensional support structure is considered.

4.2 Hydrodynamic Loads

4.2.1 Morison's Equation

In DNV-OS-J101 [28] Morison's equation is suggested for the wave force calculation of slender structures such as jacket structures. Morison's equation gives a horizontal force on a vertical element $\mathrm{d}z$ of the structure at level z. It consists of an inertia force term and a drag force term, but neglects wave diffraction, radiation and viscous effects. It is furthermore assumed that the wave is not affected or disturbed by the structure itself. These neglected effects are usually only important for large volume structures. Inertia and drag term are 90° out of phase and simply superimposed:

$$\mathrm{d}F = \mathrm{d}F_{\mathrm{M}} + \mathrm{d}F_{\mathrm{D}} \tag{4.1}$$

$$\mathrm{d}F = \frac{1}{4}\pi\rho_{\mathrm{w}}C_{\mathrm{M}}D^2\dot{u}_{\mathrm{rel}}\,\mathrm{d}z + \frac{1}{2}\rho_{\mathrm{w}}C_{\mathrm{D}}Du_{\mathrm{rel}}|u_{\mathrm{rel}}|\,\mathrm{d}z \tag{4.2}$$

u_{rel} and $\dot{u}_{\mathrm{rel}}$ are the relative velocity and acceleration between the water flow and the structure's motion perpendicular to the member, D is the diameter of the cylinder, ρ_{w} is the density of sea water, C_{M} and C_{D} are the dimensionless inertia and drag coefficients. The coordinate system's origin is located at still water level with the z-axis pointing upwards. This contradicts the coordinate system definition of the support structure model, where the origin is located at the sea bed, and some modifications have to be made.

The drag and inertia coefficient are dependent, among others, on the Reynolds number, the Keulegan-Carpenter number and the relative roughness. At a typical southern North Sea site in 30 to 40 m water depth, they can be assumed as $C_{\mathrm{M}} = 1.6$ and $C_{\mathrm{D}} = 0.8$ according to DNV's offshore standard DNV-OS-J101 [28].

4.2.2 Regular Wave Kinematics

To determine the hydrodynamic loads by means of the Morison equation, the water kinematics must be known. Linear wave theory is assumed. Contributions from wave crests are ignored (profile extension methods, constant extension, how implemented?).

The following formulae and derivations are loosely based on the book "Offshore Hydromechanics" [29]. For a regular wave, the wave profile can be defined as

$$\zeta(x,t) = \zeta_{\mathrm{a}}\sin(\omega t - kx) \tag{4.3}$$

The linear wave potential ϕ in shallow water is

$$\phi(x,z,t) = -\frac{g\zeta_{\mathrm{a}}}{\omega}\frac{\cosh k(z+h)}{\cosh kh}\cos(\omega t - kx) \tag{4.4}$$

Wave frequency ω and wave number k are related by the dispersion relation. In shallow water it is defined as

$$\omega^2 = gk\tanh kh \tag{4.5}$$

The horizontal and vertical wave velocity u and w are obtained by taking the derivative of the wave potential with respect to the x and the z coordinate, respectively:

$$u(x,z,t) = \frac{\partial\phi}{\partial x} = \omega\zeta_{\mathrm{a}}\frac{\cosh k(z+h)}{\sinh kh}\sin(\omega t - kx) \tag{4.6}$$

$$w(x,z,t) = \frac{\partial\phi}{\partial z} = \omega\zeta_{\mathrm{a}}\frac{\sinh k(z+h)}{\sinh kh}\cos(\omega t - kx) \tag{4.7}$$

The time derivatives of the wave velocities u and w yield the horizontal and vertical water particle acceleration a_x and a_z:

$$a_x(x,z,t) = \frac{\partial u}{\partial t} = \omega^2 \zeta_\mathrm{a} \frac{\cosh k(z+h)}{\sinh kh} \cos(\omega t - kx) \tag{4.8}$$

$$a_z(x,z,t) = \frac{\partial w}{\partial t} = -\omega^2 \zeta_\mathrm{a} \frac{\sinh k(z+h)}{\sinh kh} \sin(\omega t - kx) \tag{4.9}$$

Equations (4.6) to (4.9) have to be adapted because of the different coordinate system definitions for the wave kinematics (origin at sea level) and for the jacket structure (origin at sea bottom). The structure's coordinate system will be used for the waves in the following as well. Hence, the wave velocities and accelerations are written as

$$u(x,z,t) = \omega \zeta_\mathrm{a} \frac{\cosh kz}{\sinh kh} \sin(\omega t - kx) \tag{4.10}$$

$$w(x,z,t) = \omega \zeta_\mathrm{a} \frac{\sinh kz}{\sinh kh} \cos(\omega t - kx) \tag{4.11}$$

$$a_x(x,z,t) = \omega^2 \zeta_\mathrm{a} \frac{\cosh kz}{\sinh kh} \cos(\omega t - kx) \tag{4.12}$$

$$a_z(x,z,t) = -\omega^2 \zeta_\mathrm{a} \frac{\sinh kz}{\sinh kh} \sin(\omega t - kx) \tag{4.13}$$

where h is the water depth.

4.2.3 Irregular Wave Kinematics

As a single linear wave is not a realistic description of a real sea state, irregular sea states are considered. An irregular sea state is obtained by linearly superposing regular wave components with different amplitude $\zeta_{\mathrm{a}i}$, frequency ω_i and phase ϵ_i. A long-crested irregular sea state propagating along the x-axis can be described by [30]

$$\zeta(x,t) = \sum_{i=1}^{n} \zeta_{\mathrm{a}i} \sin(\omega_i t - k_i x + \epsilon_i) \tag{4.14}$$

Along the lines of equations (4.6) to (4.9), the wave velocities and accelerations u, w, a_x and a_z for an irregular sea state can be written as

$$u(x,z,t) = \sum_{i=1}^{n} \omega_i \zeta_{\mathrm{a}i} \frac{\cosh k_i(z+h)}{\sinh k_i h} \sin(\omega_i t - k_i x + \epsilon_i) \tag{4.15}$$

$$w(x,z,t) = \sum_{i=1}^{n} \omega_i \zeta_{\mathrm{a}i} \frac{\sinh k_i(z+h)}{\sinh k_i h} \cos(\omega_i t - k_i x + \epsilon_i) \tag{4.16}$$

$$a_x(x,z,t) = \sum_{i=1}^{n} \omega_i^2 \zeta_{\mathrm{a}i} \frac{\cosh k_i(z+h)}{\sinh k_i h} \cos(\omega_i t - k_i x + \epsilon_i) \tag{4.17}$$

$$a_z(x,z,t) = \sum_{i=1}^{n} -\omega_i^2 \zeta_{\mathrm{a}i} \frac{\sinh k_i(z+h)}{\sinh k_i h} \sin(\omega_i t - k_i x + \epsilon_i) \tag{4.18}$$

A wave spectrum $S_\zeta(f)$ can be used to describe an irregular sea state. As the term 'state' indicates, the sea is then assumed to be a stationary random process. The spectrum describes how the amplitude of the wave components is distributed over the frequency components.

The JONSWAP spectrum is applied here as it specifically accounts for the limited fetch length in the North Sea, which leads to a not fully developed sea state. The spectrum is defined by its significant wave height H_s and the mean wave period T_p and given as

$$S_\zeta(f) = 0.3125 \cdot H_s^2 \cdot T_p \cdot \left(\frac{f}{f_p}\right)^{-5} \cdot (1 - 0.287 \ln \gamma) \cdot \exp\left(-1.25 \left(\frac{f}{f_\mathrm{p}}\right)^{-4}\right) \cdot \gamma^A \tag{4.19}$$

where A and γ are defined as

$$A = \exp\left(-0.5 \left(\frac{f - f_p}{\sigma f_p}\right)^2\right) \tag{4.20}$$

$$\gamma = \begin{cases} 5 & \text{for} \quad \frac{T_p}{\sqrt{H_s}} \leq 3.6 \\ \exp\left(5.75 - 1.15 \frac{T_p}{\sqrt{H_s}}\right) & \text{for} \quad 3.6 \leq \frac{T_p}{\sqrt{H_s}} \leq 5 \\ 1 & \text{for} \quad \frac{T_p}{\sqrt{H_s}} > 5 \end{cases} \tag{4.21}$$

The JONSWAP spectrum is shown in Figure 4.1 for a significant wave height H_s of 6 m and a spectral peak period T_p of 10 s.

The wave components' amplitude $\zeta_{\mathrm{a}i}$ can be retrieved from the wave spectrum by the following formula [30]:

$$S_\zeta(f_i)\Delta f = \frac{1}{2}\zeta_{\mathrm{a}i}^2 \tag{4.22}$$

$$\zeta_{\mathrm{a}i} = \sqrt{2 S_\zeta(f_i)\Delta f} \tag{4.23}$$

To create a realistic sea state in the time domain as required for the wave force calculation with the Morison equation, equation (4.14) is used. The JONSWAP wave spectrum is discretised into equally spaced frequency intervals Δf, for each of which a wave amplitude $\zeta_{\mathrm{a}i}$ can be computed by equation (4.23). The wave frequency ω_i is obtained by multiplying the discrete frequency values by 2π. The phase ϵ_i for every wave component is a random value between 0 and 2π.

The JONSWAP spectrum used for the wave load calculation is created with a frequency range from 0.001 Hz to 0.5 Hz in steps of 0.001 Hz. This corresponds to the superposition of 500 regular waves with periods between 2 s and 1 000 s.

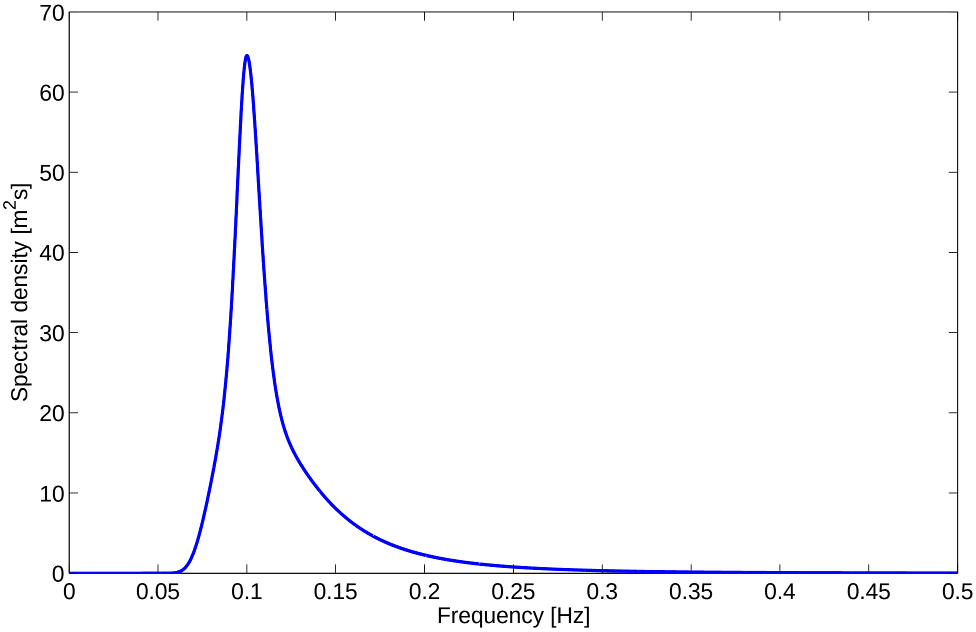

Figure 4.1: JONSWAP spectrum ($H_\mathrm{s} = 6\,\mathrm{m}$, $T_\mathrm{p} = 10\,\mathrm{s}$)

4.2.4 Inertia Term

Now, that the wave kinematics are defined, the force calculation can be performed. First, the inertia term of the Morison equation is presented as it is used in this report's calculations.

The phenomenon of an oscillating cylinder in an oscillating flow can be subdivided into a fixed cylinder in an oscillating flow plus the effect of the oscillating cylinder. A fixed cylinder in oscillating flow is subjected to a Froude-Krilov force and a disturbance force. The Froude-Krilov force arises from the time-dependent pressure gradient in the ambient flow, the disturbance force is a result of the flow disturbance of the impermeable cylinder. The cylinder oscillation obviously affects the disturbance force term. However, the Froude-Krilov force is not influenced by the cylinder's oscillation because it is assumed that the motion of the cylinder is small relative to the motion of the wave. Hence, the inertia term of the Morison equation for an oscillating cylinder in oscillating flow is written as

$$\mathrm{d}F_\mathrm{M} = \underbrace{\frac{1}{4}\pi D^2 \rho_\mathrm{w} C_\mathrm{M} a_\mathrm{n}}_{\mathrm{d}F_\mathrm{WF}} - \underbrace{\frac{1}{4}\pi D^2 \rho_\mathrm{w} (C_\mathrm{M} - 1)\ddot{x}_\mathrm{n}}_{\mathrm{d}F_\mathrm{A}} \tag{4.24}$$

a_n is the flow acceleration normal to the member's longitudinal axis, $\ddot{x}_\mathrm{n}$ is the structure's acceleration normal to its longitudinal axis.

The left term is the wave force term $\mathrm{d}F_\mathrm{WF}$ consisting of the Froude-Krilov force and the disturbance force, the right term is often denoted as added mass term $\mathrm{d}F_\mathrm{A}$. As the added mass term on the right-hand side is uncoupled from the rest of the force, it can be moved to the left-hand side of the equation of motion:

$$(\mathbf{M}+\mathbf{A})\,\ddot{\mathbf{x}}+\mathbf{C}\dot{\mathbf{x}}+\mathbf{K}\mathbf{x}=\mathbf{f} \tag{4.25}$$

The added mass matrix $\mathbf{A}$ is derived equivalently to the mass matrix. The mass matrix for an element is defined in Section A.2 as

$$\mathbf{M}=\rho A\int_0^L \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x=\frac{1}{2}\rho AL\int_{-1}^{1}\mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\xi \tag{4.26}$$

The term ρA is constant along the length of the element and describes the mass per unit length. From the added mass term, the added mass per unit length m_{a} can be extracted as well:

$$\mathrm{d}F_{\mathrm{A}}=m\ddot{x}=\underbrace{\rho_{\mathrm{w}}A_{\mathrm{d}}C_{\mathrm{A}}}_{m_{\mathrm{a}}}\ddot{x} \tag{4.27}$$

$A_{\mathrm{d}}=\frac{1}{4}\pi D^2$ is the displaced water volume per unit length (thus, in fact, a cross-section area). The added mass matrix $\mathbf{A}$ is thus defined as

$$\mathbf{A}=\rho_{\mathrm{w}}A_{\mathrm{d}}C_{\mathrm{A}}\int_0^L \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x=\frac{1}{2}\rho_{\mathrm{w}}A_{\mathrm{d}}LC_{\mathrm{A}}\int_{-1}^{1}\mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\xi \tag{4.28}$$

$$\mathbf{A}=\frac{\rho_{\mathrm{w}}V_{\mathrm{d}}C_{\mathrm{A}}}{420}\begin{bmatrix}140 & 0 & 0 & 70 & 0 & 0\\ 0 & 156 & 22L & 0 & 54 & -13L\\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2\\ 70 & 0 & 0 & 140 & 0 & 0\\ 0 & 54 & 13L & 0 & 156 & -22L\\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2\end{bmatrix} \tag{4.29}$$

$C_{\mathrm{A}}=C_{\mathrm{M}}-1$ is the added mass coefficient and $V_{\mathrm{d}}=A_{\mathrm{d}}L$ is the displaced water volume.

The part of the inertia force $\mathrm{d}F_{\mathrm{M}}$ remaining on the right-hand side of the equation of motion is:

$$\mathrm{d}F_{\mathrm{M}}=\frac{1}{4}\pi D^2\rho_{\mathrm{w}}C_{\mathrm{M}}a_{\mathrm{n}} \tag{4.30}$$

It can be determined with the wave acceleration perpendicular to the member, which is determined to be

$$a_{\mathrm{n}}=a_x\sin(\varphi)+a_z\cos(\varphi) \tag{4.31}$$

The angle φ represents the angle between the global coordinate system and the element's local coordinate system.

4.2.5 Drag Term

Drag is a phenomenon occurring in real flows that is caused by flow disturbance and wake formation. The drag term in the Morison equation is given by:

$$\mathrm{d}F_{\mathrm{D}}=\frac{1}{2}\rho_w C_{\mathrm{D}}Du_{\mathrm{rel}}|u_{\mathrm{rel}}| \tag{4.32}$$

In the case of an oscillating cylinder in an oscillating fluid, u_{rel} denotes the relative velocity between the flow and the structure. The relative velocity is given as

$$u = u_{\mathrm{n}} - \dot{x}_{\mathrm{n}} \tag{4.33}$$

so that the drag term becomes

$$\mathrm{d}F_{\mathrm{D}} = \frac{1}{2}\rho_w C_{\mathrm{D}} D \left(u_{\mathrm{n}} - \dot{x}_{\mathrm{n}}\right) \left|u_{\mathrm{n}} - \dot{x}_{\mathrm{n}}\right| \tag{4.34}$$

u_{n} is the flow's velocity normal to the member's axis and $\dot{x}_{\mathrm{n}}$ is the structure's velocity normal to the member's axis. The flow's velocity consists not only of the velocity due to the waves, but also of the current velocity u_{c}:

$$u_{\mathrm{n}} = u\sin(\varphi) + w\cos(\varphi) + u_{\mathrm{c}}\sin(\varphi) \tag{4.35}$$

The drag force in equation (4.34) is coupled with the left-hand side of the equation of motion, more specifically with the damping term. Therefore, the equations of motion have to be solved iteratively. This has been implemented in the calculations by means of the `ode` solver in Matlab.

In the later analysis and damage identification techniques, non-linear forces are very cumbersome to incorporate in the model. Thus, a fixed cylinder in oscillating fluid is considered and implemented as well to obtain a linear relationship:

$$\mathrm{d}F_{\mathrm{D}} = \frac{1}{2}\rho_{\mathrm{w}} C_{\mathrm{D}} D u_{\mathrm{n}} |u_{\mathrm{n}}| \tag{4.36}$$

4.3 Wind Loads

4.3.1 Introduction

The variable nature of wind can be described by a mean wind speed superimposed by turbulent fluctuations. The mean wind speed depends on seasonal, synoptic and diurnal effects and varies on a time scale of one to several hours or days, whereas the turbulent fluctuations vary on a relatively small time scale of a few minutes and have a zero mean value if averaged over a short time span [31]. In this report the time-dependent and varying wind velocity V is represented by two terms, as suggested. However, the mean wind speed is assumed to be a time-independent and constant mean value $\bar{V}$ because in this investigation a time scale in the order of minutes is of interest and hourly or daily variations can be neglected. The turbulent part of the wind velocity is described by a wind spectrum. Only horizontal wind components are considered, so that

$$V_x(t) = \bar{V}_x + \tilde{V}_x(t) \tag{4.37}$$

The wind speed variations are characterised by their frequency ω. Since long-term variations as seasonal, daily or hourly variations are not of interest, only turbulent variations are taken into account in the varying part of the wind force. As can be seen in Figure 4.2, the turbulence peak in the wind speed spectrum is associated with a period T of approximately one minute.

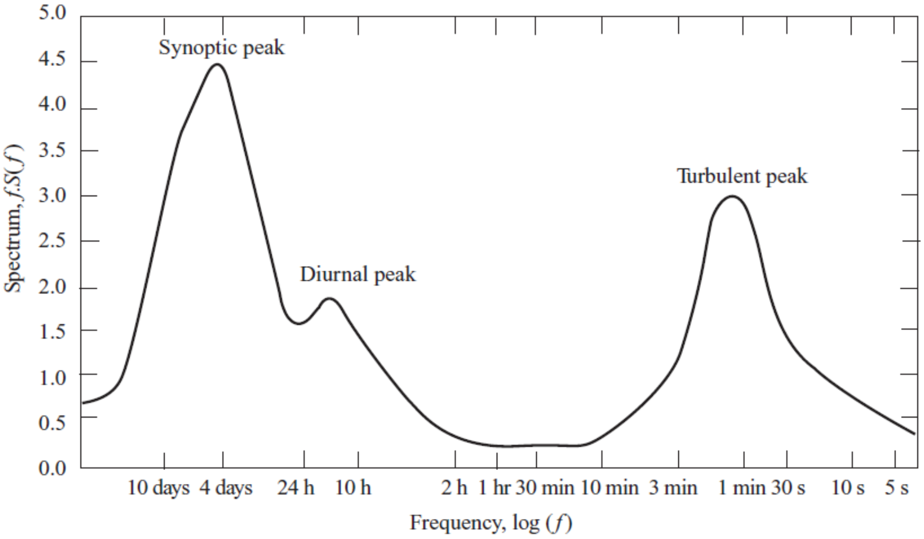

Figure 4.2: Typical wind speed spectrum [31]

4.3.2 Wind Conditions

The wind conditions at a certain height are described by the mean wind speed $\bar{V}$ and the turbulence intensity I. The turbulence intensity is defined as

$$I = \frac{\sigma}{\bar{V}} \tag{4.38}$$

The mean wind speed at hub height V_{hub} is taken as $12\,\text{m/s}$. With the normal wind profile model from the IEC-61400-1 [32] the variation of the mean wind speed with height can be determined:

$$V(z) = V_{\text{hub}} \left(\frac{z}{z_{\text{hub}}} \right)^{\alpha} \tag{4.39}$$

α is the power law exponent and assumed to be 0.2.

The turbulence standard deviation σ is defined for the normal turbulence model as

$$\sigma = I_{\text{ref}} \left(0.75 V_{\text{hub}} + 5.6\,\text{m/s} \right) \tag{4.40}$$

I_{ref} is the expected value of the hub height turbulence intensity at a 10 minute average wind speed of $15\,\text{m/s}$. In the IEC-61400-1 values for different categories are specified. Category C is chosen, where I_{ref} is given as $12\,\%$, because in an offshore environment lower turbulence characteristics are common.

With $\bar{V}$ and σ known, the turbulence intensity I in equation (4.38) can determined for different heights.

4.3.3 Kaimal Spectrum

To describe the varying and irregular nature of wind, a Kaimal spectrum is employed as it is one of the spectra recommended both in the "Wind Energy Handbook" [31] and in the IEC 61400-1 [32]. The Kaimal spectrum for the longitudinal wind direction is defined in the IEC 61400-1 [32] as

$$\frac{fS_V(f)}{\sigma^2} = \frac{4fL/\bar{V}}{(1 + 6fL/\bar{V}^{5/3})} \tag{4.41}$$

where f is the frequency in Hertz, $S_V(f)$ is the power spectral density function for the longitudinal wind velocity component, L is the longitudinal wind velocity component integral scale parameter, σ is the longitudinal wind velocity standard deviation and $\bar{V}$ is the mean wind speed.

Employing equation (4.38), equation (4.41) becomes

$$S_V(f) = \frac{4I^2L\bar{V}}{(1 + 6fL/\bar{V})^{5/3}} \tag{4.42}$$

L is defined as

$$L = 8.1\Lambda \tag{4.43}$$

where Λ is the turbulence scale parameter at height z and defined as

$$\Lambda = \begin{cases} 0.7z & z \leq 60\,\mathrm{m} \\ 42\,\mathrm{m} & z > 60\,\mathrm{m} \end{cases} \tag{4.44}$$

With $\bar{V}$ and I changing for different heights, a Kalman spectrum has to be calculated for every height of interest. The Kaimal spectrum is calculated in the frequency range from 0.001 Hz to 0.5 Hz in steps of 0.001 Hz. In Figure 4.3 a Kaimal spectrum is plotted at a hub height of 120 m and a wind speed at hub height of 12 m/s.

To obtain a realistic wind speed time series, the same procedure as in Section 4.2.3 for the wave time series is followed. The turbulent wind speed part is defined as

$$\tilde{V}(x,t) = \sum_{i=1}^{n} \tilde{V}_{\mathrm{a}i} \sin(\omega_i t + \epsilon_i) \tag{4.45}$$

The wind speed time series is then obtained as:

$$V(x,t) = \bar{V} + \sum_{i=1}^{n} \tilde{V}_{\mathrm{a}i} \sin(\omega_i t + \epsilon_i) \tag{4.46}$$

From the Kaimal spectrum, the amplitudes $\tilde{V}_{\mathrm{a}i}$ are obtained as function of the wind frequency f:

$$\tilde{V}_{\mathrm{a}i} = \sqrt{2S_V(f_i)\Delta f} \tag{4.47}$$

From the wind time series, the wind loads on the structure can be calculated. The wind load on the turbine tower is a drag force and thus determined by the (linear) drag term of the Morison equation:

$$\mathrm{d}F_\mathrm{D} = \frac{1}{2}\rho_\mathrm{a} C_\mathrm{D} D V_\mathrm{n} |V_\mathrm{n}| \tag{4.48}$$

The density of air ρ_a is 1.225 kg/m^3 and the drag coefficient C_D is assumed to be 0.8. V_n is the wind velocity perpendicular to the element:

$$V_\mathrm{n} = V \sin\varphi \tag{4.49}$$

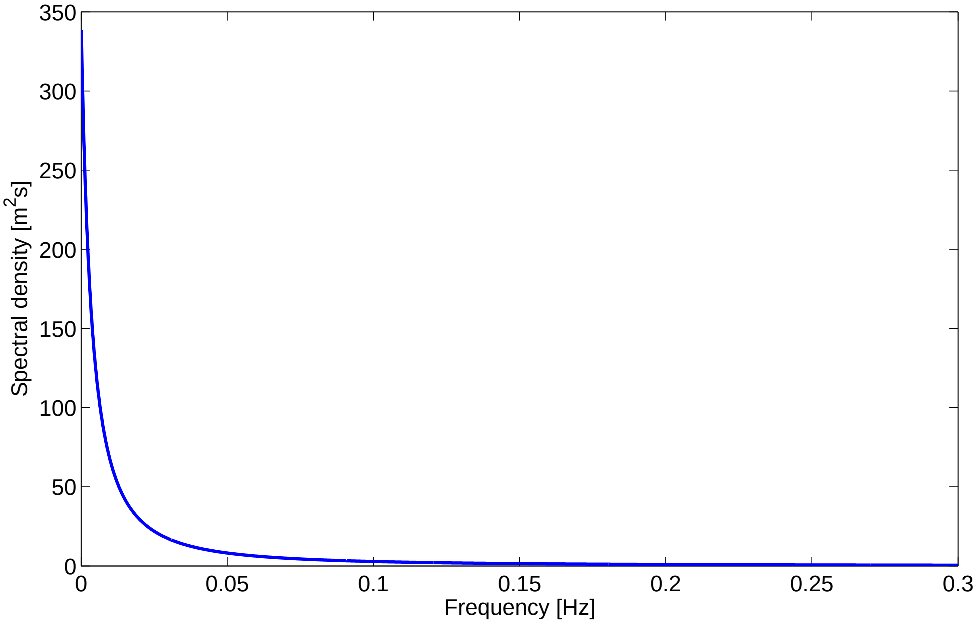

Figure 4.3: Kaimal spectrum of longitudinal wind velocities

4.3.4 Estimation of Thrust Force

The thrust force on the rotor of the wind turbine shall be modelled as a point force at the tower top here. The thrust force is determined with a slightly altered drag force formulation:

$$F_{\mathrm{T}} = \frac{1}{2}\rho_{\mathrm{a}} C_{\mathrm{T}} A_{\mathrm{R}} V_{\mathrm{n}} |V_{\mathrm{n}}| \tag{4.50}$$

where A_{R} is the rotor area. To find a suitable and realistic value for the thrust coefficient C_{T}, the thrust force of the NREL 5 MW offshore wind turbine [11] is determined first.

