Hydrodynamic Modeling of a Catenary-Moored Floating Offshore Wind Turbine using the Moving Frame Method

Olga Rakvåg

Thesis for Master of Science Degree at the University of Bergen, Norway

Supervised by: Dr. David Lande-Sudall

2024

Abstract

Floating offshore wind turbines are seen as vital to harnessing the greater power potential of higher wind speeds offshore, in deep water. Since offshore turbines are coupled with mooring lines, an advanced model of mooring system is often required.

In this project, a novel approach of mathematical modelling of mooring system is presented. It bases on Moving Frame Method, extensively utilizing Cartan's concept of Moving Frames, theory of Lie Groups, especially Special Euclidean SE(3) and Special Orthogonal SO(3) groups, combined with concise notation derived by Frankel. By means of these principles, the multi-segmented model of mooring line has been developed.

Hydrodynamic loads have been extended by second order forces, in the form of sum- and difference frequency forces.

Equation of motion have been obtained via Principles of Virtual Work and Hamilton's Principle. The resulting equation is then solved numerically using a fourth-order Runge-Kutta method and generalized coordinates are obtained. The overall dynamic response of FOWT is analysed in frequency domain and both wave frequency range (WF) and low frequency range (LF) are investigated.

As a larger goal, this project aims to establish a strong base for utilizing MFM in more advanced analyses that can handle intricate influences. While main focus of MFM is kinetics and robotics, this study has ambiguous idea to open the doors for MFM theory to the world of mooring line systems, enhancing also the latter one. Advantages of both world should meet in one project , and therefore a new, advanced dynamics can cooperate with hydrodynamics.

Preface

This project is a part of integrated master's programme in Ocean Technology at the University of Bergen in cooperation with Western Norway University of Applied Sciences.

Dynamics and hydrodynamics of offshore wind turbine is the theme of this thesis supervised by Dr. David Lande-Sudall.

I am grateful for all the encouragement I have received from my supervisor and for the support I have gotten from Bjørn Tore Hjertaker and Irlin Nyland during all the years of my education.

Table of Contents

List of Figures

List of Tables

Nomenclature

$[B]$	B-matrix
C	Center of mass
$[D]$	Combined angular velocity matrix
E	Frame connection matrix
$\dot{E}$	Time derivative of frame connection matrix
e	Frame
$\{F^*\}$	Generalized force
$\{F\}$	Force and moment list
H	Angular Momentum
$\{H\}$	Generalized momenta
I_3	3×3 Identity matrix
$J_{(a)}$	3×3 Mass moment of inertia matrix
J_C	Joint
K	Kinetic energy
$\{L\}$	Linear momentum
L	Linear momentum
Ł	Lagrangian
$[M]$	Mass matrix
$[M^*]$	Reduced mass matrix
$[N^*]$	Reduced nonlinear velocity matrix
q	Generalized position
$\dot{q}$	Generalized velocity
$\ddot{q}$	Generalized acceleration
$\{q\}$	General velocity variable list
$\{\dot{q}\}$	Generalized acceleration variable list
R	Rotation matrix
$\dot{R}$	Time derivative of rotation matrix
r	Absolute position vector
s	Position vector

U	Potential energy
δU	Variation of potential energy
δW	Virtual work
$\{X\}$	Velocity list
$\{x\}$	Virtual displacements
x	Position vector
$\dot{x}$	Linear velocity vector
α	Index counter
δ	Variation
$\delta\Pi$	Variation of frame connection matrix
θ	Angle
$\delta\theta$	Virtual rotational displacement
Ω	Time rate of the frame connection matrix
ω	Angular velocity vector
$\tilde{\omega}$	Skew symmetric angular velocity matrix
FOWT	Floating Offshore Wind Turbines
MFM	Moving frame method
FSI	Fluid-Structure Interactions
$RK4$	4th order Runge-Kutta method
PVW	Principle of Virtual Work
EOM	Equation of Motion

1 Introduction

In this project a new model of mooring line is developed and incorporated into catenary mooring line system of spar-buoy type floating offshore wind turbine (FOWT), aiming to contribute to development of existing renewable energy technologies and sustainable industrialization. The thesis investigates the dynamic behavior of a FOWT by applying an advanced numerical computational Moving Frame Method (MFM).

The dynamic behaviour of FOWTs is influenced by a variety of factors, including wind, currents, and waves, which greatly contribute to its translational and rotational movement. As a result, analysing FOWTs becomes a complex multi-body dynamic problem. However, by utilizing the MFM, this process can be generalized by calculating the translational and rotational motion, expressed by generalized coordinates, presented in chapter in Chapter 4.3 for platform and RNA and in Section 5.2 for mooring lines.

The project is in line with the global Paris Agreement and the European Green Deal which is at the Core of European Union's ambition to become climate neutral by 2050, meeting several Sustainable Development Goals (SDGs).

The main contribution targets Goal 7: Affordable and Clean Energy, Goal 9: Industry, Innovation, and Infrastructure, and Goal 13: Climate Action [7], [8], [9]. But indirectly other sustainable goals are met as well, such as Goal 8: Decent Work and Economic and Goal 12: Responsible Consumption and Production [10], [11].

When it comes to bigger scale problem, it worth mentioning that, FOWT generates renewable energy, ensuring everyone has access to inexpensive, dependable, clean green energy, while consuming much less fuel derived from fossils [7]. Offshore turbines have an advantage compared to land based turbines, since offshore areas generally experiences a stronger and more regular wind, increasing the production of renewable energy, because the power generated by a wind generator is a function of the cube of wind velocity. FOWT plays a crucial part in reducing climate change through production of non-polluting power that reduces greenhouse effect[9].

In addition, FOWTs' deployment gives rise to innovation in renewable energy technologies by driving resilient infrastructure development that can foster growth within the sector of offshore wind industry[8]. Addressing the latter SDG target, smaller challenges are introduced and solved by applying MFM for modelling of offshore turbine of type spar buoy and catenary type of mooring lines.

1.1 Problem Statement and Objectives

The project focuses on tackling several issues, one of them concerns improvement of multi-body system modelling, such as FOWT and mooring system, by addressing effective representation of multi-body system in 3D space.

Although existing methods in dynamics for modelling of complex systems are reliable, they are non-extensible due to some of the modelling issues, such as effective representation of translational and rotational motions of physical bodies in 3D space [12]. The uniqueness of the Moving Frame Method (MFM) lies in deriving equations of motion (EOM) from complex spatial movements of multi-body systems, such as FOWT, while retaining the simplicity of Cartan's notation in both 2D and 3D spaces [13], [14]. Employing MFM in this context allows to avoid the customary limitations of free-body diagrams, while facilitating the integration of joint reactions and hydrodynamic forces, not only for a better dynamic response of the system but also making it easier to compare with experimentation as well as any other modelling approach without much deviation in their results. The overreliance on vector algebra is reduced to simplicity and replaced by a Lie algebraic framework, making appliance of MFM promising, since it offers a more efficient way of modelling complex dynamics of both single and multi-bodies. In Chapter 4.3 a multi-body system is represented by an assembly of platform, tower with RNA (nacelle and rotor) and by a multi-segmented model of catenary line in Section 5.2.

Another issue this project faces is challenges related to implementation of hydrodynamics into MFM-based model. Since, according to principles of virtual work (PVW), the hydrodynamic forces applied to multi-body system must be at first defined in Cartesian coordinates, before their transformation into generalized coordinates, as presented in Chapter 6. This means, that the forces acting on the multi-body system, should be defined for each body separately, which can be done by linearisation of a well-known equation presented by W.E. Cummins [15]. The process is described in Chapters 3 and Chapter 6.

However, to investigate non-linear motion of the system, it is necessary to incorporate second order forces and such effects as waves loads of sum and difference-frequencies [16]. Utilization of second order transfer function (QTF) enables implementation of the second order forces into hydrodynamic loads, that acts on the system.

Then, the whole virtual work of a system is used for derivation of the equation of motion, which in the form of second-order ordinary differential equation is solved via Runge-Kutta numerical method.

From the above, the main objectives of the thesis are deduced as follows:

- Development of multi-segmented model of FOWT and catenary mooring line, α-bodies, based on Moving Frame Method, with further integration to spar- buoys mooring system. Description of the process is found in Chapter 4.3 for FOWT, and for MFM-based mooring line in Section 5.2.

- Implementation of the hydrodynamic forces into the full-coupled MFM model of FOWT, and incorporating as well second order forces, represented by difference-frequency second-order wave forces, through QTF's from OrcaFlex[17], [18], as shown in Section 3.1.1.

- Simulation in irregular seas, by solving the equations of motion (EOM), presented in the form of differential equations through 4th order Runge-Kutta method, as described in Chapter 7.

2 Literature Overview

Recent researches have revealed that FOWT is susceptible to unexpected movements in the lower frequency range [19]. First order excitation forces takes mostly into account response of FOWT with the same frequency as the passing wave. While a turbine will additionally be subjected to forces in the low frequency range. Difference-frequency wave excitation on a FOWT vary slowly and can create resonance if they coincide with the eigenfrequency or period of FOWT. Therefore, the results of such a research can be crucial for risk assessment and are important for increasing safety. Depending on environmental conditions, there is a need to incorporate second order forces into MFM model, through implementation of second order transfer function (QTF), to be able to better model the hydrodynamics of the FOWT.

Wave-induced second-order forces significantly affect the design and study of spar-buoy platforms through their role in the slow-drift motion and global stability of the system. Extensive work on this topic has been summarized by Weggel and Roesset's study conducted in 1996 which has produced valuable information on the dynamic behaviour of the immersed object in waves [20]. Over the recent years, researchers have been focusing on enhancement of accuracy and efficiency of mooring line models when subjected to hydrodynamic forces. The work of Ormberg [21], presents enhanced modelling techniques which take into account complex behaviour of mooring system under hydrodynamic load.

Moreover, the advancement of traditional particle dynamics methods particularly in the area of mooring line systems cannot remain unacknowledged. These have been altered to reflect the complex behavior of mooring lines under different load situations in different ways. It is the assertion of recent research like that of Kim and Park [22] that such models based on traditional particle dynamics can significantly enhance the simulation of mooring line dynamics because they simulate contact and deformation more accurately.

It has been thought that there should be created a more advanced mooring system, that would be able to withstand extreme weather conditions, minimising losses and damage.

One critical area of research is the hydrodynamic modeling of mooring systems, which focuses on understanding the effect of wave, current and wind forces. Computational Fluid Dynamics (CFD) have made it possible to simulate these forces more accurately and to understand how mooring lines respond to them. The work done by Bai and Teng [23] is a notable achievement that has seen the development of a holistic model that combines hydrodynamic forces with structural responses of mooring lines thus offering better predictive capability.

Even with all of these improvements, there are still some issues when it comes to accurately predicting how mooring lines move. One major problem is developing an integrated model comprising numerous forces and material properties that would allow it to perform well during real-time simulations.

This project attempts to solve this problem.

3 Hydrodynamics

In order to accurately predict the hydrodynamic interaction between the fluid and the platform, loads that influence the system over time are considered. To represent the hydrodynamic forces acting on a dynamic system in the time-domain, we can use Cummins equation, Eqn. 3.1, that shows how a marine structure responds to waves, separating the variables and not relying on added mass frequency. Cummins equation shows how to represent the equations of motion in the time-domain.

$$(M+A_{\infty})\,\ddot{q}^{\,tot}+\int_0^t K(t-\tau)\dot{q}^{\text{tot}}(\tau)d\tau+C_{\text{hydro}}\,q^{\text{ tot}}=F_{ex}+f_{external} \tag{3.1}$$

Here

M is the mass and inertia matrix of the float

A_{∞} is the added-mass matrix

$\int_0^t K(t-\tau)\dot{q}^{\text{tot}}(\tau)d\tau$ is radiation forces

C_{hydro} is the hydrostatic restoring forces

$f_{external}$ displays external forces that acts on the platform

F_{ex} is excitation force, represented by the sim of the first and second order excitation forces ${F_{ex}}^{(1)}$ and ${F_{ex}}^{(2)}$

The forces outlined above by the Cummins equation establish an essential foundation for constructioin of the force vector F^*, Eqn. 6.19, and processed in the equations of motion Eqn. 6.29. This force vector is structured in a way that is conducive to numerical integration using the RK4 method.

3.1 Excitation Force

Excitation force F_{ex} is often refered as a wave excitation force. It consists of first and second-order excitation loads which are described by the terms ${F_{ex}}^{(1)}$ and ${F_{ex}}^{(2)}$, see Eqn. 3.2.

$$F_{ex} = {F_{ex}}^{(1)} + {F_{\mathrm{e\,x}}}^{(2)} \tag{3.2}$$

The wave excitation force comprises two parts. First, there is the Froude-Krylov force. This occurs when the pressure field from undisturbed waves is integrated. Next is the diffraction force, caused by wave scattering when a body is present.

The first-order loads caused by single incoming wave trains are computed utilizing Eqn. 3.3

$${F_{ex}}^{(1)}{}_i = \mathrm{Re}\left(AX_i e^{j\omega t}\right), \quad i = 1, 2, \ldots, 6 \tag{3.3}$$

Here, A symbolizes the complex incoming wave's complex amplitude, providing amplitude and phase, j stands for the imaginary number $\sqrt{-1}$, while X_i describes the first-order load i^{th} component for each amplitude. For long-crested irregular sea states lacking directional spreading, the total load often is represented as a superposition of different wave-frequency components.

$${F_{ex}}^{(1)}{}_i = \mathrm{Re}\left(\sum_{k=1}^{N} A_k X_i\left(\omega_k\right) e^{j\omega_k t}\right), \quad i = 1, 2, \ldots, 6, \quad \omega_k = (k-1)d\omega \tag{3.4}$$

3.1.1 Second-Order Force

The long-crested second order wave model, is the simples representation of the wave propagation with N^2 corrections that spreads over all the sum frequencies, and another N^2 corrections covering all difference frequencies.

Longuett-Higgins developed a simple model of random wave for long-crested second order wave, expressed in exponential form as second order wave excitation force ${F_{ex}}^{(2)}$ Eq. 3.5 [24]

$$ {F_{ex}}^{(2)}{}_i = \mathrm{Re}\left(\sum_{m=1}^{N}\sum_{n=1}^{N}\left[A_m A_n QTF_i^{+}\left(\omega_m,\omega_n\right)e^{j(\psi_m+\psi_n)t} + A_m {A_n}^{*} QTF_i^{-}\left(\omega_m,\omega_n\right)e^{j(\psi_m-\psi_n)t}\right]\right) \tag{3.5} $$

As seen from Eqn. 3.5, the model of the wave includes additional N^2 terms, and contains sum frequency component, defined as $QTF_t^{+}(\omega_m,\omega_n)$, and difference frequencies component. Together they form a quadratic transfer function of surface elevation. QTF has a symmetric nature and therefore can be expressed by the Eqn. 3.6.

$$ QTF_t^{+}(\omega_m,\omega_n) = QTF_t^{+}(\omega_n,\omega_m) \quad \text{and} \quad QTF_t^{-}(\omega_m,\omega_n) = QTF_t^{-}(\omega_n,\omega_m)^{*} \tag{3.6} $$

Here the asterisk $(*)$ is the complex conjugate, ω_m and ω_n are wave frequencies.

Utilizing Euler's notation, the real part of the Eqn. 3.5 is written as:

$$ \begin{aligned} F^{(2+)} &= \sum_{m=1}^{N}\sum_{n=1}^{N} A_m A_n QTF_{mn}^{+}\cos\left(\psi_m+\psi_n\right) \\ F^{(2-)} &= \sum_{m=1}^{N}\sum_{n=1}^{N} A_m A_n QTF_{mn}^{-}\cos\left(\psi_m-\psi_n\right) \end{aligned} \tag{3.7} $$

The correction terms of Eqn. 3.7 including sub-harmonics and super-harmonics are found as sum - and difference frequency components:

$$ \begin{aligned} \psi_m+\psi_n &= (\omega_m+\omega_n)t+\phi_m+\phi_n \\ \psi_m-\psi_n &= (\omega_m-\omega_n)t+\phi_m-\phi_n \end{aligned} \tag{3.8} $$

Therefore Eqn. 3.5 is expressed in equivalent form:

$$ \begin{aligned} F^{(2)} = &\sum_{n=1}^{N}\sum_{m=1}^{N} A_m A_n QTF_{mn}^{+}\cos\left[(\omega_m+\omega_n)t+(\varepsilon_m+\varepsilon_n)\right] \\ &+\sum_{n=1}^{N}\sum_{m=1}^{N} A_m A_n QTF_{mn}^{-}\cos\left[(\omega_m-\omega_n)t+(\varepsilon_m-\varepsilon_n)\right] \end{aligned} \tag{3.9} $$

The amplitude of QTF in equation Eqn. 3.7 is presented in the form of a symmetric matrix, as shown in Table 3.1, where diagonal coefficients form the mean drift force, $\overline{F_i}$, calculated by direct pressure integration over the wet surface in equation Eqn. 3.10. While the off-diagonal coefficients are related to the slow drift force, F_i^{SV} [16].

$$\overline{F_i} = \frac{\rho g \zeta_a^2}{2} \int_{L_1} sin^2(\theta + \beta)\eta_i dl \tag{3.10}$$

where ζ_a is the wave amplitude, β is the propagation direction of a wave and θ is the directional heading of the wave (in global coordinates).

Based on Eqn. 3.9 second order wave excitation forces are calculated, in the form of sum- and difference frequency forces. QTF's are obtained in OrcaWave, quadratic and potential loads are summed up, as it is defined by Orcina, described in Eqn. 3.11 [17].

$\omega_j = \omega_5$	$Sym.$	$Sym.$	$Sym.$	$Sym.$	$\lvert QTF_{55}\rvert$
$\omega_j = \omega_4$	$Sym.$	$Sym.$	$Sym.$	$\lvert QTF_{44}\rvert$	$\lvert QTF_{54}\rvert$
$\omega_j = \omega_3$	$Sym.$	$Sym.$	$\lvert QTF_{33}\rvert$	$\lvert QTF_{43}\rvert$	$\lvert QTF_{53}\rvert$
$\omega_j = \omega_2$	$Sym.$	$\lvert QTF_{22}\rvert$	$\lvert QTF_{32}\rvert$	$\lvert QTF_{42}\rvert$	$\lvert QTF_{52}\rvert$
$\omega_j = \omega_1$	$\lvert QTF_{11}\rvert$	$\lvert QTF_{21}\rvert$	$\lvert QTF_{31}\rvert$	$\lvert QTF_{41}\rvert$	$\lvert QTF_{51}\rvert$
	$\omega_i = \omega_1$	$\omega_i = \omega_2$	$\omega_i = \omega_3$	$\omega_i = \omega_4$	$\omega_i = \omega_5$

Table 3.1: $\left|QTF(\omega_i, \omega_j)\right|$ matrix for five wave frequencies [6].

$$QTF_{OrcaFlex} = F_q + F_p \tag{3.11}$$

Where F_q is quadratic load, and F_p is potential load.

Mean drift load have been mapped onto matrix of quadratic load, since potential load falls to zero for zero -difference frequencies , i.e. $\Delta\omega = 0$ when $\omega_i = \omega_j$. The overview over QTFs in all 6 DOF and MATLAB code are presented in Appendix A.

3.2 Radiation forces

The radiation forces act in the opposite direction to the excitation forces as they are emitted away from the platform. The convolution integral $\int_0^t K(t-\tau)\dot{q}^{\mathrm{tot}}\,(\tau)d\tau$ determines radiation force by taking into account the memory effects of the fluid and relies on past time, τ [15]. The radiation impulse response function, $K(t)$ is given by integral:

$$K(t) = \frac{2}{\pi}\int_0^\infty B(\omega)\cos(\omega t)d\omega \tag{3.12}$$

Here $B(\omega)$ represents the amplitude of the response components to a unit amplitude forcing function with frequency, ω [15].

3.3 Added mass forces

Added mass forces are caused by the water's inertia around the body. These forces are important for simulating the time-dependent movement of an irregular sea state. Under waves of difference periods, the amount of added mass varies. To separate these contributions out, Cummin has eveloped this equation [15]. Added forces are calculated at infinite frequency by letting ω approach inifinity, as given in Eqn. 3.13.

$$A_\infty = \lim_{\omega\to\infty} A_(\omega) \tag{3.13}$$

3.4 External forces

External forces applied on the model are represented by a combination of several forces acting on the system.

3.4.1 Drag force

The drag term of Morison's equation is applied to calculate the drag forces acting on the platform. The viscous drag force opposes the relative motion of an object relative to the fluid.

$$F_H = -\frac{1}{2}\rho r d_1 C_{d_x} \dot{x}|\dot{x}| \tag{3.14}$$

$$F_V = -\frac{1}{2}\rho \pi r^2 C_{d_z} \dot{z}|\dot{z}| \tag{3.15}$$

$$M_{\text{Drag}} = -F_V dz - F_H \frac{d}{2} \tag{3.16}$$

Here C_{d_x} and C_{d_z} are the drag coefficients, $\dot{x}$, $\dot{z}$ is the relative fluid velocity, while d and r are diameter and radius of the platform. F_{Drag} include all the drag components calculated above:

$$F_{\text{Drag}} = \begin{bmatrix} F_H \\ F_V \\ M_{\text{Drag}} \end{bmatrix}. \tag{3.17}$$

3.4.2 Hydrostatic restoring force

The hydrostatic restoring force is the force acting on a body caused by the change in hydrostatic pressure on its wetted surface as it moves from its original position. Only heave and pitch displacements are concerned, since there is no restoring force in horisontal direction. The hydrostatic stiffness for vertical (heave) S_z and rotational (pitch) S_θ directions, as well as the restoring forces and moments, are described as follows:

$$S_z = \rho g \pi r^2 \tag{3.18}$$

$$S_\theta = \frac{\rho g \pi r^4}{4} \tag{3.19}$$

$$F_z = -S_z z \tag{3.20}$$

$$M_\theta = -S_\theta \theta. \tag{3.21}$$

The actual vertical hydrostatic stiffness S_z and rotational S_θ are taken from OrcaWave diffraction model.

3.4.3 Simplification of the Force Vector

A real wind turbine has an operating rotor, subject to aerodynamic thrust forces, within a turbulent wind field, mounted at the top of a large tower, which provides a large overturning moment. During storm-conditions, with wind speeds > 25 m/s, the turbine will be shut down. Thus we can simplify the force contribution from the rotor and nacelle. For describing the main forces influencing the nacelle and rotor, a scenario in which the only acting force is the force of gravity is concerned. It implies that the case with no wind and no rotation is applied.

Still, even with no force and rotation, the weight of the nacelle might be the reason of bending or torsional moments. But, since it is an inertial force it provides resistance to motion. However, the moment that acts on the nacelle due to gravity is zero, since the is no arm between the axis of the tower and center of gravity of the nacelle, for the system analysed in this project. Otherwise it would be calculated with Eqn. 3.22.

$$M_{\text{nacelle}} = F_g\, d \tag{3.22}$$

Where F_g is the gravitational force and d is the horizontal distance from the axis of the tower to the center of gravity of the nacelle. By this simplification the force vector can be constructed, including simplified model of wind load that act on nacelle and rotor, see Eqn. 6.20.

The foundation for more comprehensive studies that will incorporate variable wind forces at a later stage is laid when no-wind conditions are thoroughly understood. This way the necessity of dividing the analysis into smaller parts as part of a systematized process for solving problems is emphasized.

4 The Moving Frame Method

In contrast to traditional vector-based dynamics, MFM adopts moving reference frame approach of Élie Cartan [25] along with the group theory and associated algebra of Sophus Lie, allowing to relate frames of a multi-body system. This method is further developed by Frankel [26], Murakami [27], and described by Impelluso [28].

4.1 Main Principles of MFM

MFM is based on principles of Lie Group, and more specifically on Special Euclidean Group $SE(3)$ and the Special Orthogonal Group $SO(3)$ which enables representation of 3D rotations through matrices, by means of associated Lie algebra $\mathfrak{so}(3)$ and body kinematics.

4.2 Body Kinematics

It can be assumed that any arbitrary point can be written without explicitly stating it, as described in Eqn. 4.1

$$\mathbf{e}' = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right) \tag{4.1}$$

Then, given that every coordinate in Eqn. 4.1 is normalized, a vector basis e^I is derived:

$$\mathbf{e}^I = (\mathbf{e}_1^I, \mathbf{e}_2^I, \mathbf{e}_3^I) \tag{4.2}$$

Eqn. 4.2 defines an inertial frame, while each individual α-body frame has it's own vector basis, Eqn. 4.3, attached to the mass center $C^{(\alpha)}$ of the respective body.

$$\mathbf{e}^{(\alpha)}(\mathrm{t}) = \left(\mathbf{e}_1^{(\alpha)}(\mathrm{t}) \mathbf{e}_2^{(\alpha)}(\mathrm{t}) \mathbf{e}_3^{(\alpha)}(\mathrm{t}) \right) \tag{4.3}$$

The body numbers are then sequentially assigned, starting with $\alpha = 1$ for the platform and moving inwards towards the nacelle. The general schematic for FOWT is shown in Figure 4.1

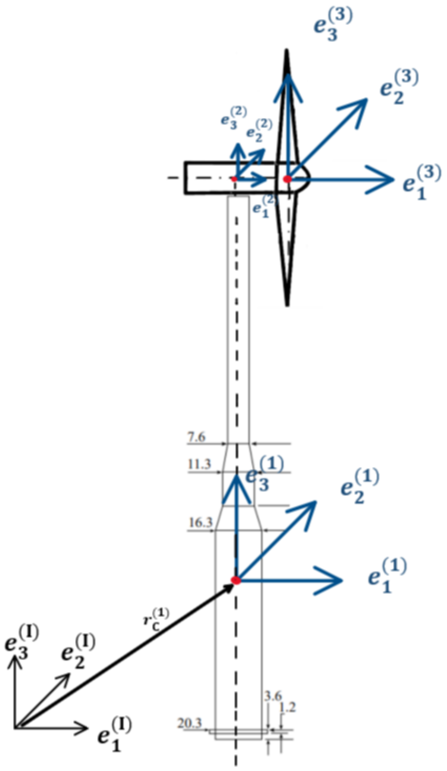


Figure 4.1: Diagram of FOWT showing position of the frames usied in this study, adapted from [1].

The translation of body-α in reference to an inertial frame perspective is denoted as $\mathrm{r}_{\mathrm{C}}^{(\alpha)}(\mathrm{t})$, where the symbol x is used to represent inertial coordinates.

$$\mathrm{r}_{\mathrm{C}}^{(a)}(\mathrm{t}) = \mathbf{e}^{\mathrm{I}}\mathrm{x}_{\mathrm{C}}^{(\alpha)}(\mathrm{t}) \tag{4.4}$$

The attitude, or orientation and rotation, of $e^{\alpha}(t)$ is measured from e^{I} and represented by a 3×3 rotation matrix, $R^{\alpha}(t)$, a member of the $SO(3)$ group, as shown in equation (4.5)

$$e^{\alpha}(t) = e^{I}(t)R^{\alpha}. \tag{4.5}$$

Equation (4.6) captures the orientation of body-$(\alpha + 1)$ with respect to body-α, as described in equation (4.5).

$$e^{(\alpha+1)}(t) = e^{(\alpha)}(t)R^{(\alpha+1/\alpha)}(t). \tag{4.6}$$

If the orientation of the body $e^{(\alpha+1)}(t)$ is desired with respect to the inertial frame, Eqn. (4.5) can be inserted into Eqn. (4.6) as follows:

$$e^{(\alpha+1)}(t) = e^{I}R^{\alpha}(t)R^{(\alpha+1/\alpha)}(t) = e^{I}R^{(\alpha+1)}(t). \tag{4.7}$$

The frames are connected through rotation matrices, therefore the inverse of Eqs. (4.5-4.7) exists. This implies that

$$R^{-1}(t) = R^{T}(t) \tag{4.8}$$

where superscript, T, represent s the square-matrix transpose.

4.2.1 Special Orthogonal Group $SO(3)$

$SO(3)$ comes with an associated algebra, denoted as $\mathfrak{so}(3)$, that produces the angular velocity matrix, as shown in Eqs. 4.11-4.12 and can state any skew-symmetric matrix.

By means of $SO3$ and $\mathfrak{so}(3)$ Eqn. 4.8 becomes identical with Eqn. 4.10

$$\mathrm{SO}(3) = \left\{R \in \mathbb{R}^{3\times 3} \mid R^T R = I, \det(R) = 1\right\} \tag{4.9}$$

Where det is determinant of matrix, R.

The vertical line in Eqn. 4.10 means that the inverse matrix A is equal to - A^*

$$\mathfrak{so}(3) = \left\{A \in \mathbb{R}^{3\times 3} \mid A^T = -A\right\} \tag{4.10}$$

Absolute from inertial frame
$$\overleftrightarrow{\omega(t)^{\alpha}} = \left(R(t)^{\alpha}\right)^T \dot{R}(t)^{\alpha} \tag{4.11}$$

Relative from abs.frame
$$\overleftrightarrow{\omega(t)^{(\alpha/\alpha-1)}} = \left(R(t)^{(\alpha/\alpha-1)}\right)^T \dot{R}(t)^{(\alpha/\alpha-1)} \tag{4.12}$$

The recursive relation to previous frame, starting from frame two and up, as shown below:

$$\overleftrightarrow{\omega^{(\alpha)}} = \left(R^{(\alpha/\alpha-1)}\right)^T \overleftrightarrow{\omega^{(\alpha-1)}} R^{(\alpha/\alpha-1)} + \left(R^{(\alpha/\alpha-1)}\right)^T \dot{R}^{(\alpha/\alpha-1)} \tag{4.13}$$

Knowing that:

$$\left(R^{(\alpha/\alpha-1)}\right)^T \overleftrightarrow{\omega^{(\alpha)}} R^{(\alpha/\alpha-1)} = \overleftrightarrow{\left(R^{(\alpha/\alpha-1)}\right)^T} \omega^{(\alpha)} \tag{4.14}$$

Thus, using Eqn. 4.12, absolute frame can be rewritten as:

$$\overleftrightarrow{\omega^{(\alpha)}} = \left(R^{(\alpha/\alpha-1)}\right)^T \overleftrightarrow{\omega^{(\alpha)}} R^{(\alpha/\alpha-1)} + \overleftrightarrow{\omega^{(\alpha/\alpha-1)}} \tag{4.15}$$

Or

$$\overleftrightarrow{\omega^{(\alpha)}} = \overleftrightarrow{\left(R^{(\alpha/\alpha-1)}\right)^T} \omega^{(\alpha)} + \overleftrightarrow{\omega^{(\alpha/\alpha-1)}} \tag{4.16}$$

Cancelling out the skew and get:

$$\omega^{(\alpha)} = \left(R^{(\alpha/\alpha-1)}\right)^T \omega^{(\alpha)} + \omega^{(\alpha/\alpha-1)} \tag{4.17}$$

Two complimentary algebraic expressions that are intertwined are give as follows:

- Reverse cross- product by installing negative sign:

$$\overleftrightarrow{\omega} s = \omega(-\overleftrightarrow{s}) = -\omega\overleftrightarrow{s} \tag{4.18}$$

- Absorb negative sign by transpose:

$$-\omega\overleftrightarrow{(s)} = \omega\overleftrightarrow{s}^T \tag{4.19}$$

4.2.2 Special Euclidean Group $SE(3)$

Now that SO (3) has been presented, the properties of the special Euclidean group, SE(3), which are used for defining the dynamics of the platform, are presented in this section. There are two types of $SE(3)$ Frame Connection Matrices, i.e. absolute and relative. The absolute frame connection matrix, expressed with Eq.4.21, connects the moving frame to the inertial frame. While the relative frame defines the relation between two moving frames, see Eqn: 4.23.

$$(e^{(\alpha)}(t)\; x_C^{(\alpha)}(t)) = (e^I\; 0)E^{(\alpha)}(t) \tag{4.20}$$

$$E^{(\alpha)}(t) = \begin{bmatrix} R^{(\alpha)}(t)\, x_C^{(\alpha)}(t) \\ 0_{1\times 3} \quad\quad 1 \end{bmatrix} \tag{4.21}$$

$$(e^{(\alpha+1/\alpha)}(t)\; r_C^{(\alpha+1/\alpha)}(t)) = (e^{(\alpha)}(t)\; r_C^{(\alpha)}(t))E^{(\alpha+1/\alpha)}(t) \tag{4.22}$$

$$E^{(\alpha+1/\alpha)}(t) = \begin{bmatrix} R^{(\alpha+1/\alpha)}(t)\, S_C^{(\alpha+1/\alpha)}(t) \\ 0_{1\times 3} \quad\quad\quad 1 \end{bmatrix} \tag{4.23}$$

Absolute frame has relation to inertial in comparison to SO(3):

$$SO(3):\quad \overleftrightarrow{\omega^{(\alpha)}} = \left(R^{(\alpha/\alpha-1)}\right)^T \overleftrightarrow{\omega^{(\alpha-1)}} R^{(\alpha/\alpha-1)} + \left(R^{(\alpha/\alpha-1)}\right)^T \dot{R}^{(\alpha/\alpha-1)} \tag{4.24}$$

$$SE(3):\quad \overleftrightarrow{\Omega^{(\alpha)}} = \left(E^{(\alpha/\alpha-1)}\right)^T \overleftrightarrow{\Omega^{(\alpha-1)}} E^{(\alpha/\alpha-1)} + \left(E^{(\alpha/\alpha-1)}\right)^T \dot{E}^{(\alpha/\alpha-1)} \tag{4.25}$$

4.2.3 Generalized Coordinates

Relation between *Cartesian coordinates* and *generalized* coordinates is expressed by Eqn. 4.26.

$$\left\{X_i\dot{(t)}(t)\right\} = \left[B_{ij}(t))\right]\left\{\dot{q}_j(t)\right\} \tag{4.26}$$

Where the Cartesian $\dot{\mathbf{X}}(t)$ is defined by linear velocity $\dot{x}_c^{(\alpha)}(t)$ and angular velocity $\omega^{(\alpha)}(t)$ at the center of mass of each body, forming $\alpha x1$ matrix for a system with α bodies and n- DOF as shown in Eqn. 4.27. While $\dot{\mathbf{q}}(t)$ is defined by Eqn. 4.28.

Cartesian ***Generalized***

$$\{\dot{\mathbf{X}}(t)\} = \begin{Bmatrix} \dot{x}_c^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}_c^{(2)}(t) \\ \omega^{(2)}(t) \\ \vdots \\ \dot{x}_c^{(\alpha)}(t) \\ \omega^{(\alpha)}(t) \end{Bmatrix} \tag{4.27}$$

$$\{\dot{\mathbf{q}}(t)\} = \begin{Bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \vdots \\ \dot{q}_n(t) \end{Bmatrix} \tag{4.28}$$

4.3 Dynamics of floating Offshore Wind Turbine

In this section the dynamics of FOWT is described in detail. The theory given in previous chapter is now applied to a model of FOWT, represented in Figure 4.1.

