Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Harald Mo
Marie Fejerskov Kongshaug

Real-time compensation of residual loads in dynamic positioning control systems

Master's thesis in Marine Technology
Supervisor: Roger Skjetne
Co-supervisor: Mathias Marley
June 2023

NTNU
Norwegian University of
Science and Technology

Harald Mo
Marie Fejerskov Kongshaug

Real-time compensation of residual loads in dynamic positioning control systems

Master's thesis in Marine Technology
Supervisor: Roger Skjetne
Co-supervisor: Mathias Marley
June 2023

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

NTNU
Norwegian University of
Science and Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER OF TECHNOLOGY THESIS DEFINITION (30 SP)

Name of the candidate:	Marie Kongshaug and Harald Mo
Field of study:	Marine cybernetics
Thesis title (Norwegian):	Sanntidskompensasjon av residualkrefter i kontrollsystemer for dynamisk posisjonering
Thesis title (English):	Real-time compensation of residual loads in dynamic positioning control systems

Background

Real-time monitoring and compensation of prevailing environmental loads is essential in dynamic positioning (DP) based marine operations. Enabling better (machine) awareness of the wave field and predicting the wave parameters (frequency, amplitude, phase, direction, and ultimately the wave loads) at the vessel location in the time domain may aid the onboard decision making and the ability of the DP control system to compensate the wave loads. This is done conventionally through feedback control, where low-frequency 2nd order wave loads are compensated through the inherent integral action mechanism of the DP controller. For wind loads, on the other hand, the DP controller uses a wind feedforward mechanism. Since feedforward acts faster than feedback, this ensures a more reactive DP response to rapidly varying wind gusts and loads. With baseline being integral action in the PID controller or bias estimation in the DP observer – the two most common methods in state-of-the-art DP controllers, we will in this project look further into use of 1) adaptive control (as a nonlinear and more accurate integral action), 2) internal model principle by improved DP observer design, and 3) machine learning techniques for compensation of wave loads with low-frequency (not necessarily zero-frequency) components.

Scope of Work

1. Perform a background and literature review to provide information and relevant references on:
   - Hydrodynamic wave models and loads on ships, included but not limited to the quadratic transfer functions (QTFs), Newman's approximation, and different wave spectra.
   - DP control system: Objective, topology, feedback/feedforward control, observer, thrust allocation and control, conventional DP control and disturbance rejection techniques such as observer disturbance estimation, and adaptive control.
   - Machine learning (ML): Supervised vs. unsupervised learning, neural networks (NN), collection and quality of data, training of ML network.
   - Disturbance rejection based on NN and Fourier Series (FS).
   - Derivative-free optimization (DFO) techniques used in tuning of observers and controllers.
   - Simulation-based testing and verification; X-in-the-loop testing. Assurance through verification. Simulation challenges and assurance of simulations.
   - MC-lab test platforms. Experimental setups, ROS, Raspberry Pi, CSAD, etc.

   Write a list with abbreviations and definitions of terms and symbols, relevant to the literature study and project report.
2. Utilize the frequency analysis from the preceding project thesis, both of simulated and experimental DP loads and responses, in a discussion of assumptions and limitations, and elaborate on how well such loads are captured by conventional design models.
3. Further develop the fidelity of the established 6DOF CSAD simulation model (together with Jan-Erik Hygen), while focusing on run-time and making the code as universal and accessible as possible, incl. the new features:
   - Shallow water effect in the dispersion relation.
   - Multi-directional waves.
   - Improvement of frequency-dependent hydrodynamic coefficients.
   - Alternative integration schemes, such as Runge-Kutta methods.

   State all assumptions and limitations of the models and present the simulation modules visually.

4. Expand the already established 6DOF CSAD simulation model into a complete DP system. This includes the new implementation of:
    - DP observer – discrete Kalman Filter (LTV-KF or EKF).
    - Baseline DP controller.
    - Thruster dynamics, thruster control, and thrust allocation.
    - DP Guidance Manager (guidance model).
5. Implement and test the FS and NN as alternative internal models for the wave disturbance, incl.:
    - Develop, train, and implement an NN disturbance rejection mechanism. Discuss the robustness and commercial feasibility of the rejection mechanism. Investigate other possibilities of using ML and NN in disturbance rejection.
    - Implement an adaptive disturbance rejection mechanism using a truncated FS based on frequency analysis, both of simulated data and experimental response.
    - Discuss observer-based disturbance rejection based on the internal model principle as an alternative method.
6. Use DFO to automate tuning of the implemented DP controllers. Test with different initial conditions for the optimization parameters and discuss achieved performance of the observer.
7. Ultimately, make the code adaptable to ROS and the MC-lab. Perform lab experiments with the most promising disturbance rejection techniques and compare with simulation results.

Specifications

Every weekend throughout the project period, the candidate shall send a status email to the supervisor and co-advisors, providing two brief bulleted lists: 1) work done recent week, and 2) work planned to be done next week.

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the steps in the deduction.

The report shall be organized in a logical structure to give a clear exposition of background, problem/research statement, design/method, analysis, and results. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is expected to be between 100-140 A4-pages, from introduction to conclusion, unless otherwise agreed. It shall be written in English (preferably US) and contain the elements: Title page, project definition, preface (incl. description of help, resources, and internal and external factors that have affected the project process), acknowledgement, abstract, list of symbols and acronyms, table of contents, introduction (project background/motivation, objectives, scope and delimitations, and contributions), technical background and literature review, problem formulation or research question(s), method/design/development, results and analysis, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The contribution of the candidate shall be clearly and explicitly described, and material taken from other sources shall be clearly identified. Chapters/sections written together with other students shall be explicitly stated at the start of the chapter/section. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. natbib Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct, which is taken very seriously by the university and will result in consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed.

The thesis shall be submitted with an electronic copy to the main supervisor and department according to NTNU administrative procedures. The final revised version of this thesis definition shall be included after the title page. Computer code, pictures, videos, data, etc., shall be available for NTNU, for education and research purposes, and be included electronically with the report.

Start date: 15 January, 2023 **Due date:** As specified by the administration.

Supervisor: Roger Skjetne
Co-advisor(s): Mathias Marley, Emir Cem Gezer.
Signatures:

Roger Skjetne

Digitally signed by rskjetne
Date: 2023.06.16 14:42:08 +02'00'

Abstract

This Master's thesis focuses on addressing the challenges associated with compensating for second-order, low-frequency wave load on DP vessels. The conventional approach of using the integral action feedback method has been found to be susceptible to transients caused by the low-frequency components of the wave loads. This may cause positional offsets of the vessel. Motivated by this challenge, the work presented in this thesis aims to explore alternative methods for real-time compensation of these residual loads.

A high-fidelity simulation platform, MCSimPython, has been developed, in collaboration with Jan-Erik Hygen. The platform provides a comprehensive descriptions of vessel kinetics and kinematics, external environmental loads, and DP functionality, providing the complete functionality for end-to-end testing of DP control system. The simulation platform is provided as a open-source PyPI Python package. Additionally, a neural network source code has also been developed, and is utilized in the control techniques.

Four different disturbance rejection methods for compensating second order, low-frequency wave loads on DP vessels has been presented and successfully implemented. The methods include direct integral action in a PID controller, DP-observer estimates utilized in a feedforward loop, an adaptive controller employing a truncated Fourier series disturbance model and a neural network bias estimate integrated into a feedforward loop. The two first methods serve as baseline approaches, while the latter two are proposed as alternative methods for potentially better compensation of residual loads. All controllers have been tuned equally, by the means of derivative-free-optimization techniques, enabling a fairer comparison foundation.

The simulation study proved that all control methods exhibited a satisfactory performance. The adaptive controller demonstrated potential for a more rapid bias rejection during transient phases, although slightly underperforming in overall performance. The neural network bias model exhibited a superiority over the other approaches in regard to bias estimation. That being said, as the controller was implemented in a direct bias compensation mechanism, relying on a perfect vessel model, it was concluded that the technique faced difficulties regarding commercial feasibility.

A sensitivity analysis is provided to assess the robustness of the two newly proposed control methods. The analysis revealed sensitive behavior related to noisy velocity estimates, which had to be addressed in the tuning process of the adaptive gain, and introduced a trade-off between rapid bias rejection and stability. The neural network controller showcased notable variations in bias estimation when subjected to errors in the control design model and input parameters. This highlights the need for extensive training for the controller can be effectively implemented in a real system.

Finally, the PID and the adaptive controller, were tested experimentally in a laboratory. Despite the need for extensive re-tuning of the controllers, the experimental tests exhibited similar behaviors and trends as observed in the simulation study. This alignment serves as an important validation of the results obtained from the simulation study.

Sammendrag

Denne masteroppgaven fokuserer på å håndtere utfordringene forbundet med kompensasjon av annenordens, lavfrekvente bølgebelastninger på DP-fartøy. Konvensjonelt brukes integralvirkning i en tilbakekoblingssløyfe for å kompensere for slike krefter. Denne metoden har derimot vist seg å være sensitiv for transiente effekter forårsaket av de lavfrekvente komponentene i bølgebelastningene. Dette kan resultere i posisjonelle avvik for fartøyet. Motivert av denne utfordringen tar arbeidet i denne oppgaven sikte på å utforske alternative metoder for sanntidskompensasjon av disse gjenværende belastningene.

En omfattende simuleringsplattform, MCSimPython, har blitt utviklet i samarbeid med Jan-Erik Hygen. Plattformen inkluderer funksjonalitet for å modellere et fartøys kinematikk og kinetikk, ytre miljøkrefter samt funksjonalitet knyttet til reguleringsteknikk. Med dette fasiliteter plattformen for å kunne kjøre omfattende testing og analyse av DP-systemer. Simuleringsplattformen er tilgjengelig for allmenheten da den kan installeres som en egen Python pakke ved bruk av PyPI.

Fire ulike metoder for kompensasjon av annenordens, lavfrekvente bølgebelastninger på DP-fartøy presenteres og er blitt implementert. Metodene inkluderer direkte integralvirkning i en PID-regulator, bias estimater fra observer brukt i en foroverkobling, en adaptiv regulator som benytter en Fourierserie bias-modell, og, bias estimering fra et nevralt nettverks integrert i en foroverkobling. De to første metodene er konvensjonelle metoder og fungerer som et sammenligningsgrunnlag for de to sistnevnte mer innovative metodene.

Det er gjennomført en simuleringsstudie som evaluerer og sammenligner de fire ulike metodene presentert ovenfor. Hvor gode regulatorene er til å holde fartøyet i ro for en gitt posisjon, samt hvor godt de klarer å følge en ønsket manøver har blitt evaluert. For den sistnevnte testen er transienter tilstede. Dette øker kompleksiteten til testen. Generelt sett viste alle kontrollmetodene tilfredsstillende resultater for de to testene. Den adaptive regulatoren viste seg å ha de overordnede dårligste resultatene. Den viste derimot potensiale for rask kompensasjon av bias under transiente faser. Den nevrale nettverksregulatoren gav den høyeste posisjonsnøyaktigheten blant alle regulatorer. Samtidig estimerte den biasesen betraktelig bedre enn de andre metodene. Til tross for de lovende resultatene, kreves det videre utvikling før denne regulatoren kan anvendes i virkelige DP-systemer.

For å vurdere robustheten til de to alternative regulatorene ble det utført en sensitivitetsanalyse. Analysen avdekket sensitiv oppførsel knyttet til støyfulle hastighetssignaler for den adaptive regulatoren. Viktigheten av regulatorens stabilitet måtte derfor veies opp mot rask bias kompensering. Den nevrale nettverksregulatoren viste betydelige variasjoner i bias estimeringen når det ble introdusert feil i kontroll design modellen og nettverkets input parametere. Dette understreker behovet for omfattende trening før regulatoren kan implementeres effektivt i et ekte system.

