NTNU

Norwegian University of
Science and Technology

Using Neural Networks to Predict the Response of a Floating Structure

Kaja Steffensen Bremer

Marine Technology
Submission date: June 2018
Supervisor: Trygve Kristiansen, IMT
Co-supervisor: Babak Ommani, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Preface

This thesis was written during Spring 2018 as part of a Master's Degree at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU) in Trondheim. The topic was suggested by Sevan Marine and Professor Trygve Kristiansen offered to supervise.

The thesis investigates the possibility of using Neural Networks to predict the response of a model, where the responses are from model experiments conducted in Ladertanken. The combination of using hydrodynamic knowledge, practical skills and analytic skills has been very rewarding. Due to my interest of programming it has been especially interesting to combine Machine Learning and Hydrodynamics. Starting from zero knowledge of Machine Learning and Python, it has been a great experience to develop a deep understanding of Neural Networks, obtaining experience in Python programming and applying the Neural Network theory in practice.

I am very thankful for Professor Trygve Kristiansen's enthusiasm and encouragement during the process and his guidance and support has been highly appreciated. I am very grateful that I got to shape my own thesis and received guidance from Trygve if I was going in the wrong direction.

Further acknowledgements involve co-supervisor Professor Babak Ommani, who contributed with his knowledge of Machine Learning which was both inspiring and helpful. The experiments would not be possible without the help of Torgeir Wahl.

At last, I want to thank Maren Eidal who has made me laugh, let me complain and made my day, several times.

Trondheim, 11.06.2018

Kaja Steffensen Bremer

Abstract

Floating structures are influenced by the ocean environment which cause responses from the structures known as surge, sway, heave, roll, pitch and yaw. These responses are commonly approximated using linear potential theory, which neglects the viscous effects created by interaction between the body and fluid. For conventional ship hulls, Ikeda developed an empirical method to estimate the roll damping [Ikeda, 1978] which is successful for traditional vessel geometries. However, the development of unconventional ship hulls has resulted in the need of better ways to predict the response and viscous effects.

This thesis investigated if Neural Networks could be utilised to predict the response of a floating structure, using experimental results. A two-dimensional model of a mid-ship section with large bilge boxes has been tested in Ladertanken, an experimental wave flume operated by SINTEF Ocean. Due to the large bilge boxes, the model represents an unconventional ship hull shape where the bilge boxes induce viscous effects. The model was freely-floating, although restrained to an area in the model using springs. A wave maker installed in the wave flume created waves with different wave period and wave steepness, which induced motions in sway, heave and roll. The comparison of the created waves versus the theoretical wave showed an average difference between 6.9 % and 7.9 % for both wave steepness and wave period. A bug in the processing software which transforms the analogue signal to a digital signal was found after the experiments were conducted, and there are indications that the results are affected. The results from the experiments could therefore not be used to analyse the development of hydrodynamic effects, however, the results were utilised as data to the response predictions.

Firstly, a Linear Regression model using Stochastic Gradient Descent was created to predict sway, heave and roll motion as well as their response amplitude operators. The Linear Regression models obtained a Coefficient of Determination ranging between 30.0 % and 42.0 %, which is unsatisfactory. The linear regression models

were included to compare to the Neural Network performance and thereby show the potential of Neural Network within non-linear problems.

One Neural Network model was built and trained for each of the measured responses, where the responses were used as targets and the wave period and wave steepness were given as features. The input data is scaled using standardisation, as well as shuffled and split into training, testing and validation data sets. The Neural Network consisted of an input layer, two hidden layers and an output layer, where the first hidden layer utilised the Rectified Linear Unit activation function and the second hidden layer utilised tangent hyperbolic. By the use of Mean Squared Error as the loss function and the Root Mean Square Propagation algorithm as optimisation method, the model trained the weights and bias to predict the targets. Using the Coefficient of Determination as the accuracy metric, both motion and response amplitude operator in sway, heave and roll resulted in an accuracy above 98.0 %, where 100.0 % is perfect prediction. From the Mean Absolute Percentage Error the motions resulted in an accuracy of 5.7 % in sway, 8.5 % in heave and 6.2 % in roll, where 0.0 % is perfect prediction. The response amplitude operator resulted in a Mean Absolute Percentage Error of 2.4 % in sway, 2.7 % in heave and 2.8 % in roll. It was suspected that the use of the wave height in the calculation of the response amplitude operator influences the prediction results, because the wave height was a feature to the Neural Network.

The prediction results showed that it was possible to utilise Neural Network to predict the response of a floating structure. Different indications of prediction error of the two accuracy metrics showed the importance of using more than one error estimate. The predictions were also plotted to visualise the difference between the prediction values and true values, which was important in the evaluation of prediction results. From the prediction plots it was concluded that the Coefficient of Determination was misleading and should not be utilised to assess the Neural Network predictions.

From the knowledge of Neural Networks and Hydrodynamics, a method has been suggested to investigate the development of the viscous terms of the model response using experiments and predictions. With the combination of freely-floating model experiments, forced oscillation model experiments and Neural Networks, a model is trained to predict the viscous terms and thereby contribute with information about the development in various sea states.

Sammendrag

Flytende konstruksjoner påvirkes av havmiljøet som forårsaker bevegelser av konstruksjonen kjent som jag, svai, hiv, rull, stamp og gir. Disse bevegelsene er vanligvis tilnærmet ved hjelp av lineær potensialteori, som neglisjerer de viskøse effektene som oppstår ved samspillet mellom skroget og fluidet. En empirisk metode utviklet av Ikeda [Ikeda, 1978] er tradisjonelt brukt til å estimere rulledempingen for konvensjonelle skipsskrog. Utviklingen av ukonvensjonelle skipsskrog resulterer imidlertid i et behov for bedre måter å estimere bevegelsene og de viskøse effektene.

Denne oppgaven har undersøkt om Nevrale Nettverk kan benyttes til å predikere bevegelsene av en flytende konstruksjon ved hjelp av eksperimentelle resultater. En to-dimensjonell modell av midt-seksjonen av et skip med store slingrekjøler ble testet i Ladertanken, en eksperimentelt bølgetank eid av SINTEF Ocean. På grunn av de store slingrekjølene representerer modellen en ukonvensjonell skipsform hvor slingrekjølene bidrar til å fremkalle viskøse effekter. Modellen var fritt flytende, men begrenset til et området av bølgetanken ved hjelp av fjærer festet på modellen. Bølgemaskinen ble satt til å lage bølger av forskjellig bølgeperiode og bratthet, som resulterte i bevegelser i svai, hiv og rull. Sammenligningen av de teoretiske og eksperimentelle bølgeperiodene og bratthetene viste en forskjell mellom 6.9 % og 7.9 %. En feil i prosesseringsprogramvaren som konverterer analoge instrumentsignaler til digitale verdier ble oppdaget etter forsøkene var gjennomført, og det ble funnet indikasjoner på at de eksperimentelle målingene var rammet av denne feilen. Grunnet dette kan resultatene ikke brukes til en hydrodynamisk analyse, men de kan likevel brukes som data i predikeringen av bevegelsene.

En lineær regresjonsmodell ble utviklet ved hjelp av stochastic gradient descent som optimaliseringsalgoritme, som predikerte både bevegelse og respons amplitude operatorene i svai, hiv og rull. De lineære regresjonsmodellene oppnådde en nøyaktighet mellom 30.0 % og 42.0 % ved hjelp av determinasjonskoeffisienten, som ikke er tilfredstillende. De lineære regresjonsmodellene er inkludert i studiet for å vise

hvordan Nevrale Nettverk yter i forhold til lineære prediksjoner.

Ett Nevralt Nettverk ble programmert og trent for svai, hiv og rull og deres respektive respons amplitude operatorer. Bølgeperiode og bølgebratthet ble brukt som input til det Nevrale Nettverket, hvor en av bevegelsene er prediksjons-objektet. Inputen ble skalert ved hjelp av standardisering, i tillegg til å bli blandet og delt i læringsdata, treningsdata og valideringsdata. Det Nevrale Nettverket besto av ett input-lag, to skjulte lag og et output-lag, hvor de to skjulte lagene benyttet seg av forskjellige aktiveringsfunksjoner. Minste kvadraters metode ble brukt som tapsfunksjon, og root mean squared propagation ble benyttet som optimaliseringsalgoritme. De Nevrale Nettverkene ble trent til å predikere svai, hiv og rulle bevegelse samt respons amplitude operatoren til de tre bevegelsene i individuelle modeller. Ved å bruke determinasjonskoeffisienten resulterer både bevegelses og respons amplitude operatorene i en nøyaktighet på over 98.0 %, hvor 100.0 % er perfekt prediksjon. Fra det gjennomsnittlige absolutt-prosentavviket ble det funnet et avvik på 5.7 % i svai, 8.5 % i hiv og 6.2 % i rull, hvor 0.0 % avvik er perfekt prediksjon. For respons amplitude operatorene ble avvikene som følger; 2.4 % i svai, 2.7 % i hiv og 2.8 % i rull. Forskjellen mellom resultatene i bevegelse og respons amplitude operatorene ble antatt å være på grunn av bruken av bølgeamplitude i beregningen av respons amplitude operatoren, som gjør at bølgehøyden er både en egenskap i nettverket og indirekte i predikeringen.

Prediksjonsresultatene viser at det er mulig å bruke Nevrale Nettverk til å predikere bevegelsen til en flytende konstruksjon. Imidlertid viste indikasjonene av de to nøyaktighetsmålene at det er viktig å benytte seg av mer enn en avviksmåling. Prediksjonene ble også visualisert ved hjelp av grafer hvor prediksjonsverdiene ble plottet mot de sanne verdiene, som viste seg å være verdifult i evalueringen av prediksjonsresultatene. Fra disse grafene ble det funnet at determinasjonskoeffisienten var misvisende og ikke burde benyttes i evalueringen av disse Nevrale Nettverkene.

Fra kunnskapen om Nevrale Nettverk og Hydrodynamikk foreslås en metode for å

analysere de viskøse leddene av bevegelsene. Ved bruk av fritt flytende modellforsøk, tvungen oscillering modellforsøk og Nevrale Nettverk kan en modell trenes til å predikere de viskøse leddene og dermed bidra til mer kunnskap om oppførselen til de viskøse leddene.

Nomenclature

Abbreviations

CFD Computational Fluid Dynamics

FPSO Floating Production, Storage and Offloading

MAPE Mean Absolute Percentage Error

MSE Mean Squared Error

MSE Mean Squared Error

PSO Particle Swarm Optimisation

RAO Response Amplitude Operator

RAO 2 Response Amplitude Operator: Sway

RAO 3 Response Amplitude Operator: Heave

RAO 4 Response Amplitude Operator: Roll

ReLU Rectified Linear Unit

RMSprop Root Mean Square Propagation

SSE Sum of Least Squares

SVM Support Vector Machine

Tanh Tangent Hyperbolic

TLP Tension Leg Platform

WP1 Wave Probe 1

WP2 Wave Probe 2

Greek Letters

η_1 Surge Motion

η_2 Sway Motion

η_3 Heave Motion

η_4 Roll Motion

η_5 Pitch Motion

η_6 Yaw Motion

λ Wave Length

ω Circular Frequency

ϕ Velocity Potential

ξ Non-Linearities

ζ Free Surface Elevation

Roman Letters

$\bar{y_i}$ Mean Value

$\bar{y}$ Mean Position in y-direction

$\hat{y_i}$ Predicted Value

A_{jk} Added Mass coefficient

b Neural Network Bias

B_{jk} Damping coefficient

C_{jk} Restoring coefficient

f	Frequency
F_{exc}	Excitation Force
F_{rad}	Radiation Force
g	Gravity
H	Wave Height
H/λ	Wave Steepness
I_{jk}	Moment of Inertia in j-direction due to motion in k-direction
$J(\theta)$	Neural Network Loss Function
k	Wave Number
M_{jk}	Mass coefficient
R^2	Coefficient of Determination
T	Wave Period
T_n	Natural Period
w	Neural Network Weights
y_i	True Value
B	Model Width
B_B	Bilge Box Width
H	Model Height
H_B	Bilge Box Height
k	Spring Stiffness
L	Model Length
L_B	Bilge Box Length
T	Model Draft

Contents

Chapter 1

Introduction

1.1 Background

Vessels and floating structures face harsh sea environments with waves, winds and currents which induce responses from the vessels and floating structures. These responses are important to approximate when the structures are designed for specific operational environments. It is of interest to reduce the vessel response to increase the operational range, and the roll motion is the response which is easiest to decrease. To enable the floating structure to be operational in a larger range of sea states, it is commonly equipped with roll damping devices. Among the common roll damping devices are bilge keels, or bilge boxes as seen in Figure 1.1, which reduce the roll motion by creating a roll damping moment. However, the presence of bilge keels and bilge boxes induce flow separation and vortex shedding due to the geometrical singularities. Flow separation and vortex shedding are viscous effects which are non-linear effects.

The vessel response is commonly approximated using linear potential theory in steady-state conditions. A consequence of assuming linearity is that only linear effects are taken into account, which means the non-linear effects are neglected. Hence the vortex shedding and flow separation are neglected, which is a good estimate in small sea states. However, in larger sea states these viscous effects are of importance and can no longer be neglected, especially in roll. For conventional

ship hulls Ikeda [Ikeda, 1978] developed an empirical method to estimate the viscous roll damping, which gives a good estimate for the traditional hull shapes. However, unconventional ship hulls can not rely on the empirical method and it is therefore of interest to obtain more knowledge of the development of the viscous effects on unconventional ship hulls. The increase of unconventional ship hulls such as FPSOs and floating wind turbines increases the need of knowledge of the viscous effects at larger sea states.

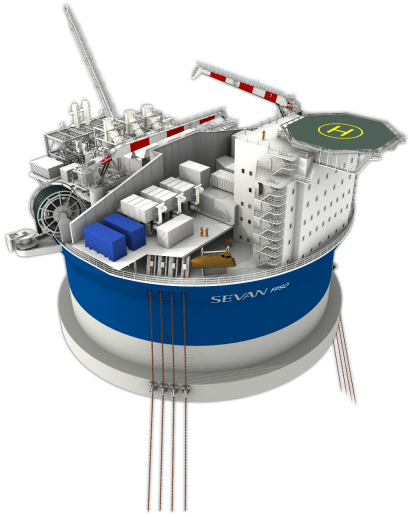


Figure 1.1: An illustration of an unconventional ship hull, namely Goliat designed by Sevan Marine [Eni, 2018]. The structure is equipped with bilge boxes which are visible at the bottom of the structure.

In the design process of unconventional ship hulls, experiments are commonly conducted which gives access to a lot of data. This opens up the possibility of using Machine Learning to investigate the response during model tests and obtain more knowledge of the viscous effects. Machine Learning is a statistical tool which is capable of finding patterns in large data sets. It is of interest to investigate if Machine Learning can be utilised to predict the response of a floating structure and thereby be helpful to analyse the viscous effects.

1.2 Literature Review

The literature review is further work of the feasibility study conducted Autumn 2017 [Bremer, 2017].

Support Vector Machine has been utilised to predict the ship manoeuvring motion together with Particle Swarm Optimisation [Luo et al., 2016]. The combined SVM and PSO was successful in predicting the ship manoeuvrability by identifying the parameters in an Abkowitz model, which is a mathematical model for ship manoeuvring. However, the paper has simplified the Abkowitz model due to the high number of parameters. This reduces the number of derivatives from sixty to thirty-six. SVM and PSO are used to approximate these parameters using 400 training sets. The results from the prediction agree well with the experimental results.

In *Modelling Ship Equations of Roll Motion using Neural Networks* [Xing and McCue, 2010] a Neural Network method to predict the ship rolling motion was tested against experimental data. By inputting the time history of roll angle and velocity in to the Neural Network model, the model was trained by minimising the difference prediction and true value of roll motion. The paper successfully estimates parameters such as damping, restoring and exciting moment.

A paper from RMIT University in Australia [Khan et al., 2005] investigates the application of Artificial Neural Network using two different training techniques to predict the ship motion. The paper utilised nine separate data sets where the last third of the data sets were used as validation of the machine learning model. The results using singular value decomposition as training technique had an accuracy of 99% for prediction of ship motion ten seconds in advance. The paper concludes that the singular value decomposition outperforms the generic algorithm as training technique and that the Artificial Neural Network model is sufficiently capable of representing the motion of a ship in open sea.

Neural Network Prediction of the Roll Motion of a Ship for Intelligent Course Control [Nicolau et al., 2007] demonstrated the capabilities Machine Learning in prediction of roll motion. The study was motivated by the influence of roll on the performance of the autopilot and the goal of developing an intelligent autopilot. An intelligent autopilot should predict the roll motion and give rudder commands which will have decreasing effects on the roll movements of the vessel. The study successfully predicts the roll motion of a ship using a Feed-Forward Neural Network Model.

Using past time series to predict the ship motion is an extremely powerful tool to further develop autonomous vessels. Most modern vessels have sensors collecting data for future purpose, such as diagnosis of components and further developing ship motion prediction tools. [Li et al., 2017] attempted to develop a more general prediction model and three learning strategies are tested - offline, online and hybrid learning. It is found that the Neural Networks perform better when constructed to target a single motion. This is due to each of the motions having an individual dependence on the features given and therefore train an exclusive Neural Network to predict each motion. The study concludes that the hybrid learning strategy was superior and can be helpful in the modelling of ship motion prediction.

Using Artificial Neural Networks to more accurately predict the heave motion is useful to support marine operations. The study conducted by *Saipem Energy Services* [Masi et al., 2011] introduces a Radial Based Neural Network to predict the heave motion of a vessel 15 s, 30 s and 50 s ahead. The goal of the study is to investigate if Neural Networks can be used to predict the vessel motion ahead of time, which could be used to improve dynamic positioning systems. The constructed Neural Network consists of input layer, a hidden layer and the output layer. The results were acceptable for predictions up to 40 s ahead, but any longer the results are progressively worse.

1.3 Present Study

The aim for the present study is to conduct experiments and predict the motion of a model using a Neural Network. Experiments will be conducted on a two-dimensional model with bilge boxes to induce viscous effects. The model will be freely-floating in waves with a spring system to restrain the model in the wave tank. A Neural Network will be programmed using Python, and considerable work has to be done to learn both Python and Machine Learning. The motivation of the present study is to investigate if Machine Learning is a way to predict the response of a floating structure and if it is possible to obtain more knowledge of the viscous effects. Because the model experiments will include the viscous effects, it is of interest to find a way to predict the added mass and damping coefficients which can give valuable information about the viscous effects of unconventional ship hulls.

1.3.1 Objectives

The following objectives are drawn up:

- Present hydrodynamic theory such as ship motions, hydrodynamic loads and motion calculations and explain viscous effects and drift loads.
- Show the analytic derivation of natural periods - coupled and uncoupled.
- A thorough introduction to basic Machine Learning with emphasis on the usage of Neural Networks
- Plan, conduct, process and present two-dimensional experiment of freely-floating model in waves
- Build and train a Neural Network do predict the response of the model using the experimental results

1.3.2 Limitations

Due to the configuration of the wave tank and model, the model is similar to a two-dimensional model and the three-dimensional effects are therefore neglected. The

model testing is time consuming and therefore limited to one bilge box size and one draft condition, and the tested wave periods and wave steepness will be limited to the test setup and wave maker. The largest possible wave periods are therefore prioritised to induce the most viscous effects.

The Neural Network prediction models are only programmed in Python using the Machine Learning Library known as Keras. There exist many other Machine Learning libraries and toolboxes, but this study is limited to utilising one programming method. Each of the models are trained on experimental data and are therefore only able to predict the motion of this specific model.

1.4 Structure of the Thesis

Chapter 2 presents the basic theory in both hydrodynamics and machine learning. In hydrodynamics the ship motions are presented together with an explanation of how to calculate the hydrodynamic loads and motions. An explanation of viscous effects and drift forces follows together with a small section about irregular waves. The equation of motion is derived to show the complexity of the problem, and the natural periods are derived using simplifications. In Machine Learning a few basic concepts such as supervised learning, linear regression and artificial neural networks are presented and the machine learning terms are presented with the use of an example.

Chapter 3 presents the experimental method which covers basics about the preparation of the experiments, instrumentation and wave generation, how the measurements are analysed and processed and an uncertainty analysis of the conducted experiment.

Chapter 4 explains how a linear regression and neural network model is build to predict the motion of the model using the experimental results. Preparation of the data points obtained from the experiments is explained as well as choice of features,

targets and parameters in the neural network.

Chapter 5 presents the results from the experiments, including wave generation and the obtained RAOs. The results from the linear regression motion predictions and the neural networks results are presented and discussed.

Chapter 6 explains how the method developed in the present study can be applied to learn more about the viscous terms in added mass and damping. Further application of Neural Networks within the hydrodynamics of floating structures are also discussed.

Chapter 7 concludes the results and the work in the present study. Several suggestions for further work are also found in this chapter.

Chapter 2

Theory

This section will introduce the reader to the theoretical basis of vessel motion and hydrodynamics as well as an introduction to Machine Learning.

2.1 Introduction to Vessel Motion

This section covers the theoretical and mathematical definitions and assumptions to give the reader a basic knowledge of vessel motion. The theory is further work from the theoretical section in the feasibility study conducted Autumn 2017 [Bremer, 2017]. The reader will be presented to the derivation of natural period and shown the complex equations of motions to understand why Machine Learning could be a helpful tool to predict ship response.

2.1.1 Motion Definitions

The definition of the coordinate system of a ship or an offshore structure is illustrated in Figure 2.1. The three translation motions are surge, sway and heave while the rotational motions are roll, pitch and yaw and together these motions form the response of a vessel. The six degrees of freedom are coupled motions which are complex to calculate. However, by assuming linear potential theory an approximation of the vessel response can be calculated.

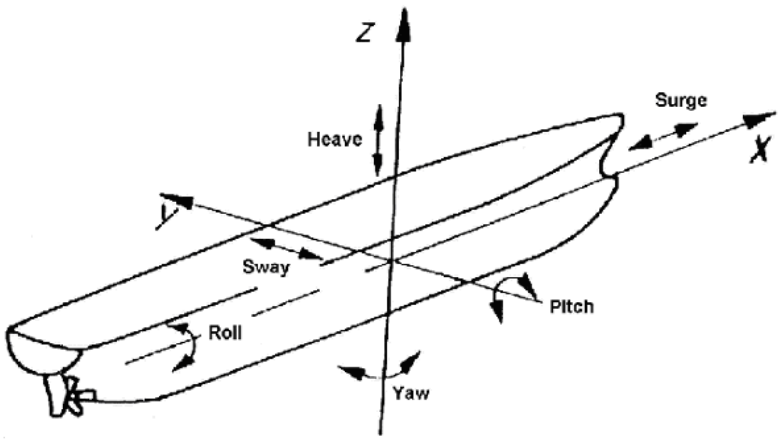


Figure 2.1: Definition of coordinate system to describe the motion of a ship or offshore structure [Benedict et al., 2003].

The motion of an arbitrary point of the vessel when the motions are small can be found from the following relation [Pettersen, 2014].

$$s = \eta_1 \boldsymbol{i} + \eta_2 \boldsymbol{j} + \eta_3 \boldsymbol{k} + \omega \times \boldsymbol{r} \tag{2.1}$$

Where ω and $\mathbf{r}$ are defined as follows:

$$\begin{aligned} \omega &= \eta_4 \boldsymbol{i} + \eta_5 \boldsymbol{j} + \eta_6 \boldsymbol{k} \\ \boldsymbol{r} &= x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k} \end{aligned} \tag{2.2}$$

Hence, one needs to find the six motions to know the motion of a vessel.

2.1.2 Frequency Domain Analysis

The solution to a hydrodynamic problem is found by describing the fluid velocity and pressure fields at the area of interest. The velocity potential is a mathematical expression which is capable of describing the velocity components and thereby also provides information about the pressure field in the fluid [Çengel, 2014]. The velocity potential for general marine problems is found by solving the Laplace Equation and assuming potential flow conditions; irrotational, inviscid and incompressible. The

boundary conditions for marine applications is illustrated in Figure 2.2.

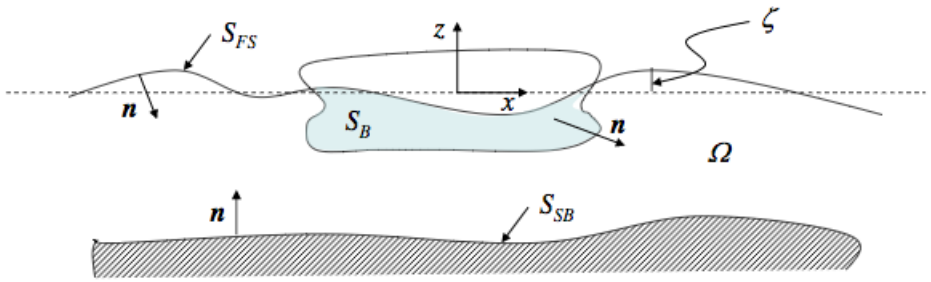


Figure 2.2: Boundary conditions for the Laplace Equation which are utilised to derive the velocity potential and free surface elevation [Greco, 2012].

Where Ω is the fluid domain, ζ is the free surface elevation, $\mathbf{n}$ is the normal and S_{FS}, S_{SB} and S_B are the surfaces of the free surface, sea bed and body respectively. The full derivation of the problem is found in *TMR4215: Sea Loads - Lecture Notes* [Greco, 2012], and the velocity potential and free surface elevation can be expressed as follows:

$$\begin{aligned} \phi &= \phi_1\xi + \phi_2\xi^2 + \phi_3\xi^3 + ... \\ \zeta &= \zeta_1\xi + \zeta_2\xi^2 + \zeta_3 + \xi^3 + ... \end{aligned} \tag{2.3}$$

Where ϕ is the velocity potential, ζ is the free surface elevation and ξ are the non-linearities in the problem.

The linear theory assumes small non-linearities which leads to neglecting the higher order terms, $\mathcal{O}(\xi^n)$, where $n < 1$ [Greco, 2012]. With the assumption of only using the first order terms, the space and time dependence can be split. This leads to the problem only being dependent on the frequency, hence a frequency domain analysis.

2.1.3 Hydrodynamic Loads

Assuming linearity, steady state and linear potential theory, the hydrodynamic loads can be found by decomposing the velocity potential, and use the superposition

principle [Greco, 2012]. The wave-body interaction problem is decomposed into two sub-problems, the excitation forces and the radiation forces as illustrated in Figure 2.3.

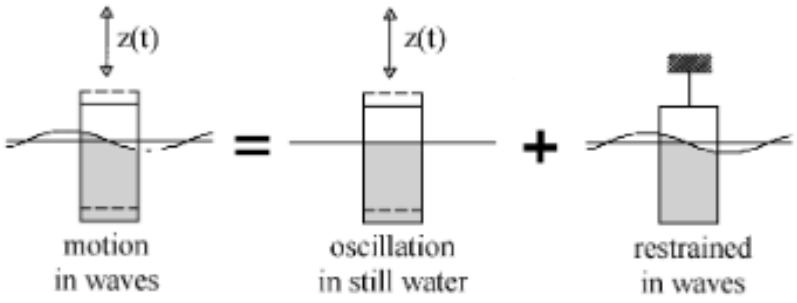


Figure 2.3: Vessel response in regular waves with the two sub-problems - restrained from oscillation and forced oscillations [Journée and Pinkster, 2002].

Excitation forces are when the body is restrained from oscillating, but there are incoming waves which lead to an excitation force. The radiation problem is when the body is oscillating with the wave excitation frequency without incident waves. This means that the problem is split into either the body is moving or the water is moving, and the two sub-problems are added together to obtain the full body response due to the assumption of linearity [Faltinsen, 2006].

The excitation force is divided into two effects, Froude-Kriloff force and diffraction force [Journée and Pinkster, 2002]. Both force contributions are pressure contributions, but the Froude-Kriloff is the undisturbed wave while the diffraction force is due to the presence of the body in the wave. Froude-Kriloff is hence the undisturbed pressure field on the body which is calculated from the dynamic pressure in linear wave theory [Faltinsen, 1990]. The undisturbed pressure field in the Froude-Kriloff force needs to be corrected to include the presence of the body. This is the diffraction force, which is the pressure field set up by the structure such that the total velocity normal to the body is zero. The Froude-Kriloff and diffraction force are calculated using the velocity potentials as follows [Greco, 2012].

$$F_{exc,k}(t) = -\int_{S_{0B}} \rho \frac{\partial \phi_0}{\partial t} n_k dS - \int_{S_{0B}} \rho \frac{\partial \phi_D}{\partial t} n_k dS \tag{2.4}$$

The radiation problem involves added mass, damping and restoring forces and moments. These come from the radiating waves created by the body oscillation which causes oscillating fluid pressures on the body surface [Faltinsen, 1990]. The added mass is in counter phase with the ship motion acceleration and the damping is in counter phase with the ship motion velocity. The restoring forces and moments on the body are due to variations of the hydrodynamic pressure, $p = -\rho g z$. The added mass and damping terms are found from the dynamic pressure due to the body motions as follows:

$$F_{rad,k}(t) = -\int_{S_{0B}} \rho \frac{\partial \phi_R}{\partial t} n_k dS \tag{2.5}$$

It can be shown that Equation 2.5 can be expressed using the added mass and damping terms and by adding the restoring term, the radiation forces can be found from the following relation:

$$F_{rad,k}(t) = \sum_{j=1}^{6} \big(- A_{kj}\ddot{\eta}_j - B_{kj}\dot{\eta}_j - C_{kj}\eta_j \big) \tag{2.6}$$

Using the hydrodynamic loads and Newtons Second Law, the Equation of Motion is found as follows [Faltinsen, 1990]:

$$\sum_{j=1}^{6} \big[\big(M_{kj} + A_{kj}\big)\ddot{\eta}_j + B_{kj}\dot{\eta}_j + C_{kj}\eta_j \big] = F_k e^{-i\omega_e t} \tag{2.7}$$

Where M_{kj} is the generalised mass, A_{kj} is the added mass, B_{kj} is the damping component and C_{kj} is the restoring component.

2.1.4 Viscous Effects

The hydrodynamic loads and motions are calculated based on the assumption of potential theory. However, potential theory is not capable of describing the flow field inside the boundary layer. When fluid is moving past a body, the no-slip condition leads to the fluid particles closest to the body having the same velocity as the body. This results in a thin layer where the velocity of the fluid particles increases, or decreases, from the body's velocity to the velocity of the outer fluid flow. This thin layer is known as the boundary layer, and is the phenomenon which potential theory is incapable of describing. The adverse pressure gradient due to the skin friction leads to flow separation, which in turn causes vortex shedding. Geometrical singularities, such as sharp edges, also cause flow separation and vortex shedding. The viscous effects are mainly divided into five contributions - the skin friction, eddy making, free surface waves, the lift effect damping and the bilge keel damping. [Chakrabarti, 2001].

The skin friction is due to the no-slip condition along the ship hull, while the eddy making is both flow separation and the vortex shedding due to sharp edges. The free surface wave damping is due to the radiated waves created by the body motions, and the lift damping is due to the lifting moment created by the ship hull. Bilge keels contribute to roll damping due to the normal force on the keel as well as the pressure variations due to the presence of the bilge keels.

Viscous effects are significant in regards to the roll motion if the body sections are nearly circular, and in heave motion for a body with a large draft compared to the diameter [Newman, 1977]. These two cases are both examples of bodies shaped such that the inertial forces are small and the frictional forces therefore become important. The viscous effects are more important for rectangular ship cross-sections compared to conventional ship sections [Himeno, 1981] and the bilge keel damping can account for as much as 50 % of the total damping. If a body is long and slender with a length such that the motion of the body does not coincide with the local wave velocity, the cross-flow drag will be of significance [Newman, 1977]. In other

words, the body is large and therefore moves relative to the waves, unlike a small body which will move with the wave amplitude and frequency. This is the case for bodies with large draft compared to diameter where the horizontal cross-flow will be of significance, or on semi-submersible offshore platforms which will be influenced by a vertical force. In the case of moored structures, the mooring lines are usually long which means there can be viscous effects present on the mooring lines. The mooring lines also restrain the body, which can cause the body to move relative to the waves leading to viscous effects being of significance.