4.4 First Frame: Platform/Tower

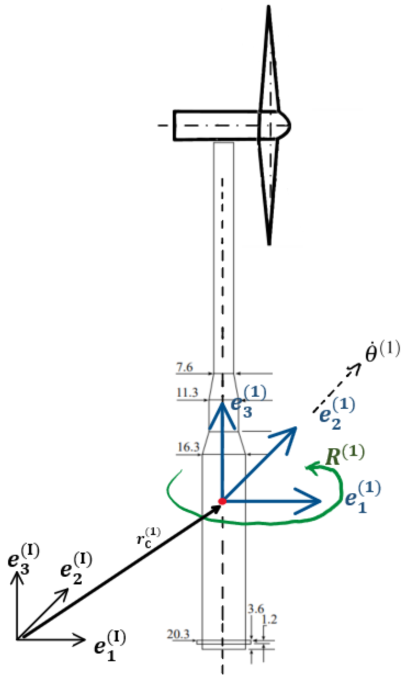


Figure 4.2: Connection between inertial frame and frame 1. [1]

First frame related to inertial frame by Eqn. 4.29:

$$\mathbf{e}^{(1)}(t) = \mathbf{e}^I R^{(1)}(t) \tag{4.29}$$

Position vector $\mathbf{r}_c^{(1)}(t)$ is defined by Eqn. 4.30:

$$\mathbf{r}_c^{(1)}(t) = \mathbf{e}^I x_c^{(1)}(t) \tag{4.30}$$

Frame connection matrix containing position and the rotation of the first frame is:

$$E^{(1)}(t) = \begin{bmatrix} R^{(1)}(t) & x^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \tag{4.31}$$

The rate of change is the derivative of the frame connection matrix

$$\dot{E}^{(1)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t) & \dot{x}_c^{(1)}(t) \\ 0^T & 0 \end{bmatrix} \tag{4.32}$$

The inverse of the frame connection matrix

$$\left(E^{(1)}(t)\right)^{-1} = \begin{bmatrix} R^{(1)T}(t) & -R^{(1)T}(t)x_c^{(1)}(t) \\ 0^T & 1 \end{bmatrix} \tag{4.33}$$

The absolute time rate of frame connection matrix for the first frame:

$$\Omega^{(1)} = \left(E^{(1)}(t)\right)^{-1} \dot{E}^{(1)}(t) = \begin{bmatrix} \overleftrightarrow{\omega^{(1)}(t)} & \left(R^{(1)}(t)\right)^T \dot{x}^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{4.34}$$

First term representing angular velocity for the first frame is extracted from the first column:

$$\dot{\mathbf{e}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)} \tag{4.35}$$

The second term is also extracted from the second column, representing linear velocity in terms of the first frame:

$$\dot{\mathbf{r}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\left(R^{(1)}(t)\right)^T \dot{x}^{(1)}(t) \tag{4.36}$$

Pulling back to inertial frame, by applying Eqn. 4.29 and simplifying with Eqn. 4.9, $\left(R^{(1)}(t)\right)^T R^{(1)}(t) = I$ and therefore term R^1 vanishes:

$$\dot{\mathbf{r}}^{(1)}(t) = \mathbf{e}^I \dot{x}^{(1)}(t) \tag{4.37}$$

4.5 Second Frame: Nacelle

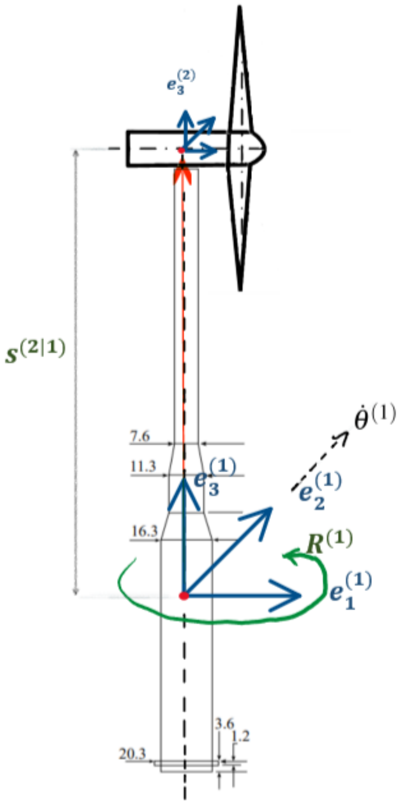


Figure 4.3: Connection between frame 1 and frame 2 through path $s^{(2/1)}$, follow the red arrow. [1]

The nacelle is considered to be in a fixed position, so rotation $R^{(2/1)}$ is eliminated. Therefore the relative frame connection is only defined by a translation from the mass center of the tower towards mass center of the nacelle, $s^{(2/1)}$, as shown in Figure 4.6.

$$E^{(2/1)}(t) = \begin{bmatrix} I_3 & s^{(2/1)} \\ 0_3^T & 1 \end{bmatrix} \tag{4.38}$$

The absolute frame connection matrix $E^{(2)}(t)$

$$E^{(2)}(t) = E^{(1)}(t)E^{(2/1)}(t) = \begin{bmatrix} R^{(1)}(t) & R^{(1)}(t)s^{(2/1)} + x^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \tag{4.39}$$

with the rate of change given as the time-derivative of the frame connection matrix, see equation below:

$$\dot{E}^{(2)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t) & \dot{R}^{(1)}(t)s^{(2/1)} + \dot{x}_c^{(1)}(t) \\ 0^T & 0 \end{bmatrix} \tag{4.40}$$

The inverse of the frame connection matrix is:

$$\left(E^{(2)}(t)\right)^{-1} = \begin{bmatrix} R^{(1)T}(t) & -R^{(1)T}(t)(R^{(1)}(t)s^{(2/1)} + x^{(1)}(t)) \\ 0^T & 1 \end{bmatrix} \tag{4.41}$$

and the absolute time rate of frame connection matrix for the second frame:

$$\Omega^{(2)}(t) = \begin{bmatrix} \left(R^{(1)}(t)\right)^T \dot{R}^{(1)}(t) & \left(R^{(1)}(t)\right)^T \left(\dot{R}^{(1)}(t)s^{(2/1)} + \dot{x}^{(1)}(t)\right) \\ 0_3^T & 0 \end{bmatrix} \tag{4.42}$$

The first term representing angular velocity for the second frame is extracted from the first column:

$$\omega^{(2)}(t) = \mathbf{e}^{(2)}(t)\left(\omega^{(1)}(t)\right) \tag{4.43}$$

By applying Eqn. 4.29 and simplifying with Eqn. 4.9,
the $\left(R^{(1)}(t)\right)^T R^{(1)}(t) = I$ vanishes and thus relation to the inertial frame is obtained as:

$$\dot{\mathbf{r}}^{(2)}(t) = e^I \left(R^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)}\, s^{(2/1)} + \dot{x}^{(1)}(t)\right) \tag{4.44}$$

By reversing cross product with Eqn. 4.18 and absorbing the negative sign by transpose as in Eqn. 4.19, final linear velocity can be expressed as:

$$\dot{\mathbf{r}}^{(2)}(t) = \mathbf{e}^I \left\{ R^{(1)}(t)\left(\overleftrightarrow{s^{(2/1)}}\right)^T \omega^{(1)}(t) + \dot{x}^{(1)}(t) \right\} \tag{4.45}$$

4.6 Third Frame: Rotor

To enable future incorporation of aerodynamic forcing, the equations are presented in this Section, assuming separate frames for each the nacelle and rotor. Those frames will be locked together as a single-rigid body in the final code. Following the same procedure developed for the first and second frame we obtain the third frame which is represented by RNA.

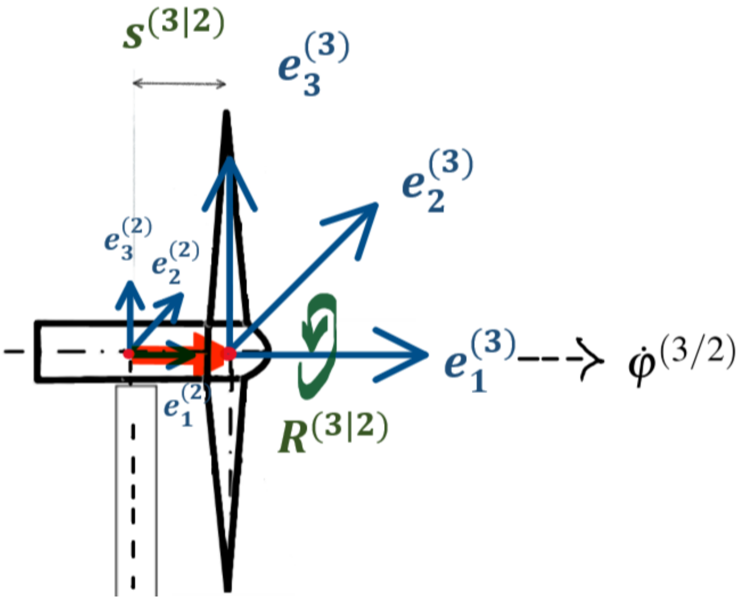


Figure 4.4: Connecting frame 2 with frame 3 through path $s^{(3/2)}$, follow the red arrow. [1]

$$\mathbf{e}^{(3)}(t) = \mathbf{e}^{(2)}(t)R^{(3/2)}(t) = \mathbf{e}^{(2)}(t)\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi^{(3/2)}(t) & -sin\ \varphi^{(3/2)}(t) \\ 0 & \sin\varphi^{(3/2)}(t) & \cos\varphi^{(3/2)}(t) \end{bmatrix} \tag{4.46}$$

$$E^{(3/2)}(t) = \begin{bmatrix} I_{3x3} & s_c^{(3/2)}(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^{(3/2)}(t) & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^{(3/2)}(t) & s^{(3/2)} \\ 0 & 1 \end{bmatrix} \tag{4.47}$$

$$E^{(3)}(t) = E^{(2)}(t)E^{(3/2)}(t) \tag{4.48}$$

$$E^{(3)}(t) = \begin{bmatrix} R^{(1)}(t) & R^{(1)}(t)s^{(2/1)} + x^{(1)}(t) \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} R^{(3/2)}(t) & s^{(3/2)} \\ 0 & 1 \end{bmatrix} \tag{4.49}$$

$$E^{(3)}(t) = \begin{bmatrix} R^{(1)}R^{(3/2)} & R^{(1)}s^{(3/2)} + R^{(1)}s^{(2/1)} + x^{(1)} \\ 0 & 1 \end{bmatrix} \tag{4.50}$$

$$\Omega^{(3)}(t) = \left(E^{(3)}(t)\right)^{-1} \dot{E}^{(3)}(t) = \begin{bmatrix} \Omega^{(3)}_{11}(t) & \Omega^{(3)}_{12}(t) \\ 0 & 0 \end{bmatrix} \tag{4.51}$$

$$\dot{E}^{(3)}(t) = \begin{bmatrix} \dot{R}^{(1)}R^{(3/2)} + R^{(1)}\dot{R}^{(3/2)} & \dot{R}^{(1)}s^{(3/2)} + \left\{\dot{R}^{(1)}s^{(2/1)} + \dot{x}^{(1)}\right\} \\ 0 & 0 \end{bmatrix} \tag{4.52}$$

Realizing that $\dot{x}^{(2)} = \dot{R}^{(1)}s^{(2/1)} + \dot{x}^{(1)}$, and therefore:

$$\dot{E}^{(3)}(t) = \begin{bmatrix} \dot{R}^{(1)}R^{(3/2)} + R^{(1)}\dot{R}^{(3/2)} & \dot{R}^{(1)}s^{(3/2)} + \dot{x}^{(2)} \\ 0 & 0 \end{bmatrix} \tag{4.53}$$

$$\Omega^{(2)}_{11}(t) = \left(R^{(1)}(t)\right)^T \left(R^{(3/2)}(t)\right)^T \left(\dot{R}^{(1)}(t)R^{(3/2)}(t) + R^{(1)}(t)\dot{R}^{(3/2)}(t)\right) \tag{4.54}$$

Acknowledging that $\overleftrightarrow{\omega^{(1)}(t)} = \left(R^{(1)}(t)\right)^T \dot{R}^{(1)}(t)$ according to Eqn. 4.11,
and $\overleftrightarrow{\omega^{(3/2)}(t)} = \left(R^{(3/2)}(t)\right)^T \dot{R}^{(3/2)}(t)$ by Eqn. 4.12. Therefore:

$$\Omega^{(2)}_{11}(t) = \left(R^{3/2)}\right)^T \overleftrightarrow{\omega^{(1)}} R^{(3/2)} + \overleftrightarrow{\omega^{(3/2)}} \tag{4.55}$$

Cancel out the skew by means of Eqn. 4.17, and extracting $\omega^{(3)}$:

$$\omega^{(3)} = \left(R^{3/2)}\right)^T \omega^{(1)} + \omega^{(3/2)} \tag{4.56}$$

$$\Omega^{(2)}_{12}(t) = \left(R^{(1)}(t)\right)^T \left(R^{(3/2)}(t)\right)^T \left(\dot{R}^{(1)}s^{(3/2)} + \dot{x}^{(2)}\right) \tag{4.57}$$

Simplifying expression by utilizing Eqn. 4.11 and multiplying Eqn. 4.57 with $R^{(3/2)}(t)$ to get rid of the transpose of this term, by identity matrix $\left(R^{(3/2)}(t)\right)^T R^{(3/2)}(t) = I$, getting:

$$\Omega^{(2)}_{12}(t) = \overleftrightarrow{\omega^{(1)}(t)}\, s^{(3/2)} + \left(R^{(1)}(t)\right)^T \dot{x}^{(2)}(t) \tag{4.58}$$

Reintroducing the inertial frame, by frame relation given in Eqns. 4.29 and 4.5

$$\dot{\mathbf{r}}^{(3)}(t) = e^I \left(R^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)}\, s^{(3/2)} + \dot{x}^{(2)}(t)\right) \tag{4.59}$$

By reversing cross- product with Eqn. 4.18 and absorbing the negative sign by transpose as in Eqn. 4.19, final linear velocity can be expressed as:

$$\dot{\mathbf{r}}^{(3)}(t) = e^{I}\left(R^{(1)}(t)\omega^{(1)}(t)\ \left(\overleftrightarrow{s^{(3/2)}}\right)^{T} + \dot{x}^{(2)}(t)\right) \tag{4.60}$$

The updated kinematics of the platform is shown in Figure 4.4 and Table 4.1.

Table 4.1: The kinematic results for frames 1-3, representing the floating offshore wind turbine in the MFM.

Body	Linear velocity	Angular velocity
1: Platform	$\dot{x}_c^{(1)}(t) = \dot{x}_c^{(1)}(t)$	$\omega^{(1)}(t) = \omega^{(1)}(t)$
2: Nacelle	$\dot{x}^{(2)}(t) = R^{(1)}(t)\left(\overleftrightarrow{s^{(2/1)}}\right)^{T}\ \omega^{(1)}(t) + \dot{x}^{(1)}(t)$	$\omega^{(2)}(t) = \omega^{(1)}(t)$
3: Rotor	$\dot{x}^{(3)}(t) = R^{(1)}(t)\ \left(\overleftrightarrow{s^{(3/2)}}\right)^{T}\ \omega^{(1)}(t) + \dot{x}^{(2)}(t)$	$\omega^{(3)}(t) = \left(R^{3/2)}\right)^{T}\ \omega^{(1)} + \omega^{(3/2)}$

4.7 Generalized Coordinates of FOWT

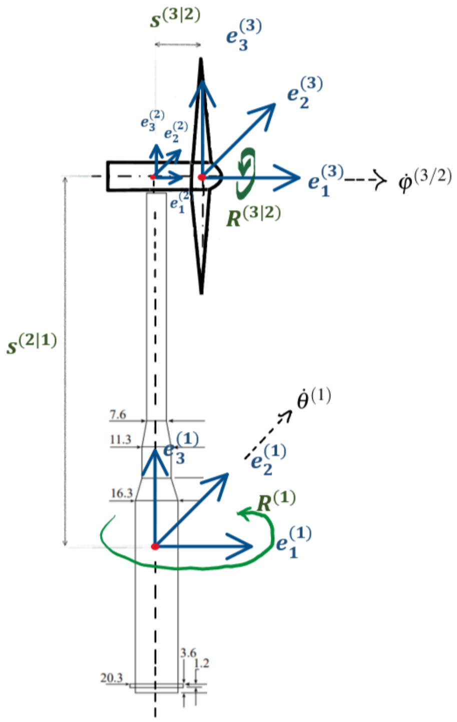


Figure 4.5: Model of FOWT with 4 general coordinates $\{\dot{\mathbf{q}}(t)\}$ enabled. [1]

To extract the minimal set of generalized coordinates $\{\dot{\mathbf{q}}(t)\}$ from the equations presented in Tabel 4.1, equations are redefined, using the respective unit vectors, according to the following assumptions:

- The platform is considered to only translate in directions 1 and 3. It simplifies the definition of the linear velocity for these bodies. Moving side to side is not a major concern of this study. Because the axisymmetric geometry and only considering unidirectonal waves, hence no transverse forcing are considered, though one should keep in mind, that this will ignore any cross-coupling terms, which will be minor.

$$\dot{x}^{(1)}(t) = \dot{x}^{(1)} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \dot{x}^{(1)} e_1 + \dot{x}^{(1)} e_3 \tag{4.61}$$

And the generalized coordinates for the platforms linear velocity are:

$$\dot{x}_1^{(1)}(t) = \dot{x}^{(1)}e_1 \tag{4.62}$$

$$\dot{x}_3^{(1)}(t) = \dot{x}^{(1)}e_3 \tag{4.63}$$

- The platform's angular velocity is limited to rotation in the 2 direction, as other directions are deemed to have negligible effects. Nacelle is assumed to be in fixed position at the top of the platform and therefore doesn't rotate.
  Therefore, $R^{(2)}(t) = R^{(1)}(t)R^{(2/1)}(t) = R^{(1)}(t)I = R^{(1)}(t)$.

$$\omega^{(1)}(t) = \dot{\theta}^{(1)} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \dot{\theta}^{(1)}e_2 \tag{4.64}$$

- The rotor rotates is set this to zero, assuming that the offshore turbine is in parked condition, though the facility is available in the code to set the rotational velocity to a time-varying value. Rotor and Nacelle are considered as a single body.

$$\omega^{(3/2)}(t) = \dot{\varphi}^{(3/2)} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \dot{\varphi}^{(3/2)}e_1 \tag{4.65}$$

The updated kinematic results are given below:

- Platform:

$$\dot{x}^{(1)}(t) = \dot{x}^{(1)}e_1 + \dot{x}^{(1)}e_3 \tag{4.66}$$

$$\omega^{(1)}(t) = \dot{\theta}e_2 \tag{4.67}$$

- Nacelle:

$$\dot{x}^{(2)}(t) = R^{(1)}(t)\left(\overleftrightarrow{s^{(2/1)}}\right)^T \underbrace{(\dot{\theta}e_2)}_{\omega^{(1)}(t)} + \underbrace{\dot{x}^{(1)}e_1 + \dot{x}^{(1)}e_3}_{\dot{x}^{(1)}(t)} \tag{4.68}$$

$$\omega^{(2)}(t) = \dot{\theta}e_2 \tag{4.69}$$

- Rotor:

$$\dot{x}^{(3)}(t) = R^{(1)}(t)\left(\overleftrightarrow{s^{(3/2)}}\right)^{T}\left(\dot{\theta}e_2\right) + \underbrace{R^{(1)}(t)\left(\overleftrightarrow{s^{(2/1)}}\right)^{T}\left(\dot{\theta}e_2\right) + \dot{x}^{(1)}e_1 + \dot{x}^{(1)}e_3}_{\dot{x}^{(2)}(t)} \quad (4.70)$$

$$\dot{x}^{(3)}(t) = \underbrace{R^{(1)}(t)\left\{\left(\overleftrightarrow{s^{(3/2)}}\right)^{T} + \left(\overleftrightarrow{s^{(2/1)}}\right)^{T}\right\}e_2}_{B_{53}}\ \dot{\theta}^{(1)} + \dot{x}^{(1)}e_1 + \dot{x}^{(1)}e_3\} \quad (4.71)$$

$$\omega^{(3)}(t) = \left(R^{(3/2)}\right)^{T}(\dot{\theta}e_2) + \dot{\varphi}^{(3/2)}e_1 \quad (4.72)$$

4.8 B -matrix of FOWT

The B-matrix has the same number of rows as Cartesian velocities $\left\{\dot{\mathbf{X}}(t)\right\}$ and the same number of columns as generalised coordinates $\{\dot{\mathbf{q}}(t)\}$. Therefore the Cartesian coordinate rates for three bodies, represented by velocity vector $\left\{\dot{\mathbf{X}}(t)\right\}$, contains a $6\alpha \times 1$ column vector of linear and angular velocities, where each element in the column serves as a 3 × 1 column for a different body.

4.8.1 B-matrix

Since the B matrix is dependent on the DOF, multi-body system and coordinates that matters, the entire Cartesian rate column for this model is 18×1 , while the full size of the B-matrix in this scenario is 18×4, as the number of generalised coordinates $\{\dot{\mathbf{q}}(t)\}$ is 4, see Eqn. 4.73 :

$$\{\dot{\mathbf{X}}(t)\} \equiv \left\{ \begin{array}{c} \dot{x}_1^{(1)}(t) \\ \dot{x}_2^{(1)}(t) \\ \dot{x}_3^{(1)}(t) \\ \omega_1^{(1)}(t) \\ \omega_2^{(1)}(t) \\ \omega_3^{(1)}(t) \\ \dot{x}_1^{(2)}(t) \\ \dot{x}_2^{(2)}(t) \\ \dot{x}_3^{(2)}(t) \\ \omega_1^{(2)}(t) \\ \omega_2^{(2)}(t) \\ \omega_3^{(2)}(t) \\ \dot{x}_1^{(3)}(t) \\ \dot{x}_2^{(3)}(t) \\ \dot{x}_3^{(3)}(t) \\ \omega_1^{(3)}(t) \\ \omega_2^{(3)}(t) \\ \omega_3^{(3)}(t) \end{array} \right\} \equiv \begin{pmatrix} \dot{x}^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}^{(2)}(t) \\ \omega^{(2)}(t) \\ \dot{x}^{(3)}(t) \\ \omega^{(3)}(t) \end{pmatrix} \qquad \{\dot{q}(t)\} \equiv \left\{ \begin{array}{c} \dot{x}_1^{(1)}(t) \\ \dot{x}_3^{(1)}(t) \\ \dot{\theta}^{(1)}(t) \\ \dot{\varphi}^{(3/2)}(t) \end{array} \right\} \tag{4.73}$$

Utilizing unit vector notation, Eqns. 4.66 -4.72 present the updated equations for the B matrix:

$$[B(t)] = \begin{bmatrix} e_1 & e_3 & \underset{3\times1}{0} & \underset{3\times1}{0} \\ \underset{3\times1}{0} & \underset{3\times1}{0} & e_2 & \underset{3\times1}{0} \\ e_1 & e_3 & R^{(1)}(t)\overleftrightarrow{s^{(2/1)}}^{T} e_2 & \underset{3\times1}{0} \\ \underset{3\times1}{0} & \underset{3\times1}{0} & e_2 & \\ e_1 & e_3 & B_{53} & \underset{3\times1}{0} \\ \underset{3\times1}{0} & \underset{3\times1}{0} & \left(R^{(3/2)}(t)\right)^{T} e_2 & e_1 \end{bmatrix} \tag{4.74}$$

Here

$$B_{53} = R^{(1)}(t)\left\{ \left(\overleftrightarrow{s^{(3/2)}}\right)^{T} + \left(\overleftrightarrow{s^{(2/1)}}\right)^{T} \right\} e_2 \tag{4.75}$$

4.8.2 Updated B -matrix

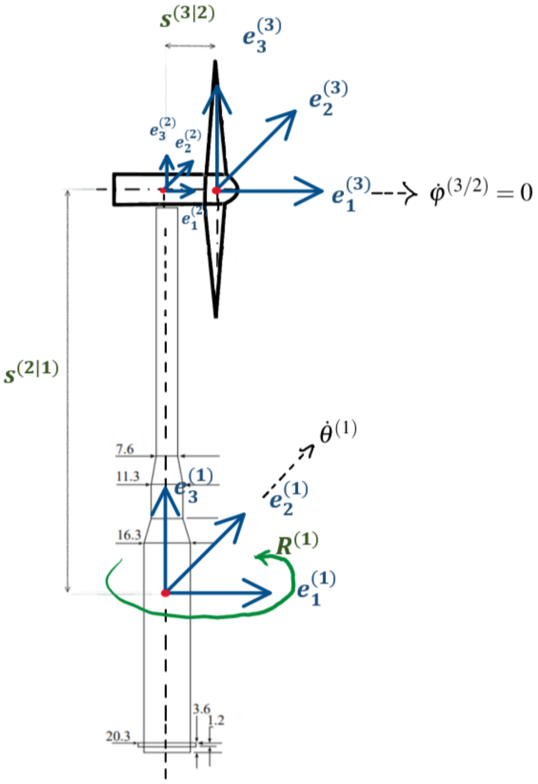


Figure 4.6: Updated model of FOWT with excluded spin rate $\dot{\varphi}^{(3/2)}$. [1]

The Cartesian velocity and generalized essential velocity are connected using the B-matrix presented in Section 4.8 which enables for the spin of the rotor, through rotation matrix $R^{(3/2)}(t)$ and rotation rate $\dot{\varphi}^{(3/2)}$. To simulate free decay or extreme wind conditions with wind speed above the turbine's cut-out speed, it may be necessary to remove the $\dot{\varphi}^{(3/2)}$ from the generalised velocities and the corresponding column of the B-matrix which links the generalised velocity to Cartesian rates. In this scenario, when the rotor does not spin, the pitch rate assumed to have prescribed constant spin rate $\dot{\varphi}^{(3/2)} = 0$; this implies that $R^{(3/2)}(t) = I$. The forces from applied unidirectional waves allows to proceed the analysis without considering sway, roll and yaw. Those can be easily added for generality, though, they are not exert the greatest load on the system. Therefore the following are the new generalized coordinates:

$$\{\dot{q}(t)\} \equiv \left\{ \begin{array}{c} \dot{x}_1^{(1)}(t) \\ \dot{x}_3^{(1)}(t) \\ \dot{\theta}^{(1)}(t) \end{array} \right\} \tag{4.76}$$

The full size of the B-matrix in this scenario is 18 × 3, as the number of generalised coordinates $\{\dot{\mathbf{q}}(t)\}$ is reduced to 3, as updated in Eqn. 4.76. The angular velocity $\omega^{(3)}(t)$, given in Eqn. 4.72, in this scenario is equal to $\omega^{(1)}(t)$, see Eqn. 4.77, while term B_{63} is represented by a unit vector, as shown in updated B-matrix, Eqn. 4.78:

$$\omega^{(3)}(t) = \dot{\theta} e_2 \tag{4.77}$$

$$[B(t)] = \begin{bmatrix} e_1 & e_3 & \underset{3\times1}{0} \\ \underset{3\times1}{0} & \underset{3\times1}{0} & e_2 \\ e_1 & e_3 & R^{(1)}(t)\overleftrightarrow{s^{(2/1)}}^{T} e_2 \\ \underset{3\times1}{0} & \underset{3\times1}{0} & e_2 \\ e_1 & e_3 & B_{53} \\ \underset{3\times1}{0} & \underset{3\times1}{0} & e_2 \end{bmatrix} \tag{4.78}$$

It is clear, that in this scenario, FOWT rotates at same angular speed, and $\omega^{(1)} = \omega^{(2)} = \omega^{(3)}$, because rows 2,4 and 6 in Eqn. 4.78 that represents $\omega(t)$, are absolutely identical.

5 Mooring model

The mooring system is represented by a unique formation of two different substructures: one is represented by a catenary quasi-static model, and another by a dynamic MFM based model. Complementing each other, like spirit and body, the initial catenary quasi-static model, brings the dynamic model to life, by analogy to the myth of Pygmalion's lifeless statue poised in readiness, getting a soul, and becoming alive.

Every subsystem of the mooring system is aimed to solve a particular task, and structured in a way that allows to redefine each model independently, while guaranteeing their seamless cooperation in establishing of overall mooring line system behaviour.

- **Quasi-Static Catenary Model :** The quasi static method solves for tension of each segment, allowing establishing the initial position of each segment, with an assumption that the system is in equilibrium state. Example of solution for 10 and 100 segments is shown in Figure 5.4. Since inertia or dampening effects do not significantly affect overall system dynamics as is evident in the comparative studies done earlier, they are not included into analysis.

- **Dynamic, MFM based Model :** Dynamic model is defined by the geometry of the initial quasi-static models. It converts Cartesian coordinates into generalized, by reading the latter and implementing them directly into B matrix. Then Virtual Work as an extension of Hamilton's principle is calculated, thus the force acting on fairlead is defined.

The duality of the catenary mooring system is reflected through theoretical framework described in subsequent sections, and incorporated into the architecture of the MATLAB code.

5.1 Quasi-static model

Development of the multi-segmented MFM based mooring line model is a tricky task. After many trials and fails, the most suitable for MFM -based mooring line model, have been established by adopting and modifying the mooring system module proposed by Jonkman, J. M. in Dynamics Modelling and Loads Analysis of an Offshore Floating Wind Turbine [3]. While quasi-static model is more complex compared to traditional idealized catenary line described with well-known catenary equation $y = a\cosh(\frac{x}{a}) - a$, it is more suitable for dynamic simulation, since it extends basic catenary theory, allowing inclusion of dynamic effects.

If catenary curve depends on hyperbolic functions, there should exist an equivalent representation of them. The inverse of hyperbolic sine is such an equivalent representation, given in Eqn. 5.1, allowing to express hyperbolic sine analytically, in different form. Therefore two non-linear equations, representing x and y position of the catenary mooring line are obtained by Eqns. 5.5, 5.6 at fairlead, and for suspended line by 5.7, 5.8 [3].

$$\sinh^{-1}(x) = \ln\left(x + \sqrt{1+x^2}\right) \tag{5.1}$$

Initial horizontal fairlead position serves as an input for establishing an initial guess, H_F^o and V_F^0, based on starting values given by Peyrot and Goulois, integrated in Newton-Raphson iteration test with Eqns. 5.2,5.3 and 5.4. This test solves for horizontal and vertical components of effective tension at fairlead T_{eff} through partial derivatives organized in Jacobian matrix [29]. Whole computational process is represented in the diagram, Figure 5.3.

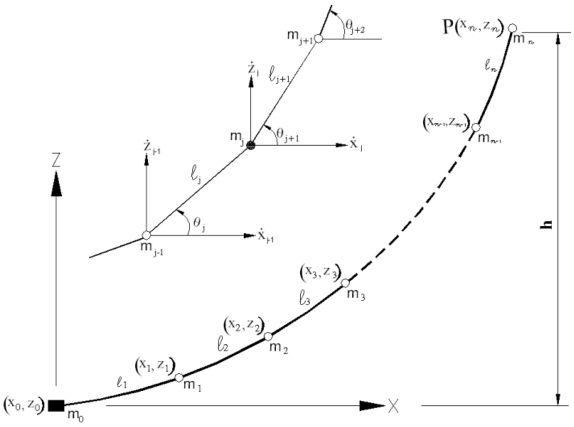


Figure 5.1: Mathematical model of an n-segments mooring line, constructed by means of hyperbolic functions. [2]

$$H_F^0 = \left| \frac{\omega x_F}{2\lambda_o} \right| \tag{5.2}$$

$$V_F^0 = \frac{\omega}{2} \left[\frac{z_F}{\tanh(\lambda_0)} + L \right] \tag{5.3}$$

Here λ_0 is dimensionless parameter, depending on conditions, used to initialize the catenary line:

$$\lambda_0 = \begin{cases} 1,000,000 & \text{for } x_F = 0 \\ 0.2 & \text{for } \sqrt{x_F^2 + z_F^2} \geq L. \\ \sqrt{3\left(\frac{L^2 - z_F^2}{x_F^2} - 1\right)} & \text{otherwise} \end{cases} \tag{5.4}$$

The initial catenary shape is set by the tension calculated at fairlead via Eqns.5.5 and 5.6, as seen from the diagram in Figure 5.3.

$$x_F(H_F, V_F) = \frac{H_F}{\omega} \left\{ \ln\left[\frac{V_F}{H_F} + \sqrt{1 + \left(\frac{V_F}{H_F}\right)^2} \right] - \ln\left[\frac{V_F - \omega L}{H_F} + \sqrt{1 + \left(\frac{V_F - \omega L}{H_F}\right)^2} \right] \right\} + \frac{H_F L}{EA} \tag{5.5}$$

$$z_F(H_F, V_F) = \frac{H_F}{\omega} \left[\sqrt{1 + \left(\frac{V_F}{H_F}\right)^2} - \sqrt{1 + \left(\frac{V_F - \omega L}{H_F}\right)^2} \right] + \frac{1}{EA}\left(V_F L - \frac{\omega L^2}{2} \right) \tag{5.6}$$

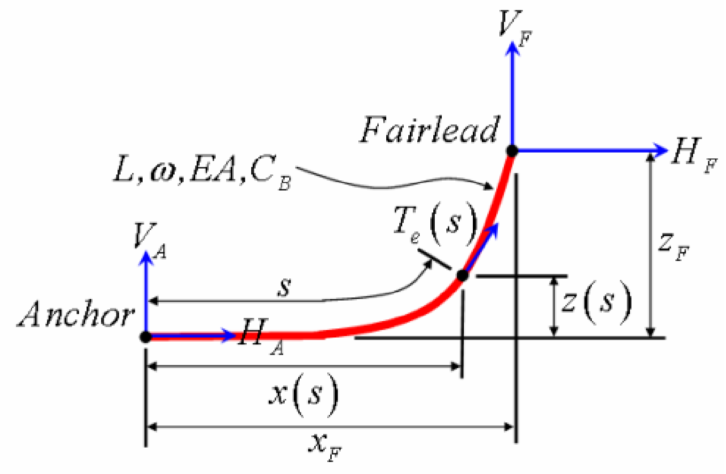


Figure 5.2: Generic catenary mooring line representation in local frame [3]

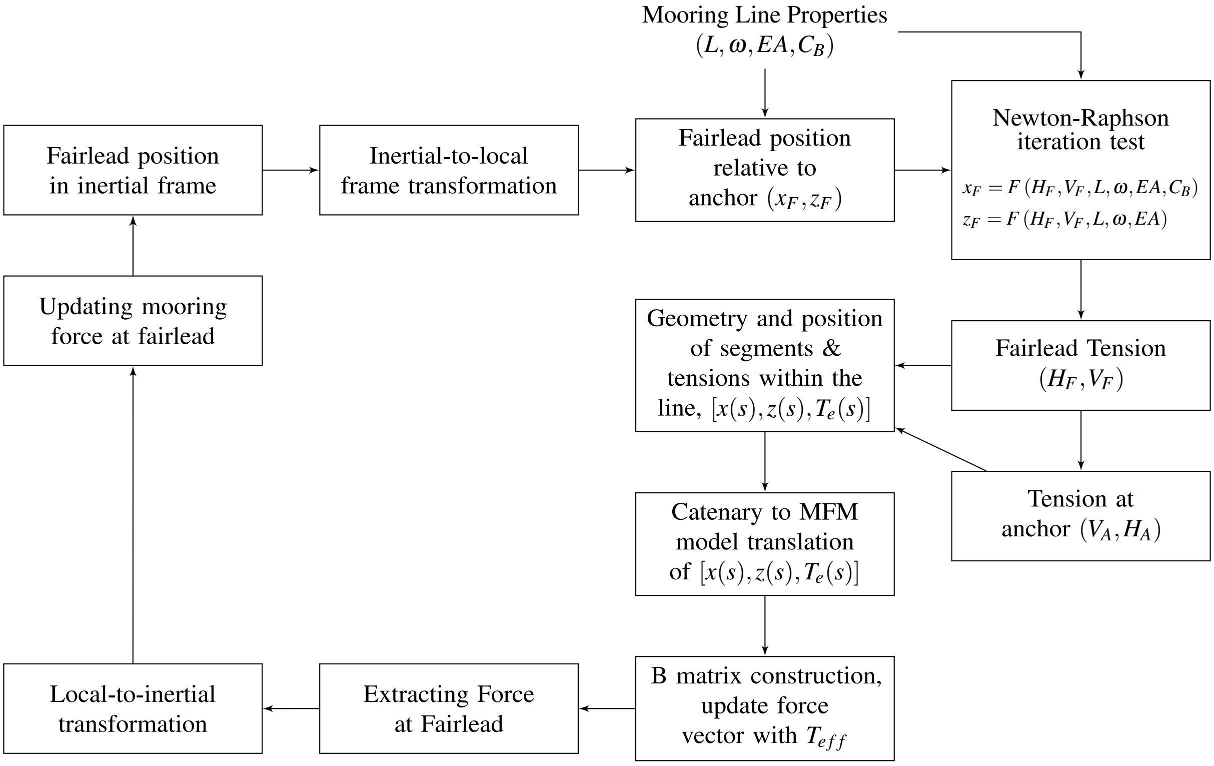

Mooring Line Properties
(L, ω, EA, C_B)
Fairlead position in inertial frame
Inertial-to-local frame transformation
Fairlead position relative to anchor (x_F, z_F)
Newton-Raphson iteration test
$x_F = F(H_F, V_F, L, \omega, EA, C_B)$
$z_F = F(H_F, V_F, L, \omega, EA)$
Updating mooring force at fairlead
Geometry and position of segments & tensions within the line, $[x(s), z(s), T_e(s)]$
Fairlead Tension (H_F, V_F)
Tension at anchor (V_A, H_A)
Catenary to MFM model translation of $[x(s), z(s), T_e(s)]$
Local-to-inertial transformation
Extracting Force at Fairlead
B matrix construction, update force vector with T_{eff}

Figure 5.3: Diagram of computational procedure.