To av kontrollmetodene, PID-regulatoren og den adaptive regulatoren, ble testet i Marin Teknisk Senters eget laboratorium. Til tross for at det var behov for omfattende justering av regulatorene, viste de eksperimentelle testene lignende oppførsel og trender som testene utført på simuleringsplattformen. Disse funnene fungerer som en validering av resultatene fra simuleringsstudien.

Preface

This thesis represents the culmination of our dedicated efforts and academic journey throughout five rewarding years at the Marine Technology master's degree program at NTNU in Trondheim. The work was started in the preceding project thesis during the fall of 2022, and it now concluded with the current master's thesis. The delivery of this thesis finalize our master's degree in marine cybernetics.

The knowledge acquired during these past months has significantly improved our understanding of the challenges currently faced by the industry regarding dynamic positioning systems. It has broadened our proficiency in programming, control theory, artificial intelligence, system modeling, and model verification and validation. We hand in this Master's thesis with motivation and optimism, as we embark on our professional journeys within the maritime industry.

We hope that this thesis serves as a meaningful contribution to the field within marine control systems and inspires further research and exploration. It is our fervent hope that the findings presented here will contribute to the advancement of knowledge regarding the study of disturbance rejection and have practical implications in marine control systems.

Acknowledgments

We would like to seize this opportunity to express our deepest gratitude to all those who have generously shared their knowledge, expertise, and time, contributing significantly to the development of this Master's thesis. Their support has been invaluable throughout this journey.

First and foremost, we would like to extend our appreciation to our supervisor, Roger Skjetne, and our co-supervisor, PhD candidate, Mathias Marley. They have both expressed a genuine interest in our work, and their insightful feedback, constructive criticism, and guidance, have played a key role in shaping the direction and content of this thesis. Throughout the entire research process, both has welcomed us to ask questions and prioritized to put aside time to talk to us. We are truly grateful for the encouragement to explore research questions that remain unanswered.

PhD candidate, Markus Fossdal, and Senior Engineer, Robert Opland, deserves a special mention for their invaluable assistance in conducting experiments in the MC-lab. Given the authors' limited prior experience in conducting lab experiments, their help has been of utmost importance for the successful completion of this thesis. Fossdal's remarkable proficiency in both software and hardware is truly impressive, and we and we consider ourselves fortunate to have such an exceptional teacher. The fact that Opland knows the MC lab like the back of his hand has been immensely helpful whenever we have faced difficulties in the lab. The willingness of both Opland and Fossdal to assist us, even outside of office hours, shows remarkable dedication. For this we are truly grateful.

Furthermore, we would like to express our gratitude to Jan-Erik Hygen for a productive and rewarding collaboration in the development of a generic simulation platform. This collaboration stands as one of the major contributions of this project. We strongly believe that our engaging discussions and different approaches to problem-solving has undoubtedly resulted in a better result than what any of us could have achieved individually.

Finally, we would like to express our sincere gratitude to our fellow students who have made the past five years truly memorable. We would like to extend a special thanks to our friends with whom we have shared our offices, the people in C317 and C319. Your positivity and sense of humor have greatly contributed to making this challenging journey more enjoyable.

To all those mentioned above and to those who have supported us in ways beyond words, we express our deepest gratitude. Your contributions have been instrumental in the completion of this master thesis, and we are truly honored to have had the opportunity to work with such remarkable individuals.
Thank you.

Marie Fejerskov Kongshaug and Harald Mo
June 2023

Contents

List of Figures

List of Tables

Abbreviations

CDM Control design model

CLF Control Lyapunov function

CSAD Cybership Artic Drillship

DFO Derivative-free optimization

DFT Discrete Fourier transform

DOF Degree of freedom

DP Dynamic positioning

EKF Extended Kalman filter

FFT Fast Fourier transform

GNC Guidance, navigation & control

IMO International Maritime Organization

HF High-frequency

HIL Hardware-in-the-loop

IMU Inertial measurement unit

ITTC International Towing Tank Conference

ISSC International Ship and Offshore Structures Congress

JONSWAP Joint North Sea Wave Project

LF Low frequency

LTV-KF Linear time-varying Kalman filter

MC Marine cybernetics

MIMO Multiple-input and multiple-output

MIL Model-in-the-loop

ML Machine learning

MPM Modified Pierson-Moskowitz

NED North-East-Down

NLO Nonlinear observer

NN Neural network

OOP Object-oriented programming

PM Pierson-Moskowitz

QTF Quadratic transfer function

RAO Response amplitude operator

RBF Radial basis function

ReLU Rectified linear unit

ROS Robotic operative system

SIL Software-in-the-loop

SV Slowly-varying

WF Wave frequency

Nomenclature

β	Wave direction
ϵ	Wave phase
η	Generalized position vector
γ	Peak enhancement factor
λ	Wave length
ν	Generalized velocity vector
ω	Wave frequency
ρ	Water density
τ	Generalized force vector
ζ	Free surface elevation
ζ_a	Wave amplitude
$\{b\}$	Body-fixed reference frame
$\{n\}$	North-East-Down reference frame
C_g	Group velocity
C_w	Phase velocity
g	Gravitational acceleration
h	Water depth
j	Imaginary unit
k	Wave number
w	Gaussian zero-mean white noise
Hs	Significant wave height
T	Wave period
T_1	Average wave period
Tp	Peak wave period
Tz	Average zero-crossing period

Chapter 1

Introduction

1.1 Motivation

Norway, stands, with the world's second longest coastline, as one of the leading offshore nations globally. With over 150 years of maritime experience, Norway is currently controlling the world's 4th largest merchant fleet (International Trade Administration, 2022). As emerging industries like offshore wind and aquaculture gain momentum, considerable growth in the international maritime industry is expected, and Norway is expected to play a central part in this transition. In a world facing both climate change and energy scarcity, advancements in the maritime industry hold immense importance, not only for the economic growth and welfare of the maritime nations such as Norway, but for the entire world.

The maritime engineering systems of today are already operating in highly challenging environments. As marine operations move further away from shore, these environments becomes even harsher, further increasing the demands with regards to safety and reliability. Dynamic positioning (DP) systems are present in most complex maritime operations today and it is, therefore, important to dedicate time and effort to make them as accurate and redundant as possible. DP systems are designed to compensate for environmental loads such as waves, currents, and wind. Currently, wind loads are compensated for with a feedforward control, while the slow variations exhibited by current and second order wave loads are commonly compensated for using integral action in a feedback loop. However, this compensation is susceptible to transients, originating from low-frequency contributions, which can manifest as positional offsets in the vessel's position. For marine operations that require high positional precision, such drift can be unacceptable. Increasing accuracy in the DP system can significantly strengthen the operational safety, enabling operations in harsher conditions than currently operated in, and expending the weather window, ultimately reducing costs. Addressing this challenge has been the driving force behind the work presented in this thesis, where efforts has been made to explore alternative methods for low-frequency wave load compensation.

1.2 Project formulation and objectives

This thesis focus on exploring the challenges related to the compensation of second order, low-frequency wave loads on DP vessels. It is possible that superior alternative disturbance rejection methods, surpassing the baseline methods, exist, and have the potential to enhance the overall performance of DP systems. This thesis aims to investigate this statement by formulating the following main objective:

- Explore the effectiveness of alternative compensation techniques, in comparison to traditional integral action, for mitigating response from low-frequency (not necessarily zero-frequency) wave load components.

In addition, a sub-objective has been established to support the achievement of the main objective:

- Develop a high fidelity simulation platform to enable full end-to-end testing of control systems in an isolated environment.

To accomplish these objectives, a scope was defined in collaboration with our supervisor, Roger Skjetne. The scope is presented in Section 1.4.

1.3 Literature review

The present study builds upon a solid foundation of existing research and theoretical frameworks in the field of dynamic positioning. The project objective is based on the preceding master's thesis written by Brørby (2022). This thesis have been used as a notable reference point, and provided valuable insight for further development in the current study.

The doctoral thesis of Værnø (2020) has played a fundamental role in this master's thesis. Specifically, the research conducted and published in the journal papers within the thesis, such as Værnø, Skjetne et al. (2019), and Værnø, Brodtkorb et al. (2019). Especially the latter, which investigates the effects of various internal bias models for compensating residual loads in a DP system, has been influential, and is frequently referenced to in the relevant scientific literature.

A large part of the theory in this thesis is collected from the text books written by Faltinsen (1990), Fossen (2021), and Nocedal and Wright (2006). These publications present in-depth descriptions of wave load theory, marine control systems, and optimization, and are considered well known within their respective field of works.

1.4 Scope and delimitations

The scope of this thesis was outlined to fulfill the objectives presented in Section 1.2. It can be summarized in the following bullet points.

- Perform an extensive background study and literature review on relevant topics to provide information and references essential for the completion of this thesis.
- Further advance the already established Python simulation platform. A simulation model, capable of simulating first and second order wave loads for various vessel models, is to be created. Focus on making the source code computational efficient.
- Extend the established simulation platform to a full DP-model by implementing guidance, navigation and control (GNC) functionality.
- Implement four different control techniques, whereas two serve as baseline controllers, commonly used in industrial DP systems of today. The remaining two should be considered as less explored methods. Evaluate their performance with regards to compensation of slowly varying wave loads.
- Utilize derivative free optimization (DFO) to tune the implemented controllers, creating an increased basis for comparisons in between.
- Ensure seamless communication between the simulation platform and hardware in the marine cybernetics laboratory (MC-lab) by implementing a Robot Operating System (ROS) wrapper.

To set realistic expectations for what can be achieved from the research related to the thesis, delimitations are presented as follow:

- The scope of the thesis mainly involve compensation of wave and current loads. Disturbances due to wind is, therefore, excluded in all test cases.
- Only a general thrust allocation algorithm has been implemented in the developed simulation platform. Constraints, such as forbidden zones and angular rate, has not been addressed.
- The created simulation model fully relies on hydrodynamic coefficients and response amplitude operators provided by ShipX (Veres) by Sintef Ocean. These functions are numerically computed from 2D strip theory, and the fidelity of the simulation platform is, thus, limited by the fidelity of the Veres data.
- The availability to the MC-lab was limited due to the fact that the lab was shared with several other Master's students. This resulted in limited time for the experimental testing.
- The model vessel used for experimental testing had malfunctioning on several of the thrusters, strongly reducing the maneuverability of the vessel.
- Due to complexity in generating current in the MC-lab, the experimental tests were conducted without the presence of current loads.

1.5 Project contributions

The contributions from this project thesis are focused on facilitating for further work related to the development of disturbance rejection methods and DP studies. Extensive work has been conducted, and the main contributions can be listed as:

- **Contributions to Knowledge**: Advancing knowledge about alternative methods for residual load compensation in DP systems.
- **Development of Simulation Platform**: In collaboration with fellow M.Sc. student Jan-Erik Hygen, a high-fidelity simulation platform has been developed. This platform is distributed as an open-source Python package, making it accessible to researchers and other students. The simulation platform serves as a valuable tool for testing and evaluating DP systems.
- **Validation of Simulation Platform**: The simulation platform has been validated through comparisons with experimental testing and theoretical analysis. This validation process ensures the accuracy and reliability of the platform, enhancing its usefulness for future studies.
- **Development of four distinct control methods** Four distinct control methods have been implemented, all with their separate bias rejection technique. Their performances are studied through an in-depth analysis, and two of the controllers are tested experimentally in the MC lab.
- **Neural Network Architecture**: A neural network architecture has been developed specifically for bias prediction. The architecture provides an innovative approach to estimate and compensate for bias loads in DP systems.
- **Comparison Study**: The project thesis includes a comprehensive comparison study that evaluates and compares four different disturbance rejection methods. This study offers insights into the strengths and weaknesses of these methods, hopefully facilitating for further research on the approaches presented.