In order to determine the constant mean part of the thrust force, results from numerical calculations are used. The Computational Fluid Dynamics (CFD) method is believed to be the most advanced and accurate method. The normal force distribution along one blade of the NREL 5 MW wind turbine is shown among other numerical calculations in Figure 4.4. There, the loads on the blade have been determined for a wind speed of 11.4 m/s and a rotational speed of the wind turbine of 12.1 rpm [33]. These conditions correspond to a normal operation condition, where the turbine turns with its rated rotational speed and produces at rated power, but is far from extreme conditions. The values for the distributed normal force are read from the graph. Only values between blade radius 15.85 m and 61.63 m are given. Between 1.5 m and 15.85 m values are linearly interpolated. At the blade tip (63 m) zero force is assumed as the chord is very small there. Since the numerical calculations are performed for one blade, the result has to be multiplied by the number of blades, which in this case is 3:

$$\bar{F}_x = 3 \cdot \bar{F}_{x,blade} \tag{4.51}$$

The determination of the thrust force is shown in Table 4.1. The entries in the column

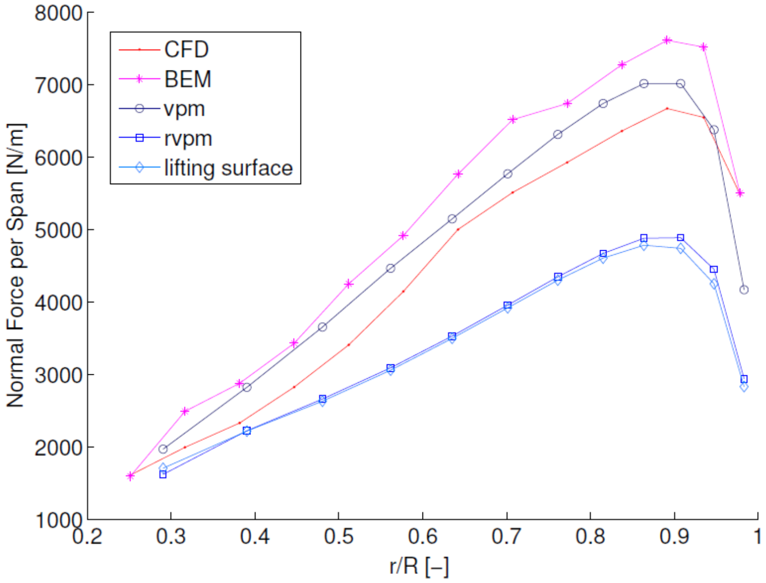

Figure 4.4: Normal force distribution along one blade [33]

labelled “Radius” are the distance from the centre of the hub. The diameter of the hub is 3 m, so that the blade starts at radius 1.5 m. The last three columns in Table 4.1 are not related to nodes as the first three columns, but to sections between two adjacent nodes. That means that e.g. section 1 (in line 2) is defined by the nodes in line 1 and line 2. The total constant mean thrust force at a wind speed of 11.4 m/s is

$$\bar{F}_x = 648.18\,\mathrm{kN} \tag{4.52}$$

This value is used to calibrate C_T in equation (4.50).

4.4 Nodal Forces

A global force vector has to be obtained in order to solve the equations of motion. To this end, the forces are calculated at every node on element level. The beam element nodal force vector is derived in Section A.2 as

$$\mathbf{f} = \int_0^L \mathbf{N}^\mathrm{T} q(x)\,\mathrm{d}x \tag{4.53}$$

The first step in obtaining the nodal forces is the calculation of the hydrodynamic loads or wind loads for every element at the two nodes of each plane frame element. The loads on the nodes are then assumed to vary linearly over the length of the element. Thus, a line load $q(x)$ is obtained:

$$q(x) = f_1 + \frac{f_2 - f_1}{x_2 - x_1}(x - x_1) \tag{4.54}$$

With $x_1 = 0$ and $x_2 = L$, the formula simplifies to

$$q(x) = f_1 + \frac{f_2 - f_1}{L} x \tag{4.55}$$

Inserting equation (4.55) into (4.53) yields:

$$\mathbf{f} = \int_0^L \mathbf{N}^\mathrm{T} \left(f_1 + \frac{f_2 - f_1}{L} x \right) \mathrm{d}x \tag{4.56}$$

The result of the integration are four force components, namely one transverse force component and one moment per node as only loads perpendicular to the elements are considered.

$$f_{z1} = \frac{1}{20}(13f_1 - 3f_2)L \tag{4.57}$$

$$f_{\theta 1} = \frac{1}{60}(7f_1 - 2f_2)L^2 \tag{4.58}$$

$$f_{z2} = \frac{1}{20}(17f_1 - 7f_2)L \tag{4.59}$$

$$f_{\theta 2} = \frac{1}{60}(-8f_1 + 3f_2)L^2 \tag{4.60}$$

The element force vector is then obtained as:

$$\mathbf{f} = \begin{bmatrix} 0 \\ f_{z1} \\ f_{\theta 1} \\ 0 \\ f_{z2} \\ f_{\theta 2} \end{bmatrix} \tag{4.61}$$

To transfer the element force vector, that is given in its local coordinate system, into the global coordinate system the following transformation is applied:

$$\mathbf{f} = \mathbf{T}^\mathrm{T} \overline{\mathbf{f}} \tag{4.62}$$

Finally, the global force vector is obtained by assembling the element force vectors.

Nodes			Sections		
Radius [m]	Norm. radius	Load [kN/m]	Length [m]	Load [kN/m]	Force [kN]
1.50	0.02	0.00			
2.87	0.05	0.15	1.37	0.08	0.10
5.60	0.09	0.46	2.73	0.30	0.83
8.33	0.13	0.76	2.73	0.61	1.66
11.75	0.19	1.14	3.42	0.95	3.26
15.85	0.25	1.60	4.10	1.37	5.62
19.95	0.32	2.00	4.10	1.80	7.38
24.05	0.38	2.30	4.10	2.15	8.82
28.15	0.45	2.90	4.10	2.60	10.66
32.25	0.51	3.40	4.10	3.15	12.92
36.35	0.58	4.10	4.10	3.75	15.38
40.45	0.64	5.00	4.10	4.55	18.66
44.55	0.71	5.50	4.10	5.25	21.53
48.65	0.77	6.00	4.10	5.75	23.58
52.75	0.84	6.40	4.10	6.20	25.42
56.17	0.89	6.60	3.42	6.50	22.23
58.90	0.93	6.50	2.73	6.55	17.88
61.63	0.98	5.50	2.73	6.00	16.38
63.00	1.00	0.00	1.37	2.75	3.77
Total normal force per blade					216.06
Total thrust force					648.18

Table 4.1: Calculation of the thrust force for the NREL wind turbine

5 State-Space Representation and State Estimation

5.1 Introduction

In the previous chapters the support structure model has been set up and explained. The process of describing a system or a real structure by mathematical equations is called system modelling and represents the first step when the response of such a structure shall be obtained. Here, the whole model can be described by one equation:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{5.1}$$

In fact, the equation of motion of the support structure is a system of n_{DOF} non-homogeneous second order linear differential equations. These equations describe the dynamic characteristics of the support structure.

In this chapter, the solution of equation (5.1) is presented. To this end, the so-called state-space representation is introduced. The state-space representation describes the physical system by input, output and state variables, which are related by first-order differential equations. It provides a convenient and straight-forward model to analyse dynamic input-output systems. The term states refers to the variables that fully describe the system (e.g. displacements, velocities).

Furthermore, the state-space representation forms the basis for the application of the Kalman filter, which accounts for the shortcomings of purely deterministic systems. The Kalman filter provides an optimal estimate of the state of a system based on input, model and measurement information. Based on the state estimate by the Kalman filter, the response of the support structure can be predicted.

5.2 State-Space Representation

5.2.1 Deterministic Continuous-Time State-Space Model

The classical continuous-time state-space formulation consists of a state equation and an observation equation:

$$\dot{\mathbf{x}} = \mathbf{A}_{\mathrm{c}}\mathbf{x} + \mathbf{B}_{\mathrm{c}}\mathbf{p} \tag{5.2}$$

$$\mathbf{d} = \mathbf{G}_{\mathrm{c}}\mathbf{x} + \mathbf{J}_{\mathrm{c}}\mathbf{p} \tag{5.3}$$

In equations (5.2) and (5.3), $\mathbf{x}$ is the state vector, a set of variables describing the system's state at time t, $\mathbf{p}$ is the input vector representing the externally applied forces, $\dot{\mathbf{x}}$ is the time derivative of the state vector and $\mathbf{d}$ is the output vector, where the measured data is stored.

For a system with n_p inputs, n_d outputs and n_s state variables, the sizes of the matrices and vectors in equations (5.2) and (5.3) are listed in Table 5.1.

Name	Variable	Size
State vector	$\mathbf{x}$	$n_\mathrm{s} \times 1$
Derivative of state vector	$\dot{\mathbf{x}}$	$n_\mathrm{s} \times 1$
Input vector	$\mathbf{p}$	$n_\mathrm{p} \times 1$
Output vector	$\mathbf{d}$	$n_\mathrm{d} \times 1$
System matrix	$\mathbf{A}_\mathrm{c}$	$n_\mathrm{s} \times n_\mathrm{s}$
Input matrix	$\mathbf{B}_\mathrm{c}$	$n_\mathrm{s} \times n_\mathrm{p}$
Output matrix	$\mathbf{G}_\mathrm{c}$	$n_\mathrm{d} \times n_\mathrm{s}$
Direct transmission matrix	$\mathbf{J}_\mathrm{c}$	$n_\mathrm{d} \times n_\mathrm{p}$

Table 5.1: Matrix and vector sizes for state-space model

The state equation is obtained by transforming the equations of motion (5.1), which are second order differential equations, into first order differential equations. To this end, the displacements $\mathbf{u}(t)$ and the velocities $\dot{\mathbf{u}}(t)$ are defined as the state variables and collected in the state vector $\mathbf{x}(t)$:

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{pmatrix} \tag{5.4}$$

The derivative of the state vector is

$$\dot{\mathbf{x}}(t) = \begin{pmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{u}}(t) \\ \ddot{\mathbf{u}}(t) \end{pmatrix} \tag{5.5}$$

Usually, the equation of motion is modified a bit to set up the state vector. The force vector $\mathbf{f}(t)$ is replaced by the product of a force selection matrix $\mathbf{S}_\mathrm{p}$ with size $n_\mathrm{DOF} \times n_\mathrm{p}$ and the output vector $\mathbf{p}(t)$ [34]:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) = \mathbf{S}_\mathrm{p}\mathbf{p}(t) \tag{5.6}$$

However, in the response calculation of the support structure model, the force selection matrix $\mathbf{S}_\mathrm{p}$ is taken as the unity matrix $\mathbf{I}$ with size $n_\mathrm{DOF} \times n_\mathrm{DOF}$ because forces in all degrees of freedom are considered, $n_\mathrm{p} = n_\mathrm{DOF}$. Thus, in fact, the force vector $\mathbf{f}$ is used in the following.

Equation (5.6) can be rearranged to obtain an expression for the accelerations $\ddot{\mathbf{u}}$

$$\ddot{\mathbf{u}}(t) = -\mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{u}}(t) - \mathbf{M}^{-1}\mathbf{K}\mathbf{u}(t) + \mathbf{M}^{-1}\mathbf{f}(t) \tag{5.7}$$

Thus, the expression for the derivative of the state vector is

$$\dot{\mathbf{x}}(t) = \begin{pmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_2 \\ -\mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{x}}_1 - \mathbf{M}^{-1}\mathbf{K}\mathbf{x}_1 + \mathbf{M}^{-1}\mathbf{f}(t) \end{pmatrix} \tag{5.8}$$

With equation (5.4) the state equation is then determined to be

$$\dot{\mathbf{x}}(t) = \mathbf{A}_\mathrm{c}\mathbf{x}(t) + \mathbf{B}_\mathrm{c}\mathbf{f}(t) \tag{5.9}$$

where the system matrix $\mathbf{A}_\mathrm{c}$ and the input matrix $\mathbf{B}_\mathrm{c}$ are defined as

$$\mathbf{A}_\mathrm{c} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{5.10}$$

$$\mathbf{B}_\mathrm{c} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \tag{5.11}$$

Since possible measurement quantities are accelerations, velocities and displacements, the observation equation is given as

$$\mathbf{d}(t) = \mathbf{S}_\mathrm{a}\ddot{\mathbf{u}}(t) + \mathbf{S}_\mathrm{v}\dot{\mathbf{u}}(t) + \mathbf{S}_\mathrm{d}\mathbf{u}(t) \tag{5.12}$$

$\mathbf{S}_\mathrm{a}$, $\mathbf{S}_\mathrm{v}$ and $\mathbf{S}_\mathrm{d}$ are selection matrices for acceleration, velocities and displacements with size $n_\mathrm{d} \times n_\mathrm{DOF}$, respectively. The selection matrices specify the measurement locations, so that the observation equation in this form collects the measured data at those degrees of freedom. With the definitions from equations (5.4) and (5.5) equation (5.12) becomes

$$\mathbf{d}(t) = \mathbf{S}_\mathrm{a}\dot{\mathbf{x}}_2 + \mathbf{S}_\mathrm{v}\dot{\mathbf{x}}_1 + \mathbf{S}_\mathrm{d}\mathbf{x}_1 \tag{5.13}$$

The observation equation in state-space form is then determined to be

$$\mathbf{d}(t) = \mathbf{G}_\mathrm{c}\mathbf{x}(t) + \mathbf{J}_\mathrm{c}\mathbf{f}(t) \tag{5.14}$$

where the output matrix $\mathbf{G}_\mathrm{c}$ and the direct transmission matrix $\mathbf{J}_\mathrm{c}$ are

$$\mathbf{G}_\mathrm{c} = \begin{bmatrix} \mathbf{S}_\mathrm{d} - \mathbf{S}_\mathrm{a}\mathbf{M}^{-1}\mathbf{K} & \mathbf{S}_\mathrm{v} - \mathbf{S}_\mathrm{a}\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{5.15}$$

$$\mathbf{J}_\mathrm{c} = \mathbf{S}_\mathrm{a}\mathbf{M}^{-1} \tag{5.16}$$

When solving the support structure model, the outputs at all degrees of freedom are considered, $n_\mathrm{d} = n_\mathrm{DOF}$. This implies that the selection matrices change their size as well. The matrices and vectors characterising the state-space representation are summarised in Table 5.2.

Name	Variable	Size
State vector	$\mathbf{x}$	$2n_{\text{DOF}} \times 1$
Derivative of state vector	$\dot{\mathbf{x}}$	$2n_{\text{DOF}} \times 1$
Input vector	$\mathbf{f}$	$n_{\text{DOF}} \times 1$
Output vector	$\mathbf{d}$	$n_{\text{DOF}} \times 1$
System matrix	$\mathbf{A}_{\text{c}}$	$2n_{\text{DOF}} \times 2n_{\text{DOF}}$
Input matrix	$\mathbf{B}_{\text{c}}$	$2n_{\text{DOF}} \times n_{\text{DOF}}$
Output matrix	$\mathbf{G}_{\text{c}}$	$n_{\text{DOF}} \times 2n_{\text{DOF}}$
Direct transmission matrix	$\mathbf{J}_{\text{c}}$	$n_{\text{DOF}} \times n_{\text{DOF}}$
Acceleration selection matrix	$\mathbf{S}_{\text{a}}$	$n_{\text{DOF}} \times n_{\text{DOF}}$
Velocity selection matrix	$\mathbf{S}_{\text{v}}$	$n_{\text{DOF}} \times n_{\text{DOF}}$
Displacement selection matrix	$\mathbf{S}_{\text{d}}$	$n_{\text{DOF}} \times n_{\text{DOF}}$

Table 5.2: Matrix and vector sizes for state-space model as applied in the solution of the system

5.2.2 Deterministic Discrete-Time State-Space Model

The derivations so far assumed a continuous-time model. To obtain a discrete-time state-space model, the time range of interest is discretised at a certain sampling rate $f_{\text{s}} = 1/\Delta t$ into N samples. For the force $\mathbf{f}(t)$ a zero-order hold is assumed. It implies that the sample values are assumed constant during one sample interval.

The discrete-time state-space model can be written as [34]

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{f}_k \tag{5.17}$$

$$\mathbf{d}_k = \mathbf{G}\mathbf{x}_k + \mathbf{J}\mathbf{f}_k \tag{5.18}$$

where $\mathbf{x}_k = \mathbf{x}(k\Delta t)$, $\mathbf{f}_k = \mathbf{f}(k\Delta t)$, $\mathbf{d}_k = \mathbf{d}(k\Delta t)$ and $k = 1, ..., N$. The discrete matrices are defined as:

$$\mathbf{A} = \text{e}^{\mathbf{A}_{\text{c}}\Delta t} \tag{5.19}$$

$$\mathbf{B} = (\mathbf{A} - \mathbf{I})\,\mathbf{A}_{\text{c}}^{-1}\mathbf{B}_{\text{c}} \tag{5.20}$$

$$\mathbf{G} = \mathbf{G}_{\text{c}} \tag{5.21}$$

$$\mathbf{J} = \mathbf{J}_{\text{c}} \tag{5.22}$$

To determine the state $\mathbf{x}_{k+1}$ at a time step $k+1$, the previous state $\mathbf{x}_k$ and force $\mathbf{f}_k$ must be known. Hence, initial conditions $\mathbf{x}_0$ are defined. Physically, this means initial conditions for displacements $\mathbf{x}_1 = \mathbf{u}$ and velocities $\mathbf{x}_2 = \dot{\mathbf{u}}$ are imposed. These are collected in a vector

$$\mathbf{x}_0 = \begin{bmatrix} \mathbf{x}_{10} \\ \mathbf{x}_{20} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0 \\ \dot{\mathbf{u}}_0 \end{bmatrix} \tag{5.23}$$

Consequently, the support structure model can be solved with equations (5.17) and (5.18). In practice, this means, that the accelerations, velocities and displacements can be calculated with equation (5.14) when one of the selection matrices $\mathbf{S}_\mathrm{a}$, $\mathbf{S}_\mathrm{v}$ or $\mathbf{S}_\mathrm{d}$ is set to the unity matrix and the others are set to zero.

5.3 State Estimation

5.3.1 Introduction

Deterministic systems analysis has some shortcomings that make a direct translation of results into reality difficult. Hence, stochastic methods are employed to account for the various sources of uncertainty in the analysis. A physical system, in this case the offshore wind turbine support structure, is modelled by mathematical equations to relate known inputs with wanted outputs. These mathematical equations are based on physical laws or empirical testing and describe the system sufficiently accurate for the engineer's purpose. The model of the system is thus never perfect because it only represents certain characteristics of the system and the mathematical model also only depicts an approximation to reality. Since the prediction of the response of the system can not be fully trusted, often the actual system response is measured with measurement devices. Offshore wind turbine support structures are frequently equipped with accelerometers in the wind turbine tower. However, these measurement devices do not provide exact data, because they might be noise corrupted or distorted. Thus, a method is desired that takes all these uncertainties into account and provides an optimal estimate of the state of the structure. The Kalman filter provides such a method.

5.3.2 Combined Deterministic-Stochastic State-Space Model

The two aforementioned sources of uncertainty, namely the noise on the model equations and on the measurement data, are taken into account by introducing two random vectors into the deterministic state-space equations.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{f}_k + \mathbf{w}_k \tag{5.24}$$

$$\mathbf{d}_k = \mathbf{G}\mathbf{x}_k + \mathbf{J}\mathbf{f}_k + \mathbf{v}_k \tag{5.25}$$

The vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise. For the Kalman filter to work, they are assumed to be white, stationary and mutually uncorrelated processes with normal probability distributions. Their mean values are zero and their covariances are denoted as process noise covariance $\mathbf{Q}$ and measurement noise covariance $\mathbf{R}$, respectively.

5.3.3 The Discrete Kalman Filter Algorithm

The Kalman filter is a recursive linear state estimator. For every time step k, the filter estimates the states based on knowledge of the process prior to step k and based on given measurements at step k. The former is denoted the a priori state estimate $\hat{\mathbf{x}}_k^-$, whereas the latter is denoted the a posteriori state estimate $\hat{\mathbf{x}}_k$. The a priori and the a posteriori estimate errors are then defined as:

$$\begin{aligned}\mathbf{e}_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- &\quad (5.26)\\ \mathbf{e}_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k &\quad (5.27)\end{aligned}$$

The a priori and a posteriori error covariances $\mathbf{P}_k^-$ and $\mathbf{P}_k$ are then

$$\begin{aligned}\mathbf{P}_k^- &= \mathbb{E}\left[\mathbf{e}_k^- \mathbf{e}_k^{-\mathrm{T}}\right] &\quad (5.28)\\ \mathbf{P}_k &= \mathbb{E}\left[\mathbf{e}_k \mathbf{e}_k^{\mathrm{T}}\right] &\quad (5.29)\end{aligned}$$

The Kalman filter determines the a posteriori estimate with the following formula [35]:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{d}_k - \mathbf{G}\hat{\mathbf{x}}_{\mathbf{k}}^- - \mathbf{J}\mathbf{f}_k\right) \tag{5.30}$$

The a posteriori state estimate $\hat{\mathbf{x}}_k$ is a linear combination of the a priori state estimate $\hat{\mathbf{x}}_k^-$ and a weighted measurement innovation. The measurement innovation $\mathbf{d}_k - \mathbf{G}\hat{\mathbf{x}}_{\mathbf{k}}^- - \mathbf{J}\mathbf{f}_k$ gives the difference between an actual measurement $\mathbf{d}_k$ and a measurement prediction $\mathbf{G}\hat{\mathbf{x}}_{\mathbf{k}}^- + \mathbf{J}\mathbf{f}_k$. The weighting is performed by the gain matrix $\mathbf{K}_k$, that minimises the a posteriori error covariance $\mathbf{P}_k$. One popular form is given in equation (5.31).

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{G}^{\mathrm{T}} \left(\mathbf{G}\mathbf{P}_k^- \mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1} \tag{5.31}$$

When the measurement error $\mathbf{R}$ is small, the gain matrix weighs the measurement innovation more heavily as the measurement is 'trusted' more. Equivalently, the gain matrix weighs the measurement innovation less heavily when the a priori estimate error is smaller because then the model equations are expected to be relatively accurate.

The Kalman filter algorithm consists of time update and measurement update equations (cf. (5.32) to (5.36)).

Time update equations

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \mathbf{A}\hat{\mathbf{x}}_{k-1}^- + \mathbf{B}\mathbf{f}_{k-1} &\quad (5.32)\\ \mathbf{P}_k^- &= \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q} &\quad (5.33)\end{aligned}$$

Measurement update equations

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{G}^{\mathrm{T}} \left(\mathbf{G}\mathbf{P}_k^- \mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1} &\quad (5.34)\\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{d}_k - \mathbf{G}\hat{\mathbf{x}}_{\mathbf{k}}^- - \mathbf{J}\mathbf{f}_k\right) &\quad (5.35)\\ \mathbf{P}_k &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{G}\right)\mathbf{P}_k^- &\quad (5.36)\end{aligned}$$

In the time update equations, the current state $\hat{\mathbf{x}}_{k-1}$ and error covariance $\mathbf{P}_{k-1}$ are projected forward in time to obtain the a priori estimates $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$ for the next time step k. The equations (5.32) and (5.33) are, in fact, predicting the state at the next time step based on the knowledge of the previous state and the system characteristics expressed by the matrices $\mathbf{A}$ and $\mathbf{B}$. In the measurement update, the a priori estimates are improved or corrected by incorporating an actual (noisy) measurement $\mathbf{d}_k$ to obtain the a posteriori estimates $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$, as has already been explained.

For the algorithm to work, a number of inputs have to be defined. The four matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ have to be determined beforehand by equations (5.19) to (5.22). Whereas the system and the input matrix do not change compared to the deterministic state-space model, the output and direct transmission matrix are adapted because the selection matrices are changed. In accordance with the objective of this thesis, accelerometers are 'placed' on the support structure at the wind turbine tower at a number of locations n_d to measure the response. Thus, the acceleration selection matrix becomes a matrix of size $n_\mathrm{d} \times n_\mathrm{DOF}$, where each row stands for a sensor and the columns for the degrees of freedom measured. A 1 indicates that this degree of freedom is measured, a 0 indicates no measurement. Naturally, there will be only one 1 per row as it is assumed that one sensor can measure only one quantity. The velocity and displacement selection matrices will be zero as those quantities are not measured.

Moreover, the input vector $\mathbf{f}$, the measurement vector $\mathbf{d}$, the process error covariance $\mathbf{Q}$ and the measurement error covariance $\mathbf{R}$ have to be known. The force vector is computed for every time step as described in Chapter 4. The measurement vector is determined by taking the acceleration output from the deterministic state-space model at the 'measured' locations and adding some noise that represents the inaccuracy of the sensors. The noise level is assumed to be 10 percent of the standard deviation of the true accelerations. A comparison between the 'true' accelerations and the measured accelerations is presented in Figure 5.1. For a smoother appearance, the sampling frequency has been increased for the plot.

For the measurement error covariance a slightly different value is chosen as in reality it can not be assumed that the measurement uncertainty is known precisely. The process error covariance is estimated at 10^{-5}.

Furthermore, initial estimates for the state vector $\hat{\mathbf{x}}_0$ and the a posteriori error $\mathbf{P}_0$ have to be assumed. The initial state vector is assumed to be a zero vector and the initial a posteriori error covariance is taken as 10^{-3}. The choice of the latter two variables is not of utmost importance because the initial state estimate is considered as a random variable and under the assumed conditions here the error covariance will stabilise quickly [34] [35].

Having determined all the inputs, the states of the system or support structure model can be estimated by means of the time update and measurement update equations for every time step. For the sake of clarity, the inputs to the Kalman filter a listed in Table 5.3.