When there is no interaction with a seabed, the position of the segments and effective tension T_e is calculated with Eqns. 5.7, 5.8 and 5.9 for every segment within the line, resulting in the suspended catenary shape of mooring line represented in Figure 5.4.

$$x(s) = \frac{H_F}{\omega}\left\{\ln\left[\frac{V_A + \omega s}{H_F} + \sqrt{1 + \left(\frac{V_A + \omega s}{H_F}\right)^2}\right] - \ln\left[\frac{V_A}{H_F} + \sqrt{1 + \left(\frac{V_A}{H_F}\right)^2}\right]\right\} + \frac{H_F s}{EA} \tag{5.7}$$

$$z(s) = \frac{H_F}{\omega}\left[\sqrt{1 + \left(\frac{V_A + \omega s}{H_F}\right)^2} - \sqrt{1 + \left(\frac{V_A}{H_F}\right)^2}\right] + \frac{l}{EA}\left(V_A s + \frac{\omega s^2}{2}\right) \tag{5.8}$$

$$T_e(s) = \sqrt{H_F^2 + (V_A + \omega s)^2} \tag{5.9}$$

Here s is unstretched distance from anchor to the α-segment.

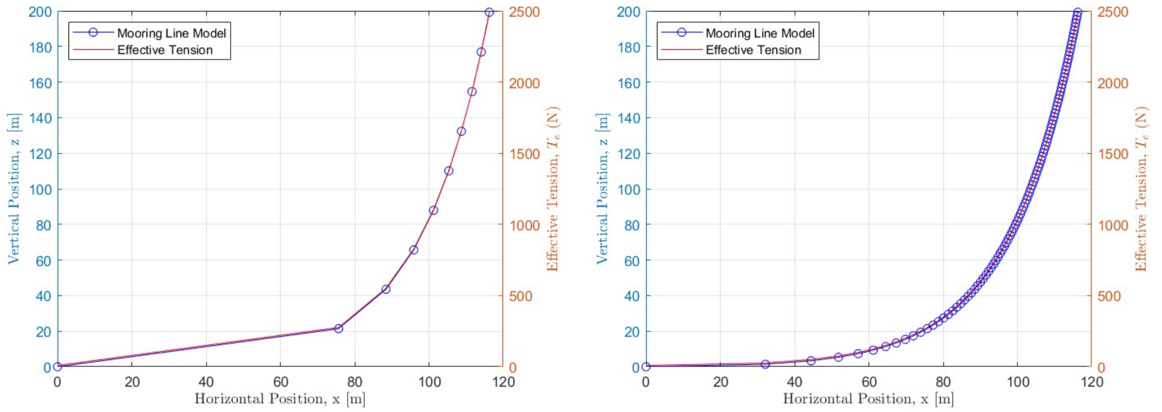


Figure 5.4: Effective tension distribution along the line and position of the 10 segments (left), and 100 segment (right).

A second set of equations, Eqns. 5.10, 5.11, and 5.12, is necessary to describe a situation, when a portion of the mooring line is lying on the seabed, as represented in Figure 5.5.

$$x(s) = \begin{cases} s & \text{for } 0 \leq s \leq L_B - \frac{H_F}{C_B\omega} \\ s + \frac{C_B\omega}{2EA}\left[\begin{array}{l} s^2 - 2\left(L_B - \frac{H_F}{C_B\omega}\right)s \\ +\left(L_B - \frac{H_F}{C_B\omega}\right)MAX\left(L_B - \frac{H_F}{C_B\omega},0\right)\end{array}\right] & \text{for } L_B - \frac{H_F}{C_B\omega} \leq s \leq L_B \\ L_B + \frac{H_F}{\omega}\ln\left[\frac{\omega(s-L_B)}{H_F} + \sqrt{1+\left(\frac{\omega(s-L_B)}{H_F}\right)^2}\right] + \frac{H_F s}{EA} & \text{for } L_B \leq s \leq L \\ +\frac{C_B\omega}{2EA}\left[-L_B^2 + \left(L_B - \frac{H_F}{C_B\omega}\right)MAX\left(L_B - \frac{H_F}{C_B\omega},0\right)\right] & \end{cases}, \tag{5.10}$$

$$z(s) = \begin{cases} 0 & \text{for } 0 \leq s \leq L_B \\ \frac{H_F}{\omega}\ln\left[\sqrt{1+\left(\frac{\omega(s-L_B)}{H_F}\right)^2} - 1\right] + \frac{\omega(s-L_B)^2}{2EA} & \text{for } L_B \leq s \leq L \end{cases}, \tag{5.11}$$

$$T_e(s) = \begin{cases} MAX\left(H_F + C_B\omega\left(s - L_B\right),0\right) & \text{for } 0 \leq s \leq L_B \\ \sqrt{H_F^2 + \left(\omega\left(s - L_B\right)\right)^2} & \text{for } L_B \leq s \leq L \end{cases}. \tag{5.12}$$

Utilizing segmetns positon and adopting solution of catenary problem presented by W. Whewell already in 1830's, the relation between segments is obtained in a similar way [30].

If the tension at the anchor is $T_e^{\alpha}(s)$, and $T_e^{\alpha+1}(s)$ is the tension measured at a distance s away from the previous segment, and the line's angle to the horizontal is equal θ, then the equilibrium balance of force components acting on each segment of the line is:

$$T_e^{\alpha}(s)cos(\theta) = T_e^{(\alpha-1)}(s) \tag{5.13}$$

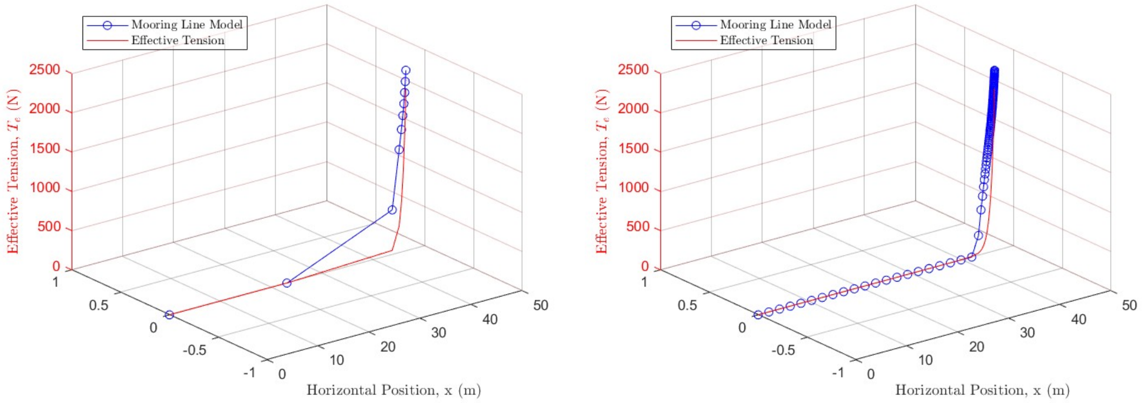


Figure 5.5: Position of the segments and shape of the catenary line, when the seabed interaction is taking into account 10 segments (left), and 100 segment (right, taking into account seabed interaction).

$$T_e^{\alpha}(s)sin(\theta) = \omega s^{(\alpha-1)} \tag{5.14}$$

Then from the line slope:

$$\tan\theta = \frac{\omega}{T_e^{(\alpha-1)}(s)} s = \frac{L}{a} \tag{5.15}$$

where a is a constant defined by relation $a = \frac{T_e^{(\alpha-1)}(s)}{\omega}$ and L is the total length of the line. Now, when angle θ is defined in terms of distance between s segments, the slope can be expressed by Eqn. 5.16, see Figure 5.2

$$\frac{dz}{dx} = \tan\theta = \frac{s}{a} \tag{5.16}$$

Utilizing the latter relation, the angles, in the form of the rate of change between adjacent segments along the line, are calculated by taking the derivatives of Eqns. 5.7, 5.7 with respect to s, as dx/ds and dz/ds. From that the angular velocity is derived as $\omega(s) = \frac{d^2z/ds^2}{d^2x/ds^2} = \frac{d}{ds}\theta(s)$, and angular acceleration is obtained, since $\dot{\omega}(s) = \frac{d}{ds}\omega(s)$. These mathematical relations are used to initialize the MFM-based model of catenary mooring line represented in Figure 5.2 and equations given in section 5.2.

5.2 MFM-based Catenary Model

Now, when the quasi-static catenary model, it's shape and geometry is defined, the linear and angular velocities for all α-segments can be obtained by establishing frames for each segment and then constructing B-matrix grouped in a $6n \times I$ matrix $\{\dot{X}(t)\}$, as described further in following sections of the chapter. The derived equations are based on the principles of Moving Frame Method, based on theory of E. Cartan, as instructed by T. J. Impelusso in Advanced Dynamics course [31].

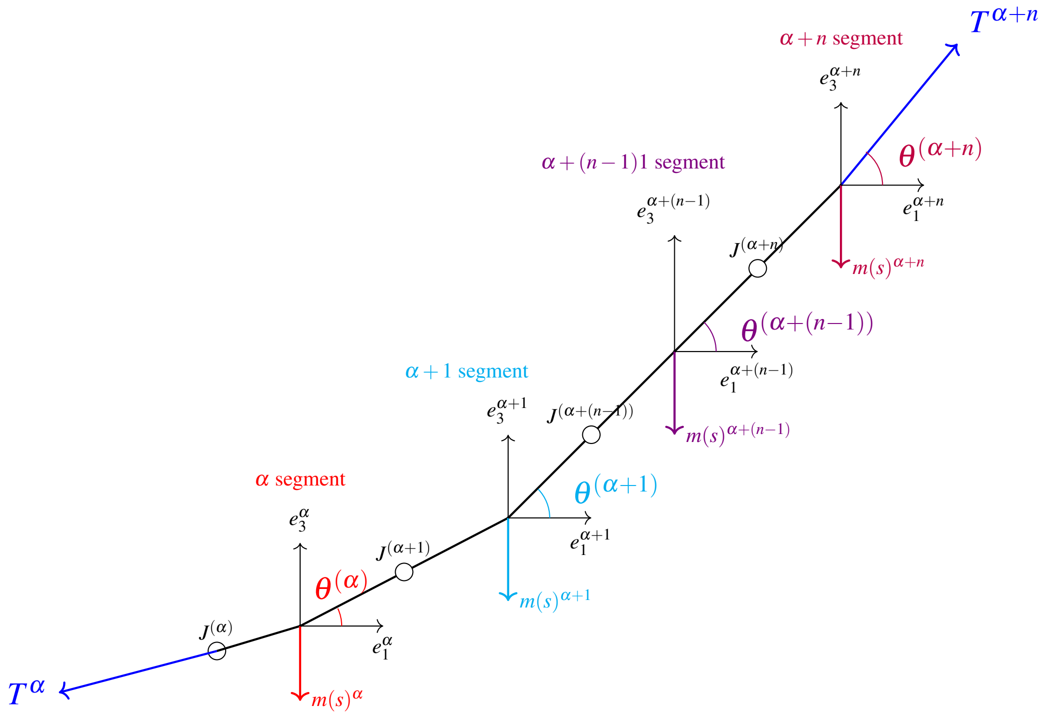


Figure 5.2: MFM-based multi-segmented model of catenary mooring line for α-segments.

Figure 5.2 shows the developed model where absolute frames e^{α} are installed at the center of mass of each α- segment between the connection points $J^{(\alpha)}$, which perform a joint function. Frames positions and each segment's length are defined by Eqs. 5.7 and 5.8, respectively, while angle $\theta^{(\alpha)}$, and tension T^{α} with Eqns. 5.15, 5.16, and 5.9, based on quasi-static model for mooring line presented in Section 5.1. Segments mass denoted as $m(s)^{(\alpha)}$ is determined by the choice of mooring lines profile. In this project High-Modulus Polyethylene (HMPE) has been applied to the model derived from the required minimum breaking load (MBL) and minimum breaking strength (MBS), in accordance to DNV standards [32], [33], [34], MEG4 standard [35], and ISO standard [36],

5.2.1 Frame of Segment 1

An inertial frame e^I is placed at the center of the first segment, which is related to moving frame, $\mathbf{e}^{(1)}(t)$, by the rotation matrix $R^{(1)}(t)$, establishing the orientation given by the theory presented in the Section 5 by equation Eqn. 4.5.

$$\mathbf{e}^{(1)}(t) = \mathbf{e}^I R^{(1)}(t) = \mathbf{e}^I(t) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta^{(1)}(t) & -\sin\theta^{(1)}(t) \\ 0 & \sin\theta^{(1)}(t) & \cos\theta^{(1)}(t) \end{bmatrix} \tag{5.17}$$

Fame connection matrix $E^{(1)}$ for the first segment is obtained from the sequence of movements from the inertial frame towards the center of mass of the first frame of the segment 1:

1. Translate by a distance $s^{(J^{(1)})}$
2. Rotate about axis of the inertial frame
3. Tranlsate by a distance $s^{(1/J^{(1)})}$

Expressing this in terms of frame connection matrices:

$$E^{(1)}(t) = \begin{bmatrix} I_{3\times 3} & s^{(J^{(1)})} \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} R^{(1)}(t) & 0 \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} I_{3\times 3} & s^{(1/J^{(1)})} \\ 0_3^T & 1 \end{bmatrix} \tag{5.18}$$

$$E^{(1)}(t) = \begin{bmatrix} R^{(1)}(t) & R^{(1)}(t) s^{(1/J^{(1)})} + s^{(J^{(1)})} \\ 0_3^T & 1 \end{bmatrix} \tag{5.19}$$

Stating first frame connnection matrix more compact as:

$$\left(\mathbf{e}^{(1)}(t)\mathbf{r}_C^{(1)}(t)\right) = \begin{pmatrix} \mathbf{e}^I & 0 \end{pmatrix} E^{(1)}(t) \tag{5.20}$$

Taking time rate of the first segment frame connection:

$$\begin{pmatrix} \dot{\mathbf{e}}^{(1)}(t) & \dot{\mathbf{r}}_C^{(1)}(t) \end{pmatrix} = \begin{pmatrix} \mathrm{e}^I & 0 \end{pmatrix} \dot{E}^{(1)}(t) \tag{5.21}$$

$$\dot{E}^{(1)}(t) = \begin{bmatrix} \dot{R}^{(1)}(t) & \dot{R}^{(1)}(t) s^{(1/J^{(1)})} \\ 0_3^T & 0 \end{bmatrix} \tag{5.22}$$

The representation of the inverse of the frame connection matrix is given by:

$$\left(E^{(1)}(t)\right)^{-1} = \begin{bmatrix} \left(R^{(1)}(t)\right)^T & -\left(R^{(1)}(t)\right)^T \left(R^{(1)}(t)s^{(1/J^{(1)})} + s^{(J^{(1)})}\right) \\ 0_3^T & 1 \end{bmatrix} \tag{5.23}$$

Following the definition of the Ω, which belongs to the special Euclidean group $SE(3)$ and its associated algebra se(3), the time rate at which the frame connection matrix changes is represented by $\Omega^{(1)}$. It is found by multiplying the inverse of $E^{(1)}(t)$ and $\dot{E}^{(1)}(t)$.

$$\Omega^{(1)} \equiv \left(E^{(1)}(t)\right)^{-1} \dot{E}^{(1)}(t) \tag{5.24}$$

$$\begin{pmatrix} \dot{\mathbf{e}}^{(1)}(t) & \dot{\mathbf{r}}_C^{(1)}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{e}^{(1)}(t) & \mathbf{r}_C^{(1)}(t) \end{pmatrix} \Omega^{(1)}(t) \tag{5.25}$$

$$\Omega^{(1)} = \begin{bmatrix} \omega^{(1)}(t) & v^{(1)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{5.26}$$

Pulling out angular velocity $\omega^{(1)}(t)$ from Eqn: 5.26

$$\dot{\mathbf{e}}^{(1)}(t) = \mathbf{e}^{(1)}(t)\overleftrightarrow{\omega^{(1)}(t)} \tag{5.27}$$

The assosiated angular velocity vector is:

$$\omega^{(1)}(t) = \mathbf{e}^{(1)}(t) \begin{pmatrix} \dot{\theta}^{(1)}(t) \\ 0 \\ 0 \end{pmatrix} \tag{5.28}$$

And therefore, based on 5.26:

$$\dot{\mathbf{r}}_C^{(1)}(t) = \mathbf{e}^{(1)}(t) v_C^{(1)}(t) = \mathbf{e}^{(1)}(t) \left(\overleftrightarrow{\omega^{(1)}(t)} s^{(1/J^{(1)})}\right) \tag{5.29}$$

Following rules of MFM, by installing a negative sign before the position vector $s^{(1/(J^{(1)})}$, the skew over $\omega^{(1)}(t)$ can be reversed, and therefore the equation Eqn. 5.29 is restated relative to the inertial frame as:

$$\dot{\mathbf{r}}_C^{(1)}(t) = -\mathbf{e}^{(1)}(t) \left(\overleftrightarrow{s^{(1/J^{(1)})}} \omega^{(1)}(t)\right) = -\mathbf{e}^I R^{(1)}(t) \left(\overleftrightarrow{s^{(1/J^{(1)})}} \omega^{(1)}(t)\right). \tag{5.30}$$

Finally, linear velocity $\dot{x}^{(1)}(t)$ for segment 1 is extracted:

$$\dot{x}^{(1)}(t) = -R^{(1)}(t)\overleftrightarrow{s^{(1/J^{(1)})}}\omega^{(1)}(t) \tag{5.31}$$

5.2.2 Frame of Segment 2

The process for defining frame 2 of the segment 2 is similar to that of frame 1. First the orientation of the second frame is established by equation Eqn. 5.32.

$$\mathbf{e}^{(2)}(t) = \mathbf{e}^{(1)}(t)R^{(2/1)}(t) = \mathbf{e}^{(1)}(t)\begin{bmatrix} \cos\theta^{(2)}(t) & -\sin\theta^{(2)}(t) & 0 \\ \sin\theta^{(2)}(t) & \cos\theta^{(2)}(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.32}$$

To reach frame 2 form frame 1, a fame connection matrix $E^{(2/1)}$ is obtained from the sequence of movements relative to the first frame:

1. Translation by a distance $s^{(J^{(2)}/1)}$
2. Rotation about common 3-axis by $R^{(2/1)}(t)$
3. Tranlsation by a distance $s^{(2/J^{(2)})}$ to the center of segment 2

Second frame is defined as:

$$\left(\mathbf{e}^{(2)}(t) \quad \mathbf{r}_C^{(2)}(t)\right) = \left(\mathbf{e}^{(1)}(t) \quad \mathbf{r}_C^{(1)}(t)\right)E^{(2/1)}(t) \tag{5.33}$$

Expressing this movement sequence in terms of frame connection matrix $E^{(2/1)}$ results in Eqn. 5.34:

$$E^{(2/1)}(t) = \begin{bmatrix} R^{(2/1)}(t) & R^{(2/1)}(t)s^{\left(2/J^{(2)}\right)} + s^{\left(J^{(2)}/1\right)} \\ 0 & 1 \end{bmatrix} \tag{5.34}$$

Finding inverse of $E^{(2/1)}(t)$:

$$\left(E^{(2/1)}\right)^{-1} = \begin{bmatrix} \left(R^{(2/1)}\right)^T(t) & -\left(R^{(2/1)}\right)^T(t)\left(R^{(2/1)}(t)s^{(2/J^{(2)})} + s^{(J^{(2)}/1)}\right) \\ 0 & 1 \end{bmatrix} \tag{5.35}$$

Therefore the rate of change is expressed by Eqn. 5.36

$$\dot{E}^{(2/1)}(t) = \begin{bmatrix} \dot{R}^{(2/1)}(t) & \dot{R}^{(2/1)}(t)s^{(2/J^{(2)})} \\ 0_3^T & 0 \end{bmatrix} \tag{5.36}$$

Relative time rate is defined by $\Omega^{(2/1)}(t)$

$$\Omega^{(2/1)}(t) = \begin{bmatrix} \overleftrightarrow{\omega^{(2/1)}(t)} & \overleftrightarrow{\omega^{(2/1)}(t)}s^{(2/J^{(2)})} \\ 0_3^T & 0 \end{bmatrix} \tag{5.37}$$

By pulling out the angular velocity term from $\Omega^{(2/1)}(t)$ and lifting the skew, the vector is defined:

$$\omega^{(2/1)}(t) = \mathbf{e}^{(2)}(t) \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}^{(2)}(t) \end{pmatrix} \tag{5.38}$$

Using definition given in chapter 5 by equation Eqn. 4.25, the absolute time rate of the frame connection matrix $\Omega^{(2)}(t)$

$$\Omega^{(2)}(t) = \left(E^{(2/1)}\right)^{-1}(t)\Omega^{(1)}(t)E^{(2/1)}(t) + \Omega^{(2/1)}(t) \tag{5.39}$$

From Eqn. 5.39 and by applying rules of MFM listed in chapter 5, the linear $\dot{x}^{(2)}(t)$ and angular velocity $\omega^{(2)}(t)$ are defined as:

$$\omega^{(2)}(t) = \left(R^{(2/1)}(t)\right)^T \omega^{(1)}(t) + \omega^{(2/1)}(t) \tag{5.40}$$

$$\dot{x}^{(2)} = R^{(1)}\left(\overleftrightarrow{R^{(2/1)}s^{(2/J^{(2)})} + s^{(J^{(2)}/1)}}\right)^T \omega^{(1)} + R^{(2)}\left(\overleftrightarrow{s^{(2/J^{(2)})}}\right)^T \omega^{(2/1)} + \dot{x}^{(1)} \tag{5.41}$$

The linear $\dot{x}^{(3)}(t)$, $\dot{x}^{(4)}(t)$ and angular velocities $\omega^{(3)}(t)$ and $\omega^{(4)}(t)$ for segments 3 and 4 are obtained in a similar way, resulting in Eqn.

$$\dot{x}^{(3)} = R^{(2)}\left(\overleftrightarrow{R^{(3/2)}s^{(3/J^{(3)})} + s^{(J^{(3)}/2)}}\right)^T \omega^{(2)} + R^{(3)}\left(\overleftrightarrow{s^{(3/J^{(3)})}}\right)^T \omega^{(3/2)} + \dot{x}^{(2)} \tag{5.42}$$

$$\begin{gathered} \omega^{(3)} = \left(R^{(3/2)}\right)^T \left(R^{(2/1)}\right)^T \omega^{(1)} + \left(R^{(3/2)}\right)^T \omega^{(2/1)} + \omega^{(3/2)} \\ \omega^{(3)} = \left(R^{(3/1)}\right)^T \omega^{(1)} + \left(R^{(3/2)}\right)^T \omega^{(2/1)} + \omega^{(3/2)} \end{gathered} \tag{5.43}$$

$$\dot{x}^{(4)} = R^{(3)}\left(\overleftrightarrow{R^{(4/3)}s^{(4/J^{(4)})}+s^{(J^{(4)}/3)}}\right)^{T}\omega^{(3)}+R^{(4)}\left(\overleftrightarrow{s^{(4/J^{(4)})}}\right)^{T}\omega^{(4/3)}+\dot{x}^{(3)} \tag{5.44}$$

Equations are simplified further by putting similar terms terms, and assigning them an S-term name, as shown in Eqns. 5.45, 5.46, and 5.45.

$$S^{(2/1)} = \left(\overleftrightarrow{R^{(2/1)}s^{(2/J^{(2)})}+s^{(J^{(2)}/1)}}\right)^{T} \tag{5.45}$$

$$S^{(3/2)} = \left(\overleftrightarrow{R^{(3/2)}s^{(3/J^{(3)})}+s^{(J^{(3)}/2)}}\right)^{T} \tag{5.46}$$

$$S^{(4/3)} = \left(\overleftrightarrow{R^{(4/3)}s^{(4/J^{(4)})}+s^{(J^{(4)}/3)}}\right)^{T} \tag{5.47}$$

5.2.3 Frames of α- Segments

The same logic of establishing frames for α-segments, based on computation of the relative frame connection, $E^{(\alpha/\alpha-1)}(t)$ is used to represent the relative position between two adjacent segments.

$$\left(\mathbf{e}^{(\alpha)}(t)\;\mathbf{r}_C^{(\alpha)}(t)\right) = \left(\mathbf{e}^{(\alpha-1)}(t)\mathbf{r}_C^{(\alpha-1)}(t)\right) E^{(\alpha/\alpha-1)}(t) \tag{5.48}$$

The relative frame connection consists of relative rotation matrix $R^{(\alpha/\alpha-1)}(t)$, see first cell i Eqn. 5.49 and relative position vector, see cell [2,1].

$$E^{(\alpha/(\alpha-1))}(t) = \begin{bmatrix} R^{(\alpha/(\alpha-1))}(t) & R^{(\alpha/(\alpha-1))}(t)s^{(\alpha/J^{(\alpha)})} + s^{(J^{(\alpha)}/(\alpha-1))} \\ 0 & 1 \end{bmatrix} \tag{5.49}$$

The inverse of the relative frame connection therefore is:

$$\left(E^{(\alpha/(\alpha-1))}\right)^{-1} = \begin{bmatrix} \left(R^{(\alpha/(\alpha-1))}\right)^{T}(t) & -\left(R^{(\alpha/(\alpha-1))}\right)^{T}(t)\left(R^{(\alpha/(\alpha-1))}(t)s^{(\alpha/J^{(\alpha)})} + s^{(J^{(\alpha)}/(\alpha-1))}\right) \\ 0 & 1 \end{bmatrix} \tag{5.50}$$

$$\dot{E}^{(\alpha/(\alpha-1))}(t) = \begin{bmatrix} \dot{R}^{(\alpha/(\alpha-1))}(t) & \dot{R}^{(\alpha/(\alpha-1))}(t)s^{(\alpha/J^{(\alpha)})} \\ 0_3^T & 0 \end{bmatrix} \tag{5.51}$$

$$\Omega^{(\alpha/\alpha-1)} \equiv \left(E^{(\alpha/\alpha-1)}(t)\right)^{-1} \dot{E}^{(\alpha/\alpha-1)}(t) \tag{5.52}$$

The time rate of absolute frame connection matrix $\Omega^{(\alpha)}$ is:

$$\Omega^{(\alpha)}(t) = \left(E^{(\alpha/\alpha-1)}\right)^{-1}(t)\Omega^{(\alpha-1)}(t)E^{(\alpha/\alpha-1)}(t) + \Omega^{(\alpha/\alpha-1)}(t) \tag{5.53}$$

$$\Omega^{(\alpha)}(t) = \begin{bmatrix} \overleftrightarrow{\omega^{(\alpha)}(t)} & v^{(\alpha)}(t) \\ 0_3^T & 0 \end{bmatrix} \tag{5.54}$$

Following same procedure, angular velocity $\overleftrightarrow{\omega^{(\alpha)}(t)}$ is extracted from the absolute time rate connection matrix $\Omega^{(\alpha)}$, and then processed through application of mathematical relation given by Eqns. 4.19 and 4.18. After that, the generalized expression of angular velocity is simplified, based on rotational relation $R^{(\alpha/\alpha)}(t) \equiv I$, allowing to express mathematically angular velocity as:

$$\omega^{(\alpha)} = \left(R^{(\alpha/1)}(t)\right)^{T} \omega^{(1)} + \left(\sum_{i=2}^{i=\alpha} \left(R^{(\alpha/i)}(t)\right)^{T} \omega^{(i/i-1)}\right) \tag{5.55}$$

$$S^{(\alpha/\alpha-1)} = \left(\overleftrightarrow{R^{(\alpha/\alpha-1)}s^{(\alpha/J_\alpha)} + s^{(\alpha/\alpha-1)}}\right)^T \tag{5.56}$$

$$\dot{x}^{(\alpha)} = R^{(\alpha-1)}S^{(\alpha/\alpha-1)}\omega^{(\alpha-1)} + R^{(\alpha)}\left(\overleftrightarrow{s^{(\alpha/J_\alpha)}}\right)^T\omega^{(\alpha/\alpha-1)} + \dot{x}^{(\alpha-1)} \tag{5.57}$$

5.3 B matrix

B matrix is organized in a similar way that for the FOWT itself defined by Eqn. 4.26.

Each column of B matrix corresponds to a certain angular velocity in the vector $\dot{q}(t)$. Linear $\dot{x}$ and angular velocities $\dot{\omega}$ of the respective α-segment fill in the B matrix from the first to the last α -segment, as shown in Eqn. 5.58.

$$\{\dot{\mathbf{X}}(t)\} \equiv \left\{\begin{matrix} \dot{x}_c^{(1)}(t) \\ \omega^{(1)}(t) \\ \dot{x}_c^{(2)}(t) \\ \omega^{(2)}(t) \\ \vdots \\ \dot{x}_c^{(\alpha)}(t) \\ \omega^{(\alpha)}(t) \end{matrix}\right\} \qquad \{\dot{q}(t)\} \equiv \left\{\begin{matrix} \omega^{(1)}(t) \\ \omega^{(2/1)}(t) \\ \omega^{(3/2)}(t) \\ \vdots \\ \omega^{(\alpha/\alpha-1)}(t) \end{matrix}\right\} \tag{5.58}$$

To represent dependancies on the rotational axis and inheritance of rotational direction from the previouse segment, the Lie generators of so(3) group is infered, see Eqn. 5.59.

$$g_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad g_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad g_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{5.59}$$

The generators can be easier understood through standard unit vectors , e_1, e_2 and e_3 with a magnitude of 1, as shown below:

$$\underset{3\times1}{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{5.60}$$

The Li generator uses alike pattern that resemble a Kronecker delta function. It assigns 1 when i equals j, otherwise it stays zero, as expressed mathematically in Eqn. 5.61

$$\delta_{i,j} \overset{\text{def}}{=} \begin{cases} 1 & \text{if} \quad i=j \\ 0 & \text{otherwise} \end{cases} \tag{5.61}$$

Using this logic, Li generator affects the calculation of B matrix and consequently the outcome of the EOM and therfore the whole dynamics of the modeled system is affected.

$$B_{(2i-1)j} = \left(1-\delta_{ij}\right)\left(R^{(i-1)}S^{(i/i-1)}R^{(i/j)}g^{(j/j-1)} + B_{(i-1)j}\right) + \delta_{ij}R^{(i)}\overleftrightarrow{\left(s^{\left(i/J^{(i-1)}\right)}\right)^T} g^{(i/i-1)} \tag{5.62}$$

$$B_{(2i)j} = \left(1-\delta_{ij}\right)\left(R^{(i/j)}\right)^T g^{(j/j-1)} + \delta_{ij}g^{(j/j-1)} \tag{5.63}$$

Based on Eqns. 5.62 and 5.63 the terms of B-matrix can bewe directly calculated for α-segments.

$$[B(t)] = \begin{bmatrix} B_{1,1} & \underset{3\times1}{0} & \underset{3\times1}{0} & \cdots & \underset{3\times1}{0} & \underset{3\times1}{0} \\ B_{2,1} & B_{2,2} & \underset{3\times1}{0} & \cdots & \underset{3\times1}{0} & \underset{3\times1}{0} \\ B_{3,1} & B_{3,2} & B_{3,3} & \cdots & \underset{3\times1}{0} & \underset{3\times1}{0} \\ B_{4,1} & B_{4,2} & \underset{3\times1}{0} & \cdots & \underset{3\times1}{0} & \underset{3\times1}{0} \\ B_{5,1} & B_{5,2} & B_{5,3} & \cdots & \underset{3\times1}{0} & \underset{3\times1}{0} \\ B_{6,1} & B_{6,2} & B_{6,3} & \cdots & \underset{3\times1}{0} & \underset{3\times1}{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ B_{(m-1),1} & B_{(m-1),2} & B_{(m-1),3} & \cdots & B_{(m-1),(n-1)} & B_{(m-1),n} \\ B_{m,1} & B_{m,2} & B_{m,3} & \cdots & B_{m,(n-1)} & B_{m,n} \end{bmatrix} \tag{5.64}$$

Taking into account the definition of generator g given by Eqns. 5.59, and 5.61, the B -terms in B matrix (Eqns. 5.64) are calculated based on the developed computational algorithm provided by Eqns. 5.62 and 5.63. B-terms are simlified by applying two important algebraic operations, Eqns. 4.19 and 4.18

- **If** $i = 1 \rightarrow \delta_{11} = 1$
  - $B_{(2\times1-1)1} = B_{1,1} = R^{(1)}\overleftrightarrow{\left(s^{\left(1/J^{(0)}\right)}\right)^T} g^{(1/0)}$
    where $g^{(1/0)} = g_1$ according to initial rotation, Eqn. 5.17.