1.6 Thesis outline

The thesis is structured into seven separate chapters. The setup aims to first provide the reader with a comprehensive background of the relevant theory, before presenting the developed simulation model, as well as all compensation methods. Furthermore, all simulation results are presented and discussed,

before, finally, some of the key findings are validated in an experimental study. The chapters are organized as followed.

Chapter 2 presents the relevant background theory used as a basis for all further work and results presented in this Master's thesis. The chapter address topics such as hydrodynamic wave theory, neural networks, DP systems, derivative-free optimization techniques, and the MC lab facilities.

Chapter 3 propose a high-fidelity, 6 degree of freedom (DOF), simulation platform developed for this master's thesis. The platform's modules, and all their functionalities, are displayed and critically discussed, before illustrating how the platform can be connected to hardware, using a ROS interface.

Chapter 4 unveil the research questions of the thesis, as well as the different control methods explored to solve them. The chapter also briefly presents the tuning processes of the controllers.

Chapter 5 presents final results from the extensive simulation study of the different disturbance rejection methods. Both quantitative and qualitative assessments of the controllers are made. The results also include a sensitivity analysis for a selected number of controller parameters.

Chapter 6 aims to investigate some of the key findings from the simulation study through experimental testing in the MC-lab. The findings are discussed, and evaluated against results in Chapter 5.

Chapter 7 summarize the key findings from the analyses presented in Chapter 5 and Chapter 6. The chapter set the findings up against the overall objectives of the thesis, and present potential further work to be done.

Chapter 2

Background

This section will present a comprehensive description of all theory relevant for the upcoming study. This include a thorough presentation of hydrodynamic wave loads, neural networks, DP systems, vessel modeling, derivative-free optimization, and the MC-lab setup. Certain parts of the presented theory are adapted from the preliminary project work in Kongshaug and Mo (2022). This is particularly the case for Section 2.2, Section 2.4, and Section 2.5.

2.1 Mathematical preliminaries and methods

2.1.1 Mathematical notation

The following chapter presents mathematical notations used in this report.

Hadamard product: The element-wise product of two matrices. The Hadamard product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted as

$$\boldsymbol{A} \circ \boldsymbol{B}. \tag{2.1}$$

Euclidean distance: The distance between two points in the n-dimensional Euclidean space,

$$d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}|| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}, \tag{2.2}$$

where $\boldsymbol{x} = [x_1, ..., x_n]$ and $\boldsymbol{y} = [y_1, ..., y_n]$.

Cross product operator: The cross product matrix operator is given by

$$\boldsymbol{\lambda} \times \boldsymbol{a} := \mathbf{S}(\boldsymbol{\lambda})\boldsymbol{a},$$

where $\boldsymbol{S}$ is given as the matrix

$$\mathbf{S}(\boldsymbol{\lambda}) = -\mathbf{S}^\top(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix}. \tag{2.3}$$

Skew-symmetric matrices: A skew-symmetric matrix is defined as

$$\boldsymbol{S} = -\boldsymbol{S}^\top, \tag{2.4}$$

implying that $s_{ij} = -s_{ji}$. A skew symmetric matrix also has the useful property for the quadratic form of a vector $\boldsymbol{\lambda}^T \mathbf{S}(\boldsymbol{\lambda})\boldsymbol{\lambda} \equiv 0$.

Big-$\mathcal{O}$ notation: The big-$\mathcal{O}$ notation describes the run time limit of a function f. The notation is often used in run time analysis within computer science, and denotes how the execution time of a script is based on the size of the input n. The limit is notated as $f = \mathcal{O}(\dots)$, where the run time is presented within the bracket, as a function of n.

2.1.2 Fourier transformations

Given a signal in the time-domain $f(t)$, the Fourier transformation will map this signal to the frequency-domain through the Fourier transformation operator $\mathcal{F}\big[\cdot\big]$. The operator is defined as in (2.5a), according to Kreyszig (2011). The output signal is then regarded as the signal spectrum, describing the intensity and presence of harmonic sinusoidal waves for different frequencies. This enables several signal processing opportunities. It can especially be of interest to examine the magnitude for different frequencies, which can indicate where the different frequencies originate from. The Fourier transform is a complex number, where the modulus corresponds to the amplitude, and the argument to the phase. Similarly to the Fourier transformation operator, the inverse Fourier transformation also exists, and is described in (2.5b).

$$\hat{f}(j\omega) = \mathcal{F}\Big[f(t)\Big] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{2.5a}$$

$$f(t) = \mathcal{F}^{-1}\left[\hat{f}(j\omega)\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(j\omega) e^{j\omega t} d\omega \tag{2.5b}$$

As computers operate in discrete time, the discrete Fourier transform (DFT), presented in (2.6), is used in software and electronic devices that generate frequency spectra. The method can be seen as a approximation of the continuous transformation, (2.5a). The validity of the DFT will depend on the time-domain signal being analyzed, as it is defined on a finite time interval for a computational signals. For efficiency the DFT is commonly implemented by the fast Fourier transform (FFT) (Brigham, 1988).

$$\hat{f}(j\omega) = \mathcal{F}\Big[f(t)\Big] = \sum_{k=0}^{N} f(t_k) e^{-j\omega t_k} \Delta t \tag{2.6}$$

There exist several different methods for spectral density estimation. One of those is Welch's method which partitions the signal into multiple periodograms and averages them to reduce noise in the estimated power spectra. Welch's method will introduce a trade-off between noise reduction and frequency resolution. This will be decided by the number of samples per periodogram and is important to bear in mind when performing spectral analysis (Solomon, 1991).

2.2 Hydrodynamic wave theory

2.2.1 Wave modeling

Wave modeling is often split into two subsections: regular and irregular wave theory.

Regular wave theory is a mathematical model that assumes that waves can be described by a constant wavelength, period, and amplitude. The theory is based on simplifying assumptions, including the disregard of wave breaking, viscosity, and turbulence effects. Regular waves can be mathematical described as the relationship between the height, length, period, and speed of the wave, as well as the water depth and gravitational acceleration. The first order solution of the wave elevation for a regular sinusoidal wave can be expressed as

$$\zeta(x,t) = \zeta_a \sin(\omega t \pm kx + \epsilon), \tag{2.7}$$

where the angular wave frequency is defined as $\omega = \frac{2\pi}{T}$, T is the wave period, k is the wave number, and ϵ is a random phase, uniformly distributed in the interval $[0, 2\pi]$. The plus/minus sign inside the trigonometric expression in (2.7) defines the direction of the wave propagation. In all upcoming calculations, the waves described will be defined to propagate in the positive x-direction, corresponding to using the minus sign. The relation between the wave frequency and wave number is given by the dispersion relation presented in (2.8a). g, defines the gravitational constant. By assuming infinite water depth ($h \to \infty$), the expression is simplified to the deep water dispersion relation, as in (2.8b) (Faltinsen, 1990, ch. 2).

$$\omega^2 = kg \tanh(kh) \tag{2.8a}$$

$$\omega^2 = kg \tag{2.8b}$$

Unlike regular waves, which can be described by a simple mathematical equation, irregular waves have a more complex nature, providing a more realistic representation of a real sea. Irregular wave theory models the sea surface as a linear superposition of regular wave components with various amplitude, phase angle and frequency. The properties of these component waves are provided by the wave spectrum, $S(\omega)$, a frequency-domain representation of the sea state. The spectrum provides information about the energy content and spatial distribution of the waves, as presented in (2.9) (Larsen et al., 2021, ch. 12).

$$\frac{E}{\rho g} = \sum_{i=1}^{N} \frac{1}{2} \zeta_{ai}^2(\omega_i) = \sum_{i=1}^{N} S(\omega_n) \Delta\omega \tag{2.9}$$

Irregular waves are often classified into long-crested waves and short-crested waves. Long-crested waves are modeled as two-dimensional, unidirectional waves, that vary in space and time. This model can be sufficient for waves generated by a storm where the waves travel with sufficiently large wave periods, typically seen for swell (Newman, 2017). The surface elevation of long crested waves are expressed as the sum of N distinct regular wave components, with a random phase angle, ϵ_i. The phase angle, uniformly distributed between 0 and 2π, represents the randomness and unpredictability of the wave field.

$$\zeta(x,t) = \sum_{i=1}^{N} \zeta_{ai} \cos(\omega_i t - k_i x + \epsilon_i) \tag{2.10}$$

It can be observed that the expression in (2.10) is a linear superposition of several representations of (2.7), each with their own energy contribution. By combining (2.9) and (2.10), the surface elevation can be expressed as

$$\zeta(x,t) = \sum_{i=1}^{N} \sqrt{2S(\omega_i)\Delta\omega} \cos(\omega_i t - k_i x + \epsilon_i). \tag{2.11}$$

The unidirectional wave assumption can be insufficient for describing a real sea surface, where waves often propagate from multiple directions. Especially for shorter and steeper waves, this directional variation is significant (Newman, 2017). Short-crested waves provides a more realistic representation of a real sea surface by superposing waves propagating from multiple directions. The surface elevation at a given position (x, y), at time t, can is then given as

$$\zeta(x,y,t) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{2S(\omega_i, \beta_j)\Delta\omega\Delta\beta} \sin(\omega_i t - k_i x \cos(\beta_j) - k_i y \sin(\beta_j) + \epsilon_{i,j}), \tag{2.12}$$

where the directional wave spectrum, $S(\omega, \beta)$ is introduced. β defines the direction of the wave propagation. N and M are the number of frequencies and directions, respectively. An increased number of

wave components will lead to increased computational time for simulations, and efforts can thus be made to reduce this computational time, for instance by choosing random frequencies and directions within each frequency- and direction-interval. Another measure to reduce computational time is to remove wave components that contain an insignificant amount of energy, as these components often have a negligible contribution to the total sea state (Fossen, 2021).

2.2.2 Wave spectra

As stated, the wave spectra portray a frequency-domain representation of the sea state. Several different wave spectra exist, each with their own characteristics. The Pierson-Moskowitz (PM) and the JONSWAP spectra are two commonly used, one-peaked spectra within ocean engineering.