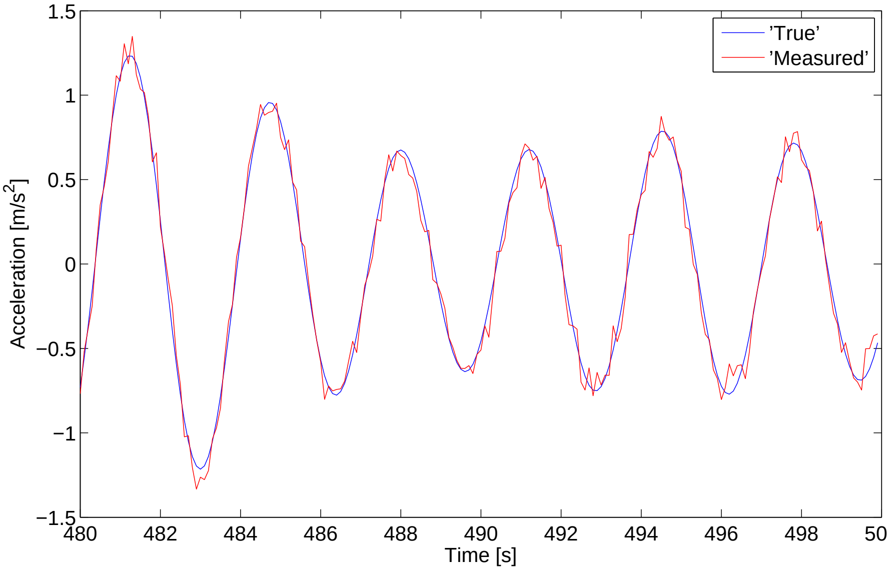

Figure 5.1: Comparison between ‘true’ and ‘measured’ accelerations

Name	Variable	Size
System matrix	$\mathbf{A}$	$2n_{\mathrm{DOF}} \times 2n_{\mathrm{DOF}}$
Input matrix	$\mathbf{B}$	$2n_{\mathrm{DOF}} \times n_{\mathrm{DOF}}$
Output matrix	$\mathbf{G}$	$n_{\mathrm{d}} \times 2n_{\mathrm{DOF}}$
Direct transmission matrix	$\mathbf{J}$	$n_{\mathrm{d}} \times n_{\mathrm{DOF}}$
Input vector	$\mathbf{f}$	$n_{\mathrm{DOF}} \times 1$
Output vector	$\mathbf{d}$	$n_{\mathrm{d}} \times 1$
Process noise covariance	$\mathbf{Q}$	$2n_{\mathrm{DOF}} \times 2n_{\mathrm{DOF}}$
Measurement noise covariance	$\mathbf{R}$	$n_{\mathrm{d}} \times n_{\mathrm{d}}$
A posteriori error covariance	$\mathbf{P}_0$	$2n_{\mathrm{DOF}} \times 2n_{\mathrm{DOF}}$
Initial state estimate	$\hat{\mathbf{x}}_0$	$2n_{\mathrm{DOF}} \times 1$

Table 5.3: Inputs for the Kalman filter

5.4 Response Prediction

The Kalman filter algorithm estimates the states $\hat{\mathbf{x}}_k$ for every time step k. Now, the response of the support structure model can be predicted at any (unmeasured) degree of freedom with the observation equation:

$$\mathbf{d}_k = \mathbf{G}\hat{\mathbf{x}}_k + \mathbf{J}\mathbf{f}_k \tag{5.37}$$

The accelerations are obtained by setting the acceleration selection matrix $\mathbf{S}_\mathrm{a}$ with size $n_\mathrm{DOF} \times n_\mathrm{DOF}$ to unity and the other two selection matrices to zero. Then, the output and direct transmission matrix $\mathbf{G}$ and $\mathbf{J}$ are calculated with equations (5.15) and (5.16) and the accelerations can be obtained with above formula. Equivalently, the velocities and displacements can be determined.

To prove the functionality of the Kalman filter algorithm, the ‘true’ and the predicted accelerations at a measured location in the turbine tower are plotted together in Figure 5.2. For a properly working filter, the two time series should be almost the same. This can be verified from Figure 5.2.

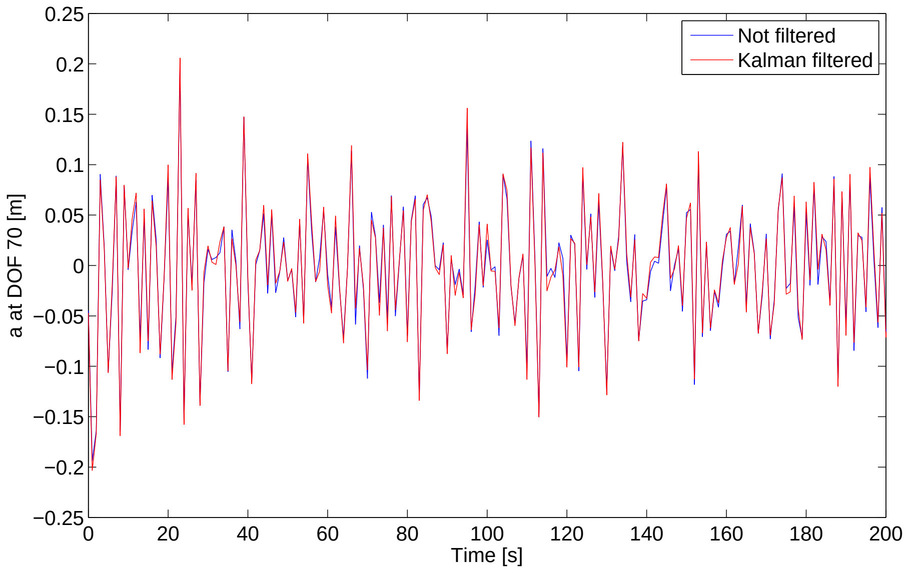

Figure 5.2: Comparison between ‘true’ and predicted accelerations

5.5 Conclusions

For clarity, the steps undertaken in this chapter are recapitulated:

- The response of the support structure model developed in Chapters 2 to 4 shall be investigated by acceleration measurements at a few locations in the turbine tower. To this end, the model is transformed into a *deterministic state-space* *model* to obtain the 'true' accelerations at the measurement locations. These accelerations are called the 'true' accelerations because the deterministic model assumes that the model equations are completely accurate. This assumption, however, does not hold in reality.

- Therefore, the deterministic state-space model is extended to a *combined deterministic-stochastic state-space model* that accounts for the uncertainties in the model equations and in the measurements. The combined deterministic-stochastic state-space model is solved by means of the Kalman filter algorithm that gives optimal estimates of the state of the model for every time step.
- In the last step, the *response prediction*, the accelerations at any degree of freedom are predicted from the state estimates.

6 Damage Identification Based on Operational Vibrations

6.1 Introduction

In this chapter the dynamic properties and the response of the modelled jacket support structure, that has been developed along the lines of the UpWind reference jacket support structure, are investigated. Two main parameters in the model are changed during the analysis. On the one hand, the crack depth at the specific location is varied from 0.01 m to 0.4 m, where the bigger values represents a crack through the half-width of the brace. On the other hand, the number of degrees of freedom is increased once in order to describe the local modes more accurately. In Table 6.1 the different crack depths are listed together with the crack coefficient. For the relative stiffness change due to the crack, confer Table 3.1.

Denotation	Crack depth	Rel. crack depth	Crack coefficient
	a [m]	a/D_{e} [–]	K
Cracked 0	0.00	0.0000	0
Cracked 1	0.01	0.0125	0.00158
Cracked 2	0.02	0.0250	0.00871
Cracked 3	0.05	0.0625	0.02155
Cracked 4	0.10	0.1250	0.03208
Cracked 5	0.20	0.2500	0.04381
Cracked 6	0.30	0.3750	0.05021
Cracked 7	0.40	0.5000	0.05607

Table 6.1: Different jacket models used in the analysis

The locations where measurements are conducted are distributed over the height of turbine tower. One accelerometer measures the horizontal accelerations at the tower top, one accelerometer measures at approximately half height and one accelerometer measures at the tower bottom.

In a first step, the eigenvalues and natural frequencies and their corresponding mode shapes (eigenvectors) are determined. A comparison between the cracked and the uncracked structure is conducted. Subsequently, the structure's response to ambient excitation by wind and waves is considered. Finally, a forced harmonic vibration is imposed on the support structure and the effect is investigated.

6.2 Eigenvalue Analysis

6.2.1 Theory

For the determination of the eigenvalues of a dynamic system free undamped vibrations are considered ($\mathbf{C} = 0$, $\mathbf{f} = 0$).

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = 0 \tag{6.1}$$

When assuming harmonic vibration

$$\mathbf{u} = \hat{\mathbf{u}} \sin(\omega t + \varphi) \tag{6.2}$$

the solution becomes

$$\left(\mathbf{K} - \omega^2 \mathbf{M}\right) \hat{\mathbf{u}} = 0 \tag{6.3}$$

Rewriting equation (6.3) yields

$$\mathbf{KV} = \mathbf{MVD} \tag{6.4}$$

where $\mathbf{V}$ is a square matrix of size n x n containing the eigenvectors $\hat{\mathbf{u}}_i$ as columns

$$\mathbf{V} = \begin{bmatrix} \hat{\mathbf{u}}_1 & \dots & \hat{\mathbf{u}}_n \end{bmatrix} \tag{6.5}$$

and $\mathbf{D}$ is a square matrix of size n x n containing the eigenvalues $\lambda_i = \omega_i^2$ on the diagonal

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \tag{6.6}$$

Equation (6.3) or (6.4) are the generalised eigenvalue problem, in which ω^2 represents the eigenvalue and $\hat{\mathbf{u}}$ the corresponding eigenvector. It has a non-trivial solution if, and only if, the determinant of the coefficient matrix is zero:

$$\det\left(\mathbf{K} - \mathbf{DM}\right) = 0 \tag{6.7}$$

Equation (6.7) is the characteristic polynomial of the generalised eigenvalue problem. For systems with n degrees of freedom n eigenvalues λ_i are found. The corresponding eigenvectors $\hat{\mathbf{u}}$ are obtained applying equation (6.3).

6.2.2 UpWind Reference Jacket

Jacket with 96 DOFs

The built-in Matlab function `eig` is used to solve the generalised eigenvalue problem described in the previous section. The UpWind reference jacket is modelled with 96 degrees of freedom, thus 96 eigenvalues and eigenvectors will be obtained. Usually the lowest natural frequencies are of most interest because the structure's response is dominated by the lowest modes and the excitation frequencies in an offshore environment with wind and waves are relatively low, so that structures are designed to have a fundamental frequency above the excitation frequencies.

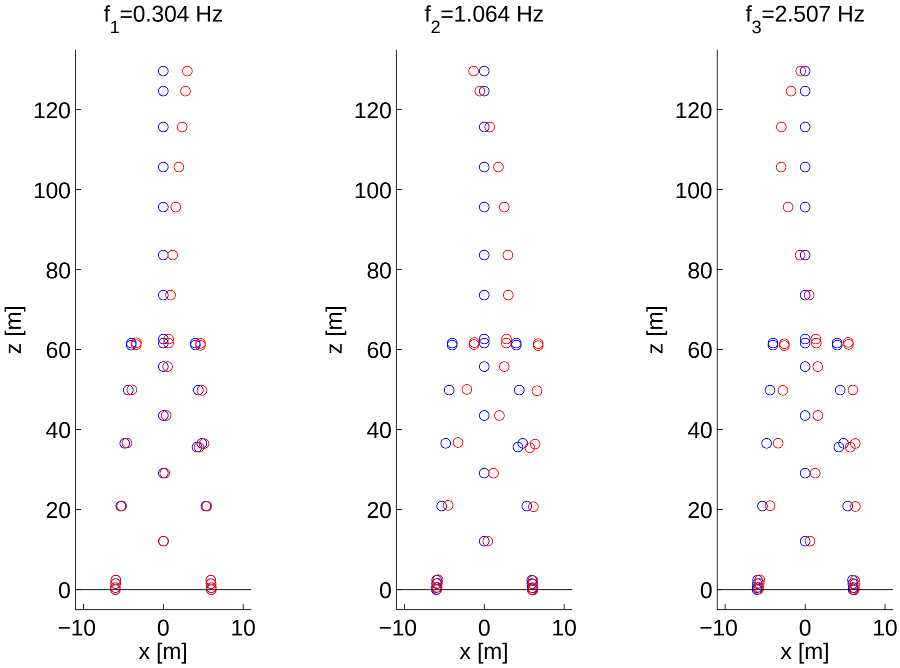

Figure 6.1: First three mode shapes of the undamaged support structure

Modelling the UpWind reference jacket in two dimension has important implications on the design. Since in this investigation the support structure's dynamic behaviour is of primary interest, the support structure model is adapted to match the first three eigenfrequencies of the UpWind reference jacket [36] (see Table 6.4). In this way, realistic response spectra can be obtained because the support structure lies above the main excitation frequencies.

The first three eigenfrequencies of the undamaged support structure are shown in Figure 6.1. Qualitatively, the mode shapes look familiar from, for example, the dynamic investigation of a cantilever beam. The jacket substructure, however, is apparently stiffer than the turbine tower leading to the larger observed displacements in the tower.

The UpWind reference jacket's first eigenfrequencies include tower fore-aft and side-to-side mode shapes as well as torsional mode shapes. The mode shapes in Figure 6.1 and eigenfrequencies in Table 6.4 refer only to the so-called tower fore-aft modes because the support structure is modelled in 2D and thus neglects the effects in the third dimension.

In Figure 6.2 the first 50 eigenfrequencies of the undamaged and the (most) damaged support structure are plotted. The absolute difference between the eigenfrequencies is almost not observable, so the relative deviation is computed and plotted in Figure 6.3.

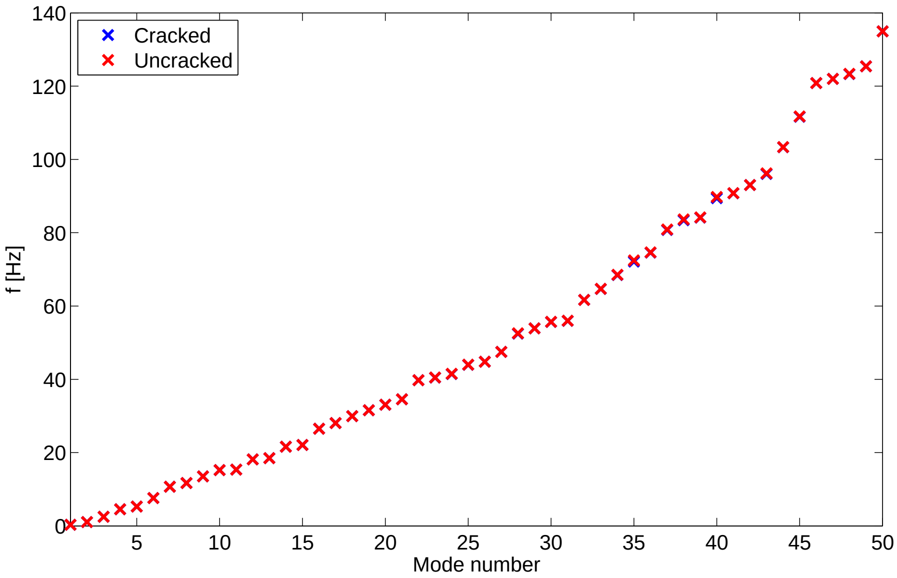

Figure 6.2: Damaged and undamaged structure's eigenfrequencies

The relative deviation between the eigenfrequencies of the undamaged support structure ω_{nu} and the eigenfrequencies of the damaged support structure ω_{nc} are calculated as follows:

$$\frac{\omega_{nu} - \omega_{nc}}{\omega_{nu}} \tag{6.8}$$

The lowest modes show almost no deviation from the undamaged structure. Mode 35 shows the highest deviation with a maximum of approximately 0.5 %. A number of modes follow in the range of 0.2 to 0.4 % deviation.

This overall behaviour can be expected as the crack in one of the joints introduces a local decrease of stiffness. This local decrease does not affect the global modes, where the whole structure is moving in a consistent manner.

The modes with the highest deviation are local modes of the brace the crack is located in. This interpretation is supported by a plot of the modes in Figure 6.4. The nodes of the crack element and the brace it is located in are displaced and/or rotated most in their respective modes. The mode shapes look almost the same in the undamaged structure and are therefore not repeated here. Another observation that can be made is the reduction of the eigenfrequencies in the damaged structure. This effect has already been explained in Section 3.2 and is due to the crack's stiffness decrease in the structure.

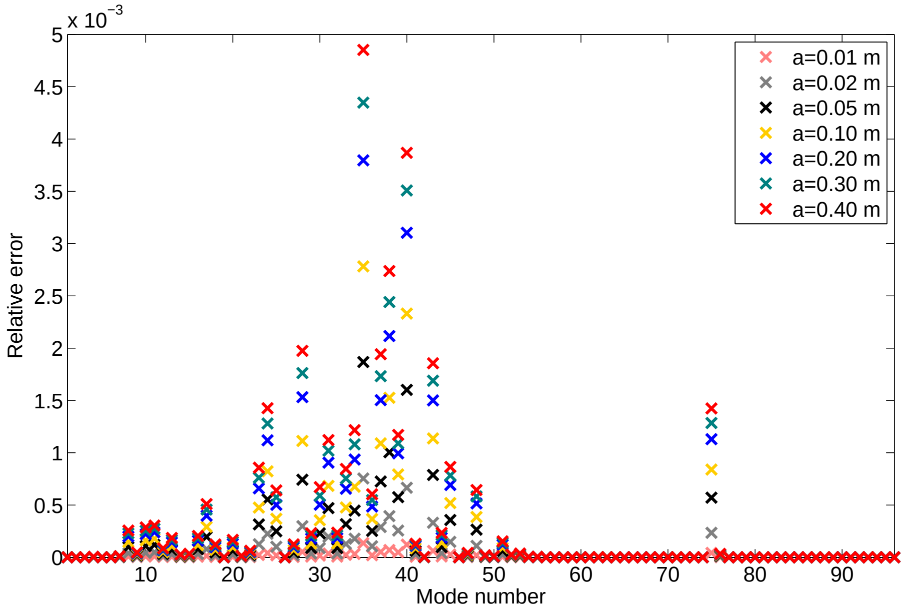

Figure 6.3: Relative difference between damaged and undamaged structure's eigenfrequencies

	Eigenfrequency [Hz]		Rel. dev. [–]
	Undamaged	Damaged	
f_1	0.304	0.304	$5.6 \cdot 10^{-7}$
f_2	1.064	1.064	$4.3 \cdot 10^{-11}$
f_3	2.507	2.507	$2.4 \cdot 10^{-7}$
f_{28}	52.562	52.459	$2.0 \cdot 10^{-3}$
f_{35}	72.456	72.104	$4.9 \cdot 10^{-3}$
f_{38}	83.628	83.399	$2.7 \cdot 10^{-3}$
f_{40}	89.755	89.408	$3.9 \cdot 10^{-3}$

Table 6.2: Comparison of eigenfrequencies between damaged and undamaged jacket

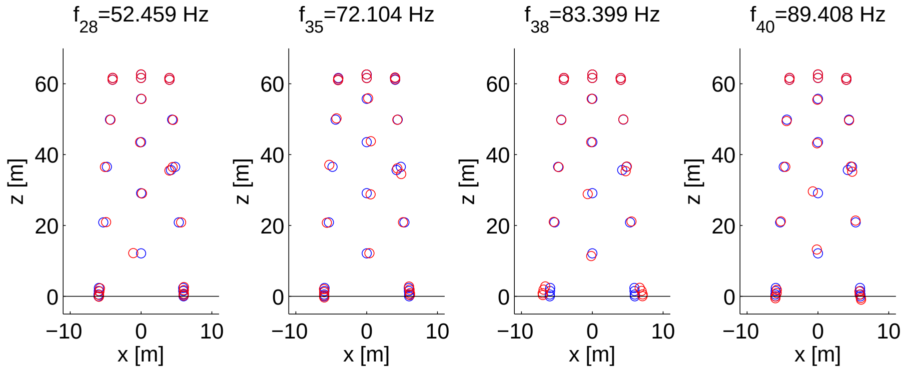

Figure 6.4: Mode shapes and eigenfrequencies of higher local modes in damaged structure

Jacket with 312 DOFs

Modelling the jacket's braces and legs with more nodes and thus more degrees of freedom is expected to increase the accuracy of the analysis because local brace or leg modes can be described. In Figure 6.5 the first three mode shapes are plotted again. In that figure the locations of the additional nodes can be clearly seen.

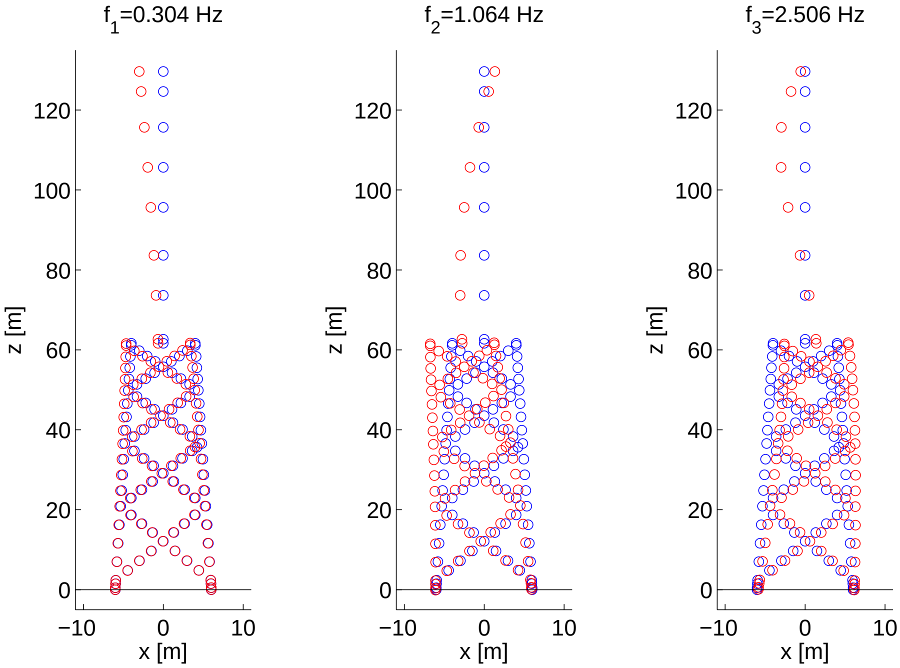

Figure 6.5: First three mode shapes of the undamaged support structure

Again, the relative deviation of the damaged structure from the undamaged structure is compared. Figure 6.6 presents the result of this comparison. In Table 6.3 the relative deviation of the first three modes and the modes with the highest deviation are presented. As before with the smaller model, the first three eigenfrequencies hardly change. The four modes with the largest relative deviation from the undamaged structure are shown in Figures 6.7 and 6.8. Now, it can be seen more clearly, that the modes that are affected most are indeed higher local modes. An interesting distinction can be made. Whereas the modes in Figure 6.7 seem to be local modes of the brace the crack is located in, the modes in Figure 6.8 seem to local 'crack modes' of the cracked finite element with much higher frequencies.

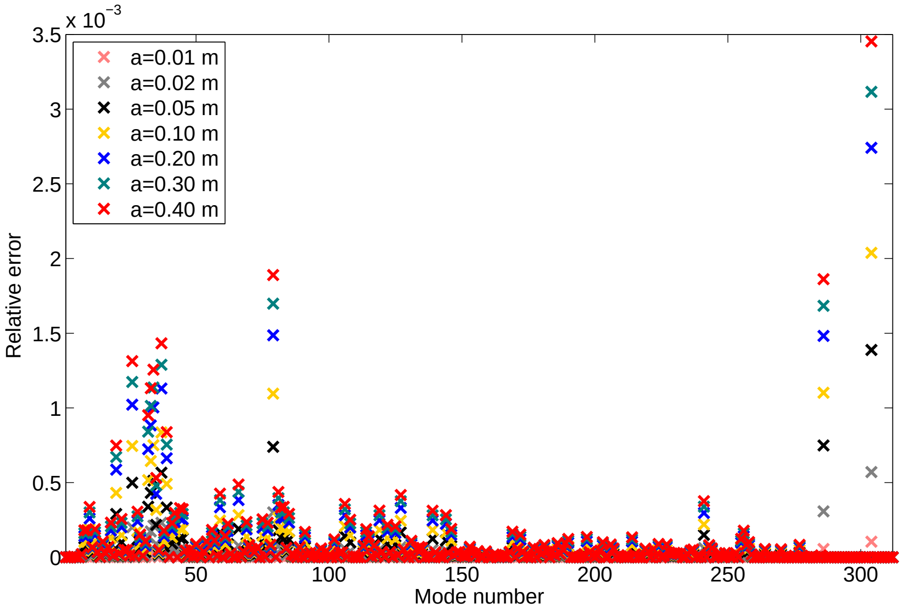

Figure 6.6: Relative difference between damaged and undamaged structure's eigenfrequencies

	Eigenfrequency [Hz]		Rel. dev. [–]
	Undamaged	Damaged	
f_1	0.304	0.304	$5.6 \cdot 10^{-7}$
f_2	1.064	1.063	$1.7 \cdot 10^{-10}$
f_3	2.506	2.506	$2.5 \cdot 10^{-7}$
f_{37}	49.916	49.845	$1.4 \cdot 10^{-3}$
f_{79}	134.825	134.570	$1.9 \cdot 10^{-3}$
f_{286}	1798.526	1795.179	$1.9 \cdot 10^{-3}$
f_{304}	3771.164	3758.142	$3.5 \cdot 10^{-3}$

Table 6.3: Comparison of eigenfrequencies between damaged and undamaged jacket

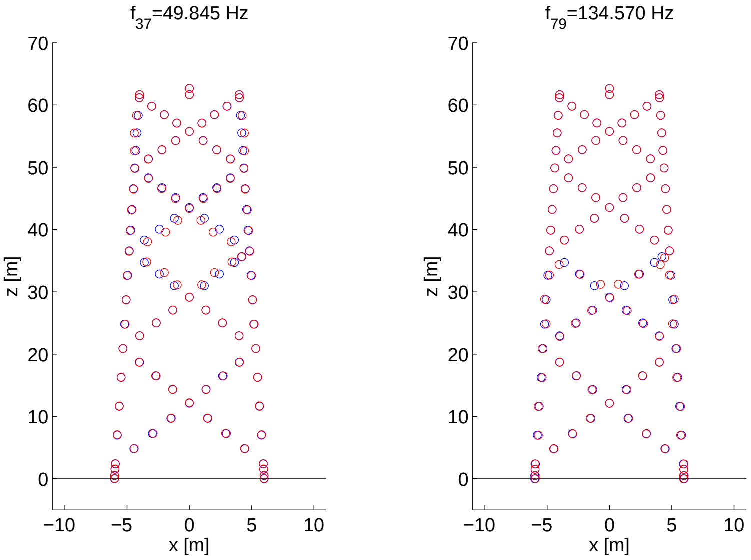

Figure 6.7: Mode shapes and eigenfrequencies of higher local modes in damaged structure

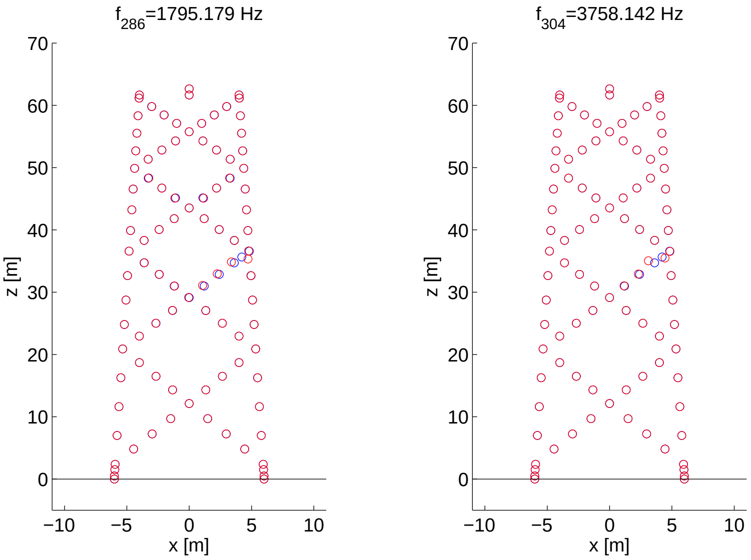

Figure 6.8: Mode shapes and eigenfrequencies of higher local modes in damaged structure

Comparison

In Figure 6.9 the eigenfrequencies of the undamaged support structure are shown for the model with 96 and 312 degrees of freedom. The eigenfrequencies cover approximately the same range, but, naturally, the larger model has more eigenfrequencies than the smaller model. That is, among others, due to the fact that a lot more and higher local brace modes can be described. This fact also leads to the large differences between the modes that are most affected by the presence of the crack. Tables 6.2 and 6.3 document this fact.