    The transpose vanishes, after introducing a negative sign under algebraic operation, Eqn. 4.19. Therefore $B_{1,1} = -R^{(1)}(t)\overleftrightarrow{s^{(1)}}g_1(t)$

- $B_{2\times 1,1} = B_{2,1} = g_1$

- **If** $i = 2 \rightarrow \delta_{21} = 0$ and $S^{(2/1)}$=$\left(\overleftrightarrow{R^{(2/1)}s^{(2/J^{(2)})} + s^{(J^{(2)}/1)}}\right)^T$ from Eqn. 5.45
  - for $j = 1$
    - * $B_{(2\times 2-1)1} = B_{3,1} = R^{(1)}S^{(2/1)}R^{(2/1)}g_1 + B_{1,1}$
    - * $B_{(2\times 2)\ 1} = B_{4,1} = (R^{(2/1)})^T g_1$
  - for $j = 2$ $\delta_{22} = 1$
    - * $B_{3,2} = -R^{(2)}\overleftrightarrow{s^{(2/J^{(1)})}}g_3$
    - * $B_{4,2} = g_3$

The other terms of B-matrix are calculated based on the same algorithm through symbolic programming in open source CAS software, Julia.jl, which uses a wrapper SymEngine $C++$ [37]. Another CAS software, as for example SymPy of Python , Mathematica of Wolfram Alfa or Maple, are capable to handle this task with similar success. Assignment of active rotational axis to odd or even indices is based on g -generator value in such a manner, that value of 1 is being assigned to rows $3, 9, 15, 21$, and 27 and column indices $1, 2, 3, 4, 5$ for even indices. And for odd indices to rows $5, 11, 17, 23$, and 29, and column indices $1, 2, 3, 4$, and 5. This algorithm have been implemented by means of *modulus* in MATLAB code and can be easily observed n B-matrix.

5.4 Frame Connection

To represent the forces acting at fairlead on the platform in the inertial frame a vector is created by identifying the vector connecting the platform and fairleads, given by Eqn. 5.74.

Finding the optimal vertical position of the fairlead requires identification of the points where no coupling between surge and pitch happens. In this way, dynamic effects, such as overturning moment, caused by wind and waves are reduced and stability of FOWT is enhanced. based on engineering principles and knowledge within the field, an appropriate location for such a deep-draught floating system will lie somewhere between COB and COG, but closer to the latter [38], [39].

There are also placed frames at points $\mathbf{s}^{L\alpha}(t)$, where the individual mooring line is connected to the respective fairlead. These points are mooring line connection points, as represented on Figure 5.6, and the distance between them is equal to $\theta_{delta\ connection} = 120\,^{\circ} = \frac{\pi}{2*3} = \frac{\pi}{6}$, while fairlead connection frame, $\mathbf{e}^{(\mathrm{Fl})}(t)$, is given in Eqn. 5.73.

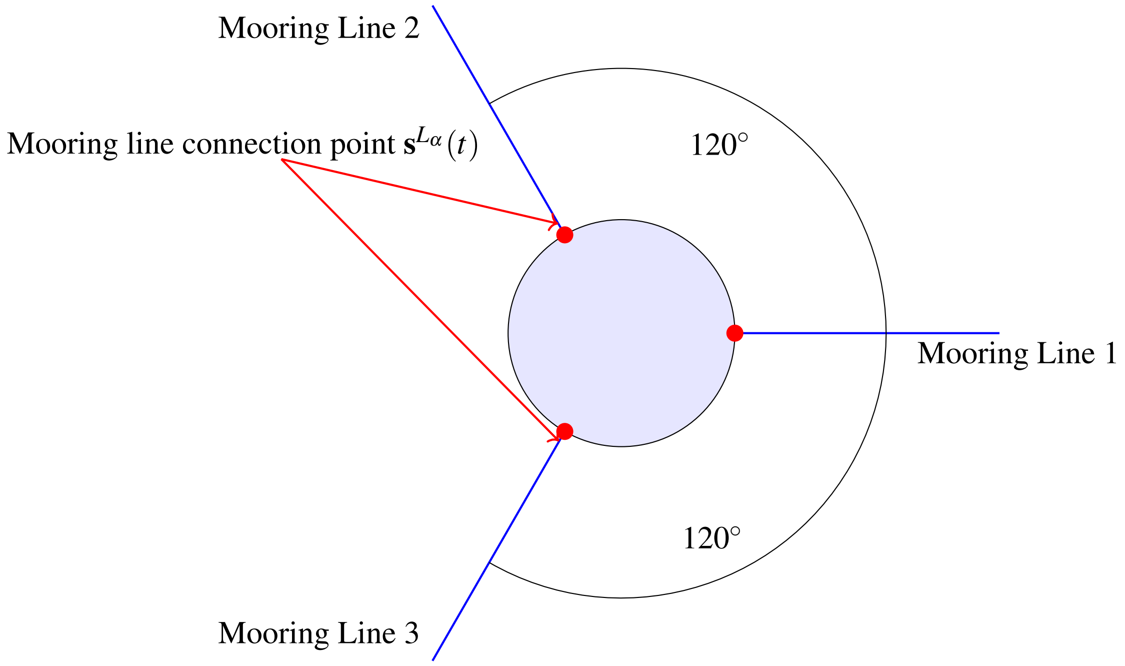


Figure 5.6: Location of the mooring line's connection point, $\mathbf{s}^{L\alpha}(t)$.

$$\mathbf{e}^{(L_1)}(t) = \mathbf{e}^{(\mathrm{Fl})}(t)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.65}$$

$$\mathbf{e}^{(L_2)}(t) = \mathbf{e}^{(\mathrm{Fl})}(t)\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.66}$$

$$\mathbf{e}^{(L_3)}(t) = \mathbf{e}^{(\mathrm{Fl}\,)}(t)\begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.67}$$

To determine the location of the mooring line's connection point with respect to the fairlead in the inertial frame, consider the following equation:

$$\vec{s}^{\,L\alpha/Fl}(t) = \vec{s}^{\,L\alpha}(t) - \vec{s}^{\,Fl}(t) \tag{5.68}$$

Where mooring line connecting point vector $\vec{s}^{\,L\alpha}(t)$ is expressend in local frame with Eqn. 5.69.

$$\mathbf{s}^{L\alpha}(t) = \mathbf{e}^{L\alpha}(t) s^{L\alpha}(t) = \mathbf{e}^{L\alpha}(t) \begin{pmatrix} s_1^{L\alpha} \\ 0 \\ s_3^{L\alpha} \end{pmatrix} \tag{5.69}$$

Here $\mathbf{e}^{L\alpha}(t)$ is a delta connection frame of the respective mooring line connection point, given in 5.65 -5.67, and $\frac{d}{2}$ is the radius of the platform, which assumed to be equal to the fairlead location, which is logical.

Now, consider formulation of the location of the upper fastening point (fairlead) of a mooring line, during a general displacement of the platform, adopted from Fossen T.I., given in the inertial frame as:

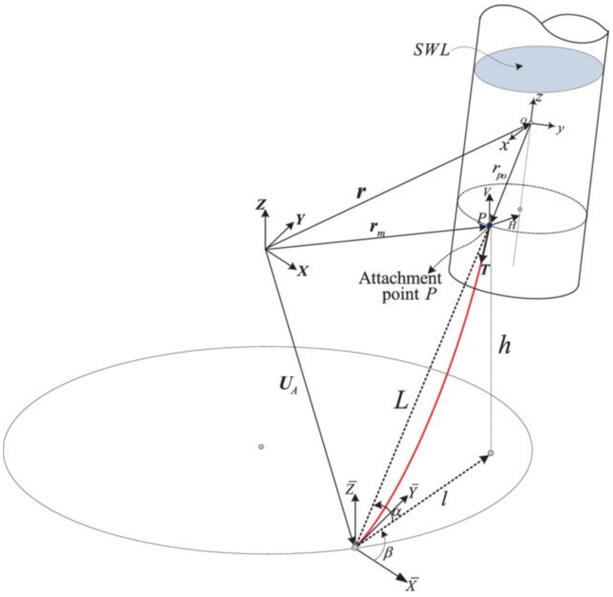


Figure 5.7: Generic mooring line geometry [4].

$$r_m = \{X_p\ Y_p\ Z_p\}^T = r + R\, r_{\text{po}} = \{r_X\ r_Y\ r_Z\}^T + R\,\{x_{\text{po}}\ y_{\text{po}}\ z_{\text{po}}\}^T \quad [40] \tag{5.70}$$

Where the displacement of the origin (x,y,z) of the body frame relative to the inertial frame is denoted by vector $r = \left\{ \begin{array}{ccc} r_X & r_Y & r_Z \end{array} \right\}^T$, while r_{po} is the position of the fairlead (upper attachment point) in the body frame, as shown in Figure 5.7. Vectors in the body frame are transformed to the inertial frame by rotation matrix of the platform R by a Euler angle triad $\left[\begin{array}{ccc} \phi & \theta & \psi \end{array} \right]^T$, Eqn. 5.71. [40]

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{5.71}$$

In the above equations, according to Fossen T.I., the change in position of the upper attachment point, relative to changes in the platform's orientation should be defined as:

$$\frac{\partial r_m}{\partial \zeta_i} = \frac{\partial R}{\partial \zeta_i} r_{\text{po}}, \quad \zeta_i = \phi, \theta, \psi \quad [40] \tag{5.72}$$

Here $\phi,\ \theta,\ \psi$ corresponds to roll, pitch and yaw angles.

In a similar way, a connection between fairlead and platform is established, by inserting a frame $\mathbf{e}^{(\text{Fl})}$ at the vertical position of the fairleads, ensuring that its main axes are aligned with platform principle axis.

$$\mathbf{e}^{(\text{Fl})} = \mathbf{e}^{(1)}(t)I \tag{5.73}$$

Here I is the identity matrix and $\mathbf{e}^{(1)}(t)$ represents platform frame given by Eqn. 4.5

Therefore the position of fairlead relative to platform frame, is expressed with equation Eqn. 5.74

$$\vec{s}^{\,(Fl/1)}(t) = \mathbf{e}^{(1)}(t) s^{(\text{Fl}\,/1)}(t) \tag{5.74}$$

While in inertial frame, fairlead is defined by vector $\vec{s}^{\,Fl}(t)$, and by using frames, $s^{\,Fl}(t)$:

$$\vec{s}^{\,Fl}(t) = \vec{s}^{\,(1)}(t) + \vec{s}^{\,(Fl/1)}(t) \tag{5.75}$$

$$\mathbf{s}^{\,Fl}(t) = \mathbf{e}^I x_c^{(1)}(t) + \mathbf{e}^{(1)}(t) s^{(Fl/1)}(t) \tag{5.76}$$

Therefore, the vector pointing in the direction the mooring line pulls on the fairlead, is defined by Eqn. 5.77:

$$\mathbf{s}^{(L\alpha/Fl)} = \mathbf{s}^{L\alpha}(t) - \mathbf{s}^{\,Fl}(t) \tag{5.77}$$

By re-assembling and unpacking back terms $\mathbf{s}^{L\alpha}(t)$, Eqn. 5.69, and $\mathbf{s}^{\,Fl}(t)$, into Eqn. 5.77, the final output which represents the vector connecting a fairlead to a mooring line connection point, is obtained:

$$\mathbf{s}^{(L\alpha/Fl)} = \mathbf{e}^{L\alpha}(t)s^{L\alpha}(t) - \left(\mathbf{e}^{I}x_c^{(1)}(t) + \mathbf{e}^{(1)}(t)s^{(Fl/1)}(t)\right) \tag{5.78}$$

In Equation 5.77, the negative sign before the final term indicates the direction of the force exerted by the mooring line due its tension towards mooring line connection point, that pulls fairlead towards the anchor point.

Upon the realization that equation in Eqn. 5.78 might be further simplified by returning some terms back to the inertial frame with Eqn. 4.29, redefined generalized vector in the inertial frame is:

$$\mathbf{s}^{(L\alpha/Fl)} = \mathbf{e}^{(I)}R^{(1)}(t)R^{L\alpha}(t)s^{L\alpha}(t) - \left\{\mathbf{e}^{I}x_c^{(1)}(t) + \mathbf{e}^{I}(t)R^{(1)}(t)s^{(Fl/1)}(t)\right\} \tag{5.79}$$

Suppose that there is a certain amount of tension T exerted by the mooring line along its length. Then, the force vector $\vec{F}$, acting in horisontal plane, at the fairlead towards the mooring line connection point, $\mathbf{s}^{(L\alpha/Fl)}$, is normalized and multiplyed with acting force, second term in Eqn. 6.19.

$$\left\|\mathbf{s}^{(L\alpha/Fl)}\right\| \tag{5.80}$$

Which in inertial frame is represented by the unit vector:

$$\frac{1}{\left\|\mathbf{s}^{(L\alpha/Fl)}\right\|}\mathbf{s}^{(L\alpha/Fl)} \tag{5.81}$$

Where $\left\|\mathbf{s}^{(L\alpha/Fl)}\right\|$ refers to the magnitude of the vector $\mathbf{s}^{(L\alpha/Fl)}$.

Then force vector in both vertical and horizontal direction is given by Eqn. 6.19

$$\mathbf{e}^{I}F^{\mathrm{L}\alpha} = \mathbf{e}^{I}\begin{pmatrix} 0 \\ 0 \\ -F_3^{(\alpha)} \end{pmatrix} - \mathbf{e}^{I}\frac{1}{\left\|\mathbf{s}^{(L\alpha/Fl)}\right\|}\mathbf{s}^{(L\alpha/Fl)}F_1^{(\alpha)} \tag{5.82}$$

The negative sign reveals the direction of the force, which acts towards the connection point of the mooring line, effectively pulling the fairlead in that direction. Note that vertical force is represented only by it's magnitude, since the direction of the force in the vertical plane is considered to be aligned with gravity and the system's vertical axis.

Therefore the total force coming from all mooring lines that acts on the platform is a sum of individual forces in inertial frame:

$$\mathbf{F}^{I,\,L_{tot}} = \sum_{\alpha=1}^{\alpha=3} \mathbf{e}^{I}(t) F^{L\alpha} \tag{5.83}$$

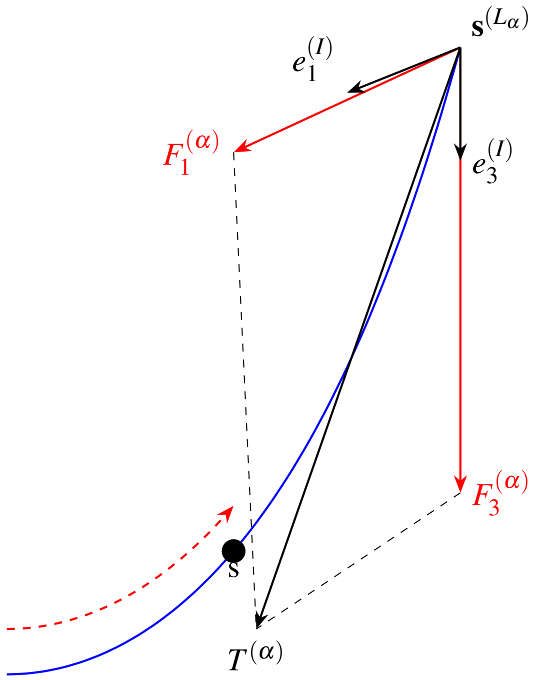
The moment from each mooring line that acts on the platform is determined by positioning a vector from the frame on the platform to the fairlead where the mooring line load is applied, at $\mathbf{s}^{(L\alpha/1)}$.

$$\vec{\mathbf{s}}^{\,(L\alpha/1)} = \vec{s}^{\,L\alpha}(t) - \vec{s}^{\,(1)}(t) \tag{5.84}$$

Where $\vec{s}^{\,L\alpha}(t)$ is found from Eqn. 5.77 and vertical distance to fairlead.

Then the moment from all lines in inertial frame is:

$$\mathbf{M}^{I,\,L_{tot}} = \sum_{\alpha=1}^{\alpha=3} \mathbf{e}^{I}(t) \overleftrightarrow{s^{(L\alpha/1)}} F^{I,\,L\alpha} \tag{5.85}$$



Grafic representation of forces that act at mooring line connection point $\mathbf{s}^{(L\alpha)}$.

6 Kinetics

While Kinematics studies movement irrespective of forces that causes them, concentrating severally on the geometrical aspect. Kinetics, better known as dynamics concentrates on how forces affects and produce motion. For simplicity, let's refer to the Lagrangian as to the difference between kinetic energy and potential energy.

$$Ł^{(\alpha)}(q(t),\dot{q}(t)) = K^{(\alpha)}(q(t),\dot{q}(t)) - U^{(\alpha)}(q(t)) \tag{6.1}$$

Then by integratiing the Lagrangian function over a period of time, the Action can be expressed as:

$$A = \int_{t_0}^{t_1} Ł^{(\alpha)}(q(t),\dot{q}(t),t)dt \tag{6.2}$$

According to Hamilton's principle, the system's motion can be determined by pushing a certain definite integral to minimum possible arbitrary variations, for example by applying prescribed and therefore known initial and final conditions to the system. This allows to set the variation of the Action to zero and obtain easily EOM:

$$\delta \int_{t_0}^{t_4} Ł^{(\alpha)}(q(t),\dot{q}(t),t)dt = 0 \tag{6.3}$$

In order to consider non-conservative forces, an expanded version of Hamilton's Principle called the Principle of Virtual Work is used. In this approach, the Lagrangian is expressed exclusively in terms of kinetic energy. All additional forces, whether conservative or non-conservative, are treated as work and are placed on the right side of the equation. Moving forward, the notation is simplified by excluding the dependencies on position and velocity.

$$\int_{t_0}^{t_1} \delta K^{(\alpha)}(t)dt = -\int_{t_0}^{t_1} \delta W^{(\alpha)}(t)dt \tag{6.4}$$

Llinear momentum $\mathrm{L}_{\mathrm{C}}^{(\alpha)}(\mathrm{t})$ and the angular momentum $\mathrm{H}_{\mathrm{C}}^{(\alpha)}(\mathrm{t})$ define the kinetic energy of each body:

$$\mathbf{H}_C^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) H_C^{(\alpha)}(t) = \mathbf{e}^{(\alpha)}(t) J_C^{(\alpha)} \omega^{(\alpha)}(t) \tag{6.5}$$

$$\mathbf{L}_C^{(\alpha)}(t) = \mathbf{e}^I L_C^{(\alpha)}(t) = \mathbf{e}^I m^{(\alpha)} \dot{x}_C^{(\alpha)}(t) \tag{6.6}$$

Where $\mathrm{J}_{\mathrm{C}}^{(\alpha)}$ is the moment of inertia for each α-body and net kinetic energy is simply given by Eqn. 6.7

$$K^{(\alpha)}(t) = \frac{1}{2}\left\{\dot{\mathbf{r}}_C^{(\alpha)} \cdot \mathbf{L}_C^{(\alpha)} + \omega^{(\alpha)} \cdot \mathbf{H}_C^{(\alpha)}\right\} \tag{6.7}$$

Then net kinetic energy is just a matrix:

$$K(t) = \frac{1}{2}\{\dot{X}(t)\}^T [M] \{\dot{X}(t)\} \tag{6.8}$$

Where generalized mass matrix [M] is formed by the terms of masses and moments of inertia for each body:

$$[M] = \begin{bmatrix} m^{(1)} I_3 & \underset{3\times3}{0} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \underset{3\times3}{0} & J_C^{(1)} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \underset{3\times3}{0} & \underset{3\times3}{0} & m^{(2)} I_3 & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \underset{3\times3}{0} & \underset{3\times3}{0} & \underset{3\times3}{0} & J_C^{(2)} & \cdots & \underset{3\times3}{0} \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ \underset{3\times3}{0} & \underset{3\times3}{0} & \underset{3\times3}{0} & \cdots & m^{(a)} I_3 & \underset{3\times3}{0} \\ \underset{3\times3}{0} & \underset{3\times3}{0} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & J_C^{(a)} \end{bmatrix} \tag{6.9}$$

Hamilton's Principle doesn't consider non-conservative forces. To address this, Hamilton's Principle is extended and represented already as the Engineering Principle of Virtual Work. This means that the work done should be calculated. The MFM restricts virtual rotation and represents it as a natural conjugate pair expressed in the moving body frame in form of angular velocity and moment.

By skewing the virtual rotational displacement $\delta\pi^{\alpha}(t)$ as shown in Eqn. 6.10, the EOM can be established then in a simpler way.

$$\overleftrightarrow{\delta\pi^{(\alpha)}(t)} = \left(R^{(\alpha)}(t)\right)^T \delta R^{(\alpha)}(t) \tag{6.10}$$

The virtual Cartesian displacements $\{\delta\widetilde{X}(t)\}$ are determined by the equation presented below.

$$\{\delta\widetilde{X}(t)\} = \begin{pmatrix} \delta x^{(1)}(t) \\ \delta\pi^{(1)}(t) \\ \delta x^{(2)}(t) \\ \delta\pi^{(2)}(t) \\ \vdots \\ \delta x^{(\alpha)}(t) \\ \delta\pi^{(\alpha)}(t) \end{pmatrix} \tag{6.11}$$

Following this, the virtual Cartesian velocities $\{\delta\dot{X}(t)\}$ represent the variation of the velocities

$$\{\delta\dot{X}(t)\} = \begin{Bmatrix} \delta\dot{x}_c^{(1)}(t) \\ \delta\omega^{(1)}(t) \\ \delta\dot{x}_C^{(2)}(t) \\ \delta\omega^{(2)}(t) \\ \vdots \\ \delta\dot{x}_c^{(\alpha)}(t) \\ \delta\omega^{(\alpha)}(t) \end{Bmatrix} \tag{6.12}$$

When it comes to linear displacement, the change in the derivative matches the derivative of the variation.

$$\delta\dot{x}_C^{(\alpha)}(t) = \frac{d}{dt}\delta x_c^{(\alpha)}(t) \tag{6.13}$$

But, there are limitations when it comes to variations in angular velocity:

$$\delta\omega^{(\alpha)}(t) = \frac{d}{dt}\delta\pi^{(\alpha)}(t) + \overleftrightarrow{\omega^{(\alpha)}(t)}\delta\pi^{(\alpha)}(t) \tag{6.14}$$

The final two equations are presented in a condensed manner as:

$$\{\delta\dot{X}(t)\} = \{\delta\dot{X}(t)\} + [D]\{\delta\widetilde{X}(t)\} \tag{6.15}$$

The angular velocity matrices for each segment are stored in a skew form inside [D] matrix.

$$[D(t)] = \begin{bmatrix} \underset{3\times3}{0} & \underset{3\times3}{0} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \underset{3\times3}{0} & \overleftrightarrow{\omega^{(1)}(t)} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \underset{3\times3}{0} & \underset{3\times3}{0} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \underset{3\times3}{0} & \underset{3\times3}{0} & \overleftrightarrow{\omega^{(2)}(t)} & \cdots & \underset{3\times3}{0} & \underset{3\times3}{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \underset{3\times3}{0} & \underset{3\times3}{0} & \underset{3\times3}{0} & \cdots & \underset{3\times3}{0} & \overleftrightarrow{\omega^{(a)}(t)} \end{bmatrix} \tag{6.16}$$

Now, the kinetic energy variation defined in cartesian coordinates by Eqn. 6.17, can be expressed in terms of genralized coordinates by Eqn.6.18.

$$\delta K(t) = \{\delta \dot{X}(t)\}^T [M] \{\dot{X}(t)\} \tag{6.17}$$

$$\delta K(t) = \left\{ \frac{d}{dt}[B(t)]\{\delta q(t)\} + [D(t)][B(t)]\{\delta q(t)\} \right\}^T [M][B(t)]\{\dot{q}(t)\} \tag{6.18}$$

For each body, a set lists a force vector containing forces and moments that should be defined as a single column matrix $\{Q(t)\}$ following the form presented below:

$$\{F(t)\} = \begin{Bmatrix} F^{(1)}_{3\times 1}(t) \\ M^{(1)}_{3\times 1}(t) \\ F^{(2)}_{3\times 1}(t) \\ M^{(2)}_{3\times 1}(t) \\ F^{(3)}_{3\times 1}(t) \\ M^{(3)}_{3\times 1}(t) \\ \vdots \\ F^{(\alpha-1)}_{3\times 1}(t) \\ M^{(\alpha-1)}_{3\times 1}(t) \\ F^{(\alpha)}_{3\times 1}(t) \\ M^{(\alpha)}_{3\times 1}(t) \end{Bmatrix} \tag{6.19}$$

The force vector for FOWT model in this project is given by Eqn. 6.20

$$\{F(t)\} = \begin{Bmatrix} F^{(1)}_{3\times 1}(t) \\ M^{(1)}_{3\times 1}(t) \\ F^{(2)}_{3\times 1}(t) \\ M^{(2)}_{3\times 1}(t) \\ F^{(3)}_{3\times 1}(t) \\ M^{(3)}_{3\times 1}(t) \end{Bmatrix} = \begin{Bmatrix} \mathrm{F}^{(1)}_{\mathrm{hydro}}(\mathrm{t}) + \mathrm{F}^{(1)}_{\mathrm{external}} \\ \mathrm{M}^{(1)}_{\mathrm{hydro}}(\mathrm{t}) + \mathrm{M}^{(1)}_{\mathrm{external}} \\ \mathrm{F}^{(2)}_{\mathrm{nacelle}}(\mathrm{t})) \\ \mathrm{M}^{(2)}_{\mathrm{nacelle}}(\mathrm{t})) \\ \mathrm{F}^{(3)}_{\mathrm{rotor}}(\mathrm{t}) \\ \mathrm{M}^{(3)}_{\mathrm{rotor}}(\mathrm{t}) \end{Bmatrix} \tag{6.20}$$

The terms are:

$$\begin{aligned} F_{hydro} &= \text{Force induced by waves on the respective body} \\ F_{external} &= \text{Force caused by external forces} \\ M_{hydro} &= \text{Moment induced by waves} \\ M_{external} &= \text{Moment caused by external forces} \end{aligned}$$

In section 3 of this work, the total hydrodynamic forces and moments caused by waves, referred as F_{hydro} and M_{hydro} are described in details. These forces include excitation, added mass, radiation damping, and external hydrostatic stiffness. Additionally, there are also discussed external forces and moments, $F_{external}$ and $M_{external}$.
Solving EOM numerically, the resulting row vector of size $3nx1$ is vtained and contains the next time step for the essential generalized velocity $\dot{q}(t)_{n+1}$. The forces and moments that do a virtual work can be expresed in cartesian coordinates by Eqn. 6.21:

$$\delta W = \{\delta \widetilde{X}(t)\}^T \{Q(t)\} \tag{6.21}$$

The B-matrix connects the Cartesian velocities$\{\dot{X}(t)\}$ to the essential generalized velocities $\{\dot{q}(t)\}$, as well as linking the virtual generalized displacements $\{\delta \widetilde{X}(t)\}$ to the essential virtual displacements $\{\delta q(t)\}$.

$$\{\delta \tilde{X}(t)\} = [B(t)]\{\delta q(t)\} \tag{6.22}$$

Updating the virtual work given in Eqn. 6.21 by the transpose the virtual work becomes:

$$\delta W(t) = \{\delta q(t)\}^T \{F^*(t)\} \tag{6.23}$$

$\{F^*(t)\}$ are the essential generalized forces given by:

$$\{F^*(t)\} = [B(t)]^T \{Q(t)\} \tag{6.24}$$

When pluging in the virtual work and the variation of the kinetic energy into Eqn. 6.27, we lay the groundwork for the EOM.

$$\int_{t_0}^{t_1} \left(\{\delta \dot{X}(t)\}^T [M]\{\dot{X}(t)\} + \{\delta q(t)\}^T \{F^*(t)\}\right) dt = 0 \tag{6.25}$$

6.1 Equation of motion

Euler's equation in the form shown by Eqn. 6.26 is extracted from minimizing the action and is simplified to Eqn. 6.29 by definitions of M^* and N^* given in Eqns. 6.27 and 6.28.

$$\left[\begin{array}{c} [B(t)^T]\left([M]\left(\begin{array}{c}\dot{B}(t)\left\{\dot{q}(t)\right\}\ + \\ B(t)]\left\{\ddot{q}(t)\right\}\end{array}\right)+[D]\,[M]\left([B(t)]\left\{\dot{q}(t)\right\}\right)\right) \\ -\left\{F^*\right\} \end{array}\right]=0 \tag{6.26}$$

$$[M^*(t)] \equiv [B(t)]^T[M][B(t)] \tag{6.27}$$

$$N^* = [B(t)]^T\left([M][\dot{B}(t)]+[D(t)][M][B(t)]\right) \tag{6.28}$$

After using integration by parts on equation Eqn. 6.25 and considering that there is no virtual displacement at the endpoints, we end up with a second-order coupled differential equation.

$$[M^*(t)]\left\{\ddot{q}(t)\right\}+[N^*(t)]\left\{\dot{q}(t)\right\}=\left\{F^*(t)\right\} \tag{6.29}$$

Extracted EOM defined in Eqn. 6.29 is solved with respect to generalized acceleration $\ddot{q}(t)$, as shown in Eqn. 6.29 discretized as in Eqn. 7.2

$$\left\{\ddot{q}(t)\right\}=[M^*(t)]^{-1}\left(\left\{F^*(t)\right\}-[N^*(t)]\left\{\dot{q}(t)\right\}\right) \tag{6.30}$$

The generalized force vector, $F^*(t)$ is representation of generalized forces derived in Chapter 3 and included in Eqns. 6.23 to 6.25.

7 Numerical method

Prior to conducting the analysis, a force vector is defined based on hydrodynamic discussed in Chapter 3 and Principles of Virtual, Eqns. 6.23 to 6.25 and incorporated into EOM in Section 6.1. Following this, a time series is computed using the simulation's time interval. The wave excitation force is then isolated from this time series for every complete cycle of the Runge-Kutta 4th-order technique and added to other forces that a acts on FOWT.

7.1 The Runge-Kutta Method

One way to calculate ordinary differential equations $(ODEs)$ with numerical integration involves taking a trial step at the midpoint of an interval in order to eliminate lower-order global truncation error terms, $O(h^k)$. The most commonly used method, the Runge-Kutta method, is especially effective and precise with a value of $k = 4$. By utilizing RK4 method, we can obtain approximate values for the solution of initial value problem (IVP) at various points such as x_0, $x_0 + h$, and $x_0 + nh$ [41].

The Runge-Kutta method (RK4) is implemented in the general s-stage explicit form, iteratively computing the slopes k_1, k_2, ..., k_s at each stage of function estimation, as shown by Eqn. 7.1. When solving first order ODE, $\dot{y} = f(t,y)$, it calculates the next value y_{n+1}, based on the previous value y_n plus a weighted average of four increments, so that:

$$\begin{aligned}
\mathrm{k}_1 &= hf\left(\mathrm{t_n} + \mathrm{c_1\ h}, \mathrm{y_n}\right) \\
\mathrm{k}_2 &= \mathrm{hf}\left(\mathrm{t_n} + \mathrm{c_2\ h}, \mathrm{y_n} + \mathrm{a}_{(2,1)}\mathrm{k}_1\right) \\
\mathrm{k}_3 &= \mathrm{hf}\left(\mathrm{t_n} + \mathrm{c_3\ h}, \mathrm{y_n} + \mathrm{a}_{(3,1)}\mathrm{k}_1 + \mathrm{a}_{(3,2)}\mathrm{k}_2\right) \\
&\vdots \\
\mathrm{k}_s &= \mathrm{hf}\left(\mathrm{t_n} + \mathrm{c_s h}, \mathrm{y_n} + \textstyle\sum_{\mathrm{i}=1}^{s-1} \mathrm{a}_{(s,\mathrm{i})}\mathrm{k_i}\right) \\
\mathrm{y_{n+1}} &= \mathrm{y_n} + \textstyle\sum_{\mathrm{j}=1}^{s} \mathrm{b_j k_j}
\end{aligned} \tag{7.1}$$

Where

- h: is the size over the interval the solution is estimated.
- t_n: represents the current time.
- y_n: intermediate approximation of the solution at time t_n.
- $c_1, c_2, \ldots, c_s$: are time coefficients of each stage, denoting the time points within the interval the slopes are estimated.
- $a_{(i,j)}$: are coefficients in the Butcher tableau represent the weights applied to slopes from previous stages to calculate the slope at stage i.
- $b_1, b_2, \ldots, b_s$: are coefficients that represent the weights applied to slopesb at each stage to get the final approximation of the solution.

The values of coefficients implemented in RK4 are given in the Butcher tableau below.

Table 7.1: Butcher tableau for RK4

$$\begin{array}{c|c} c & a \\ \hline & b \end{array} = \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

- c_i are time coefficients at which the function evaluations given by Eqn. 6.30 are performed.
- The weighting coefficients a_{ij} are used to estimate the weighted sum of intermediate stages of function evaluations. These coefficients makes each calculation stage dependable on the previous calculation stage. The diagonal terms a_{ii} shows the weights of the intermediate calculations, while the off-diagonal terms a_{ij} reveal the dependencies between stages.
- b_i are values of final approximation that shows the weighted sum of the function calculations at each stage.

Let define $\dot{q}_n$ at next n time step as $\dot{q}_{n+1}$, and $\ddot{q}_n$ as $\ddot{q}_{n+1}$, then Eqn. 6.30 can be descretized as [31]:

$$\ddot{q}_{n+1} = [M_n^*]^{-1}[F_n^* - N_n^* \dot{q}_n] \tag{7.2}$$

Applying Eqn. 7.1 together with coefficients provided by Butcher tableau to Eqn. 6.30, results in function approximation of the solution represented by equation Eqn. 7.7 and therefore provides a solution for angular acceleration given in Eqn. 7.2.

$$k_1 = [M_n^*]^{-1}[F_n^* - N_n^* \dot{q}_n] \tag{7.3}$$

$$k_2 = [M_{n+0.5\Delta t}^*]^{-1} \left[F_{n+0.5\Delta t}^* - N_{n+0.5\Delta t}^* \left(\dot{q}_n + \frac{1}{2} k_1 \Delta t\right)\right] \tag{7.4}$$

$$k_3 = [M_{n+0.5\Delta t}^*]^{-1} \left[F_{n+0.5\Delta t}^* - N_{n+0.5\Delta t}^* \left(\dot{q}_n + \frac{1}{2} k_2 \Delta t\right)\right] \tag{7.5}$$

$$k_4 = [M_{n+\Delta t}^*]^{-1} \left[F_{n+\Delta t}^* - N_{n+\Delta t}^* (\dot{q}_n + k_3 \Delta t)\right] \tag{7.6}$$

Then, the next value is given by:

$$\dot{q}_{n+1} = \dot{q}_n + \Delta t \frac{1}{6} (k_1 + k_2 + k_3 + k_4) \tag{7.7}$$

7.2 Post-processing

It is assumed that the motion of a FOWT is linearly related to surface waves. This implies that in irregular waves it may be computed as a sum of motions arising from the regular waves of different amplitudes, frequencies and direction of wave propagation of the relevant sea state. As such, turbine motion for every sea state is calculated using Fast Fourier Transform (FFT) and is presented as one-sided, unsmoothed, discrete power density function (PSD), described with Eqn. 7.8 [19]. For this, MATLAB code has been written which is based on code from NTNU in connection with Data Analysis linked to the course "Experimental Methods in Marine Hydrodynamics".

The FFT of a time-serie$x(t)$ is sampled at N points: [42]. Figure 7.1 shows transformation principle of wave energy into response energy [5].