For conventional ship hulls the damping coefficient is approximated using an empirical method mainly developed by Ikeda [Ikeda, 1978]. For unconventional hull shapes these approximations are not sufficiently accurate, and therefore rely on model testing or numerical modelling to estimate the viscous damping. The geometrical singularities around the bilge boxes on the cylindrical FPSO designed by Sevan, illustrated in Figure 1.1, cause the viscous effects to affect the damping, especially in roll and heave. This is also observed in similar structures such as floating wind turbines which also have large geometrical singularities causing vortex shedding and flow separation. This is commonly approximated as a series expansion:

$$B\dot{\eta} = B_1\dot{\eta} + B_2|\dot{\eta}|\dot{\eta} \tag{2.8}$$

Equation 2.8 above includes the non-linear damping term, B_2 which can be found from decay tests.

The added mass can be expressed similarly as the damping and as shown by Muhammad Muklash in *Roll Damping Investigation of Two-Dimensional Ship Section with Bilge-Boxes* [Mukhlas, 2017], the added mass can be extracted from forced oscillation model tests.

Computational Fluid Dynamics is another method of estimating the viscous effects. From the CFD solution, one can extract the added mass and damping terms. However, CFD has trouble approximating the viscous effects of blunt bodies. This will be further discussed in Section 6 where the application of Neural Networks is discussed.

2.1.5 Second-Order Drift Forces

Second-Order wave drift forces are due to the body capability in generating waves. A body generates waves with the same frequency as the incident waves, which can be calculated using the linear theory solution. In the derivations of the linear hydrodynamic loads the higher order terms are neglected. The linear solution has a zero mean and the loads and motions oscillate with the frequency of the incident wave. However, for a moored vessel, the second-order drift forces cannot be neglected. The second-order effects are due to the hydrodynamic pressure of the first-order wave, and can be divided into mean drift forces, difference-wave frequency forces and sum-wave frequency forces.

Drift forces are important in the design of moored structures, such as an FPSO or a floating wind turbine. In Section 2.1.2 it was assumed that the higher order non-linear terms could be neglected. However, as the waves grow steeper the second order terms can no longer be neglected due to the second-order drift forces. When the second-order effects are included, $\mathcal{O}(\xi^2)$, the error in the approximation is of order $\mathcal{O}(\xi^3)$ [Greco, 2012].

The solution of the second-order effects are mean forces, forces oscillating with the sum-frequency and forces oscillating with the difference-frequency [Faltinsen, 1990]. The sum-frequency is the behaviour which is due to $\omega_i + \omega_j$ while the difference-frequency effects are due to $\omega_i - \omega_j$. The difference-frequency is commonly around the resonance period of moored structures, while the sum-frequency can excite resonance oscillations in heave, pitch and roll for a TLP.

There are viscous effects on the mean wave forces which are proportional to the cube of the wave amplitude, which means these drift forces are of third order. These effects are of significance for semi-submersibles where the incident wave amplitude is large relative to the cross-sectional dimensions [Faltinsen, 1990].

2.1.6 Irregular Waves

The hydrodynamic loads and motions discussed above are all assumed for regular waves which are waves describes with a sine, or cosine, function. The waves in the ocean are not regular and several regular wave combinations are used to describe the irregular waves. By adding together regular waves with different wave amplitude, wave length and direction of propagation, the real sea state is described [Faltinsen, 1990]. Due to the assumption of linearity, the superposition principle is valid and the regular waves can be summarised to describe the irregular waves.

The principal assumption that linear superposition applies lead to the following [Newman, 1977]: "The body response in irregular waves can be found by superposition of the response in regular waves". In other words, by summing the contributions of regular waves with different wave amplitude, wave length and propagation direction, the body response in irregular wave can be found.

2.1.7 Derivation of the Equations of Motion

In the following section the derivation of coupled motion in sway, heave and roll will be shown. Calculation of the natural periods in both coupled and uncoupled motions will also be presented. The section is included to show the complexity of the problem, how it is commonly approximated and why the present study is of interest.

Non-Linear Coupled Motion: Sway, Heave and Roll

The motions defined in Section 2.1.1 are coupled motions, meaning they depend on each other. In the following section the non-linear coupled motions will be shown,

which is already known in literature. When the motions are non-linear, meaning linear potential theory is not assumed, heave is also coupled with sway and roll. Due to symmetry, surge, pitch and yaw can be uncoupled from the system.

The components of the inertial force which are associated with the body mass are written in the mass matrix [Newman, 1977]. Only the sway, heave and roll terms are considered.

$$M_{jk} = \begin{bmatrix} M & 0 & -Mz_G \\ 0 & M & 0 \\ -Mz_G & 0 & I_{44} \end{bmatrix} \tag{2.9}$$

Similarly the added mass and damping coefficient matrices are simplified to the following:

$$A_{jk} = \begin{bmatrix} A_{22} & A_{23} & A_{24} \\ A_{32} & A_{33} & A_{34} \\ A_{42} & A_{43} & A_{44} \end{bmatrix} \tag{2.10}$$

$$B_{jk} = \begin{bmatrix} B_{22} & B_{23} & B_{24} \\ B_{32} & B_{33} & B_{34} \\ B_{42} & B_{43} & B_{44} \end{bmatrix} \tag{2.11}$$

Each of the coefficients in the matrices involve non-linear terms which can be written as seen in Equation 2.8. The restoring coefficient matrix includes the restoring force from the springs in the system, C_{22}. These springs are either restoring forces in form

of anchor lines or springs in an experimental setup.

$$C_{jk} = \begin{bmatrix} C_{22} & 0 & C_{24} \\ 0 & C_{33} & 0 \\ C_{42} & 0 & C_{44} \end{bmatrix} \tag{2.12}$$

The time derivatives of sway, heave and roll are defined as follows:

$$\eta_j = \begin{bmatrix} \eta_2 \\ \eta_3 \\ \eta_4 \end{bmatrix} \tag{2.13}$$

$$\dot{\eta_j} = \begin{bmatrix} \dot{\eta_2} \\ \dot{\eta_3} \\ \dot{\eta_4} \end{bmatrix} \tag{2.14}$$

$$\ddot{\eta_j} = \begin{bmatrix} \ddot{\eta_2} \\ \ddot{\eta_3} \\ \ddot{\eta_4} \end{bmatrix} \tag{2.15}$$

The equations of the coupled motions then become:

$$\left(\begin{bmatrix} M & 0 & -Mz_G \\ 0 & M & 0 \\ -Mz_G & 0 & I_{44} \end{bmatrix} + \begin{bmatrix} A_{22} & A_{23} & A_{24} \\ A_{32} & A_{33} & A_{34} \\ A_{42} & A_{43} & A_{44} \end{bmatrix} \right) \begin{bmatrix} \ddot{\eta_2} \\ \ddot{\eta_3} \\ \ddot{\eta_4} \end{bmatrix} + \begin{bmatrix} B_{22} & B_{23} & B_{24} \\ B_{32} & B_{33} & B_{34} \\ B_{42} & B_{43} & B_{44} \end{bmatrix} \begin{bmatrix} \dot{\eta_2} \\ \dot{\eta_3} \\ \dot{\eta_4} \end{bmatrix} + \begin{bmatrix} C_{22} & 0 & 0 \\ 0 & C_{33} & 0 \\ 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \eta_2 \\ \eta_3 \\ \eta_4 \end{bmatrix} = \begin{bmatrix} F_2(t) \\ F_3(t) \\ F_4(t) \end{bmatrix} \tag{2.16}$$

Equation 2.16 above is written with simple terms for added mass, damping and restoring coefficient. However, due to the non-linearity, these will in fact involve higher order terms from the higher order solution of the velocity potential. For the damping terms this is equivalent to $B_{jk}\dot{\eta_j} = B_{jk,1}\dot{\eta_j} + B_{jk,2}|\dot{\eta_j}|\dot{\eta_j}$ and similar relations can be developed for the other terms. This leads to a very complex mathematical expression which is hard to solve and simplifications are made to solve the problem.

Linear Coupled Motion: Sway and Roll

Equation 2.16 is simplified by assuming linear, coupled motion. Due to the linear potential theory, heave motion is no longer coupled with sway and roll. Linearity also neglects the higher order terms, hence, the coefficients now only involve the single terms A_{jk}, B_{jk} and C_{jk} and no non-linear contributions. This derivation was conducted in the feasibility study [Bremer, 2017].

The mass coefficient matrix simplifies to only including the roll and sway values.

$$M_{jk} = \begin{bmatrix} M & -Mz_G \\ -Mz_G & I_{44} \end{bmatrix} \tag{2.17}$$

The same way as the mass coefficient matrix was simplified, the added mass, damping and restoring coefficient matrices can be simplified.

$$A_{jk} = \begin{bmatrix} A_{22} & A_{24} \\ A_{42} & A_{44} \end{bmatrix} \quad (2.18) \qquad B_{jk} = \begin{bmatrix} B_{22} & B_{24} \\ B_{42} & B_{44} \end{bmatrix} \quad (2.19) \qquad C_{jk} = \begin{bmatrix} C_{22} & 0 \\ 0 & C_{44} \end{bmatrix} \quad (2.20)$$

Sway and roll are defined with their time derivatives:

$$\eta_j = \begin{bmatrix} \eta_2 \\ \eta_4 \end{bmatrix} \quad (2.21) \qquad \dot{\eta_j} = \begin{bmatrix} \dot{\eta_2} \\ \dot{\eta_4} \end{bmatrix} \quad (2.22) \qquad \ddot{\eta_j} = \begin{bmatrix} \ddot{\eta_2} \\ \ddot{\eta_4} \end{bmatrix} \quad (2.23)$$

And the equation of motion as mentioned in Section 2.1.3.

$$\big(M_{jk} + A_{jk}\big)\ddot{\eta_j} + B_{jk}\dot{\eta_j} + C_{jk}\eta_j = F_j(t) \tag{2.24}$$

Which can be written in matrix form with the mass matrix, added mass, damping

and restoring matrices.

$$\left(\begin{bmatrix} M & -Mz_G \\ -Mz_G & I_{44} \end{bmatrix} + \begin{bmatrix} A_{22} & A_{24} \\ A_{42} & A_{44} \end{bmatrix} \right) \begin{bmatrix} \ddot{\eta_2} \\ \ddot{\eta_4} \end{bmatrix} + \begin{bmatrix} B_{22} & B_{24} \\ B_{42} & B_{44} \end{bmatrix} \begin{bmatrix} \dot{\eta_2} \\ \dot{\eta_4} \end{bmatrix} + \begin{bmatrix} C_{22} & 0 \\ 0 & C_{44} \end{bmatrix} \begin{bmatrix} \eta_2 \\ \eta_4 \end{bmatrix} = \begin{bmatrix} F_2(t) \\ F_4(t) \end{bmatrix} \quad (2.25)$$

In order to find the sway and roll motion of a structure Equation 2.25 is solved with respect to sway, η_2, and roll, η_4.

Coupled Natural Period

The coupled natural period can be found from Equation 2.25 by assuming undamped and no excitation force. The expression then simplifies to:

$$\left(\begin{bmatrix} M & -Mz_G \\ -Mz_G & I_{44} \end{bmatrix} + \begin{bmatrix} A_{22} & A_{24} \\ A_{42} & A_{44} \end{bmatrix} \right) \begin{bmatrix} \ddot{\eta_2} \\ \ddot{\eta_4} \end{bmatrix} + \begin{bmatrix} C_{22} & 0 \\ 0 & C_{44} \end{bmatrix} \begin{bmatrix} \eta_2 \\ \eta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.26)$$

The solution of roll and sway is assumed to be of the form $\eta_j = \eta_{ja} e^{iwt}$, which has the following derivative

$$\ddot{\eta_j} = -\omega^2 \eta_{ja} e^{iwt} \quad (2.27)$$

Because Equation 2.26 only consists of η_j and $\ddot{\eta_j}$ the term e^{iwt} can be divided out of the equation.

$$\left(\begin{bmatrix} M & -Mz_G \\ -Mz_G & I_{44} \end{bmatrix} + \begin{bmatrix} A_{22} & A_{24} \\ A_{42} & A_{44} \end{bmatrix} \right) \begin{bmatrix} (-\omega^2)\, \eta_{2a} \\ (-\omega^2)\, \eta_{4a} \end{bmatrix} + \begin{bmatrix} C_{22} & 0 \\ 0 & C_{44} \end{bmatrix} \begin{bmatrix} \eta_{2a} \\ \eta_{4a} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.28)$$

Gathering the expressions:

$$\begin{bmatrix} (-\omega^2)\,(M+A_{22}) & (-\omega^2)\,(A_{24}-Mz_G) \\ (-\omega^2)\,(A_{42}-Mz_G) & (-\omega^2)\,(I_{44}+A_{44}) \end{bmatrix} \begin{bmatrix} \eta_{2a} \\ \eta_{4a} \end{bmatrix} + \begin{bmatrix} C_{22} & 0 \\ 0 & C_{44} \end{bmatrix} \begin{bmatrix} \eta_{2a} \\ \eta_{4a} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.29}$$

Further gathering:

$$\begin{bmatrix} (-\omega^2)\,(M+A_{22})+C_{22} & (-\omega^2)\,(A_{24}-Mz_G) \\ (-\omega^2)\,(A_{42}-Mz_G) & (-\omega^2)\,(I_{44}+A_{44})+C_{44} \end{bmatrix} \begin{bmatrix} \eta_{2a} \\ \eta_{4a} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.30}$$

The natural frequency can then be found by solving for the determinant of the left matrix which shall be zero.

$$\Big[\big((-\omega^2)\,(M+A_{22})+C_{22}\big)\times\big((-\omega^2)\,(I_{44}+A_{44})+C_{44}\big)\Big]-\Big[(-\omega^2)(A_{24}-Mz_G)\times(-\omega^2)(A_{42}-Mz_G)\Big]=0 \tag{2.31}$$

Multiplying out of the parentheses:

$$\begin{aligned} \big[&\omega^4 MI_{44}+\omega^4 MA_{44}-\omega^2 MC_{44}+\omega^4 A_{22}I_{44}+\omega^4 A_{22}A_{44} \\ &-\omega^2 A_{22}C_{44}-\omega^2 C_{22}I_{44}-\omega^2 A_{44}C_{22}+C_{22}C_{44}\big]- \\ \big[&\omega^4 A_{24}A_{42}-\omega^4 A_{24}Mz_G-\omega^4 Mz_G A_{42}+\omega^4 M^2 z_G^2\big]=0 \end{aligned} \tag{2.32}$$

Remove parenthesis:

$$\begin{aligned} &\omega^4 MI_{44}+\omega^4 MA_{44}-\omega^2 MC_{44}+\omega^4 A_{22}I_{44}+\omega^4 A_{22}A_{44}-\omega^2 A_{22}C_{44}-\omega^2 C_{22}I_{44} \\ &-\omega^2 A_{44}C_{22}+C_{22}C_{44}-\omega^4 A_{24}A_{42}+\omega^4 A_{24}Mz_G+\omega^4 Mz_G A_{42}-\omega^4 M^2 z_G^2=0 \end{aligned} \tag{2.33}$$

The expressions with ω^4 and ω^2 are gathered:

$$\begin{aligned} \omega^4\big(&MI_{44}+MA_{44}+A_{22}I_{44}+A_{22}A_{44}+A_{24}A_{42}+A_{24}Mz_G+Mz_G A_{42}-M^2 z_G^2\big)+ \\ &\omega^2\big(-MC_{44}-A_{22}C_{44}-C_{22}I_{44}-A_{44}C_{22}\big)+C_{22}C_{44}=0 \end{aligned} \tag{2.34}$$

Equation 2.34 can be written as the following expression

$$\lambda^2 A + \lambda B + C = 0 \tag{2.35}$$

Where the constants A, B and C are defined as

$$\begin{aligned} A &= MI_{44} + MA_{44} + A_{22}I_{44} + A_{22}A_{44} + A_{24}A_{42} + A_{24}Mz_G + Mz_GA_{42} - M^2z_G^2 \\ B &= -MC_{44} - A_{22}C_{44} - C_{22}I_{44} - A_{44}C_{22} \\ C &= C_{22}C_{44} \\ \lambda &= \omega^2 \end{aligned} \tag{2.36}$$

The solution of Equation 2.34 is of the form

$$\lambda = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{2.37}$$

The natural frequency of interest are only positive, thus the natural frequencies of the coupled, undamped system are:

$$\begin{aligned} \omega_1 &= \sqrt{\frac{-B + \sqrt{B^2 - 4AC}}{2A}} \\ \omega_2 &= \sqrt{\frac{-B - \sqrt{B^2 - 4AC}}{2A}} \end{aligned} \tag{2.38}$$

The natural periods can then be found from the two following expressions with the defined A, B and C values as in Equation 2.36.

$$\begin{aligned} T_1 &= 2\pi\sqrt{\frac{2A}{-B + \sqrt{B^2 - 4AC}}} \\ T_2 &= 2\pi\sqrt{\frac{2A}{-B - \sqrt{B^2 - 4AC}}} \end{aligned} \tag{2.39}$$

For the above derivation damping is assumed zero and the two coupled natural periods are found. The natural periods for the damped and coupled motion can be found by starting with Equation 2.24 with the damping terms. The solution for the motions are then of the form $\eta_j = \eta_{ja}e^{st}$ where $s = \alpha t + i\beta t$. The undamped solution is only the term $e^{i\beta t}$ which are the oscillations. The term $e^{\alpha t}$, which is included in the damped solution, is the function which reduces the amplitude of the oscillations.

Uncoupled Natural Period

Assuming the motions are uncoupled and undamped the natural periods can be found the same way as the coupled natural periods were found. The uncoupled natural periods have the form of Equation 2.40.

$$T_{ni} = 2\pi\sqrt{\frac{M_{ii} + A_{ii}}{C_{ii}}} \tag{2.40}$$

For an unmoored structure there are no natural periods in surge, sway and yaw [Faltinsen, 1990] because there are no restoring forces. However, in the experiments the model will be freely floating with springs attached. The springs will act the same way as moorings and will contribute with restoring forces to the structure.

$$T_2 = 2\pi\sqrt{\frac{M + A_{22}}{C_{22}}}\,, \quad T_3 = 2\pi\sqrt{\frac{M + A_{33}}{C_{33}}}\,, \quad T_4 = 2\pi\sqrt{\frac{I_{44} + A_{44}}{C_{44}}} \tag{2.41}$$

The difference between the coupled and uncoupled natural periods are the coupled terms such as A_{24}, A_{42} and Mz_G.

From the derivations in Section 2.1.7 it is seen that the calculation of a vessel response is both complex and challenging. Several assumptions are utilised to estimate the response which leads to neglecting effects which can be of importance.

2.2 An Introduction to Machine Learning

Learning is the art of getting better at a task over a period of time. Machine learning, a subcategory of Artificial Intelligence, is when computers learn in a similar way as humans. These learning algorithms are divided into three categories - *supervised learning*, *unsupervised learning* and *reinforcement learning*. Supervised learning is where a machine is trained on known input and output data, in order to predict the output of new and unseen input data. In unsupervised learning, the computer analyses input data and has to find a pattern in the data without external supervision in order to sort the output in classes or categories. Reinforcement learning are techniques where the model is trained to take actions by giving the model rewards or punishment. Some of the most common applications of the algorithms today are image and speech recognition, medical diagnosis, statistical arbitrage, weather prediction and spam e-mail classification.

As a small example - look at the numbers listed below

$$2,\ 4,\ 6,\ 8,\ ? \tag{2.42}$$

The human brain knows from experience that the next number in the sequence is ten. This kind of behaviour, learning from experience, is the goal of Machine Learning. Teaching the computer to learn from experience and being able to predict the next number in the sequence.

As the problem at hand is a *supervised* learning problem, only supervised learning algorithms will be explained in further detail.

2.2.1 Supervised Learning

A supervised learning algorithm is given known input and output data and trained to predict the output. This is similar to finding the function which maps the input x to the output y, known as $F(x)$. Supervised learning algorithms are either used to solve

classification or regression problems. Classification problems are discrete problems where the input has to be labelled into the right category [Pedregosa et al., 2011]. Examples of classification problems is sorting e-mails as spam or recognising hand written digits. A regression problem in supervised learning is when the output is continuous, such as predicting house prices.

Predicting the motion of a vessel requires predicting a continuous output, which is why a regression method is necessary. In the following sections Neural Networks will be presented, however, to ensure proper background knowledge, the classical linear regression method is briefly presented.

2.2.2 Linear Regression

Linear regression models assume there exists a linear relationship between the input x and the output y. The model either assumes that the regression function $E(Y|X)$ is linear or that assuming a linear model is a reasonable approximation [Hastie, 2009].This linear approximated relationship can be written as seen in Equation 2.43 below.

$$y = b_0 + b_1 x \tag{2.43}$$

Where:

- y is the output
- x is the input
- b_0 is the interception point

In Equation 2.43 above the relationship between the input data and the output data is deterministic, meaning it is exact. This is generally not the case in engineering or scientific problems, which is why there is a need of more input variables to further explain the output prediction. Multiple linear regression explains the relationship between one output variable and two or more input variables. The multiple regression is known as seen in Equation 2.44 [Ziegel et al., 1999] and is used to calculate

the best fit line with more information and thereby more accurately predict the outcome.

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n \tag{2.44}$$

Where:

- y is the output
- x are the input variables
- b_0 is the interception point
- b_n is the coefficient of input variable n

The coefficients b are estimated from the known data set $(x_1, y_1)...(x_n, y_n)$ using a *cost function*, also known as *loss function*. A cost function is a function which measures the inaccuracy of the prediction. There exist different cost functions which can be implemented in the linear regression model and one of the most popular methods is the *least squares method* [Hastie, 2009]. The least squares method utilises the residuals, which is the error in a fitted model, to calculate the sum of the square of the residuals as seen in Equation 2.45. The least squares calculation is minimised to optimise the linear regression model.

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 \tag{2.45}$$

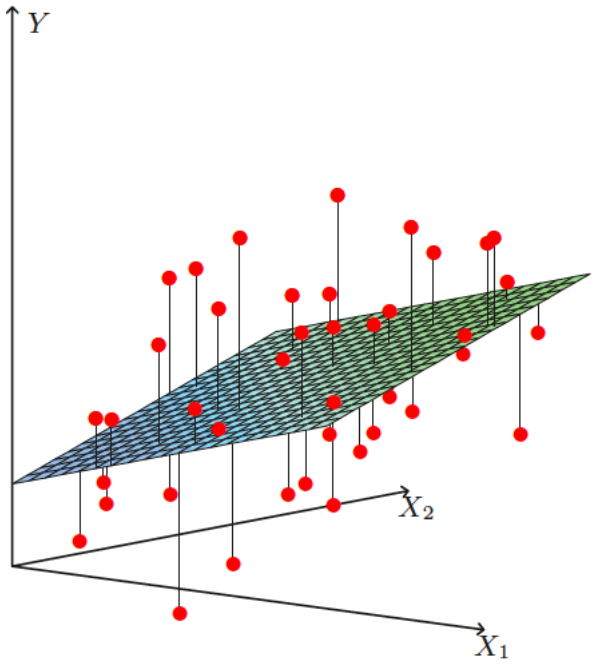


Figure 2.4: The best fit plane for the given data points where the sum of least squares is optimised. The distance between the data points to the best fit plane is the loss of each data point. [Hastie, 2009]

The cost function calculates the loss in the model prediction, and the model is therefore optimised by finding the line, plane or hyperplane which gives the smallest loss. In the case where the least squares method is used to calculate the loss, the optimal plane would be where the sum of least squares is the smallest. In Figure 2.4 above, the best fit plane is shown together with the data points and their losses.

2.2.3 Artificial Neural Networks

Artificial Neural Networks are machine learning algorithms which are inspired by the human brain. A neural network consists of nodes, or *neurons*, which are connected and sorted in layers. There are different ways of connecting the nodes between layers, which create different kinds of neural networks. How the weights and biases are optimised is another difference and this leads to a huge amount of algorithms which all are neural networks. The general approach of a feed forward neural network is presented in this section.

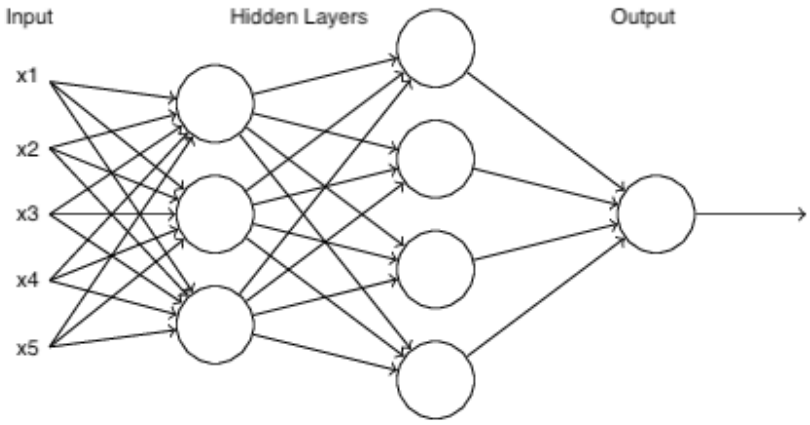


Figure 2.5: Illustration of a Neural Network with two hidden layers, input layer and output layer. The fully-connected Neural Network shows how all neurons in each layer are connected to both the neurons in the previous layer and the next layer [Nielsen, 2015].

In Figure 2.5 a neuron is illustrated as a circle and the input layer is illustrated by the inputs x. The two hidden layers are the columns with neurons in the middle and the output layer is the final neuron at the right. The Neural Network in the illustration is a three layer Neural Network, meaning the input layer does not count as a layer because it can only receive input. The neurons in the hidden layer calculate values which will never be seen by the programmer, which is why these layers are called hidden. If a Neural Network has more than one hidden layer it is known under the category Deep Learning.

Neuron

The Neurons in the Artificial Neural Networks are inspired by the human brain. In the human brain there are neurons which either send a signal or not, given the processed input. This is the behaviour an artificial neuron is trying to mimic. An artificial neuron, from now on referred to as a neuron, passes a value further in to the Neural Network using two calculation steps - summation and activation function.

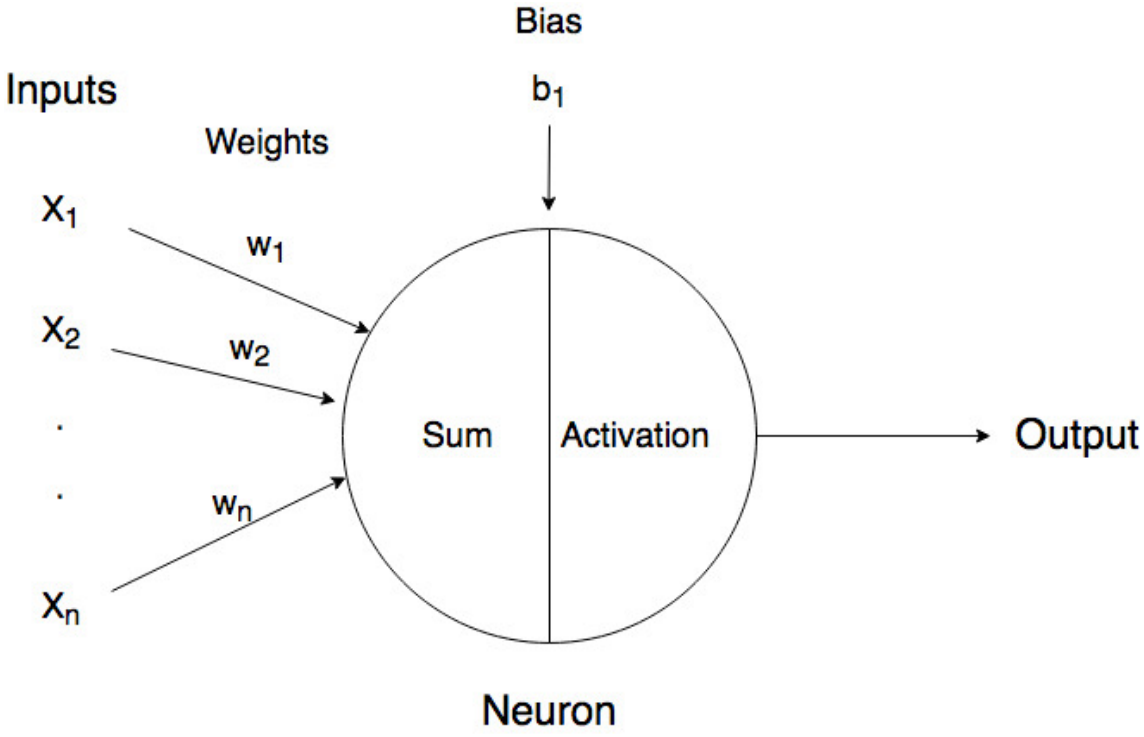


Figure 2.6: Illustration of a single neuron which receives inputs with weights and bias, sums all inputs and utilises an activation function to calculate a scalar value which is sent as output from the neuron.

Figure 2.6 illustrates a single neuron with inputs, connections, bias and output. Each neuron receives inputs which are weighted, illustrated by the arrow connections. The input is written as x, weight as w, bias as b. Each neuron calculates the sum of the weighted inputs and ads the bias of the neuron. The bias is set for the individual neuron. Every neuron then uses the weighted sum and bias to calculate a scalar value, which is used as input for an activation function in the second step in the neuron. The scalar value after the activation function is the output of the neuron. The sum is calculated as shown in Equation 2.46.

$$z = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b_1 \tag{2.46}$$

A complete Neural Network will have many neurons and layers, and the above notation is therefore simplified to indicate both layer and neuron in the specific layer. A simple fully connected Neural Network is illustrated in Figure 2.7.

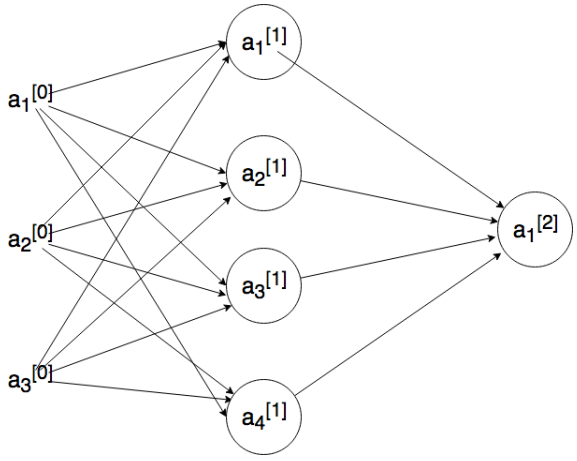


Figure 2.7: An illustration of a one-layer Neural Network where the subscript indicates which neuron it is in the layer. The superscript indicates which layer in the Neural Network.

In Figure 2.7 the input is written as $a_n^{[0]}$, where the square brackets indicate the layer and the subscript indicates the neuron in the layer. The input layer is commonly named layer 0, because there are no calculations done in the input layer. The weighted sum calculated in the first node in the hidden layer, layer 1, is calculated as in Equation 2.47.

$$z_1^{[1]} = w_{1,1}^{[1]} a_1^{[0]} + w_{1,2}^{[1]} a_2^{[0]} + w_{1,3}^{[1]} a_3^{[0]} + b_1^{[1]} \tag{2.47}$$

With a growing number of neurons and layers this calculation becomes large, but the system of equations can be nicely written in vector form. For the hidden layer, layer 1, the following matrix describes the summation step of the neuron calculations.

$$\begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} = \begin{bmatrix} w_{1,1}^{[1]1} & w_{1,2}^{[1]} & w_{1,3}^{[1]} & w_{1,4}^{[1]} \\ w_{2,1}^{[1]1} & w_{2,2}^{[1]} & w_{2,3}^{[1]} & w_{2,4}^{[1]} \\ w_{3,1}^{[1]1} & w_{3,2}^{[1]} & w_{3,3}^{[1]} & w_{3,4}^{[1]} \\ w_{4,1}^{[1]1} & w_{4,2}^{[1]} & w_{4,3}^{[1]} & w_{4,4}^{[1]} \end{bmatrix} \begin{bmatrix} a_1^{[0]} \\ a_2^{[0]} \\ a_3^{[0]} \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \end{bmatrix} \tag{2.48}$$

By writing the weights as a matrix and the input and bias as vectors and generalised for the number of layer, Equation 2.48 can be simplified to the following:

$$z^{[n]} = W^{[n]}a^{[n-1]} + b^{[n]} \tag{2.49}$$

Activation Function

After the neuron has calculated the weighted sum, z, an activation function is used to decide if the neuron is activated or not. If the neuron is activated, the neuron *fires* and the output is sent to the next neuron. If the neuron is not activated, the neuron does not fire. Without the activation function in the neurons, this would simply be a linear regression model as presented in Section 2.2.2. There are many options when choosing an activation function, but it is important to choose a non-linear activation function [Ketkar, 2017]. If it is a linear function, the hidden layers of the Neural Network lose their advantage and the network will perform the same way as a single-layer network. With a non-linear activation function, the neural network is theoretically proven to approximate any function. Three activation functions are plotted in Figure 2.8.