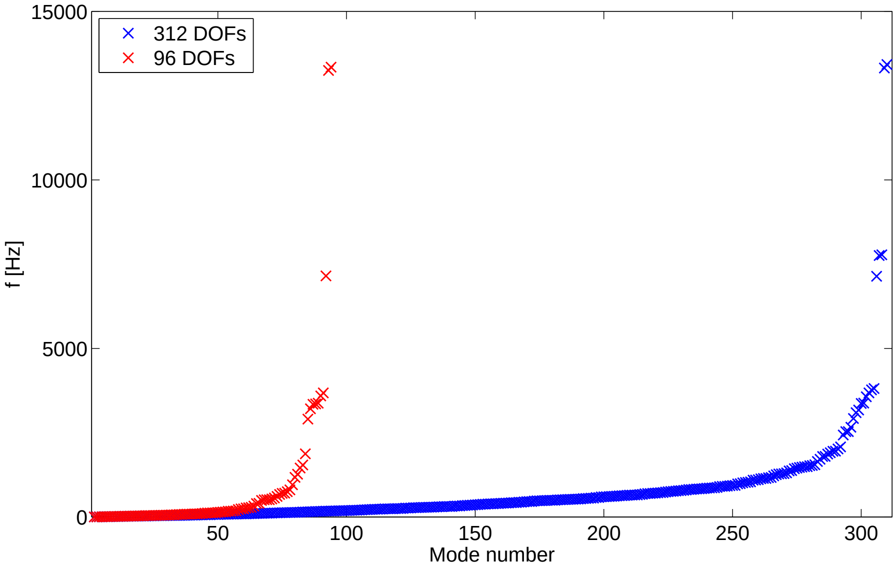

Figure 6.9: Eigenfrequencies of structure with 96 DOFs and 312 DOFs

The lowest three eigenfrequencies are almost the same for both models as can be seen in Table 6.4. Based on different assumptions regarding, among others, the flexibility of the foundation and marine growth, a range of eigenfrequencies is given for the UpWind reference jacket [36].

	Eigenfrequency [Hz]		
	UpWind	Model 1	Model 2
1st	0.291–0.310	0.304	0.304
2nd	0.813–1.104	1.063	1.063
3rd	2.001–2.622	2.507	2.506

Table 6.4: Comparison of eigenfrequencies in the model and the UpWind reference jacket

6.2.3 Conclusions

The introduction of a crack in the support structure model affects the structure's eigenfrequencies. Whereas the first modes of the support structure show almost no deviation in the eigenfrequencies, there exist higher modes that are local brace or even crack modes and that show small deviations from the uncracked structure. Since the model with 96 degrees of freedom did not represent those local modes well, a model with 312 degrees of freedom was modelled. The more precise description of the local brace modes lead to an much better description of the local modes and is also assumed to be much more accurate.

The model with more degrees of freedom seems to have no influence on the lowest eigenfrequencies compared to the model with less degrees of freedom. Therefore, the model with 96 degrees of freedom is used in the following for the ambient and forced vibration tests because only the frequency range up to 0.5 Hz is excited.

6.3 Ambient Vibration Testing

6.3.1 Introduction

Ambient vibration testing uses the natural excitation sources like wind and waves and the operational vibrations due to the interaction of the wind flow and the (moving) structure to determine the modal characteristics of the structure. Accelerometers measure the response spectra at the wind turbine tower. Kalman filtering is applied on the measurements and the model and the response is predicted for the whole structure. The excitation frequencies range from 0.001 Hz to 0.5 Hz, thus only the first eigenfrequency is excited.

It was concluded from the eigenvalue analysis that the difference between the lowest eigenfrequencies of the undamaged support structure and the damaged support structure is negligible and thus probably not detectable. However, there exists the possibility that a difference between the damaged and the undamaged support structure can be seen in the response spectra, for example in the height of the peaks.

6.3.2 Time Series

As a first step, the response in the time domain is considered. Wave, wind, structure's displacement, velocities and acceleration time series for a certain degree of freedom are plotted. The first plot shows the wave time series. The waves have zero mean and varying amplitude up to around 6 m as has been defined in the wave spectrum. The wind time series in the second plot varies around a mean value of 12 m/s and varies between approximately 7 and 17 m/s. The last three plots show the structure's response to this loading at the wind turbine tower bottom in the horizontal direction. The structure's displacement has a small positive mean value, indicating a mean force that acts on the structure (due to wind). The structure's velocity and acceleration vary around zero, The acceleration's amplitude is larger than the velocity's amplitude, which, in turn, is larger then the displacement's amplitude.

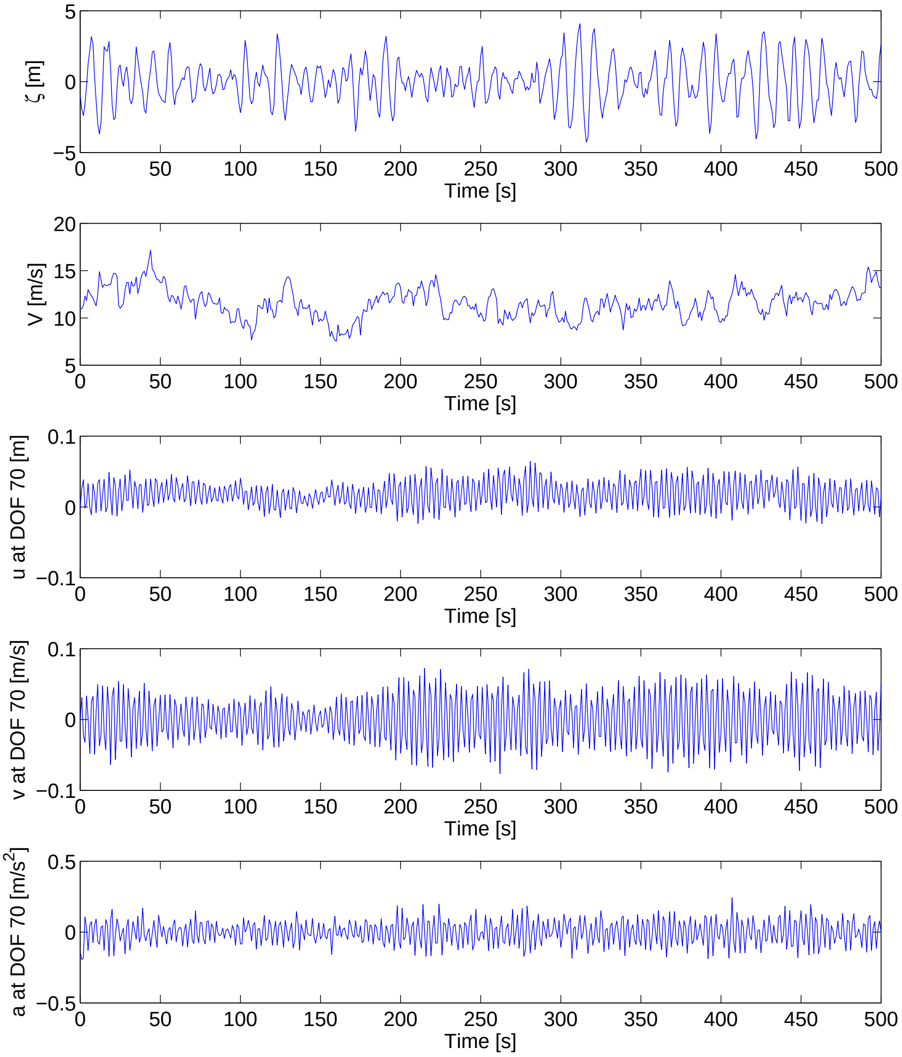

Figure 6.10: Excitation and response time series of undamaged support structure at tower bottom

6.3.3 Frequency Spectra

The excitations and responses shown in the previous section in time domain are transformed into frequency domain to obtain frequency spectra. In Appendix D important parameters are explained. A representation in frequency domain allows an easy insight into the frequency content of the response (and the excitation).

Figure 6.11 presents the frequency spectra of wave and wind time series in the first two plots. The spectra resemble closely the spectra (JONSWAP and Kaimal) that were used to create the time series. The response spectra all show a clear peak around 0.3 Hz, where the first eigenfrequency of the structure is located. The displacement spectrum also contains some low frequency content due to the wind excitation.

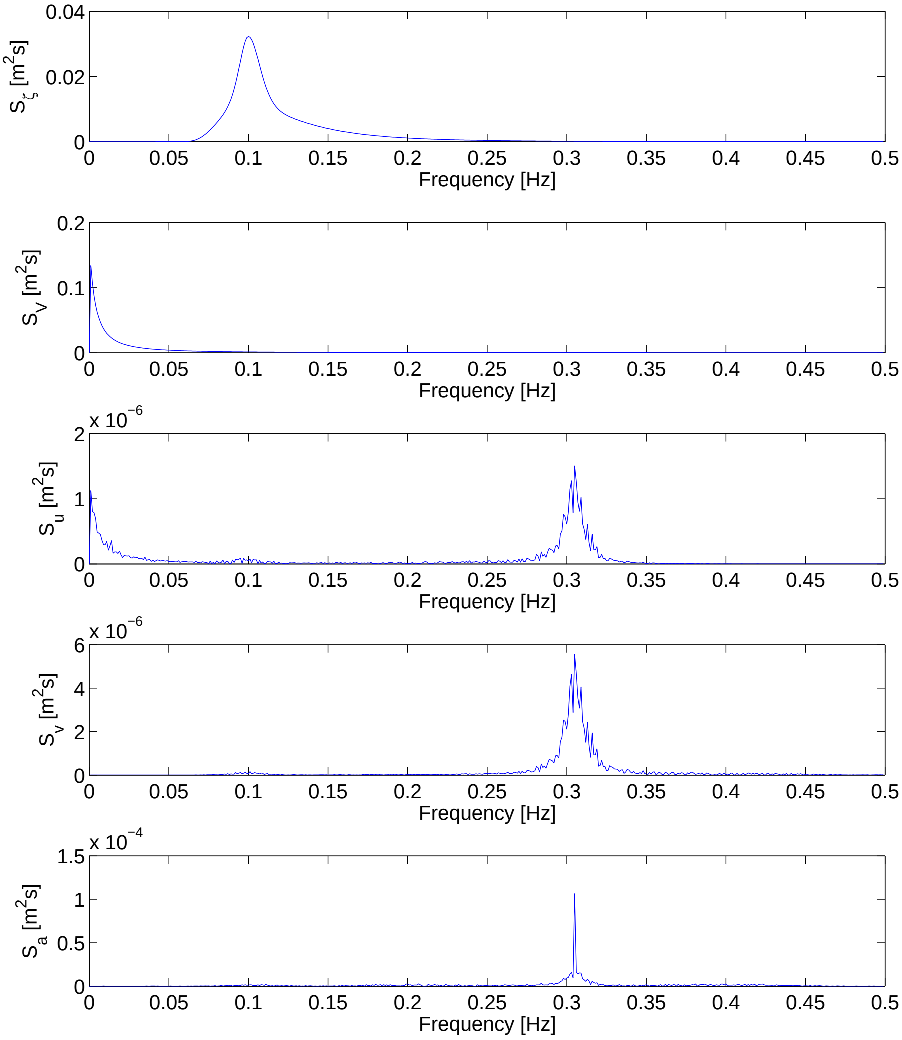

Figure 6.11: Excitation and response frequency spectra of undamaged support structure at tower bottom

In Figure 6.12 the acceleration spectrum at the tower bottom is plotted for the Kalman filtered response and for different damage conditions. In Figure 6.13 only the range around the first eigenfrequency peak is presented for better visibility. Some differences can be observed between the different damage conditions, however they seem to be random. This becomes clearer when Figures 6.14 and 6.15 are considered, where the same spectra are plotted for the unfiltered response. As can be seen, hardly any difference exists between the damaged and undamaged structures. Thus, the greater variability for the filtered spectra can be explained with the Kalman filter algorithm that gives an optimal estimate, however not always the same. The scatter due to the

Kalman filter can be reduced if either the measurement uncertainty of the sensors is reduced or the model accuracy is increased or both.

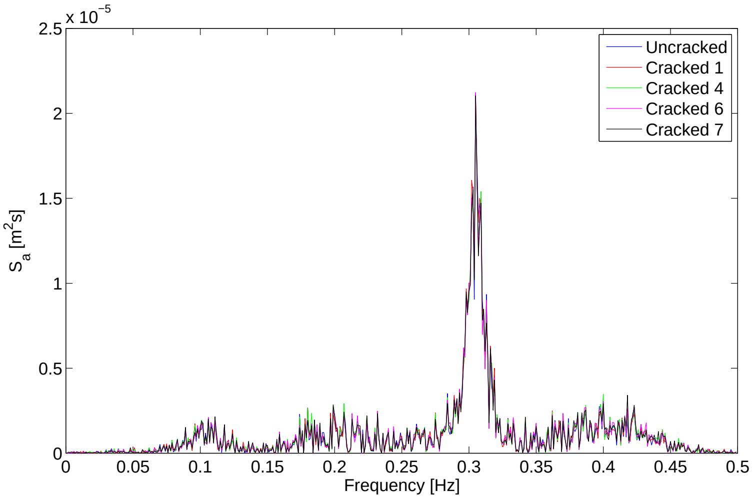

Figure 6.12: Comparison of filtered acceleration response frequency spectra at tower bottom

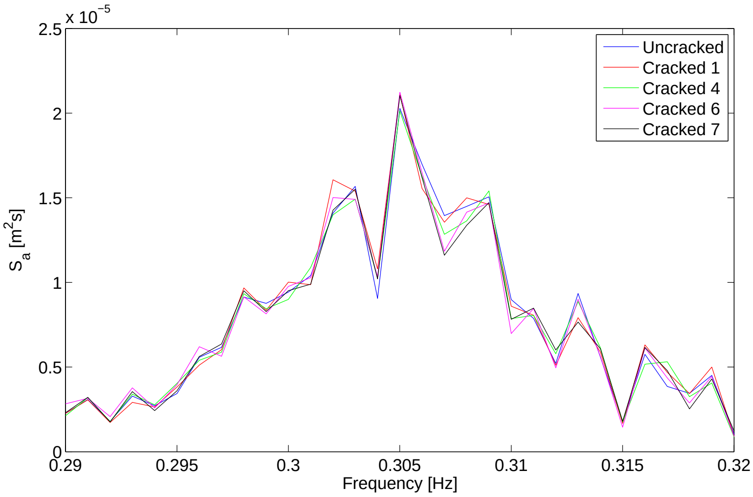

Figure 6.13: Detail around first eigenfrequency of Figure 6.12

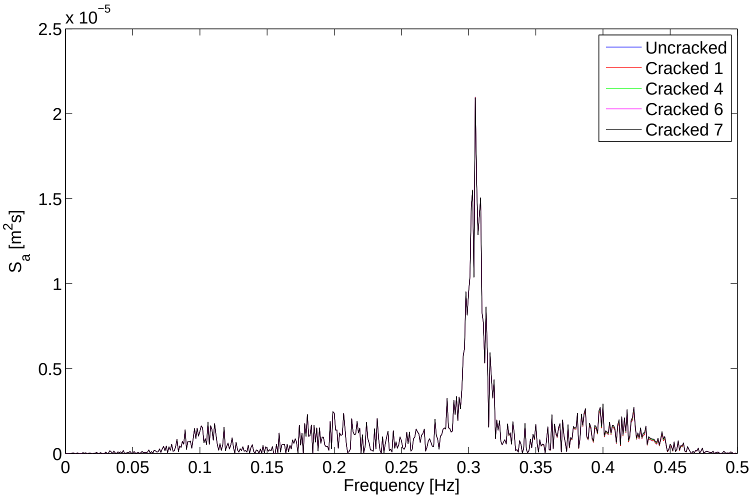

Figure 6.14: Comparison of acceleration response frequency spectra at tower bottom

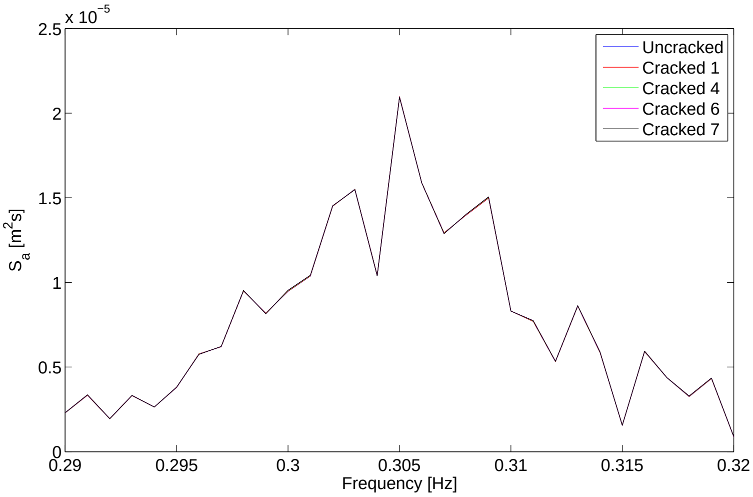

Figure 6.15: Detail around first eigenfrequency of Figure 6.14

6.3.4 Conclusions

As predicted from the results of the eigenvalue analysis, damage identification by ambient vibration testing is not possible for the investigated complex structure as there is hardly any difference visible in the response spectra. Actually, the variability due to the uncertainty of the Kalman filter applied here is larger than the difference between damaged and undamaged structure.

6.4 Forced Vibration Testing

In contrast to ambient vibration testing, forced vibration testing uses an (additional) artificial excitation source to excite the structure at a specific location with a certain frequency. So-called shakers can be employed to excite the structure with a harmonic force.

The harmonic excitation force is applied to the support structure model to investigate how the response or dynamic characteristics are changed. The considered spectrum from 0.001 Hz to 0.5 Hz covers only the first eigenfrequency, where a peak occurs in the response spectra. The objective is thus to obtain a more distinct difference in the response spectral peak of a damaged and an undamaged structure. To this end, the harmonic excitation force is taken with the same frequency as the first eigenfrequency of the structure.

This investigation obviously constitutes a proof of concept as the application of a harmonic excitation close to the first eigenfrequency is a very delicate matter.

The harmonic force is defined by

$$F_{\mathrm{h}} = A_{\mathrm{h}} \sin(\omega_{\mathrm{h}} t) \tag{6.9}$$

where A_{h} is the amplitude of the force and ω_{h} is the frequency of the force:

$$\omega_{\mathrm{h}} = 2\pi f_{\mathrm{h}} \tag{6.10}$$

It is applied at the tower bottom in the horizontal direction as the first eigenfrequency deflects in that direction.

Figure 6.16 and 6.17 show the acceleration power spectrum at the tower bottom in horizontal direction. There is no significant improvement in the detectability of the damage. When these figures are compared to the spectra of the previous section, it is noted that the peak at the first eigenfrequency is increased significantly as can be expected when a harmonic force with that frequency is applied.

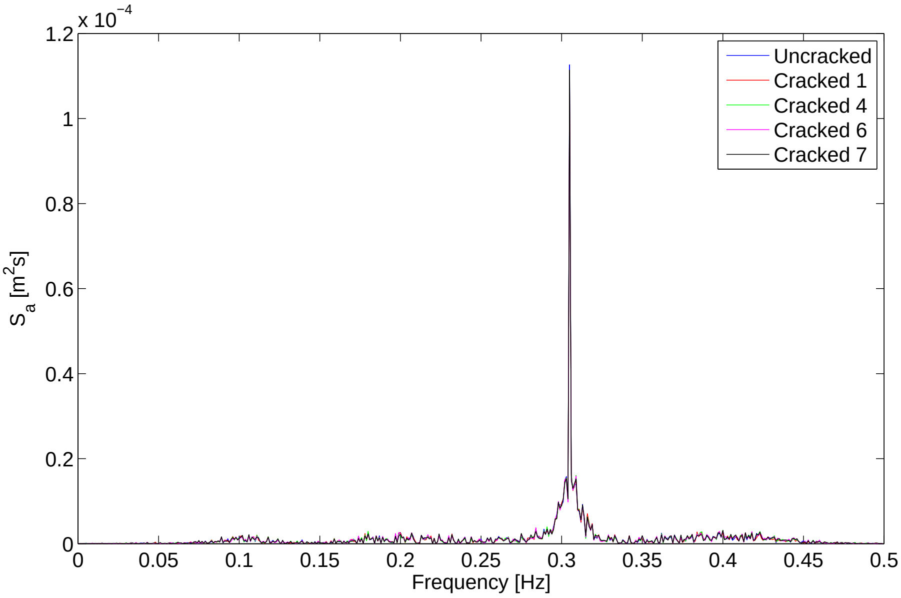

Figure 6.16: Comparison of acceleration response frequency spectra at tower bottom with added harmonic force

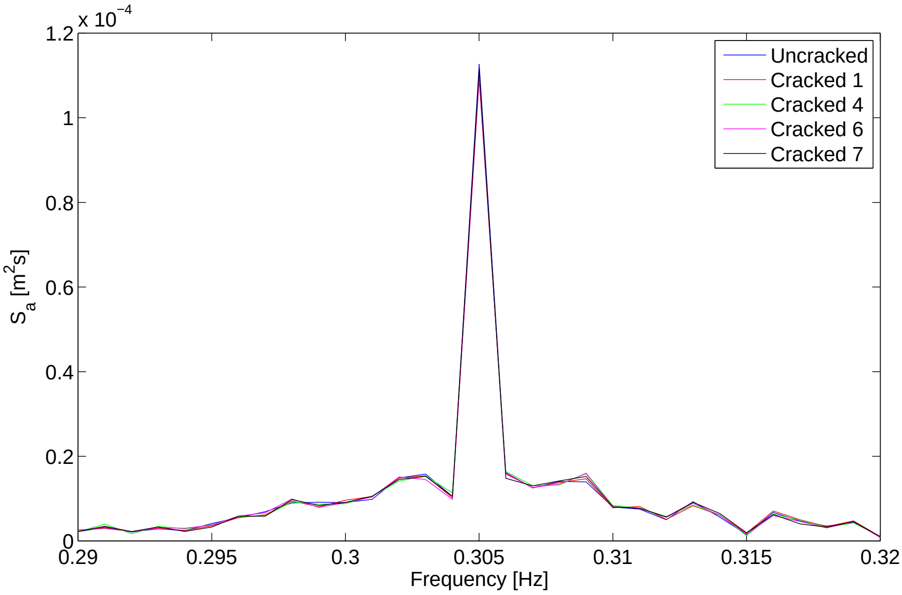

Figure 6.17: Detail around first eigenfrequency of Figure 6.16

6.5 Conclusions

Three kinds of investigations have been conducted in this chapter to analyse whether it is possible to detect and identify the presence of a crack in a complex system as an offshore wind turbine jacket support structure.

The eigenvalue analysis revealed that there are some changes in the eigenfrequencies, but mainly for higher local brace modes, where the crack is located. Even those deviations proved to be relatively small. In the ambient vibration test the results from the eigenvalue analysis were confirmed. In the excitation frequency band there is no visible and reliable difference in the response spectrum between the undamaged and the damaged support structure.

In a final step, forced vibration testing was conducted on the support structure at the first eigenfrequency to see whether it is possible to amplify the differences in the spectrum that were hardly visible in the ambient vibration test. It was found that the peak in the spectrum increases clearly, but the difference between the damaged and the undamaged support structure did not increase.

7 Conclusions

7.1 Recapitulation

In this thesis, the question has been investigated whether a crack in a complex structure as an offshore wind turbine jacket support structure can be detected by online measurements. As measured quantity accelerations have been chosen because they can be measured directly by accelerometers. The measurement sensors were placed on the turbine tower, where these days accelerometers are already often installed. The investigation was based on comparisons between the undamaged and the damaged structure because only a relative approach could possibly yield useful results.

A support structure model has been developed and implemented, that is described in Chapters 2 to 4. The crack was introduced into the model below water as the detection of cracks in the submerged part of the support structure is especially difficult, yet desirable. To account for the model and measurement uncertainties, a Kalman filter has been implemented and the responses have been predicted.

An eigenvalue analysis showed that the presence of the crack has mainly influence on higher local brace or crack modes. Still, the deviation were found to be small, which makes the identification difficult. The lowest eigenfrequencies in the damaged structure, which were of primary interest in this thesis, because damage detection based on ambient vibrations has been considered as a primary goal, showed no deviation from the undamaged structure.

In a second step, the responses of the damaged and the undamaged structure due to ambient vibrations were compared. The results from the eigenvalue analysis were confirmed as there was no visible shift of the first eigenfrequency or a significant difference in the peak of the spectra. Finally, in an attempt to amplify the deviation between undamaged and damaged support structure, a forced harmonic vibration with the frequency of the first eigenfrequency has been added to the support structure at the tower bottom in the horizontal direction. It was, however, not possible to increase the deviation.

To conclude the investigations so far, it can be said that it is not possible or very unlikely to identify small damage in a complex and large structure as an offshore wind turbine jacket support structure based on ambient vibrations. The main reason for this is the small frequency band that is excited by wind and waves.

7.2 Recommendations for Future Work

This thesis constitutes a first step towards the damage identification of large and complex offshore wind turbine support structures based on ambient and forced vibrations. Several issues should be tackled in future work to improve the quality and possibility to identify damage.

- The results of the eigenvalue analysis showed that higher local modes are affected most by the presence of the crack. With a model that is able to describe these local modes, the forced vibration test should be repeated with a harmonic force at one of the eigenfrequencies of the support structure that were affected most.
- There exist extensions of the Kalman filter that are able to estimate the force input as well, so that it must not be known beforehand. Also a non-linear Kalman filter could be implemented to account for the motions of the structure in the force calculations.
- In the literature, many different concepts exist to represent a crack. Other concepts could be tested to see how they influence the results of identification.
- Only one sea state has been considered in this analysis. In a statistical analysis it could be investigated how changing the sea state and wind conditions affects the results of the identification.

Appendix

A Derivation of Finite Element Theory for Bars and Beams

A.1 Bar Element

A.1.1 Definition Bar/Assumptions

A bar is a one-dimensional, continuous and straight structure. It describes axial displacements due to axial forces. The axial displacement $u(x,t)$ is a function of the longitudinal space coordinate x and time t. It is assumed that the bar's vibrations are small and that the material is linear-elastic, thus obeying Hooke's law.

A.1.2 Equation of Motion/Strong Form

A bar with length L, cross-section area A, Young's modulus E and distributed axial line load n is assumed.

The bar's motion is described by the axial displacement $u(x,t)$. To derive the bar's equation of motion the displacement u is linked to the strain by the following kinematic relation

$$\epsilon = \frac{\partial u}{\partial x} \tag{A.1}$$

Hooke's Law is applied as constitutive relation to relate stresses and strains.

$$\sigma = E\epsilon \tag{A.2}$$

The normal force N is given as

$$N = \int \sigma \, \mathrm{d}A = \sigma A \tag{A.3}$$

Combining equations (A.1) to (A.3) yields

$$N = EAu' \tag{A.4}$$

Dynamic equilibrium in axial direction of an infinitesimal part of a straight bar is

$$\rho A \Delta x \frac{\partial^2 u}{\partial t^2} = -N(x) + N(x + \Delta x) + n(x)\Delta x \tag{A.5}$$

Taylor expansion of $N(x + \Delta x)$ yields

$$N(x + \Delta x) = N(x) + \frac{\partial N}{\partial x}\Delta x \tag{A.6}$$

Inserting equation (A.6) into (A.5) and dividing by Δx equation (A.7) is obtained.

$$\rho A \frac{\partial^2 u}{\partial t^2} = \frac{\partial N}{\partial x} + n \tag{A.7}$$

After application of equation (A.4) the bar's equation of motion eventually reads

$$\rho A \frac{\partial^2 u(x,t)}{\partial t^2} - EA \frac{\partial^2 u(x,t)}{\partial x^2} = n(x,t) \tag{A.8}$$

This is also denoted as the strong form of the initial value problem(?).