The PM spectrum is a wave spectrum, developed in 1963, for fully developed wind-generated sea states in the North Atlantic Ocean. The spectrum is known to have a steep front at low frequencies, and is used as a basis for several spectral formulations. It can be expressed as

$$S(\omega) = \frac{A}{\omega^5}\exp\left(-\frac{B}{\omega^4}\right). \tag{2.13}$$

In the original formulation, the parameters A, and B, are given according to Fossen (2021), as

$$A = 8.1 \cdot 10^{-3} g^2, \qquad B = 0.74\Big(\frac{g}{V_{19.5}}\Big)^4.$$

The standard PM spectrum is defined by $V_{19.5}$, the wind speed 19.5 m above the sea level. However, ISSC and ITTC recommends a modified version for a deep-water fully developed sea with no swell and unlimited fetch. This modified PM spectrum (MPM) is a two-parameter wave spectrum, as opposed to the standard one-parameter PM spectrum, and is defined by the significant wave height, Hs, and the average zero-crossing period, Tz. Its parameters are then defined in Fossen (2021), as

$$A = \frac{4\pi^3 \mathrm{Hs}^2}{\mathrm{Tz}^4}, \qquad B = \frac{16\pi^3}{\mathrm{Tz}^4}.$$

The JONSWAP spectrum is an ITTC stadard spectrum, created to resemble non-fully developed sea in the North Sea. The spectrum was developed during the extensive measurement program known as the *Joint North Sea Wave Project*. Mathematically, it can be represented as a PM-like spectrum, scaled by a peak enhancement factor, denoted as γ. Hence, it is often recognized by its sharp peak compared to the PM spectrum. The spectrum can be expressed, according to Faltinsen (1990), as

$$S(\omega) = 155\frac{\mathrm{Hs}^2}{\mathrm{T}_1^4\omega^5}\exp\left(-\frac{944}{\mathrm{T}_1^4\omega^4}\right)\gamma^{\exp\left[-\left(\frac{0.191\omega\mathrm{T}_1-1}{\sqrt{2}\sigma}\right)^2\right]} \tag{2.14}$$

$$\sigma = \begin{cases}\sigma_a, & \text{for } \omega < \omega_p \\ \sigma_b, & \text{for } \omega > \omega_p\end{cases}, \tag{2.15}$$

where T_1 describes the average wave period, and ω_p represents the spectral peak frequency. Results from the JONSWAP project suggested $\gamma = 3.3$, $\sigma_a = 0.07$ and $\sigma_b = 0.09$ (Torsethaugen, 2004). Notice that when $\gamma = 1$, the JONSWAP spectrum equals the PM spectrum. This can be observed from Figure 2.1, as the blue and the orange lines coincide.

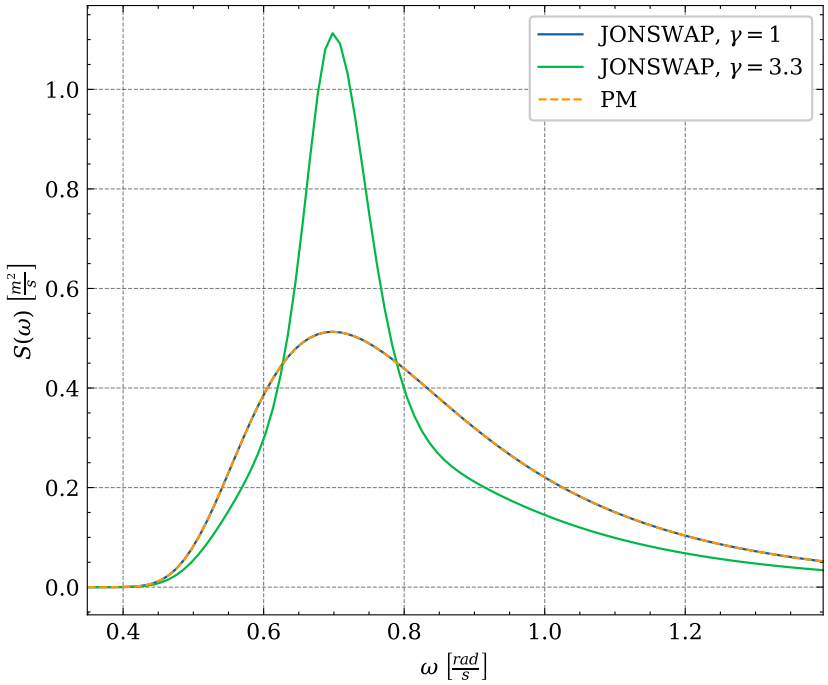


Figure 2.1: Comparison of different 1D wave spectra.

Although the two spectra presented both are considered as one-peaked spectra, the sea state can also be modeled by a two-peaked spectra, such as the Torsethaugen spectrum. The Torsethaugen spectrum is an empirical spectrum derived from fitting two JONSWAP-shaped spectra to a spectrum averaged over different combinations of Hs and Tp from the Norwegian Continental Shelf. The additional low-frequent peak introduced in the spectrum originates from swell (Torsethaugen, 2004).

Figure 2.1 only presents unidirectional spectra, capable of generating waves in one direction. One may, however, wish to model multidirectional wave fields for a more accurate representation of the sea state. This requires directional wave spectra. The energy of the wave spectrum is then spread out over different angles through a spreading function

$$D(\theta) = 2^{(2s-1)} s! \frac{(s-1)!}{\pi(2s-1)!} \cos\left(\theta - \theta_p\right)^{(2s)}, \tag{2.16}$$

such that

$$S(\omega, \theta) = S(\omega) D(\theta), \tag{2.17}$$

where D is here assumed independent of frequency. It is important to note that this assumption is a simplification and may not necessarily hold true in all cases (Larsen et al., 2021, ch. 13).

2.2.3 Wave loads

The wave loads exerted on a vessel can be modelled as a sum of first and second order wave loads. Higher-order terms can also be included to additionally increase the precision, even though theory up to second order often is considered sufficient.

First order wave loads, $\boldsymbol{\tau}^{(1\text{st})}$, operate in the same frequency interval as the incident wave frequencies. They will, thus, oscillate in parallel with the wave elevation, but with a different phase. As for the second order loads, $\boldsymbol{\tau}^{(2\text{nd})}$, it is common to split them up into three separate loads; sum-frequency, difference-frequency, and mean-drift loads. The sum- and difference-frequency loads occur at respectively higher

and lower frequency compared to the first order wave-frequency loads. The mean-drift loads do not oscillate and will therefore provide a constant force on an object in waves (Faltinsen, 1990). The total force acting on a body in waves is then modeled as

$$\boldsymbol{\tau}_{wave} = \boldsymbol{\tau}^{(1\text{st})} + \boldsymbol{\tau}^{(2\text{nd})}. \tag{2.18}$$

Figure 2.2 displays the partition of the first and second order wave loads, separated according to which frequency interval the loads operate within.

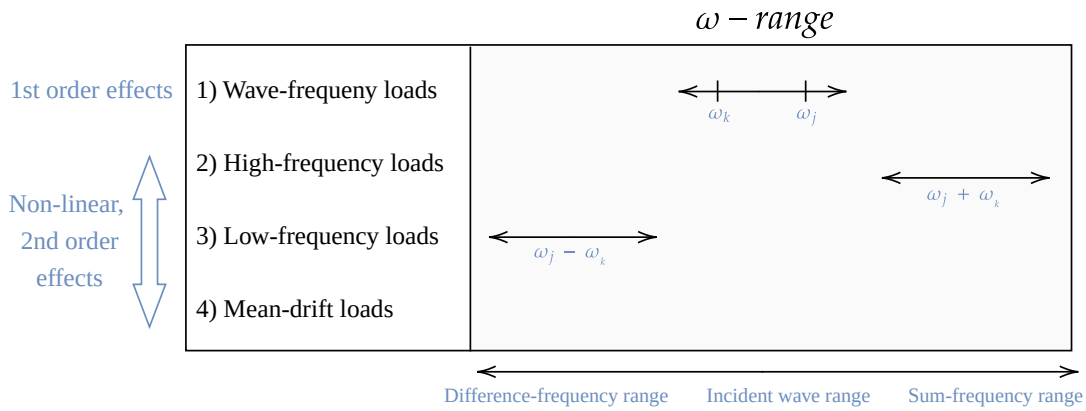


Figure 2.2: Wave loads divided according to frequency.

The first order wave loads, often denoted as the wave-frequency (WF) loads, usually represent the largest contribution of the wave loads. These loads can be represented as a sum of each individual wave's load contribution (OrcaFlex, 2022). Force *Response Amplitude Operators*, (RAOs), can be used to translate the incident waves to wave-induced loads. These force RAOs are first order transfer functions, dependent on ship geometry, and contains both an imaginary and a real part (amplitude and phase).

Given a force RAO

$$\text{RAO}_{force}(\omega, \beta) = |F_j(\omega_k, \beta_i)| + j\angle F_j(\omega_k, \beta_i), \tag{2.19}$$

the wave-induced, first order load, in DOF m, can be calculated as

$$\tau_m = \sum_{k=1}^{N} \sum_{i=1}^{M} \rho g \zeta_{a,k} \, |F_m(\omega_k, \beta_i)| \, \cos\Big(\omega t + \angle F_m(\omega_k, \beta_i) + \epsilon_k\Big), \tag{2.20}$$

where $F_m(\omega_k, \beta_i)$ is the force RAO's amplitude, and $\angle F_m(\omega_k, \beta_i)$ is the force RAO's phase (Fossen, 2021). j is defined as the imaginary unit.

Second order wave loads are included in wave load calculations to enhance the accuracy of the wave loads acting on a vessel. In second order wave theory, all terms in the velocity potential, proportional to the square of the wave amplitude, are taken into consideration and superposed with the linear terms, according to the perturbation method. This results in a smaller approximation error, described by the upper approximation error limit of $\mathcal{O}(\zeta^3)$ (Faltinsen, 1990).

The total second order wave loads are obtained by summing up the three components present in the second order model, as in (2.21). SV and HF do here refer to slowly-varying and high-frequency, respectively. The sum-frequency loads are often neglected as these, in general, are smaller in magnitude and yield little effect on large vessels. However, for some marine constructions, such as tension leg

platforms, the sum-frequency forces can be of great significance as they can excite resonant oscillations (Fossen, 2021). That being said, for DP-vessels, the slowly varying motions are the ones of the greatest interest, as these can cause the vessel to drift from its set point.

$$\boldsymbol{\tau}^{(2\text{nd})} = \bar{\boldsymbol{\tau}} + \boldsymbol{\tau}^{\text{SV}} + \boldsymbol{\tau}^{\text{HF}} \tag{2.21}$$

The slowly-varying excitation loads occur when two regular waves with different frequencies interfere. The resulting wave load can be measured to have a frequency equal to the difference of the two regular wave frequencies. A general formula for these loads can be derived utilizing quadratic transfer functions (QTFs), according to OrcaFlex (2022). The slowly varying load in DOF m, for an irregular sea state with N regular wave components, can then be written as

$$\tau_m^{\text{SV}} = \sum_{k=1}^{N}\sum_{i=1}^{N} \text{Re}\Big\{\zeta_k\zeta_i\, T_m(\beta_k,\beta_i,\omega_k,\omega_i) \exp\Big(j(\omega_k-\omega_i)t - (\epsilon_k-\epsilon_i)\Big)\Big\}, \tag{2.22}$$

where ζ_k, ω_k, and ϵ_k represent the amplitude, frequency, and phase of wave component k, while $T_m(\beta_k,\beta_i,\omega_k,\omega_i)$ denotes the complex-valued QTF for the wave pair (k,i), in DOF m, consisting of both amplitude and phase information. Re denotes taking the real part of the complex value. The QTFs can, for all N wave components, be collected in a QTF matrix. For a fixed relative angle β, this can be represented by the matrix $\boldsymbol{T}(\beta) \in \mathbb{R}^{N\times N}$, as

$$\boldsymbol{T}(\beta) = \begin{bmatrix} T(\beta,\omega_1,\omega_1) & \dots & T(\beta,\omega_1,\omega_N) \\ \vdots & \ddots & \dots \\ T(\beta,\omega_N,\omega_1) & \dots & T(\beta,\omega_N,\omega_N) \end{bmatrix} \tag{2.23}$$

Finding the low-frequency wave load in (2.22) can be a challenging task. First of all, the double summation can lead to difficulties regarding computational capabilities. However, the more complex problem lies in determining the quadratic transfer functions. This can be done experimentally, but due to the substantial costs involved in large-scale tests, it is more commonly done numerically, for instance with programs such as WAMIT (3D potential theory) or VERES (2D strip theory). To avoid calculating the full QTF matrix in (2.23), a simplified approach, known as the *Newman's approximation* can be used. The approximation only requires knowledge of the diagonal terms, defined as the drift-coefficients, to approximate the full QTF. The approximation is widely used in the industry, and provide significantly less CPU time. The drift coefficients are also derived purely by first order analysis, simplifying the calculations even further.