Other characteristics of the activation function are listed in the 2.2.3 [Ketkar, 2017]:

- In order to use gradient-methods to optimise the model, the activation function has to be continuous

- Activation functions with a finite range have a more stable performance

- Activation functions are typically symmetric around the origin

Two of the most common activation functions for regression Neural Networks are the Sigmoid Function and Tangent Hyperbolic. Both functions are continuous, have a finite range and are symmetric around the origin. The Sigmoid Function is defined as shown in Equation 2.50, including the derivative of the function.

$$f(z) = \frac{1}{1+e^{-z}}\,, \quad f'(z) = \frac{1}{1+e^{-z}} \cdot \big(1 - \frac{1}{1+e^{-z}}\big) \tag{2.50}$$

The Tangent Hyperbolic derivative is shown in Equation 2.51 as well as the derivative.

$$f(z) = \tanh(z)\,, \quad f'(z) = 1 - \tanh(z)^2 \tag{2.51}$$

The main difference between the two functions is the range, where Sigmoid has a range between $[0, 1]$ while Tangent Hyperbolic is in range $[-1, 1]$. Both Sigmoid and Tanh will cause the neurons to almost always be activated to process the output. The two functions have gradual changes and the output are float numbers between the ranges. This is the same as the neuron answering *maybe* to the input given [Gulli and Pal, 2017]. In a large neural network with many layers and neurons, this would lead to a time consuming and costly computation. To avoid all neurons being activated, a Rectified Linear Unit activation function can be utilised.

$$f(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}\,, \quad f'(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases} \tag{2.52}$$

The ReLU function equals z for values larger than zero and is zero otherwise as seen in Equation 2.52 above. This function is also non-linear, but the range of ReLU is $[0, \text{inf}]$. ReLU is limited because gradients tend towards zero, which means that the neurons will stop responding to new input because of the gradient being zero, meaning there are no changes.

Due to the different characteristics of the activation functions it is common to combine and use different activation functions in different layers. There is no specific answer to why, and which, function works best in a given problem and this is part of the further development of using Neural Networks. Figure 2.8 shows the three activation functions.

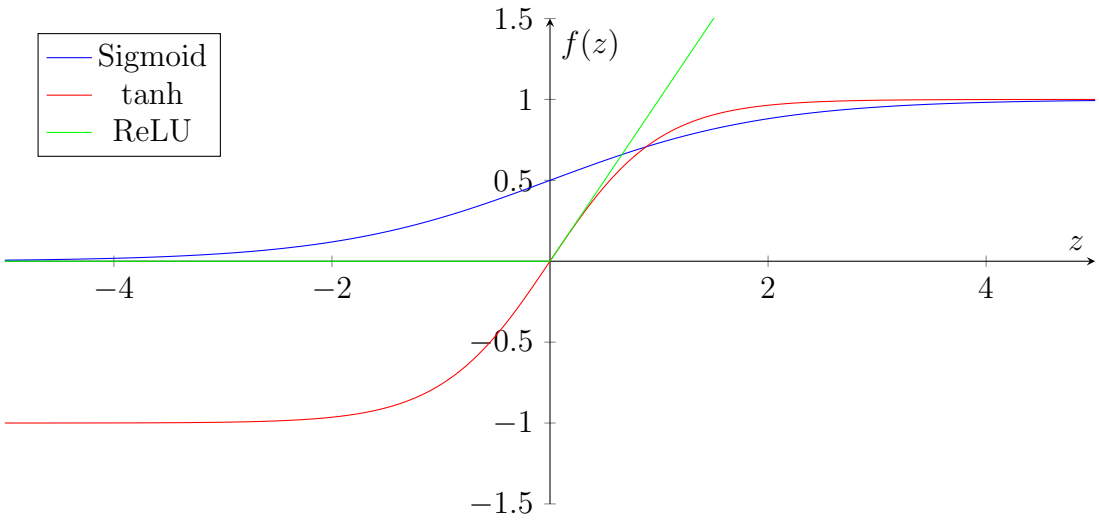


Figure 2.8: Activation Functions: Sigmoid, ReLU and Tangent Hyperbolic. The three activation functions are commonly utilised in Neural Networks and have different characteristics.

2.2.4 Loss Function and Optimisation

Loss functions are used to calculate the disagreement between the predicted value and the actual value [Ketkar, 2017]. The loss function is used to optimise the weights and the process of optimisation is often known as loss minimisation. There are several common loss functions and one of the most used loss functions for regression models is the *Mean Squared Error* which is seen in Equation 2.53 and is recommended for use in regression problems [Ketkar, 2017].

$$MSE = \frac{\sum \left(y_i - \hat{y}_i\right)^2}{N} \tag{2.53}$$

Where y_i is the true value, $\hat{y}_i$ is the predicted value and N is the number of data points.

The optimisation algorithms use the loss function to optimise the weights in the Neural Network. This is done by minimising the loss calculated with the loss function, thus minimising the error. Two common optimisation algorithms are the *Stochastic Gradient Descent* and *Root Mean Square Propagation*.

Stochastic Gradient Descent finds the minimum of the loss function to optimise the weights in the Neural Network by utilising the gradient. The derivative of the loss function is calculated in order to find the direction of decrease and thereby update the weights so the loss is lower in the next iteration. Figure 2.9 illustrates the iteration to find the minimum loss function.

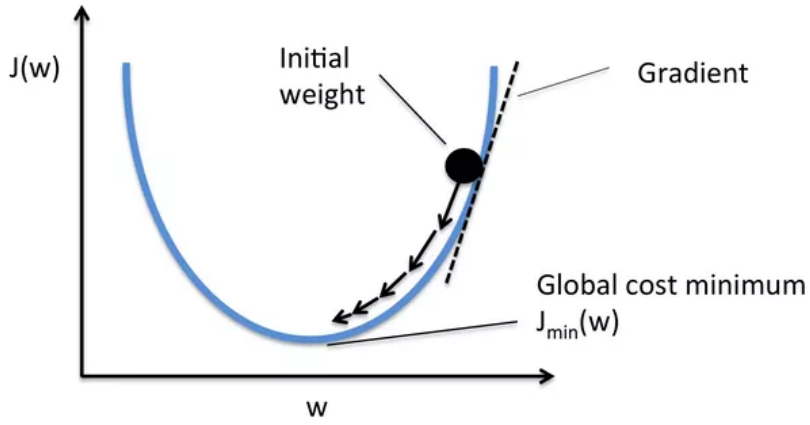


Figure 2.9: Illustration of the optimisation of a loss function. The optimisation algorithm finds the global minimum of the loss function, also known as cost function. [Gulli and Pal, 2017]

The weights are updated with a *learning rate*, which if set too low could lead to the optimisation missing the lowest point of the loss function. If the learning rate is set too high, the learning algorithm will be slow and the model will need a higher number of iterations to reach the minimum point. The mathematical expression for the stochastic gradient descent is presented in Equation 2.54.

$$\theta^{k+1} = \theta^k - \eta \frac{\partial}{\partial \theta^k} J\big(\theta\big) \tag{2.54}$$

Where θ are the weights, η is the learning rate and $J(\theta)$ is the loss function.

The stochastic gradient descent optimisation is used once for each iteration, contrary to optimising each data point. It chooses a random data point in each iteration and optimises the weights with that single data point, which is why it is called *stochastic* gradient descent [Google Developers, 2018].

Root Mean Square Propagation is a further development of stochastic gradient descent which falls under the category of adaptive gradient descent algorithms. Root Mean Square Propagation utilises the gradient as well as dividing by the size of the gradient [Hinton, 2014]. The method is unpublished, but widely used in the field of Machine Learning and was developed by Professor G. Hinton and Professor T. Tieleman at the University of Toronto where it was presented in a Lecture [Hinton, 2014]. Several studies have been successful at using the Root Mean Square Propagation as optimisation algorithm, such as *Training of Deep Neural Networks based on Distance Measures using RMSProp* [Kurbiel, Thomas and Khaleghian, Shahrzad, 2017]. The mathematical expression for the Root Mean Square Propagation is presented below in Equation 2.55.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{RMS[g_{t,i}^2]}} g_{t,i} \tag{2.55}$$

where $g_{t,i}$ and $RMS[g_{t,i}^2]$ are defined as follows

$$\begin{aligned} g_{t,i} &= \frac{\partial J(\theta_{t,i})}{\partial \theta} \\ RMS[g_{t,i}^2] &= 0.9\, RMS[g_{t-1,i}^2] + 0.1\, g_{t,i}^2 \end{aligned} \tag{2.56}$$

The Root Mean Square Propagation method is thus similar to using the Stochastic Gradient Descent, but dividing by a moving squared gradient for each weight [Hinton, 2014].

2.2.5 Neural Network Model

As a short summary, the Neural Network models consists of layers with neurons which are connected. These neurons receive input with weights which is summed and an activation function is used to calculate a scalar value which is sent as the neurons output. As a whole, loss functions are utilised to calculate the error of the output, using the training data where the outputs are known. The Neural Network is optimised by minimising the error of the loss function, thus minimising the error

between the prediction and the known output.

The Neural Network is build using the methods described above. However, it is important to understand that the training of the model is part of the creation of the model and that the final product is a complete, trained model. Hence, if a model has good results on the testing data it can be finalised and saved as a product. Using new test data the saved model can be utilised to predict an output of new data or further trained using more data.

2.2.6 Machine Learning in Practice

The above sections have explained the theoretical background of Linear Regression and Neural Networks. This section will explain the application of these algorithms with an example to introduce machine learning terminology.

Predicting Housing Prices

Prediction of housing prices is a good example where Machine Learning algorithms can be utilised. There are many factors which play a role in the pricing of a house and there is a lot of data available.

Target and Features

The target of the housing price prediction are house prices and is the output of the model. The inputs to a Neural Network are called features. If a person was to approximate a house price he or she would find a similar sized house in the same area and compare. Each house would be an observation and by comparing number of bedrooms, square meters, number of bathrooms and other features, the person would be able to make a suggestion. A feature is thus a factor which affects the house price and is also the input to the Neural Network. The model, just like the human brain, needs information to make a qualified guess. The higher amount of features, the more patterns the algorithm can find and the better approximation. The Neural Network therefore needs to be given features such as number of bed-

rooms, number of bathrooms, location and postal code, number of floors and the year the house was build.

It is important that the features given to the model are independent of each other because every feature needs to give new information to the model. If the model is given too many features the algorithm can fit the data too well, which means it is not generalised and will perform badly on unseen data. An over fitted model has learned the noise and random fluctuations in the seen data, which is why it is said to not be generalised. In Figure 2.10 the right hand side plot illustrates an over fitted algorithm.

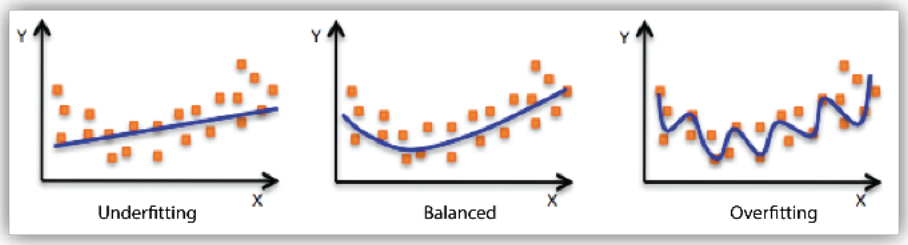


Figure 2.10: Illustration of under and over Fitting. Under fitting is when the model is not capable of predicting the target values, while over fitting is when the model has learned the noise in the input data and is predicting the given data too well. An over fitted model is not predicting unseen data points well, due to learning the noise and is therefore not general enough to predict the target value of unseen data. [Amazon, 2018]

The left hand plot in Figure 2.10 illustrates an under fitted model. If a Neural Network is under fitted, the model is not capable of predicting the seen or unseen data. When a model is under fit, it is not capable of mapping the relation between the features and the target. This commonly happens when there are not enough features to describe the data [Ketkar, 2017].

A model with fifty features and fifty data points would be a perfect fit model, hence, no learning necessary. The model would simply memorise the relations between input and output, but given new, unseen data it would not have learned any patterns and would therefore not be able to predict the output.

Data Set

The training process of a Neural Network is an iteration process where an *epoch* is an iteration of the whole training data set. The input data to the Neural Network is split into training, testing and validation data sets. The training data is the data which is read by the model and used to change the weight of the neurons and consequently train the model. In each epoch iteration of the training, a small data set called test data is used to test the accuracy of the weights in the epoch. This is done to track the development of the model accuracy during the learning process. When the model is fully trained on the training data, the model is validated against the third data set split which is the validation data set. It is important for the validation of the Neural Network that it is validated against data the model has never seen before. This is why the original data set is split into three, training, testing and validation. This final validation shows the accuracy of the Neural Network and its ability to predict or classify correctly.

Feature Scaling

The features in a data set can have a very different scale, for example one feature can range between $0.1 - 1.0$ while another feature can range between $1000 - 10000$. The features in the case of house price prediction will also have different scaling, such as the number of bathrooms and the square meters of a house.

Different scaling of the features slows down the optimisation of the prediction [Ng, 2012]. If the features have a different magnitude, some weights in the neural network can update quicker than others which leads to the model prioritising these weights and they will dominate the prediction. Other machine learning algorithms calculate the distance between points and if a feature has a large distance this feature can dominate the learning process. It is found that scaled features contribute to a faster learning algorithm due to faster optimisation convergence [Ng, 2012].

The features can be *normalised*, meaning it is scaled between 0 and 1. If the feature contains negative values, the normalisation will lead to a range between -1 and 1.

A common way of normalising the features is using Equation 2.57 below.

$$x' = \frac{x_i - min(x)}{max(x) - min(x)} \tag{2.57}$$

Standardisation is another common way of scaling the features where the feature distribution is centred around 0 and the standard deviation is 1. The standardisation of a feature is done using the following equation.

$$x' = \frac{x_i - \mu}{\sigma} \tag{2.58}$$

Where x' is the standardised value, μ is the mean value of x_i and σ is the standard deviation of x_i.

The choice of normalising and standardisation depends on the features and the choice of machine learning algorithm. There is no certain way of knowing which will work best for a given cause and both should be tested to know which suits the problem at hand [Ng, 2012].

Loss Function and Optimisation

The loss function calculates the loss of the prediction compared to the true value. In order to optimise the loss function, meaning finding the all-time low value of the loss function, the gradient is calculated. This can be compared with the way a river always looks for the easiest way down. The optimisation algorithm is used to find where the loss functions decreases the most and change the weights or coefficients to move in that direction.

Model Fitting and Evaluation

The model is fitted by training the model on the given data to predict the target value. The performance is evaluated to find the accuracy of the prediction and compare model configurations. In the house pricing predictions, several configurations of the neural network are possible. In order to find the best configuration of number

of layers and nodes, activation function, loss function and optimisation algorithm the different configurations need to be compared.

For regression there are three common metrics to evaluate the predictions listed below.

- Mean Absolute Percentage Error, $MAPE$
- Mean Squared Error, MSE
- Coefficient of Determination, R^2

$$MAPE = \frac{100\%}{N} \sum \frac{\left| y_i - \hat{y}_i \right|}{y_i}, \quad MSE = \frac{1}{N} \sum \left(y_i - \hat{y}_i \right)^2, \quad R^2 = 1 - \frac{\sum \left(y_i - \hat{y}_i \right)^2}{\sum \left(y_i - \bar{y}_i \right)^2} \tag{2.59}$$

Where y_i is the true value, $\hat{y}_i$ is the predicted value and $\bar{y}_i$ is the mean value of the true values.

When a model with an acceptable performance is built, it can be used to predict the house prices of future houses. The model does not need to be trained for each time it is used, the trained model is the final product which is the tool that can be utilised for future predictions.

Chapter 3

Experimental Method

This section aims to give an understanding of the testing procedures, facilities and test data analysis.

3.1 Preparation Procedures

3.1.1 The Model

The model is a two-dimensional mid-section of a ship with bilge boxes. The main dimensions of the model is seen in Figure 3.1. Its length is almost the same as the width of the tank, and each side of the model has a $0.003 - 0.005$ m gap. The size of the gap depends on the position along the tank, because the tank walls are not perfectly straight. This gap between the model and tank acted as a lubrication layer where the water helped preventing the model from hitting the tank wall. However, due to initial problems with the bilge boxes hitting the tank wall in steep waves, the bilge boxes length was cut to 0.586 m, meaning 2 mm was cut of each end.

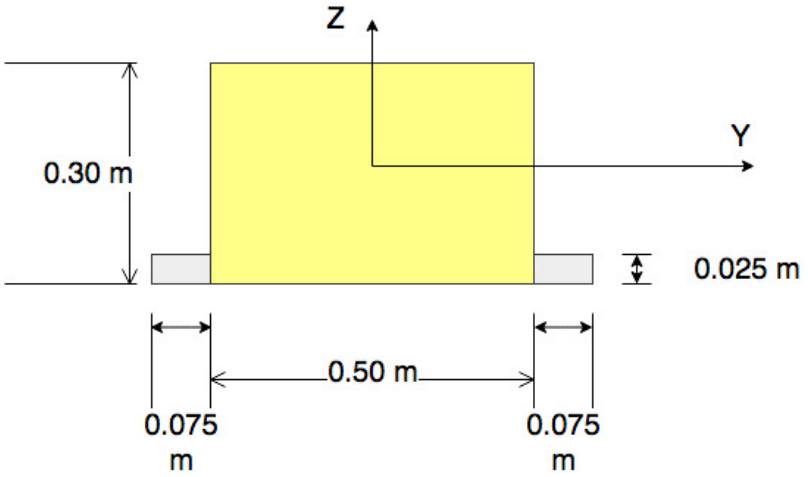


Figure 3.1: Illustration of the side of the model with dimensions and coordinate system. The wave tank extends in y-direction and the waves are propagating in the same direction.

The coordinate system of the model was defined such that the bilge keels follow the length of the full scale ship. The beam of the model is the width, which is in the length direction of the wave tank. Figure 3.1 shows the side of the model, which is the equivalent to the beam of a full scale vessel. The main dimensions of the model are presented in Table 3.1.

Table 3.1: Main dimensions of the Experimental Model with bilge boxes.

Dimension	**Abbreviation**	**Size**
Length	L	0.500 m
Width	B	0.590 m
Height	H	0.300 m
Draft	T	0.148 m
Bilge Length	L_B	0.586 m
Bilge Width	B_B	0.075 m
Bilge Height	H_B	0.025 m

The model was ballasted with 30 kg which were symmetrically distributed on each side of the model as seen in Figure 3.2. The ballast weight was optimised to obtain a suited natural period of the system. A ballast of 30 kg corresponds to a draft of 0.148 m.

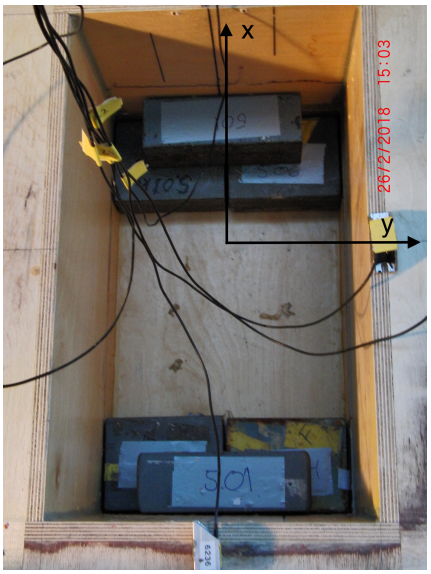
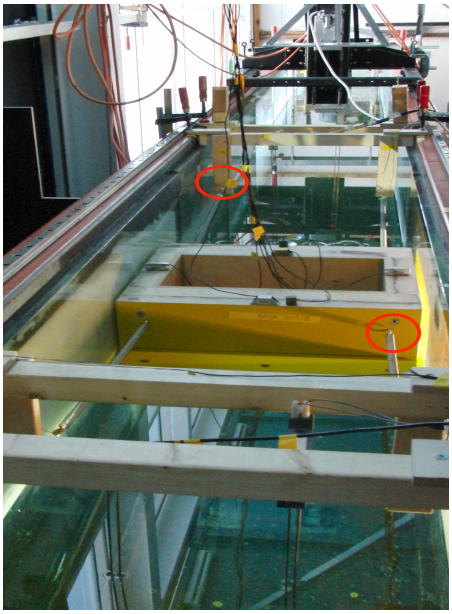


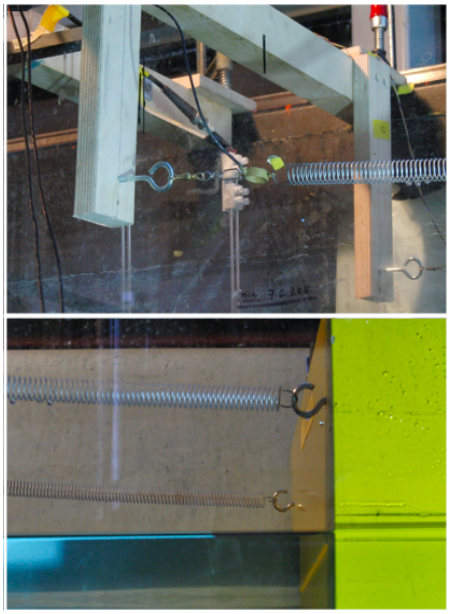
Figure 3.2: Weight distribution of the ballast weights in the model. The weights are placed symmetrically around both the x-axis and the y-axis.

3.1.2 Model Setup

In order for the model to be freely-floating in waves it has to be attached to springs on both ends to restrain it in the middle of the wave tank. The setup used in these experiments was developed during the pre-project in October 2017, where the frames and spring attachments were designed and optimised [Bremer, 2017]. There are two springs on each end of the model which were attached to frames suspended over the wave tank. Figure 3.3 below shows the model and frames as well as two detailed pictures of spring attachments on a frame and the model.



(a) An overview of the model with the frames where the springs are attached. The spring attachments on the frame and model are circled in red.



(b) Upper Picture: Detailed picture of the spring attachment on one of the two frames. Lower Picture: Detailed picture of the spring attachment on the model

Figure 3.3: Figure 3.3a shows the model setup with the frames and spring attachments. Figure 3.3b is a close-up of the spring attachments on the frame and the model.

Figure 3.3a shows the model setup with the springs mounted to the model and frames. The red circles show where the springs are attached to the frames and model, and Figure 3.3b shows a more detailed picture of these attachments. During the setup testing in October 2017 it was found that the springs with a spring stiffness of 23 N/m gave the system the most suitable natural frequency.

3.1.3 Test Facilities

The experiments were conducted in Ladertanken at the Norwegian University of Science and Technology, a wave tank operated by SINTEF Ocean. Ladertanken is a wave flume which has a length of 13.0 m and a width of 0.60 m. The wave maker is a flap type wave maker installed at the one end of the tank, while the other end has a beach to reduce wave reflection, illustrated in Figure 3.4.

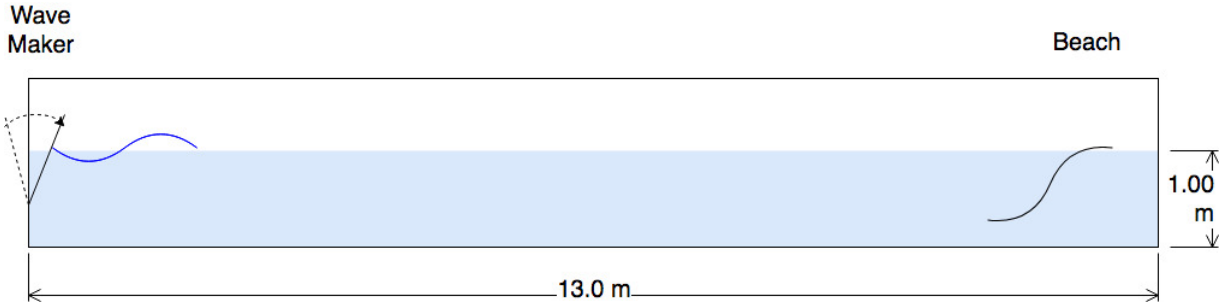


Figure 3.4: Side view of Ladertanken with the dimension, wave maker and beach.

3.1.4 Instrumentation

Wave Probes

Six wave probes were mounted in the wave flume in order to measure the wave elevation. Two probes were placed just after the wave maker, two in front of the model and two behind the model. The position of the wave probes is illustrated in Figure 3.5.

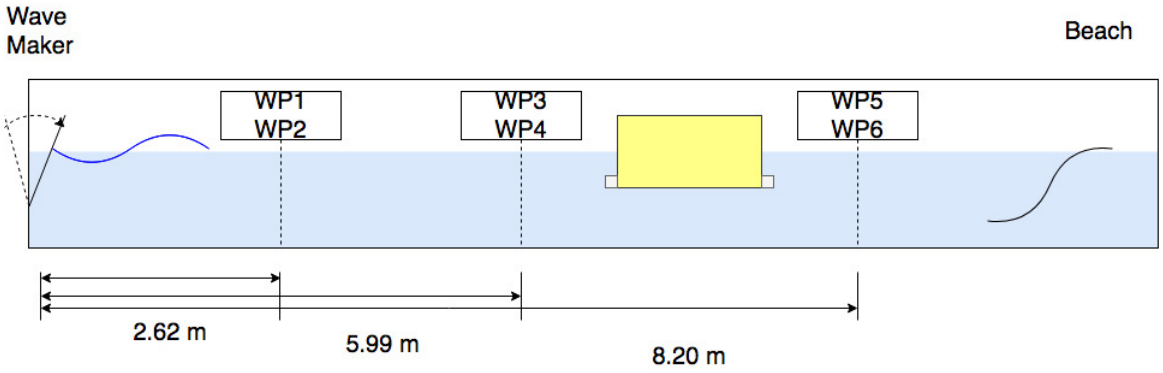


Figure 3.5: Illustration of the wave tank with the model and wave probes. The distance from the wave maker to the wave probe pairs is indicated on the figure.

Accelerometers

The model had six accelerometers mounted to measure the acceleration. Three accelerometers measured the motion in z-direction, two in y-direction and one in x-direction. The accelerometer measuring accelerations in x-direction was used to know if the model hit the tank wall. The accelerometer arrangement is illustrated in Figure 3.6.

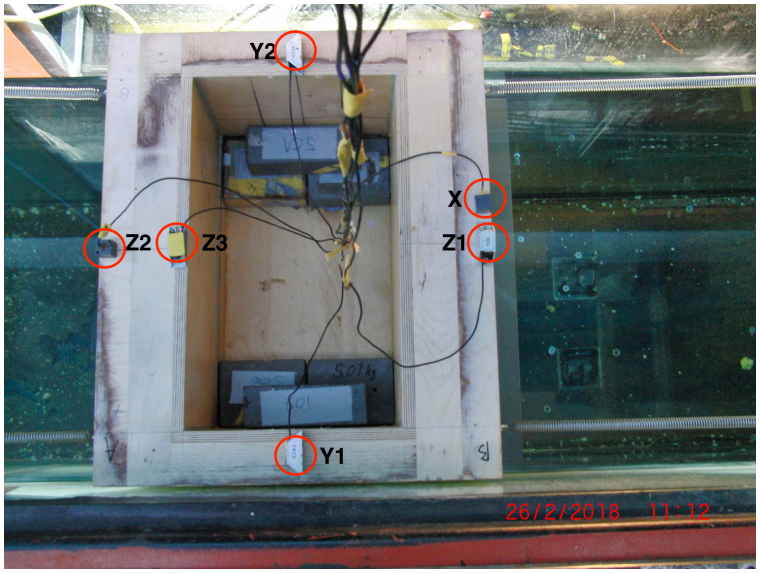


Figure 3.6: Accelerometers mounted on the model. The picture was taken from the opposite side of the tank, due to a better angle. The waves are coming from the right in the picture, propagating towards the left.

The accelerometers measuring the motion in z-direction were placed as far out from the model centre line as possible, to increase the measurement accuracy of the roll motion. The third z-direction accelerometer was used as a redundancy to ensure accuracy of the heave acceleration.

Force Gauges

There were four force gauges in the system, one on each spring. These measured the restoring forces in the springs and can be seen in Figure 3.7.

Figure 3.7: Close-up of the force gauges utilised to measure the force in the springs.

Sampling Frequency

The sampling frequency of the instrumentation determines how often the measurements are made. If the sampling frequency is too low, the measurements will not capture enough of the model motion. This will give a wrong impression of the model motion, as illustrated in Figure 3.8. However, having a too high sampling frequency will lead to unnecessary large data files. The sampling frequency was set to 200 Hz to ensure the vessel motion was accurately captured.

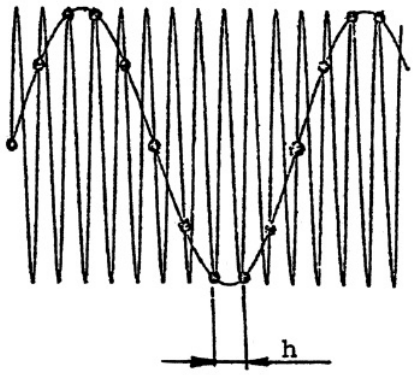


Figure 3.8: The consequence of too low sampling frequency [Steen, 2014]. As seen in the figure, a low sampling frequency is not capable of capturing the full shape of the signal.

Calibration

Calibration of measurement instruments means finding the right factor between the output signal and the physical property. The instruments give a change in voltage, which needs to be connected to the change of physical value. Thus, for an accelerometer the change of measured voltage will be converted to a change of acceleration and how much it changes needs to be checked by calibration.

Calibration utilises the relation in Equation 3.1 to find the ratio between voltage and physical property. By making three or four measurement with a known change in x, the results are plotted. The factor a is found as the computed calibration factor.

$$y = ax + b \tag{3.1}$$

Changing the physical value x on a wave probe means changing the submergence in the water. The accelerometers were rotated 90°and the force gauges were applied a known force. Before every experimental run, a zero measurement of all instrumentation equipment is made. This means finding the b value of Equation 3.1, making sure the measurement are made relating to the changed zero value.

3.1.5 Instrumentation Acceleration to Motion Acceleration

The acceleration in roll is an angular acceleration and is found using the measurements in z-direction made by two of the accelerometers. Accelerometer Z1 and Z2 were placed with a maximal distance between them, as illustrated in Figure 3.9. The acceleration in roll is the difference between the acceleration measured in z-direction, divided by the distance between them. Because the two accelerometers were placed on the opposite side of the centre of the model, one was measuring downwards as the other measures upwards.

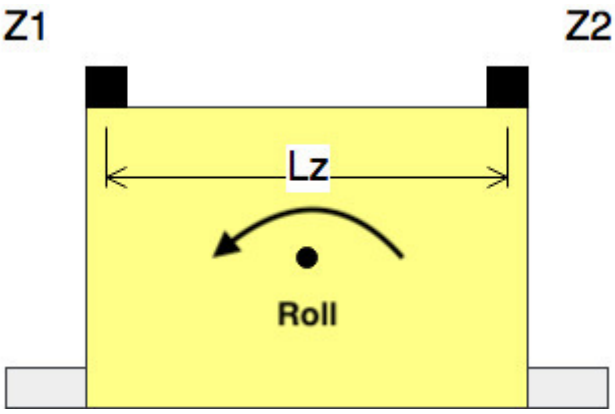


Figure 3.9: The distance l_z is defined as the distance between the two accelerometers, a_{z1} and a_{z2}. The acceleration in roll was found by calculating the difference between the two accelerometers and dividing by the distance l_z.