A.1.3 Variational Formulation/Weak Form

The variational formulation of equation (A.8) is written as:

$$\int \delta u \left(\rho A \frac{\partial^2 u}{\partial t^2} - EA \frac{\partial^2 u}{\partial x^2} - n \right) \mathrm{d}x = 0 \tag{A.9}$$

Rearranging yields:

$$\int \delta u \rho A \frac{\partial^2 u}{\partial t^2} \,\mathrm{d}x - \int \delta u EA \frac{\partial^2 u}{\partial x^2} \,\mathrm{d}x - \int \delta u n \,\mathrm{d}x = 0 \tag{A.10}$$

Integration by parts is defined as

$$\int_a^b u v' \,\mathrm{d}x = [uv]_a^b - \int_a^b u' v \,\mathrm{d}x \tag{A.11}$$

Substitution of $u = \delta u$ and $v = u''$

$$- \int \delta u EA u'' \,\mathrm{d}x = - \left[\delta u \underbrace{EAu'}_{N} \right]_0^L + \int \delta u' EA u' \,\mathrm{d}x \tag{A.12}$$

Finally, the weak form is:

$$\int \delta u \rho A \ddot{u} \,\mathrm{d}x + \int \delta u' EA u' \,\mathrm{d}x - \int \delta u n \,\mathrm{d}x = 0 \tag{A.13}$$

A.1.4 Bar Finite Element

Shape Functions

The bar finite element has two degrees of freedom u

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^{\mathrm{T}} \tag{A.14}$$

The displacement $u(x,t)$ is approximated by (separation of variables)

$$u(x,t) = \mathbf{N}\mathbf{u} = \mathbf{N}(x)\mathbf{u}(t) \tag{A.15}$$

The derivatives are

$$\dot{u}(x,t) = \mathbf{N}\dot{\mathbf{u}} = \mathbf{N}(x)\dot{\mathbf{u}}(t) \tag{A.16}$$
$$\ddot{u}(x,t) = \mathbf{N}\ddot{\mathbf{u}} = \mathbf{N}(x)\ddot{\mathbf{u}}(t) \tag{A.17}$$

Equation (A.15) can be written in matrix form:

$$u(x,t) = \begin{bmatrix} N_{u1} & N_{u2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{A.18}$$

The shape functions $\mathbf{N}$ are given as

$$\begin{aligned} N_{u1}(\xi) &= 1 - \xi \\ N_{u2}(\xi) &= \xi \end{aligned} \tag{A.19}$$

The shape functions are defined in terms of the space coordinate ξ

$$\xi = \frac{x}{L} \tag{A.20}$$

$$u'(x,t) = \frac{\partial u}{\partial x} = \frac{1}{L}\frac{\partial u}{\partial \xi} = \frac{1}{L}\frac{\mathrm{d}\mathbf{N}}{\mathrm{d}\xi}\mathbf{u} = \mathbf{B}\mathbf{u} = \mathbf{N}'\mathbf{u} = \mathbf{N}'(x)\mathbf{u}(t) \tag{A.21}$$

$\mathbf{B}$ is the strain-displacement matrix.

$$\mathbf{B} = \frac{1}{L}\begin{bmatrix} -1 & 1 \end{bmatrix} \tag{A.22}$$

Same interpolation functions for virtual displacements δu as for unknown displacements u

$$\begin{aligned} \delta u &= \mathbf{N}\delta\mathbf{u} \\ \delta\dot{u} &= \mathbf{N}\delta\dot{\mathbf{u}} \\ \delta\ddot{u} &= \mathbf{N}\delta\ddot{\mathbf{u}} \\ \delta u' &= \mathbf{N}'\delta\mathbf{u} = \mathbf{B}\delta\mathbf{u} \end{aligned} \tag{A.23}$$

Inserting those expressions yields:

$$\int \delta\mathbf{u}^{\mathrm{T}}\rho A\mathbf{N}^{\mathrm{T}}\mathbf{N}\ddot{\mathbf{u}}\,\mathrm{d}x + \int \delta\mathbf{u}^{\mathrm{T}}EA\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{u}\,\mathrm{d}x - \int \delta\mathbf{u}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}n\,\mathrm{d}x = 0 \tag{A.24}$$

Rearranging

$$\delta\mathbf{u}^{\mathrm{T}}\int \rho A\mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x\ddot{\mathbf{u}} + \delta\mathbf{u}^{\mathrm{T}}\int EA\mathbf{B}^{\mathrm{T}}\mathbf{B}\,\mathrm{d}x\mathbf{u} - \delta\mathbf{u}^{\mathrm{T}}\int \mathbf{N}^{\mathrm{T}}n\,\mathrm{d}x = 0 \tag{A.25}$$

The equation of motion for the bar element in weak form is then:

$$\underbrace{\int \rho A\mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x}_{\mathbf{M}}\;\ddot{\mathbf{u}} + \underbrace{\int EA\mathbf{B}^{\mathrm{T}}\mathbf{B}\,\mathrm{d}x}_{\mathbf{K}}\;\mathbf{u} = \underbrace{\int \mathbf{N}^{\mathrm{T}}n\,\mathrm{d}x}_{\mathbf{f}} \tag{A.26}$$

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{A.27}$$

Mass Matrix

Every structural element has a distributed mass characterised by its material density. Associated with these distributed mass there are inertia forces occurring when the body is subject to accelerations. In the process of constructing a mass matrix the distributed inertia forces are transformed to nodal forces. There are two standard methods for the construction of mass matrices on the element level: direct mass lumping and variational mass lumping.

If direct mass lumping is applied, a diagonally lumped mass matrix is obtained, where no interaction or coupling effects between the nodal masses occur. Variational mass lumping is based on a variational formulation as performed in section xxx.

According to equation (A.13) the mass matrix is then given by

$$\mathbf{M} = \int \rho A \mathbf{N}^{\mathrm{T}} \mathbf{N} \, \mathrm{d}x \tag{A.28}$$

The mass matrix $\mathbf{M}$ is called a consistent mass matrix because the applied shape functions $\mathbf{N}$ are the same shape functions as applied in the definition of the stiffness matrix $\mathbf{K}$ and the force vector $\mathbf{f}$.

For a bar with uniform density ρ, constant cross-section area A and length L the consistent mass matrix of the bar finite element is defined as

$$\mathbf{M} = \rho A \int_0^L \mathbf{N}^{\mathrm{T}} \mathbf{N} \, \mathrm{d}x = \frac{1}{2} \rho A L \int_0^1 \mathbf{N}^{\mathrm{T}} \mathbf{N} \, \mathrm{d}\xi \tag{A.29}$$

$$\mathbf{M} = \frac{1}{6} \rho A L \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{A.30}$$

Stiffness Matrix

$$\mathbf{K} = \int_0^L EA \mathbf{B}^{\mathrm{T}} \mathbf{B} \, \mathrm{d}x = EAL \int_0^1 \mathbf{B}^{\mathrm{T}} \mathbf{B} \, \mathrm{d}\xi \tag{A.31}$$

Performing the integration

$$\mathbf{K} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{A.32}$$

Nodal Force Vector

$$\mathbf{f} = \int_0^L \mathbf{N}^{\mathrm{T}} n \, \mathrm{d}x = L \int_0^1 \mathbf{N}^{\mathrm{T}} n \, \mathrm{d}\xi \tag{A.33}$$

For a constant load n

$$\mathbf{f} = \frac{1}{2} n L \begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathrm{T}} \tag{A.34}$$

A.2 Beam Element

A.2.1 Definition Beam/Assumptions

The model of a beam in bending is a special case of a one-dimensional continuous structure. The beam describes lateral displacements due to moments and lateral loads. Euler-Bernoulli beam theory is assumed, which implies the assumption of small vibrations. Euler-Bernoulli beam theory furthermore states that the beam's cross-section is infinitely rigid in its own plane, i.e. no deformations occur in the plane of the cross-section, and that the cross-section remains plane and normal to the beam's deformed axis (normality assumption). The material obeys Hooke's law, which means that linear-elastic material is considered.

A.2.2 Equation of Motion/Strong Form

The equation of motion for the transverse motion of a bending beam is derived. To this end, an infinitesimal part of a straight beam under a transverse distributed external load $q(x,t)$ is considered. The beam has a bending stiffness EI, a cross-sectional area A and a mass density ρ. The beam's motion is described by a two dimensional displacement field

$$\begin{bmatrix} u(x,z,t) \\ w(x,z,t) \end{bmatrix} \tag{A.35}$$

u and w are the axial and the transverse displacement respectively. In order to simplify the beam problem to one dimension kinematic relations between u and w are introduced. To this end, the static case is considered. From the assumption that plane cross-sections remain plane and perpendicular to the neutral axis the following kinematic relations can be derived

$$u(x,z) = z\theta = z\phi = -z\frac{\partial w}{\partial x} = -zw' \tag{A.36}$$

$$w(x,z) = w(x) \tag{A.37}$$

The beam's transverse displacement $w(x)$ is a continuous function of coordinate x only. The axial displacement $u(x,z)$ is expressed in terms of the transverse displacement $w(x)$ and the z coordinate. The bending due to the distributed external load q causes axial stresses and strains in the beam. Using relationship (A.36) the strain can be expressed as

$$\epsilon = \frac{\partial u}{\partial x} = -z\frac{\partial^2 w}{\partial x^2} = -zw'' \tag{A.38}$$

The constitutive relation, which relates stresses and strains, is based on Hooke's law.

$$\sigma = E\epsilon \tag{A.39}$$

Inserting (A.38) into (A.39) the stress reads

$$\sigma = -zEw'' \tag{A.40}$$

Consequently, bending moment and curvature are related as

$$M = \int z\sigma \,\mathrm{d}A = -w''E \int z^2 \,\mathrm{d}A = -EIw'' \tag{A.41}$$

Here, the definition for the area moment of inertia I was used. From the vertical equilibrium the equation of motion can be derived using Newton's second law:

$$\rho A \Delta x \frac{\partial^2 w}{\partial t^2} = -V(x) + V(x + \Delta x) + q\Delta x \tag{A.42}$$

The shear forces are assumed to be vertical because of the condition of small vibrations introduced in section xxx. Taylor expansion of the second term on the right-hand side yields

$$V(x + \Delta x) = V(x) + \frac{\partial V}{\partial x}\Delta x \tag{A.43}$$

Equation (A.42) simplifies to

$$\rho A \frac{\partial^2 w}{\partial t^2} = \frac{\partial V}{\partial x} + q \tag{A.44}$$

Neglecting rotational inertia the moment equilibrium reads

$$V = \frac{\partial M}{\partial x} \tag{A.45}$$

Insertion of equation (A.45) into the equation of motion (A.44) yields

$$\rho A \frac{\partial^2 w}{\partial t^2} = \frac{\partial^2 M}{\partial x^2} + q \tag{A.46}$$

Applying equation (A.41) and assuming EI to be constant the equation of motion can be written as

$$\rho A \frac{\partial^2 w(x,t)}{\partial t^2} + EI \frac{\partial^4 w(x,t)}{\partial x^4} = q(x,t) \tag{A.47}$$

The equation of motion is an inhomogeneous partial differential equation of the fourth order and describes the response of the beam to the initial conditions and the applied external loading $q(x,t)$. It does not take into account the type of supporting as it applies to an infinitesimal element of the beam. The manner the beam is supported has a large impact on the dynamic behaviour and is introduced by means of the boundary conditions. Interface conditions are applied if an abrupt change of one of the distributed parameters occurs. The equation of motion is valid as long as the assumptions of small vibrations, the Euler-Bernoulli beam and neglected rotational inertia are not violated.

A.2.3 Variational Formulation/Weak Form

The static equation is a special case of the dynamic equation of motion, where velocities and accelerations are set to zero. The resulting ordinary differential equation for beam bending together with the corresponding boundary conditions represents the strong form of the boundary value problem.

$$\rho A \frac{\partial^2 w(x,t)}{\partial t^2} + EI \frac{\partial^4 w(x,t)}{\partial x^4} - q(x,t) = 0 \tag{A.48}$$

The strong form may be reformulated into an equivalent weak form, where equilibrium is satisfied in an integral (averaged) sense as opposed to the pointwise equilibrium in the strong form.

$$\int \delta w \left(\rho A \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^4} - q \right) \mathrm{d}x = 0 \tag{A.49}$$

δw is the virtual displacement. Rearranging (and assuming EI to be constant) yields

$$\int \delta w \rho A \frac{\partial^2 w}{\partial t^2} \, \mathrm{d}x + \int \delta w EI \frac{\partial^4 w}{\partial x^4} \, \mathrm{d}x - \int \delta w q \, \mathrm{d}x = 0 \tag{A.50}$$

Stiffness term: Integration by parts is defined as

$$\int_a^b u v' \, \mathrm{d}x = [uv]_a^b - \int_a^b u' v \, \mathrm{d}x \tag{A.51}$$

Substitution of $u = \delta w$ and $v = w^{\text{iv}}$ results in equation (A.52).

$$\int \delta w EI w^{\text{iv}} \, \mathrm{d}x = \left[\delta w \underbrace{EI w'''}_{V} \right]_0^L - \int \delta w' EI w''' \, \mathrm{d}x \tag{A.52}$$

The second term on the right-hand side is integrated by parts again with $u = \delta w'$ and $v = w'''$

$$- \int \delta w' EI w''' \, \mathrm{d}x = - \left[\delta w' \underbrace{EI w''}_{M} \right]_0^L + \int \delta w'' EI w'' \, \mathrm{d}x \tag{A.53}$$

The weak form of the boundary value problem finally reads (nodal forces go into q)

$$\int \delta w \rho A \ddot{w} \, \mathrm{d}x + \int \delta w'' EI w'' \, \mathrm{d}x - \int \delta w q \, \mathrm{d}x = 0 \tag{A.54}$$

This is the principle of virtual displacements (work?)/D'Alembert's principle.

Explanation

A.2.4 Beam Finite Element

Shape Functions

To arrive at a finite element formulation, the beam structure or the weak form respectively is discretised into one or more elements. Since the deflection $w(x)$ is the primary unknown, a displacement pattern in each element is assumed. The variational index of equation (**??**) is 2, because it includes derivatives of the second order in the space coordinate x. Thus, admissible displacements must be C^1 continuous, meaning that displacements and their first derivatives must be continuous within the element and especially between elements. Each beam element has two end nodes 1 and 2 and four degrees of freedom. The nodal displacement vector is

$$\mathbf{u} = \begin{bmatrix} w_1 & \theta_1 & w_2 & \theta_2 \end{bmatrix}^{\mathrm{T}} \tag{A.55}$$

Figure beam element

The displacement $w(x)$ is defined by the degrees of freedom $\mathbf{u}$ and shape functions $\mathbf{N}$, which approximate the displacement pattern. The simplest shape functions satisfying the C^1 continuity are the Hermitian cubic shape functions.

$$w(x) = \begin{bmatrix} N_{w1} & N_{\theta 1} & N_{w2} & N_{\theta 2} \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{bmatrix} = \mathbf{Nu} \tag{A.56}$$

The shape functions are expressed in terms of a dimensionless coordinate ξ, which ranges from $\xi = -1$ at node 1 and $\xi = 1$ at node 2.

$$\xi = \frac{2x}{L} - 1 \tag{A.57}$$

The shape functions are (figure shape functions)

$$\begin{aligned} N_{w1}(\xi) &= \frac{1}{4}(1-\xi)^2(2+\xi) \\ N_{\theta 1}(\xi) &= \frac{1}{8}L(1-\xi)^2(1+\xi) \\ N_{w2}(\xi) &= \frac{1}{4}(1+\xi)^2(2-\xi) \\ N_{\theta 2}(\xi) &= -\frac{1}{8}L(1+\xi)^2(1-\xi) \end{aligned} \tag{A.58}$$

It is furthermore assumed that the displacement $w(x)$ consists of a space- and a time-dependent part. The principle of separation of variable is applied. The shape functions $\mathbf{N}$ are only dependent on the x (or ξ coordinate respectively) coordinate and the degrees of freedom $\mathbf{u}$ are only dependent on time t:

$$w(x) = \mathbf{Nu} = \mathbf{N}(x)\mathbf{u}(t) \tag{A.59}$$

The respective time derivatives (nodal velocities and nodal accelerations) are

$$\dot{w}(x) = \mathbf{N}\dot{\mathbf{u}} = \mathbf{N}(x)\dot{\mathbf{u}}(t) \tag{A.60}$$
$$\ddot{w}(x) = \mathbf{N}\ddot{\mathbf{u}} = \mathbf{N}(x)\ddot{\mathbf{u}}(t) \tag{A.61}$$

The derivatives with respect to the space coordinate can be obtained in a similar manner. The curvature w'' can be expressed in terms of the assumed displacement field by

$$w' = \frac{\mathrm{d}w}{\mathrm{d}x} = \frac{2}{L}\frac{\mathrm{d}w}{\mathrm{d}\xi} = \frac{2}{L}\frac{\mathrm{d}\mathbf{N}}{\mathrm{d}\xi}\mathbf{u} = \mathbf{N}'\mathbf{u} = \mathbf{N}'(x)\mathbf{u}(t) \tag{A.62}$$
$$w'' = \frac{\mathrm{d}^2w}{\mathrm{d}x^2} = \frac{4}{L^2}\frac{\mathrm{d}^2w}{\mathrm{d}\xi^2} = \frac{4}{L^2}\frac{\mathrm{d}^2\mathbf{N}}{\mathrm{d}\xi^2}\mathbf{u} = \mathbf{B}\mathbf{u} = \mathbf{N}''\mathbf{u} = \mathbf{N}''(x)\mathbf{u}(t) \tag{A.63}$$

$\mathbf{B}$ is the curvature-displacement matrix and is determined to be

$$\mathbf{B} = \frac{4}{L^2}\frac{\mathrm{d}^2\mathbf{N}}{\mathrm{d}\xi^2} = \frac{1}{L}\begin{bmatrix} 6\frac{\xi}{L} & 3\xi - 1 & -6\frac{\xi}{L} & 3\xi + 1 \end{bmatrix} \tag{A.64}$$

The factor $4/L^2$ is introduced into equation (A.62) by the switch from space coordinate x to the dimensionless coordinate ξ as shown below.

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{\mathrm{d}f(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x} = \frac{2}{L}\frac{\mathrm{d}f(\xi)}{\mathrm{d}\xi} \tag{A.65}$$

$$\frac{\mathrm{d}^2f(x)}{\mathrm{d}x^2} = \frac{\mathrm{d}(2/L)}{\mathrm{d}x}\frac{\mathrm{d}f(\xi)}{\mathrm{d}\xi} + \frac{2}{L}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}f(\xi)}{\mathrm{d}\xi}\right) = \frac{4}{L^2}\frac{\mathrm{d}^2f(\xi)}{\mathrm{d}\xi^2} \tag{A.66}$$

The virtual displacement field δw and their derivatives in time and space are interpolated with the same set of interpolation/shape functions as the unknown displacement field w.

$$\begin{aligned} \delta w &= \mathbf{N}\delta\mathbf{u} \\ \delta\dot{w} &= \mathbf{N}\delta\dot{\mathbf{u}} \\ \delta\ddot{w} &= \mathbf{N}\delta\ddot{\mathbf{u}} \\ \delta w' &= \mathbf{N}'\delta\mathbf{u} \\ \delta w'' &= \mathbf{B}\delta\mathbf{u} \end{aligned} \tag{A.67}$$

Compatibility between virtual displacements and virtual curvatures (strain) yields

Substituting the derived relationships into equation (A.54) results in equation (A.68).

$$\int \delta\mathbf{u}^{\mathrm{T}}\rho A\mathbf{N}^{\mathrm{T}}\mathbf{N}\ddot{\mathbf{u}}\,\mathrm{d}x + \int \delta\mathbf{u}^{\mathrm{T}}EI\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{u}\,\mathrm{d}x - \int \delta\mathbf{u}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}q\,\mathrm{d}x = 0 \tag{A.68}$$

Rearranging

$$\delta\mathbf{u}^{\mathrm{T}}\int \rho A\mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x\,\ddot{\mathbf{u}} + \delta\mathbf{u}^{\mathrm{T}}\int EI\mathbf{B}^{\mathrm{T}}\mathbf{B}\,\mathrm{d}x\,\mathbf{u} - \delta\mathbf{u}^{\mathrm{T}}\int \mathbf{N}^{\mathrm{T}}q\,\mathrm{d}x = 0 \tag{A.69}$$

Taking into account that δw and w are independent (and thus $\delta\mathbf{u}$ and $\mathbf{u}$) (A.69) becomes

$$\underbrace{\int \rho A \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x}_{\mathbf{M}}\ \ddot{\mathbf{u}} + \underbrace{\int EI\mathbf{B}^{\mathrm{T}}\mathbf{B}\,\mathrm{d}x}_{\mathbf{K}}\ \mathbf{u} = \underbrace{\int \mathbf{N}^{\mathrm{T}}q\,\mathrm{d}x}_{\mathbf{f}} \tag{A.70}$$

Equation (A.70) represents the well-known form of the equation of motion in matrix form obtained by a variational formulation.

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{A.71}$$

Mass Matrix

For a beam with uniform density ρ, constant cross-section area A and length L the consistent mass matrix is defined as

$$\mathbf{M} = \rho A \int_0^L \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}x = \frac{1}{2}\rho AL \int_{-1}^{1} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\xi \tag{A.72}$$

Evaluation of the integral yields the consistent mass matrix for a beam finite element:

$$\mathbf{M} = \frac{\rho AL}{420}\begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix} \tag{A.73}$$

Stiffness Matrix

The element stiffness matrix $\mathbf{K}$ is defined in (A.70) as

$$\mathbf{K} = \int_0^L EI\mathbf{B}^T\mathbf{B}\,\mathrm{d}x \tag{A.74}$$

Assuming a prismatic beam with constant stiffness EI equation (A.74) becomes

$$\mathbf{K} = \frac{1}{2}LEI \int_{-1}^{1} \mathbf{B}^T\mathbf{B}\,\mathrm{d}\xi \tag{A.75}$$

Performing the integration eventually yields the stiffness matrix for a beam finite element with four degrees of freedom.

$$\mathbf{K} = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{A.76}$$

Nodal Force Vector

The consistent element node force vector **f** is according to equation (A.70)

$$\mathbf{f} = \int_0^L \mathbf{N}^{\mathrm{T}} q \, \mathrm{d}x \tag{A.77}$$

The coordinate change yields

$$\mathbf{f} = \frac{1}{2} L \int_{-1}^{1} \mathbf{N}^{\mathrm{T}} q \; \mathrm{d}\xi \tag{A.78}$$

For a constant load q the result finally reads

$$\mathbf{f} = \frac{1}{2} q L \begin{bmatrix} 1 & \frac{1}{6} & 1 & -\frac{1}{6} \end{bmatrix}^{\mathrm{T}} \tag{A.79}$$

B Support Structure Model

Node	x [m]	z [m]	Node	x [m]	z [m]
1	-6.000	0.000	17	4.016	61.151
2	-6.000	0.500	18	4.000	61.650
3	-5.967	1.499	19	0.000	12.127
4	-5.939	2.373	20	0.000	29.129
5	-5.333	20.886	21	0.000	43.542
6	6.000	0.000	22	0.000	55.762
7	6.000	0.500	23	0.000	61.650
8	5.967	1.499	24	0.000	62.650
9	5.939	2.373	25	0.000	73.650
10	5.333	20.886	26	0.000	83.650
11	-4.820	36.578	27	0.000	95.650
12	-4.385	49.878	28	0.000	105.650
13	-4.016	61.151	29	0.000	115.650
14	-4.000	61.650	30	0.000	124.650
15	4.820	36.578	31	0.000	129.650
16	4.385	49.878	32	4.277	35.738

Table B.1: Nodal coordinates of the jacket support structure

Node	DOFs	Node	DOFs
1	1, 2, 3	17	49, 50, 51
2	4, 5, 6	18	52, 53, 54
3	7, 8, 9	19	55, 56, 57
4	10, 11, 12	20	58, 59, 60
5	13, 14, 15	21	61, 62, 63
6	16, 17, 18	22	64, 65, 66
7	19, 20, 21	23	67, 68, 69
8	22, 23, 24	24	70, 71, 72
9	25, 26, 27	25	73, 74, 75
10	28, 29, 30	26	76, 77, 78
11	31, 32, 33	27	79, 80, 81
12	34, 35, 36	28	82, 83, 84
13	37, 38, 39	29	85, 86, 87
14	40, 41, 42	30	88, 89, 90
15	43, 44, 45	31	91, 92, 93
16	46, 47, 48	32	94, 95, 96

Table B.2: Global degrees of freedom of the jacket support structure

Member	Node 1	Node 2	Prop.	Member	Node 1	Node 2	Prop.
1	1	2	2	23	10	20	1
2	2	3	2	24	20	11	1
3	3	4	2	25	20	32	1
4	4	5	2	26	32	15	1
5	6	7	2	27	11	21	1
6	7	8	2	28	15	21	1
7	8	9	2	29	21	12	1
8	9	10	2	30	21	16	1
9	5	11	3	31	12	22	1
10	11	12	3	32	16	22	1
11	12	13	3	33	22	13	1
12	13	14	3	34	22	17	1
13	10	15	3	35	14	23	14
14	15	16	3	36	23	18	14
15	16	17	3	37	23	24	6
16	17	18	3	38	24	25	7
17	3	8	1	39	25	26	8
18	4	19	1	40	26	27	9
19	9	19	1	41	27	28	10
20	19	5	1	42	28	29	11
21	19	10	1	43	29	30	12
22	5	20	1	44	30	31	13

Table B.3: Element node list of the jacket support structure

	Jacket support structure			
Property set	D [m]	t [m]	E [N/m^2]	ρ [kg/m^3]
1	0.800	0.020	$2.1 \cdot 10^{11}$	7850
2	1.200	0.050	$2.1 \cdot 10^{11}$	7850
3	1.200	0.035	$2.1 \cdot 10^{11}$	7850
4	1.200	0.040	$2.1 \cdot 10^{11}$	7850
	Foundation piles			
5	2.082	0.060	$2.1 \cdot 10^{11}$	7850
	Tower			
6	5.589	0.032	$2.1 \cdot 10^{11}$	7850
7	5.448	0.031	$2.1 \cdot 10^{11}$	7850
8	5.200	0.029	$2.1 \cdot 10^{11}$	7850
9	4.941	0.026	$2.1 \cdot 10^{11}$	7850
10	4.683	0.023	$2.1 \cdot 10^{11}$	7850
11	4.447	0.021	$2.1 \cdot 10^{11}$	7850
12	4.224	0.025	$2.1 \cdot 10^{11}$	7850
13	4.059	0.030	$2.1 \cdot 10^{11}$	7850
	Transition piece			
	A [m^2]	I [m^4]	E [N/m^2]	ρ [kg/m^3]
14	41.625	51.200	$3.0 \cdot 10^{10}$	2000