2.2.4 Newman's approximation

Newman's approximation was first proposed by Newman (1974), and is widely used for estimating the off-diagonal terms of the quadratic transfer function in (2.23). The approximation is stated in OrcaFlex (2022), and in Faltinsen (1990), with slightly different notation. This thesis use the definition and mathematical expressions as

Definition. ***Newman's Approximation:***
The value of the QTF for (ω_i,ω_k) can be found as the average of its value for (ω_i,ω_i) and (ω_k,ω_k).

$$\underbrace{T(\omega_i,\omega_k) = T(\omega_k,\omega_i)}_{\text{Newman's definition}} = \underbrace{0.5\,(T(\omega_i,\omega_i) + T(\omega_k,\omega_k))}_{\text{Newman's approximation}} \tag{2.24}$$

According to Standing et al. (1987), the arithmetic mean in Equation 2.24 can be replaced by a geometric mean to further improve the approximation. This can be justified as, provided that there is little spread of wave directions, the far-from-diagonal QTF values are less crucial as they provide higher frequency

contributions. High-frequency loads are, for most vessels, often associated with little effects, due to their large inertia. Figure 2.3 displays how an arbitrary point in a fictive QTF can be calculated from its belonging diagonal with Newman's approximation.

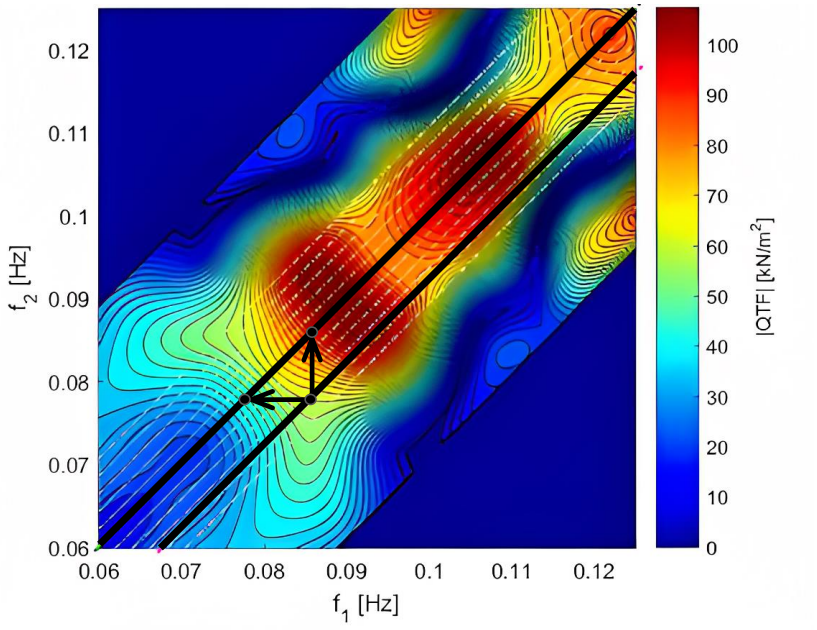


Figure 2.3: Heat map of a fictive QTF, displaying Newman's approximation. Courtesy of Fonseca and Stansberg (2017).

The approximation can be said to be suitable if the real part of the QTF at relevant difference-frequencies (such as natural frequencies) are close to the value at the main diagonal. As the imaginary part (phase) of the QTF is zero along the diagonal, Newman's approximation will lead to zero phase for all elements in the QTF. Thus, the approximation is best suited if the actual phase also is close to zero. Newman's approximation does however often result in satisfactory outcome, as $T(\omega_i, \omega_k)$ generally do not alternate a lot with frequency. Care must nevertheless be taken if ω is close to the resonance frequency as this can generate large motions (OrcaFlex, 2022). Figure 2.3 further strengthen the understanding of how local peaks in the QTF potentially can influence the validity of the approximation.

Two crucial cases for which care must be taken should also be mentioned when discussing Newman's approximation. Firstly, sea floor, or tank wall, effects may influence the validity of the approximation. Consequently the approximation can be insufficient in shallow waters. Secondly, waves with adjacent periods but distinct directions use far-from-diagonal QTF values. Thus, for vessels subject to a significant spread of wave directions caution should be exercised. In general the full QTF is to be preferred in both of these cases.

Mean drift forces can be described as a special case of difference-frequency loads that occur when $\omega_j = \omega_k$ in (2.22). This equality evidently leads to zero-frequency loads, indicating that a constant force is acting on the vessel. A term for the mean drift load can be derived by evaluating the expression for slowly-varying loads in (2.22). The zero-frequency definition will then eliminate the complex exponential term, resulting in the following more straightforward equation.

$$\overline{\tau}_m = \sum_{i=1}^{N} \zeta_i^2 T_m(\beta, \omega_i, \omega_i) \tag{2.25}$$

$T(\beta, \omega_i, \omega_i)$ is here taken as the real value of the diagonals in the QTF in (2.23). As expected, wave components pair of equal frequencies give a time-independent contribution which can give rise to a mean static offset of the vessel if not sufficiently compensated for.

2.3 Neural networks

Neural networks (NN) are a subset of machine learning (ML) and are considered to be at the heart of deep learning algorithms. The computing systems are based on the principle of the neurons in the brain and are designed to recognize patterns. A network is composed of interconnected nodes arranged in multiple layers, which process information and make predictions based on input data. NNs have been applied to a wide range of fields, including computer vision, natural language processing, predictive analysis, and control theory (IBM, 2023).

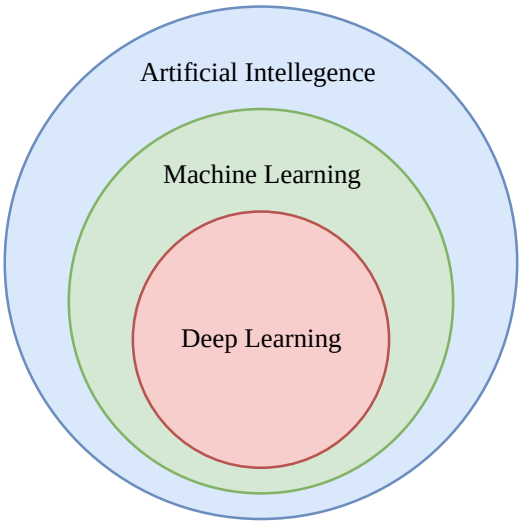


Figure 2.4: The relation between deep learning, machine learning and artificial intelligence.

The incorporation of neural networks in control theory dates back to the early 1990s. Since then, extensive attention has been given to develop different techniques and approaches to integrate NNs into a growing range of control applications. Within control theory, NNs have two major utility values; to approximate dynamic programming for solving optimal control problems, and to be used in closed-loop feedback control. However, implementing NNs to feedback control may introduce challenges related to the closed-loop stability, as well as to the performance of the system. It is, therefore, important to select a suitable control system structure and to tune the network properly (Al-Aubidy, 2023).

NN offer potential solutions to various aspects and challenges in control theory, including plant dynamics modeling. Modeling the dynamics of a system is known to be a challenging task, especially when there is limited or no information available about the plant model. By treating the plant model as a *black box*, NNs can be used to provide numerical information about the behavior of the system without requiring any mathematical description of it.

The fast learning capabilities of NNs make them suitable to for controllers that aim to adapt to changing environments (Al-Aubidy, 2023). NNs are also well-suited for making predictions about future outcomes, through processing historical data. In the context of marine control theory, accurate predictions, for instance of the unknown bias load, can be used to increase the accuracy and performance of a DP controller. This aspect will be further investigated in Chapter 4.

2.3.1 Topology

NN algorithms are structured as a stack of multiple different layers of artificial neurons. The layers can generally be categorized in three, being the input layer, the hidden layer, and the output layer, where

the neurons of one layer is connected to its adjacent layers through weighted channels. Data is initially inserted into the input layer and weighted based on its impact on the output. The weighted inputs are them passed through one or more hidden layers. The hidden layers process the data through activation functions, $h(\cdot)$, specifically selected to capture the desired characteristics of the process. The nodes in the hidden layer will have specific threshold values assigned to them. If the value of a node exceeds its threshold, the node is activated and the weighted signal is transmitted to the next layer. Conversely, if the output has a value below the threshold, the signal is not passed through the node. Ultimately, the output layer combines the processed data into the expected output $\boldsymbol{z}$ (IBM, 2023). The complete process is illustrated in Figure 2.5. The constant node in the figure represents a bias in the network.

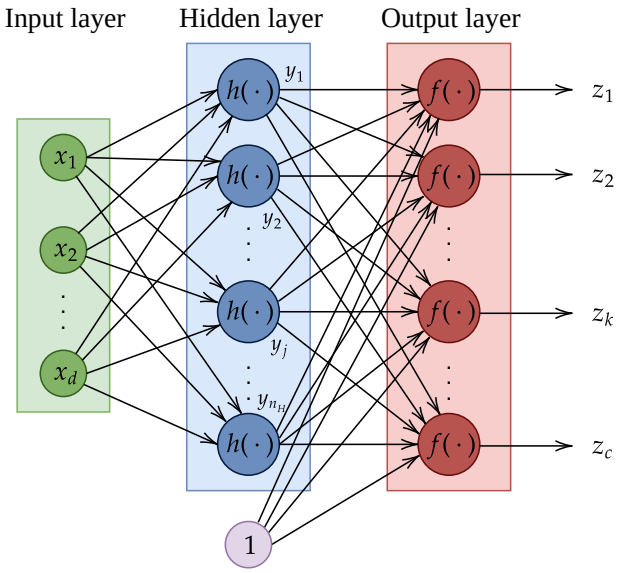


Figure 2.5: Neural network with one hidden layer.

Increasing the number of layers in a NN, generally improves its accuracy. However, the complexity of the network also increases, and the network can easily become computationally demanding. Hence, it is important to find the right balance between efficiency and accuracy. In many cases, it can be sufficient with a relatively shallow network to achieve satisfactory results, provided that the NN is well constructed, i.e. if well suited activation functions have been chosen.

2.3.2 Activation functions

The performance of a neural network is heavily linked with the selection of activation functions in its layers. The activation functions transfer the summed weight of input from the previous layer into an output value to be fed into the following layer. Consequently, they play a crucial role in capturing nonlinearities of the observed dynamics. Without an activation function, the neural network would simply be reduced to a linear function of its input, which can be considered purposeless as most realistic processes involve significant nonlinearities.

When selecting an activation function, there are several important factors to consider. Firstly, the function should be continuous and differentiable. Non-continuous and non-differentiable activation functions can provide complications when training the network with backpropagation, to be described in Section 2.3.3. It is also important to adjust the activation function based on the type of prediction problem that is being solved. These task-specific considerations can involve the dynamics of the system, as well as the characteristics of the input data, and must be evaluated in relation with the properties of the activation functions. Overall, this often lead to a trade-off between the performance of the predicted results, and challenges related to complexity and computational time.

The activation function can be chosen from a variety of alternatives, where some commonly used options are presented in the following bullet points. The information is gathered from Baheti (2021), and Sharma (2017).

- **Sigmoid function**: $f(x) = \dfrac{1}{1+e^{-x}}$

  The Sigmoid function is a smooth, s-shaped function that can take a value between 0 and 1. It is thus commonly utilized in networks that aims to solve binary identification and probability problems. Even though its derivative is continuous and easily derived, it can often lead to issues related to the *problem of vanishing gradient*. This problem occur when the value of the gradient becomes adequately small, which can cause the the weights of the lower layers to be updated very slowly, or not at all.