The equation for the roll acceleration then becomes

$$\ddot{\eta_4} = \frac{a_{Z1} - a_{Z2}}{l_Z} \tag{3.2}$$

In y-direction the accelerometers are already measuring pure sway motion, although a contribution from the roll angle was not included. The accelerometers were calibrated when the model is still, meaning there is no roll angle. However, when the model has a roll angle, there is a contribution from the z-accelerometers which shall be included in the sway motion. Figure 3.10 shows the contribution in the sway acceleration from a roll angle, the component g_y. Due to the two accelerometers in y-direction both measuring pure sway motion, the mean of the two values was used to reduce the error.

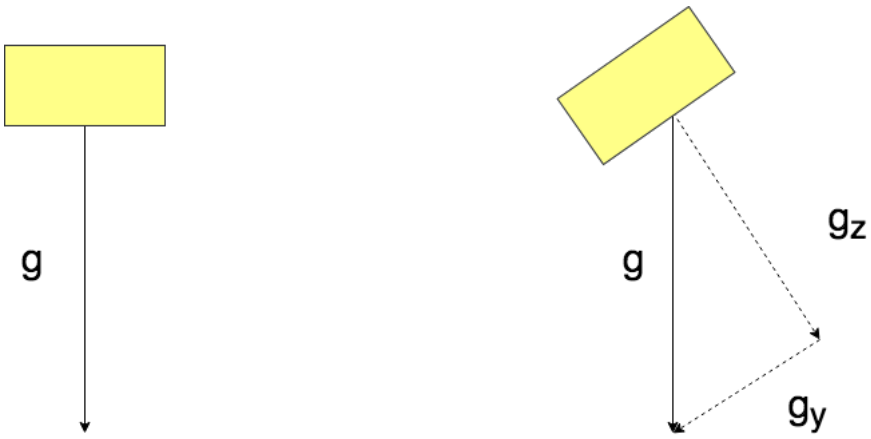


Figure 3.10: The figure illustrates the gravity component when the model has no rolling angle versus a rolling angle. With a rolling angle the gravity has a component in y-direction as well as the z-direction. This component has to be corrected for in the calculation of the motion in y-direction, the sway motion.

By assuming a small roll angle the expression for the sway acceleration was found from the mean of the y-direction measurements and the roll acceleration correction as follows:

$$\ddot{\eta}_2 = \frac{a_{Y2} + a_{Y1}}{2} - \eta_4 g \tag{3.3}$$

The acceleration in heave was found from the z-direction accelerations, however, to extract pure heave motion the mean of the z-acceleration is calculated from the following expression:

$$\ddot{\eta}_3 = \frac{a_{Z1} + a_{Z2}}{2} \tag{3.4}$$

3.1.6 Wave Generation

The wave maker in Ladertanken is automated and can run from an input file with sine wave oscillations. Because of this, it is possible to create a series of waves with a pre-set waiting time between each wave created by the wave maker. This means the wave maker can run for hours, and the limiting factor is the memory storage of the measurement data.

The waves were generated using the Matlab scripts *testinfoRegular.m* and *generateRegular.m* both developed by Prof. Trygve Kristiansen and manipulated to suit this particular experimental setup. The test matrix was given to the wave generation program in *testinfoRegular.m* while *generateRegular.m* creates the sine wave flap oscillations. The script creates sine wave oscillations according to the wave flap theory derived in [Hughes, 1993]. A summary of the wave period, wave steepness and flap amplitude of each wave created in a time series was written into a file which is utilised in the data processing. This was found necessary because the waves tested in the experiments require large flap amplitudes and some waves are not created because the flap amplitude exceeds the limitation of the wave maker. In order to keep track of which waves were created, it is therefore important to write the created wave to a file which can be utilised in the data analysis process.

A mechanical transfer function was developed from the ratio between the measured wave flap amplitude and the desired flap amplitude. By running waves in the wave tank without the model, the flap amplitude was measured and the ratio between the measured flap amplitude and theoretical flap amplitude can be found. Using this ratio, a transfer function was created which is capable of interpolating between the tested wave periods and wave steepnesses. This was necessary to ensure the measured wave flap amplitude is similar to the theoretical flap amplitude given to the wave generator. Due to large differences between the measured and theoretical flap amplitude, this process was found time consuming and left no time to create transfer functions for the wave amplitude and wave steepness.

3.1.7 Decay Test

To find the natural period in sway, heave and roll the model was excited and the motion of the model was measured. The natural period in sway and roll is found by pushing the model in the given direction. It was found hard to only excite the model in heave, thus the roll decay measurements were used and the heave accelerations were extracted using Equation 3.5.

$$\ddot{\eta}_3 = \frac{a_{Z1} + a_{Z2}}{2} \tag{3.5}$$

The decay tests were conducted on the full experimental setup, which means that the natural periods found from the decay tests are coupled natural periods.

3.1.8 Test Matrix

In order to have enough data to train a Machine Learning Model it is important to test as many different waves as possible. The two main parameters which were manipulated are the wave period and wave steepness. The larger model motion the greater non-linear effects around the bilge keels, thus the focus was on creating the largest wave combinations possible. Due to the restriction on the wave maker flap amplitude, the largest wave period was found to be 1.78 s and $\frac{1}{20}$ steepness. In order to run as many different waves as possible the following ranges were set for wave period and wave steepness.

Table 3.2: Experimental range of wave period and wave steepness

	Minimum	**Maximum**	**Step Size**
Period	0.88 s	1.78 s	0.02 s
Steepness	1/60	1/20	1/2.5

The test matrix which was given to the wave maker in *testinfoRegular.m* contains information about the waves to be created. Main parameters such as minimum, maximum and step size of wave period and wave steepness need to be decided.

Other parameters such as wave ramp up-and down, pause between each wave and how many wave oscillations in each wave also need to be declared in the test matrix.

The main test matrix is seen in Table 3.3 below where these ranges are implemented to five main runs. Because of limitations in the flap amplitude of the wave maker, not all these waves were produced. If a given wave steepness and wave period combination leads to a flap amplitude larger than 0.20 m the wave is not created. Hence, the produced wave file was created to know the parameters of the created waves.

Table 3.3: Experimental Test Matrix

Test	**Tmin**	**Tmax**	**ΔT**	$\frac{H}{\lambda}min$	$\frac{H}{\lambda}max$	$\Delta\frac{H}{\lambda}$	nT	**Pause**
1001	0.88 s	1.00 s	0.02 s	1/60	1/20	2.5	20	240 s
1002	1.00 s	1.12 s	0.02 s	1/60	1/20	2.5	20	240 s
1003	1.12 s	1.22 s	0.02 s	1/60	1/20	2.5	20	240 s
1004	1.22 s	1.34 s	0.02 s	1/60	1/20	2.5	20	240 s
1005	1.34 s	1.44 s	0.02 s	1/60	1/20	2.5	20	240 s
1006	1.44 s	1.56 s	0.02 s	1/60	1/20	2.5	20	240 s
1007	1.56 s	1.66 s	0.02 s	1/60	1/20	2.5	20	240 s
1008	1.66 s	1.78 s	0.02 s	1/60	1/20	2.5	20	240 s

3.1.9 Time Series

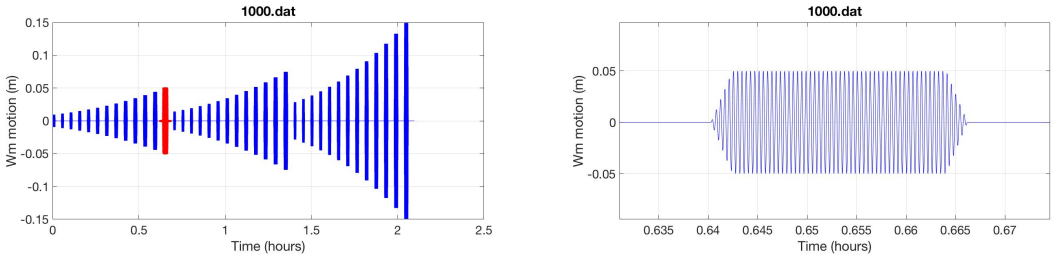
As seen in the Test Matrix in Table 3.3 the wave testing range has been split into eight series to limit the length of each run. It was found with the amount of instrumentation and the sampling frequency used in the experiments, the memory storage could handle series up to thirteen hours. Therefore, the series were split into runs of around eight hours to ensure enough memory.

The number of different waves and the duration of each series is summarised in Table 3.4. It is seen that the amount of waves decreases as the wave period increases. This is due to the wave flap amplitude restriction, when the wave period is longer and the waves are steep the required flap amplitude is larger than the limit. Therefore, not all waves in the test matrix were created.

Table 3.4: Number of waves and duration of run for each wave series

Time Series	**Number of Waves**	**Duration**
1	102	7.5 hours
2	102	7.6 hours
3	102	7.7 hours
4	102	7.8 hours
5	97	7.4 hours
6	84	6.5 hours
7	68	5.3 hours
8	49	3.8 hours
Total	706	53.6 hours

In Figure 3.11a a full test series of an example run is shown. In Figure 3.11b one wave is illustrated which is the wave oscillations marked in red in Figure 3.11a. Identifying these individual waves is part of the analysis procedure which is explained in the next section.



(a) A full time series of a wave set created by the wave maker. The red oscillations is one single wave.

(b) The wave maker oscillations which creates waves with a given wave steepness and wave period.

Figure 3.11: Figure 3.11a shows the full time series of a set of waves. One wave in the set is marked as red and a close-up of the oscillations in the specific wave are seen in Figure 3.11b.

3.2 Analysis Procedure

3.2.1 Time Series

The time series were saved in .bin files containing all instrumentation measurements and the time series time step. These files were opened in Matlab where the individual waves were identified and the instrumentation variables were sorted. Because the measurement system is measuring continuously during the whole time series, the time window where each wave is located has to be identified.

Time Window

In order to select the correct time window for each instrument the distance between the wave maker and the specific instrument needs to be known, to know the time it takes the wave to travel from the wave maker to the instrument.

During the process of post-processing the results, it was found that the wave maker has a small delay each time it applied a new wave steepness. This had to be accounted for when locating the time window of each wave and the exact formula for this was found by trial and error. It was found that the time delay had the following

relation

$$delay = 0.1\, f \times (0.45\, N_S) \tag{3.6}$$

Where f is the frequency of measurement and N_S counts the changes of the steepness and thereby increases the factor when the steepness is changed.

3.2.2 Filtering

The raw data was filtered using a band-pass filter in Matlab. A band-pass filter allows the frequencies within a range to pass and takes out the frequencies outside of the range. This is done in Matlab with the script *bpass2.m* developed by Prof. Trygve Kristiansen, where the signal is transformed using Fast-Fourier Transformation. The high and low pass frequencies are set to $\frac{1}{T} \pm 0.2$ Hz. Because the band-pass filter is applied using Fast Fourier Transformation the oscillations need to be complete, which is why it is given the full wave oscillation signal with ramp up, ramp down and the pause.

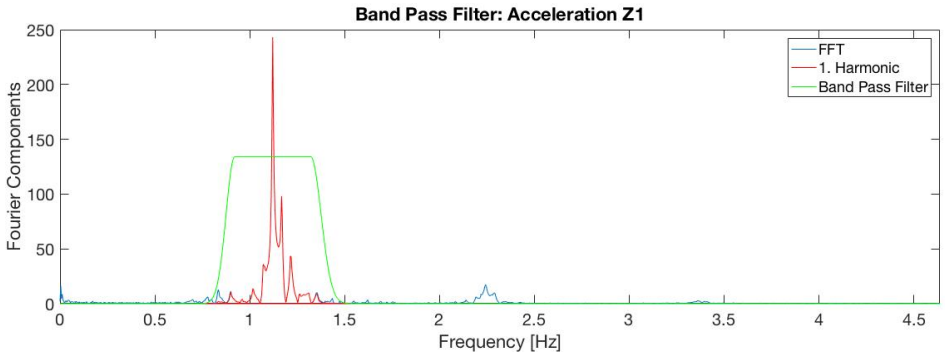
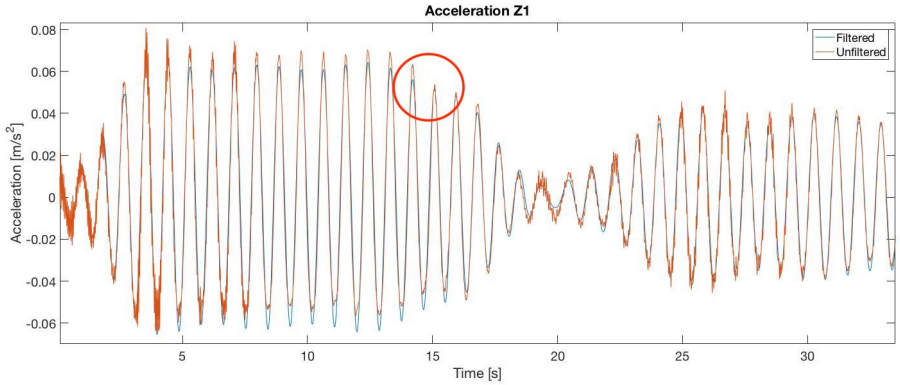


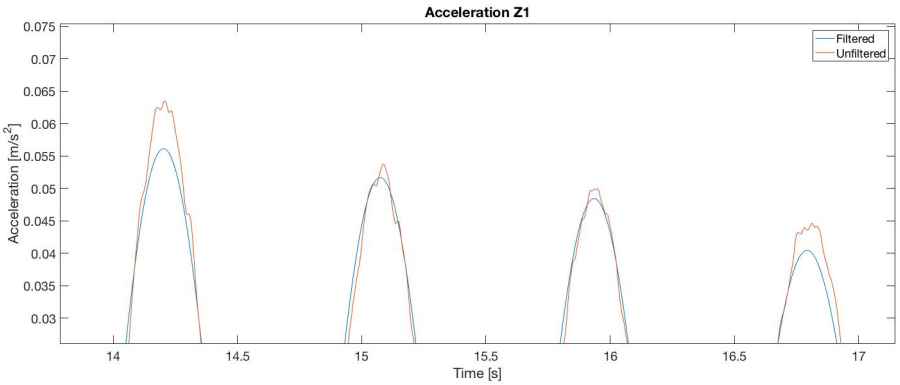
Figure 3.12: The band-pass filtering plots the frequency domain of the signal and shows the filtering range in green. The red signal is the filtered signal, while the blue signal is the unfiltered signal.

In Figure 3.12 above the red section of the signal was kept while the higher harmonics in red were set to zero. Thus, the frequency range within the green frequency range is the band-pass filter where the signal passes and is kept.

The filtered and unfiltered data of accelerometer Z1 is plotted in Figure 3.13.



(a) The filtered signal from accelerometer Z1 is plotted in blue, while the unfiltered signal is plotted in red.



(b) A close up of the red circle in Figure 3.13a above. The blue, filtered signal has less noise compared to the red, unfiltered signal.

Figure 3.13: Figure 3.13a shows the filtered signal in blue and the unfiltered signal in red. Figure 3.13b is a close-up of the circled area in Figure 3.13a.

All instrumentation measurements are filtered using the method described above. The filtered data is saved in a binary file and loaded in to Python where the data is analysed and processed.

Decay Tests

The measured accelerations in the decay tests were used to calculate the measured motions using the relations presented in Section 3.1.5. From the acceleration in the three motions, the position in each of them is found using the integration script written by Prof. Trygve Kristiansen, *acc2pos.m*. The integration is performed by transforming the signal using Fast-Fourier Transformation and integrating in the frequency domain before the signal is transformed back to the time domain. In the

sway direction the measurements of the force gauges were also used to calculate the position of the model in sway.

The force measurements and the spring stiffness of each spring are used to calculate the mean model position during the decay test using Equation 3.7 below.

$$\bar{y} = \frac{F1 + F2 + F3 + F4}{4\text{k}} \tag{3.7}$$

The position measurements were used as input to the frequency domain calculation in *bpass2.m*, where the peak frequency is the natural period in the given direction. The results from the decay tests are presented in Section 5.1.2.

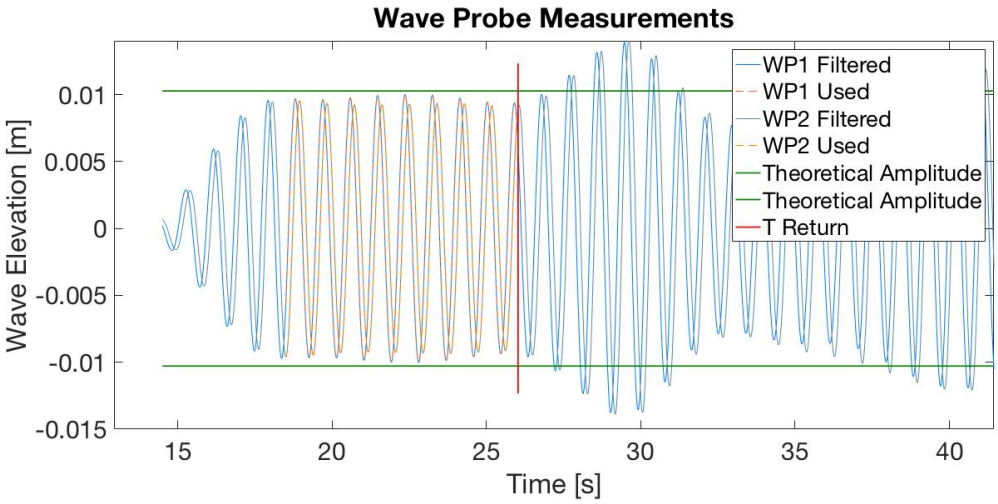
3.2.3 Wave Reflection

The wave tank has a length restriction and wave reflection is therefore a significant problem in the wave tank for certain wave periods. It is important to only use data measurements of the wave probes where wave reflection does not occur, because wave reflection adds or subtract to the original wave and thereby change the wave amplitude in the wave tank.

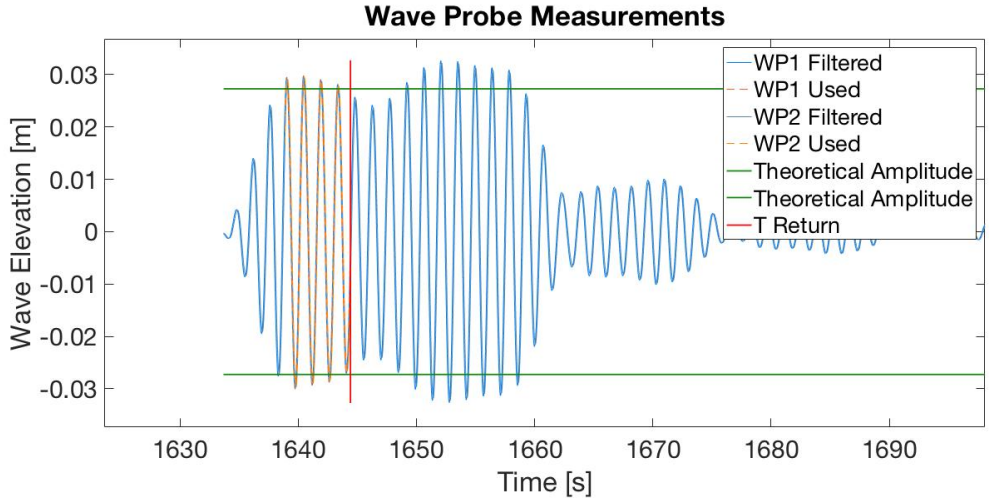
It was assumed that the beach in the end of the wave tank dissipates the waves adequately, which was based on visual observations during the experiments and inspection of the measurements. The wave reflection of significance is therefore the reflected wave by the model.

In order to cut the time window before the reflected waves disturb the wave oscillations, it is important to know the time it takes for the reflected waves to reflect back to the wave probes. This was calculated using the travelled distance and the group velocity.

During inspection of the wave elevation it was found that the ramp up waves reflected and significantly disturbed the wave oscillation for wave periods shorter than 10.0 s. For the longer wave periods the ramp up waves did not cause significant wave reflection, but for longer wave periods the main wave oscillations cause reflections which also occur earlier in the wave run. This was due to the group velocity increasing with increasing wave period. For the intermediate wave periods it was therefore found that there were very few steady-state wave oscillations to measure before wave reflection disturbed the wave oscillations. In Figure 3.14 the short period wave amplitude is plotted in Figure 3.14a and an intermediate wave amplitude is plotted in Figure 3.14b. The plots clearly show the difference in steady-state duration and the time when wave reflection starts is marked with a red line.



(a) Period: 0.88 s, Steepness: 1/60



(b) Period: 1.44 s, Steepness: 1/57.5

Figure 3.14: Figure 3.14a shows a short wave period where the return waves arrive after around eight complete wave oscillations. Figure 3.14b shows a medium wave period where the return waves arrive after around four wave oscillations and the wave measurements show a more unstable wave elevation compared to the shorter wave period. The red line is the point where the return wave has reached the wave probes.

For the purpose of the experiments in this investigation it is assumed that the results with a few stable oscillations are acceptable. Because the objective is to investigate the possibility to use Machine Learning the importance of having enough data outweighs the desire to use highly accurate data. Using data with less stable oscillations will lead to a lower accuracy of the results, because the hydrodynamic effects may not be fully developed.

For the longest wave periods tested during the experiments, there is very little wave reflection and more of the measured data can be used. This was discovered during the experiments and visual inspection of the data confirms this. Figure 3.15 contains the obtained data in yellow and the time of wave reflection as the red line.

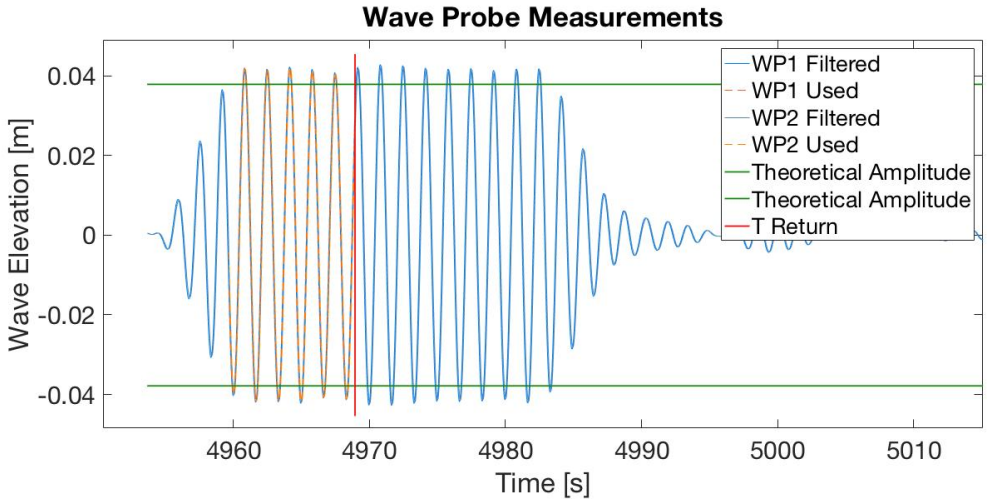


Figure 3.15: Wave Period 1.68 s - Wave Steepness 1/52.5. The red line is the point where the return wave has reached the wave probes. As seen in the plots, the wave elevation is stable also after the return wave has reached back to the wave probes.

3.2.4 Data Analysis

The wave amplitude was found using the two wave probes placed closest to the wave maker in Figure 3.5, hence WP1 and WP2. The mean value of the average of the wave height, measured by the two wave probes, was used as the experimental wave height. In Figure 3.16 below the calculation of the wave height is illustrated.

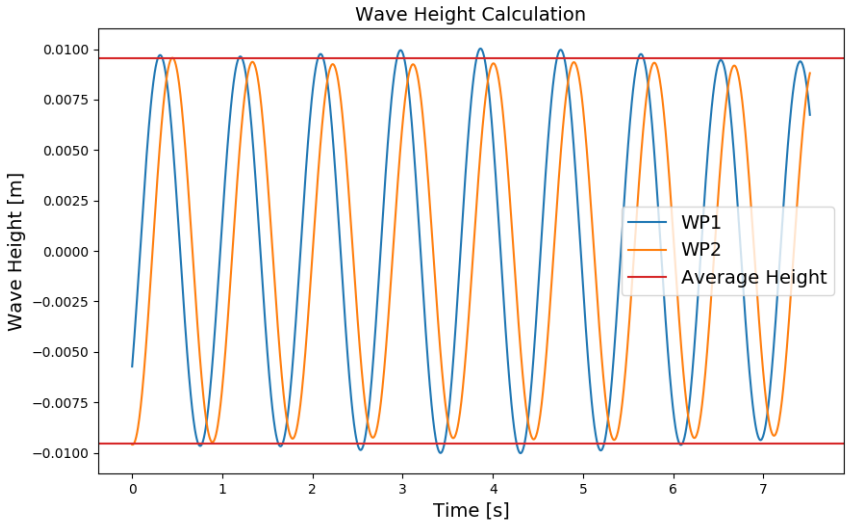


Figure 3.16: The wave height is calculated as the mean of the average wave height and marked in red in the figure. The two wave probe measurements are plotted in blue and orange.

A very small difference between the measured wave height is observed between two wave probes, which could be due to small errors during calibration. The difference is consistent and very small, so it is assumed that it is not due to transverse waves in the wave tank.

The initial wave probes, WP1 and WP2, were also used to find the experimental wave period. Using the time window measurement of the wave probes, the mean distance between the zero upwards and downwards crossings is the average wave period of the wave in the time window. This experimental wave period was further used to find the experimental wave length, wave number and wave steepness.

The instrumental accelerations were used to calculate the acceleration in sway, heave and roll by using the acceleration equations in Section 3.1.5. The accelerations in sway, heave and roll were further used to calculate the Response Amplitude Operator, RAO, in each of the three directions. The RAOs are defined as follows in Equation 3.8 below.

$$RAO_2 = \frac{\eta_{2a}}{\zeta_a}, \quad RAO_3 = \frac{\eta_{3a}}{\zeta_a}, \quad RAO_4 = \frac{\eta_{4a}}{k\zeta_a} \tag{3.8}$$

Where η_a is the motion amplitude, ζ_a is the wave amplitude and k is the wave number. The characteristic length used for the calculations is the width of the model.

3.3 Uncertainty Analysis

There are two main types of errors in experiments, *bias errors* and *precision errors* [Steen, 2014]. The bias errors are systematic errors which will not be revealed in the repetition of experimental cases. Due to the nature of this investigation where the results are used in Machine Learning, the bias errors will not affect the accuracy of the Machine Learning prediction. Since bias errors are systematic, they will be systematically in the data given to the machine learning model, which means that the learning algorithm will include the bias errors in the learning process. The bias errors will therefore not be attempted to quantify, but a short description of possible bias errors follows in Section 3.3.1.

3.3.1 Bias Errors

Friction between the glass wall and the model was closely monitored and it was made sure that the model never actually hit the wall. However, the thin layer of water between the model and glass wall could cause friction if the boundary layer developed to a turbulent layer.

Deflection of the frames which hold the springs could cause an error in the model motion. The springs are attached symmetrically on the model to minimise the error of deflection, but there is still a possibility that this will affect the motion of the model.

Spring Stiffness is assumed constant during the whole experimental process which lasted for around three weeks. However, the spring stiffness can decrease during the experiments due to being under tension which can cause fatigue.

Gaps between the bilge keel and model in the bilge keel attachment could cause disturbances in the fluid flow around the bilge keel.

Temperature fluctuations in the laboratory was measured to ± 1.5°which could lead to inaccurate instrumental measurements.

Water level in the wave tank was measured before each experimental run to ensure a constant water level. During the preliminary testing it was found that the water level decreased if the water pump system was open. The valve for the water filling system was therefore completely shut to maintain a constant water level during each of the time series.

The wave maker was not completely calibrated for the wave amplitude, which can lead to the waves created not being exactly as the theoretical wave. This was corrected for as the experimental wave data are used in the calculations of the experimental results. The wave maker width is a little less than the wave tank width, which leads to some back flow at the edge of the wave maker. The created wave is therefore not the full width of the wave tank. The tank walls are not fully straight, which can disturb the fluid flow and create non-linear effects. Friction between the fluid and the tank wall can lead to deflection of the moving waves, which can be measured by comparing the wave elevation at each pair of wave probes. If one of the wave probes in a pair measures differently, the wave is deflected or there are transverse waves in the tank. Neither the deflection of waves or vortex shedding was observed under visual inspection of the created waves, but these are still effects that cannot be neglected.

The beach at the end of the wave tank was placed so it would dissipate the waves created by the wave maker. However, due to the constant position of the beach, the wave dissipation was not always optimal for the given wave oscillation. Some wave radiation can therefore be expected, which can influence the motion of the model.

3.3.2 Precision Error

Precision errors are calculated by repeating the experiments and ideally all cases of the experiments should be repeated. This is seldom done due to time constraints and it is therefore common to run repetition tests on one or a couple of cases, depending on the experiment, and assume this is valid for all cases. The method of calculation is presented in Appendix A and is based on [Steen, 2014].

The precision limit and uncertainty of a single test is the error of interest, but in order to calculate this for a single test a number of repeated tests are necessary.

For these experiments the time series number eight in Table 3.4 was repeated four times, corresponding to test number 1008 in Table 3.3. The precision limit and uncertainty for both the mean of the repeated tests and a single test are calculated for all three directions and response amplitude operators using the Matlab and Python scripts. The results are presented in Table 3.5 for period 15.0 s and two steepnesses, 1/60 and 1/50.

Table 3.5: Precision limit and uncertainty of the experiments

	Steepness: 1/60 **Mean Uncertainty**	**Steepness:** 1/60 **Single Test Uncertainty**	**Steepness:** 1/50 **Mean Uncertainty**	**Steepness:** 1/50 **Single Test Uncertainty**
Sway	1.3 %	2.6 %	1.1 %	2.1 %
Heave	0.2 %	0.5 %	0.5 %	0.9 %
Roll	0.7 %	1.3 %	0.9 %	1.8 %
RAO 2	1.5 %	3.0 %	0.9 %	1.8 %
RAO 3	0.6 %	1.2 %	0.5 %	1.1 %
RAO 4	0.6 %	1.2 %	0.5 %	1.1 %

In Table 3.5 above it is seen that the uncertainty for a single test is 3.0 % at the

highest and it is therefore assumed that the experimental results are all within the acceptable uncertainty of 5.0 %.

Chapter 4

Machine Learning Method

In the following chapter the methods of linear regression and neural networks will be presented. Linear regression is done to show the difference between a linear prediction and a neural network non-linear prediction. It is expected that the reader is familiar with Python and only key concepts will be explained.

4.1 Linear Regression: Response Prediction

The linear regression model is developed to predict the motion and response amplitude operators of the model. The Python library *Scikit-Learn* is used to develop the regression model. *Scikit-Learn* is an open-source Machine Learning library in Python with emphasis on being user friendly, having high performance and proper documentation [Pedregosa et al., 2011].

The linear regression model is able to predict a target using features, similarly to a Neural Network. The features in the linear model are the wave period and the wave height, while the target is sway, heave and roll and the corresponding RAOs in individual linear regression models.

The data is split into a training data set and a testing data set, where 20 % of the data is used for testing. The linear regression is done on the training data set and the testing data is used to find the accuracy of the model prediction.

The data is split using the built in function in *scikit-learn*, `sklearn.model_selection.train_test_split(X, y, test_split=0.2)`. The features, target and the chosen value to split the data has to be given to the function and the function also shuffles the data points.

After the data is split into test and training data sets, the model is created and fitted using the two following functions.

```
model = SGDRegressor(loss='squared_loss')

model.fit(X_train, y_train)
```

The model utilises the mean squared error as the loss function and stochastic gradient descent as the optimisation algorithm. With the created model the predictions are made with the following function, `model.prediction(X_test)`. The performance of the linear regression prediction is calculated using the coefficient of determination, R^2, as seen in Section 2.2.6. The coefficient of determination is used as evaluation metric due to the range from 0 to 1. *Scikit-Learn* has a built in function which calculates the R^2-value of the prediction, `model.score(X_test, y_test)`.

The goal of creating these six small linear regressions is to show the difference between linear curve fitting and the non-linear curve fitting which Neural Networks is capable of doing.