Table B.4: Member properties

C Mathematica Code

C.1 Determination of Stiffness Matrix for Finite Beam Element with Crack

```
In[1]:= A = {{1, 0, 0, 0, 0, 0, 0, 0}, {0, 1, 0, 0, 0, 0, 0, 0},
            {0, 0, 0, 0, 1, L, L^2, L^3}, {0, 0, 0, 0, 0, 1, 2 * L, 3 * L^2},
            {1, L / 2, L^2 / 4, L^3 / 8, -1, -L / 2, -L^2 / 4, -L^3 / 8},
            {0, 1, L, 3 / 4 * L^2, 0, -1, -L + 2 * K, -3 / 4 * L^2 + 3 * K * L},
            {0, 0, 2, 3 * L, 0, 0, -2, -3 * L}, {0, 0, 0, 6, 0, 0, 0, -6}};

In[2]:= q = {q1, q2, q3, q4, 0, 0, 0, 0};

In[3]:= Simplify[LinearSolve[A, q]];

In[4]:= {a, Cm} = Simplify[Normal[CoefficientArrays[{q1 == a1, q2 == a2,
            -(3 K (2 q1 - 2 q3 + L (q2 + q4)) + 2 L (3 q1 - 3 q3 + L (2 q2 + q4)))/(2 L^2 (K + L)) == a3,
            (2 q1 - 2 q3 + L (q2 + q4))/L^3 == a4, (2 L q1 + K (2 q1 + L (q2 - q4)))/(2 (K + L)) == a5, (L q2 + K q4)/(K + L) == a6,
            -(3 K (2 q1 - 2 q3 + L (q2 + q4)) + 2 L (3 q1 - 3 q3 + L (2 q2 + q4)))/(2 L^2 (K + L)) == a7,
            (2 q1 - 2 q3 + L (q2 + q4))/L^3 == a8}, {q1, q2, q3, q4}]]];

In[6]:= pol = {1, x, x^2, x^3};

In[7]:= N1 = pol.Cm[[1 ;; 4, 1 ;; 4]];

In[8]:= N2 = pol.Cm[[5 ;; 8, 1 ;; 4]];

In[9]:= B1 = D[N1, {x, 2}];

In[10]:= B2 = D[N2, {x, 2}];

In[11]:= K1 = Simplify[Integrate[Outer[Times, B1, B1], {x, 0, L / 2}]];

In[12]:= K2 = Simplify[Integrate[Outer[Times, B2, B2], {x, L / 2, L}]];

In[13]:= Km = Simplify[K1 + K2];
```

C.2 Determination of Crack Coefficient

```
In[1]:= De = 0.8;

In[2]:= t = 0.02;

In[3]:= Di = De - 2 * t;

In[4]:= β = Sqrt[x - x^2 - (1 - γ^2) / 4];

In[5]:= γ = Di / De;

In[6]:= Imom = Pi / 64 * (De^4 - (De - 2 * t) ^4);

In[7]:= Em = 2.1 * 10^11;

In[8]:= a = Table[0.001 * i, {i, 1, 400}];

In[9]:= na = Length[a];

In[10]:= int = Table[0, {na}];

In[11]:= int1 = Table[0, {na}];

In[12]:= int2 = Table[0, {na}];

In[13]:= int3 = Table[0, {na}];

In[14]:= c = Table[0, {na}];

In[15]:= Kr = Table[0, {na}];

In[16]:= xp = (2 * x + Sqrt[1 - 4 * y^2] - 1) / (2 * Sqrt[1 - 4 * y^2]);

In[17]:= F = Sqrt[2 / (Pi * xp) * Tan[Pi * xp / 2]] *
           (0.923 + 0.199 * (1 - Sin[Pi * xp / 2]) ^4) / (Cos[Pi * xp / 2]);

In[18]:= For[i = 1, i < 401, i++, i;
          If[a[[i]] ≤ t,
           int[[i]] = NIntegrate[(1 - 4 * y^2) * (2 * x + Sqrt[1 - 4 * y^2] - 1) * F^2,
             {x, 0, a[[i]] / De}, {y, -Sqrt[x - x^2], Sqrt[x - x^2]}],
           int1[[i]] = NIntegrate[(1 - 4 * y^2) * (2 * x + Sqrt[1 - 4 * y^2] - 1) * F^2,
             {x, 0, t / De}, {y, -Sqrt[x - x^2], Sqrt[x - x^2]}];
           int2[[i]] = NIntegrate[(1 - 4 * y^2) * (2 * x + Sqrt[1 - 4 * y^2] - 1) * F^2,
             {x, t / De, a[[i]] / De}, {y, -Sqrt[x - x^2], -β}];
           int3[[i]] = NIntegrate[(1 - 4 * y^2) * (2 * x + Sqrt[1 - 4 * y^2] - 1) * F^2,
             {x, t / De, a[[i]] / De}, {y, β, Sqrt[x - x^2]}];
           int[[i]] = int1[[i]] + int2[[i]] + int3[[i]]];
          c[[i]] = 1024 / (Em * De^3 * Pi * (1 - γ^4) ^2) * int[[i]];
          Kr[[i]] = Em * Imom * c[[i]];
         ];

In[19]:= list1 = Table[{a[[i]], Re[Kr[[i]]]}, {i, 1, 400}];
```

D Frequency Domain Representation

D.1 Introduction

The structure's response can be presented in time domain and in frequency domain. The frequency domain has the advantage of providing a clear overview over the frequency content of the response. It thus allows for comparisons of responses, which is not easily possible in time domain because mathematically it is a superposition of randomly shifted sinusoidally waves.

D.2 Frequency Spectra

A discrete time series of data points has a certain sample frequency f_s, number of samples N and duration T. Sample frequency and time step Δt are related by

$$f_\mathrm{s} = \frac{1}{\Delta t} \tag{D.1}$$

The time series can hence be represented by

$$t_n = n\Delta t \qquad n = 1, ..., N \tag{D.2}$$

The time step, number of samples and duration are related by

$$\Delta t = \frac{T}{N} \tag{D.3}$$

In the frequency domain, the frequencies are defined as

$$f_q = q\Delta f \tag{D.4}$$

The lowest frequency in the considered frequency range f_min is simultaneously the frequency resolution Δf:

$$f_\mathrm{min} = f_1 = \Delta f = \frac{1}{T} \tag{D.5}$$

Equation (D.5) determines the relation between the smallest frequency f_min and the simulation duration T. The longer the simulation duration is chosen, the smaller the lowest frequency in the spectrum and the bigger the frequency resolution. The upper end of the frequency range $f_{\max}$ is usually chosen. Wind and wave excitation frequencies do not exceed 0.5 Hz, which is therefore chosen as upper limit here. $f_{\max}$ is defined in equation (D.6) [37].

$$f_{\max} = f_{N/2} = \frac{N}{2T} \tag{D.6}$$

With equations (D.1) and (D.3) a suitable sample frequency (and thus time step) can be determined:

$$f_{\mathrm{s}} = 2f_{\max} \tag{D.7}$$

The sample frequency f_{s} is thus (at least) 1 Hz. The number of samples N then automatically follows from equation (D.3).

E Matlab Code

E.1 Jacket Model with 96 DOF

```
% 2D jacket structure according to Vorpahl's UpWind Reference Jacket
% 96 DOF

clc
clear all
close all
tic

%% Input values

global eom f zeta omega k phase_JS u_c Cm Cd d f_wi df_wi phase_list ...
    V_hub sigma1 nodcoor elemnod elemD elemSubm elemWind Kc

% Crack 'number'
cn = 8;                 % 1 to 8
cn_str = num2str(cn-1);

% Time span of solution
fs = 1;
dt = 1/fs;              % Sample size/time step
T = 1200;
t = 0:dt:T-dt;          % [s]
N = length(t);

% Node coordinates
nodcoor = [-6.000   0.000;...   % 1
           -6.000   0.500;...   % 2
           -5.967   1.499;...   % 3
           -5.939   2.373;...   % 4
           -5.333  20.886;...   % 5
            6.000   0.000;...   % 6
            6.000   0.500;...   % 7
            5.967   1.499;...   % 8
            5.939   2.373;...   % 9
            5.333  20.886;...   % 10
           -4.820  36.578;...   % 11
           -4.385  49.878;...   % 12
           -4.016  61.151;...   % 13
           -4.000  61.650;...   % 14
            4.820  36.578;...   % 15
            4.385  49.878;...   % 16
            4.016  61.151;...   % 17
            4.000  61.650;...   % 18
            0.000  12.127;...   % 19
            0.000  29.129;...   % 20
            0.000  43.542;...   % 21
            0.000  55.762;...   % 22
            0.000  61.650;...   % 23
            0.000  62.650;...   % 24
            0.000  73.650;...   % 25
            0.000  83.650;...   % 26
            0.000  95.650;...   % 27
            0.000 105.650;...   % 28
            0.000 115.650;...   % 29
            0.000 124.650;...   % 30
            0.000 129.650;...   % 31
            4.218  35.647];     % 32

nDOF = size(nodcoor,1)*3;

% Element nodes
elemnod = [ 1, 2;  2, 3;  3, 4;  4, 5;...
            6, 7;  7, 8;  8, 9;  9,10;...
            5,11; 11,12; 12,13; 13,14;...
           10,15; 15,16; 16,17; 17,18;...
            3, 8;  4,19;  9,19; 19, 5;...
           19,10;  5,20; 10,20; 20,11;...
           20,32; 32,15; 11,21; 15,21;...
           21,12; 21,16; 12,22; 16,22;...
           22,13; 22,17; 14,23; 23,18;...
           23,24; 24,25; 25,26; 26,27;...
           27,28; 28,29; 29,30; 30,31];

% Material properties
Es = 2.1e11;        % [N/m2]
Ec = 3e10;          % [N/m2]
rho_s = 7850;       % [kg/m3]
rho_c = 2000;       % [kg/m3]

elemE = [Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es,...
         Es, Es, Ec, Ec,...
         Es, Es, Es, Es,...
         Es, Es, Es, Es];

elemrho = [rho_s, rho_s, rho_s, rho_s,...
```

```
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_c, rho_c,...
           rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s];

% Member properties

% Jacket
D1 = 0.8;
D2 = 1.2;
D3 = 1.2;
D4 = 1.2;
t1 = 0.02;
t2 = 0.05;
t3 = 0.035;
t4 = 0.04;
A1 = pi/4*(D1^2-(D1-2*t1)^2);
A2 = pi/4*(D2^2-(D2-2*t2)^2);
A3 = pi/4*(D3^2-(D3-2*t3)^2);
A4 = pi/4*(D4^2-(D4-2*t4)^2);
I1 = pi/64*(D1^4-(D1-2*t1)^4);
I2 = pi/64*(D2^4-(D2-2*t2)^4);
I3 = pi/64*(D3^4-(D3-2*t3)^4);
I4 = pi/64*(D4^4-(D4-2*t4)^4);

% Tower
% Dt = [5.600/1.95; 5.577/1.95; 5.318/1.95; 5.082/1.95; 4.800/1.95;...
%       4.565/1.95; 4.329/1.95; 4.118/1.95; 4.000/1.95];
% tt = [0.032/2; 0.032/2.3; 0.030/2; 0.028/2; 0.024/2; 0.022/2; 0.020/2;...
%       0.030/2; 0.030/2];
Dt = [5.600/1; 5.577/1; 5.318/1; 5.082/1; 4.800/1; 4.565/1; 4.329/1;...
      4.118/1; 4.000/1];
tt = [0.032/1; 0.032/1; 0.030/1; 0.028/1; 0.024/1; 0.022/1; 0.020/1;...
      0.030/1; 0.030/1];
Dte = zeros(length(Dt)-1,1);
tte = zeros(length(Dt)-1,1);
At = zeros(length(Dt)-1,1);
It = zeros(length(Dt)-1,1);
for i = 2:length(Dt)
    Dte(i-1) = (Dt(i-1)+Dt(i))/2;
    tte(i-1) = (tt(i-1)+tt(i))/2;
    At(i-1) = pi/4*(Dte(i-1)^2-(Dte(i-1)-2*tte(i-1))^2);
    It(i-1) = pi/64*(Dte(i-1)^4-(Dte(i-1)-2*tte(i-1))^4);
end

% Transition piece
```

```
b_TP = 9.6;                 % [m]
h_TP = 4;                   % [m]
l_TP = 8;                   % [m]
I_TP = b_TP*h_TP^3/12;      % [m4]
m_TP = 666000;              % [kg]
A_TP = m_TP/(rho_c*l_TP);   % [m2]

elemA = [A2, A2, A2, A2,...
         A2, A2, A2, A2,...
         A3, A3, A3, A3,...
         A3, A3, A3, A3,...
         A1, A1, A1, A1,...
         A1, A1, A1, A1,...
         A1, A1, A1, A1,...
         A1, A1, A1, A1,...
         A1, A1, A_TP/2, A_TP/2,...
         At(1), At(2), At(3), At(4),...
         At(5), At(6), At(7), At(8)];
%          A1, A1, A_TP/2.5, A_TP/2.5,...

elemI = [I2, I2, I2, I2,...
         I2, I2, I2, I2,...
         I3, I3, I3, I3,...
         I3, I3, I3, I3,...
         I1, I1, I1, I1,...
         I1, I1, I1, I1,...
         I1, I1, I1, I1,...
         I1, I1, I1, I1,...
         I1, I1, I_TP/2, I_TP/2,...
         It(1), It(2), It(3), It(4),...
         It(5), It(6), It(7), It(8)];
%          I1, I1, I_TP/30, I_TP/30,...

elemD = [D2, D2, D2, D2,...
         D2, D2, D2, D2,...
         D3, D3, D3, D3,...
         D3, D3, D3, D3,...
         D1, D1, D1, D1,...
         D1, D1, D1, D1,...
         D1, D1, D1, D1,...
         D1, D1, D1, D1,...
         D1, D1, 1, 1,...
         Dt(1), Dt(2), Dt(3), Dt(4),...
         Dt(5), Dt(6), Dt(7), Dt(8)];

elemSubm = ones(size(elemnod,1),1);    % Element submerged? 0=false, 1=true
subm = [11,12,15,16,31:44];
elemSubm(subm) = 0;                    % Insert elements that are NOT submerged

elemWind = zeros(size(elemnod,1),1);   % Subject to wind load? 0=false, 1=true
elemWind(37:44) = 1;                   % Insert elements that are subject to wind
```

```
elemCrack = zeros(size(elemnod,1),1);  % Element cracked? 0=false, 1=true
elemCrack(26) = 1;

Kc = [0, 0.00157821, 0.00871166, 0.0215477, 0.0320845, 0.0438089, 0.0502119,...
      0.0560688]; % Crack depth (0.01 m 0.02 m 0.05 m 0.10 m 0.20 m 0.30 m 0.40 m)
Kc = Kc(cn);



%% System Matrices

% System mass and stiffness matrices M and K
[M,K] = SystemMatrices(nodcoor,elemnod,elemSubm,elemCrack,elemE,elemrho,...
                       elemA,elemI,elemD);

% Check for singularity of stiffness matrix separately in Mathematica

% Boundary conditions

% Top mass
% m_RNA = 350000/10;                    % Top mass in [kg] (NREL 5 MW 350t RNA)
m_RNA = 350000/1.5;
m_R = 110000;                           % Rotor mass: 110,000 kg
m_NA = 240000;                          % Nacelle mass: 240,000 kg
r_R = 63;                               % Rotor diameter: 63 m
l_NA = 14;                              % Length of nacelle: 14 m
h_NA = 3.5;                             % Height of nacelle: 3.5 m
I_R = m_R*r_R^2/4;
I_NA = 1/12*m_NA*(l_NA^2+h_NA^2);
I_RNA = I_R + I_NA;
% m_top = 8000;
M(91,91) = M(91,91) + m_RNA;
M(92,92) = M(92,92) + m_RNA;
M(93,93) = M(93,93) + I_RNA/3;

% Additional tower point masses
m_t1 = 1900;
% m_t1 = 500;
M(67,67) = M(67,67) + m_t1;
M(68,68) = M(68,68) + m_t1;

m_t2 = 1400;
% m_t2 = 300;
M(79,79) = M(79,79) + m_t2;
M(80,80) = M(80,80) + m_t2;

m_t3 = 1000;
% m_t3 = 200;
M(91,91) = M(91,91) + m_t3;
M(92,92) = M(92,92) + m_t3;

% Soil springs
```

```
k_x = 6.7821e+08;                       % Soil stiffness in x direction
k_xr = 1.7969e+09;                      % Soil stifness coupling between x and r
k_z = 2.1940e+09;                       % Soil stiffness in z direction
k_r = 1.3061e+10;                       % Soil stiffness in rotational direction

K(1,1) = K(1,1) + k_x;                  % Leg 1
K(2,2) = K(2,2) + k_z;
K(3,3) = K(3,3) + k_r;
K(1,3) = K(1,3) + k_xr;
K(3,1) = K(3,1) + k_xr;

K(16,16) = K(16,16) + k_x;              % Leg 2
K(17,17) = K(17,17) + k_z;
K(18,18) = K(18,18) + k_r;
K(16,18) = K(16,18) + k_xr;
K(18,16) = K(18,16) + k_xr;

% Rayleigh damping matrix
% C = zeros(size(M,1),size(M,1));
C = 0.0001*M + 0.0001*K;

% Soil dashpots
c_x = 8.8800e+07;                       % Damping value in x direction
c_z = 2.8727e+08;                       % Damping value in z direction
c_r = 1.7101e+09;                       % Damping value in rotational direction
c_xr = 2.3527e+08;

C(1,1) = C(1,1) + c_x;                  % Leg 1
C(2,2) = C(2,2) + c_z;
C(3,3) = C(3,3) + c_r;
C(1,3) = C(1,3) + c_xr;
C(3,1) = C(3,1) + c_xr;

C(16,16) = C(16,16) + c_x;              % Leg 2
C(17,17) = C(17,17) + c_z;
C(18,18) = C(18,18) + c_r;
C(16,18) = C(16,18) + c_xr;
C(18,16) = C(18,16) + c_xr;

% Mass matrix
eom.M = M;
Mi = M^(-1);
eom.Mi = Mi;

% Stiffness matrix
eom.K = K;

% Damping matrix
eom.C = C;

% Matrix multiplications
eom.MK = -eom.Mi * eom.K;
eom.MC = -eom.Mi * eom.C;


%% Hydrodynamic load

% Irregular wave properties
Hs = 6;                 % Significant wave height [m]
Tp = 10;                % Spectral peak period [s]
u_c = 1;                % Current velocity [m/s]
Cm = 1.6;               % Inertia coefficient
Cd = 0.8;               % Drag coefficient
d = 49.878;             % Water depth [m]
fp = 1/Tp;              % Spectral peak frequency [Hz]

% Frequency band
df = 0.001;
f = df:df:0.5;
% phase_JS = 2*pi*rand(length(f),1);      % Random phases
load('phase_JS')

S_JS = zeros(length(f),1);
zeta = zeros(length(f),1);
omega = zeros(length(f),1);
k = zeros(length(f),1);
eta = 0;


% JONSWAP spectrum
if Tp/sqrt(Hs) <= 3.6
    gamma = 5;
elseif Tp/sqrt(Hs) > 3.6 && Tp/sqrt(Hs) <= 5
    gamma = exp(5.75-1.15*Tp/sqrt(Hs));
else
    gamma = 1;
end

for i = 1:length(f)

    if f(i) <= fp
        sigma = 0.07;
    else
        sigma = 0.09;
    end

    S_JS(i) = 0.3125*Hs^2*Tp*(f(i)/fp)^(-5)*(1-0.287*log(gamma))...
              *exp(-1.25*(f(i)/fp)^(-4))*gamma^exp(-0.5*((f(i)-fp)/(sigma*fp))^2);

    zeta(i) = sqrt(2*S_JS(i)*df);   % Wave amplitude [m]
    omega(i) = 2*pi*f(i);           % Wave frequency [rad/s]
    k(i) = kSolve(omega(i),d);      % Wave number [rad/m]
    % Wave elevation
    eta = eta + zeta(i)*sin(omega(i)*t+phase_JS(i));
end



%% Wind load

% Wind properties
V_hub = 12;                         % Mean wind speed at hub height
Iref = 0.12;                        % Turbulence intensity
sigma1 = Iref*(0.75*V_hub+5.6);     % Standard deviation (invariant with height)

% Frequency band
df_wi = 0.001;
f_wi = df_wi:df_wi:0.5;
% phase_list = 2*pi*rand(10,length(f_wi));
load('phase_list')

S_KM = zeros(length(f_wi),1);
amp_wi = zeros(length(f_wi),1);
omega_wi = zeros(length(f_wi),1);
V_var = 0;

z_hub = 129.65;
if z_hub-d < 60
    Lk = 5.67*(z_hub-d);            % Integral scale parameter Lk
else
    Lk = 8.1*42;
end
I = sigma1/V_hub;

for j = 1:length(f_wi)
    S_KM(j) = 4*I^2*V_hub*Lk/((1+6*f_wi(j)*Lk/V_hub)^(5/3)); % Kaimal spectrum
    amp_wi(j) = sqrt(2*S_KM(j)*df_wi);                       % Wind amplitude [m/s]
    omega_wi(j) = 2*pi*f_wi(j);                              % Wind frequency [rad/s
    V_var = V_var + amp_wi(j)*sin(omega_wi(j)*t+phase_list(1,j));
end

V_mean = V_hub;           % Mean wind speed at hub height
V = V_mean + V_var;



%% Force vector

f_hydro = zeros(nDOF,N);
f_wind = zeros(nDOF,N);
f_total = zeros(nDOF,N);
for w = 1:N
    v = t(w);
    f_hydro(:,w) = Hydro_Morison_irr_lin(v);
    f_wind(:,w) = Wind_Morison_lin(v);
    f_total(:,w) = f_hydro(:,w) + f_wind(:,w);
end

% Add harmonic force with f=0.305 Hz
A_h = 50;           % [N]
f_h = 0.305;        % [Hz]
omega_h = 2*pi*f_h; % [rad/s]
f_harm = A_h*sin(omega_h*t);
f_total(70,:) = f_total(70,:) + f_harm;

% Wave velocities and accelerations
u_x1 = 0;
u_z1 = 0;
u_x2 = 0;
u_z2 = 0;
a_x1 = 0;
a_z1 = 0;
a_x2 = 0;
a_z2 = 0;

x1 = -4.385;
z1 = 49.878;
for l = 1:length(f)
    u_x1 = u_x1 + omega(l)*zeta(l)*cosh(k(l)*z1)/sinh(k(l)*d)...
                  *sin(omega(l)*t-k(l)*x1+phase_JS(l));
    u_z1 = u_z1 + omega(l)*zeta(l)*sinh(k(l)*z1)/sinh(k(l)*d)...
                  *cos(omega(l)*t-k(l)*x1+phase_JS(l));
    a_x1 = a_x1 + omega(l)^2*zeta(l)*cosh(k(l)*z1)/sinh(k(l)*d)...
                  *cos(omega(l)*t-k(l)*x1+phase_JS(l));
    a_z1 = a_z1 - omega(l)^2*zeta(l)*sinh(k(l)*z1)/sinh(k(l)*d)...
                  *sin(omega(l)*t-k(l)*x1+phase_JS(l));
end



%% Solution

% 1) Initialise solution
n_p = nDOF;                  % Number of forces
n_d = nDOF;                  % Number of observations
x0 = zeros(2*nDOF,1);        % Initial conditions

% Selection matrices
S_v = zeros(n_d,nDOF);       % Velocity
S_d = zeros(n_d,nDOF);       % Displacements
S_p = eye(nDOF);             % Forces
S_a = eye(n_d,nDOF);         % Accelerations (all degrees of freedom)


% 2) Construct the (discrete-time) state-space matrices A, B, C and D
% System matrix A
```

```
A_c = [zeros(nDOF,nDOF) eye(nDOF); -Mi*K -Mi*C];
A = expm(A_c*dt);

% Input matrix B
B_c = [zeros(nDOF,n_p); Mi*S_p];
B = (A-eye(2*nDOF))*A_c^(-1)*B_c;

% Output matrix G
G = [S_d-S_a*Mi*K S_v-S_a*Mi*C];

% Direct transmission matrix J
J = S_a*Mi*S_p;

% Force matrix
p = f_total;

% State matrices
state.A = A;
state.B = B;
state.G = G;
state.J = J;


% 3) Solve
x_k = zeros(2*nDOF,N);      % Preallocation
for time = 1:N
    %   Time update
    if time==1
        x_k(:,time) = A*x0;
    else
        x_k(:,time) = A*x_k(:,time-1)+B*p(:,time-1);
    end
end


% 4) Response determination
n_d = nDOF;                             % Number of output positions
% Displacements
S_d1 = eye(n_d,nDOF);
S_v1 = zeros(n_d,nDOF);
S_a1 = zeros(n_d,nDOF);
G1 = [S_d1-S_a1*Mi*K S_v1-S_a1*Mi*C];
J1 = S_a1*Mi*S_p;
u_k = G1*x_k+J1*p;                    % Displacements [n_d x N], here [nDOF x N]

% Velocities
S_d2 = zeros(n_d,nDOF);
S_v2 = eye(n_d,nDOF);
S_a2 = zeros(n_d,nDOF);
G2 = [S_d2-S_a2*Mi*K S_v2-S_a2*Mi*C];
J2 = S_a2*Mi*S_p;
v_k = G2*x_k+J2*p;                    % Velocities [n_d x N], here [nDOF x N]
```

```
% Accelerations
S_d3 = zeros(n_d,nDOF);
S_v3 = zeros(n_d,nDOF);
S_a3 = eye(n_d,nDOF);
G3 = [S_d3-S_a3*Mi*K S_v3-S_a3*Mi*C];
J3 = S_a3*Mi*S_p;
a_k = G3*x_k+J3*p;                      % Accelerations [n_d x N], here [nDOF x N]


% 4) Save
% cd('C:\Users\s.brauer\Documents\01 Thesis\04 Model Setup\2 MATLAB\Solver\UpWind Jacket\
% dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Cracked ' cn_str '\1Hz
dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Cracked ' cn_str '\harmo
% dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Cracked ' cn_str' '\10
% dir = 'W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Uncracked\stat1';
cd(dir);

save('eom.mat','eom')
save('state.mat','state');
save('u_k.mat','u_k');
save('v_k.mat','v_k');
save('a_k.mat','a_k');
save('t.mat','t');
save('p.mat','p');
save('eta.mat','eta');
save('V.mat','V');

toc
```