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N} \tag{7.8}$$

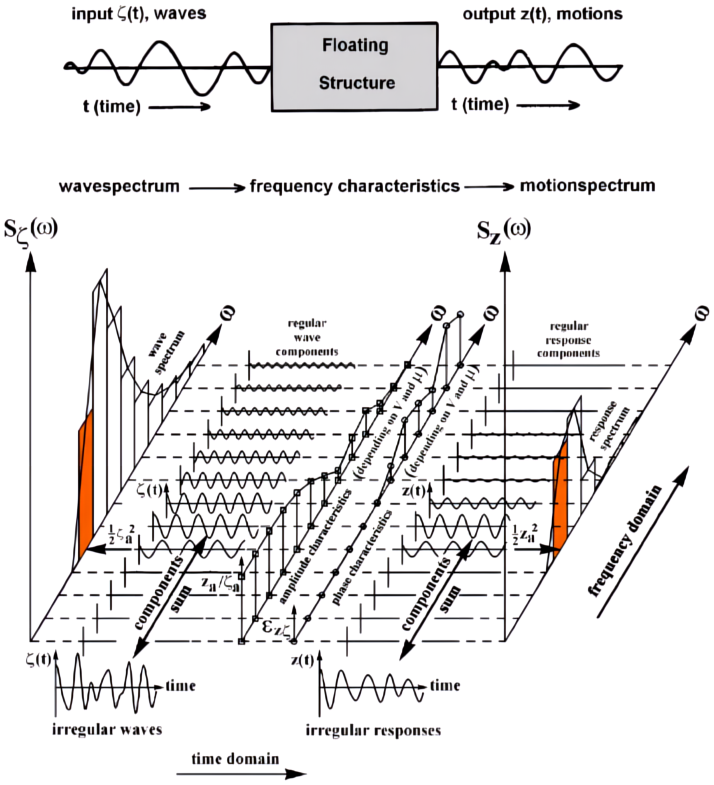


Figure 7.1: Principle of Transfer of Waves into Responses [5].

where $x[n]$ is the signal value at sample n, and k is the index in the frequency domain ranging from 0 to $N-1$. The Power Spectral Density (PSD) is computed from the FFT as:

$$S[k] = \frac{2\Delta t}{N}|X[k]|^2 \tag{7.9}$$

where Δt is the sampling interval, N is the total number of samples, and $|X[k]|^2$ is the squared magnitude of the FFT at frequency bin k. The factor of 2 accounts for the one-sided spectrum of a real-valued signal.

The frequencies that corresponds to the FFT data are given by:

$$f[k] = \frac{k}{N\Delta t} \tag{7.10}$$

for $k = 0, 1, 2, \ldots, \frac{N}{2}$, where $\frac{N}{2}$ is the Nyquist limit.

7.3 Computation in MATLAB

The simulation code structure is complex and consists of several substructures, submodules and functions. The main module handles the outer loop, solving EOM via Runge-Kutta numerical method presented in Chapter 7.

The other submodules are responsible for initialization of the catenary mooring line. The first module initializes quasi-static catenary model, through Newton-Raphson test as discussed in Section 5.1.

The second module takes over parameters established in the first module, through partial derivation of multi-segmented line, as described in Section 5.2

MFM-based model is coupled to spar FOWT model through linkage of parallel computational processes, such as B matrix and total forces F^* computations in parallel loop, establishing overall EOM of the system, according to process described in Section 6.1. Figure 7.2 shows the flowchart of the coupled analysis. Code is presented in Appendix B.

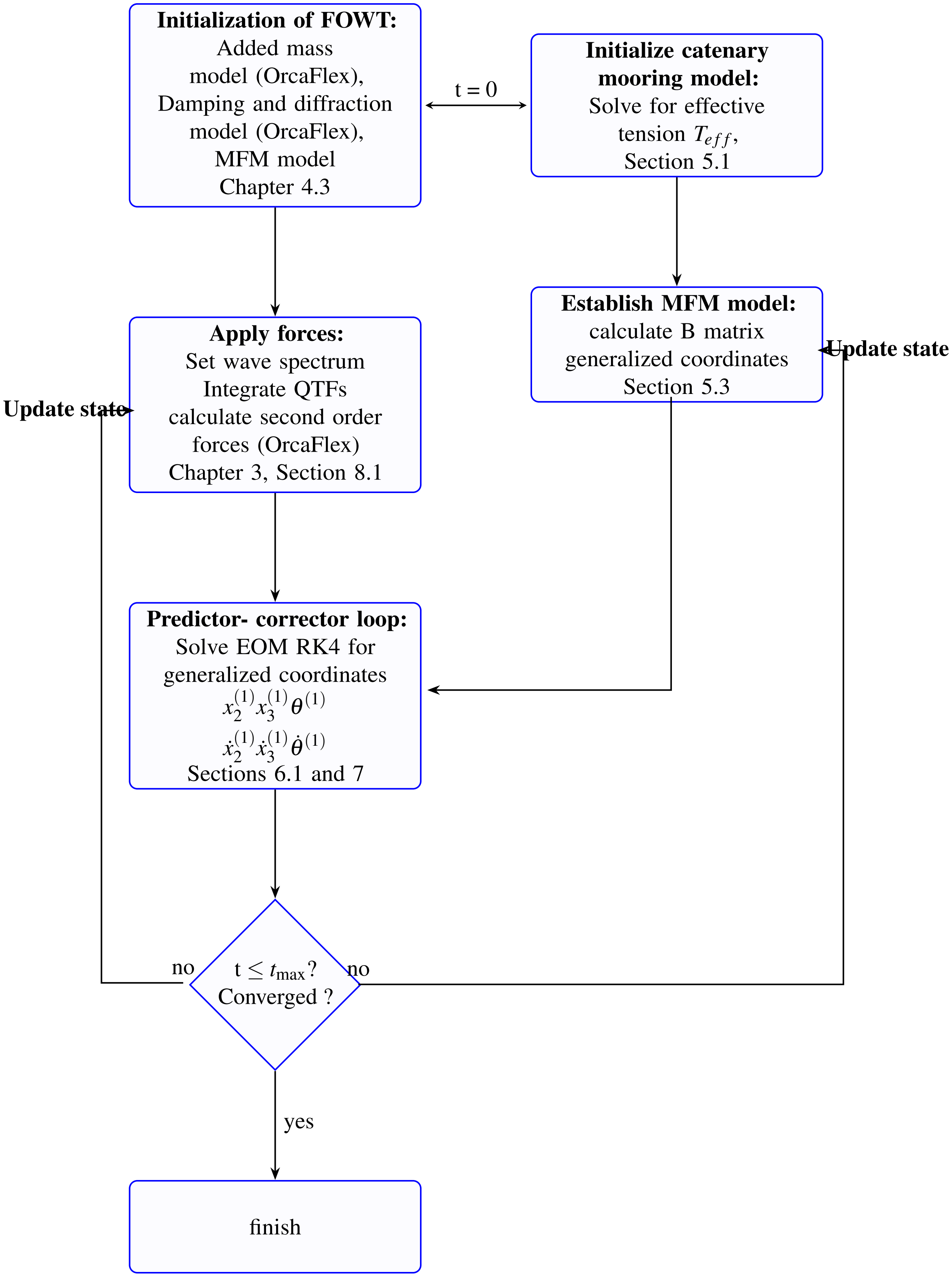

Initialization of FOWT:
Added mass model (OrcaFlex),
Damping and diffraction model (OrcaFlex),
MFM model
Chapter 4.3
t = 0
Initialize catenary mooring model:
Solve for effective tension T_{eff},
Section 5.1
Establish MFM model:
calculate B matrix
generalized coordinates
Section 5.3
Update state
Apply forces:
Set wave spectrum
Integrate QTFs
calculate second order forces (OrcaFlex)
Chapter 3, Section 8.1
Update state
Predictor- corrector loop:
Solve EOM RK4 for generalized coordinates
$x_2^{(1)} x_3^{(1)} \theta^{(1)}$
$\dot{x}_2^{(1)} \dot{x}_3^{(1)} \dot{\theta}^{(1)}$
Sections 6.1 and 7
no
$t \leq t_{\max}$?
Converged ?
no
yes
finish

Figure 7.2: Flowchart of the code for coupled analysis in MATLAB.

8 Results

This section presents the simulation results in MATLAB, obtained for different combinations of the model under irregular sea state. The aim is to examine how developed MFM-based mooring system and calculated second order forces affect the overall dynamic response of the FOWT and how the hydrodynamic behaviour changes under different loading condition and different configurations. Several configurations of the test are considered:

As seen from the Table 8.1 mooring line system is coupled to the spar-type platform only in configuration 3. This way the baseline performance of the dynamic of offshore turbine itself can be better investigated. Second order forces, presented in Section 3.1.1, are added in Configuration 2, enabling more accurate observation of motion of the FOWT in the low-frequency range (LF). Simulation is performed in time-domain and the obtained data are post processed in frequency domain, using method presented in Section 7.2. Due to a broad scope of the work and extensive computational tasks in this project, time required for more prolonged simulation shortened to 500 seconds [43]. While the selected time step is set to $\Delta t = 0.05$ s in accordance to guidelines recommended by NREL and to avoid singularities in matrix operations.

Configuration	**Description**
1	1st order wave excitation forces without coupled mooring system
2	1st and 2d order wave excitation forces without coupled mooring system
3	1st and 2d order wave excitation forces with coupled mooring system

Table 8.1: Summary of Model Configurations

Scale of	Wind Speed	Open Ocean Areas (Bretschneider)			North Sea Areas (JONSWAP)			
Beaufort	at 19.5 m above sea	$H_{1/3}$(m)	T_1	T_2	$H_{1/3}$(m)	T_1	T_2	γ
1	2.0	1.10	5.80	5.35	0.50	3.50	3.25	3.3
2	5.0	1.20	5.90	5.45	0.65	3.80	3.55	3.3
3	8.5	1.40	6.00	5.55	0.80	4.20	3.90	3.3
4	13.5	1.70	6.10	5.60	1.10	4.60	4.30	3.3
5	19.0	2.15	6.50	6.00	1.65	5.10	4.75	3.3
6	24.5	2.90	7.20	6.65	2.50	5.70	5.30	3.3
7	30.5	3.75	7.80	7.20	3.60	6.70	6.25	3.3
8	37.0	4.90	8.40	7.75	4.85	7.90	7.35	3.3
9	44.0	6.10	9.00	8.30	6.10	8.80	8.20	3.3
10	51.5	7.45	9.60	8.80	7.45	9.50	8.85	3.3
11	59.5	8.70	10.10	9.30	8.70	10.00	9.30	3.3
12	> 64.0	10.25	10.50	9.65	10.25	10.50	9.80	3.3

Table 8.2: Wave Spectrum Parameter Estimates [5]

8.1 Wave Spectrum Model

To simulate response of the offshore platform more realistically, two irregular wave spectra are chosen based on environmental conditions (mild, moderate and severe) presented in Table 8.2 and Figure 8.1.

The first is for a *developing sea*, JONSWAP spectrum, which captures the impact of wind retained for a shorter duration or fetch on it, applicable in cases waves are still maturing. In such seas, there typically is higher energy at the peak frequency due to clustering around more recent winds while lesser energy toward the sides of the local spectrums. In the case when peak-enhancement factor $\gamma = 1$, the JONSWAP spectrum simplifies to the Pierson-Moskowitz spectrum, since JONSWAP is an extension of the PM spectrum with additional parameters which accounts for wind-generated sea state.

The next is for *fully developed sea state*, Pierson-Moskowitz spectrum, assuming that the wave field is homogeneous (uniform over an area), and isotropic (same in all directions).

Peak wave period , T_P is calculated with mathematical relation $Tp = 1.199T_1$, where T_1 is taken from Table 8.2, as given by Johan Journee and William Massie. [5]. To identify potential resonance, eigenfrequencies from Tabel 8.3 of the offshore turbine are plotted in the same Figure 8.1. Since they are at some safe distance away from the peak of the wave spectrum, the wave energy is distributed on other frequencies, and the resonance should not happen. P-M wave spectrum, shown in the figure, 8.1 has the peak period at 11.39 s and a low gamma value, $\gamma = 1$. A low peak peak-enhancement factor $\gamma = 1$, results in a narrower spectral curve and a greater

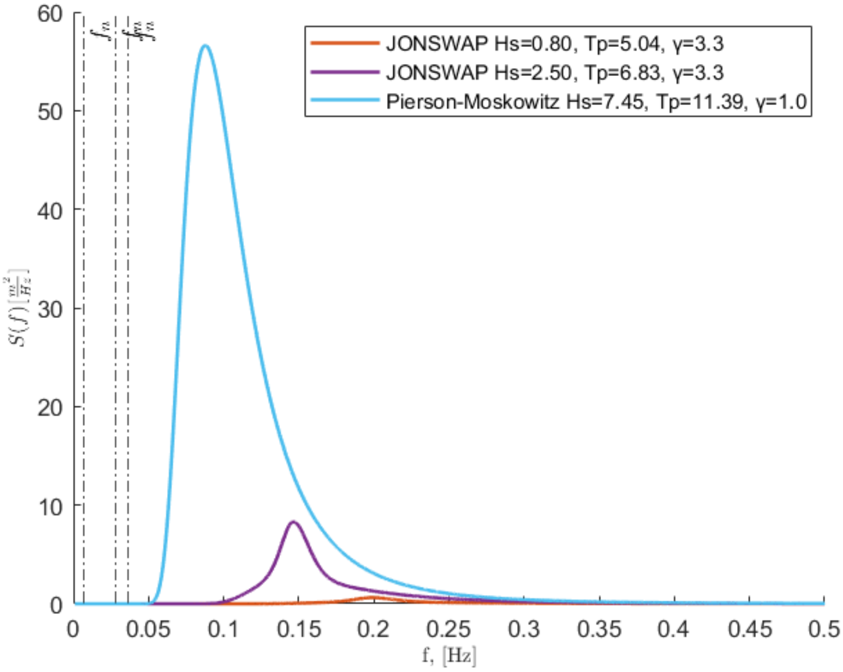


Figure 8.1: Theoretical wave spectra used in the simulation. f_n is eigenfrequencies given in a Tabel 8.3.

proportion of the energy is concentrated in the low frequencies. The spectrum is steep and thus it is expected that movements in low frequencies range (LF) excited by the second-order forces will be noticeable.

DOF	T_0 [s]	f_n [Hz]	ω_n [rad/s]
Surge, X	149.3	0.0067	0.042
Heave, Z	27.7	0.0361	0.227
Pitch, Θ	35.7	0.0280	0.176

Table 8.3: Spar-buoy undamped eigenperiods, T_0, and eigenfrequencies in Hz and rad/s. [1]

One should notice, that FOWT is assumed to be in a "parked" position, intentionally halted and secured. Hydrodynamic forces are examined independently from the influence of wind forces, aerodynamics is not concerned in this project.

8.1.1 Load Cases

Based on presented wave spectrum several load cases are defined according to different sea states, presented in the Table 8.4 below: Frequency limits presented in Table 8.5 are determined utilizing the symmetry around the peak frequency of the related wave spectra, by including an octave above and below $1/T_p$ in the interval $\frac{1/T_p}{2}$ to $2/T_p$.

Sea State	Load Case	Spectrum	H_s [m]	T_p [s]	$1/T_p$ [s^{-1}]
Mild (Mild Breeze)	O1	JONSWAP, $\gamma = 3.3$	0.80	5.04	0.198
Moderate (Strong Breeze)	M1	JONSWAP, $\gamma = 3.3$	2.50	6.83	0.146
Severe (Storm)	S1	Pierson Moskowitz, $\gamma = 1$	7.45	11.39	0.088

Table 8.4: Load cases specification

Load Case	Low-Frequency Range [Hz]	Wave-Frequency Range [Hz]
O1	0.005 to 0.098	0.099 to 0.396
M1	0.005 to 0.072	0.073 to 0.292
S1	0.005 to 0.043	0.044 to 0.176

Table 8.5: Frequency Limits of Power Spectral Density for all cases.

8.1.2 QTF Description

QTF have been plotted in MATLAB, using Eqn. 3.11 as presented in Section3.1.1. The results for DOF1, 3, and 5 in Figures 13a, 13b and 13c presented in Appendix A, show sharp resonant peaks at specific frequency pairs (f_m, f_n), which imply possible critical operational conditions. The identified frequencies as seen from contour and surface plots are defined by following frequency pairs:

- Surge: max. 2399.41 $\frac{kN}{m^2}$ at ($f_m = 0.03$ Hz , $f_n = 0.18$ Hz)
- Heave: max. 374.27.41 $\frac{kN}{m^2}$ at ($f_m = 0.03$ Hz , $f_n = 0.15$ Hz)
- Pitch: max. 89588.31 $\frac{kNm}{m^2}$ at ($f_m = 0.03$ Hz , $f_n = 0.03$ Hz)

The identified critical frequencies are situated in LF -range and it is expected noticable dynamic response in this region. It is observed that QTF potential force is equal zero at $\Delta f = 0$. Threfore the main diagonal, drawn as a red diagonal line, is defined by mean drift force, where equality between wave frequencies, f_m = f_n is observed.

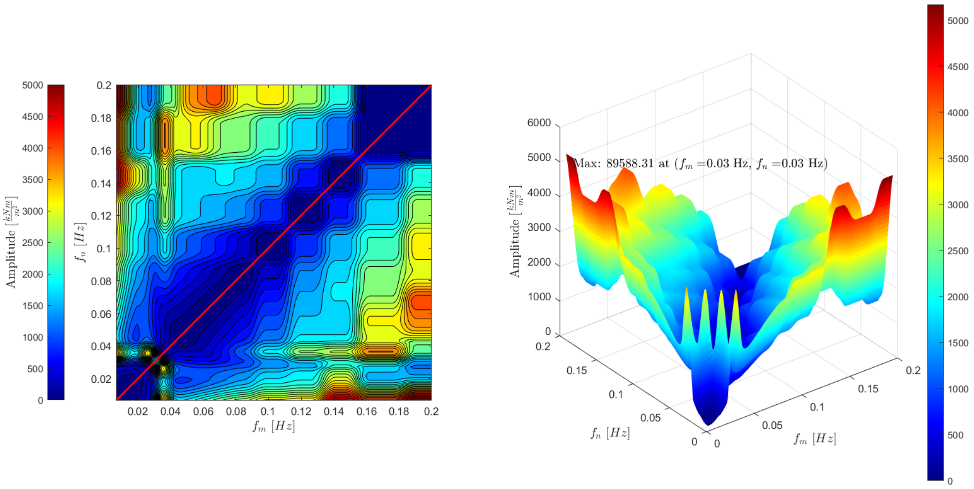


Figure 8.2: Contour plot (left) and surface plot (right) of QTF Moment for DOF5.

8.2 Dynamic Response

As seen from Figures 20 and 21, there is noticable second order induced motion in low-frequency range (LF) for DOF3 and 5. All model configurations seems to produce similar result, except for DOF1, in surge direction, where the biggest excitation in LF -range is mooring line induced. Same result is observed for load case M1 in moderate sea state and S1 in severe state. Though it is observed wave excited motion for DOF5 in wave frequency range (WF) in case S1 in severe state.

The results presented in Table 8.6 shows that the biggest displacement is in surge direction, DOF1 for case S1 in severe sea state. The same yields pitch rotation in DOF2, represented by $\theta^{(1)}$, depicting how much the platform tilts forward and backward from its equilibrium position.

Load Cases	$x_1^{(1)}$ $[\frac{m^2}{Hz}]$	$x_3^{(1)}$ $[\frac{m^2}{Hz}]$	$\theta^{(1)}$ $[\frac{deg^2}{Hz}]$
O1	1126.2	9684.8	5.2×10^{-7}
M1	752.1	9621.8	4.0×10^{-5}
S1	48331.1	9188.9	0.061

Table 8.6: Summary of maximum motion of the platform in Degrees of Freedom (DOFs) 1, 3, and 5.

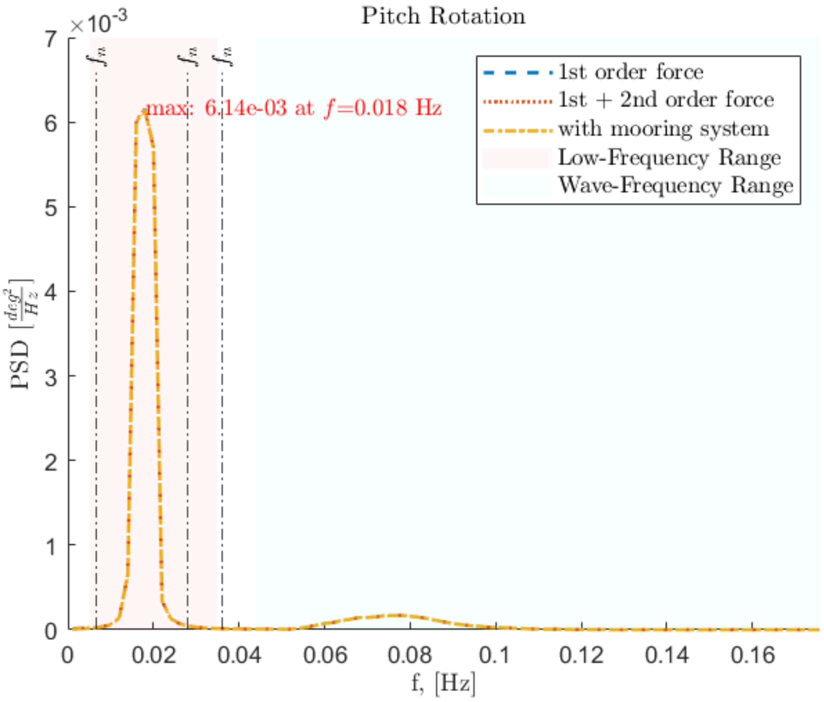


Figure 8.3: Rotation in pitch, $\theta^{(1)}$ for load case S1 presented as PSD in frequency domain.

8.3 Force Excitation

The transient behaviour of the offshore platform can be evaluated based on data collected through simulation presented in Tables 1, 2 and 3 for each case respectively. According to values calculated with Eqn. 8.1, first order excitation force, has highest max value and standard deviation σ compared to other forces in load cases M1 and S1 as well as variance, while total force F^* takes leadership in load case O1, in mild sea state and also shows higher variability in force magnitude over time, possibly due to wave interactions and changing directions. Calculated percentage increase in variance from LCO1 to LCS1 between first order excitation force and total force F^* is equal to approximately 39.035% and increase in standard deviation is estimated to 1.878.25%.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i-1}^{N}(x_i-\mu)^2} \tag{8.1}$$

Here σ is standard deviation, μ is the mean, and $N-1$ represents Bessel's correction.

Figures 8.5 and 8.8 show the transformation of excitation forces that acts on the platform. The total force F^* dominates in load case O1, at identified critical frequency $f = 0.03$, as discussed in the Section 8.1.2. In addition this effect is amplified by presense of hiv and pitch eigenfrequencies, in the low- frequency range.

So the distinct peak occurs at $f = 0.034$ Hz, as seen from Figure 8.5. The force $F*$ migrates to wave frequency region, where first order excitation force takes over. And now, the highest amplitude that dominates wave frequency range belongs to the first order excitation force, as demonstrated in Figure 8.8 .

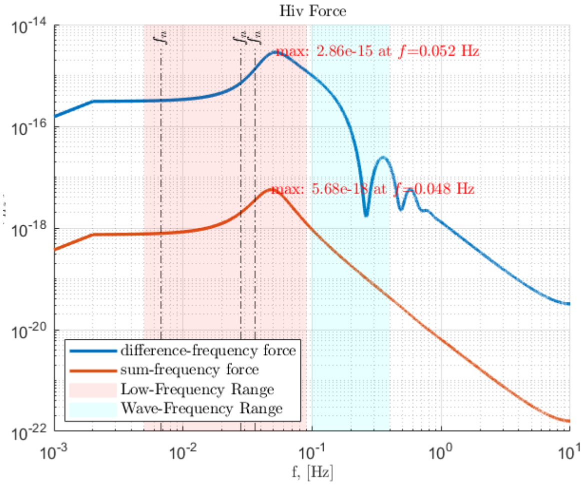


Figure 8.4

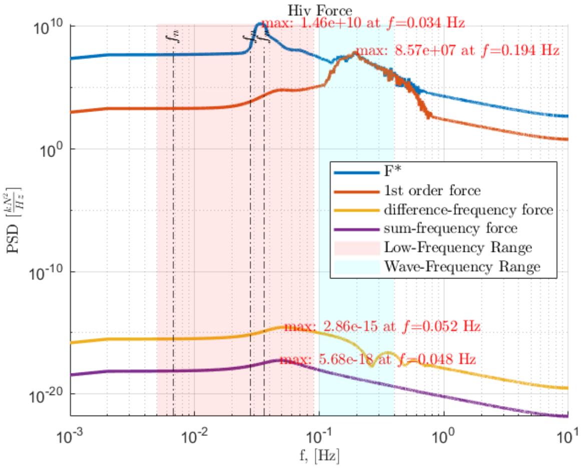


Figure 8.5

Figure 8.6: All forces compared to difference frequency force in hiv for load case O1 presented as PSD in frequency domain

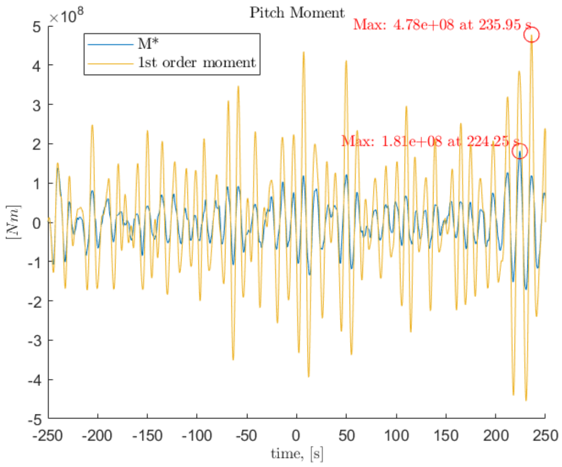


Figure 8.7

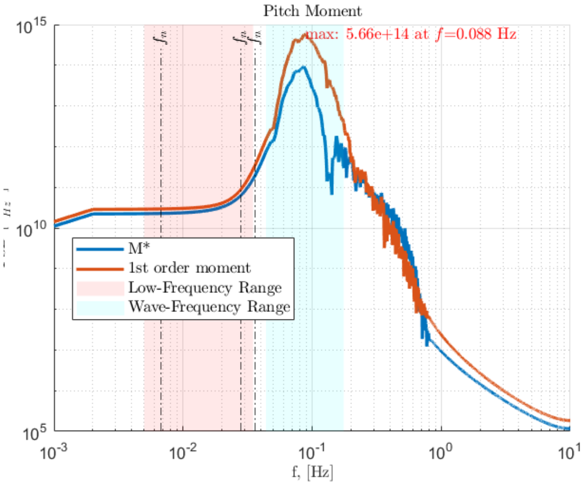


Figure 8.8

Figure 8.9: Total Moment* compared to first order excitation moment in pitch for load case S1 presented as time-series (left) and as PSD in frequency domain (right).

9 Conclusions

An innovative approach based on Moving Frame Method for effectively modelling and analysing FOWT of spar buoy type and catenary mooring system, designed as multi-segmented entity with adjustable number of segments, has been developed and employed into analysis. The applied modelling principle is novel, because the most of acknowledgement MFM have gotten in robotic field, while it has not been used for modelling of mooring system before.

The established mooring model is based on Jonkmans proposal of quasi-static catenary mooring module, which has been greatly modified and adopted to fit the projects needs, as presented in section 5.1. The catenary model is used to initialize an MFM-based multi-segmented mooring line through partial derivation, dx/ds and dz/ds of Eqns. 5.5 and 5.6, which defines suspended mooring line.

A multi-segmented mooring line has been constructed according to the principles of MFM, extensively using *relative* and *absolute* frames in the matrix form. The generalized coordinates, q, have been defined for both spar buoy and mooring line, through $B-$matrix, calculated in C++ based Julia.jl, as presented in the Sections 4.8 and 5.3

The uniqueness of developed MFM-based multi-segmented model of mooring line lies in generic nature of constructed model, that can be extended up to α-segments, allowing as well initialization of the segments to be defined and changed according to desired line type (for example taut), component type, (for example lying on the seabed), and by material the components are made of, for by usual for deep water chain$-$ wire/line $-$chain components. Though, to simplify the task of the project, suspended catenary line with uniform material properties of high modulus polyethylene (HPME) is being used.

Hydrodynamic forces incorporated into the system, have been extended by second order forces, utilizing second transfer function (QTF) from OrcaFlex, enabling representation of the motion in the lower frequency range, as shown in Section 3.1.1, Chapter 8 and overview of QTFs is given in Appendix A.

EOM is derived via Principles of Virtual Work, based on Eqns. 6.23 to 6.25, as shown in Section 6.1 and, in the form of differential equations, solved through 4th order Runge-Kutta method, following method described in the Section 7.1.

Results obtained from the simulation in time domain, shows that total force F^* is mostly characterized by the contribution of the first order excitation force in wave frequency range (WF), as described in Section 8.3. Though it is observed noticeable dynamic response in low-frequency range, from 0.03 to 0.20 Hz, as shown in Section 8.2. Based on analyse of QTFs performed in Sections 8.1.2 and results obtained for second order forces from simulation, shown in Section 8.3 it is assumed that dynamic response in low -frequency range is induced by second order excitation forces.

The findings of this project provides additional evidence and confirms results obtained by specialists at the National Renewable Energy Laboratory (NREL) [19]. Whether developed system has contribution to this motion is still not obvious and remain a subject for a further research.

9.1 Futher Work

To investigate the reason why hydrodynamic response is observable mostly in low frequency range, the overall dynamics of the offshore turbine should be investigated more thoroughly, by performing several tests in different sea states. In addition, to overcome transient time, the simulation length should be extended. There could be a need to include additional DOFs and extend the minimum set of the generalized coordinates of both developed mooring system and of the FOWT itself. Work has already started and will continue as the inclusion of the Moving Frame Method cannot be underestimated in hydrodynamic computations.

References

[1] Aminda M. T. Ripe and David R. Lande-Sudall. A comparison of extreme mooring loads and response of a spar-buoy wind turbine using conditional waves. *Journal of Physics: Conference Series*, 2626(1):012062, oct 2023.

[2] U.M. Ba, University of Newcastle upon Tyne. School of Marine Science, and Technology. *Analysis of Mooring and Steel Catenary Risers System in Ultra Deepwater*. University of Newcastle upon Tyne, 2012.

[3] J. M. Jonkman. Dynamics modeling and loads analysis of an offshore floating wind turbine. 12 2007.

[4] Mohammed Khair Al-Solihat and Meyer Nahon. Stiffness of Slack and Taut Moorings. *Ships and Offshore Structures*, 11(8):890–904, 2016.

[5] Johan M.J. Journee and William W. Massie. *Offshore Hydromechanics*. Delft University of Technology, Delft, Netherlands, 2001.

[6] *Second-Order Difference-Frequency Loads on FPSOs by Full QTF and Relevant Approximations*, volume Volume 1: Offshore Technology of *International Conference on Offshore Mechanics and Arctic Engineering*, 08 2020.

[7] International Renewable Energy Agency (IRENA). Renewable power generation costs in 2020, 2020. Retrieved from `https://www.irena.org/publications/2021/Jun/Renewable-Power-Costs-in-2020`.

[8] Global Wind Energy Council (GWEC). Global offshore wind report 2021, 2021. Retrieved from `https://gwec.net/global-offshore-wind-report-2021`.

[9] Intergovernmental Panel on Climate Change (IPCC). Climate change 2021: The physical science basis. *Contribution of Working Group I to the Sixth Assessment Report of the*

Intergovernmental Panel on Climate Change, 2021. Retrieved from `https://www.ipcc.ch/report/ar6/wg1/`.

[10] International Labour Organization (ILO). The renewable energy industry and job creation, 2020. Retrieved from `https://www.ilo.org/global/topics/green-jobs/publications/WCMS_596875/lang--en/index.htm`.

[11] United Nations Environment Programme (UNEP). Responsible consumption and production: A global challenge, 2021. Retrieved from `https://www.unep.org/explore-topics/resource-efficiency/what-we-do/sustainable-consumption-and-production-policies`.

[12] R. L. Huston. Multibody dynamics — modeling and analysis methods. *Applied Mechanics Reviews*, 44(3):109–117, 1991.

[13] Élie Cartan. *Exterior Differential Systems and Their Geometric Applications*. Springer-Verlag, 1971.

[14] Élie Cartan. *La Méthode de Repère Mobile, la Théorie des Groupes Continus, et les Espaces Généralisés*. Hermann, 1935.

[15] W.E. Cummins and United States. Navy. *The Impulse Response Functions and Ship Motions*. David Taylor Model Basin. United States Department of the Navy, David Taylor Model Basin, 1962.

[16] O. Faltinsen. *Sea loads on ships and offshore structures*, volume 1. Cambridge university press, 1993.

[17] Orcina. Theory: Second-order equations. *OrcaWave Documentation*, 2023. Available online: `https://www.orcina.com/webhelp/OrcaWave/Content/html/Theory,Second-orderequations.htm`.

[18] Orcina. Data: Qtfs. *OrcaWave Documentation*, 2023. Available online: `https://www.orcina.com/webhelp/OrcaWave/Content/html/Data,QTFs.htm`.

[19] A. N. Robertson, S. Gueydon, and E. Bachynski et al. Oc6 phase i: Investigating the underprediction of low-frequency hydrodynamic loads and responses of a floating wind turbine. *Journal of Physics: Conference Series*, 1618(3):032033, sep 2020.

[20] CD Weggel and JM Roesset. Dynamic behavior of spar platforms in waves. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering*. ASME, 1996.

[21] H. Ormberg et al. Enhanced modeling techniques for mooring line dynamics. *Journal of Offshore Mechanics and Arctic Engineering*, 140(2):021901, 2018.

[22] D. Kim and J. Park. Application of particle dynamics in mooring line simulations. *Ocean Engineering*, 175:90–101, 2019.

[23] W. Bai and B. Teng. Comprehensive hydrodynamic modeling of mooring systems. *Marine Structures*, 72:102759, 2020.

[24] Michael S. Longuet-Higgins and R.W. Stewart. Radiation Stresses in Water Waves: A Physical Discussion, with Applications. *Deep Sea Research and Oceanographic Abstracts*, 11:529–562, 1964.

[25] E. Cartan. *On Manifolds with an Affine Connection and the Theory of General Relativity*. Monographs and textbooks in physical science. Bibliopolis, 1986.

[26] T. Frankel. *The Geometry of Physics: An Introduction*. Cambridge University Press, 3 edition, 2011.

[27] H. Murakami. A Moving Frame Method for Multi-Body Dynamics. volume Volume 4A: Dynamics, Vibration and Control of *ASME International Mechanical Engineering Congress and Exposition*, page V04AT04A079, 11 2013.

[28] T. J. Impelluso. The moving frame method in dynamics: Reforming a curriculum and assessment. *International Journal of Mechanical Engineering Education*, 46(2):158–191, 2018.

[29] A. H. Pevrot and A. M. Goulois. Analysis of cable structures. *Computers & Structures*, 10(5):805–813, 1979.

[30] William Whewell. *The Mechanics of Engineering*. J. and J. J. Deighton, Cambridge, UK, 1837.

[31] Western Norway University of Applied Sciences. *MAS313 Advanced Dynamics*, Autumn 2023. Professor T.J.Impelluso.

[32] DNVGL-OS-E301: Position Mooring, 2018.

[33] DNV-ST-0119: Floating Wind Turbine Structures, 2021.

[34] DNV-RP-H103: Modelling and Analysis of Marine Operations, 2011.

[35] Mooring Equipment Guidelines (MEG4), 2018.

[36] DIN EN ISO 1141:Fibre ropes – Polyester – 3-, 4-, 8- and 12-strand ropes, 2021. Fifth edition, Reference number ISO 1141:2021(E).