- **Hyperbolic tangent**: $f(x) = \tanh x$

  The hyperbolic tangent function has large similarities to the Sigmoid function. The function is continuous, easily differentiated, and also takes the same s-shape. Its output values are however zero centered and span from -1 to 1. This enables the network to more easily map the output values as strongly negative, neutral, or strongly positive. The hyperbolic tangent does also face issues related to a vanishing gradient, but is often preferred over the Sigmoid function, as it has a steeper, and more substantial gradient. Due to the zero centering of the function, the hyperbolic tangent is also more capable of capturing nonlinearities, as the gradient is not restricted to move in a certain direction.

- **Gaussian radial basis function** $f(x) = \exp\left[-\beta||x-c||^2\right]$

  Gaussian radial basis functions (RBFs) can be highly practical to use in NNs. They are considered particularly useful in function approximation tasks, as the functions have the ability to interpolate any set of input-output pairs exactly, as long as enough RBFs are applied. The functions are also highly nonlinear, and thus, excellent at capturing nonlinearities in the training data (Liu, 2013). However, it is important to note that RBFs are also prone to overfitting, where the model fails to generalize well to unseen data. Additionally, the RBFs can be computationally more demanding compared to other activation functions. The functions also require center initialization, which often is considered a difficult task, but, nonetheless, plays a crucial role for achieving optimal performance of the network.

- **ReLU and leaky ReLU functions** $f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0.01x, & \text{if } x < 0 \end{cases}$

  The rectified linear unit (ReLU) function is a piecewise linear function that returns the input, given that the input is positive. If not, zero is returned. The function may intuitively give the impression of a linear function, even though it still has a derivative function, which consequently allow backpropagation to be applied. The ReLU function is therefore one of the more popular activation functions in the world today. An issue does nonetheless arise due to the fact that all negative input values will become zero. This affects the model's ability to fit the weights to the data set during backpropagation properly, which in turn affects the resulting graph by not mapping the negative values adequately. This is often referred to as the *dying ReLu problem*, and can be avoided by implementing the leaky ReLU function. The leaky ReLU function will, similarly to the original function, return the input value for all positive numbers. For negative inputs however, it has a small positive slope, ensuring that the gradient of the function never will come to be zero for negative inputs. This will eventually guarantee that all neurons play a part in updating the weights during backpropagation.

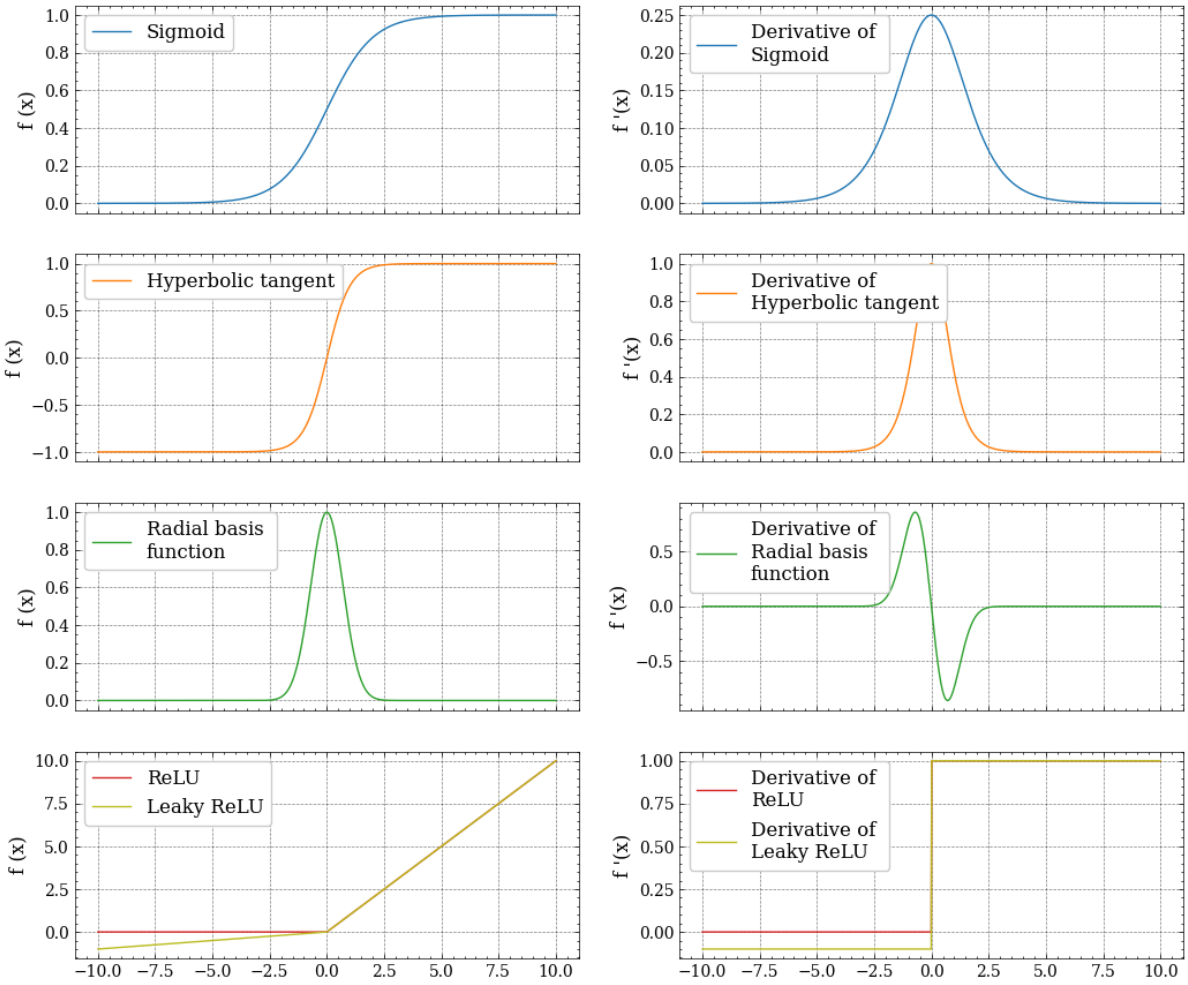


Figure 2.6: Different activation functions and their derivative in a neural network.

2.3.3 Training a neural network

Neural networks are designed to learn and adapt from data, and in order for networks to work as intended, proper training must be conducted. Through the process of adjusting the weights of the network, the neural network can learn to identify patterns and generate predictions based on new inputs in the future. The procedure is therefore pivotal for the performance of the network, and rely on a proper and representative set of data.

Two important concepts in the process of training a neural network, are *backpropagation*, and *gradient descend*. Backpropagation is the process of calculating the error rate of the output of a network, and using the chain rule to propagate this rate backwards through the network. The calculated error gradients can then be utilized to modify the weight parameters in order to minimize an overall error function. This must be accomplished by an optimization algorithm, such as the commonly used gradient descend. The algorithm adjust the weights in the direction of the negative gradient of the loss function, and the size of the update is determined by a given learning rate to be selected by the creator of the network. The learning rate must be selected in such a manner that the global minimum of the error function is reached.

The process of training a neural network can be summarized in the four following steps.

1. Feed the network with input from a training data set, and propagate through the network to calculate an output.
2. Compare the calculated output from the network with the actual data, and calculate an error

based on a selected error function.

3. Backpropagate through the network and adjust the weights using for instance a gradient descent technique.

4. Repeat step 1-3 for the complete dataset.

Tuning a neural network can also give rise to some unforeseen complications. One of the most common of which is the problem of *overfitting*. This concept occurs when a network is trained to fit its training data perfectly, at the expense of the algorithm's performance against upcoming and unseen data. This may be difficult to notice, as the network will show no sign of overfitting during the training process. It will however most likely result in poor performance when tested on different data sets. As mentioned in Section 2.3.2, problems also occur if the value of the gradient of the activation functions become too small. This is especially the case for deep neural networks with several hidden layers, as it can lead to an increased difficulty to update the weights and biases efficiently.

2.3.4 Developing a neural network

The development of a neural network can be a complicated and time consuming process. Several different library tools which can be of use, such as the widely used Python package, PyTorch, already exist. Functionality from this library has been used for this thesis. Nonetheless, to guarantee that the network behaves as desired for a specific problem, it is important to understand the fundamentals of how to implement a neural network without such assistance. The following paragraphs describe how to implement a simple neural network. The procedure is heavily inspired by the process described in Aflak (2018).

In general, the implementation of a neural network consists of defining two fundamental layers, namely a *dense layer* and an *activation layer*. The dense layer describes the process of moving from one layer in a neural network to its adjacent layer, while the activation layer describes the process of sending a data through an activation function, as described in Section 2.3.2. When moving forward (forward propagation) in a neural network, both layers take in a vector, $\boldsymbol{X} \in \mathbb{R}^n$, and returns a vector, $\boldsymbol{Y} \in \mathbb{R}^m$. Similarly, when moving backwards (backpropagation) in the network, the error gradient with respect to the output, $\frac{\partial E}{\partial \boldsymbol{Y}} \in \mathbb{R}^m$, is handled as an input, before returning the error gradient with respect to the original input, $\frac{\partial E}{\partial \boldsymbol{X}} \in \mathbb{R}^n$. The general structure can be seen in Figure 2.7. The dense layer must also differentiate the error with respect to the weights in the layer, $\frac{\partial E}{\partial \boldsymbol{W}} \in \mathbb{R}^{m \times n}$, to be used when updating the weights using gradient descend.

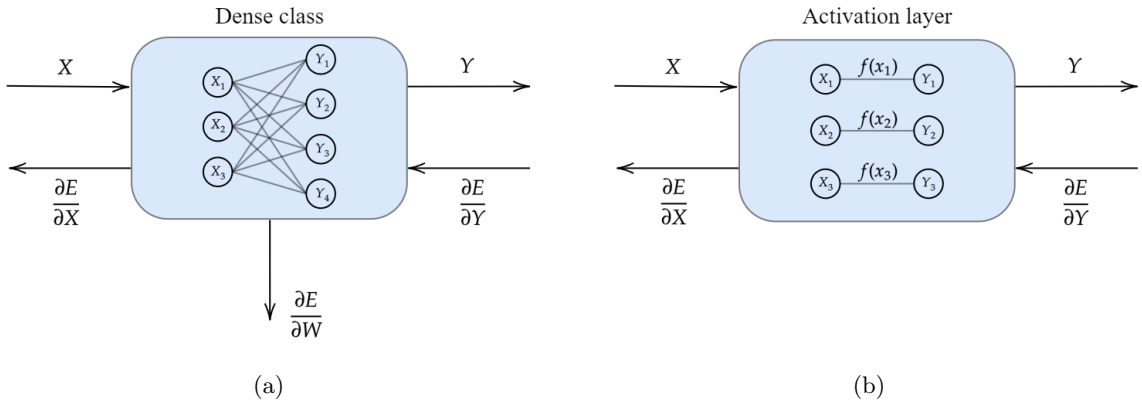


Figure 2.7: Dense and activation layer structure.