4.2 Neural Network: Response Prediction

The Neural Network is developed in *Keras*, which is an open-source Neural Network library for Python developed by Google Engineer Francois Chollet [Gulli and Pal, 2017]. The library runs on top of *TensorFlow* which is a Google Developed Machine Learning Library. TensorFlow is a mathematical library which is used for Machine Learning, however due to the complexity of TensorFlow, Keras was developed by the Google engineer to create a user-friendly tool enabling fast experimentation with

Neural Networks. Keras is now supported by both Google and Microsoft and is a rapidly growing library for Machine Learning.

Several Python libraries are utilised to build the Neural Network, such as *Keras*, *Scikit-Learn*, *Numpy* and *Pandas*. In the following sections the key concepts and code implementations will be explained.

4.2.1 Targets and Features

There are six targets of interest in total; the motion and response amplitude operator in sway, heave and roll. A model can only predict one target at a time, so six different models were trained, one for each of the targets. Two features were used in the neural network, wave period and wave height. These are the two inputs which each give new information about the conditions. Parameters like wave length and wave steepness are calculated from using either period or height, thus they are dependent on the two features. Hence, only wave period and wave height can be given to the model as features in order to avoid redundant features which reduce the accuracy of the prediction.

Feature Scaling

As mentioned in Section 2.2.6, feature scaling is different for each problem. In this investigation both standardisation and normalisation was attempted in order to find which was best suited for the problem at hand. *Scikit-Learn* was used to pre-process the data for Machine Learning.

Standardisation was conducted using the `StandardScaler()` which scales each feature independently. This means the features can be implemented at the same time to the function and it will transform the feature to have a mean value of 0 and a standard deviation of 1. The code is implemented as follows:

```
scaler = preprocessing.StandardScaler().fit(feature)

X = scaler.transform(feature)
```

The normalisation was performed using the same pre-processing library in *Scikit-Learn*. The function `MinMax()` scales the features individually to have a range between 0 and 1. Neither the wave period or the wave heights are negative, thus the range for these two features will be between $0 - 1$. The normalisation code lines are as follows:

```
scaler = preprocessing.MinMax().fit(feature)

X = scaler.transform(feature)
```

After trial and error using standardisation and normalisation it was found that the two methods performed with a similar accuracy. Standardisation is used in the final Neural Networks.

4.2.2 Split Data

The data was split into testing and training data using a function in the *Scikit-Learn* Machine Learning library. The function `train_test_split()` splits the data into a training set and a testing set. The user sets the proportion of the data to be included in the test split [Pedregosa et al., 2011]. The function shuffles the data before the splitting, due to the experimental data being arranged with increasing wave period.

The application of the function looks like this:

```
model_selection.train_test_split(X, y, test_size=0.2
```

The validation data was set to 20 % of the total data points, leaving 80 % of the data for training and testing. As seen in the command above, the splitting of the validation data is called `test_size` which can be confusing as it is actually splitting

out the validation data set. The explanation of the roles of the training, testing and validation data sets is found in Section 2.2.6.

4.2.3 Building the Model

The simplest model in Keras is the `Sequential()` model, which is a linear stack of layers [Gulli and Pal, 2017]. A Neural Network model was built by adding layers with their assigned activation function and number of neurons. The model consists of three layers, as well as the input layer, where all layers are fully connected. The first layer has 32 neurons, the second has 24 neurons and the output layer has 1 neuron. The first hidden layer utilises the *ReLU* activation function, while the second hidden layer has the *tanh* activation function.

```
model = Sequential()

model.add(Dense(32, input_dim=2, activation='relu'))

model.add(Dense(24, activation='tanh'))

model.add(Dense(1))
```

After the layers and nodes were created, the model had to be given a loss function, optimisation method and a choice of metric to evaluate the models predictions. This is done in the following line of code:

```
model.compile(loss='mse', optimizer='rmsprop', metrics=['mape', coeff_determination])
```

The loss function was chosen as Mean Squared Error, the optimisation was done using the *RMSprop* and the evaluation metric was the Mean Absolute Percentage Error and Coefficient of Determination. Due to the Mean Squared Error being used as the loss function, the two other measures of accuracy mentioned in Section 2.2.6 are used in order to have more than one measure of accuracy.

The model was trained on the training data set using the line of code below:

```
model.fit(X_train, y_train, epochs=1200, batch_size=10, validation_split=0.2, verbose=1)
```

The `epochs` is the number of times the model goes through the whole data set and updates the weights. Because one epoch is large, the data set is split in to batches and the batch size is chosen as 10. During training the third split of the data set happens, where 20 % of the training data is set aside to validate the predictions of the model during the training, as explained in Section 2.2.6. The last input is the `verbose=1` which means that the calculated loss and accuracy of each epoch is written to the terminal in order to observe the development of the training. This was useful to ensure the model is learning during the training process. The development of the coefficient of determination was also plotted after the training to monitor the learning process.

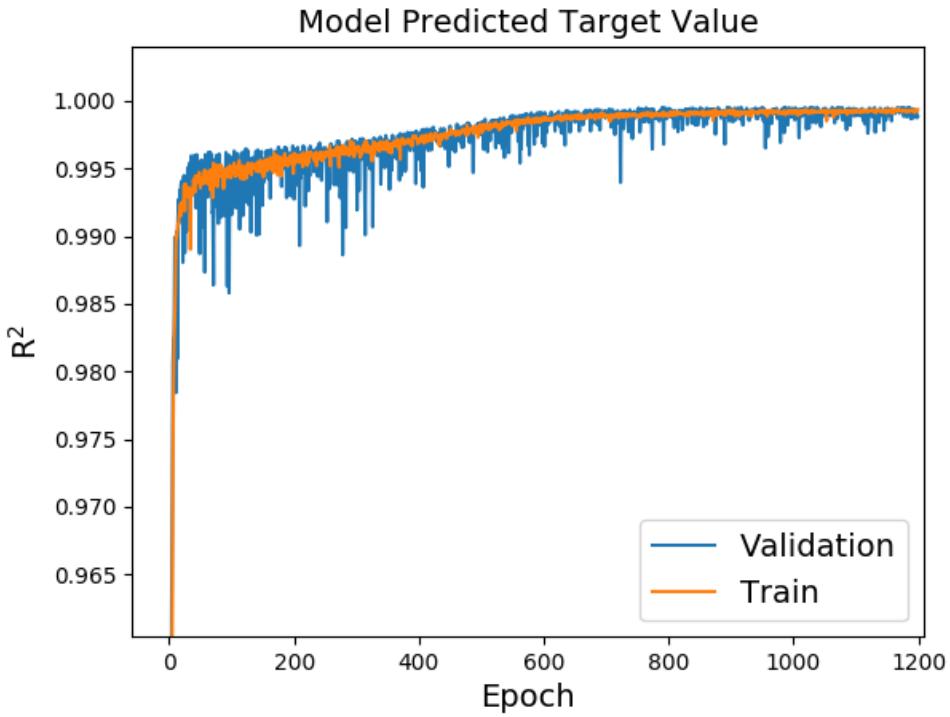


Figure 4.1: The plot shows the development of the Coefficient of Determination during a prediction of the RAO in roll. The plot shows the development of R^2 of both validation and training data. The training error is converging towards a stable value.

Figure 4.1 shows the development of the coefficient of determination for each epoch in the prediction of the RAO in roll. The number of epochs has to be large enough so the coefficient converges, which Figure 4.1 shows. The blue line labelled *Validation* is the value of the validation *during* training, which is done using the testing data set, as explained in Section 2.2.6. The orange line is the R^2 value of the training data. Figure 4.1 is also used to observe if the model is overfitted, which is when the training data maintains a high R^2 at the same time as the validation data has a decreasing R^2. The high values of R^2 are discussed in Section 5.3.2.

The final testing of the model is when the model predicts an output given only input data. Two functions were used, one function to evaluate the predictions to obtain the accuracy metrics and another to obtain the predicted values.

```
model.evaluate(X_test, y_test, batch_size=10)
```

```
y_pred = model.predict(X_test)
```

`model.evaluate()` is the function which gives the metrics of the model, meaning the Coefficient of Determination and the Mean Absolute Percentage Error. `model.predict()` uses the trained model to predict the target value given only features. These features given to the prediction are unseen data points, thus the *validation* data set.

As mentioned before, the notation in theory and practice of the three data sets can be confusing and the notations in *Keras* is defined different to the definitions in Section 2.2.6. The important note is that there are three data sets, one for training, one to validate during training and one to validate the final model.

4.2.4 Finding the Best Neural Network Configuration

The main challenge with building a Neural Network is finding the combination of parameters which gives the best prediction for the problem at hand. With the

number of layers, number of nodes in each layer, activation function in each layers, scaling method of input data, choice of loss function and optimisation, number of epochs and batch size, there are a large number of parameters which all can be combined to a large amount of Neural Network configurations. In order to find the best fit model trial and error with different configurations were tested and the Coefficient of Determination and Mean Absolute Percentage Error compared.

The challenge with Machine Learning and Neural Networks is that there are no clear guidelines to when a certain parameter choice works best. This is a part of the unknown characteristics of Neural Networks, it is not always known *why* something works well. A recurring issue is therefore that trial and error of different methods has to be used to find the best model for the given problem. During the trial and error process, the settings of the model were changed so it always chooses the same data sets for training and testing. This is done to eliminate the uncertainty of new data sets, and to ensure the change of accuracy is due to the parameter changes.

The process of building the Neural Network started by creating a small Neural Network with only two nodes in the first layer and four nodes in the second layer, using $ReLU$ for both layers, with the Mean Squared Error as loss function and *RMSprop* for optimisation. The number of epochs was set to 1000 for an initial estimate, each batch was a size of 10 and the target was the sway motion of the model. Inspecting the results showed that the number of epochs could be too low, and the number of epochs was increased to 5000. From this the model size was increased, different activation functions, loss functions and optimisation functions were tested which are summarised in Appendix D.1. The final result may not be the optimal configuration for this problem, but due to trial and error being a time consuming process and the goal of this study being to show if the method works, it was not prioritised to further optimise the model.

For the final test of convergence of the prediction results, the data sets were randomly chosen in order to include the statistical variation in prediction results due to

picking different test data each time. Because Neural Networks is a statistical tool, there are variations in the accuracy of the prediction due to the different testing data each training session. The average accuracy of ten tests is used as the final accuracy of the trained model which are summarised in Table D.2 in Appendix D.

The trial and error of the model configurations was conducted on the sway motion model, and it is assumed that the same parameters can be used on the five other models. The features given to the models are identical and it is therefore assumed that the model configuration found from the sway motion will be of high performance also for the five remaining targets. The ten repeated tests for all six targets are found in Table D.2 in Appendix D.

Chapter 5

Results and Discussion

This section presents and discusses the experimental results including wave generation, decay tests and the measured response of the model. Results from the Linear Regression and Neural Network predictions will also be presented and discussed.

5.1 Experimental Results

A bug was discovered in the computer software which transforms the measurement signals from analogue to digital values in May. The experimental results are therefore inspected to see if these experiments are affected by the bug in the system.

5.1.1 Wave Generation

The wave generation is described in Section 3.1.6 and the performance of the wave maker is investigated by comparing the experimental wave height and wave steepness with the theoretical wave height and wave steepness. The percentage error of the generated wave height of waves with theoretical steepness of 1/60, 1/45 and 1/30 are presented in Table 5.1.

Table 5.1: Experimental wave generation results. A comparison between theoretical and experimental wave height and steepness.

Theoretical Steepness	**Average Experimental Steepness**	**Percentage Error: Wave Height**	**Percentage Error: Wave Steepness**
1/60	1/55.7	7.9	7.2
1/45	1/41.4	7.8	7.9
1/30	1/27.9	6.9	6.9

The generated waves are around seven percent different from the theoretical value for both wave height and wave steepness. The comparison plot below, Figure 5.1, shows the experimental wave amplitude together with the theoretical wave amplitude for three wave steepnesses.

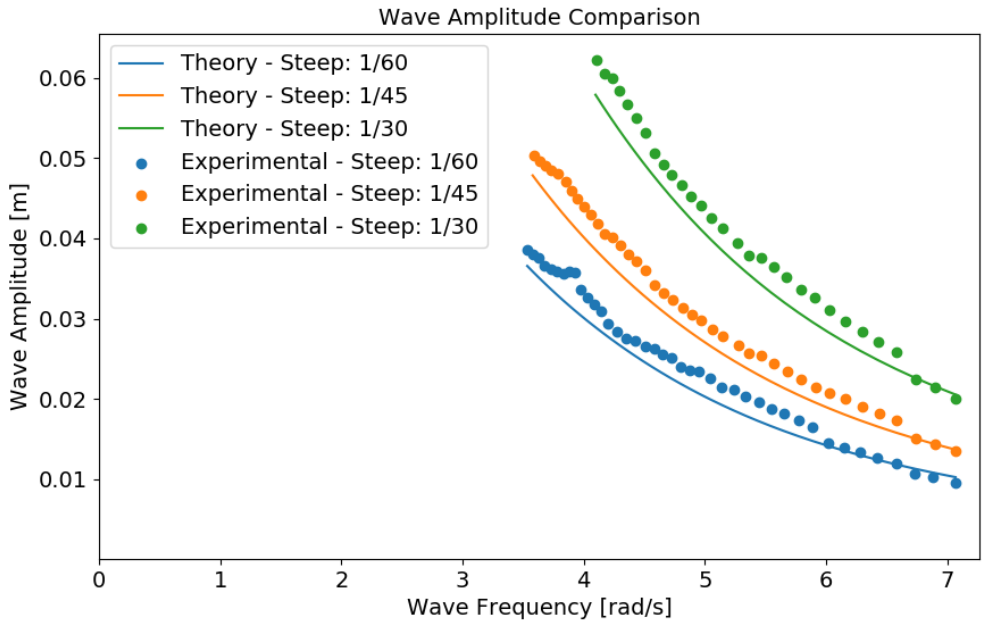


Figure 5.1: Comparison plot of theoretical and experimental wave amplitude. Wave steepness 1/60, 1/45 and 1/30 are plotted and show that the experimental waves are generally higher than the theoretical waves. The least steepest wave, plotted in blue, has inconsistent wave amplitude which varies more than the two other plotted wave steepnesses, however it is generally closer to the theoretical wave amplitude.

The experimental wave amplitude follows the same trend as the theoretical wave amplitude, although the experimental waves are generally higher. Interestingly, the

wave amplitude for the three highest frequencies of all three steepnesses plotted, are lower or equal to the theoretical value. At frequencies lower than approximately 6.6 rad/s the measured wave amplitude stabilises a little higher than the theoretical value. The least steepest wave with steepness 1/60, is seen to be less consistent than the two other steepnesses. In spite of the higher variation, the wave steepness 1/60 seems to lie generally closer to the theoretical wave amplitude.

The wave steepness comparison is plotted in Figure 5.2. The measured wave steepness tends to be steeper than the theoretical wave steepness.

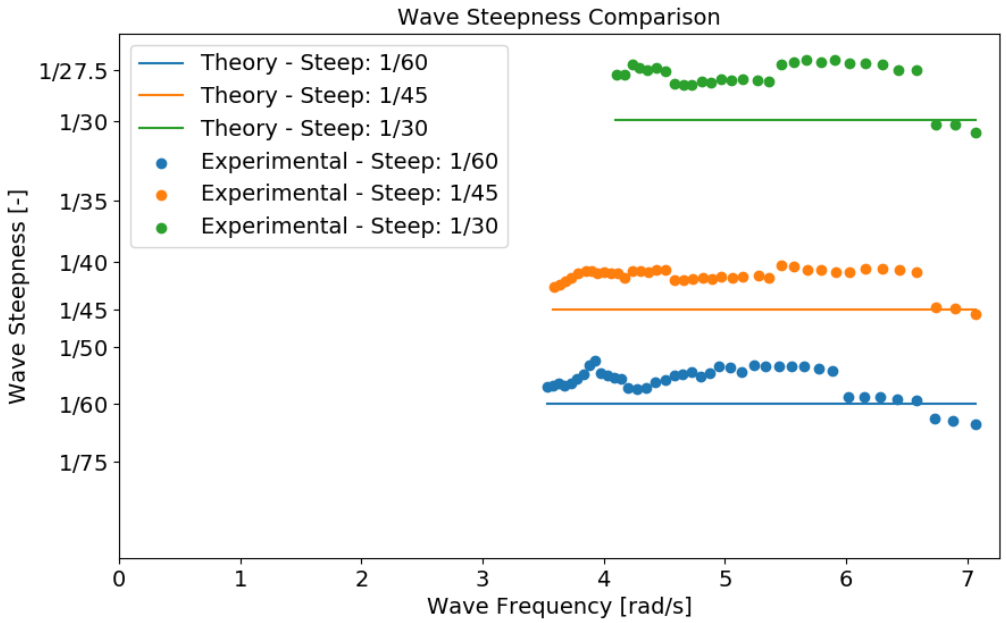


Figure 5.2: Comparison plot of the theoretical and experimental wave steepness. Theoretical wave steepness 1/60, 1/45 and 1/30 are plotted against the experimental result. All three wave steepnesses have a steeper wave than the intended theoretical wave steepness, but they are all fairly accurate at the highest wave frequencies. The two steepest waves, 1/30 and 1/45, are consistently steeper for all other frequencies. The least steepest wave is close to the theoretical steepness at both the highest and lowest tested frequencies, but shows a steeper wave at frequencies between $5.0 - 6.0$ rad/s and a higher variation of wave steepness in general.

Figure 5.2 above shows the comparison between the theoretical and experimental wave steepness for three waves. The three waves coincide well with the theoretical value at the highest tested wave frequencies which are around 7.0 rad/s. The two

steepest waves, 1/30 and 1/45, stabilise to a steeper wave compared to theoretical steepness at wave frequencies lower than around 6.8 rad/s. The least steepest wave, 1/60 is close to the theoretical steepness in the frequency range $6.0 - 7.0$ rad/s. At the mid-range wave frequencies all three waves are steeper than the theoretical waves, but the least steepest wave moves closer to the theoretical steepness at the lowest tested frequencies around $3.5 - 4.5$ rad/s. The two steepest waves, 1/60 and 1/45, are more consistent while the least steepest wave 1/30 shows more variation in the wave steepness. The experimental results are after this presented with the average measured wave steepness instead of the theoretical wave steepness.

In both wave amplitude and wave steepness the wave maker produces different values than the theoretical values. The percentage error is summarised in Table 5.1. The differences are due to imperfections in regards to both wave maker and wave tank, as discussed in Section 3.3.1. These differences could have been minimised by creating two transfer functions, one with respect to the wave amplitude and one for the wave steepness. A transfer function for the theoretical and experimental flap amplitude was developed and implemented in the wave generation, but as seen in the comparisons it is not enough to remove all differences between theoretical and experimental waves. The two additional transfer functions were not created due to time constraints, but would involve comparing the theoretical and measured wave amplitude and wave steepness and create a transfer functions which would be included in the wave generation script. Hence, the difference between the theoretical and experimental wave amplitude and wave steepness would be utilised to tune the wave maker. These transfer functions reduce the errors due to the imperfections of the wave maker and wave tank discussed in Section 3.3.1 and consequently lead to a more accurate experimental wave.

In Section 3.3.1 where the errors in the experiments are presented, it is mentioned that one can investigate wave deflection and transverse waves by comparing the measurements of the wave probe pairs. However, the experimental waves were measured during the experiment with the model in the water, thus the waves are

disturbed and can not be compared. Only the wave probes closest to the wave maker are not affected by the model, although as soon as the radiated waves from the model return to these wave probes, these are also disturbed by the model. It was therefore not possible to investigate the development of the wave along the wave tank with regards to wave deflection and transverse waves. Due to the nature of the experiment where the measurements are used for Machine Learning, it was decided that the priority was to obtain enough results to utilise in Machine Learning instead of increasing the extent of the hydrodynamic investigation in the wave tank.

5.1.2 Decay Tests

The decay tests are described throughout Section 3 in the Experimental Method, and mainly consists of filtering the signal and plotting the frequency domain of the model position during decay tests. From the decay tests the natural frequencies and periods for the model in sway, heave and roll can be quantified. The natural periods found from the decay tests are summarised in Table 5.2 below.

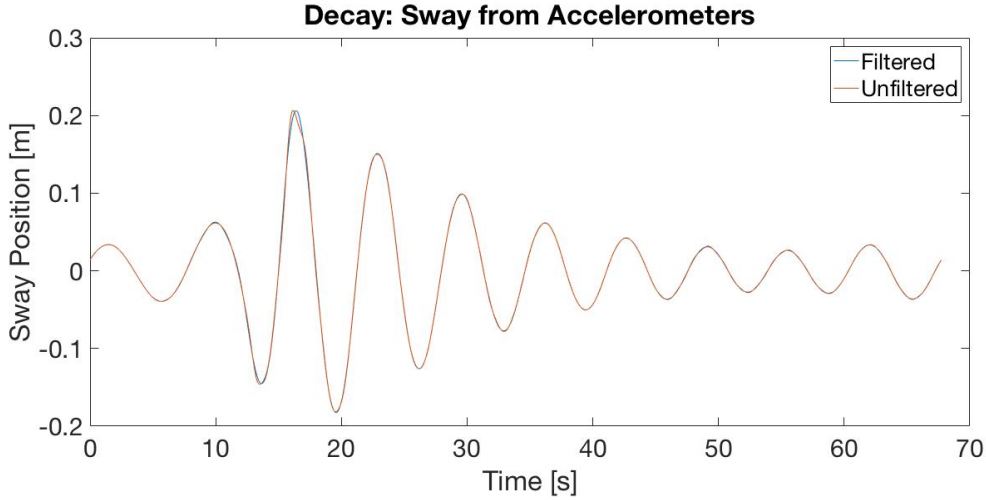
Table 5.2: The natural period and frequency of sway, heave and roll found from decay tests

Motion Direction	**Natural Period [s]**	**Natural Frequency [rad/s]**
Sway	6.77	0.93
Heave	1.43	4.40
Roll	1.33	4.71

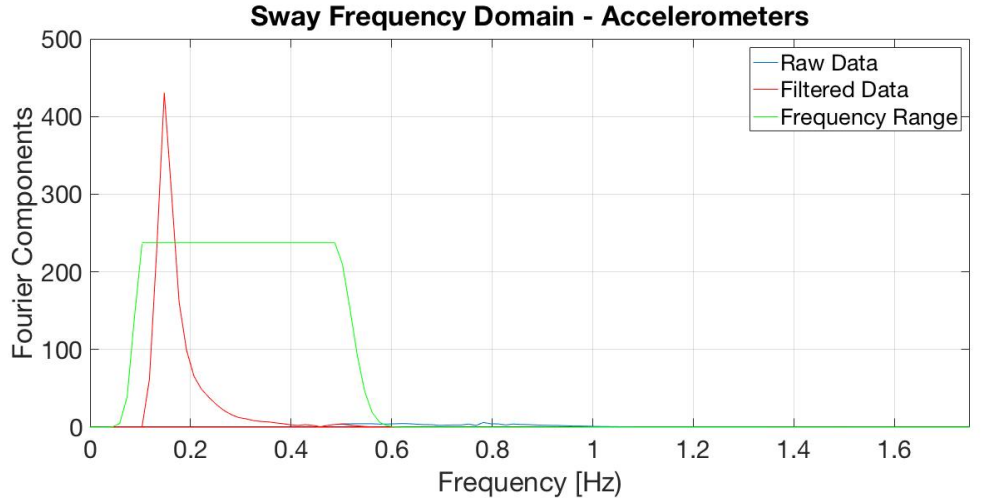
Table 5.2 shows that the measured heave and roll natural periods are very similar. These are hard to observe by the naked eye during the decay test, but the similarity indicates that one of the can be affected by the bug in the computer software for signal processing.

The decay test in sway excites the model in sway direction and the motion can be found from both the accelerometer measurements and the force gauges. Figure

5.3 shows the filtering and the frequency domain of the sway motion using the accelerometer measurements.



(a) Decay Sway - Accelerometers: The filtered and unfiltered position signal of the model obtained from the accelerometers.

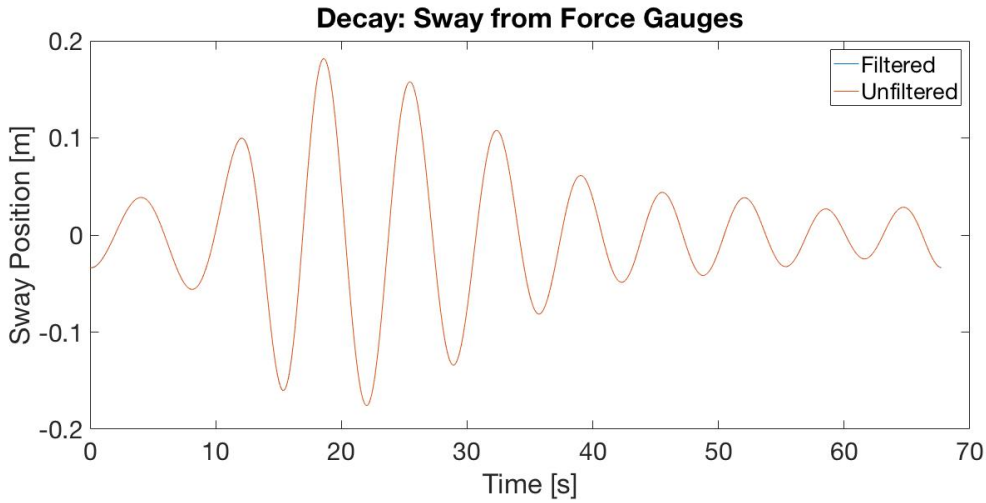


(b) Decay Sway - Accelerometers: The frequency domain of the position signal measured by the accelerometer.

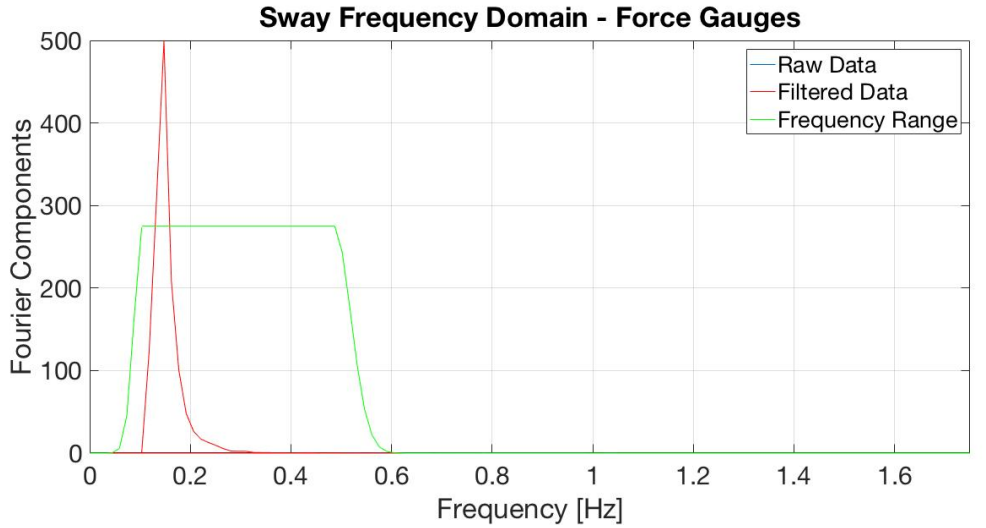
Figure 5.3: Decay Sway - Accelerometers: Figure 5.3a shows the filtered and unfiltered position signal during sway decay, where the signal is found from the accelerometers. Figure 5.3b is the corresponding frequency domain for the sway decay test using the accelerometer measurements. The peak in the frequency domain is the natural frequency of sway which equals a natural period of 6.77 s.

Figure 5.3 shows the filtering and frequency domain obtained by using the accelerometers in sway direction on the model. The frequency domain in Figure 5.3b shows one peak which results in a natural period of 6.77 s. A low-pass frequency of 0.12 Hz is utilised in the frequency domain to filter out the noise in the signal. However,

there are uncertainties in regards to the frequency range of the noise and to ensure the correct low-pass frequency is utilised, the decay test in sway is also investigated using the measurements of the force gauges. The method is described in Section 3.2.2 and Figure 5.4 shows the filtering and frequency domain of the force gauge measurement during a sway decay test.



(a) Decay Sway - Force Gauges: The filtered and unfiltered position signal of the model obtained from the force gauges.

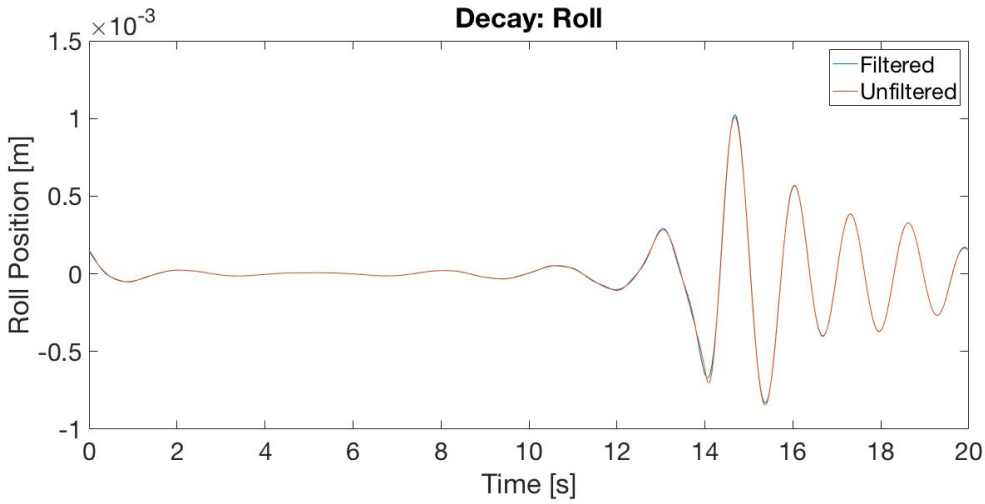


(b) Decay Sway - Force Gauges: The frequency domain of the position signal measured by the force gauges.

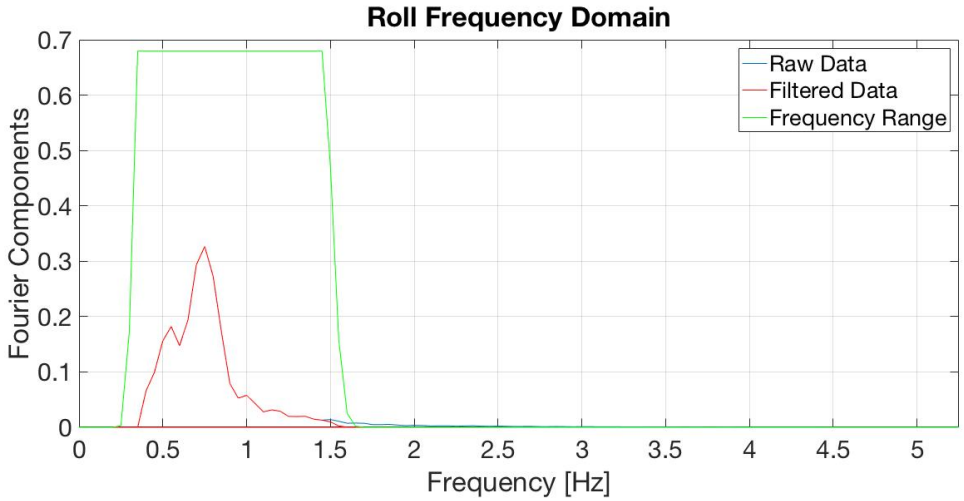
Figure 5.4: Decay Sway - Force Gauges: Figure 5.4a shows the filtered and unfiltered position signal during sway decay, where the signal is found from the force gauges. Figure 5.4b is the corresponding frequency domain for the sway decay test using the force gauge measurements. The peak in the frequency domain is the natural frequency of sway which equals a natural period of 6.77 s, similar to the natural period found from the accelerometers.

The decay signal from the force gauges is identical to the signal from the accelerometers. The peak of the frequency domain in Figure 5.4b is as evident, and at the same frequency, as the accelerometer results. From experience force gauges have less noise at low frequencies, hence the similar results indicate that the chosen low-pass frequency is filtering out the noise in the signal.

The decay test in roll involved exciting the model in roll, which was done by pushing the model on one end while the other end was freely-floating. The filtered measurement and the frequency domain are plotted in Figure 5.5.