[37] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[38] Subrata K. Chakrabarti. *Offshore Structure Analysis, Inc.: Volume I.* Elsevier, Plainfield, Illinois, USA, 2005.

[39] DNV-RP-F205: Global Performance Analysis of Deepwater Floating Structures, October 2010. Recommended Practice.

[40] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* Wiley, 2021.

[41] C. A. Barker. Numerical methods for solving differential equations: The runge-kutta method, theoretical introduction, 2017. [Online; accessed 28-November-2023].

[42] Valentin Chabaud. Data analysis in experimental methods in marine hydrodynamics. Lecture notes for Experimental Methods in Marine Hydrodynamics, Lecture in week 36, 2015. Overview of data analysis methods used in experimental marine hydrodynamics with examples using Matlab. Lecture on behalf of Pr. Sverre Steen. Covers Chapter 10 in the Lecture Notes.

[43] Amy Robertson and Jason et al Jonkman. Simulation-Length Requirements in the Loads Analysis of Offshore Floating Wind Turbines. In *Proceedings of the ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering*, Nantes, France, June 2013. ASME. NREL/CP-5000-58153.

Appendices

The appendices contain additional material that backs up the main theory and methodology presented in this thesis. They aim to provide the reader with a more thorough comprehension of the methods used and the data supporting the research results. Overview of appendices:

Appendix A: QTF for all DOF
For a detailed explanation, see Section Second-Order Force on page 8.

Appendix B:: Excitation Forces
The results obtained from simulation are provided for all cases in DOF1, 3, and 5. The forces that are listed there are the total force F^*, first order excitation force, sum-and difference frequency forces. These are plotted in one figure for comparison.

Appendix C:: Dynamic Response
In the same way as forces, the dynamic response of FOWT is presented in DOF1,3 and 5 for all cases. Model configurations are compared against each other.

Appendix D::MATLAB code
Detailed MATLAB code used to represent methodology of initialization of quasi-static model of catenary mooring line, according to procedure described in the Section 5.1.

Appendix E:: MATLAB code
In this appendix a set of different functions and scripts for post-processing of obtained data is presented. Fast Fourier Transformation (FFT) to Power Spectral Density function is among them. All scripts are annotated to help understand the computational steps and algorithms used in processing and analysing the dataset.

The main code with MFM -based developed mooring line together with Outer loop code and calculation of QTFs are to be found in separate .zip file, because of the size of the code itself. As it is too large and would interfere the consistency of the text of the thesis.

Appendix A

QTF

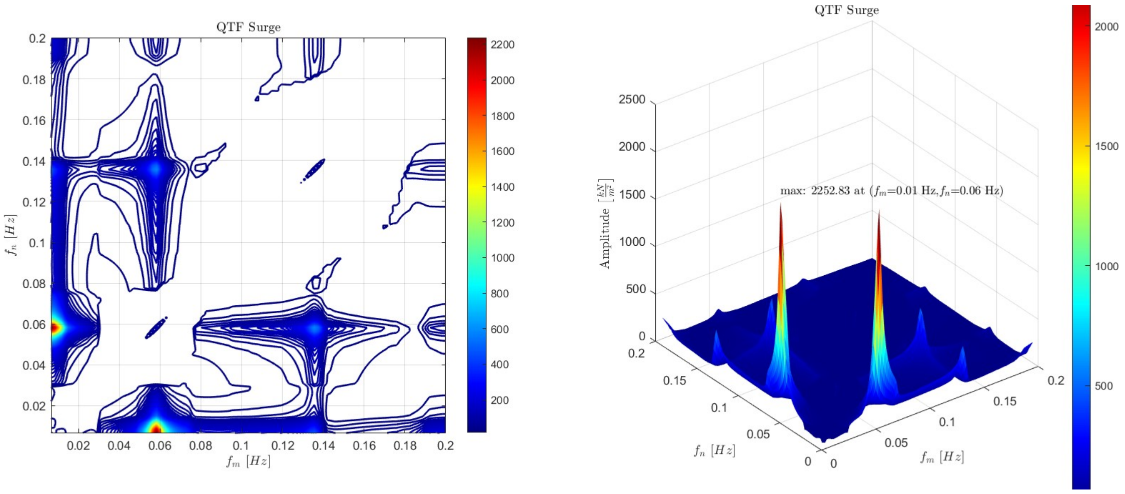


Figure 1: Amplitude of quadratic QTF force, surge.

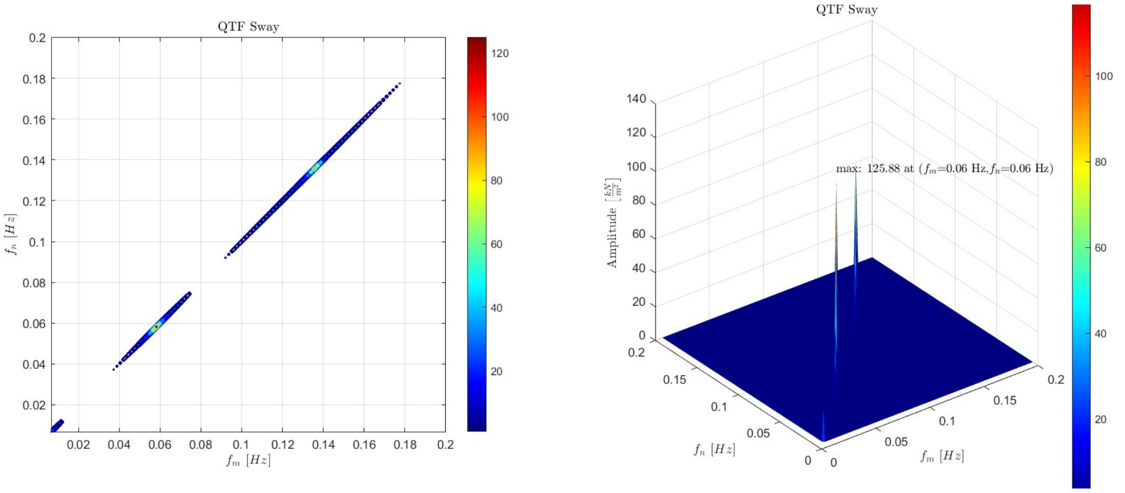


Figure 2: Amplitude of quadratic QTF force, sway.

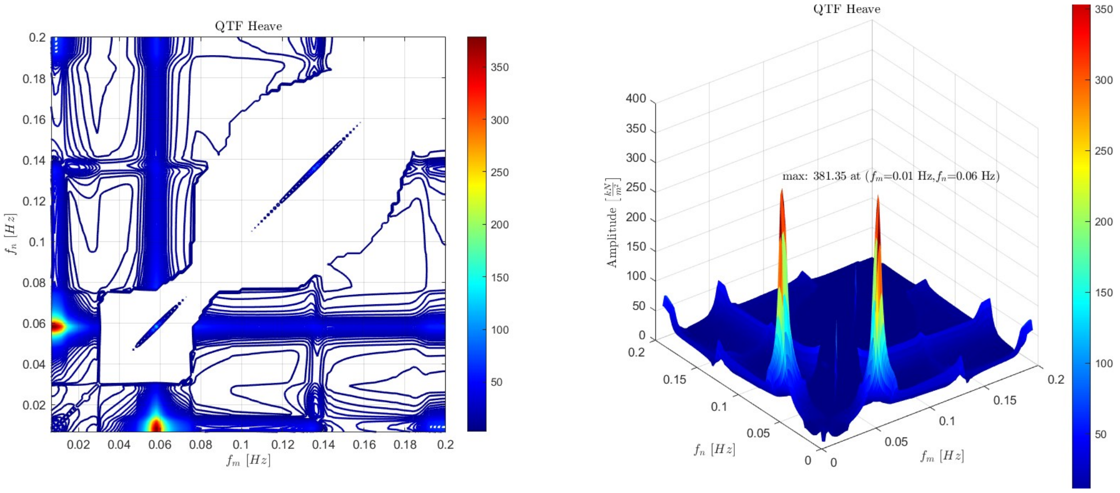


Figure 3: Amplitude of quadratic QTF force, heave.

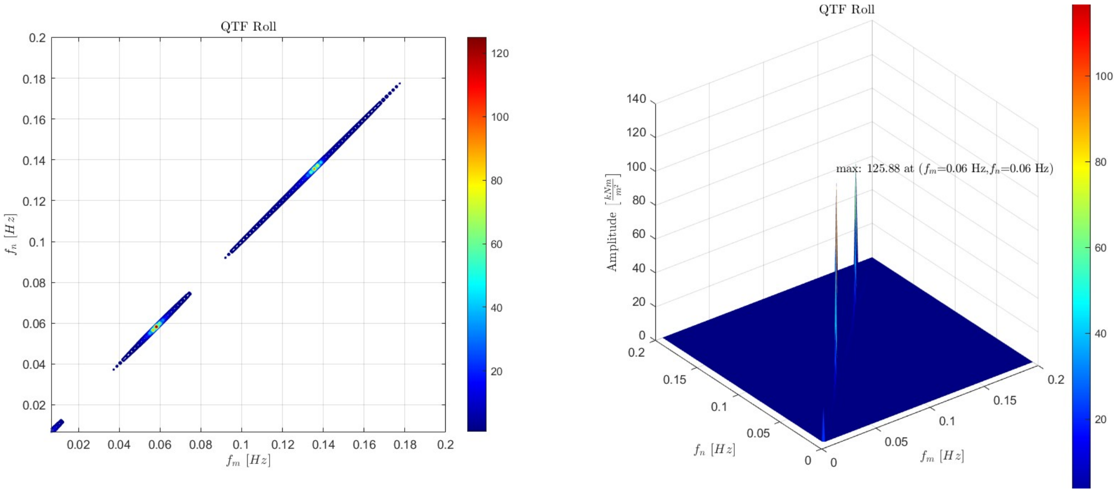


Figure 4: Amplitude of quadratic QTF moment, roll.

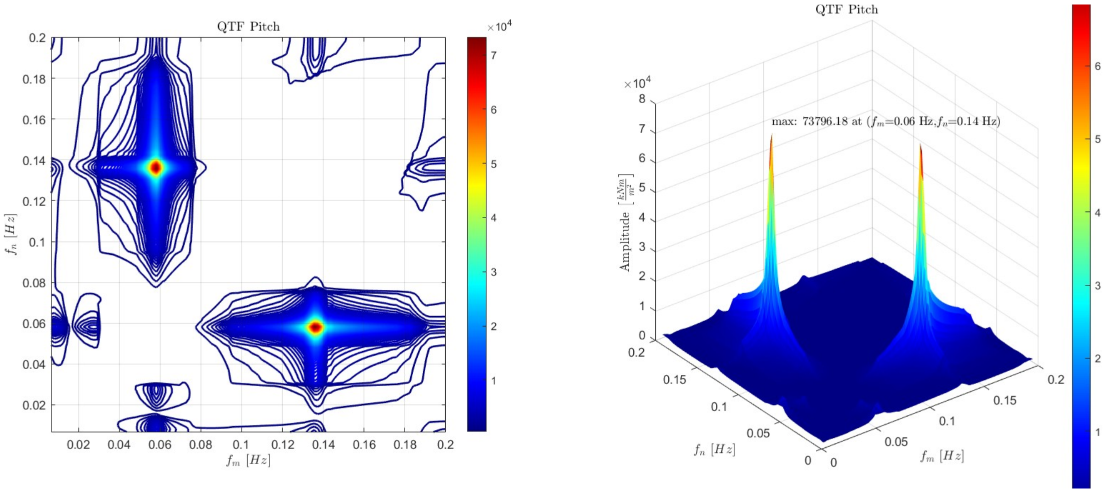


Figure 5: Amplitude of quadratic QTF moment, pitch.

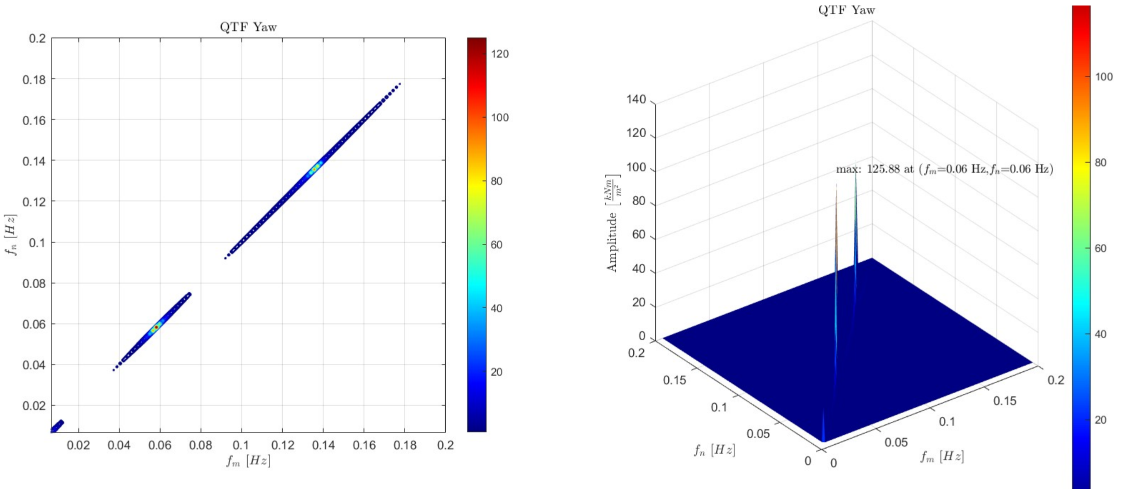


Figure 6: Amplitude of quadratic QTF moment, yaw

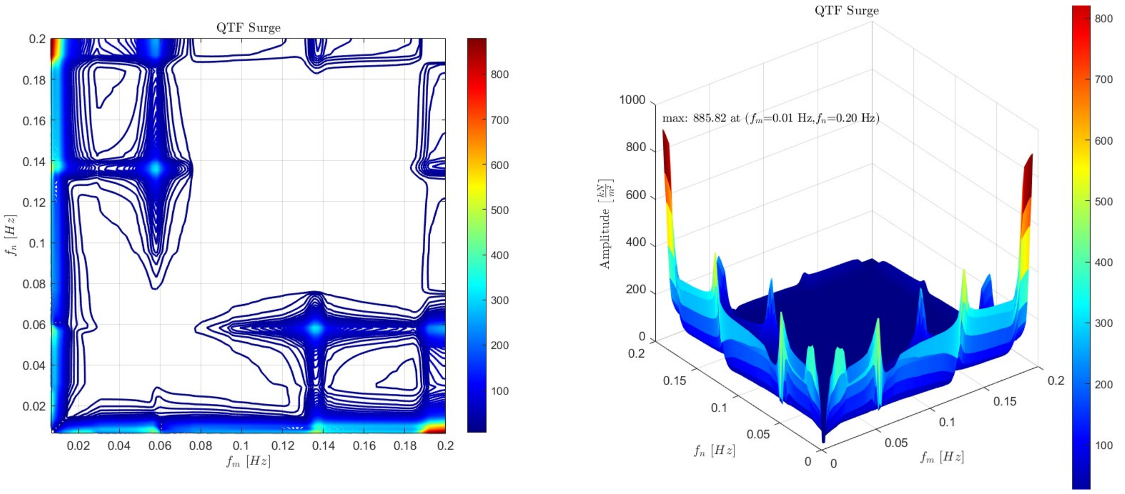


Figure 7: Amplitude of potential QTF force, surge.

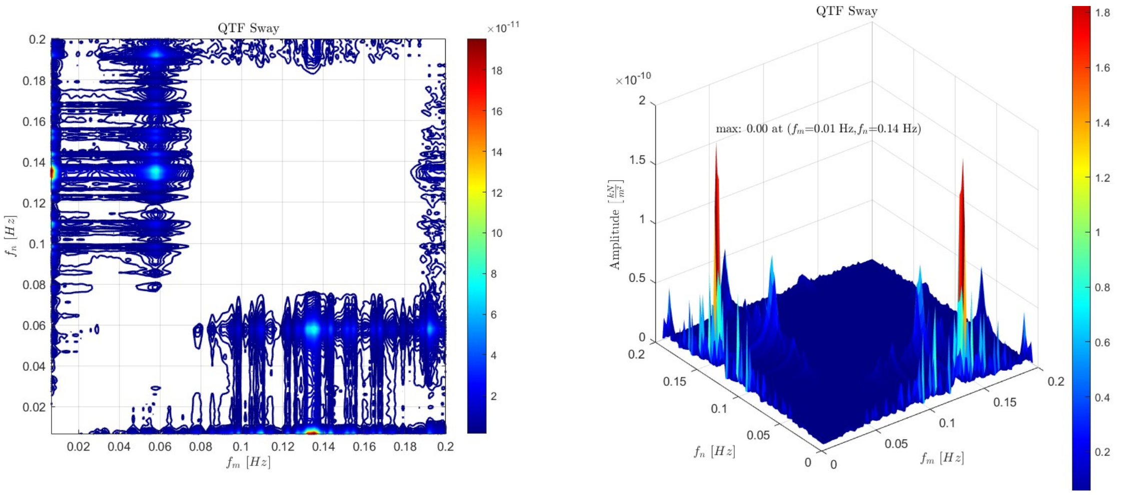


Figure 8: Amplitude of potential QTF force, sway.

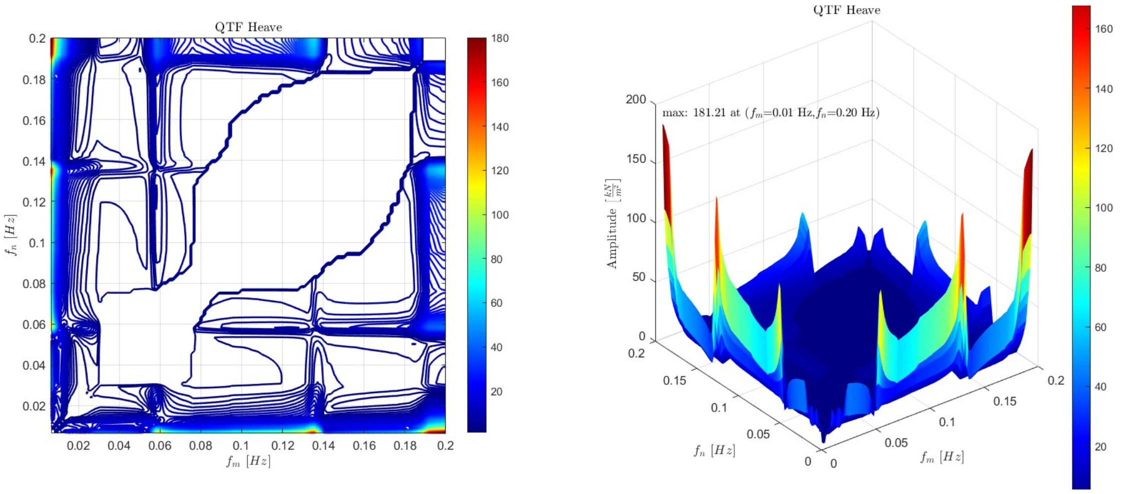


Figure 9: Amplitude of potential QTF force, heave.

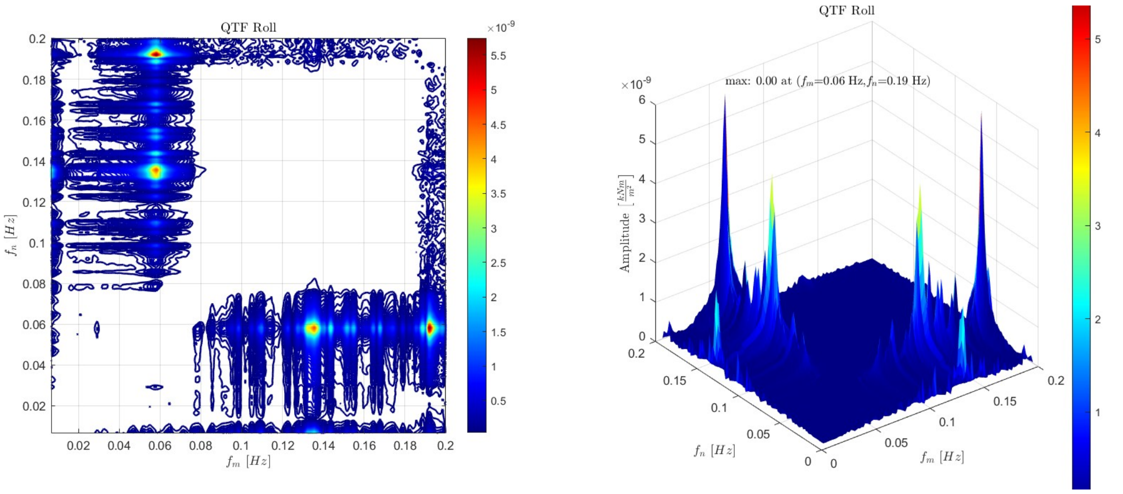


Figure 10: Amplitude of potential QTF moment, roll.

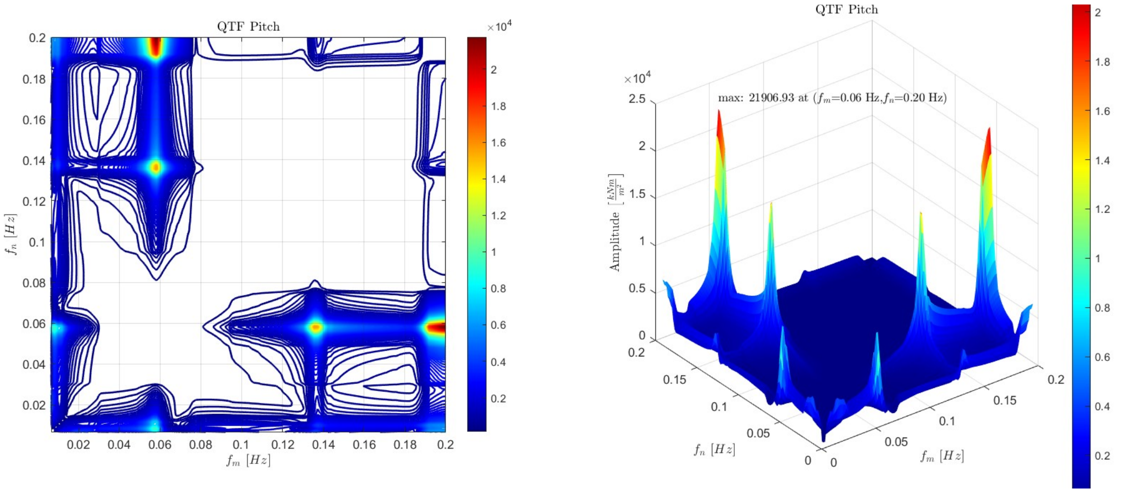


Figure 11: Amplitude of potential QTF moment, pitch.

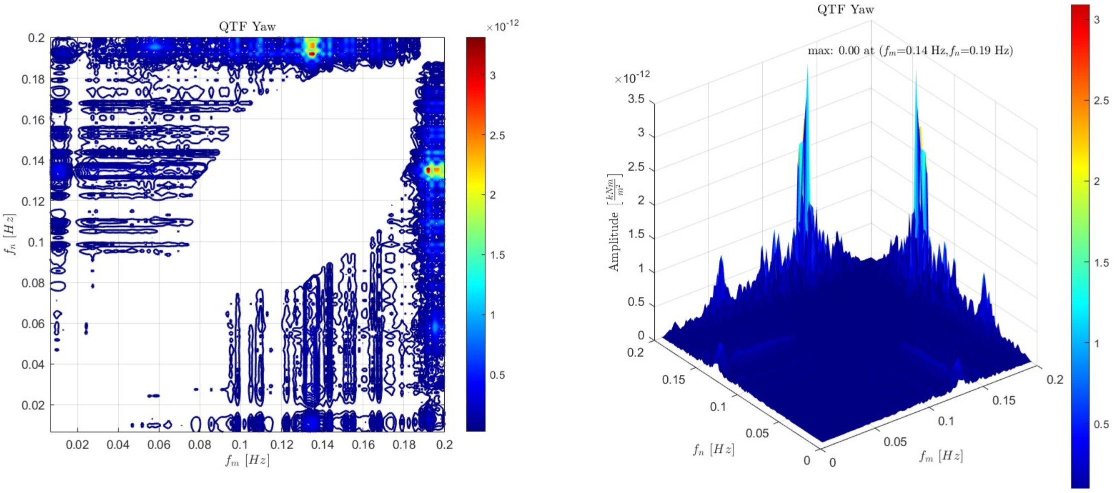
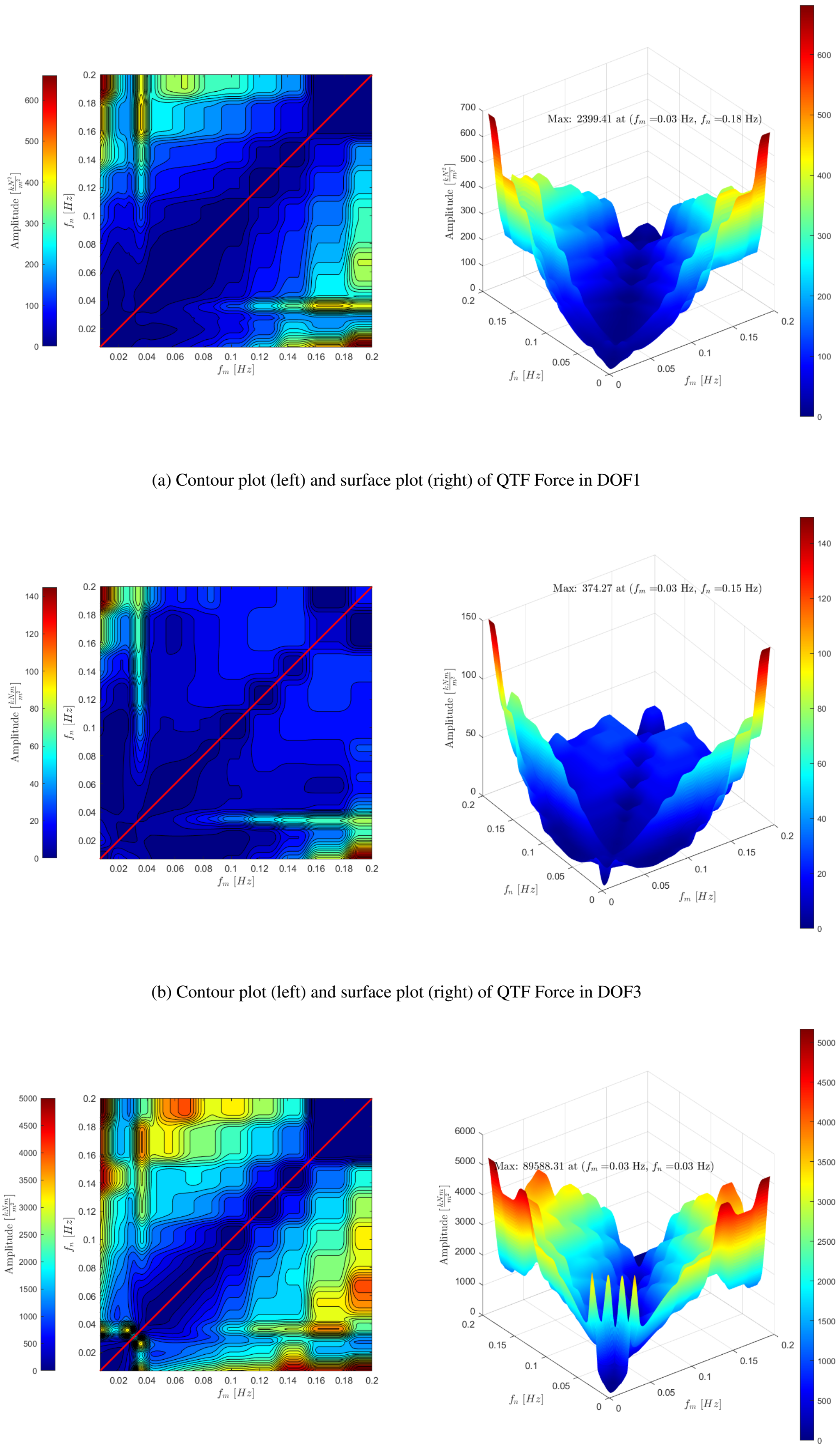


Figure 12: Amplitude of potential QTF moment, yaw



(a) Contour plot (left) and surface plot (right) of QTF Force in DOF1

(b) Contour plot (left) and surface plot (right) of QTF Force in DOF3

(c) Contour plot (left) and surface plot (right) of QTF Moment for DOF5

Figure 13: Contour plots of QTF Force for DOF1, 3, and QTF Moment for DOF 5. The red line represents mean drift force, $\Delta\omega = 0$.

Appendix B:

Excitation Forces

Force	Min	Max	σ	μ	σ^2
F_1	-1.0147×10^7	1.1220×10^7	2.0788×10^6	1.7320×10^3	4.3216×10^{12}
$F^{(2-)}$	-662.1226	3.6761×10^3	1.1791×10^3	2.3144×10^3	1.3904×10^6
$F^{(2+)}$	-825.0153	3.6761×10^3	1.1885×10^3	2.3105×10^3	1.4125×10^6
F^*	-1.1082×10^7	1.1213×10^7	4.3629×10^6	3.6157×10^4	1.9035×10^{13}

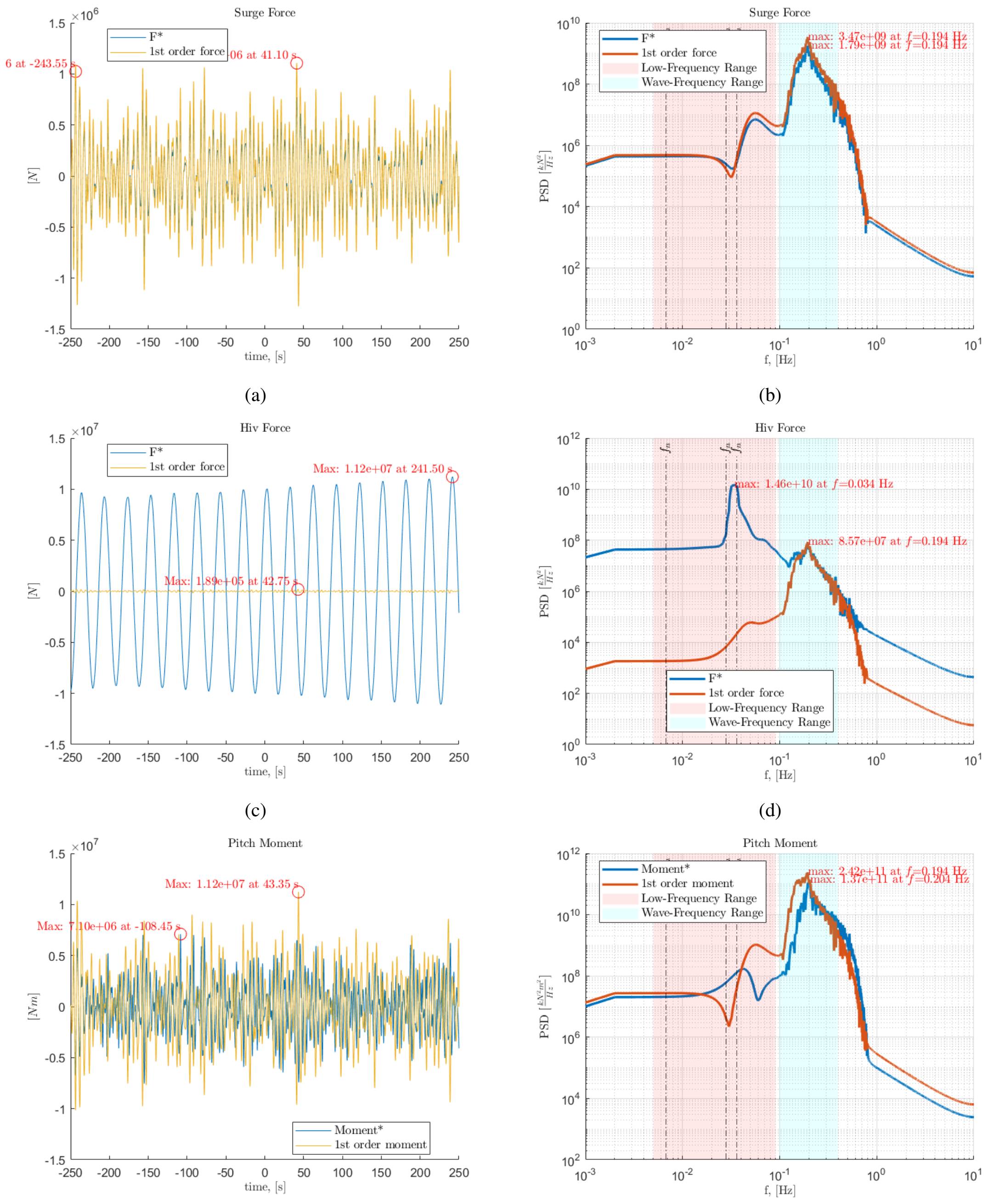
Table 1: Standard deviation, σ, variance, σ^2, mean, maximum, and minimum values obtained for Load Case O1.

Force	Min	Max	σ	μ	σ^2
F_1	-7.37×10^7	6.96×10^7	1.49×10^7	-1.5403×10^4	2.21×10^{14}
$F^{(2-)}$	-2.42×10^3	1.10×10^4	3.76×10^3	2.43×10^3	1.42×10^7
$F^{(2+)}$	-2.42×10^3	1.10×10^4	3.76×10^3	2.44×10^3	1.41×10^7
F^*	-2.75×10^7	3.61×10^7	5.63×10^6	3.17×10^4	3.23×10^{13}

Table 2: Standard deviation, σ, variance, σ^2, mean, maximum, and minimum values obtained for Load Case 2.

Force	Min	Max	σ	μ	σ^2
F_1	-4.5470×10^8	4.7758×10^8	8.6309×10^7	1.1557×10^5	7.4493×10^{15}
$F^{(2-)}$	-3.7823×10^3	1.7242×10^4	5.6217×10^3	1.0795×10^4	3.1604×10^7
$F^{(2+)}$	-3.7823×10^3	1.7242×10^4	5.5633×10^3	1.0833×10^4	3.0905×10^7
F^*	-1.7224×10^8	1.8097×10^8	3.1218×10^7	3.7248×10^4	9.7456×10^{14}

Table 3: Standard deviation, σ, variance, σ^2, mean, maximum, and minimum values obtained for Load Case 3.