By evaluating the structure of Figure 2.7, it can be seen that both layers must be able to handle both forward propagation and backpropagation. During forward propagation, the input, $\boldsymbol{X}$, must be transformed into an output, $\boldsymbol{Y}$. This can for both layers be considered as a rather straightforward

process. For backpropagation however, the error gradient, $\frac{\partial E}{\partial \boldsymbol{Y}}$, is handled as input, and must be utilized in order to calculate $\frac{\partial E}{\partial \boldsymbol{X}}$ and $\frac{\partial E}{\partial \boldsymbol{W}}$. This procedure is derived using the chain rule, as summarized in (2.26)-(2.28). The reader is referred to Aflak (2018) for a more detailed derivation.

$$\frac{\partial E}{\partial \boldsymbol{X}} = \frac{\partial E}{\partial \boldsymbol{Y}}\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{X}} = \begin{cases} \boldsymbol{W}^\top \dfrac{\partial E}{\partial \boldsymbol{Y}}, & \text{if dense layer.} \\ \dfrac{\partial E}{\partial \boldsymbol{Y}} \odot f'(\boldsymbol{X}), & \text{if activation layer.} \end{cases} \tag{2.26}$$

$$\frac{\partial E}{\partial \boldsymbol{W}} = \frac{\partial E}{\partial \boldsymbol{Y}}\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{W}} = \frac{\partial E}{\partial \boldsymbol{Y}}\boldsymbol{X}^\top \tag{2.27}$$

$$\boldsymbol{Y} = \begin{cases} \boldsymbol{W}\boldsymbol{X} + \boldsymbol{B}, & \text{if dense layer.} \\ f(\boldsymbol{X}), & \text{if activation layer.} \end{cases} \tag{2.28}$$

A selected number of the two layers in Figure 2.7 can then be connected together in series, to form a complete NN. The output of one layer will then function as input in the sequential, both when propagating forward and backward, and can be seen in Figure 2.8. These layers are not to be confused with the amount of layers presented in Figure 2.5, and do not represent the depth of the NN.

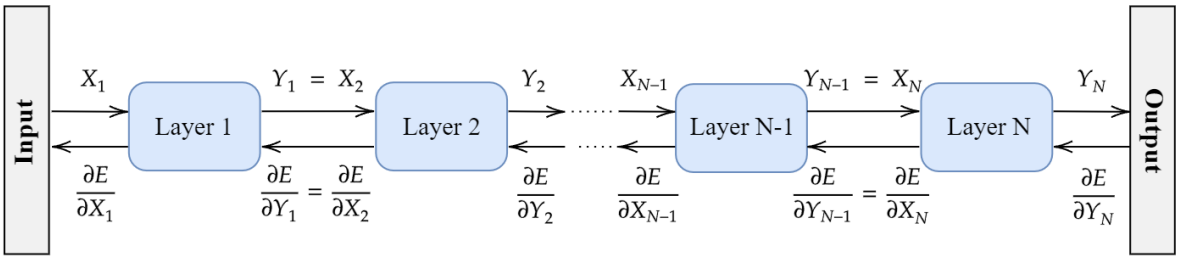


Figure 2.8: Layers structured in series, forming a neural network.

In the layout above, only the final layer requires the input to be uniquely defined while propagating backwards, as it does not receive any error derivative from its neighboring layer. A differentiable error function must therefore be specified by the developer. Given the output of the final layer $\boldsymbol{Y}$, and the actual desired output of the network, denoted as $\boldsymbol{Y}^*$, the mean square error (MSE) can be defined as

$$E_{MSE} = \frac{1}{n}\sum_i (y_i^* - y_i)^2, \tag{2.29}$$

with the derivative

$$\frac{\partial E_{MSE}}{\partial \boldsymbol{Y}} = \frac{\partial}{\partial \boldsymbol{Y}}\left[\frac{1}{n}\sum_i (y^* - y)^2\right] = \frac{2}{n}(\boldsymbol{Y} - \boldsymbol{Y}^*). \tag{2.30}$$

The selection of error function is by no means limited only to MSE as shown above. Pastor-Pellicer et al. (2013) does, among others, propose an error backpropagation algorithm based on the *F-Measure*, while Tian et al. (2022) discusses several more complicated loss functions and their advantages for deep learning neural networks.

2.4 Dynamic positioning control system

A Dynamic Positioning vessel is by definition a vessel that automatically maintains a desired position and heading exclusively by means of active thrusters (IMO, 1994). The technology was first introduced in the 1960s, even though it would take several years before it was developed into the advanced systems known today. The motivation behind the evolution originated from the offshore oil and gas industry. As new oil fields progressed into more challenging depths, the traditional bottom fixed platforms could no longer be utilized. As a result, floating production rigs were developed and taken into use. Anchoring at such depths was nonetheless vastly unpractical, forcing engineers to think outside the box and explore the possibilities of implementing dynamic positioning.

DP systems have traditionally been utilized for low-speed maneuvering and stationkeeping, but in later years standard autopilots and waypoint-tracking functionalities have also been implemented into the DP system. A trend seen today, nonetheless, is the realization of high-speed functionalities, where the ultimate goal is that this can merge with the more traditional DP systems and create a unified system for all speed ranges (Fossen, 2021, ch. 15.3).

It is normal to classify DP systems according to their reliability. The International Maritime Organization (IMO) operates with three equipment classes, all defined by their worst case failure (IMO, 1994) . For DP class I, loss of position may occur in the event of a single fault. DP class II, loss of position is not to occur in the event of a single fault in any component of the system. All components must thus be redundant. The system must consist of at least two independent computer systems, as well as at least three separate positioning systems and three sensor systems. DP class III includes all regulations in class II, with the addition of redundancy in technical design and physical arrangement to prevent failure during for instance flooding and fire.

A commercial DP system must be robust and able to compensate for environmental forces and unmodelled dynamics. These are some of the most important design requirements, as a full-state feedback controller will fail in bad weather conditions unless the environmental forces are included in the design specifications. The following requirements must therefore be present in an industrial vessel DP control system, according to Fossen (2021):

- Disturbance rejection to compensate for slowly-varying loads.
- Wind feedforward control to compensate for wind gusts.
- Wave filtering to suppress first order wave frequency motion from entering the feedback loop and cause wear and tear of actuators.
- State observer for noise filtering and the estimation of unmeasured states such as velocity.
- Optimal allocation of thrust to ensure that the desired thrust is provided by the propulsion system on board.
- Signal filtering and fault detection/mitigation of measurements.

Understandably, other requirements do exist when designing a motion control system, but the aforementioned can be considered as the most essentials in a DP system.

2.4.1 Objective

Seagoing vessels are continuously subjected to external forces from wind, waves and current. A DP system aims to counteract these forces by controlling the propulsion system, obliging the vessel to maintain either a fixed or dynamic position. The system must thus be able to evaluate the effects of wind, wave and current on a vessel and initiate measures to prevent these. The difference between the pose of the vessel, $\boldsymbol{\eta}$, and the desired reference pose, $\boldsymbol{\eta}_d$, can be defined as the error, often denoted

$\tilde{\boldsymbol{\eta}}$. The objective of the system can then be said to ensure a convergence of $\tilde{\boldsymbol{\eta}}(t)$ towards zero, written mathematically as

$$\tilde{\boldsymbol{\eta}}(t) = \boldsymbol{\eta}(t) - \boldsymbol{\eta}_d(t) \to \mathbf{0} \text{ as } t \to \infty. \tag{2.31}$$

Similarly, the desired velocity, $\boldsymbol{\nu}_d$ can be specified. The DP-system must then also consider the convergence of the velocity error of the vessel when calculating the desired thrust force, that is,

$$\tilde{\boldsymbol{\nu}}(t) = \boldsymbol{\nu}(t) - \boldsymbol{\nu}_d(t) \to \mathbf{0} \text{ as } t \to \infty. \tag{2.32}$$

2.4.2 Reference frames

When modelling and analyzing a DP vessel it can be convenient to first define the different reference frames, for which the vessel's motion can be described about. A broad variety of frames exist, but this thesis will mainly utilize two of the most frequently used in the industry. These are the geographical NED (North-East-Down) frame, denoted $\{n\}$, and the body-fixed BODY frame, denoted $\{b\}$. The NED-frame is defined as a tangent plane located on the Earth's ellipsoid, with its axis pointing towards true North, East and downwards normal to the Earth's surface. The BODY-frame has its origin fixed to a moving vessel, usually located midships in the waterline.

When expressing the position of a marine craft, it is normal to do so in the geographical NED-frame. The generalized position is then defined as

$$\boldsymbol{\eta}^n = [x^n \quad y^n \quad z^n \quad \phi \quad \theta \quad \psi]^\top. \tag{2.33}$$

The orientation of the vessel is here expressed using Euler angles, $\boldsymbol{\Theta}$. It is then possible to extract the generalized velocities of the vessel by taking the time derivative of the expression, that is

$$\dot{\boldsymbol{\eta}}^n = [\dot{x}^n \quad \dot{y}^n \quad \dot{z}^n \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^\top. \tag{2.34}$$

It is, however, often more advantageous to express these velocities in the BODY-frame when deriving the equation of motion. The velocities are then denoted as

$$\boldsymbol{\nu}^b = [u \quad v \quad w \quad p \quad q \quad r]^\top. \tag{2.35}$$

The superscript in the aforestated equations will be dismissed in further equations. It is important that all forces and motions in an equation are expressed in the same reference frame. The ability to rotate a vector in between the two defined frames is therefore a key property when deriving the equation of motions. This can be done by a rotation matrix $\mathbf{J}_\Theta$ according to Fossen (2021), such that

$$\mathbf{J}_\Theta(\boldsymbol{\eta}) = \begin{bmatrix} \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{T}_\Theta(\boldsymbol{\Theta}_{nb}) \end{bmatrix}. \tag{2.36}$$

The matrix utilizes the orientation of the vessel, expressed by its Euler angles, and rotates both the linear and angular velocity from $\{b\}$ to $\{n\}$, by the relation $\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}$. Its inverse can also be used to transform the other way around. Other options to describe the orientation of the vessel, such as quaternions, also exist, but will not be granted any further attention in this thesis. A broader and more detailed derivation of the rotation matrix is given by Fossen (2021).

When modeling a DP-vessel, it is often sufficient to only consider horizontal motion. By assuming small angular motions in roll and pitch, the 6 DOF kinematics represented above can then be reduced to only consider surge, sway, and yaw. The rotation matrix will then also be simplified, using only the yaw angle as input, giving the kinematics

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.37a}$$

$$\begin{bmatrix} \dot{x}^n \\ \dot{y}^n \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}, \tag{2.37b}$$

where the rotation matrix has the useful property of $\mathbf{R}^{-1}(\psi) = \mathbf{R}^\top(\psi)$.

2.4.3 Typical topology

Similar to other motion control systems, the DP system is often separated into three independent blocks; Guidance, Navigation and Control (Fossen, 2022, ch.11). These blocks can be complex and consist of several subsystems. A brief description is given underneath. Figure 2.9 shows a block diagram of a complete DP system where the Guidance, Navigation and Control systems all are marked.