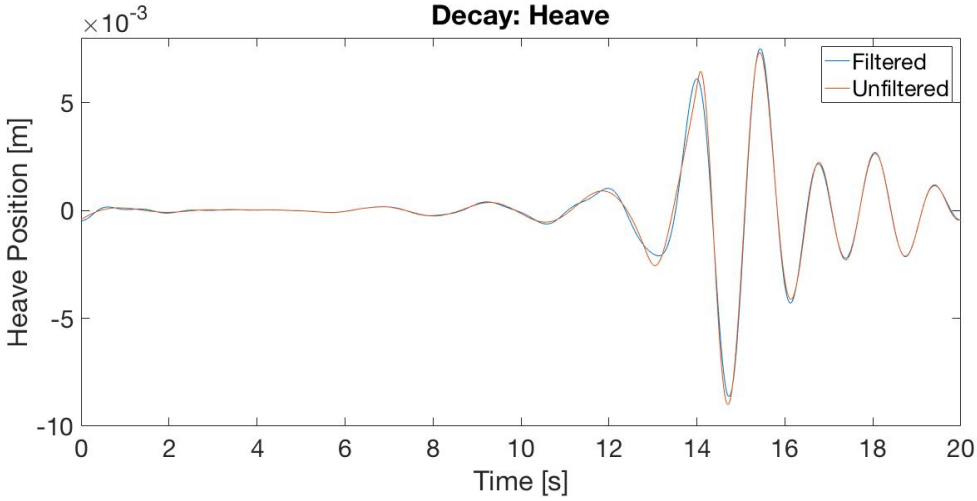
(a) Decay Roll: The filtered and unfiltered position signal obtained from the accelerometer measurements.



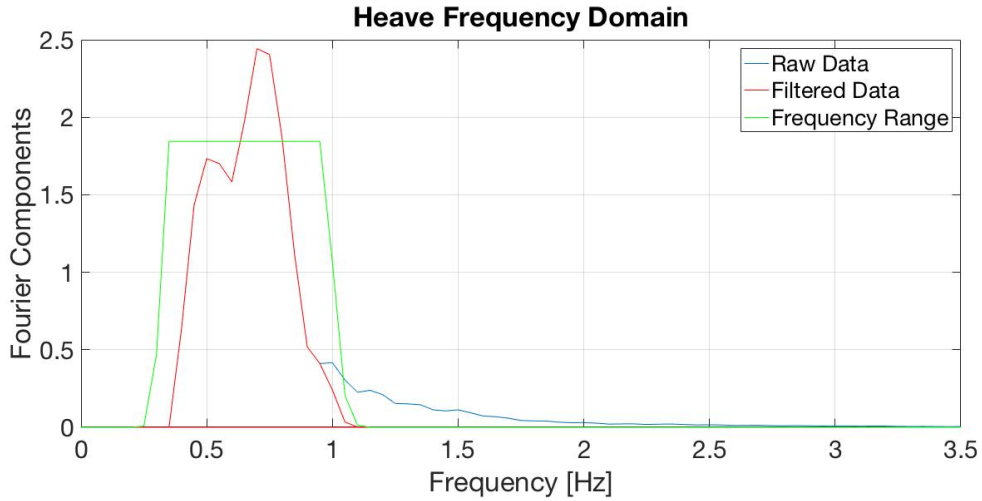
(b) Decay Roll: The frequency domain of the signal is plotted and from the frequency domain the natural period is found. The natural period corresponds to the highest peak in the frequency domain.

Figure 5.5: Decay Roll: 5.5a shows the filtered and unfiltered position of the model while Figure 5.5b plots the frequency domain of the position. From the peak in the frequency domain the natural period in roll is found to be 1.33 s.

The frequency domain of the roll motion shows a main peak at 0.75 Hz, which results in a natural period of 1.33 s. A low-pass frequency of 0.4 Hz is utilised to filter out the noise in the signal and it is observed that a lower low-pass frequency results in a peak around 0.1 Hz, which is noise in the accelerometer measurements. As seen in Figure 5.5a, the roll decay test is cut eight seconds after the model is excited, due to return waves which disturb the decay test.



(a) Decay Heave: The filtered and unfiltered position of the model during decay test in heave.



(b) Decay Heave: The frequency domain of the position signal is plotted and the peak of the plot corresponds to the natural frequency.

Figure 5.6: Decay Heave: Figure 5.6a shows the filtered and unfiltered signal for the position of the model found from the accelerometer measurements. Figure 5.6b plots the frequency domain of the position signal and the peak of the frequency domain corresponds to the natural frequency in heave. The natural period found from this peak equals 1.43 s.

The heave decay signal is extracted mathematically from the roll decay measurements using Equation 3.5, because it was found difficult to excite the model in heave. Due to the width of the model it easily crashed in the tank walls and it was found too challenging to excite the model evenly on both sides. The results from the heave decay are plotted in the Figure 5.6.

The peak of the frequency domain in Figure 5.6b is found at 0.70 Hz which equals a natural period of 1.43 s. Because the heave decay is found from the roll motion decay, the signal is again cut after around eight seconds to avoid the return waves. The heave signal has a lot of noise and the low-pass frequency is set to 0.4 Hz, which results in the filtering and frequency domain observed in Figure 5.6. With a lower low-pass frequency, the frequency domain is full of peaks and no conclusion can be made about the natural period. Due to the high low-pass frequency, another method of calculation was utilised to estimate the natural period in heave to confirm the choice of low-pass frequency.

The natural period in heave can be calculated from Equation 2.41 in Section 2.1.7, with the equation seen below.

$$T_3 = 2\pi\sqrt{\frac{M + A_{33}}{C_{33}}} \tag{5.1}$$

The mass of the model is known and the restoring term can be calculated using the following relation

$$C_{33} = \rho g A_w \tag{5.2}$$

Where A_w is the water plane area of the model, which in this case equals $A_w = 0.59$ m$\times$0.50 m$= 0.195$ m^2.

The two-dimensional added mass is estimated from numerical results in *The Added Mass of Two-Dimensional Cylinders Heaving in Water of Finite Depth* [Bai, 1977]. The paper involves circular and square cylinders without bilge boxes. The model is therefore approximated as a square cylinder with a beam the size of the model, including the bilge boxes. This is illustrated in Figure 5.7, where the blue box are the dimensions utilised in the calculations while the shaded model is seen in the background.

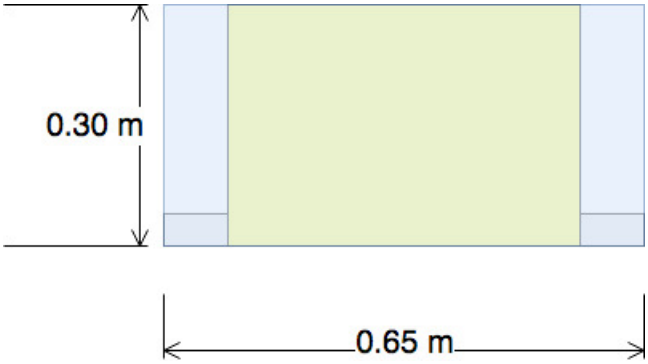


Figure 5.7: The blue box with dimensions is utilised in the calculation of the added mass coefficient, while the actual shape of the model is seen in a shade behind the box.

The estimation results in a natural period in heave which equals 1.22 s and a frequency of 0.82 Hz. Another calculation which includes the bilge boxes by using flat plate approximation to estimate a value of the two-dimensional added mass of the bilge boxes results in a heave natural period of 1.28 s and a natural frequency of 0.78 Hz. The estimates show that the natural frequency has a value above the chosen low-pass frequency and thereby confirms the choice of filtering. The two calculations of the natural period utilise the uncoupled natural period described in Section 2.1.7. These natural periods are uncoupled and undamped and the difference between the calculated and measured natural periods are the coupled terms and the damping in the system.

5.1.3 Response Amplitude Operator

The amplitude of the motions are used to plot the RAO for each direction, namely sway, heave and roll. Section 3.2.4 describes the calculation of the RAOs from the accelerometer measurements. The motions of the model are plotted in Appendix B and only the response amplitude operator results will be discussed.

In the Figure 5.8 the response amplitude operator in sway is plotted for the tested wave frequencies and three of the wave steepnesses.

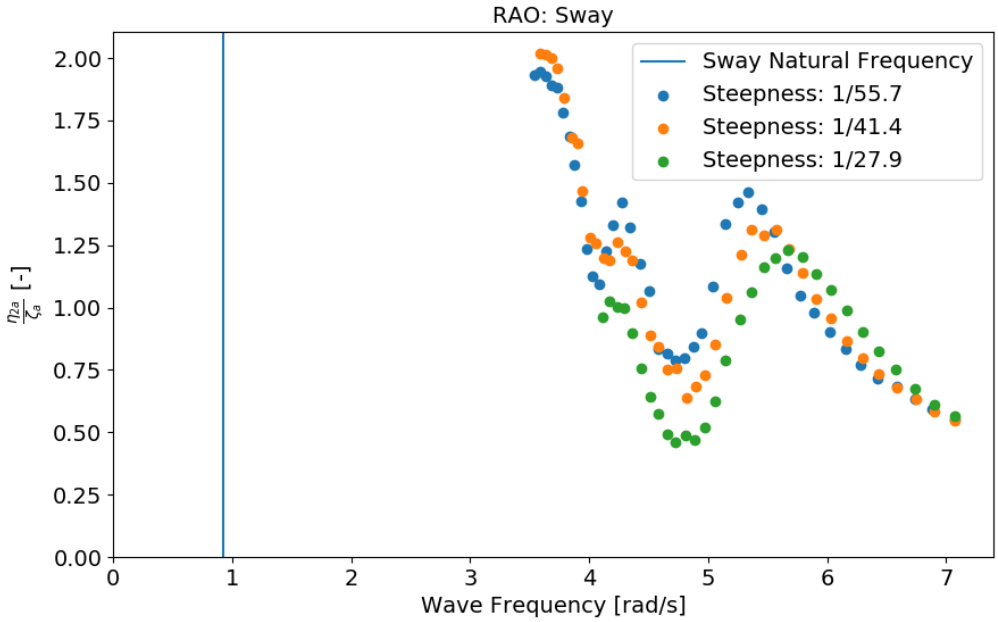


Figure 5.8: Experimental Results - Sway RAO: The Response Amplitude Operator in Sway is plotted for three wave steepnesses. The natural period in sway is indicated with the blue line.

The sway motion has a natural frequency lower than the tested frequency range. This means that the RAO peak is not reached in the measured motion. For the lowest tested frequencies the sway RAO jumps from around 1.0 to 2.0, which is most likely due to the inconsistency of the wave steepness of the experimental waves and because the model is moving in transient-state instead of steady-state motion. The three plotted wave steepnesses show a similar trend in sway RAO, but as the wave frequency decreases the steepest wave has a distinct lower sway RAO compared to the two other wave steepnesses.

The RAO development seen in Figure 5.8 does not show any sign of being affected by the bug in the signal processing system. Around 4.7 rad/s the sway is reduced, which is due to the coupled sway-roll-motion since the roll motion has a natural frequency at 4.71 rad/s. Due to the discovery of the bug in the signal processing system in Ladertanken, the accelerometer measurements were inspected. In Figure 5.9 below the accelerometer measurements are plotted for one wave run.

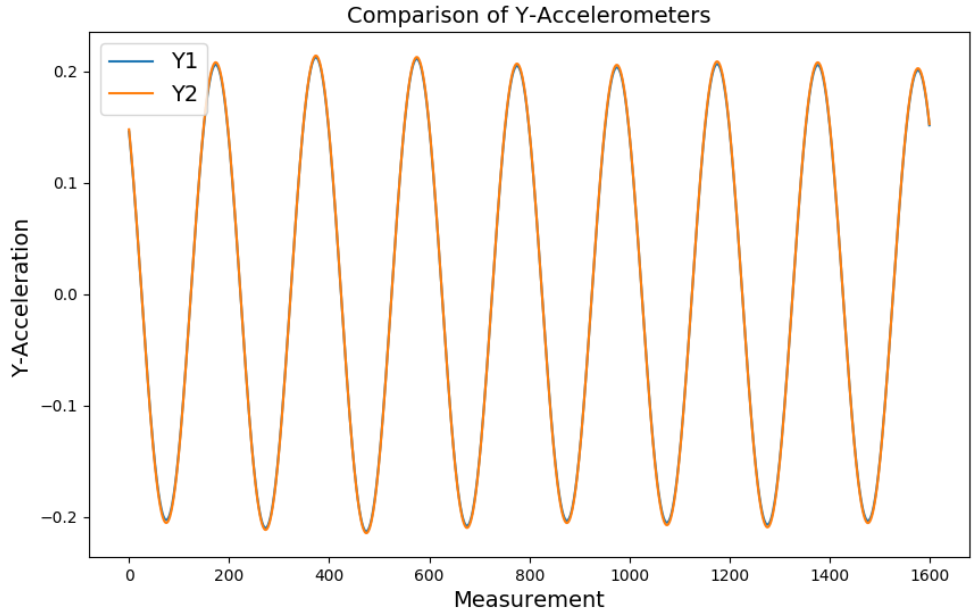


Figure 5.9: Comparison of the accelerometers measuring the motion of the model in y-direction. The two accelerometers show an identical amplitude and are therefore not affected by the bug in the signal processing software.

The above representation is only valid for one of the tested waves, but around hundred random waves were investigated and showed the same result. Both accelerometers show the same amplitude and it is assumed that this is valid for all tested wave conditions. Since the two accelerometers measure a close to identical value, the accelerometers are not affected by the bug in the software.

The RAO in heave is plotted in Figure 5.10 for the tested wave frequencies and three chosen wave steepnesses.

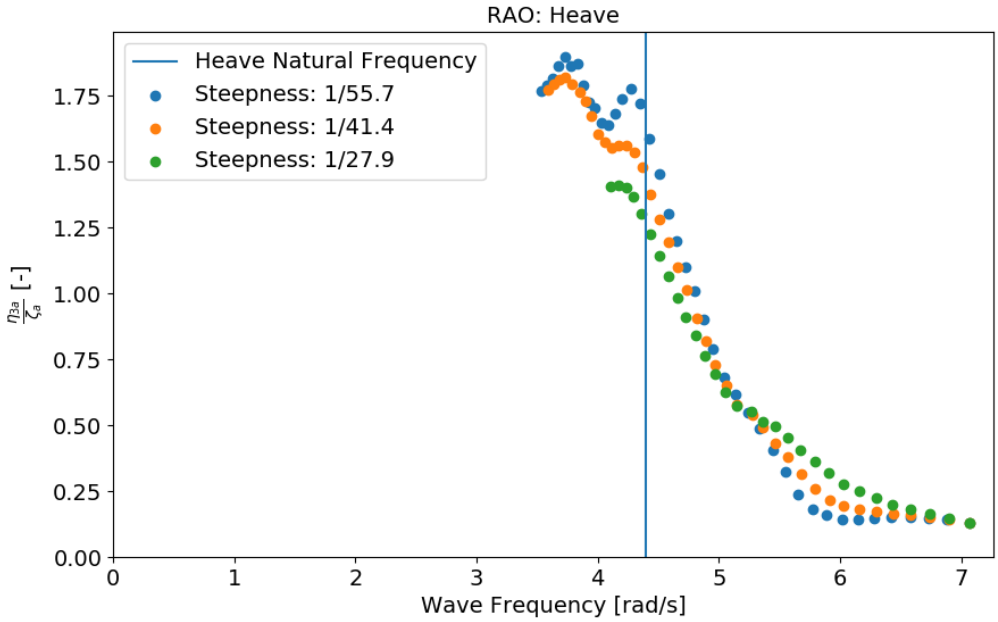


Figure 5.10: Experimental Results - Heave RAO: The Response Amplitude Operator in Heave is plotted for three wave steepnesses. The natural frequency is indicated as the blue line which corresponds well with the peak in the heave motion.

The natural frequency in heave is around the peak of the heave RAO, as seen in Figure 5.10. A higher peak is observed at lower frequencies for two of the wave steepnesses, however the low frequencies correspond to long wave periods which is when the model has a longer transient-state. The peak at the lower frequencies is therefore suspected to be due to the model not reaching steady-state motion at the lower wave frequencies. The peak of the heave RAO is expected around the natural frequency of heave, which is the case in the Figure 5.10, if the low wave frequency results are disregarded. This leads to assuming that the measurements are not affected by the bug discovered in the signal processing system.

The roll response amplitude operator plot is seen in Figure 5.11 below.

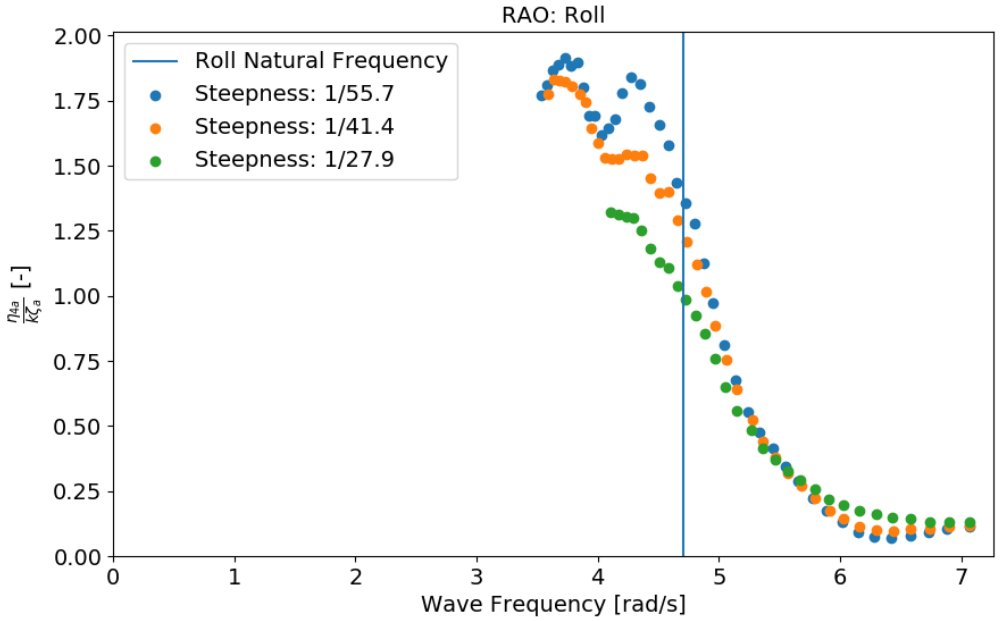
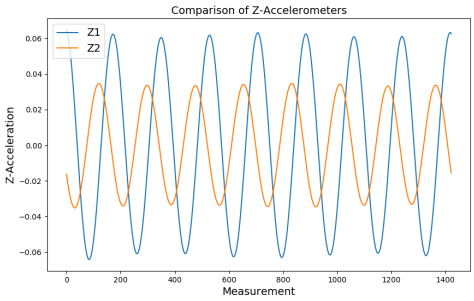


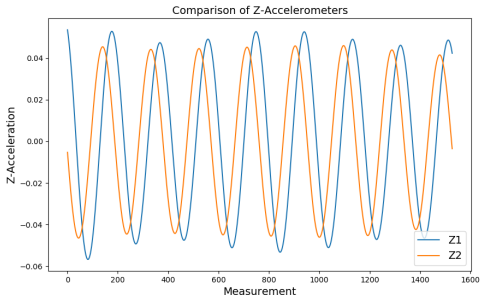
Figure 5.11: Experimental Results - Roll RAO: The Response Amplitude Operator in Roll is plotted for three wave steepnesses together with the natural period in roll. The roll motion is very similar to the heave motion and the peak is found closer to the heave natural frequency than the roll natural frequency. This could be an indication that the measurements are affected by the bug in the software.

The roll RAO plot in Figure 5.11 is very similar to the heave RAO in Figure 5.10. The peak of the roll RAO is closer to the natural frequency in heave, 4.40 Hz, than the natural frequency in roll, 4.71 Hz. This could be an indication that the accelerometers are affected by the bug in the signal processing systems, hence the measurements of the accelerometers in z-direction are analysed.

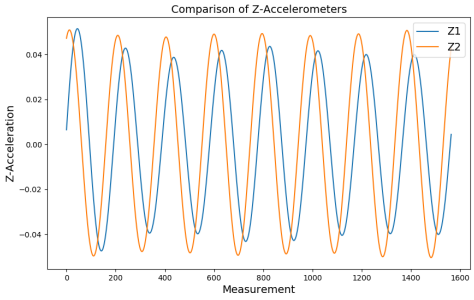
In order to investigate if the z-accelerometers are affected by the bug in the signal processing system, the z-accelerometers are compared in comparison plots. However, the z-accelerometers will capture both the heave and roll motion of the model. Thus, the accelerometers do not show the exact same values, which is seen in the comparison plots in Figure 5.12.



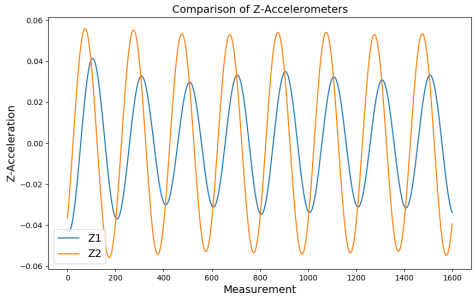
(a) T = 0.88 s - Steep = 1/55.7



(b) T = 0.96 s - Steep = 1/55.7



(c) T = 0.98 s - Steep = 1/55.7



(d) T = 1.00 s - Steep = 1/55.7

Figure 5.12: Comparison of the accelerometers measuring in z-direction. The comparison shows that the accelerometers measure a different amplitude in z-direction, which can be explained by the physical motion of the model. The model will move both in heave and in roll, hence the two accelerometers will measure different amplitudes in z-direction. It is therefore hard to draw a conclusion about the bug in the signal processing software.

The accelerometers in z-direction measure different amplitudes and there is no consistency in which accelerometers measures the highest value, as seen in Figure 5.12. However, the difference can be explained by the physical motion of the model which will include both heave and roll motion. Hence, the model can move upwards in heave as well as rotating in roll which leads to different amplitude of the measurements in z-accelerometers. This physical motion is drawn in Figure 5.13, where the model is moved upwards and rotated, thus moved in both heave and roll motion.

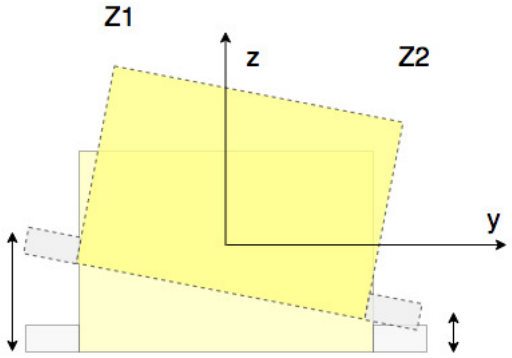


Figure 5.13: An illustration of a snap shot while the model is moving in waves with both heave and roll motion. The heave leads to the model lifting upwards and the roll motion rotates the model.

Figure 5.13 explains the physical motion of the model and the reason for the difference in the accelerometer measurements. Due to uncertainty of the expected difference between the accelerometer measurements on the two opposite sides of the model, one can not rule out that the accelerometers in z-direction are affected by the bug in the signal processing system. The similarity between the heave and roll RAO plots indicate that there is an error and the roll RAO is expected to have a peak around the natural frequency of roll. Both heave and roll utilise the z-accelerometers in the calculation of motions, as seen in Equation 3.2 and Equation 3.4 in Section 3.1.5. Due to the roll motion using the difference between the accelerometer measurements, it is expected that the error is more evident in the roll results compared to the heave results. The heave motion is calculated using the average of the two accelerometers, hence a factor of two between the accelerometers will be less prominent.

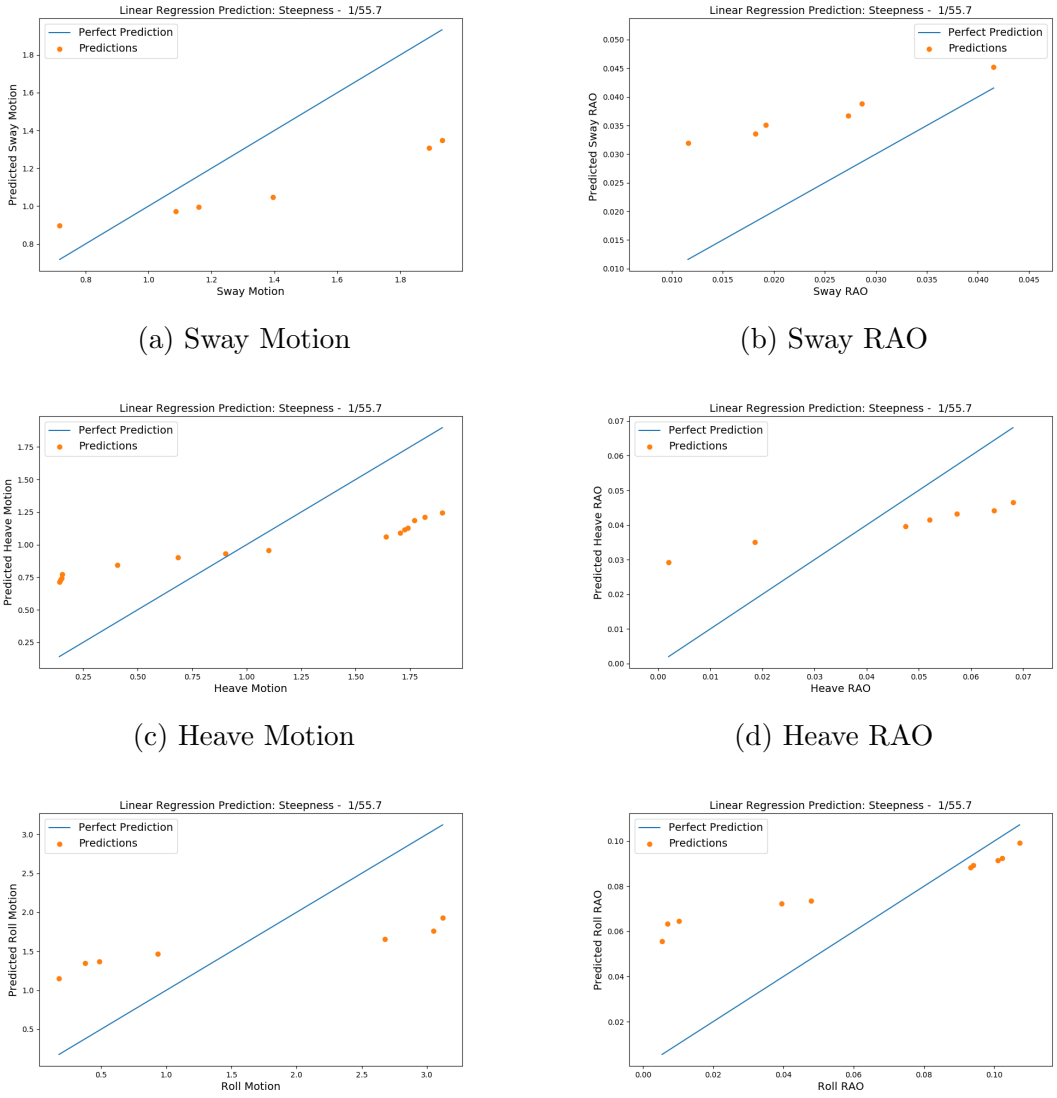
The similarity of the RAO in roll and heave can also be due to wave reflections and the model not achieving steady-state motion. The measurements are obtained from a small time window to avoid wave reflection, as explained in Section 3.2.3. By using the group velocity of the waves, the time window is cut before the wave return to the wave probes. However, the front of the wave will move faster and the results can therefore include some wave reflections. This leads to uncertainty if the model motion has reached steady-state, and most likely the model is still in transient state.

The results of the RAO in roll can indicate that the motion has not reached steady-state motion and the RAO and motion results are not from steady-state conditions.

The focus of this thesis is investigating if Neural Networks can be used to predict the motion of a floating structure. The uncertainties and possible errors in the experimental data are therefore not important, because these will be included by the Neural Network predictions. However, the results can not be utilised to investigate hydrodynamic effects because of the errors in the experiment.

5.2 Linear Regression

The linear regression models which utilises the Mean Squared Error as loss function and Stochastic Gradient Descent to optimise the prediction are presented in this section. The wave period and wave heights are the features, while the model motion and RAOs are the targets. In the plots below, Figure 5.14, the predictions are plotted versus the known value for wave steepness 1/55.7. Results for wave steepness 1/41.4 and 1/27.9 are plotted in Appendix C.



(a) Sway Motion

(b) Sway RAO

(c) Heave Motion

(d) Heave RAO

(e) Roll Motion

(f) Roll RAO

Figure 5.14: Linear Regression Results: Prediction for wave steepness 1/55.7 for motion and response amplitude operator in sway, heave and roll. The plots show that Linear Regression is not a suitable tool to predict the motion of the model.

In Figure 5.14, the scatter plots are the predicted values while the blue line is the line of perfect prediction. If the prediction is the exact same value as the true value from the experiments, it lies on the blue line which equals $x = y$ and is known as the line of perfect prediction. It is clear from the prediction results that a simple linear regression model is not capable of predicting either the motion or the RAO of the model. This shows that the problem at hand consists of non-linear relations which needs a Neural Network in order to make more accurate predictions. The Coefficient

of Determination for the six vessel responses are given in Table 5.3 below.

Table 5.3: Linear regression response prediction results using the Coefficient of Determination

Response	**R^2**
Sway	37.0 %
Heave	42.0 %
Roll	39.0 %
RAO 2	34.0 %
RAO 3	30.0 %
RAO 4	37.0 %

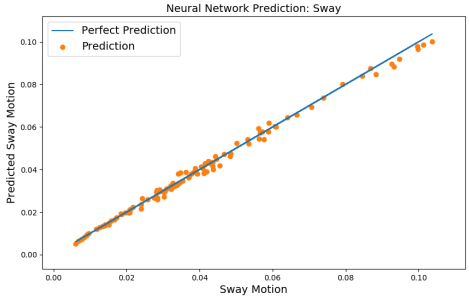
None of the linear regression models are capable of predicting anything above 41 % of the values. The linear regression model is included to show how much better a neural network can predict values when the features and targets can not be described by a linear relationship.

5.3 Neural Network

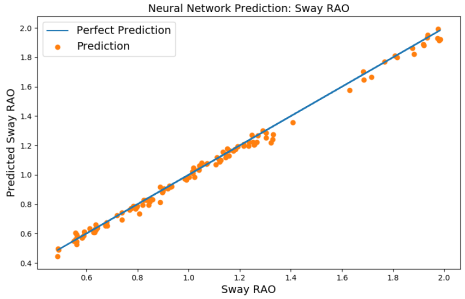
The following section presents the results of the response prediction using Neural Network and discusses the aspects of Neural Networks. The prediction results for sway, heave, roll as well as the corresponding RAOs are found as larger plots in Appendix D.

5.3.1 Response Predictions

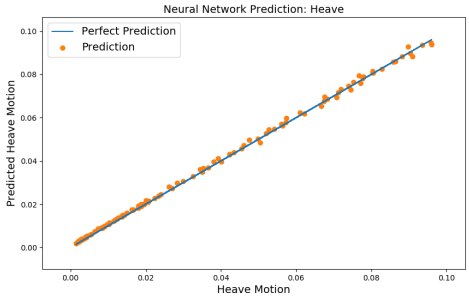
The predictions of the response of the model are plotted against the true value in Figure 5.15.