E.2 Jacket Model with 312 DOF

```
% 2D jacket structure according to Vorpahl's UpWind Reference Jacket
% 312 DOF

clc
clear all
close all
tic

global eom f zeta omega k phase_JS u_c Cm Cd d f_wi df_wi phase_list V_hub...
       sigma1 nodcoor elemnod elemD elemSubm elemWind Kc

%% Input values

% Crack 'number'
cn = 1;                  % 1 to 8
cn_str = num2str(cn-1);

% Time span of solution
```

```
fs = 300;
dt = 1/fs;              % Sample size/time step
T = 1200;
t = 0:dt:T-dt;          % [s]
N = length(t);

% Node coordinates
nodcoor = [-6.000   0.000;...   % 1      -
           -6.000   0.500;...   % 2      -
           -5.967   1.499;...   % 3      -
           -5.939   2.373;...   % 4      -
           -5.788   7.001;...   % 5
           -5.636  11.630;...   % 6
           -5.485  16.258;...   % 7
           -5.333  20.886;...   % 8      -

            6.000   0.000;...   % 9      -
            6.000   0.500;...   % 10     -
            5.967   1.499;...   % 11     -
            5.939   2.373;...   % 12     -
            5.788   7.001;...   % 13
            5.636  11.630;...   % 14
            5.485  16.258;...   % 15
            5.333  20.886;...   % 16     -

           -5.205  24.809;...   % 17
           -5.077  28.732;...   % 18
           -4.948  32.655;...   % 19
           -4.820  36.578;...   % 20     -
           -4.711  39.903;...   % 21
           -4.603  43.228;...   % 22
           -4.494  46.553;...   % 23
           -4.385  49.878;...   % 24     -
           -4.293  52.696;...   % 25
           -4.201  55.515;...   % 26
           -4.108  58.333;...   % 27
           -4.016  61.151;...   % 28     -
           -4.000  61.650;...   % 29     -

            5.205  24.809;...   % 30
            5.077  28.732;...   % 31
            4.948  32.655;...   % 32
            4.820  36.578;...   % 33     -
            4.711  39.903;...   % 34
            4.603  43.228;...   % 35
            4.494  46.553;...   % 36
            4.385  49.878;...   % 37     -
            4.293  52.696;...   % 38
            4.201  55.515;...   % 39
            4.108  58.333;...   % 40
            4.016  61.151;...   % 41     -
            4.000  61.650;...   % 42     -
```

```
-4.454   4.812;...   % 43
-2.970   7.250;...   % 44
-1.485   9.689;...   % 45
 4.454   4.812;...   % 46
 2.970   7.250;...   % 47
 1.485   9.689;...   % 48

 0.000  12.127;...   % 49    -

-1.333  14.317;...   % 50
-2.667  16.507;...   % 51
-4.000  18.696;...   % 52
 1.333  14.317;...   % 53
 2.667  16.507;...   % 54
 4.000  18.696;...   % 55

-4.000  22.947;...   % 56
-2.667  25.008;...   % 57
-1.333  27.068;...   % 58
 4.000  22.947;...   % 59
 2.667  25.008;...   % 60
 1.333  27.068;...   % 61

 0.000  29.129;...   % 62    -

-1.205  30.991;...   % 63
-2.410  32.854;...   % 64
-3.615  34.716;...   % 65
 1.205  30.991;...   % 66
 2.410  32.854;...   % 67
 3.615  34.716;...   % 68

-3.615  38.319;...   % 69
-2.410  40.060;...   % 70
-1.205  41.801;...   % 71
 3.615  38.319;...   % 72
 2.410  40.060;...   % 73
 1.205  41.801;...   % 74

 0.000  43.542;...   % 75    -

-1.096  45.126;...   % 76
-2.193  46.710;...   % 77
-3.289  48.294;...   % 78
 1.096  45.126;...   % 79
 2.193  46.710;...   % 80
 3.289  48.294;...   % 81

-3.289  51.349;...   % 82
-2.193  52.820;...   % 83
-1.096  54.291;...   % 84
```

```
            3.289  51.349;...    % 85
            2.193  52.820;...    % 86
            1.096  54.291;...    % 87

            0.000  55.762;...    % 88     -

           -1.004  57.109;...    % 89
           -2.008  58.457;...    % 90
           -3.012  59.804;...    % 91
            1.004  57.109;...    % 92
            2.008  58.457;...    % 93
            3.012  59.804;...    % 94

            0.000  61.650;...    % 95     -
            0.000  62.650;...    % 96     -
            0.000  73.650;...    % 97     -
            0.000  83.650;...    % 98     -
            0.000  95.650;...    % 99     -
            0.000 105.650;...    % 100    -
            0.000 115.650;...    % 101    -
            0.000 124.650;...    % 102    -
            0.000 129.650;...    % 103    -
            4.218  35.647];      % 104

nDOF = size(nodcoor,1)*3;

% Element nodes
elemnod = [ 1, 2;  2, 3;  3, 4;  4, 5;  5, 6;  6, 7;  7, 8;...   % Legs (1-14)
            9,10; 10,11; 11,12; 12,13; 13,14; 14,15; 15,16;...
            8,17; 17,18; 18,19; 19,20; 20,21; 21,22; 22,23; 23,24; 24,25;...
           25,26; 26,27; 27,28; 28,29;...   % Legs (15-40)
           16,30; 30,31; 31,32; 32,33; 33,34; 34,35; 35,36; 36,37; 37,38;...
           38,39; 39,40; 40,41; 41,42;...
            4,43; 43,44; 44,45; 45,49;...                   % X-braces (41-104)
           12,46; 46,47; 47,48; 48,49;...
           49,50; 50,51; 51,52; 52, 8;...
           49,53; 53,54; 54,55; 55,16;...
            8,56; 56,57; 57,58; 58,62;...
           16,59; 59,60; 60,61; 61,62;...
           62,63; 63,64; 64,65; 65,20;...
           62,66; 66,67; 67,68; 68,104; 104,33;...
           20,69; 69,70; 70,71; 71,75;...
           33,72; 72,73; 73,74; 74,75;...
           75,76; 76,77; 77,78; 78,24;...
           75,79; 79,80; 80,81; 81,37;...
           24,82; 82,83; 83,84; 84,88;...
           37,85; 85,86; 86,87; 87,88;...
           88,89; 89,90; 90,91; 91,28;...
           88,92; 92,93; 93,94; 94,41;...
            3,11; 29,95; 95,42;...                          % Horizontal (105-107)
           95,96; 96,97; 97,98; 98,99; 99,100; 100,101; 101,102; 102,103];
                                                            % Tower (108-115)
% Material properties
Es = 2.1e11;        % [N/m2]
Ec = 3e10;          % [N/m2]
rho_s = 7850;       % [kg/m3]
rho_c = 2000;       % [kg/m3]

elemE = [Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Es, Es, Es, Es, Es, Es, Es,...
         Es, Ec, Ec,...
         Es, Es, Es, Es, Es, Es, Es, Es];

elemrho = [rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s,...
           rho_s, rho_c, rho_c,...
           rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s, rho_s];

% Member properties

% Jacket
D1 = 0.8;
D2 = 1.2;
D3 = 1.2;
D4 = 1.2;
t1 = 0.02;
t2 = 0.05;
t3 = 0.035;
t4 = 0.04;
A1 = pi/4*(D1^2-(D1-2*t1)^2);
A2 = pi/4*(D2^2-(D2-2*t2)^2);
A3 = pi/4*(D3^2-(D3-2*t3)^2);
```

```
A4 = pi/4*(D4^2-(D4-2*t4)^2);
I1 = pi/64*(D1^4-(D1-2*t1)^4);
I2 = pi/64*(D2^4-(D2-2*t2)^4);
I3 = pi/64*(D3^4-(D3-2*t3)^4);
I4 = pi/64*(D4^4-(D4-2*t4)^4);

% Tower
% Dt = [5.600/1.95; 5.577/1.95; 5.318/1.95; 5.082/1.95; 4.800/1.95; 4.565/1.95;...
%       4.329/1.95; 4.118/1.95; 4.000/1.95];
% tt = [0.032/2; 0.032/2.3; 0.030/2; 0.028/2; 0.024/2; 0.022/2; 0.020/2;...
%       0.030/2; 0.030/2];
Dt = [5.600/1; 5.577/1; 5.318/1; 5.082/1; 4.800/1; 4.565/1; 4.329/1;...
      4.118/1; 4.000/1];
tt = [0.032/1; 0.032/1; 0.030/1; 0.028/1; 0.024/1; 0.022/1; 0.020/1;...
      0.030/1; 0.030/1];
Dte = zeros(length(Dt)-1,1);
tte = zeros(length(Dt)-1,1);
At = zeros(length(Dt)-1,1);
It = zeros(length(Dt)-1,1);
for i = 2:length(Dt)
    Dte(i-1) = (Dt(i-1)+Dt(i))/2;
    tte(i-1) = (tt(i-1)+tt(i))/2;
    At(i-1) = pi/4*(Dte(i-1)^2-(Dte(i-1)-2*tte(i-1))^2);
    It(i-1) = pi/64*(Dte(i-1)^4-(Dte(i-1)-2*tte(i-1))^4);
end

% Transition piece
b_TP = 9.6;                 % [m]
h_TP = 4;                   % [m]
l_TP = 8;                   % [m]
I_TP = b_TP*h_TP^3/12;      % [m4]
m_TP = 666000;              % [kg]
A_TP = m_TP/(rho_c*l_TP);   % [m2]

elemA = [A2, A2, A2, A2, A2, A2, A2, A2, A2, A2, A2, A2, A2, A2,...
         A3, A3, A3, A3, A3, A3, A3, A3, A3, A3, A3, A3, A3,...
         A3, A3, A3, A3, A3, A3, A3, A3, A3, A3, A3, A3, A3,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A1, A1, A1, A1, A1, A1, A1,...
         A1, A_TP/2, A_TP/2,...
         At(1), At(2), At(3), At(4), At(5), At(6), At(7), At(8)];

elemI = [I2, I2, I2, I2, I2, I2, I2, I2, I2, I2, I2, I2, I2, I2,...
         I3, I3, I3, I3, I3, I3, I3, I3, I3, I3, I3, I3, I3,...
         I3, I3, I3, I3, I3, I3, I3, I3, I3, I3, I3, I3, I3,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I1, I1, I1, I1, I1, I1, I1,...
         I1, I_TP/2, I_TP/2,...
         It(1), It(2), It(3), It(4), It(5), It(6), It(7), It(8)];

elemD = [D2, D2, D2, D2, D2, D2, D2, D2, D2, D2, D2, D2, D2, D2,...
         D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3,...
         D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, D1, D1, D1, D1, D1, D1, D1,...
         D1, 1, 1,...
         Dt(1), Dt(2), Dt(3), Dt(4), Dt(5), Dt(6), Dt(7), Dt(8)];

elemSubm = ones(size(elemnod,1),1);  % Element submerged? 0=false, 1=true
subm = [23:27,36:40,90:116];         % Insert elements that are NOT submerged
elemSubm(subm) = 0;

elemWind = zeros(size(elemnod,1),1); % Element subject to wind load? 0=false, 1=true
elemWind(109:116) = 1;               % Insert elements that are subject to wind

elemCrack = zeros(size(elemnod,1),1);  % Element cracked? 0=false, 1=true
elemCrack(73) = 1;

Kc = [0, 0.00157821, 0.00871166, 0.0215477, 0.0320845, 0.0438089, 0.0502119,...
      0.0560688]; % Crack depth (0.01 m 0.02 m 0.05 m 0.10 m 0.20 m 0.30 m 0.40 m)
Kc = Kc(cn);



%% System Matrices

% System mass and stiffness matrices M and K
[M,K] = SystemMatrices(nodcoor,elemnod,elemSubm,elemCrack,elemE,elemrho,...
                       elemA,elemI,elemD);

% Check for singularity of stiffness matrix separately in Mathematica

% Boundary conditions

% Top mass
% m_RNA = 350000/10;                   % Top mass in [kg] (NREL 5 MW 350t RNA)
```

```
m_RNA = 350000/1.5;
m_R = 110000;                           % Rotor mass: 110,000 kg
m_NA = 240000;                          % Nacelle mass: 240,000 kg
r_R = 63;                               % Rotor diameter: 63 m
l_NA = 14;                              % Length of nacelle: 14 m
h_NA = 3.5;                             % Height of nacelle: 3.5 m
I_R = m_R*r_R^2/4;
I_NA = 1/12*m_NA*(l_NA^2+h_NA^2);
I_RNA = I_R + I_NA;
% m_top = 8000;
M(307,307) = M(307,307) + m_RNA;
M(308,308) = M(308,308) + m_RNA;
M(309,309) = M(309,309) + I_RNA/3;

% Additional tower point masses
m_t1 = 1900;
% m_t1 = 500;
M(283,283) = M(283,283) + m_t1;
M(284,284) = M(284,284) + m_t1;

m_t2 = 1400;
% m_t2 = 300;
M(295,295) = M(295,295) + m_t2;
M(296,296) = M(296,296) + m_t2;

m_t3 = 1000;
% m_t3 = 200;
M(307,307) = M(307,307) + m_t3;
M(308,308) = M(308,308) + m_t3;

% Soil springs
k_x = 6.7821e+08;                     % Soil stiffness in x direction
k_xr = 1.7969e+09;                    % Soil stifness coupling between x and r
k_z = 2.1940e+09;                     % Soil stiffness in z direction
k_r = 1.3061e+10;                     % Soil stiffness in rotational direction

K(1,1) = K(1,1) + k_x;                % Leg 1
K(2,2) = K(2,2) + k_z;
K(3,3) = K(3,3) + k_r;
K(1,3) = K(1,3) + k_xr;
K(3,1) = K(3,1) + k_xr;

K(25,25) = K(25,25) + k_x;            % Leg 2
K(26,26) = K(26,26) + k_z;
K(27,27) = K(27,27) + k_r;
K(25,27) = K(25,27) + k_xr;
K(27,25) = K(27,25) + k_xr;

% Rayleigh damping matrix
% C = zeros(size(M,1),size(M,1));
C = 0.0001*M + 0.0001*K;
```

```
% Soil dashpots
c_x = 8.8800e+07;                     % Damping value in x direction
c_z = 2.8727e+08;                     % Damping value in z direction
c_r = 1.7101e+09;                     % Damping value in rotational direction
c_xr = 2.3527e+08;

C(1,1) = C(1,1) + c_x;                % Leg 1
C(2,2) = C(2,2) + c_z;
C(3,3) = C(3,3) + c_r;
C(1,3) = C(1,3) + c_xr;
C(3,1) = C(3,1) + c_xr;

C(25,25) = C(25,25) + c_x;            % Leg 2
C(26,26) = C(26,26) + c_z;
C(27,27) = C(27,27) + c_r;
C(25,27) = C(25,27) + c_xr;
C(27,25) = C(27,25) + c_xr;

% Mass matrix
eom.M = M;
Mi = M^(-1);
eom.Mi = Mi;

% Stiffness matrix
eom.K = K;

% Damping matrix
eom.C = C;

% Matrix multiplications
eom.MK = -eom.Mi * eom.K;
eom.MC = -eom.Mi * eom.C;



%% Hydrodynamic load

% Irregular wave properties
Hs = 6;                 % Significant wave height [m]
Tp = 10;                % Spectral peak period [s]
u_c = 1;                % Current velocity [m/s]
Cm = 1.6;               % Inertia coefficient
Cd = 0.8;               % Drag coefficient
d = 49.878;             % Water depth [m]
fp = 1/Tp;              % Spectral peak frequency [Hz]

% Frequency band
df = 0.001;
f = df:df:0.5;
% phase_JS = 2*pi*rand(length(f),1);      % Random phases
load('phase_JS')
```

```
S_JS = zeros(length(f),1);
zeta = zeros(length(f),1);
omega = zeros(length(f),1);
k = zeros(length(f),1);
eta = 0;


% JONSWAP spectrum
if Tp/sqrt(Hs) <= 3.6
    gamma = 5;
elseif Tp/sqrt(Hs) > 3.6 && Tp/sqrt(Hs) <= 5
    gamma = exp(5.75-1.15*Tp/sqrt(Hs));
else
    gamma = 1;
end

for i = 1:length(f)

    if f(i) <= fp
        sigma = 0.07;
    else
        sigma = 0.09;
    end

    S_JS(i) = 0.3125*Hs^2*Tp*(f(i)/fp)^(-5)*(1-0.287*log(gamma))...
              *exp(-1.25*(f(i)/fp)^(-4))*gamma^exp(-0.5*((f(i)-fp)/(sigma*fp))^2);

    zeta(i) = sqrt(2*S_JS(i)*df);   % Wave amplitude [m]
    omega(i) = 2*pi*f(i);           % Wave frequency [rad/s]
    k(i) = kSolve(omega(i),d);      % Wave number [rad/m]

    % Wave elevation
    eta = eta + zeta(i)*sin(omega(i)*t+phase_JS(i));
end



%% Wind load

% Wind properties
V_hub = 12;                         % Mean wind speed at hub height
Iref = 0.12;                        % Turbulence intensity
sigma1 = Iref*(0.75*V_hub+5.6);     % Standard deviation (invariant with height)

% Frequency band
df_wi = 0.001;
f_wi = df_wi:df_wi:0.5;
% phase_list = 2*pi*rand(10,length(f_wi));
load('phase_list')

S_KM = zeros(length(f_wi),1);
amp_wi = zeros(length(f_wi),1);
```

```
omega_wi = zeros(length(f_wi),1);
V_var = 0;

z_hub = 129.65;
if z_hub-d < 60
    Lk = 5.67*(z_hub-d);            % Integral scale parameter Lk
else
    Lk = 8.1*42;
end
I = sigma1/V_hub;

for j = 1:length(f_wi)
    S_KM(j) = 4*I^2*V_hub*Lk/((1+6*f_wi(j)*Lk/V_hub)^(5/3)); % Kaimal spectrum
    amp_wi(j) = sqrt(2*S_KM(j)*df_wi);                       % Wind amplitude [m/s]
    omega_wi(j) = 2*pi*f_wi(j);                              % Wind frequency [rad/s]
    V_var = V_var + amp_wi(j)*sin(omega_wi(j)*t+phase_list(1,j));
end

V_mean = V_hub;           % Mean wind speed at hub height
V = V_mean + V_var;



%% Force vector

f_hydro = zeros(nDOF,N);
f_wind = zeros(nDOF,N);
f_total = zeros(nDOF,N);
for w = 1:N
    v = t(w);
    f_hydro(:,w) = Hydro_Morison_irr_lin(v);
    f_wind(:,w) = Wind_Morison_lin(v);
    f_total(:,w) = f_hydro(:,w) + f_wind(:,w);
end

% Add harmonic force with w=0.305 Hz
% A_h = 50;           % [N]
% f_h = 0.305;        % [Hz]
% omega_h = 2*pi*f_h; % [rad/s]
% f_harm = A_h*sin(omega_h*t);
% f_total(70,:) = f_total(70,:) + f_harm;

% Wave velocities and accelerations
u_x1 = 0;
u_z1 = 0;
u_x2 = 0;
u_z2 = 0;
a_x1 = 0;
a_z1 = 0;
a_x2 = 0;
a_z2 = 0;
```

```
x1 = -4.385;
z1 = 49.878;
for l = 1:length(f)
    u_x1 = u_x1 + omega(l)*zeta(l)*cosh(k(l)*z1)/sinh(k(l)*d)...
                  *sin(omega(l)*t-k(l)*x1+phase_JS(l));
    u_z1 = u_z1 + omega(l)*zeta(l)*sinh(k(l)*z1)/sinh(k(l)*d)...
                  *cos(omega(l)*t-k(l)*x1+phase_JS(l));
    a_x1 = a_x1 + omega(l)^2*zeta(l)*cosh(k(l)*z1)/sinh(k(l)*d)...
                  *cos(omega(l)*t-k(l)*x1+phase_JS(l));
    a_z1 = a_z1 - omega(l)^2*zeta(l)*sinh(k(l)*z1)/sinh(k(l)*d)...
                  *sin(omega(l)*t-k(l)*x1+phase_JS(l));
end



%% Solution

% 1) Initialise solution
n_p = nDOF;                 % Number of forces
n_d = nDOF;                 % Number of observations
x0 = zeros(2*nDOF,1);       % Initial conditions

% Selection matrices
S_v = zeros(n_d,nDOF);      % Velocity
S_d = zeros(n_d,nDOF);      % Displacements
S_p = eye(nDOF);            % Forces
S_a = eye(n_d,nDOF);        % Accelerations (all degrees of freedom)


% 2) Construct the (discrete-time) state-space matrices A, B, C and D
% System matrix A
A_c = [zeros(nDOF,nDOF) eye(nDOF); -Mi*K -Mi*C];
A = expm(A_c*dt);

% Input matrix B
B_c = [zeros(nDOF,n_p); Mi*S_p];
B = (A-eye(2*nDOF))*A_c^(-1)*B_c;

% Output matrix G
G = [S_d-S_a*Mi*K S_v-S_a*Mi*C];

% Direct transmission matrix J
J = S_a*Mi*S_p;

% Force matrix
p = f_total;

% State matrices
state.A = A;
state.B = B;
state.G = G;
state.J = J;
```

```
% 3) Solve
x_k = zeros(2*nDOF,N);      % Preallocation
for time = 1:N
    %   Time update
    if time==1
        x_k(:,time) = A*x0;
    else
        x_k(:,time) = A*x_k(:,time-1)+B*p(:,time-1);
    end
end


% 4) Response determination
n_d = nDOF;                          % Number of output positions
% Displacements
S_d1 = eye(n_d,nDOF);
S_v1 = zeros(n_d,nDOF);
S_a1 = zeros(n_d,nDOF);
G1 = [S_d1-S_a1*Mi*K S_v1-S_a1*Mi*C];
J1 = S_a1*Mi*S_p;
u_k = G1*x_k+J1*p;                   % Displacements [n_d x N], here [nDOF x N]

% Velocities
S_d2 = zeros(n_d,nDOF);
S_v2 = eye(n_d,nDOF);
S_a2 = zeros(n_d,nDOF);
G2 = [S_d2-S_a2*Mi*K S_v2-S_a2*Mi*C];
J2 = S_a2*Mi*S_p;
v_k = G2*x_k+J2*p;                   % Velocities [n_d x N], here [nDOF x N]

% Accelerations
S_d3 = zeros(n_d,nDOF);
S_v3 = zeros(n_d,nDOF);
S_a3 = eye(n_d,nDOF);
G3 = [S_d3-S_a3*Mi*K S_v3-S_a3*Mi*C];
J3 = S_a3*Mi*S_p;
a_k = G3*x_k+J3*p;                   % Accelerations [n_d x N], here [nDOF x N]

% 4) Save
dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 2\Cracked ' cn_str '\300Hz
cd(dir);

save('eom.mat','eom')
save('state.mat','state');
save('u_k.mat','u_k');
save('v_k.mat','v_k');
save('a_k.mat','a_k');
save('t.mat','t');
save('p.mat','p');
save('eta.mat','eta');
save('V.mat','V');

toc
```

E.3 Function: Element Matrix

```
function [Me,Ke] = ElementMatrices(nodcoor,subm,crack,E,rho,A,I,D)
% ElementMatrices Establishes element stiffness and mass matrix
%   nodcoor: Nodal coordinates [2x2 matrix]
%   E: Young's modulus (assumed constant over length)
%   A: Cross-section area (assumed constant over length)
%   I: Area moment of inertia (assumed constant over length)
%   rho: Material density (assumed constant over length)

global Kc

% Local coordinates
x1 = nodcoor(1,1);
x2 = nodcoor(2,1);
z1 = nodcoor(1,2);
z2 = nodcoor(2,2);

L = sqrt((x2-x1)^2+(z2-z1)^2);      % Length [m]

% Stiffness and mass matrix
if ~crack                           % Element cracked?
    Kr = 0;
else
    Kr = Kc;
end

% Stiffness matrix with crack
a = A/(I*L);
b = 3*Kr^2+6*Kr*L+4*L^2;
c = 3*Kr^2+6*Kr*L+2*L^2;
Ke_loc = E*I*[ a       0               0 -a        0               0;
               0  12/L^3           6/L^2  0  -12/L^3           6/L^2;
               0   6/L^2  b/(L*(Kr+L)^2)  0   -6/L^2  c/(L*(Kr+L)^2);
              -a       0               0  a        0               0;
               0 -12/L^3          -6/L^2  0   12/L^3          -6/L^2;
               0   6/L^2  c/(L*(Kr+L)^2)  0   -6/L^2  b/(L*(Kr+L)^2)];

%     a = A*L^2/I;
%     Ke_loc = E*I/L^3*[ a    0      0  -a     0      0;      % Stiffness matrix
%                        0   12    6*L   0   -12    6*L;
%                        0  6*L  4*L^2   0  -6*L  2*L^2;
%                       -a    0      0   a     0      0;
%                        0  -12   -6*L   0    12   -6*L;
%                        0  6*L  2*L^2   0  -6*L  4*L^2];
% Body mass matrix
Mbe_loc = rho*A*L/420*[140      0       0   70      0       0;
                         0    156    22*L    0     54   -13*L;
                         0   22*L   4*L^2    0   13*L  -3*L^2;
                        70      0       0  140      0       0;
                         0     54    13*L    0    156   -22*L;
                         0  -13*L  -3*L^2    0  -22*L   4*L^2];

% Added mass matrix
if subm                     % Element submerged?
    rho_w = 1025;           % Density of sea water [kg/m3]
    Cm = 1.6;               % Inertia coefficient (see also Morison equation)
    Ca = Cm-1;              % Added mass coefficient
    V_d = pi*D^2/4*L;       % Displaced volume [m3]

    Mae_loc = rho_w*Ca*V_d/420*[140      0       0   70      0       0;
                                  0    156    22*L    0     54   -13*L;
                                  0   22*L   4*L^2    0   13*L  -3*L^2;
                                 70      0       0  140      0       0;
                                  0     54    13*L    0    156   -22*L;
                                  0  -13*L  -3*L^2    0  -22*L   4*L^2];

else
    Mae_loc = 0;
end

Me_loc = Mbe_loc + Mae_loc;     % Total mass matrix

% Transformation to global coordinate system
c = (x2-x1)/L;
s = (z2-z1)/L;

T = [c  s  0  0  0  0;
    -s  c  0  0  0  0;
     0  0  1  0  0  0;
     0  0  0  c  s  0;
     0  0  0 -s  c  0;
     0  0  0  0  0  1];

Ke = T.'*Ke_loc*T;
Me = T.'*Me_loc*T;
```

E.4 Function: System Matrix

```
function [M,K] = SystemMatrices(nodcoor,elemnod,elemSubm,elemCrack,elemE,...
                                elemrho,elemA,elemI,elemD)
% SystemMatrices Establishes system stiffness and mass matrix
%   nodcoor: List of nodal coordinates [nnod x 2 matrix]
%   elemnod: List of element nodes [nelem x 2 matrix]
%   elemE: List of elements' Young's moduli
%   elemA: List of elements' cross-section area
%   elemI: List of elements' area moment of inertia
%   elemrho: List of elements' material density

nnod = size(nodcoor,1);
nelem = size(elemnod,1);

M = zeros(3*nnod,3*nnod);
K = zeros(3*nnod,3*nnod);

for i = 1:nelem                  % Loop over all elements

    % Global coordinates
    nod1 = elemnod(i,1);
    nod2 = elemnod(i,2);
    eft = [3*nod1-2, 3*nod1-1, 3*nod1, 3*nod2-2, 3*nod2-1, 3*nod2];

    % Local coordinates
    elemnodcoor = [nodcoor(nod1,:); nodcoor(nod2,:)];

    % Member's properties
    E = elemE(i);
    rho = elemrho(i);
    A = elemA(i);
    I = elemI(i);
    D = elemD(i);

    % Member submerged?
    subm = elemSubm(i);

    % Member cracked?
    crack = elemCrack(i);

    % Element matrices
    [Me,Ke] = ElementMatrices(elemnodcoor,subm,crack,E,rho,A,I,D);

    % Assembly to global matrices
    for j = 1:6                             % Loop over all rows of el. matrix
        jj = eft(j);                        % jj: Global degree of freedom
        for k = 1:6                         % Loop over all columns of el. matrix
            kk = eft(k);                    % kk: Global degree of freedom
            M(jj,kk) = M(jj,kk) + Me(j,k);
            K(jj,kk) = K(jj,kk) + Ke(j,k);
        end
    end
end
```