(a)

(b)

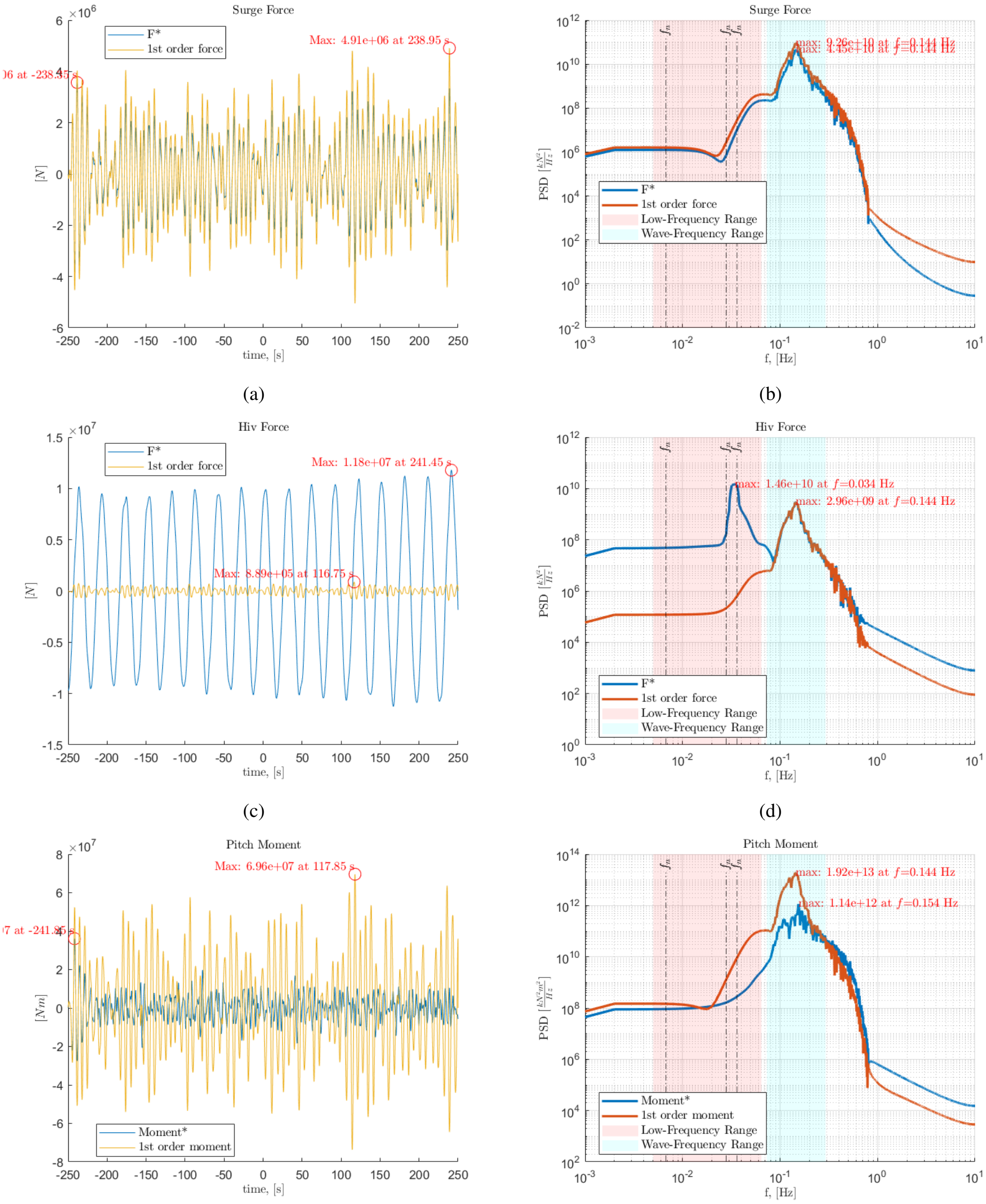
(c)

(d)

(e)

(f)

Figure 14: Total force F^* and Moment* compared to 1st order excitation force and moment in surge (a, b), hiv (c, d) and pitch (e, f), for load case O1 presented as time-series (left) and as PSD in frequency domain (right)



(a)

(b)

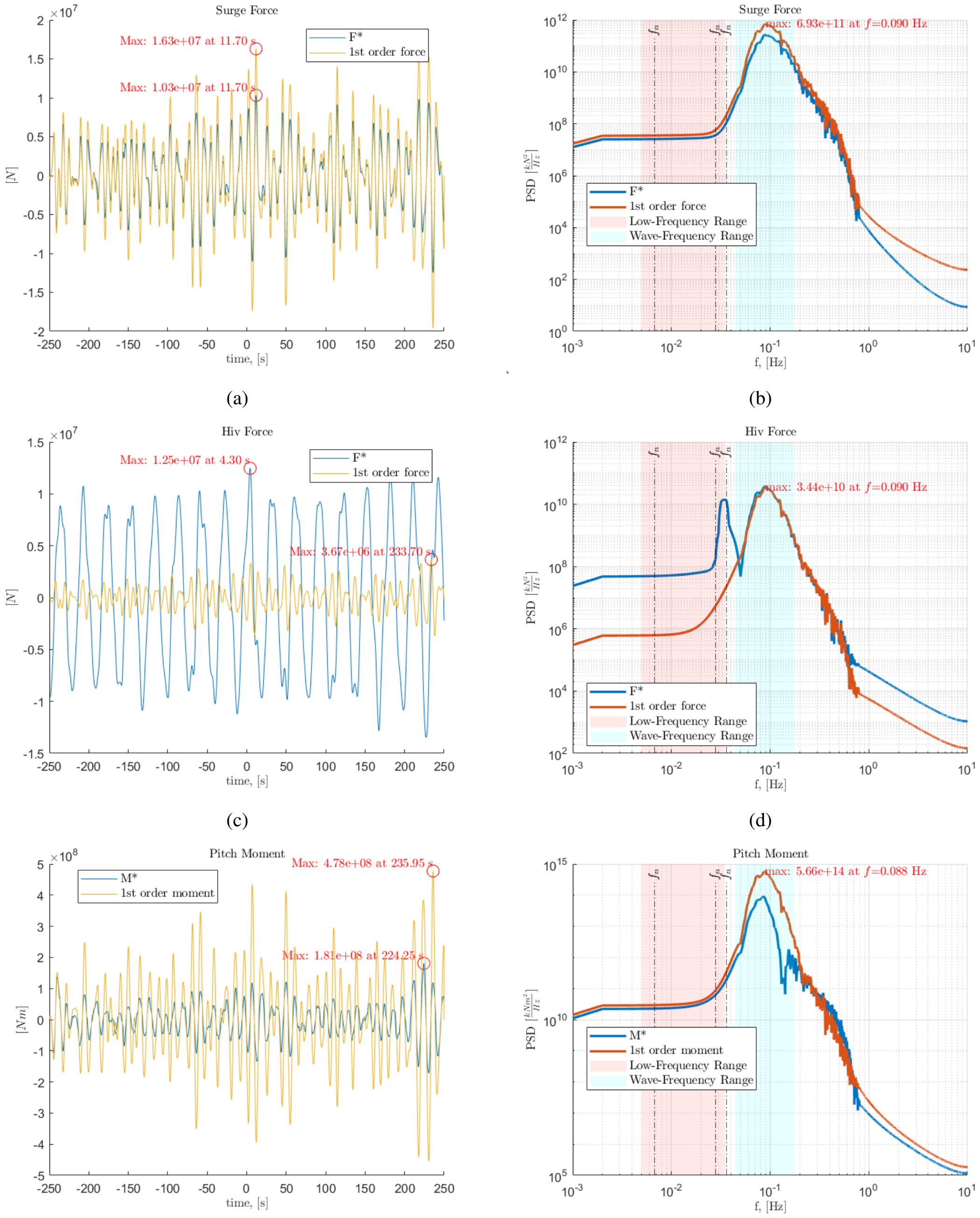
(c)

(d)

(e)

(f)

Figure 15: Total force F^* and Moment* compared to 1st order excitation force and moment in surge (a, b), hiv (c, d) and pitch (e, f), for load case M1 presented as time-series (left) and as PSD in frequency domain (right)



(a)

(b)

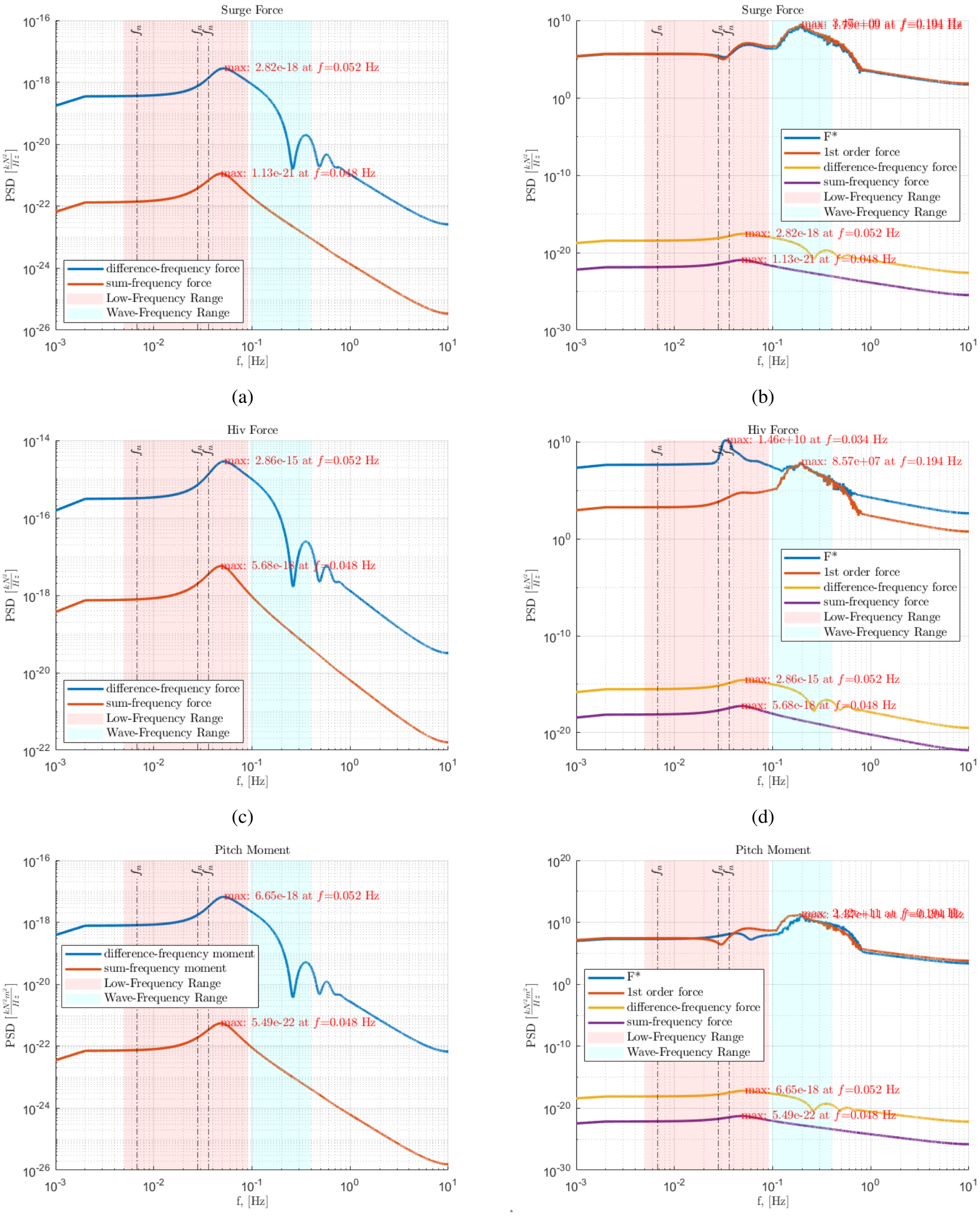
(c)

(d)

(e)

(f)

Figure 16: Total force F^* and Moment* compared to 1st order excitation force and moment in surge (a, b), hiv (c, d) and pitch (e, f), for load case S1 presented as time-series (left) and as PSD in frequency domain (right)



(a)

(b)

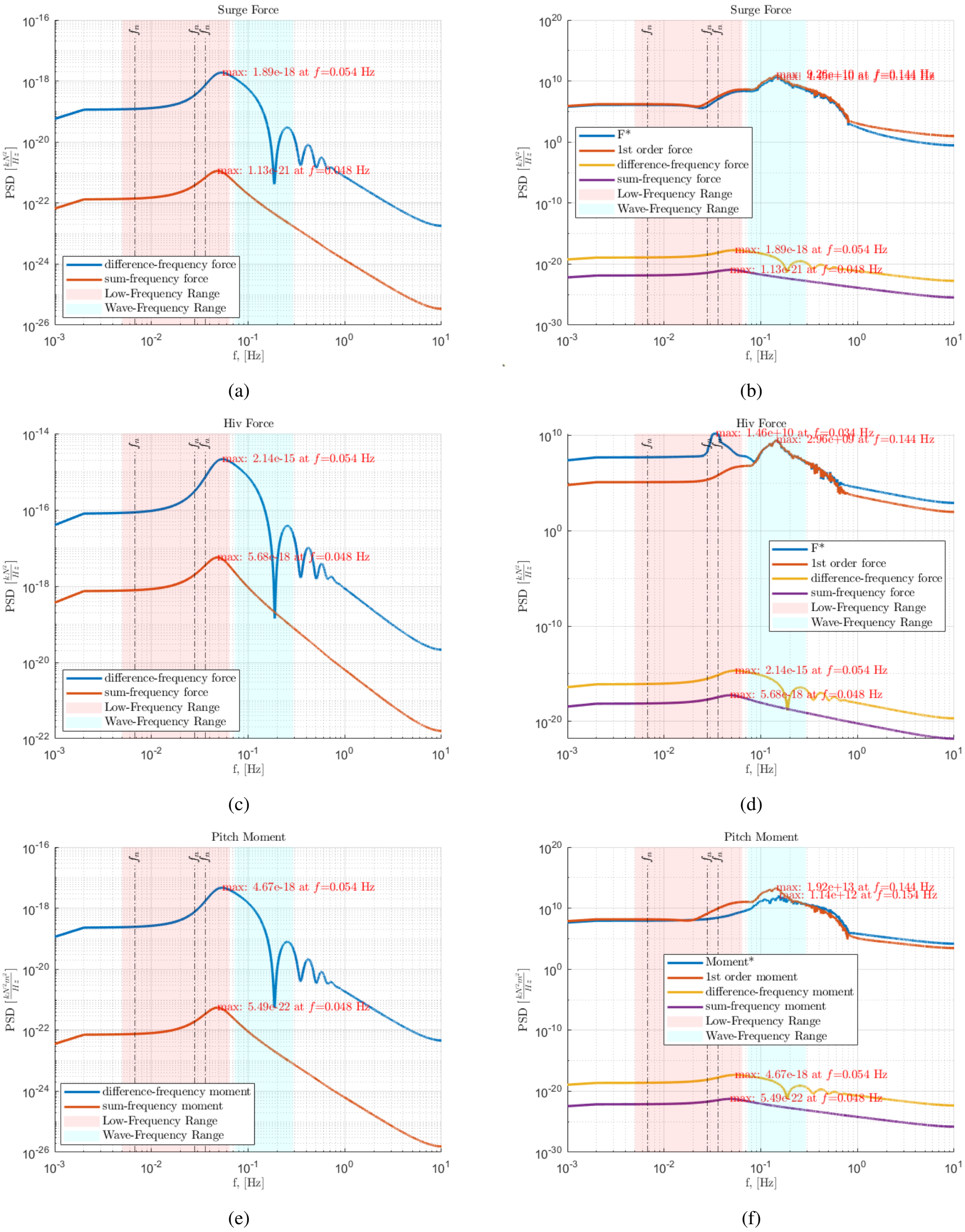
(c)

(d)

(e)

(f)

Figure 17: All forces compared to differance frequency force and moment in surge (a, b), hiv (c, d) and pitch (e, f), for load case O1 presented as PSD in frequency domain



(a)

(b)

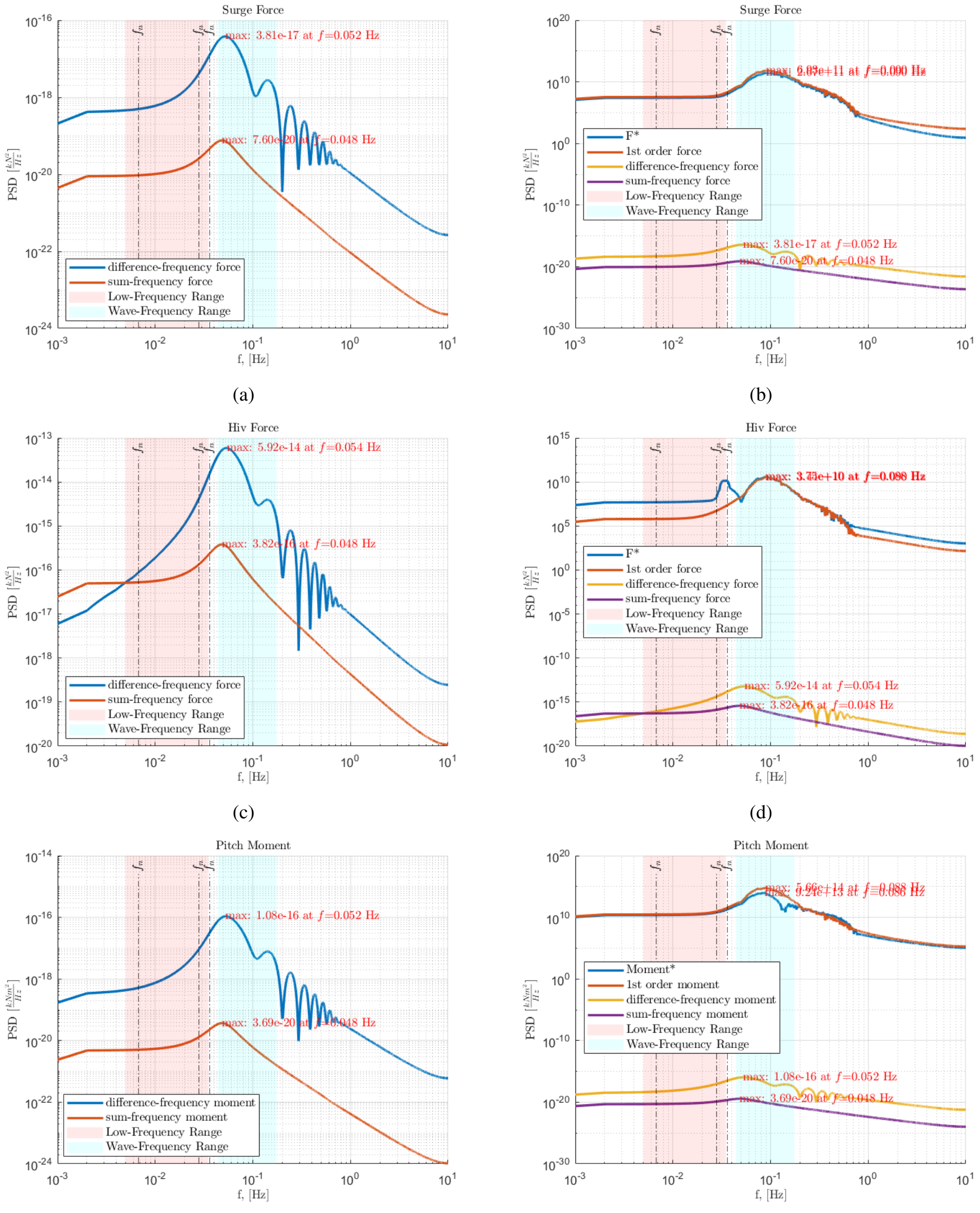
(c)

(d)

(e)

(f)

Figure 18: All forces compared to differance frequency force and moment in surge (a, b), hiv (c, d) and pitch (e, f), for load case M1 presented as PSD in frequency domain



(a)

(b)

(c)

(d)

(e)

(f)

Figure 19: All forces compared to differance frequency forces and moment in surge (a, b), hiv (c, d) and pitch (e, f), for load case S1 presented as PSD in frequency domain

Appendix C:

Dynamic Response

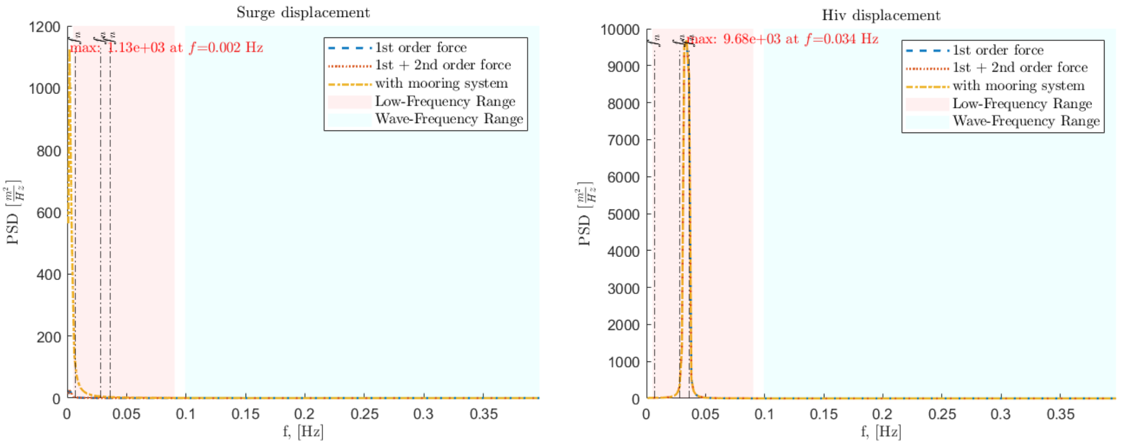


Figure 20: Displacement in hiv direction, $x_3^{(1)}$ for load case O1 presented as PSD in frequency domain.

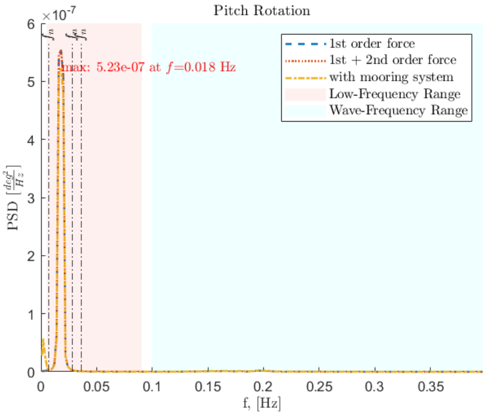


Figure 21: Rotation in pitch, $\theta^{(1)}$ for load case O1 presented as PSD in frequency domain.

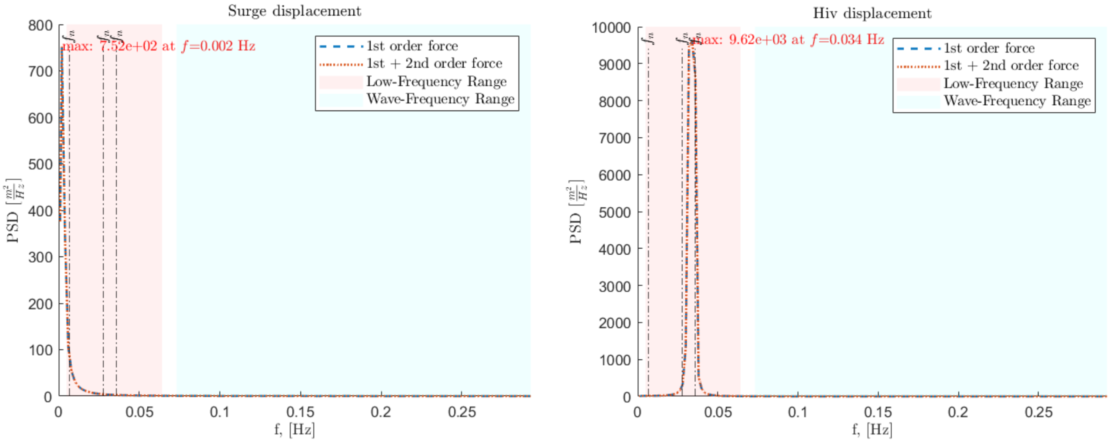


Figure 22: Displacement in hiv direction, $x_3^{(1)}$ for load case M1 presented as PSD in frequency domain.

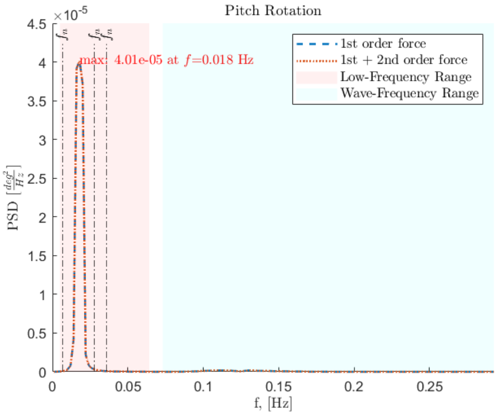


Figure 23: Rotation in pitch, $\theta^{(1)}$ for load case M1 presented as PSD in frequency domain.

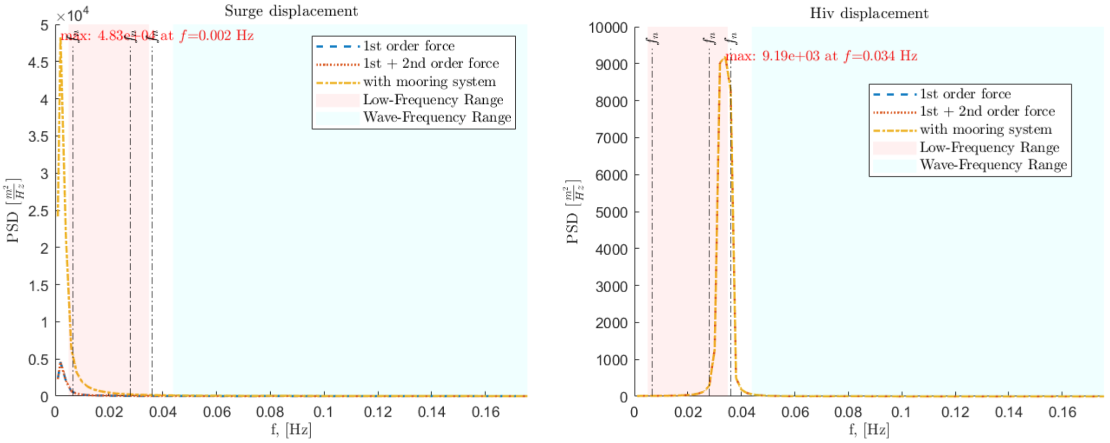


Figure 24: Displacement in hiv direction, $x_3^{(1)}$ for load case S1 presented as PSD in frequency domain.

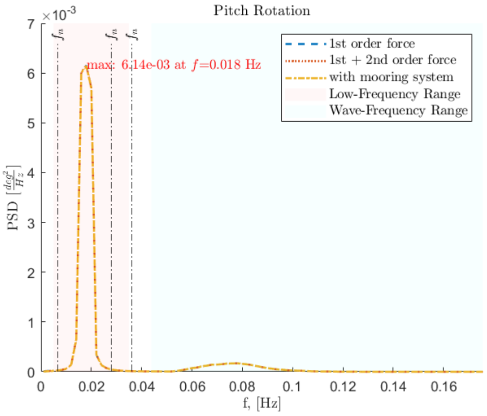


Figure 25: Rotation in pitch, $\theta^{(1)}$ for load case S1 presented as PSD in frequency domain.

Appendix D:

MATLAB code: Quasi-static Catenary Model

```matlab
    clear all; close all; clc;
      % Initializes quasi-static catenary line model
    % Calculates position for eachs segment and effective Tension, Teff
    % based on Jonkman model and equations, described
   % in chapter Mooring model, section Quasi-static model.
   [omega, EA,L,s] = Cpar();
      load('CData.mat');
            Cgen=CparData.Cgen;
            Cpar=CparData.Cpar;
            Ppar=CparData.Ppar;
            h=Cgen.h;
            s=Cgen.s;  % segments along the line
            CB=Cpar.CB;
            x_F=Cpar.x_F;
    x = zeros(size(s));
    z = zeros(size(s));
    Teff = zeros(size(s));
   for i = 1:length(s)
       [H_F, V_F, T_H, T_V, V_A, L_B, H_norm, V_norm, Teff] = getfairleadTension();
          %suspended catenary line
       [x(i), z(i)] = get_Position(s(i), H_F, V_A, omega, EA);
      %uncomment for  line laid down on the seabed
      % [x(i), z(i), Teff(i)] = get_Position(s(i), H_F, V_A, omega, EA, L_B, CB, L);
   end
% %Plotting
% x=Cpar.scale*x; % to represent in local frame
% % segments vs Teff separate plot
% figure;
%  subplot(2, 1, 1);
%     plot(x, z, 'bo-', 'DisplayName', 'Mooring line model');
%     xlabel('Horizontal Position, x (m)', 'Interpreter','latex');
%     ylabel('Vertical Position, z (m)', 'Interpreter','latex');
%     % title('Local coordinat system of the mooring line.');
%     grid on;
%     legend;
% subplot(2, 1, 2);
%     plot(s, Teff, 'ko-', 'DisplayName', 'Effective tension');
%     xlabel('length of segments, s (m)', 'Interpreter','latex');
%     ylabel('Effective Tension, $T_eff$ (N)', 'Interpreter','latex');
%     title('Effective tension, $T_eff$, along the line', 'Interpreter','latex');
%     grid on;
%     legend;
% % segments position vs Te in one 2Dplot
%  figure;
% % Plot x vs z on left y-axis
% yyaxis left;
%   plot(x, z, 'bo-', 'DisplayName', 'Mooring Line Model');
% xlabel('Horizontal Position, x [m]','Interpreter','latex', 'HandleVisibility','off');
% ylabel('Vertical Position, z [m]', 'Interpreter','latex');
% % ymax = max(z);
% % ylim([0, ymax * 1.01]);  % adding ca % 10% above the max value
% % title('Effective tension, $T_eff$ along the line', 'Interpreter','latex');
% grid on;
% hold on;
% % Plot x vs Te on right y-axis
% yyaxis right;
%  plot(x, Teff, 'r-', 'DisplayName', 'Effective Tension');
% ylabel('Effective Tension, $T_eff$ (N)', 'Interpreter','latex');
% legend('show', 'Location', 'northwest');
% %%3D representation of position vs Te
% figure;
% ax1 = axes;
%  plot3(ax1, x, zeros(size(s)), z, 'bo-', 'DisplayName', 'Mooring Line Model');
% % plot3(ax1, x, s, z, 'bo-', 'DisplayName', 'Mooring Line Model');
% xlabel('Horizontal Position, x (m)',  'Interpreter','latex');
% ylabel(' ');
% zlabel(' ');
% grid on;
% % title('3D Plot of Mooring Line Model and Effective Tension');
% hold on;
% ax2 = axes;
%  plot3(ax2, x, zeros(size(s)), Teff, 'r-', 'DisplayName', 'Effective Tension');
% % plot3(ax2, x, s, Teff, 'r-', 'DisplayName', 'Effective Tension');
% ax2.Color = 'none';
% ax2.XAxis.Visible = 'off';
% ax2.YAxis.Visible = 'off';
% ax1.ZAxis.Visible = 'off';
% ax2.ZAxis.Visible = 'on';
% ax2.ZColor = 'r';
% ax2.Position = ax1.Position;
% zlabel(ax2, 'Effective Tension, $T_eff$ (N)', 'Interpreter','latex');
% hold on;
% linkprop([ax1, ax2], 'View');
% grid on;
% legend([ax1.Children; ax2.Children], ...
%     {'Mooring Line Model', 'Effective Tension'}, 'Interpreter','latex','Location', 'northwest');
```

```

function [H_F, V_F, T_H, T_V, V_A, L_B, H_norm, V_norm, Teff] = getfairleadTension();
     % clear all; close all; clc;
        load('CData.mat');
            Cgen=CparData.Cgen;
            Cpar=CparData.Cpar;
            Ppar=CparData.Ppar;
            h=Cgen.h;
            s=Cgen.s;
            CB=Cpar.CB;
            xf=Ppar.xfl;
            x_F=xf; %in platform frame
             % x_F=Cpar.x_F;%in anchor frame
            z_F=Cpar.z_F;% in anchor frame
            % % zf=Ppar.zfl;
            % % z_F=zf; in platform frame

    % lambda_0 based on fairlead position
    if x_F == 0
        lambda_0 = 1000000;
    elseif sqrt(x_F^2 + z_F^2) >= L
        lambda_0 = 0.2;
    else
        lambda_0 = sqrt(3 * ((L^2 - z_F^2) / x_F^2 - 1)); %eq. (2-40)
    end

    % Initial guesses for tensions based on Peyrot and Goulois
    T_H = abs(omega * x_F/ (2 * lambda_0)); % eq. (2-39a) Initial horizontal tension guess
    T_V = (omega / 2) * (z_F/ tanh(lambda_0) + L); % eq. (2-39b)  Initial vertical tension guess

    % Newton-Raphson Method to solve for T_H and T_V

   [T_H, T_V] = estimateTensions(T_H, T_V, omega, L, EA, x_F, z_F);
    H_F=T_H;
    V_F = T_V;


% components of the effective tension in the mooring line at the anchor, HA and VA,
       % Determine H_F V_A based on seabed condition
    %
     L_B = Cpar.L - V_F / omega;  % eq:(2-38) s.39 Length of mooring line resting on the seabed (m)

    if L_B > 0
        H_A = max(H_F - Cpar.CB * omega * L_B, 0); % eq. (2-42a) s.41
        V_A = 0;% (2-42b) s. 41
    else
    H_A = H_F; %eq. (2-41a)
    V_A = V_F - omega * Cpar.L;% (2-42a) s. 41
  end

  % Calculate effective tension and normalized components for each segment
  Teff = sqrt(H_F^2 + (V_A + omega * s).^2);
  H_norm = H_F * Teff ./ sqrt(H_F^2 + (V_A + omega * s).^2);
  V_norm = (V_A + omega * s) .* Teff ./ sqrt(H_F^2 + (V_A + omega * s).^2);
% % uncomment below lines for case with seabed
% % Calculate the effective tension T_e(s) and its components
% for i = 1:length(s)
%  if s(i) <= L_B
%    Teff(i) = max(H_F + CB * omega * (s(i) - L_B), 0); % for 0 <= s <= L_B
%    % Horizontal and vertical components
%    H_norm(i) = H_F;
%    V_norm(i) = 0;
%  else % for L_B <= s <= L
%    Teff(i) = sqrt(H_F^2 + (omega * (s(i) - L_B))^2);
%    H_norm(i) = H_F * Teff(i) / sqrt(H_F^2 + (V_A + omega * s(i))^2);
%    V_norm(i) = (V_A + omega * s(i)) * Teff(i) / sqrt(H_F^2 + (V_A + omega * s(i))^2);
%  end
% end
%
%
 end
```

```
function [x, z] = get_Position(s, H_F, V_A, omega, EA)
%based on Jonkmans equation given in https://www.nrel.gov/docs/fy08osti/41958.pdf
  % horizontal position x(s) of segments
x = (H_F / omega) * (log((V_A + omega * s) / H_F + sqrt(1 + ...
  ((V_A + omega * s) / H_F)^2)) ...
    - log((V_A / H_F) + sqrt(1 + (V_A / H_F)^2))) + ...
(H_F * s / EA); % eq. (2-43a) s 41

  % vertical position z(s) of segments
  z = (H_F / omega) * (sqrt(1 + ((V_A + omega * s) / H_F)^2) ...
    - sqrt(1 + (V_A / H_F)^2)) ...
    + (1 / EA) * (V_A * s + (omega * s^2) / 2); % eq (2-43b) s 41

end
% %%%%%%%% uncomment for case with  line laid down
% on the seabed equations 2-45 2-46
% function [x, z, Te] = get_Position(s, H_F, V_A, ...
% omega, EA, L_B, CB, L)
%  % Calculate the horizontal position x(s) based on extended conditions
%  if s <= L_B - (H_F / (CB * omega))
%    x = s; % for 0 <= s <= L_B - H_F / ...
% (C_B * omega)
%  elseif s <= L_B
%         x = s + (CB * omega / (2 * EA)) * ...
%             (s^2 - 2 * (L_B - (H_F / (CB * omega))) * s + ...
%             (L_B - (H_F / (CB * omega))) * max(L_B -
% (H_F / (CB * omega)), 0));
%     else % for L_B <= s <= L
%         x = L_B + (H_F / omega) * ...
% log((omega * (s - L_B) / H_F) + sqrt(1 + ...
% (omega * (s - L_B) / H_F).^2)) + ...
%             (H_F * s / EA) + (CB * omega / (2 * EA)) * ...
%             (-L_B^2 + (L_B - ...
% (H_F / (CB * omega))) * max(L_B - (H_F / (CB * omega)), 0));
%     end
%
%     % Calculate the vertical
% position z(s) based on extended conditions
%     if s <= L_B
%         z = 0; % for 0 <= s <= L_B
%     else % for L_B <= s <= L
%         z = (H_F / omega) * log(sqrt(1 + ...
% (omega * (s - L_B) / H_F).^2) - 1) + ...
%             (omega * (s - L_B).^2 / (2 * EA));
%     end
%     if s <= L_B
%         Te = max(H_F + CB * omega * ...
% (s - L_B), 0); % for 0 <= s <= L_B
%         % Horizontal and vertical components
%         H_normalized = H_F;
%         V_normalized = 0;
%     else % for L_B <= s <= L
%         Te= sqrt(H_F^2 + (omega * ...
% (s - L_B))^2);
%         H_normalized = H_F * Te / sqrt(H_F^2 + (V_A + omega * s)^2);
%         V_normalized = (V_A + omega * s) ...
% * Te / sqrt(H_F^2 + (V_A + omega * s)^2);
%     end
%
% end
```

```
function [T_H, T_V] = estimateTensions(T_H, T_V, omega, L, EA, x_F, z_F)

tolerance = 1e-6; % Convergence tolerance
    max_iter = 100; % Maximum number of iterations
    for iter = 1:max_iter
    %estimate x z for fairlead, based guessed and Newton-Raphson test
        [x_est, z_est] = mooringLineEquations(T_H, T_V, omega, L, EA);

        % errors
        error_x = x_F - x_est;
        error_z = z_F - z_est;
    H_F=T_H;
    V_F = T_V;
        % Establishing Jacobian with partial derivatives
dx_dH=-(log((V_F-L.*omega)./H_F+sqrt(1.0./H_F.^2.*...
    (V_F-L.*omega).^2+1.0))-log(V_F./H_F+...
    sqrt(1.0./H_F.^2.*V_F.^2+1.0)))./omega+L./EA...
    +(H_F.*((1.0./H_F.^2.*(V_F-L.*omega)...
    +1.0./H_F.^3.*1.0./sqrt(1.0./H_F.^2.*(V_F-L.*omega).^2 ...
    +1.0).*(V_F-L.*omega).^2)./((V_F-L.*omega)./H_F+ ...
    sqrt(1.0./H_F.^2.*(V_F-L.*omega).^2+1.0))- ...
    (1.0./H_F.^2.*V_F+1.0./H_F.^3.*V_F.^2.*1.0./ ...
    sqrt(1.0./H_F.^2.*V_F.^2+1.0))./(V_F./H_F+ ...
    sqrt(1.0./H_F.^2.*V_F.^2+1.0))))./omega;
dx_dV=-(H_F.*((1.0./H_F+(1.0./H_F.^2.*(V_F.*2.0 ...
    -L.*omega.*2.0).*1.0./sqrt(1.0./H_F.^2.*(V_F ...
    -L.*omega).^2+1.0))./2.0)./((V_F ...
    -L.*omega)./H_F+sqrt(1.0./H_F.^2.*(V_F ...
    -L.*omega).^2+1.0))-(1.0./H_F+1.0./H_F.^2.*V_F.*1.0./ ...
    sqrt(1.0./H_F.^2.*V_F.^2+1.0))./(V_F./H_F ...
    +sqrt(1.0./H_F.^2.*V_F.^2+1.0))))./omega;
dz_dH=(sqrt(1.0./H_F.^2.*V_F.^2+1.0)- ...
    sqrt(1.0./H_F.^2.*(V_F-L.*omega).^2 ...
    +1.0))./omega+(H_F.*(1.0./H_F.^3.* ...
    1.0./sqrt(1.0./H_F.^2.*(V_F-L.*omega).^2 ...
    +1.0).*(V_F-L.*omega).^2- ...
    1.0./H_F.^3.*V_F.^2.*1.0./sqrt(1.0./H_F.^2.* ...
    V_F.^2+1.0)))./omega;
dz_dV=L./EA-(H_F.*((1.0./H_F.^2.*(V_F.*2.0-L.* ...
    omega.*2.0).*1.0./sqrt(1.0./H_F.^2.*(V_F-L.*omega).^2 ...
    +1.0))./2.0-1.0./H_F.^2.*V_F.* ...
    1.0./sqrt(1.0./H_F.^2.*V_F.^2+1.0)))./omega;

             J = [dx_dH, dx_dV;
             dz_dH, dz_dV];
        % Updating Newton-Raphson
        delta = J \ [error_x; error_z];
        T_H = T_H + delta(1);
        T_V = T_V + delta(2);
        % Cheking Convergence
        if norm(delta) < tolerance
            % fprintf('Converged after %d iterations.\n', iter);
            return;
        end
    end

end
```

```
 function [iseg] = Mech();
    %%The High-Modulus Polyethylene (HMPE) line designed specifically
    % for the model used in simulation
    % according to DIN EN ISO 1141 standard and
    % OCIMF Mooring Equipment Guidelines (MEG4)
    %Diameter
     iseg = struct();
     iseg.d = 0.18;  % [m] %based on calculations performed for designed moorig line
     d=iseg.d;% approx. 24.8107 [kg/m]

    iseg.A = pi * iseg.d^2/4; % approx. 0.0254 [m^2]

    iseg.rho_HMPE = 975;  % Density of HMPE [kg/m^3]

    % Mass per unit length
    iseg.masspm=iseg.A*iseg.rho_HMPE;
    masspm=iseg.masspm;
    %exporting s segments length and applying it on the MFM model
    [omega, EA,L,s] = Cpar();
    L1 = s(2)-s(1); % [m]
    L2 = s(3)-s(2); % [m]

    % Mass per segment in kg
    mass1 = iseg.masspm;
    mass2 = iseg.masspm;


    iseg.m1 = mass1;
    iseg.m2 = mass1;
    iseg.L1 = L1;
    iseg.L2 = L2;

    iseg.J11 = pi*d^4/64; %approx. [kgm^2]
    iseg.J12 = pi*d^4/64;

 %J13 =J23 is small, therefore negligible.
    iseg.J13 =0;
    iseg.J21 = pi*d^4/64;
    iseg.J22 = pi*d^4/64;
    iseg.J23 =0;

MechData = struct('iseg', iseg);
save 'MechData.mat' -v7.3;
 end
```