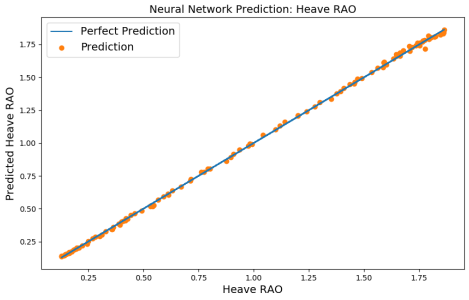
(a) Sway Motion Prediction



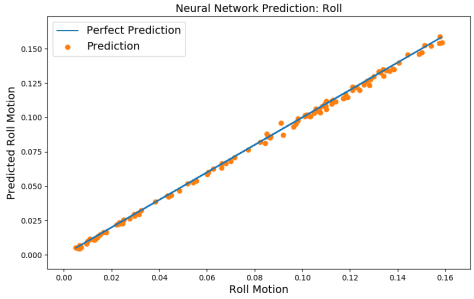
(b) RAO Sway Prediction



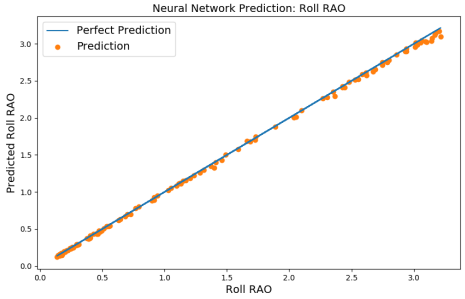
(c) Heave Motion Prediction



(d) RAO Heave Prediction



(e) Roll Motion Prediction



(f) RAO Roll Prediction

Figure 5.15: Neural Network - Response Predictions: The above plots show the predicted value versus the true value of the model response. The line of perfect prediction is plotted in blue and the orange scatter plots are the predicted values. The Neural Network is capable of predicting the target with a high accuracy, however, one can observe that the sway motion and sway RAO has a larger deviation from the line of perfect prediction.

Figure 5.15 generally indicates very good results for the Neural Network predictions. The prediction results plotted as scatter plots are very close to the perfect prediction line which is plotted in blue. Heave motion and heave RAO looks especially good with the scatter plot following the perfect prediction line closely. The sway motion and RAO are the two plots which have the largest deviation from the line of perfect

prediction, which will be discussed in Section 5.3.2.

The model accuracy oscillates around a value due to the random data set split and due to the model training slightly different each time. The model has been optimised to a certain extent, but the variation of model accuracy indicates that the model can still be further optimised with regards to the number of layers, number of nodes in each layer and other configurations, as mentioned in Section 4.2.4. Each model is trained ten times in order to evaluate the variation in the model accuracy and calculate the standard deviation of the accuracy metrics. Table D.2 in Appendix D summarises the prediction accuracy for all ten runs in all motions and RAOs, and the calculated average which is used as the final prediction accuracy. Table 5.4 summarises the average Coefficient of Determination, the average Mean Absolute Percentage Error for the motions and RAOs and the standard deviation of both metrics.

Table 5.4: Neural Network prediction results using Coefficient of Determination and Mean Absolute Percentage Error, as well as their standard deviations

Response	$\mathbf{R^2}$	**MAPE**	$\sigma(\mathbf{R^2})$	$\sigma(\mathbf{MAPE})$
Sway	98.0 %	5.7 %	1.0 %	1.9 %
Heave	99.6 %	8.5 %	0.3 %	3.2 %
Roll	99.5 %	6.2 %	0.5 %	3.5 %
RAO Sway	99.0 %	2.4 %	0.6 %	0.4 %
RAO Heave	99.9 %	2.7 %	0.0 %	1.0 %
RAO Roll	99.9 %	2.8 %	0.0 %	0.7 %

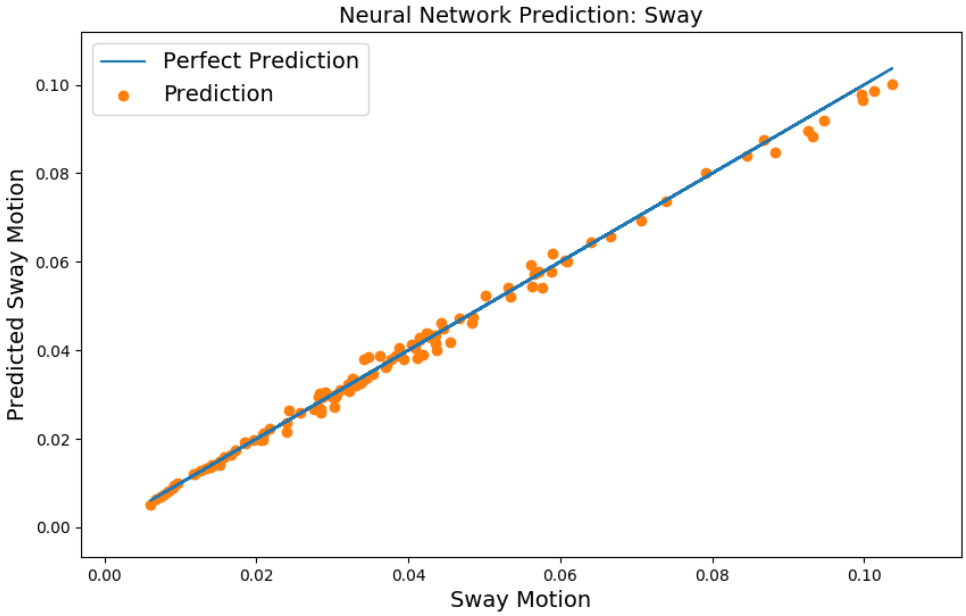
All six models have an R^2 accuracy above 98 % and the $MAPE$ values are all below 10 %. However, one can observe from Table 5.4 that there is an inconsistency between the two accuracy metrics. The standard deviation of the two metrics show that the R^2 has stabilised, while there is still some deviation in the $MAPE$ metric for the three motions, sway, heave and roll. Due to a time consuming optimisation

process of the Neural Network, the obtained standard deviations were deemed small enough to show the use of Neural Networks to predict model motion. The values do however show that there is room for optimisation for the three motion models, although the RAO models indicate being close to optimal.

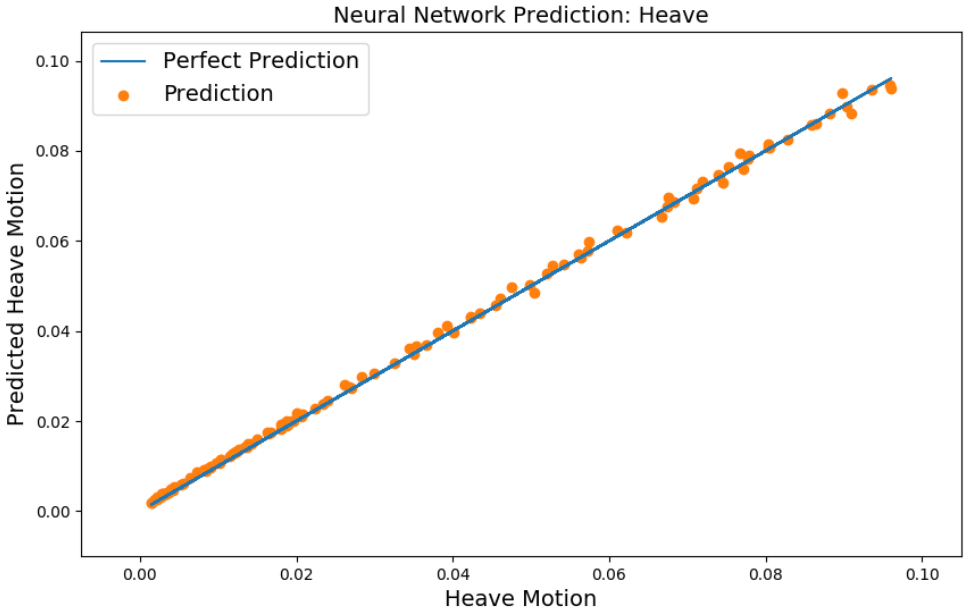
5.3.2 Accuracy Metrics

From the R^2 metric the accuracy is very high for all six models, while the Mean Absolute Percentage Error paints a different picture. All six targets have a Coefficient of Determination equal to 98.0 % or higher, all very close to perfect prediction which is $R^2 = 100\%$. As observed on the prediction plots in Figure 5.15, the predictions are not flawless and do have a difference from the perfection prediction line. Using the Mean Absolute Percentage Error the accuracy of the six targets ranges from 8.5 % down to a value of 2.4 %, where 0.0% is perfect. The difference between the two measures of error are due to the mathematical definition of the metrics, R^2 utilises the mean value of the target which $MAPE$ does not. All six targets have a non-linear development of the target values and using the mean value of the target therefore does not describe the problem sufficiently.

Looking closely at the prediction plots in Figure 5.15, the predictions in sway motion and RAO seem to deviate the most from the perfect prediction line. The sway and heave motion plots are plotted in larger format in Figure 5.16.



(a) A larger plot of the Neural Network sway prediction.



(b) A larger plot of the Neural Network heave prediction.

Figure 5.16: Larger plots of the Neural Network predictions in sway and heave. The heave predictions are closer to the perfect prediction line, while the sway predictions have a larger error. From the two plots it is seen that the predictions in sway have a higher error at larger motions, which is not as evident in heave.

Comparing the prediction plots between sway and heave motion, which both have the same x- and y-scale, the sway motion clearly deviates more from perfect predic-

tion. However, Table 5.4 shows that the $MAPE$ value for heave is larger than for sway motion, meaning sway is more accurate according to $MAPE$. The numerical error estimate is an average value of ten trained models, while the plot is a single trained model plot. Thus, the average $MAPE$ value might not correspond to the plotted prediction plot.

Another important aspect of the misleading value of R^2 is the convergence plotted in Figure 4.1. The Coefficient of Determination is utilised to monitor the learning process and ensure that the amount of epochs is high enough. However, as seen in Figure 4.1, the value of R^2 is higher than 99.0 % already at around 10 epochs. This would indicate that the Coefficient of Determination has converged, but during optimisation it is observed that the Mean Absolute Percentage Error converges later than R^2. Hence, the number of epochs necessary is found higher than the indication of the convergence plot of R^2 in Figure 4.1 and $MAPE$ should have been utilised in the convergence plots.

5.3.3 Aspects of Neural Network

The Neural Network models created to predict the motion and RAO are successful and all hold a $MAPE$ value below 10%, corresponding to an accuracy of above 90 % for all six targets. Compared with the results from the Linear Regression predictions in Figure 5.14 the Neural Network is clearly superior. Linear Regressions are not commonly utilised to predict the motion of floating structure, hence it does not prove the performance of the Neural Networks. Empirical methods and numerical calculations are common methods to calculate the response of a model as mentioned in Section 2.1.4, and it is of interest to compare the performance of the Neural Network against the empirical and numerical results. This is out of the scope of this thesis, one can therefore not conclude if the Neural Network is outperforming the common methods.

The general indication of the R^2, $MAPE$ and prediction plots is that the Neural Network can successfully predict the motion and RAO of the experimental model.

The Neural Network has the highest accuracy on the three RAOs, where the motions are made dimensionless using the wave amplitude and wave number. It is unclear exactly why the Neural Network works better with the RAOs compared to motions, but it is most likely due to the use of wave amplitude in the RAOs. The Response Amplitude Operators are calculated by dividing the motion amplitude by the wave amplitude, as seen in Equation 3.8 in Section 3.2.4. Hence, the wave amplitude is indirectly in the target as well as being one of the features. This could be why the RAO predictions have a higher accuracy than the motions, because the RAO includes one of the features. Situations like these, where it is not known why the Neural Network performs better for some features and targets compared to others is part of the challenge with Machine Learning. Mathematically, the process of calculations for Neural Networks is known. However, the randomly initialised weights and biases are unknown and this contributes to the uncertainty of the behaviour of Neural Networks. It is possible to train two separate models with different weights, but with the same results. Additionally, one can not trace back using the weights and obtain the function $F(x)$ which maps the input to the output. This leads to Neural Networks and Machine Learning in general having a reputation of being a *black box* tool. This is also why it is hard to understand *why* the Neural Network reacts the way it does to changes in the configuration or features.

As mentioned in the theory section, Section 2.2.5, the final product is a complete, trained Neural Network. This investigation has compared the average of ten trained models and utilised this as the accuracy of the model. However, if the models are to be used one can pick the model with the highest accuracy out of the ten. The analysis of the performance of the models is done to show that the models are not fully optimised and can be tuned to obtain a higher accuracy. This was found to be very time consuming and due to time constraints this was not prioritised.

Another important note is the use of Neural Network to predict outside the range of the input data. Neural Networks, and Machine Learning in general, are simply an advanced methods of curve-fitting. Extrapolation is therefore advised against,

because one does not know the behaviour of the Neural Network outside of the input data range. The performance of the Neural Network if used for extrapolation is unknown, random and highly discouraged.

Chapter 6

Discussion of Application

The following section will elaborate the way Neural Network predictions can be used to learn more about viscous residuals as well as in full scale floating structures and in the design process.

6.1 Viscous Residuals from Model Experiments

As seen in the derivation of non-linear coupled motions in Section 2.1.7, the full equation of motion with non-linear terms is both complex and difficult to solve. Several methods of simplification is presented in Section 2.1.7, with both linear, coupled natural periods and uncoupled natural periods. The present study suggests to investigate the possibility of using Neural Networks to investigate the viscous terms.

Using the measured motion and the force in the springs in the experiment, the unknowns in the Equation of Motion are reduced to the Added Mass, Damping and Excitation Forces. Equation 6.1 below shows the equation of the uncoupled roll motion.

$$\big(I_{44} + A_{44}\big)\ddot{\eta}_4 + B_{44}\dot{\eta}_4 + C_{44}\eta_4 = F_{exc,4} \tag{6.1}$$

However, the added mass, damping and excitation terms can be written as a poten-

tial term and a viscous term as follows.

$$\left(I_{44} + A_{44}^{pot} + A_{44}^{visc}\right)\ddot{\eta}_4 + \left(B_{44}^{pot} + B_{44}^{visc}\right)\dot{\eta}_4 + C_{44}\eta_4 = F_{exc,4}^{pot} + F_{exc,4}^{visc} \tag{6.2}$$

The potential terms can be found numerically by using a potential flow solver, which leaves only the viscous terms. These can be collected to a viscous residual, $F^{residual}$.

$$\begin{aligned}\left(I_{44} + A_{44}^{pot}\right)\ddot{\eta}_4 + \left(B_{44}^{pot}\right)\dot{\eta}_4 + C_{44}\eta_4 - F_{exc,4}^{pot} &= F^{residual} \\ \left(A_{44}^{visc} + B_{44}^{visc} + F_{exc,4}^{visc}\right) &= F^{residual}\end{aligned} \tag{6.3}$$

The viscous damping term, B_{44}^{visc} can be found from forced oscillation experiments as the non-linear term in Equation 2.8, narrowing down the unknowns to only $A_{44}^{visc} + F_{exc,4}^{visc}$. Using Muhammad Mukhlas method to extract the added mass from forced oscillation experiments [Mukhlas, 2017], the residual is left to only $F_{exc,4}^{visc}$.

The above separation of the viscous terms can be utilised to implement the viscous damping, added mass and excitation force to create Neural Networks which are trained to predict added mass, damping and excitation forces more accurately. The above derivations are necessary to split the problem from one force residual to the individual viscous coefficient terms. This method can be utilised if both freely floating experiments and forced oscillation experiments are conducted for the model, and the features are the wave height and wave steepness.

Using forced oscillations to find viscous damping and added mass, a Neural Network can be trained to predict the viscous excitation force. However, one can also include the viscous excitation force and focus on the added mass or the damping term, and investigate their development for different sea states. The Neural Network can find patterns in the viscous terms which enhances the understanding of the phenomenon.

Section 2.1.4 mentioned CFD as a method to approximate the added mass and damping terms. In Muhammad Muklash's investigation of roll damping of a two-dimensional ship section with bilge boxes [Mukhlas, 2017], CFD was utilised to

approximate the added mass and damping terms. The numerical method is capable of predicting the added mass and damping terms up until a certain roll amplitude, where it is suspected that the motion of the model affects the free surface. Due to the linear modelling of the free surface, these effects are not captured by the CFD approximation. This shows the limitation of CFD and this is where the Neural Network predictions of added mass and damping can be of interest. Using the above method utilising Neural Networks, one can further investigate the development of added mass and damping by conducting experiments and training a Neural Network to predict the values. One can from the predictions plot the development of the added mass and damping and learn more about the effects of the bilge boxes and flow separation on the specific model.

It is important to mention that the Neural Network method to investigate the viscous effects utilises assumptions which can affect the results. It is assumed that the viscous added mass and damping terms can be approximated from the forced oscillations. This is equivalent to solving the radiation problem and assume this is valid for the freely-floating solution, as presented in Section 2.1.3. Hence, the effect of the excitation force on the viscous terms is neglected.

6.2 Moored Structures

Modern vessels are full of sensors measuring the motion, forces in anchors and mooring lines and the condition of equipment on board. This gives access to large data sets with huge potential for Machine Learning. Similarly to the model developed in this thesis, a model can be trained to predict the motion of a full scale moored structure.

The Neural Network motion prediction utilised the wave period, wave steepness and the measured motion in the direction of interest. Modern floating structures have sensors which monitor their position, meaning the motion of the structure is known. There are wave probes available which can give information about the wave period

and wave steepness at a given time, thus all the information needed to train a Neural Network is available. The model can be trained and used to predict the position of the structure in the present sea state. If there is a large deviation from predicted and actual motion, it can be an indication of defect in the mooring lines.

6.3 Design Process

As mentioned in the introduction in Section 1, the development of unconventional ship hulls increases the need of better ways to estimate the viscous effects. Unconventional ship hulls commonly have geometrical singularities which increase the viscous effects, such as flow separation and vortex shedding. This leads to the need of tools to estimate the added mass and damping coefficients where viscous effects are of importance. However, the Neural Network predictions are dependent on data from model experiments or full scale structures. It is therefore not a tool which can be utilised in the design process. The method developed to obtain the viscous residual as described in Section 6.1 can be used to learn about the development of viscous residuals and viscous terms in different sea states. This new knowledge can possibly be utilised to develop better numerical or empirical methods to estimate the response of unconventional ship hulls, but is on its own not capable of doing so.

Chapter 7

Conclusion and Further Work

7.1 Conclusion

This Master's Thesis has investigated the possibility of training a Neural Network to predict the motion of a floating structure. A model shaped like the mid-section of a ship with large bilge boxes is utilised in the experiments conducted in Ladertanken. The experiment simulates a freely-floating structure, however the model is restrained using two springs on each side to restrict it to an area of the wave tank. Decay tests are conducted to identify the natural periods of the model in sway, heave and roll. The natural periods are found to be 6.77 s in sway, 1.43 s in heave and 1.33 s in roll. The wave periods tested range from 0.88 s to 1.78 s in model scale and the wave steepness ranges from 1/60 to 1/20. There are some uncertainties about the experimental results due to a bug in the software which converts the instrumental measurements from analogue to digital values. However, because the results are only used as data for the Neural Network, they are accepted. The Neural Networks will include the errors and the results can therefore still be used to judge if Neural Networks can be used for motion predictions

A Linear Regression Model is created to predict the motion of the model, which is shown to perform poorly. The Linear Regression Model is not capable of predicting the response of a model because the responses are non-linear. This is used to show the power of Neural Networks as a prediction tool of non-linear targets.

A total of six Neural Networks are trained to predict the response of the model, namely the motion in sway, heave and roll as well as the Response Amplitude Operator in sway, heave and roll. Experimental wave height and wave period are utilised as the inputs to the Neural Network while the model response are set as targets in their respective models. All six Neural Network Models are developed in Keras, which is a Machine Learning library in Python. The inputs are standardised and shuffled, as well as split into training, testing and validation data sets. The Coefficient of Determination and Mean Absolute Percentage Error are used as accuracy metrics, as well as plotting the predictions against the true value. From the accuracy metrics an accuracy of 98.0 % or higher is obtained using the Coefficient of Determination. Utilising the Mean Absolute Percentage Error, all six Neural Networks are below 10 % and the Response Amplitude Operator predictions in sway, heave and roll result in 2.4 %, 2.7 % and 2.8 % respectively. The difference between the Coefficient of Determination and Mean Absolute Percentage Error is due to the mathematical definition of the accuracy metrics, and it is found that the Coefficient of Determination is misleading. Using the prediction plots where the prediction values are plotted against the true value of the target, it is observed that the Mean Absolute Percentage Error coincides better with the indication of the plots. This shows the importance of using more than one accuracy metric and analysing the results carefully.

The overall conclusion is that the Neural Networks are capable of predicting the response of the freely-floating structure with an error below 10 % in general. The Response Amplitude Operators in sway, heave and roll all obtain an error below 3.0 %, using the Mean Absolute Percentage Error. The present study presents a method to utilise the results to investigate the viscous terms of added mass and damping, which can give valuable information about the development of the viscous effects in larger sea states.

7.2 Further Work

A couple of comments about further work with respect to the topic investigated in the present study.

The damping for the model investigated in this thesis is expected to be mainly due to viscous effects and it would be of interest to investigate this. The damping coefficient can be extracted from the RAO in Roll and by comparing with the empirical damping value from Figure 3.17 in *Sea Loads on Ships and Offshore Structures* [Faltinsen, 1990], one can prove that the damping coefficient is mainly due to viscous effects.

As mentioned in Section 6.1, a potential flow solver can be utilised to estimate the linear added mass and damping. These linear coefficients can be used to further investigate the viscous part of the added mass and damping terms. Another interesting aspect is comparing the results of the Neural Network predictions of added mass and damping to the results of numerical calculations using Computational Fluid Dynamics.

Sevan Marine has expressed interest in the topic and they have conducted experiments with a model in the Ocean Basin at SINTEF Ocean. It would be interesting to train a Neural Network using their model data, which has a lot more data points and larger ranges of wave period and wave steepness. Using more data points will increase the data available for training and thereby also increase the accuracy.

The Neural Network developed in this thesis is only able to predict the response of the specific model from the experiments and is not generalised. One can not predict the motion of a model with a different bilge keel and expect it to be able to predict the response of the new model. For further work it would be interesting to do more experiments with other main dimensions, such as draft, length and bilge keel size. This would train a more generalised Neural Network which can be used to compare

different load conditions or model dimensions.

Only one programming method is utilised to create and build the Neural Networks, and it could be interesting to compare the performance to other Machine Learning libraries and toolboxes. Tensorflow is a popular Machine Learning library in Python and it would be interesting to compare the performance of the Keras models versus a Tensorflow model. The Machine Learning toolbox implemented in Matlab is also of interest and can be compared to the other methods.

Bibliography

[Amazon, 2018] Amazon (2018). Amazon Machine Learning: Developer Guide.

[Bai, 1977] Bai, K. J. (1977). The added mass of two-dimensional cylinders heaving in water of finite depth. *Journal of Fluid Mechanics*, 81(1):85–105.

[Benedict et al., 2003] Benedict, K., Baldauf, M., and Kirchoff, M. (2003). Estimating Potential Danger of Roll Sesonance of Ship Operation. `http://schiw.sf.hs-wismar.de/siw/paper/heft5/beitrag10`.

[Bremer, 2017] Bremer, K. S. (2017). Machine Learning and Cylindrical FPSO Motion.

[Chakrabarti, 2001] Chakrabarti, S. (2001). Empirical calculation of roll damping for ships and barges. *Ocean Engineering*, 28(7):915 – 932.

[Eni, 2018] Eni (2018). 3D Modelling Pictures of Goliat. `http://www.eninorge.com/Global/Bildegalleri/Bildebibliotek/Goliat%203d-bilder/visco-eni_goliat_02.jpg`.

[Faltinsen, 1990] Faltinsen, O. (1990). *Sea Loads on Ships and Offshore Structures*. Cambridge University Press.

[Faltinsen, 2006] Faltinsen, O. M. (2006). *Hydrodynamics of High-Speed Marine Vehicles*. Cambridge University Press.

[Google Developers, 2018] Google Developers (2018). Machine Learning Crash Course.

[Greco, 2012] Greco, M. (2012). *TMR 4215: Sea Loads - Lecture Notes*. NTNU.

[Gulli and Pal, 2017] Gulli, A. and Pal, S. (2017). *Deep Learning With Keras*. Packt Publishing Ltd.

[Hastie, 2009] Hastie, T. J. (2009). The Elements of Statistical Learning: Data Mining, Inference and Prediction.

[Himeno, 1981] Himeno, Y. (1981). Prediction of Ship Roll Damping - State of the Art. *Department of Naval Architecture and Marine Engineering, The University of Michigan*.

[Hinton, 2014] Hinton, G. (2014). Neural Networks for Machine Learning. `http://www.cs.toronto.edu/~tijmen/csc321/lecture_notes.shtml`.

[Hughes, 1993] Hughes, S. A. (1993). *Physical Models and Laboratory Techniques in Coastal Engineering*. World Scientific Publishing Co. Pte. Ltd.

[Ikeda, 1978] Ikeda, Y. (1978). A prediction method for ship roll damping.

[Journée and Pinkster, 2002] Journée, J. and Pinkster, J. (2002). *Introduction in Ship Hydrodynamics*. Delft University of Technology.

[Ketkar, 2017] Ketkar, N. (2017). *Deep Learning with Python: A Hands-on Introduction*. Apress, Berkeley, CA.

[Khan et al., 2005] Khan, A., Bil, C., and Marion, K. E. (2005). *Theory and Application of Artificial Neural Networks for the Real Time Prediction of Ship Motion*, pages 1064–1069. Springer Berlin Heidelberg.

[Kurbiel, Thomas and Khaleghian, Shahrzad, 2017] Kurbiel, Thomas and Khaleghian, Shahrzad (2017). Training of Deep Neural Networks based on Distance Measures using RMSProp.

[Li et al., 2017] Li, G., Kawan, B., Wang, H., and Zhang, H. (2017). Neural network-based modelling and analysis for time series prediction of ship motion.

[Luo et al., 2016] Luo, W., Soares, C. G., and Zou, Z. (2016). Parameter Identification of Ship Maneuvering Model Based on Support Vector Machines and Particle

Swarm Optimization. *Journal of Offshore Mechanics and Arctic Engineering*, 138(3).

[Masi et al., 2011] Masi, G. D., Gaggiotti, F., Bruschi, R., and Venturi, M. (2011). Ship motion prediction by radial basis neural networks. In *2011 IEEE Workshop On Hybrid Intelligent Models And Applications*, pages 28–32.

[Mukhlas, 2017] Mukhlas, M. (2017). Roll damping investigation of two-dimensional ship section with bilge-boxes. NTNU.

[Newman, 1977] Newman, J. N. (1977). *Marine hydrodynamics*. MIT Press, Cambridge, Mass.

[Ng, 2012] Ng, A. (2012). Machine Learning. `http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning`.

[Nicolau et al., 2007] Nicolau, V., Palade, V., Aiordachioaie, D., and Miholca, C. (2007). *Neural Network Prediction of the Roll Motion of a Ship for Intelligent Course Control*, pages 284–291. Springer Berlin Heidelberg.

[Nielsen, 2015] Nielsen, M. (2015). Neural Networks and Deep Learning.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pettersen, 2014] Pettersen, B. (2014). *Marin Teknikk 3 - Hydrodynamikk*. Akademika Forlag.

[Steen, 2014] Steen, S. (2014). *Experimental Methods in Marine Hydrodynamics*. Department of Marine Technology.

[Xing and McCue, 2010] Xing, Z. and McCue, L. (2010). Modeling ship equations of roll motion using neural networks. *Naval Engineers Journal*, 122(3):49–60.

[Ziegel et al., 1999] Ziegel, E. R., Walpole, R. E., Meyers, R. H., and Meyers, S. L. (1999). Probability and Statistics for Engineers and Scientists. *Technometrics*, 41(2).

[Çengel, 2014] Çengel, Y. A. (2014). Fluid Mechanics: Fundamentals and Applications.

Appendix A

Precision Error

It is assumed that the measurements will follow a Gaussian distribution around a mean value if the measurements are repeated infinitely many times. The Gaussian distribution is known as

$$f(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(X-\mu)^2}{2\sigma^2}} \tag{A.1}$$

where X is the stochastic variable, μ is the mean value and σ is the standard deviation. The mean value of the stochastic variable X from N repeated tests is given as

$$\bar{X} = \frac{1}{N} \sum_{j=1}^{N} X_j \tag{A.2}$$

and the standard deviation as

$$S_X = \sqrt{\frac{1}{N-1}} \sum_{j=1}^{N} \left(X_j - \bar{X}_j\right)^2 \tag{A.3}$$

The mean of the stochastic variable, $\bar{X}$, is itself normally distributed with a mean of μ. The standard deviation of the means depends on the number of repetitions as follows

$$S_{\bar{X}} = \frac{S_X}{\sqrt{N}} \tag{A.4}$$

The Student's t-cumulative distribution with N-1 degree of freedom and a confidence interval of 95 % is used and gives the following relation for precision error of the

mean sample

$$P_{\bar{X}} = tS_{\bar{X}} \tag{A.5}$$

and the uncertainty of the mean sample is found to be

$$\frac{P_{\bar{X}}}{\mu} \tag{A.6}$$

However, it is also of interest to find the precision limit and uncertainty of a single test. These are calculated from the two following relations.

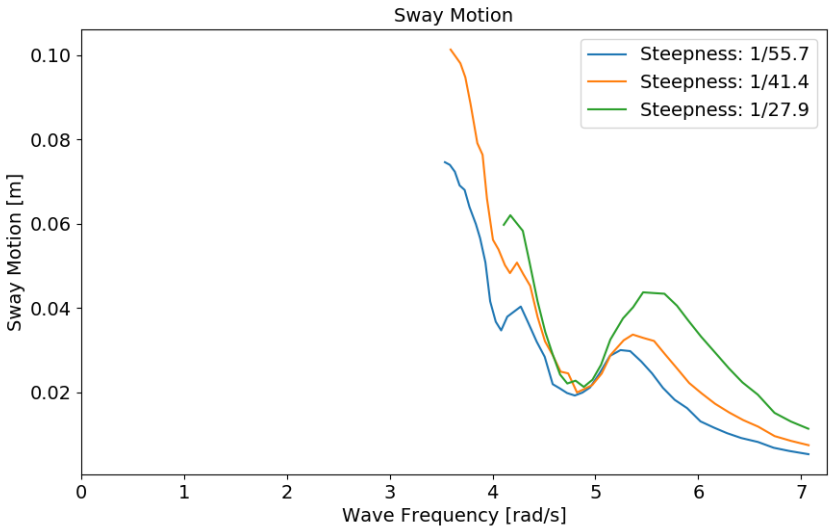
$$P_X = tS_X \ , \quad \frac{P_X}{\mu} \tag{A.7}$$

Appendix B

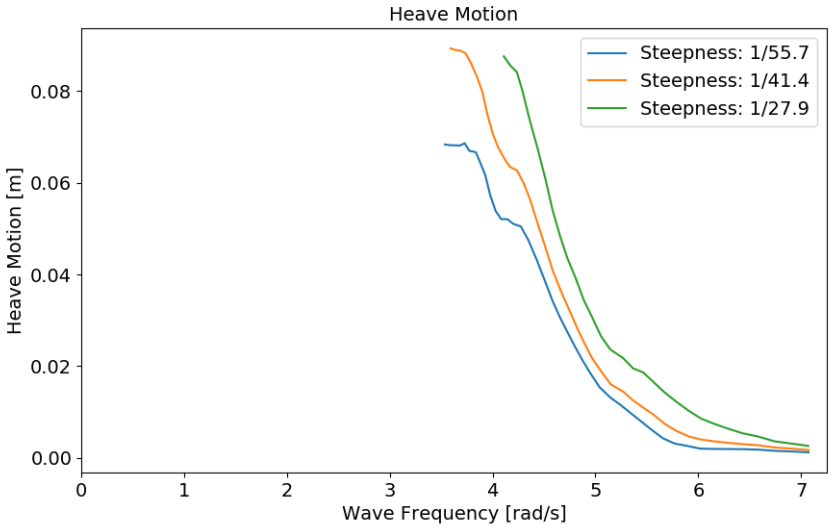
Experimental Results

The measured motions are plotted against wave frequency for three wave steepnesses.