E.5 Function: Aerodynamic Load

```
function f_wind = Wind_Morison_lin(t)
% Calculates force vector
%   t: Time [s]

global f_wi df_wi phase_list V_hub sigma1 d nodcoor elemnod elemD elemWind

rho_w = 1.225;   % Density of air [kg/m3]
Cd = 0.8;        % Drag coefficient

% Automatic sizing of y vector
aa = size(nodcoor,1)*3;
fd = zeros(aa,1);
counter = 0;

nelem = size(elemnod,1);    % Number of elements
for i = 1:nelem             % Loop over all elements

    % Global coordinates
    nod1 = elemnod(i,1);
    nod2 = elemnod(i,2);
    eft = [3*nod1-2, 3*nod1-1, 3*nod1, 3*nod2-2, 3*nod2-1, 3*nod2];

    if elemWind(i)
        counter = counter+1;
        phase_KM = phase_list(counter,:);

        % Local coordinates
        elemnodcoor = [nodcoor(nod1,:); nodcoor(nod2,:)];
        x1 = elemnodcoor(1,1);
        x2 = elemnodcoor(2,1);
        z1 = elemnodcoor(1,2);
        z2 = elemnodcoor(2,2);

        nod_hub = elemnod(end,2);   % Assuming hub has always highest node number
        z_hub = nodcoor(nod_hub,2);

        S_KM1 = zeros(length(f_wi),1);
        S_KM2 = zeros(length(f_wi),1);
        amp_wi1 = zeros(length(f_wi),1);
        amp_wi2 = zeros(length(f_wi),1);
        omega_wi = zeros(length(f_wi),1);
        V_var1 = 0;
        V_var2 = 0;

        alpha = 0.2;
        V_mean1 = V_hub*((z1-d)/(z_hub-d))^alpha;  % Mean wind speed at node 1
        V_mean2 = V_hub*((z2-d)/(z_hub-d))^alpha;  % Mean wind speed at node 2
        I1 = sigma1/V_mean1;
        I2 = sigma1/V_mean2;

        if z1-d < 60
            Lk1 = 5.67*(z_hub-d);            % Integral scale parameter Lk1
        else
            Lk1 = 8.1*42;
        end

        if z2-d < 60
            Lk2 = 5.67*(z_hub-d);            % Integral scale parameter Lk2
        else
            Lk2 = 8.1*42;
        end

        for k = 1:length(f_wi)
            % Kaimal spectrum node 1
            S_KM1(k) = 4*I1^2*V_mean1*Lk1/((1+6*f_wi(k)*Lk1/V_mean1)^(5/3));
            % Kaimal spectrum node 2
            S_KM2(k) = 4*I2^2*V_mean2*Lk2/((1+6*f_wi(k)*Lk2/V_mean2)^(5/3));
            amp_wi1(k) = sqrt(2*S_KM1(k)*df_wi);      % Wind amplitude [m/s]
            amp_wi2(k) = sqrt(2*S_KM2(k)*df_wi);
            omega_wi(k) = 2*pi*f_wi(k);               % Wind frequency [rad/s]
            V_var1 = V_var1 + amp_wi1(k)*sin(omega_wi(k)*t+phase_KM(k));
            V_var2 = V_var2 + amp_wi2(k)*sin(omega_wi(k)*t+phase_KM(k));
        end

        % Member's properties
        D = elemD(i);                       % Member's diameter [m]
        L = sqrt((x2-x1)^2+(z2-z1)^2);      % Member's length

        c = (x2-x1)/L; % cos(phi)
        s = (z2-z1)/L; % sin(phi)

        T = [c  s  0  0  0  0;      % Transformation matrix
            -s  c  0  0  0  0;
             0  0  1  0  0  0;
             0  0  0  c  s  0;
             0  0  0 -s  c  0;
             0  0  0  0  0  1];

        % Drag force term
        fd1 = 0.5*rho_w*Cd*D * ((V_mean1+V_var1)*s) * abs((V_mean1+V_var1)*s);
        fd2 = 0.5*rho_w*Cd*D * ((V_mean2+V_var2)*s) * abs((V_mean2+V_var2)*s);

        % Integration of linear wave force distribution
        fd_1 = 1/20*(13*fd1-3*fd2)*L;
        fd_2 = 1/60*(7*fd1-2*fd2)*L^2;
        fd_3 = 1/20*(17*fd1-7*fd2)*L;
        fd_4 = 1/60*(-8*fd1+3*fd2)*L^2;

        fde = [0;fd_1;fd_2;0;fd_3;fd_4];

        fde = T.'*fde;      % Tranformation

    else
        fde = zeros(6,1);
    end

    % Assembly to global force vector
    for m = 1:6
        mm = eft(m);
        fd(mm) = fd(mm)+fde(m);
    end
end

% 'Thrust' force
eft_hub = [3*nod_hub-2, 3*nod_hub-1, 3*nod_hub];
I = sigma1/V_hub;
if z_hub-d < 60
    Lk = 5.67*(z_hub-d);            % Integral scale parameter Lk1
else
    Lk = 8.1*42;
end

V_var_hub = 0;
S_KM = zeros(length(omega_wi),1);
amp_wi = zeros(length(omega_wi),1);
for p = 1:length(omega_wi)
    S_KM(p) = 4*I^2*V_hub*Lk/((1+6*f_wi(p)*Lk/V_hub)^(5/3));
    amp_wi(p) = sqrt(2*S_KM(p)*df_wi);
    V_var_hub = V_var_hub + amp_wi(p)*sin(omega_wi(p)*t+phase_KM(p));
end

V_thr = V_hub + V_var_hub;          % Same as V

% 'Rotor' properties
Cd_hub = 0.1;                % Higher drag coefficient to account for rotor
D_rot = 126;                  % Rotor diameter [m]
A_rot = pi*D_rot^2/4;         % Rotor area [m2]

% Drag force term
fd_hub = 0.5*rho_w*Cd_hub*A_rot*V_thr * abs(V_thr);      % Local

% Total global force vector
f_wind = fd;
f_wind(eft_hub(1)) = f_wind(eft_hub(1)) + fd_hub;
```

E.6 Function: Hydrodynamic Load

```
function f_hydro = Hydro_Morison_irr_lin(t)
% Calculates force vector
%   t: Time [s]

global f zeta omega k phase_JS u_c Cm Cd d nodcoor elemnod elemD elemSubm

rho_w = 1025;                           % Density of sea water [kg/m3]
```

```
% Automatic sizing of y vector
aa = size(nodcoor,1)*3;
fi = zeros(aa,1);
fd = zeros(aa,1);

nelem = size(elemnod,1);    % Number of elements
for j = 1:nelem             % Loop over all elements

    % Global coordinates
    nod1 = elemnod(j,1);
    nod2 = elemnod(j,2);
    eft = [3*nod1-2, 3*nod1-1, 3*nod1, 3*nod2-2, 3*nod2-1, 3*nod2];

    if elemSubm(j)

        % Local coordinates
        elemnodcoor = [nodcoor(nod1,:); nodcoor(nod2,:)];
        x1 = elemnodcoor(1,1);
        x2 = elemnodcoor(2,1);
        z1 = elemnodcoor(1,2);
        z2 = elemnodcoor(2,2);

        % Wave velocities and accelerations
        u_x1 = 0;
        u_z1 = 0;
        u_x2 = 0;
        u_z2 = 0;
        a_x1 = 0;
        a_z1 = 0;
        a_x2 = 0;
        a_z2 = 0;

        for l = 1:length(f)
            u_x1 = u_x1 + omega(l)*zeta(l)*cosh(k(l)*z1)/sinh(k(l)*d)...
                          *sin(omega(l)*t-k(l)*x1+phase_JS(l));
            u_z1 = u_z1 + omega(l)*zeta(l)*sinh(k(l)*z1)/sinh(k(l)*d)...
                          *cos(omega(l)*t-k(l)*x1+phase_JS(l));
            u_x2 = u_x2 + omega(l)*zeta(l)*cosh(k(l)*z2)/sinh(k(l)*d)...
                          *sin(omega(l)*t-k(l)*x2+phase_JS(l));
            u_z2 = u_z2 + omega(l)*zeta(l)*sinh(k(l)*z2)/sinh(k(l)*d)...
                          *cos(omega(l)*t-k(l)*x2+phase_JS(l));
            a_x1 = a_x1 + omega(l)^2*zeta(l)*cosh(k(l)*z1)/sinh(k(l)*d)...
                          *cos(omega(l)*t-k(l)*x1+phase_JS(l));
            a_z1 = a_z1 - omega(l)^2*zeta(l)*sinh(k(l)*z1)/sinh(k(l)*d)...
                          *sin(omega(l)*t-k(l)*x1+phase_JS(l));
            a_x2 = a_x2 + omega(l)^2*zeta(l)*cosh(k(l)*z2)/sinh(k(l)*d)...
                          *cos(omega(l)*t-k(l)*x2+phase_JS(l));
            a_z2 = a_z2 - omega(l)^2*zeta(l)*sinh(k(l)*z2)/sinh(k(l)*d)...
                          *sin(omega(l)*t-k(l)*x2+phase_JS(l));
        end
        % Member's properties
        D = elemD(j);                       % Member's diameter [m]
        A = 1/4*pi*D^2;                     % Member's cross-section area [m^2]
        L = sqrt((x2-x1)^2+(z2-z1)^2);      % Member's length

        c = (x2-x1)/L; % cos(phi)
        s = (z2-z1)/L; % sin(phi)

        T = [c  s  0  0  0  0;      % Transformation matrix
            -s  c  0  0  0  0;
             0  0  1  0  0  0;
             0  0  0  c  s  0;
             0  0  0 -s  c  0;
             0  0  0  0  0  1];


        % Inertia force term
        fi1 = rho_w*Cm*A*(a_x1*s+a_z1*c);
        fi2 = rho_w*Cm*A*(a_x2*s+a_z2*c);

        % Integration of linear wave force distribution from 0 to L
        fi_1 = 1/20*(13*fi1-3*fi2)*L;
        fi_2 = 1/60*(7*fi1-2*fi2)*L^2;
        fi_3 = 1/20*(17*fi1-7*fi2)*L;
        fi_4 = 1/60*(-8*fi1+3*fi2)*L^2;

        fie = [0;fi_1;fi_2;0;fi_3;fi_4];

        fie = T.'*fie;      % Tranformation

        % Drag force term
        fd1 = 0.5*rho_w*Cd*D * (u_x1*s+u_z1*c+u_c*s) * abs(u_x1*s+u_z1*c+u_c*s);
        fd2 = 0.5*rho_w*Cd*D * (u_x2*s+u_z2*c+u_c*s) * abs(u_x2*s+u_z2*c+u_c*s);

        % Integration of linear wave force distribution
        fd_1 = 1/20*(13*fd1-3*fd2)*L;
        fd_2 = 1/60*(7*fd1-2*fd2)*L^2;
        fd_3 = 1/20*(17*fd1-7*fd2)*L;
        fd_4 = 1/60*(-8*fd1+3*fd2)*L^2;

        fde = [0;fd_1;fd_2;0;fd_3;fd_4];

        fde = T.'*fde;      % Tranformation

    else

       fie = zeros(6,1);
       fde = zeros(6,1);

    end

    % Assembly to global force vector
```

```
    for m = 1:6
        mm = eft(m);
        fi(mm)  = fi(mm)+fie(m);
        fd(mm)  = fd(mm)+fde(m);
    end

end

% Total global force vector
f_hydro = fi+fd;
```

E.7 State-Space Representation

```
% Generate the discrete-time state space model and determine the filtered
% data by means of the Kalman filter

clear all
close all
clc

% Crack 'number'
cn = 8;                 % 1 to 8
cn_str = num2str(cn-1);

% 1) Generate and load the data and add noise
% dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Cracked ' cn_str
dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Cracked ' cn_str ''
% dir = ['W:\Master Thesis\UpWind Jacket\Response\Results\Jacket 1\Cracked ' cn_str

cd(dir);
load('eom.mat')
load('state.mat')
load('a_k.mat')
load('t.mat')           % Time/sample vector [N x 1]
load('p.mat')           % Force matrix [nDOF x N]

A = state.A;            % System matrix A
B = state.B;            % Input matrix B

dt = t(2)-t(1);         % Sample size/time step
nDOF = size(A,1)/2;     % Number of degrees of freedom
N = size(t,2);          % Number of samples
n_p = nDOF;             % Number of forces
n_d = 3;                % Number of observations


% 2) Create observation/measurement data
% Selection matrices
S_d = zeros(n_d,nDOF);      % Velocity
S_v = zeros(n_d,nDOF);      % Displacements
S_p = eye(nDOF);            % Forces
S_a = zeros(n_d,nDOF);      % Selection matrix for accelerations

a = a_k;                    % Acceleration matrix [nDOF x N]
loc = [70 79 91];           % Degrees of freedom where measured [a1 a2 ...]
S_a(1,loc(1)) = 1;
S_a(2,loc(2)) = 1;
S_a(3,loc(3)) = 1;
d = S_a*a;                  % Observation matrix [n_d x N]

% Add measurement noise
nl = 0.1;                  % Noise level
noise = zeros(n_d,N);
d_n = zeros(n_d,N);
for ind = 1:n_d
    noise(ind,:) = nl*std(d(ind,:))*randn(1,N);
    d_n(ind,:) = d(ind,:) + noise(ind,:);
end


% 3) Construct the (discrete-time) state-space matrices A, B, C and D
Mi = eom.Mi;                        % Inverse mass matrix [nDOF x nDOF]
C = eom.C;                          % Damping matrix [nDOF x nDOF]
K = eom.K;                          % Stiffness matrix [nDOF x nDOF]

G = [S_d-S_a*Mi*K S_v-S_a*Mi*C];    % Output matrix G [n_d x 2nDOF]
J = S_a*Mi*S_p;                     % Direct transmission matrix J [n_d x nDOF]


% 4) State estimation
Q = 1e-5*eye(nDOF*2);               % Process noise covariance matrix
R = 0.1*diag(std(noise,0,2));       % Measurement noise covariance matrix
x0 = zeros(nDOF*2,1);               % Initial state estimate
P0 = 1e-3*eye(nDOF*2);              % Initial state error covariance matrix

[x_id,P_id] = kalman_filter(A,B,G,J,Q,R,P0,x0,p,d_n);    % x_id [2nDOF x N]


% 5) Response prediction
n_d = nDOF;                   % Number of output positions
% Displacements
S_d1 = eye(n_d,nDOF);
S_v1 = zeros(n_d,nDOF);
S_a1 = zeros(n_d,nDOF);
G1 = [S_d1-S_a1*Mi*K S_v1-S_a1*Mi*C];
J1 = S_a1*Mi*S_p;
u_pr = G1*x_id+J1*p;          % Predicted displacements [n_d x N], here [nDOF x N]

% Velocities
S_d2 = zeros(n_d,nDOF);
S_v2 = eye(n_d,nDOF);
S_a2 = zeros(n_d,nDOF);
```

```
G2 = [S_d2-S_a2*Mi*K S_v2-S_a2*Mi*C];
J2 = S_a2*Mi*S_p;
v_pr = G2*x_id+J2*p;          % Predicted velocities [n_d x N], here [nDOF x N]

% Accelerations
S_d3 = zeros(n_d,nDOF);
S_v3 = zeros(n_d,nDOF);
S_a3 = eye(n_d,nDOF);
G3 = [S_d3-S_a3*Mi*K S_v3-S_a3*Mi*C];
J3 = S_a3*Mi*S_p;
a_pr = G3*x_id+J3*p;          % Predicted accelerations [n_d x N], here [nDOF x N]

% Save
% dir = ['W:\Master Thesis\UpWind Jacket\Kalman filter\Results\Jacket 1\Cracked ' cr
dir = ['W:\Master Thesis\UpWind Jacket\Kalman filter\Results\Jacket 1\Cracked ' cn_s
% dir = ['W:\Master Thesis\UpWind Jacket\Kalman filter\Results\Jacket 1\Cracked ' cr

cd(dir);
save('u_pr.mat','u_pr')
save('v_pr.mat','v_pr')
save('a_pr.mat','a_pr')
save('t.mat','t')
% save('P_id.mat','P_id')
```

E.8 Function: Kalman Filter

```
function [x_f,P_f] = kalman_filter(A,B,C,D,Q,R,P0,x0,p,d)

% DISCRETE-TIME STATE-SPACE MODEL:
% x_k+1 = A*x_k + B*p_k + w_k
% d_k = C*x_k + D*p_k + v_k
% Q:    process noise covariance [2*nDOF x 2*nDOF]
% R:    measurement noise covariance [2*nDOF x 2*nDOF]
% x0:   initial state estimate at t = 0 [2*nDOF x 1]
% P0:   initial error covariance at t = 0 [2*nDOF x 2*nDOF]
% N:    number of samples
% p:    applied forces [n_p x N]
% d:    data vector [n_d x N]

N = size(d,2);                  % Number of samples
nDOF = size(A,1)/2;             % Number of degrees of freedom
x_f = zeros(2*nDOF,N);          % Preallocation
P_f = zeros(2*nDOF,2*nDOF);     % Preallocation

for k = 1:N
    %   Time update
    if k==1
        x_f(:,k) = A*x0;
        P_f(:,:,k) = A*P0*A' + Q;
    else
```

```
        x_f(:,k) = A*x_f(:,k-1)+B*p(:,k-1);
        P_f(:,:,k) = A*P_f(:,:,k-1)*A' + Q;
    end

    %   Measurement update
    K_k = P_f(:,:,k)*C'*(C*P_f(:,:,k)*C'+R)^(-1);
    x_f(:,k) = x_f(:,k) + K_k*(d(:,k)-C*x_f(:,k)-D*p(:,k));
    P_f(:,:,1) = P_f(:,:,k) - K_k*C*P_f(:,:,k);
end
```

Bibliography

[1] European Wind Energy Association, ed. *The European Offshore Wind Industry. Key Trends and Statistics 1st Half Year 2013.* 2013.

[2] European Commission, ed. URL: `http://ec.europa.eu/energy/renewables/index_en.htm` (visited on 13th Dec. 2013).

[3] Engineering and Technology Magazine, ed. URL: `http://eandt.theiet.org/magazine/2014/05/windfarm-foundations.cfm` (visited on 21st July 2014).

[4] Ballast Nedam Beheer. *Snapshot of an Remotely Operated Vehicle.* 2013.

[5] A. Raine. 'Cost Behefit Applications Using the Alternating Current Field Measurement Technique'. In: *Finite Elements in Analysis and Design* 34.1 (2000), pp. 61–73.

[6] F. Vorpahl et al. *Description of a Basic Model of the 'UpWind Reference Jacket' for Code Comparison in the OC4 Project Under IEA Wind Annex 30.* Technical Report. Fraunhofer Institute for Wind Energy and Energy System Technology IWES, 2013.

[7] Carlos A. Felippa. *Introduction to Finite Element Methods.* Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado, 2004.

[8] M. B. Zaaijer. 'Foundation Models for the Dynamic Response of Offshore Wind Turbines'. In: *Proceedings of the International Conference on Marine Renewable Energy.* (Newcastle, UK). London: The Institute of Marine Engineering, Science and Technology, 2002.

[9] J. Carter et al. 'Analysis of Laterally Loaded Shafts in Rock'. In: *Journal of the Geotechnical Engineering* 118.6 (1992).

[10] M. Randolph et al. 'Analysis of Deformation of Vertically Loaded Piles'. In: *Journal of the Geotechnical Engineering Division, Proceedings of the American Society of Civil Engineers* 104.GT12 (1978).

[11] J. Jonkman et al. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development.* Technical Report. National Renewable Energy Laboratory, 2009.

[12] H. Okamura et al. 'A Cracked Column Under Compression'. In: *Engineering Fracture Mechanics* 1.3 (1969), pp. 547–564.

[13] R. D. Adams et al. 'A Vibration Technique for Non-Destructively Assessing the Integrity of Structures'. In: *Journal of Mechanical Engineering Science* 20.2 (1978), pp. 93–100.

[14] T. G. Chondros et al. 'Identification of Cracks in Welded Joints of Complex Structures'. In: *Journal of Sound and Vibration* 69.4 (1980), pp. 531–538.

[15] P. F. Rizos et al. 'Identification of Crack Location and Magnitude in a Cantilever Beam From the Vibration Modes'. In: *Journal of Sound and Vibration* 138.3 (1990), pp. 381–388.

[16] W. M. Ostachowicz et al. 'Analysis of the Effect of Cracks on the Natural Frequencies of a Cantilever Beam'. In: *Journal of Sound and Vibration* 150.2 (1991), pp. 191–201.

[17] Y. Narkis. 'Identification of Crack Location in Vibrating Simply Supported Beams'. In: *Journal of Sound and Vibration* 172.4 (1994), pp. 549–558.

[18] D. Liu et al. 'Crack Detection in Hollow Section Structures Through Coupled Response Measurements'. In: *Journal of Sound and Vibration* 261.1 (2003), pp. 17–29.

[19] S. Christides et al. 'One-Dimensional Theory of Cracked Bernoulli-Euler Beams'. In: *International Journal of Mechanical Sciences* 26.11–12 (1984), pp. 639–648.

[20] T. G. Chondros et al. 'A Continuous Cracked Beam Vibration Theory'. In: *Journal of Sound and Vibration* 215.1 (1998), pp. 17–34.

[21] A. D. Dimarogonas. 'Vibration of Cracked Structures: A State of the Art Review'. In: *Engineering Fracture Mechanics* 55.5 (1996), pp. 831–857.

[22] D. Y. Zheng et al. 'Vibration and Stability of Cracked Hollow-Sectional Beams'. In: *Journal of Sound and Vibration* 267.4 (2003), pp. 933–954.

[23] G. Gounaris et al. 'A Finite Element of a Cracked Prismatic Beam for Structural Analysis'. In: *Computers & Structures* 28.3 (1988), pp. 309–313.

[24] M. Krawczuk et al. 'Elastic Beam Finite Element With a Transverse Elasto-Plastic Crack'. In: *Finite Elements in Analysis and Design* 34.1 (2000), pp. 61–73.

[25] M. Skrinar. 'Dynamical Analysis of Frame Type Structures with Local Stiffness Reductions by a New Finite Element Based on the Dynamic Stiffness and Mass Matrices'. In: 11th International Conference on Computational Methods and Experimental Measurements. (Halkidiki, Greece). 2003, pp. 3–12.

[26] H. Yoon et al. 'Dynamic Behaviour of Cracked Simply Supported Pipe Conveying Fluid with Moving Mass'. In: *Journal of Sound and Vibration* 292.3–5 (2006), pp. 941–953.

[27] European Committee for Electrotechnical Standardization, ed. *IEC 61400-3. Wind Turbines - Part 3: Design Requirements for Offshore Wind Turbines.* Apr. 2009.

[28] Det Norske Veritas, ed. *DNV-OS-J101. Design of Offshore Wind Turbine Structures*. Sept. 2011.

[29] J. M. J. Journée et al. *Offshore Hydromechanics*. Delft University of Technology, 2001.

[30] O. M. Faltinsen. *Sea Loads on Ships and Offshore Structures*. Cambridge University Press, 1990.

[31] T. Burton. *Wind Energy Handbook*. John Wiley & Sons, Ltd., 2011.

[32] European Committee for Electrotechnical Standardization, ed. *IEC 61400-1. Wind Turbines - Part 1: Design Requirements*. Aug. 2005.

[33] K. M. Schweigler. 'Aerodynamic Analysis of the NREL 5-MW Wind Turbine using Vortex Panel Methods'. Master's Thesis. Gothenburg, Sweden: Chalmers University of Technology, 2012.

[34] E. Lourens. 'Force Identification in Structural Dynamics'. PhD Thesis. Leuven, Belgium: Faculty of Engineering, Katholieke Universiteit Leuven, 2012.

[35] G. Welch et al. *An Introduction to the Kalman Filter*. Technical Report TR 95-041. Department of Computer Science, University of North Carolina at Chapel Hill, 2006.

[36] N. K. Vemula. *Design Solution for the UpWind Reference Offshore Support Structure. Deliverable D4.2.5*. Technical Report. UpWind WP4 Offshore Foundations and Support Structures, 2013.

[37] J. van der Tempel. 'Design of Support Structures for Offshore Wind Turbines'. PhD Thesis. Delft, Netherlands: Offshore Engineering, Technische Universiteit Delft, 2006.

Nomenclature

δu	Virtual horizontal displacement	[m]
ϵ	Phase angle	[rad]
$\mathbf{A}$	System matrix	
A	Cross-section area	[m^2]
a_x	Horizontal fluid acceleration	[m/s^2]
a_z	Vertical fluid acceleration	[m/s^2]
$\mathbf{B}$	Curvature-displacement matrix	
$\mathbf{B}$	Input matrix	
$\mathbf{B}$	Strain-displacement matrix	
C	Local flexibility coefficient	[rad/Nm]
C_{D}	Drag coefficient	
C_{M}	Inertia coefficient	
δw	Vertical virtual displacement	[m]
$\mathbf{d}$	Output/measurement vector	
D	Diameter	[m]
D	Diameter	[m]
ϵ	Strain	[–]
E	Young's modulus	[N/m^2]
$\mathbf{f}$	Force vector	
f	Frequency	[Hz]
f_{p}	Peak frequency	[Hz]
$\mathbf{G}$	Output matrix	
h	Height	[m]
H_{s}	Significant wave height	[m]
I	Area moment of inertia	[m^4]
I	Mass moment of inertia	[kgm^2]
$\mathbf{J}$	Direct transmission matrix	
$\mathbf{K}$	Stiffness matrix	
K	Crack coefficient	[m]
k	Wave number	[rad/m]
K_{r}	Rotational spring coefficient	[Nm/rad]
L	Length	[m]
l_{NA}	Length	[m]
$\mathbf{M}$	Mass matrix	
M	Bending moment	[Nm]
m	Mass	[kg]
$\mathbf{N}$	Shape function matrix	

N	Normal force	[N]
n	Distributed axial line load	[N/m]
n_{DOF}	Number of degrees of freedom	
n_{d}	Number of outputs/measurements	
n_{p}	Number of inputs/forces	
ω	Wave frequency	[rad/s]
$\mathbf{P}$	A posteriori error covariance	
$\mathbf{P}^{-}$	A priori error covariance	
$\mathbf{P}_0$	Initial a posteriori error covariance	
ϕ	Rotation of neutral axis	
ϕ	Wave potential	
φ	Rotation angle	[°]
$\mathbf{Q}$	Process noise covariance	
q	Vertical distributed load	[N/m]
$\mathbf{R}$	Measurement noise covariance	
ρ	Mass density	[kg/m^3]
σ	Stress	[N/m^2]
$\mathbf{T}$	Transformation matrix	
θ	Rotation of the cross-section	
t	Time coordinate	[s]
T_{p}	Spectral peak period	[s]
$\ddot{u}$	Horizontal acceleration	[m/s^2]
$\dot{u}$	Horizontal velocity	[m/s]
$\dot{u}_{\mathrm{rel}}$	Relative acceleration between water flow and structure's motion	[m/s^2]
$\mathbf{u}$	Vector with degrees of freedom	
u	Horizontal displacement	[m]
u	Horizontal wave velocity	[m/s]
u_{rel}	Relative velocity between water flow and structure's motion	[m/s]
V	Shear force	[N]
V_{r}	Rated wind speed	[m/s]
w	Vertical displacement	[m]
w	Vertical wave velocity	[m/s]
w'	Slope of w	
w''	Curvature of w	
$\hat{\mathbf{x}}$	A posteriori state estimate	
$\hat{\mathbf{x}}^{-}$	A priori state estimate	
$\hat{\mathbf{x}}_0$	Initial state vector	
$\mathbf{x}$	State vector	
ξ	Natural coordinate	[–]
x	Space coordinate	[m]
ζ	Wave elevation	[m]
ζ_{a}	Wave amplitude	[m]
z	Vertical space coordinate	[m]