Appendix E:

MATLAB code: Post-processing

```matlab
clc; clear; close all;
%This function  calculates Power Spectral Density (PSD)
% of a time-domain signal using Fast Fourier Transformation
%described in 7.2 Post-processing
%%%%%CONTROL Combinations
%Default ALL=0, three wave forces, 1st order vs sum-differance frequency forces
%All=0, except includeFstar -> plot all four forces
%All=0, except exclude1ordF=1-> only 2d order forces
%All=0, except excludeSDForce=1-> plot F* vs 1 order force
logScale=1; % loglog xy, switch between scales, bedre for cluster signals
includeFstar=1; %includes Fstar in plot
exclude1ordF =0; %excludes 1st order wave force, to see only sum - differancies forces
excludeSOForces=0; % excludes 2-nd order wave forces,  to check Fstar vs 1st order Force
% Eigenperiods and corresponding frequencies
T0= [149.3, 27.7, 35.7]; %eigenperiods, T0
ef= 1 ./ T0; %

setSurge=0; % surge active
setHeave=0; % aktivere/deaktivere hiv
setPitch=0; %  pitch rotation
% Defines the cases and selects data files accordinly
case1 = 0; % Mild Sea State (Case 1)
case2 = 0; % Moderate Sea state (Case 2)
case3 = 1; % Severe Sea State (Case 3)
%%%%%%%%%%%%%%%%%%% Defines Wave Spectrum Model
if case1
      getName = {'Case1_uML.mat'};
     T1 = 4.20; % Defines Wave Spectrum Model
elseif case2
    getName = {'Case2_uML.mat'};
     T1 = 5.70;
elseif case3
         getName = {'Case3_uML.mat'};
      T1 = 9.50;
else
```

```matlab
    getName = {'Case1_uQTF_uML.mat'};
end
  keepTrackLegend = {'1st order force only', ...
      'sum-differance force included',...
      'all forces+mooring lines'};
Tp = 1.199 * T1;
fp = 1 / Tp;
   LF= [0.005, fp/2-0.009]; % defines low-frequency range based on Tp
  WF = [fp/2, 2*fp]; %defines wave-frequency range based in Tp
df1 = 0.0001;%step1 in LF, to get better resolution
df2 = 0.001; % step in LF
f = [0.001:df1:0.99, 1:df2:10];  % Frequency vector up to 10 Hz

% Data processing parameters
dt = 0.05; % simulation step
fNyq = 1 / (2 * dt); %nyquist frequency
cutoff = [0.3, 4] / (2 * pi); % apply filter to remove noise

figure;
hold on;
setMultLegend= {};  %  For several time-series


for k = 1:length(getName)
    data = load(getName{k});%load data

    if includeFstar == 1
        % Include all forces, including Fstar
         forceChoice = {'Fstar', 'Fdiff', 'Fdiff2', 'Fsum2'};
                  if setPitch==1
            renameLegend= {'Moment*', '1st order moment',...
                'difference-frequency moment', 'sum-frequency moment'};
               else
            renameLegend = {'F*', '1st order force', ...
                'difference-frequency force', 'sum-frequency force'};
             end

    elseif includeFstar ~= 1 && exclude1ordF ~= 1 ...
        && excludeSOForces == 0
        % Include only 1st order and sum-difference forces,
        % exclude Fstar if flag is not set to include it
        forceChoice = {'Fdiff', 'Fdiff2', 'Fsum2'};

        if setPitch==1
             renameLegend = {'1st order moment', ...
                 'difference-frequency moment', 'sum-frequency moment'};

        else
        renameLegend = {'1st order force', ...
```

```
            'difference-frequency force', 'sum-frequency force'};

        end
    elseif excludeSOForces == 1 && includeFstar == 0

        forceChoice = {'Fstar', 'Fdiff'};
        if setPitch==1
             renameLegend = {'Moment*', '1st order moment'};
        else
        renameLegend = {'F*', '1st order force'};
        end
    elseif exclude1ordF == 1

        forceChoice = {'Fdiff2', 'Fsum2'};

              if setPitch==1
                 renameLegend = {'difference-frequency moment', ...
                     'sum-frequency moment'};
             else
         renameLegend = {'difference-frequency force', ...
             'sum-frequency force'};

            end
    else
      %Default
        forceChoice = {};
        renameLegend = {};
        disp('Undefined control combination. Please, reset.');
        continue;  %do nothing in for loop
    end

    % Check if Fstar is present and loaded properly
    fstarPresent = isfield(data, 'Fstar');

    % Adjust time and duration based on the presence of Fstar
    if fstarPresent
        simLength = 499;  % Fstar data
        time = data.t(2:end);  % adjust to Fstar by skipping
    else
        simLength = 500;
        time = data.t;  % Use the full time vector
    end
% for time-serie with different length compared to other. Fstar is less
    for j = 1:length(forceChoice)
        varName = forceChoice{j};
        if isfield(data, varName)
            resMatrix = real(getfield(data, varName));
            [nRows, nCols] = size(resMatrix);
            % Special processing if Fstar is present
            % and not currently processing Fstar
            if fstarPresent && strcmp(varName, 'Fstar') ...
                    == 0 && nCols > 1
                % Skip the first column for
                % all forces other than Fstar
                if setSurge == 1 && nRows >= 1
                    response = resMatrix(1, 2:end);
                elseif setHeave == 1 && nRows >= 2
                    response = resMatrix(2, 2:end);
                elseif nRows >= 3
                    response = resMatrix(3, 2:end);
                else
                    error(['Less rows than expected %s ...' ...
                        'in file %s'], varName, getName{k});
                end
            elseif fstarPresent == 0 || strcmp(varName, ...
                    'Fstar') == 1
                %if Fstar is not included in analyse,
                % proceed as normal
                if setSurge == 1 && nRows >= 1
                    response = resMatrix(1, :);
                elseif setHeave == 1 && nRows >= 2
                    response = resMatrix(2, :);
                elseif nRows >= 3
                    response = resMatrix(3, :);
                else
                    error(['Less rows than expected %s ...' ...
                        'in file %s'], varName, getName{k});
                end
            end
        else
            warning('Variable %s not found in file %s', ...
                varName, getName{k});
        end

%interpolate to get unifrm time
            t = -simLength/2:dt:simLength/2;
            xint = interp1(time, response, t);

            % Apply bandpass filter
            [b, a] = butter(4, cutoff / fNyq, 'bandpass');
            xint_filtered = filtfilt(b, a, xint);

            % Calculate PSD
            SxxN = psd_fft(xint_filtered' - mean(xint_filtered), 3, f, 1 / dt);
             Sxx=SxxN/1000; %to kN

              currentLegend = sprintf('%s - %s', keepTrackLegend{k}, varName);
            % Plot
            if logScale ==1
                %inlogaritmic scale
                loglog(f, Sxx, 'LineWidth', 2, 'DisplayName',...
                    currentLegend);
                set(gca, 'XScale', 'log', 'YScale', 'log');
                % Set both axes to logarithmic
                grid on;
            else
            plot (f, Sxx, 'LineWidth', 2, 'DisplayName', currentLegend);
            set(gca, 'FontSize', 10);
             xlim([0 2*fp]);
            end
             keepTrackLegend{end + 1} = currentLegend; % Collect legend entry

            % if exclude1ordF||excludeSDForces ==1
            % Annotate directly on the plot with transparent background
            %for 2 vars or if max have long distance from each other
            % Find indices and values of the maximum Z value
                [maxValue, linearIndexesOfMaxes] = max(Sxx(:));
                [rowOfMax, colOfMax] = ind2sub(size(Sxx), linearIndexesOfMaxes);
                maxX = f(rowOfMax, colOfMax);
                text(maxX,  maxValue + 0.05*maxValue, ...
                    sprintf('max: %.2e at $f$=%.3f Hz',...
                    maxValue, maxX), 'Interpreter', 'latex', ...
                    'Color', 'red', 'FontSize', 10, ...
                    'BackgroundColor', 'none', 'Position', [maxX, maxValue]);
                hold on;
            % else
            %     %Do nothing
            % end
    end
end


if setSurge ==1
     hy=ylabel(' PSD $[\frac{kN^2}{Hz}]$'); %surge force
     ht=title('Surge Force', 'fontsize', 12)

elseif setHeave ==1
        hy=ylabel(' PSD $[\frac{kN^2}{Hz}]$'); %hiv Force
        ht=title('Hiv Force', 'fontsize', 12)
 else

        hy=ylabel(' PSD $[\frac{kN^2m^2}{Hz}]$'); %pitch moment
        ht=title('Pitch Moment', 'fontsize', 12)
end


hx=xlabel('f, [Hz]');
```

```
% hl= legend(legendEntries, 'Location', 'northeast');
hl= legend( renameLegend, 'Location', 'best');
% ht=title(' ');

set([hx,hy,hl, ht],'fontsize',10, 'Interpreter','latex');

y_limits = ylim;
%draw LF range and WF range
fill([LF(1), LF(1), LF(2), LF(2)], ...
     [y_limits(1), y_limits(2), y_limits(2), y_limits(1)], ...
     'r', 'FaceAlpha', 0.09, 'EdgeColor', 'none', 'DisplayName',...
     'Low-Frequency Range');

% Wave Frequency Window
fill([WF(1), WF(1), WF(2), WF(2)], ...
     [y_limits(1), y_limits(2), y_limits(2), y_limits(1)], ...
     'c', 'FaceAlpha', 0.09, 'EdgeColor', 'none', ...
     'DisplayName', 'Wave-Frequency Range');

%eigenfrequencies of the FOWT
%rad/s
% hxl=xline(ef(1:3),'k-.',' $\omega_n$',...
% 'LabelHorizontalAlignment','right', 'HandleVisibility','off', ...
% 'Interpreter','latex'); hold on
%Hz
hxl=xline(ef(1:3),'k-.',' $f_n$','LabelHorizontalAlignment',...
    'center', 'HandleVisibility','off', 'Interpreter','latex'); hold on
hold off;
```

```
clc; clear; close all;
% % Eigenperiods and corresponding frequencies
T0 = [149.3, 27.7, 35.7];
ef= 1 ./ T0; %
%DOF
setSurge=0; % surge active
setHeave=0; % aktivere/deaktivere hiv
setPitch=1; % yrotation

% Define load cases based on sea states
case1 = 0; % Mild Sea State (Case 1)
case2 = 0; % Moderate Sea state (Case 2)
case3 = 1; % Severe Sea State (Case 3)
%all three configurations,i.e. only 1st order force,
% w QTFs, m mooring
%system
if case1==1
    getName = {'Case1_uML_uQTF.mat','Case1_uML.mat',...
        'Full_Case1.mat'};
   T1 = 4.20;
```

```matlab
elseif case2==1
getName = {'Case2_uML_uQTF.mat','Case2_uML.mat',...
    'Full_Case2.mat'};
    T1 = 5.70;
 elseif case3==1
    getName = {'Case3_uML_uQTF.mat','Case3_uML.mat',...
        'Full_Case3.mat'};
      T1 = 9.50;
    else
   getName = {'Case1_uML_uQTF.mat','Case1_uML.mat',...
       'Full_Case1.mat'};
end
 Tp = 1.199 * T1;  % Calculate Tp based on T1
 fp=1/Tp;
  LF= [0.005, fp/2-0.009]; % Adjusted to prevent overlapping
  WF = [fp/2, 2*fp]
fileContent = {'1st order force', '1st + 2nd order force',...
    'with mooring system'};

simLength = 500;
dt = 0.05;
fNyq = 1 / (2 * dt);
fs = 2 * fNyq; % Sampling frequency

figure;
hold on;

nameLegends = {}; % Initialize outside the loop to collect all entries
% Define line styles
lineStyles = {'--', ':', '-.'};  % List of line styles
for k = 1:length(getName)
    data = load(getName{k});
    time = data.t;

    if setSurge == 1
        response = data.q_surge1(1,:);
        % dataDescription = 'Surge displacement';
        dataDescription =' ';
    elseif setHeave == 1
           response=real(data.q_heave1(1,:));
         % dataDescription = 'Hiv displacement';
           dataDescription =' ';
    elseif setPitch==1
           response=data.q_pitch1(1,:);
           % dataDescription = 'Pitch rotation';
             dataDescription =' ';
    elseif FdiffFlag ==1
      response=data.Fdiff(3,:);
      response=real(data.Fdiff(3,:));
      dataDescription = 'Wave Force';
    else
        continue; % do nothing
    end
 t=-simLength/2:dt:simLength/2;
xint=interp1(time,response,t); %Interpolate data
cutoff=[0.3 4]/(2*pi); %Cut-off frequencies
[b,a]=butter(4,cutoff/fNyq,'bandpass'); %Get filter coefficients
df1=0.0001;
df2=0.001;
f1=0.001:df1:0.99;
f2=1:df2:fNyq;
f=[f1 f2]; % appends frequencies
Sxx=psd_fft(xint'-mean(xint),3,f,1/dt); %PSD of unfiltered data
 plot(f, Sxx, 'LineStyle', ...
     lineStyles{mod(k-1, length(lineStyles)) + 1}, 'LineWidth', 1.5);
     currentLegend = sprintf('%s  %s', fileContent{k}, dataDescription);
    nameLegends{end + 1} = currentLegend; % Collect legend entry
end
%eigenfrequencies of the FOWT
%rad/s
% hxl=xline(ef(1:3),'k-.',' $\omega_n$',...
% 'LabelHorizontalAlignment','right', ...
% 'HandleVisibility','off', 'Interpreter','latex'); hold on
%Hz
hxl=xline(ef(1:3),'k-.',' $f_n$',...
    'LabelHorizontalAlignment','center', ...
    'HandleVisibility','off', 'Interpreter','latex'); hold on

if setSurge ==1
     hy=ylabel(' PSD $[\frac{m^2}{Hz}]$');
     ht=title('Surge displacement', 'fontsize', 12)

elseif setHeave ==1
        hy=ylabel(' PSD $[\frac{m^2}{Hz}]$');
        ht=title('Hiv displacement', 'fontsize', 12)
 else

        hy=ylabel(' PSD $[\frac{deg^2}{Hz}]$');
        ht=title('Pitch Rotation', 'fontsize', 12)
end
xlim([0 2*fp]);

hx=xlabel('f, [Hz]');
hl= legend(nameLegends, 'Location', 'northeast');

% ht=title(' ');

set([hx,hy,hl, ht],'fontsize',10, 'Interpreter','latex');
```

```
% Find indices and values of the maximum Z value
[maxValue, linearIndexesOfMaxes] = max(Sxx(:));
[rowOfMax, colOfMax] = ind2sub(size(Sxx), linearIndexesOfMaxes);
maxX = f(rowOfMax, colOfMax);

% Annotate directly on the plot with transparent background
text(maxX,  maxValue + 0.05*maxValue, ...
    sprintf('max: %.2e at $f$=%.3f Hz', ...
    maxValue, maxX), 'Interpreter', 'latex', ...
    'Color', 'red', 'FontSize', 10, ...
    'BackgroundColor', 'none', 'Position', [maxX, maxValue]);

y_limits = ylim;

% Low Frequency Range
fill([LF(1), LF(1), LF(2), LF(2)], ...
     [y_limits(1), y_limits(2), ...
     y_limits(2), y_limits(1)], 'r', ...
     'FaceAlpha', 0.06, 'EdgeColor', ...
     'none', 'DisplayName', 'Low-Frequency Range');

% Wave Frequency Range
fill([WF(1), WF(1), WF(2), WF(2)], ...
     [y_limits(1), y_limits(2), y_limits(2), ...
     y_limits(1)], 'c', 'FaceAlpha', 0.06, ...
     'EdgeColor', 'none', 'DisplayName', 'Wave-Frequency Range');
hold off;
set(gca, 'FontSize', 10);
```

```
clc; clear; close all;
%This function  calculates Power Spectral Density (PSD)
% of a time-domain signal using Fast Fourier Transformation
%described in 7.2 Post-processing
%%%%%CONTROL Combinations
%Default ALL=0, three wave forces, 1st order
% vs sum-differance frequency forces
%All=0, except includeFstar -> plot all four forces
%All=0, except exclude1ordF=1-> only 2d order forces
%All=0, except excludeSDForce=1-> plot F* vs 1 order force
logScale=1; % loglog xy, switch between scales,
% bedre for cluster signals
includeFstar=0; %includes Fstar in plot
exclude1ordF =0; %excludes 1st order wave force, to
% see only sum - differancies forces
excludeSOForces=1; % excludes 2-nd order wave forces,
% to check Fstar vs 1st order Force
% Eigenperiods and corresponding frequencies
T0= [149.3, 27.7, 35.7]; %eigenperiods, T0
```

```
ef= 1 ./ T0; %

setSurge=0; % surge active
setHeave=0; % aktivere/deaktivere hiv
setPitch=1; %  pitch rotation
% Defines the cases and selects data files accordinly
case1 = 0; % Mild Sea State (Case 1)
case2 = 0; % Moderate Sea state (Case 2)
case3 = 1; % Severe Sea State (Case 3)
%%%%%%%%%%%%%%%%%%% Defines Wave Spectrum Model
if case1
      getName = {'Case1_uML.mat'};
     T1 = 4.20; % Defines Wave Spectrum Model
elseif case2
    getName = {'Case2_uML.mat'};
     T1 = 5.70;
elseif case3
         getName = {'Case3_uML.mat'};
      T1 = 9.50;
else
    getName = {'Case1_uQTF_uML.mat'};
end
  keepTrackLegend = {'1st order force only',
      'sum-differance force included','all forces+mooring lines'};
Tp = 1.199 * T1;
fp = 1 / Tp;
   LF= [0.005, fp/2-0.009]; % defines low-frequency range based on Tp
  WF = [fp/2, 2*fp]; %defines wave-frequency range based in Tp
df1 = 0.0001;%step1 in LF, to get better resolution
df2 = 0.001; % step in LF
f = [0.001:df1:0.99, 1:df2:10];  % Frequency vector up to 10 Hz

% Data processing parameters
dt = 0.05; % simulation step
fNyq = 1 / (2 * dt); %nyquist frequency
cutoff = [0.3, 4] / (2 * pi); % apply filter to remove noise

figure;
hold on;
setMultLegend= {};  %  For several time-series


for k = 1:length(getName)
    data = load(getName{k});%load data

    if includeFstar == 1
        % Include all forces, including Fstar
         forceChoice = {'Fstar', 'Fdiff', 'Fdiff2',...
             'Fsum2'};
```

```
                  if setPitch==1
            renameLegend= {'Moment*', '1st order moment',...
                'difference-frequency moment', 'sum-frequency moment'};
               else
            renameLegend = {'F*', '1st order force', ...
                'difference-frequency force', 'sum-frequency force'};
             end

    elseif includeFstar ~= 1 && exclude1ordF ~= 1...
        && excludeSOForces == 0
        % Include only 1st order and sum-difference
        % forces, exclude Fstar if flag is not set to include it
        forceChoice = {'Fdiff', 'Fdiff2', 'Fsum2'};

        if setPitch==1
             renameLegend = {'1st order moment', ...
                 'difference-frequency moment', 'sum-frequency moment'};

        else
        renameLegend = {'1st order force', ...
            'difference-frequency force', 'sum-frequency force'};

        end
    elseif excludeSOForces == 1 && includeFstar == 0

        forceChoice = {'Fstar', 'Fdiff'};
        if setPitch==1
             renameLegend = {'Moment*', '1st order moment'};
        else
        renameLegend = {'F*', '1st order force'};
        end
    elseif exclude1ordF == 1

        forceChoice = {'Fdiff2', 'Fsum2'};

              if setPitch==1
                 renameLegend = {'difference-frequency moment', ...
                     'sum-frequency moment'};
             else
         renameLegend = {'difference-frequency force', ...
             'sum-frequency force'};

            end
    else
      %Default
        forceChoice = {};
        renameLegend = {};
        disp('Undefined control combination. Please, reset.');
        continue;  %do nothing in for loop
    end

    % Check if Fstar is present and loaded properly
    fstarPresent = isfield(data, 'Fstar');

    % Adjust time and duration based on the presence of Fstar
    if fstarPresent
        simLength = 499;  % Fstar data
        time = data.t(2:end);  % adjust to Fstar by skipping
    else
        simLength = 500;
        time = data.t;  % Use the full time vector
    end
% for time-serie with different length compared to other. Fstar is less
    for j = 1:length(forceChoice)
        varName = forceChoice{j};
        if isfield(data, varName)
            resMatrix = real(getfield(data, varName));
            [nRows, nCols] = size(resMatrix);
            % Special processing if Fstar is present
            % and not currently processing Fstar
            if fstarPresent && strcmp(varName,.. ...
                    'Fstar') == 0 && nCols > 1
                % Skip the first column for all
                % forces other than Fstar
                if setSurge == 1 && nRows >= 1
                    response = resMatrix(1, 2:end);
                elseif setHeave == 1 && nRows >= 2
                    response = resMatrix(2, 2:end);
                elseif nRows >= 3
                    response = resMatrix(3, 2:end);
                else
                    error(['Less rows than expected ...%s ' ...
                        'in file %s'], varName, getName{k});
                end
            elseif fstarPresent == 0 || strcmp(varName, ...
                    'Fstar') == 1
                %if Fstar is not included in analyse,
                % proceed as normal
                if setSurge == 1 && nRows >= 1
                    response = resMatrix(1, :);
                elseif setHeave == 1 && nRows >= 2
                    response = resMatrix(2, :);
                elseif nRows >= 3
                    response = resMatrix(3, :);
                else
                    error(['Less rows than expected %s ' ...
                        'in file %s'], varName, getName{k});
                end
            end
        else
            warning(['Variable %s not found in ' ...
                'file %s'], varName, getName{k});
        end

    plot(time, response); hold on

[maxResponse, idxMax] = max(response);  % Get the maximum
alue and its index
maxTime = time(idxMax);  % Find the corresponding
% time of the maximum

%picks up max and draw circle
plot(maxTime, maxResponse, 'ro', 'MarkerSize', ...
    10, 'MarkerFaceColor', 'none', 'HandleVisibility', 'off');

text(maxTime, maxResponse, sprintf(['Max: %.2e at' ...
    ' %.2f s'], maxResponse, maxTime), ...
     'VerticalAlignment', 'bottom', ...
     'HorizontalAlignment', 'right', 'Interpreter', ...
     'latex', 'Color','red');

if setSurge ==1
     hy=ylabel(' $[N]$');
     ht=title('Surge Force', 'fontsize', 12)

elseif setHeave ==1
        hy=ylabel(' $[{N}]$');
        ht=title('Hiv Force', 'fontsize', 12)
 else
        hy=ylabel('  $[Nm]$');
        ht=title('Pitch Moment', 'fontsize', 12)
end

    end
end
hx=xlabel('time, [s]');
hl= legend( renameLegend, 'Location', 'best');
% ht=title(' '); omit legend
set([hx,hy,hl, ht],'fontsize',10, 'Interpreter','latex');
```

```
function [S,Sraw]=psd_fft(x,Ns,f,fs)

Nt=floor(size(x,1)/2)*2;
x=x(1:Nt,:);
dt=1/fs; %Step size
T=Nt*dt; %Record length
S=fft(x); %Compute dft by fft. fft specificities (added zeroes) are handled
```

```
% internally.
S=2*dt/Nt*abs(S(1:Nt/2+1,:)).^2; %Compute PSD: one-sided (multiply by 2),
% distributed (divide by sampling freq) average of fft (divide by number of
% points) squared
fS = 1/dt*(0:(Nt/2))/Nt; %Output frequencies, up to number of points/2 (higher
% frequencies only show a folded version of low frequencies)
for i=1:size(x,2)
S(:,i)=[S(1,i)/2;smooth(S(2:end,i),Ns)]; %Smoothing by moving average
end
Sraw.S=S;
Sraw.f=fS;
S=interp1(fS,S,f); %Interpolation to desired output frequencies
```

```
clc; clear; close all;
%This function calculates theoretical JONSWAP in rad/s and in Hz
HztoradFlag =0; %represents initial Hz in rad/s
% Initialize sea state cases
seaProfile = [
    struct(’Hs’, 0.8, ’T1’, 4.20, ’gamma’, 3.3),   % Mild sea state
    struct(’Hs’, 2.5, ’T1’, 5.70, ’gamma’, 3.3),   % Moderate sea state
    struct(’Hs’, 7.45, ’T1’, 9.50, ’gamma’, 1)     % Severe sea state
];
%initialaize frequency
df1 = 0.0001;
df2 = 0.001;
f1 = 0.001:df1:0.99;  % low frequencies
f2 = 1:df2:2;
f = [f1 f2];
figure;
hold on;
nLegends= {};  % Initialize legend entries

% Loop through each sea state case and plot
for i = 1:length(seaProfile)
    caseData = seaProfile(i);
    Tp = 1.199 * caseData.T1;  % Calculate peak period based on T1
    T0 = [149.3, 27.7, 35.7];
    ef= 1 ./ T0; % eigenfrequencies
    fp=1/Tp;
  %Legend shift according to seaProfile
    if caseData.gamma == 1
      legendText = sprintf([’Pierson-Moskowitz Hs=%.2f,’ ...
          ’ Tp=%.2f, gamma=%.1f’], caseData.Hs, Tp, ...
          caseData.gamma);
    else
        legendText = sprintf([’JONSWAP Hs=%.2f,’ ...
            ’ Tp=%.2f, gamma=%.1f’], caseData.Hs, Tp, ...
            caseData.gamma);
    end
```

```
if HztoradFlag ==1
    ef=ef*2*pi;
    % rad/s
    Sxx_theory=jonswapDLS( f, caseData.Hs, ...
        Tp, caseData.gamma, 'rad');
    %i J.M.J. Journee and W.W. Massie
OFFSHORE HYDROMECHANICS",
    % First Edition  eqs 5.112 5.113 fw/df=2pi ->S(w)=initial S(f)/2pi
 % https://ocw.tudelft.nl/wp-content/...
 % uploads/OffshoreHydromechanics_Journee_Massie.pdf
    plot( f*(2*pi),Sxx_theory, 'k:');%
    hx=xlabel('$\omega, [\frac{rad}{s}]$')
    hy=ylabel('$S(\omega)[\frac{m^2rad}{s}]$');

    plotHandle = plot(f*2*pi, Sxx_theory, ...
        'DisplayName', legendText);
    nLegends{end+1} = plotHandle;  % Collect handles for legend
      xlim([0 3])
else
  ef=ef;
%Hz
  Sxx_theory=jonswapDLS( f, caseData.Hs, Tp, ...
      caseData.gamma, 'Hz');
  plot(f, Sxx_theory, 'b', LineStyle='-.');
    hx=xlabel('f, [Hz]')
    hy=ylabel('$S(f)[\frac{m^2}{Hz}]$');
    xlim([0 0.5])
     % Plot spectrum
    plotHandle = plot(f, Sxx_theory, 'DisplayName', ...
        legendText,'LineWidth',1.5);
    nLegends{end+1} = plotHandle;  % Collect
andles for legend

end
end
if HztoradFlag ==1
  hxl=xline(ef(1:3),'k-.',' $\omega_n$', ...
      'LabelHorizontalAlignment','right', ...
      'HandleVisibility','off', 'Interpreter','latex');
else
    hxl=xline(ef(1:3),'k-.',' $f_n$', ...
        'LabelHorizontalAlignment','right', ...
        'HandleVisibility','off', 'Interpreter','latex');
end
hl=legend([nLegends{:}], 'Location', ...
    'best','Interpreter','latex');
ht=title('JONSWAP Spectra for Various Sea States');
ht=title(' ');
set([hx,hy,hl,ht],'fontsize',9, 'Interpreter','latex');
```

```
hold off; %stop adding to the plot
```

```
function [ S,f ] = jonswapDLS( f,Hs,Tp,gamma,domain )
g=9.81;

fp=1./Tp; %spectral peak frequency

if isempty(f)
f=0:0.001:10; %frequency range to plot over;
f=repmat(f,[size(fp,2),1]);
end
switch domain
    case 'Hz'
    f=f;
    fp=fp;
    case 'rad'
        f=2*pi.*f;
        fp=2*pi*fp;
end

alpha = (5/16)*(Hs^2)*(fp^4)/(g^2)*(1-0.287.*log(gamma));

sigma=zeros(size(f,1),size(f,2));
%spectral width parameter:
sigma(f<=fp)=0.07;
sigma(f>fp)=0.09;

S=alpha*g^2.*(f.^-5).*exp((-5/4).*((f./fp).^-4)).*gamma.^(exp(-0.5.*((f-fp)./(sigma.*fp)).^2));


end
```