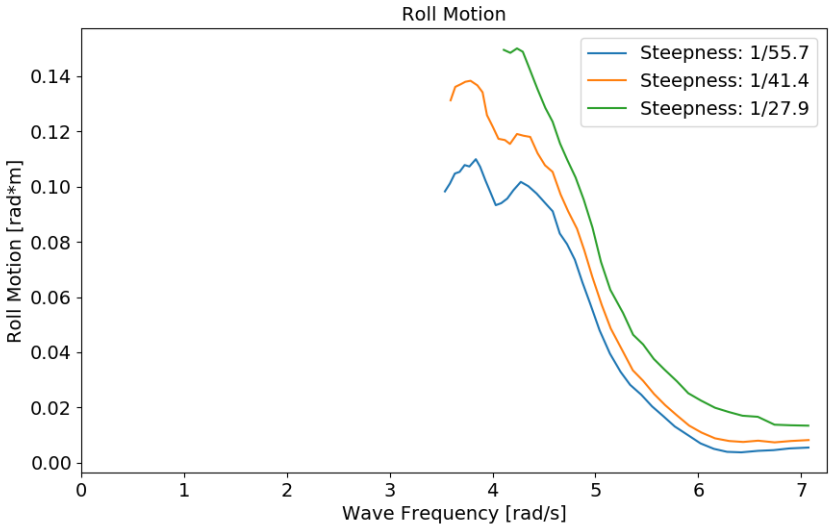
B.1 Sway Motion



B.2 Heave Motion



B.3 Roll Motion



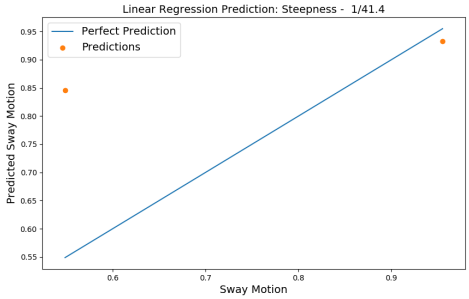
Appendix C

Linear Regression Results

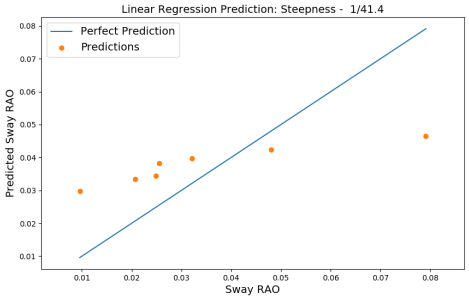
In Section C.1 the results from the linear regression model for wave steepness 1/41.4 are plotted for all motions and RAOs.

In Section C.2 the results from the linear regression model for wave steepness 1/27.9 are plotted for all motions and RAOs.

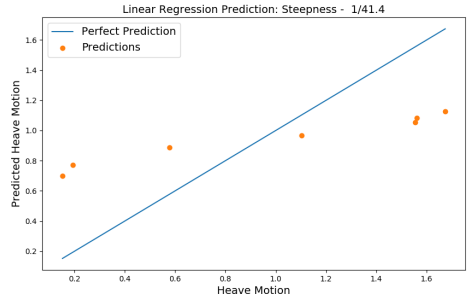
C.1 Linear Regression: Wave Steepness 1/41.4



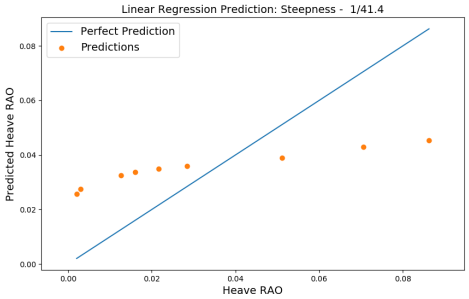
(a) Sway Motion



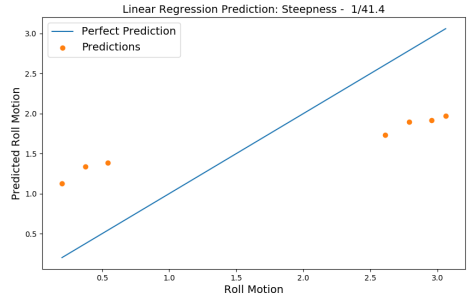
(b) Sway RAO



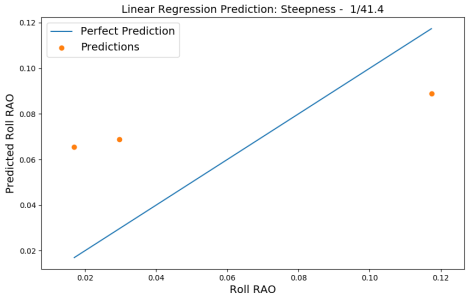
(c) Heave Motion



(d) Heave RAO



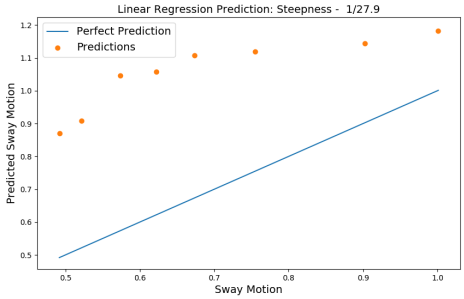
(e) Roll Motion



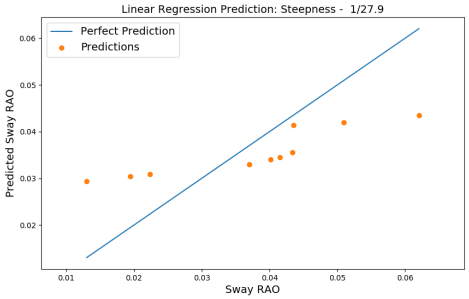
(f) Roll RAO

Figure C.1: Linear Regression Results for Steepness 1/41.4

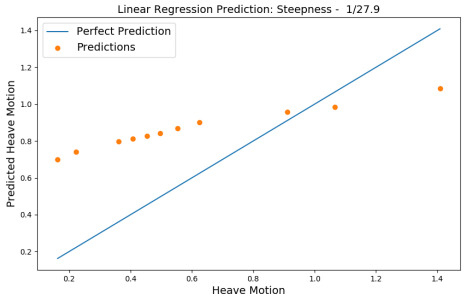
C.2 Linear Regression: Wave Steepness 1/27.9



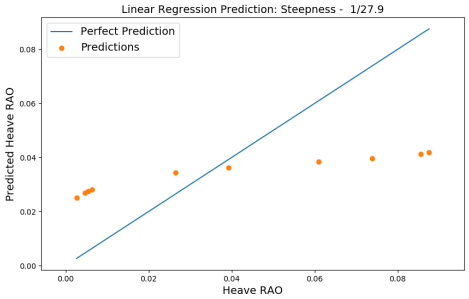
(a) Sway Motion



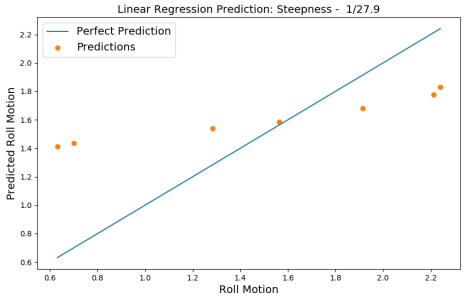
(b) Sway RAO



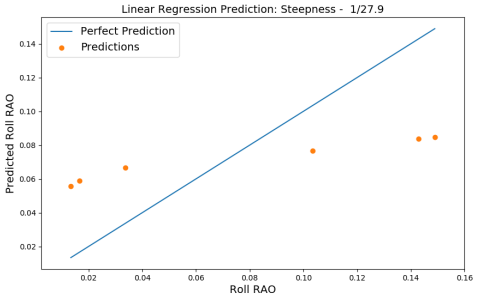
(c) Heave Motion



(d) Heave RAO



(e) Roll Motion



(f) Roll RAO

Figure C.2: Linear Regression Results for Steepness 1/27.9

Appendix D

Neural Network

Section D.1 plots the Neural Network prediction in Sway Motion and RAO.

Section D.2 plots the Neural Network prediction in Heave Motion and RAO.

Section D.3 plots the Neural Network prediction in Roll Motion and RAO.

Figure D.1 plots the results for wave steepness 1/55.7.

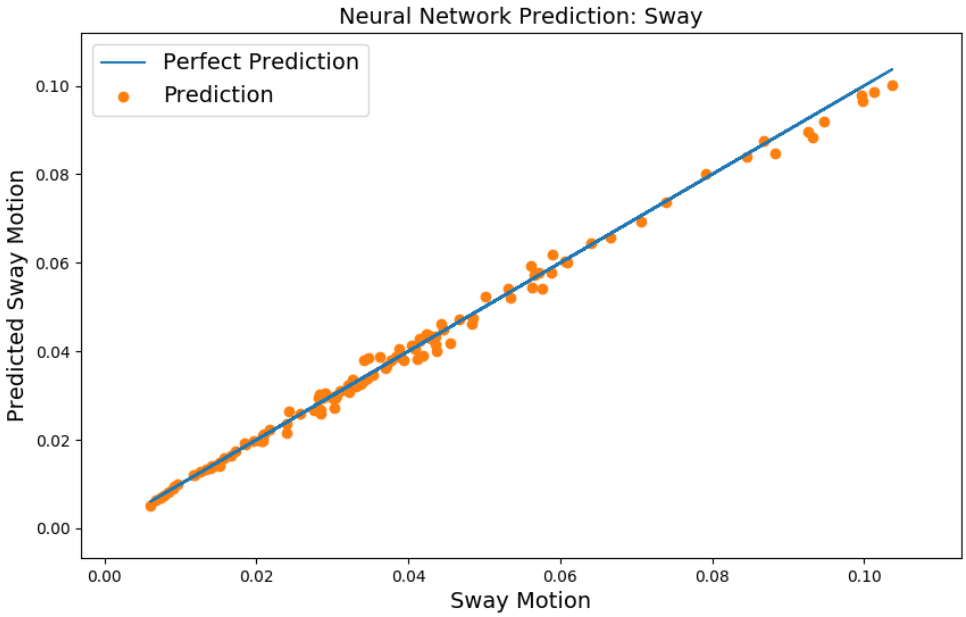
Figure D.2 plots the results for wave steepness 1/41.4.

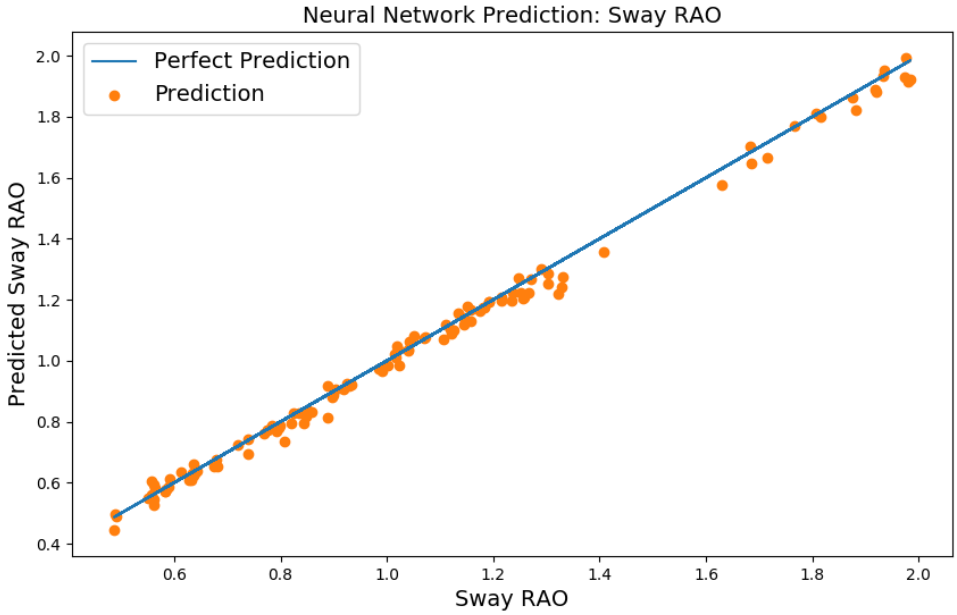
Figure D.3 plots the results for wave steepness 1/27.9.

Table D.1 shows the combinations of configurations tested for the Neural Network optimisation. The two columns to the right show the Coefficient of Determination and the Mean Absolute Percentage Error of the predictions made by the tested configuration.

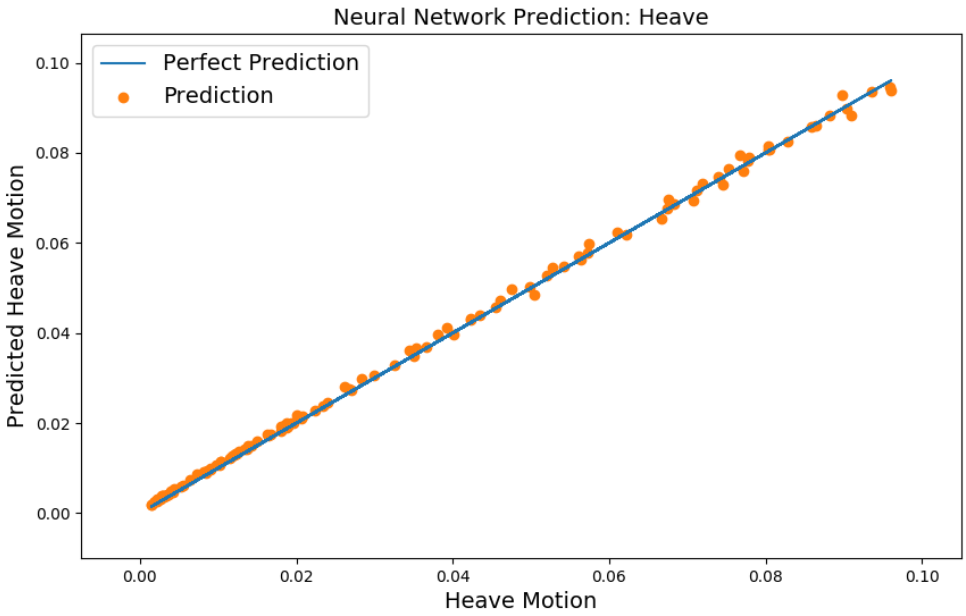
In Table D.2 the results for ten runs for each of the motions and RAO models are presented and the average value of the errors, R^2 and $MAPE$.

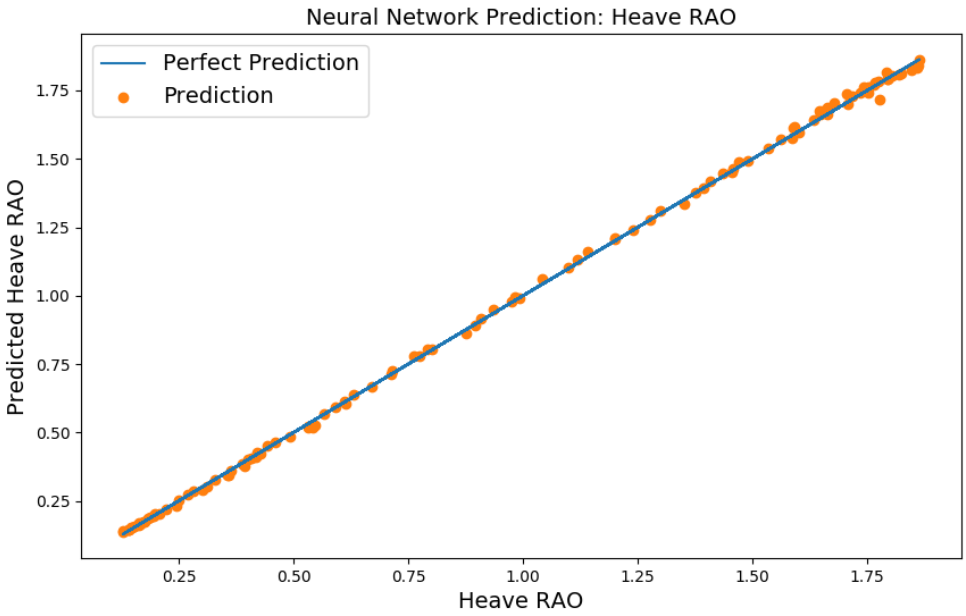
D.1 Neural Network: Sway Prediction



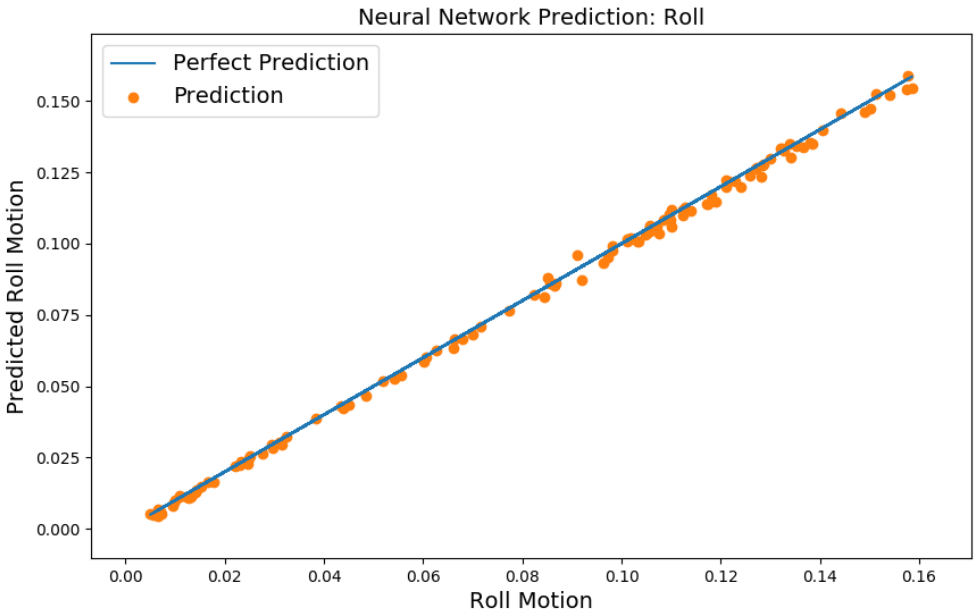


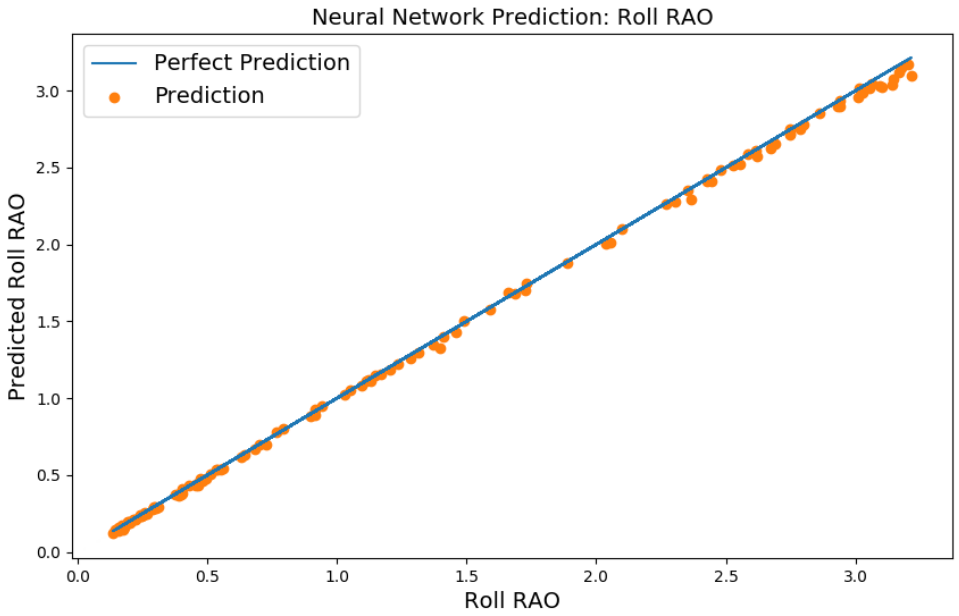
D.2 Neural Network: Heave Prediction



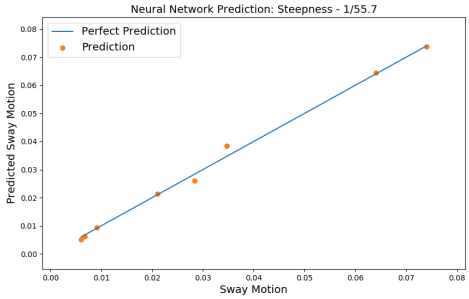


D.3 Neural Network: Roll Prediction

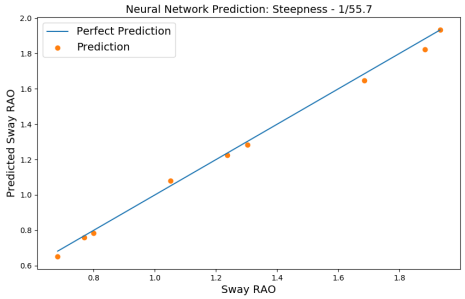




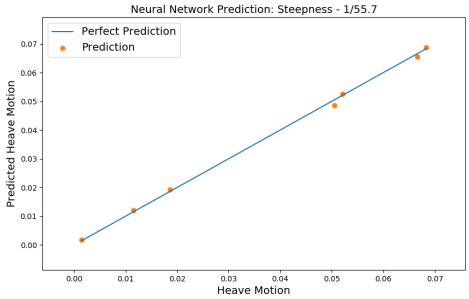
D.4 Neural Network: Wave Steepness 1/55.7



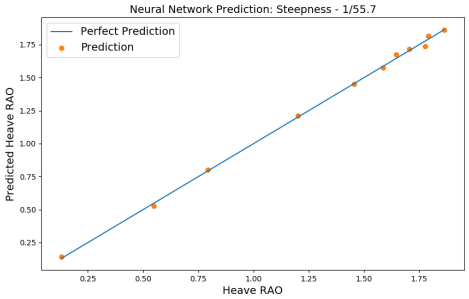
(a) Sway Motion Prediction



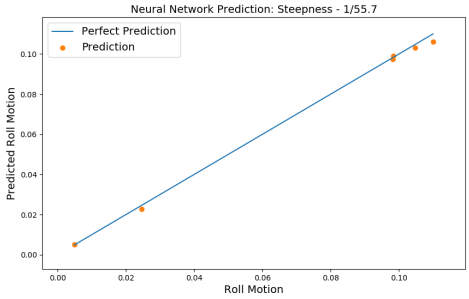
(b) RAO Sway Prediction



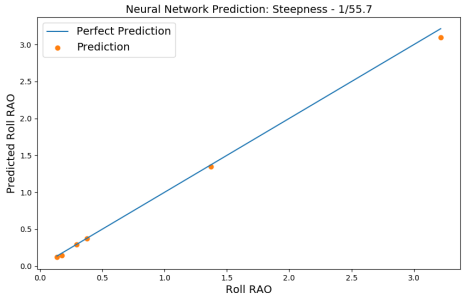
(c) Heave Motion Prediction



(d) RAO Heave Prediction



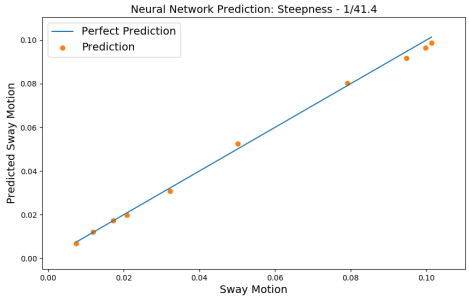
(e) Roll Motion Prediction



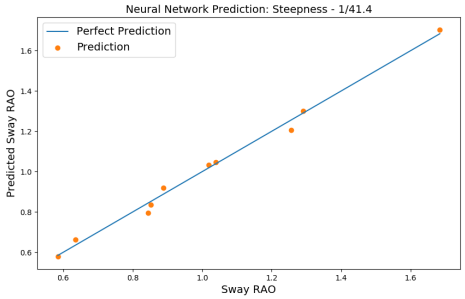
(f) RAO Roll Prediction

Figure D.1: Neural Network: Response Predictions for Wave Steepness 1/55.7

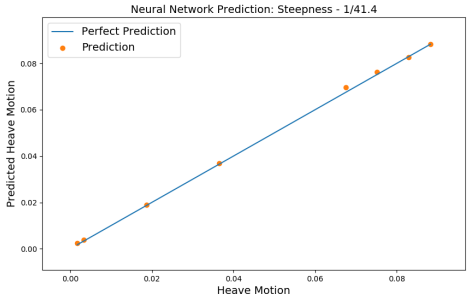
D.5 Neural Network: Wave Steepness 1/41.4



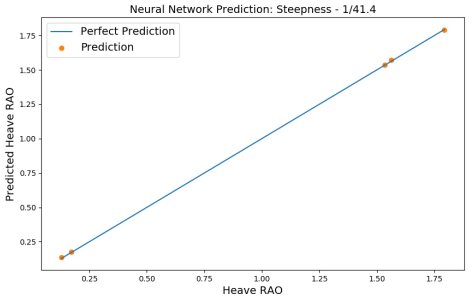
(a) Sway Motion Prediction



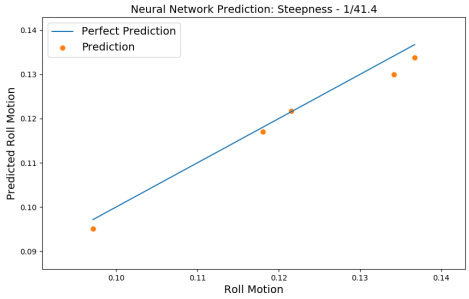
(b) RAO Sway Prediction



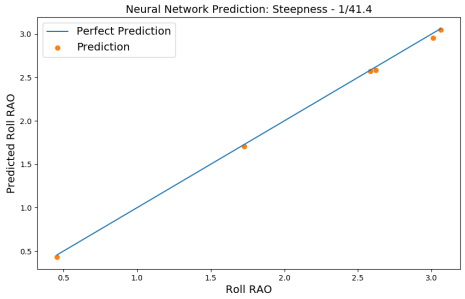
(c) Heave Motion Prediction



(d) RAO Heave Prediction



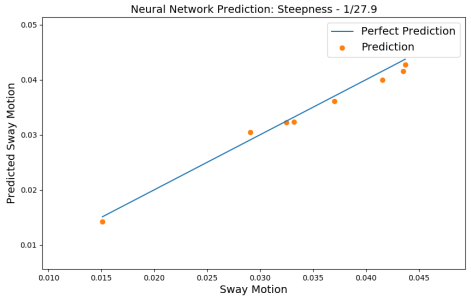
(e) Roll Motion Prediction



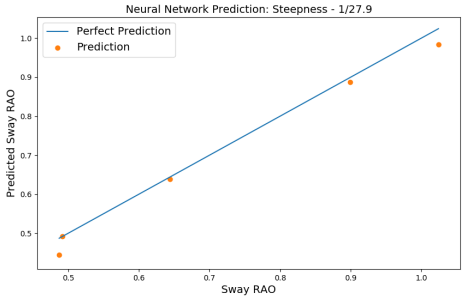
(f) RAO Roll Prediction

Figure D.2: Neural Network: Response Predictions for Wave Steepness 1/41.4

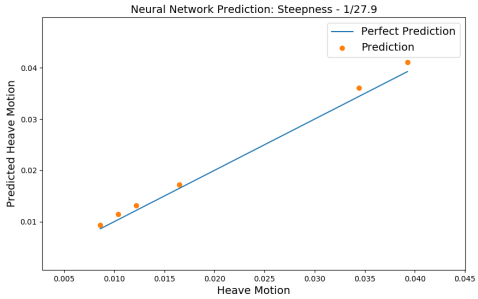
D.6 Neural Network: Wave Steepness 1/27.9



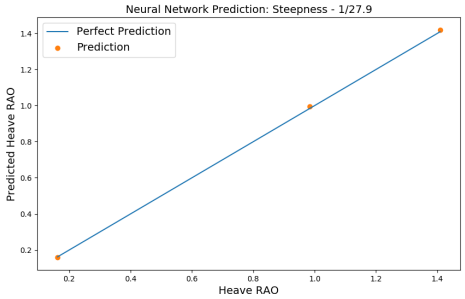
(a) Sway Motion Prediction



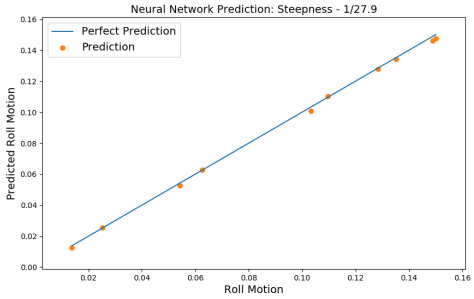
(b) RAO Sway Prediction



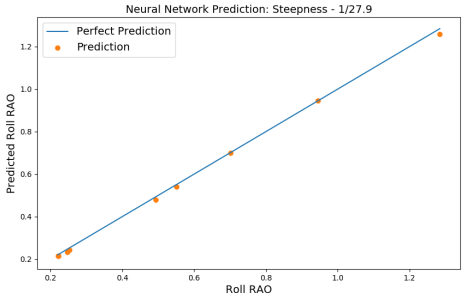
(c) Heave Motion Prediction



(d) RAO Heave Prediction



(e) Roll Motion Prediction



(f) RAO Roll Prediction

Figure D.3: Neural Network: Response Predictions for Wave Steepness 1/27.9

Table D.1: Test matrix of the Neural Network optimisation process with the different configurations

Test No.	Layer 1 No. Nodes	Layer 1 Activation	Layer 2 No. Nodes	Layer 2 Activation	Loss	Optimisation	No. Epochs	R^2	Mean Abs. % Error
1	2	ReLU	4	ReLU	MSE	RMSprop	1000	67.1 %	44.3 %
2	2	Sigmoid	4	Sigmoid	MSE	RMSprop	1000	75.9 %	25.5 %
3	2	Tanh	2	Tanh	MSE	RMSprop	1000	68.7 %	25.7 %
4	2	ReLU	4	Tanh	MSE	RMSprop	1000	86.8 %	19.6 %
5	12	Sigmoid	8	Sigmoid	MSE	RMSprop	1000	93.3 %	11.7 %
6	12	ReLU	8	Tanh	MSE	RMSprop	1000	96.7 %	11.4 %
7	24	ReLU	12	Tanh	MSE	RMSprop	1000	97.5 %	5.4 %
8	32	ReLU	24	Tanh	MSE	RMSprop	1000	98.1 %	4.8 %
9	32	ReLU	24	Tanh	MSE	RMSprop	5000	99.2 %	4.0 %
10	32	ReLU	24	Tanh	MSE	sgd	1000	90.0 %	15.3 %
11	32	ReLU	24	Tanh	MSE	RMSprop	1200	98.0 %	5.6 %

Table D.2: Neural Network prediction results for ten training sessions together with the average result for each response.

	Sway		Heave		Roll		RAO 2		RAO 3		RAO 4	
Test No.	**R^2**	**MAPE**	**R^2**	**MAPE**	**R^2**	**MAPE**	**R^2**	**MAPE**	**R^2**	**MAPE**	**R^2**	**MAPE**
1	99.3 %	4.4 %	99.8 %	7.1 %	99.1 %	3.1 %	97.5 %	3.5 %	99.9 %	2.9 %	99.9 %	2.6 %
2	98.3 %	6.6 %	99.4 %	12.1 %	98.3 %	13.6 %	98.9 %	2.3 %	99.9 %	1.4 %	99.9 %	2.4 %
3	98.1 %	5.1 %	99.0 %	13.3 %	99.7 %	3.2 %	99.3 %	2.2 %	99.9 %	2.2 %	99.9 %	3.3 %
4	97.7 %	4.7 %	99.6 %	7.5 %	99.2 %	9.6 %	99.0 %	2.2 %	99.9 %	2.6 %	99.9 %	3.4 %
5	98.7 %	4.1 %	99.8 %	8.3 %	99.9 %	2.9 %	99.2 %	2.4 %	99.9 %	2.3 %	99.9 %	3.5 %
6	95.6 %	7.0 %	99.9 %	6.5 %	99.9 %	3.0 %	99.3 %	2.2 %	99.9 %	3.9 %	99.9 %	2.3 %
7	97.7 %	4.3 %	99.8 %	10.5 %	99.7 %	5.9 %	99.2 %	2.2 %	99.9 %	2.5 %	99.9 %	3.8 %
8	99.1 %	4.5 %	99.5 %	11.0 %	99.8 %	6.3 %	99.0 %	2.5 %	99.9 %	2.6 %	99.9 %	2.0 %
9	98.0 %	5.6 %	99.8 %	4.1 %	99.8 %	7.1 %	99.4 %	2.2 %	99.8 %	4.9 %	99.9 %	2.7 %
10	97.7 %	10.0 %	99.7 %	4.3 %	99.3 %	7.2 %	98.9 %	2.3 %	99.9 %	1.8 %	99.9 %	1.8 %
Average	98.0 %	5.7 %	99.6 %	8.5 %	99.5 %	6.2 %	99.0 %	2.4 %	99.9 %	2.7 %	99.9 %	2.8 %