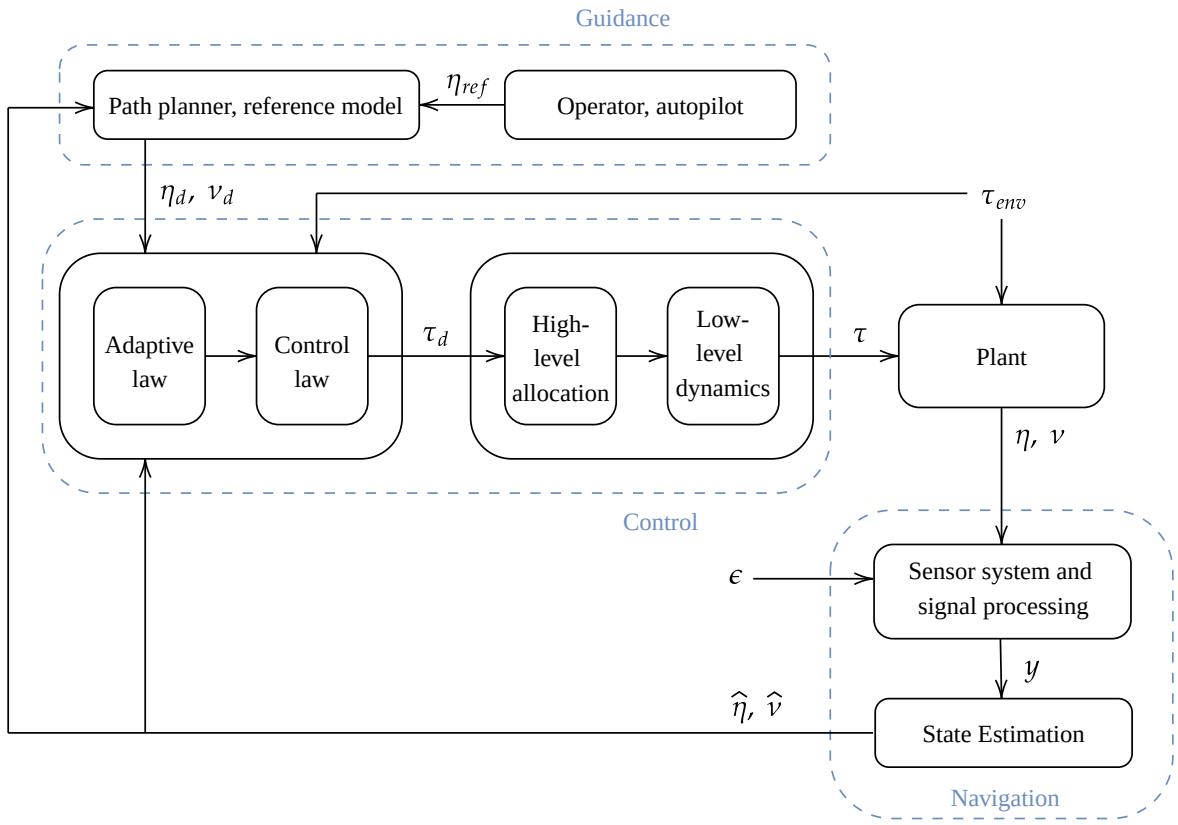


Figure 2.9: Block diagram displaying a typical DP system, separated into Guidance, Navigation and Control.

Guidance: The guidance system has the objective of computing the desired position, velocity, and heading of the vessel based on some given setpoints. These parameters are usually inserted and utilized in the control system block. The basic guidance system often consists of a path planner and a reference model, where advanced optimization techniques can be implemented to obtain an optimal trajectory.

Navigation: The navigation system has the objective of directing the vessel by determining its states (position, velocity, and attitude). The block consists of sensor systems, signal processing, and state estimators, with the desire to provide both the guidance and control system with accurate and clean signals.

Control: The control system has the objective of determining the necessary action required to satisfy the control objective. The control objective is often seen in conjunction with the guidance system, and it can involve the intention of minimizing energy, tracking trajectories, and maneuvering control. In addition to the control law, consisting of both feedback and feedforward control laws, control allocation is also a part of the control block.

Figure 2.9 presents how these three blocks are outlined as well as how they interact with each other. At the core of the DP system, the controller is located. The controller receives signal about desired position and velocity from the guidance block, as well as information about the actual position and velocity from the navigation block, and utilizes this knowledge in a control law. The control law calculates the desired

force required to obtain the DP objective, mentioned in Section 2.4.1. It can make use of the calculated state error in a feedback loop, as well as measurements of the wind velocity in a feedforward signal. Signals from the navigation block can also be used to create estimates of unknown parameters to be utilized in an adaptive law.

2.4.4 Thrust allocation

The high-level thrust allocation is a part of the control block in a general DP system which receives the desired control loads calculated in the control law, and transform these to control inputs for the onboard propulsion system. DP vessels tend to be overactuated, meaning that an optimization problem must be solved in order to find the desired control inputs. This thesis will purely focus on thrust allocation for azimuth thrusters, as this is the only propulsion unit installed on the the model vessel used for this thesis. Other propulsion systems such as tunnel thrusters, main propellers, and rudders are, however, also normally installed on DP vessels.

In general the high-level thrust allocation can be summarized as in (2.38) (Fossen, 2021). The actuator forces and moments, $\boldsymbol{\tau} \in \mathbb{R}^n$ relate to the control forces and moments, $\boldsymbol{f} \in \mathbb{R}^r$, by (2.38a), where $\boldsymbol{T}(\boldsymbol{\alpha}) \in \mathbb{R}^{n \times r}$ is the thrust configuration matrix dependent on the azimuth angles, $\alpha \in \mathbb{R}^r$. For a vessel equipped with r azimuth thrusters for operation in n DOFs, the thrust configuration matrix describes the geometry or locations of the actuators. The control forces and moments can again be expressed as in (2.38b), where $\boldsymbol{K} \in \mathbb{R}^{r \times r}$ is the diagonal force coefficient matrix and $\boldsymbol{u} \in \mathbb{R}^r$ is the control inputs.

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} \tag{2.38a}$$
$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{u} \tag{2.38b}$$

(2.38) presents a nonlinear allocation problem due to the configuration matrix' dependence on the azimuth angles. When solving the thrust allocation problem for rotatable thrusters, it can be desirable to formulate the problem as in (2.39). The subscript e in (2.39) indicates that the vectors and matrices are extended. This is done to make the problem independent of the azimuth angles and hence, easier to solve.

$$\boldsymbol{\tau} = \boldsymbol{T}_e \boldsymbol{f}_e \tag{2.39a}$$
$$\boldsymbol{f}_e = \boldsymbol{K}_e \boldsymbol{u}_e, \tag{2.39b}$$

In (2.39), the rotatable forces are treated as two forces. The extended thrust configuration matrix, $\boldsymbol{T}_e$, is defined with a set of column vectors (two for each azimuth propeller), describing the unit force in x- and y-direction, as well as the moment arm in yaw. $\boldsymbol{T}_e$ will thus be defined as a $n \times 2r$-matrix. The thrust coefficient matrix is simply a diagonal matrix describing the force coefficients, K_i, of each azimuth propeller. For a DP system, when $n = 3$, the expression in (2.38) can be written out as

$$\underbrace{\begin{bmatrix} X \\ Y \\ N \end{bmatrix}}_{\mathbf{3\times 1}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 1 & \cdots \\ 1 & 0 & 1 & 0 & \cdots \\ l_{x_1} & l_{y_1} & l_{x_2} & l_{y_2} & \cdots \end{bmatrix}}_{\mathbf{3\times 2r}} \underbrace{\begin{bmatrix} K_1 & 0 & \cdots & 0 \\ 0 & K_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_r \end{bmatrix}}_{\mathbf{2r\times 2r}} \underbrace{\begin{bmatrix} u_{1_x} \\ u_{1_y} \\ u_{2_x} \\ u_{2_y} \\ \vdots \\ u_{p_y} \end{bmatrix}}_{\mathbf{2r\times 1}}.$$

The extended control inputs can then, given the desired thrust vector, be calculated by

$$\boldsymbol{u}_e = \boldsymbol{K}_e^{-1} \boldsymbol{T}_e^{\dagger} \boldsymbol{\tau}, \tag{2.40}$$

where $\boldsymbol{T}_e^\dagger$ is the right *Moore-Penrose pseudoinverse* (Penrose, 1955) of $\boldsymbol{T}_e$. The actual control inputs and angles can be calculated by mapping the pairs (u_{i_x}, u_{i_y}) according to (2.41), which have to be modified to low level machinery signal [rpm],

$$u_i = \sqrt{u_{i_x}^2 + u_{i_y}^2}, \qquad \alpha_i = \texttt{atan2}(u_{i_y}, u_{i_x}), \qquad i = 1, 2, ..., p. \tag{2.41}$$

The allocation problem is a an optimization problem, where the optimal solution should fulfill all given constraints as sufficient as possible. In industrial propulsion systems, both physical constraints, such as saturation and forbidden zones, as well as manually set constraints, like fuel or energy optimization must be considered.

Thruster bias, which is often implemented for DP systems, can complicate the allocation problem. However, it has its benefits. By having the thrusters deliver a minimum thrust at all times at a favorable directional spreading, the DP vessels will be more responsive and will easier overcome slow azimuth response by improving their ability to produce resulting forces in any direction (Kaasen, 2015).

2.4.5 Reference model

In order for the control block to provide the vessel with control signals, the guidance system has to distribute a desired position and velocity , which is provided by the reference model in the guidance block. A reference model computes the desired trajectory based on a given set of waypoint. The reference position given by an operator (or other trajectory generators), $\boldsymbol{x}_{ref}$, does then behave as the input to the reference model, which generates a smooth signal without any steps $\boldsymbol{x}_d$. This prevents sudden jumps in the error state, as well as avoiding saturation in acceleration and velocity of the vessel. It is vital that the bandwidth of the chosen reference model is selected lower than the bandwidth of the DP system in order to obtain satisfactory tracking performance and stability.

The reference model can, according to Fossen, 2022, ch.12, be modelled as a second order mass-spring-damper system connected in cascade with a first order low-pass filter. Mathematically this can be represented by the state space representation shown below. For the model, $\boldsymbol{\Delta} = \text{diag}\{\zeta_1, \zeta_2, \zeta_3\}$ is the relative damping matrix, and $\boldsymbol{\Omega} = \text{diag}\{\omega_1, \omega_2, \omega_3\}$ is the reference model's natural frequency matrix. Both of these matrices are tuned according to the desired performance of the reference system.

$$\dot{\boldsymbol{x}}_d = \mathbf{A}_d \boldsymbol{x}_d + \mathbf{B}_d \mathbf{x}_{ref} \tag{2.42}$$

where $\boldsymbol{x}_d = [\boldsymbol{\eta}_d^\top \quad \dot{\boldsymbol{\eta}}_d^\top \quad \ddot{\boldsymbol{\eta}}_d^\top]^\top \in \mathbb{R}^9$ and

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \\ -\boldsymbol{\Omega}^3 & -(2\boldsymbol{\Delta} + \mathbf{I}_{3x3})\boldsymbol{\Omega}^2 & -(2\boldsymbol{\Delta} + \mathbf{I}_{3x3})\boldsymbol{\Omega} \end{bmatrix}, \mathbf{B}_d = \begin{bmatrix} \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} \\ \boldsymbol{\Omega}^3 \end{bmatrix} \tag{2.43}$$

Both the controller block and guidance block are depending on reliable system state signals from the navigation block. The position and orientation of the vessel is measured in the sensor system, which can be influenced by noise, bias and signal failures. The recurrence of the latter one can be reduced by enabling measures such as different error tests (signal range, variance, and wildpoint test) as well as utilizing voting and weighting of redundant sensors (Sørensen, 2018, ch.5). As for extracting the actual state from sensor bias and noise, a state observer can be implemented as in Section 2.4.7.

2.4.6 Control design model

Several blocks in Figure 2.9 are so-called model based components. Thus, they require that an adequate vessel model is implemented. For that reason a *control design model* (CDM) is derived. The CDM is used in model-based observers and controllers and is a simplified model of the simulation model. Its objective is to describe the most essential behaviour of the system. A more complete, high-fidelity

system model is described in Section 2.5. The CDM can also, according to Sørensen (2018), be used to evaluate the stability of the system. It is often advantageous to divide the control plant model into a low frequency (LF) and a wave frequency model. As the model is to be used in a horizontal plane DP system, it only contains three DOFs (surge, sway and yaw).

The simplifications in the CDM are based on the assumption that ϕ and θ are small. This simplifies the diagonal elements in the transformation matrix presented in (2.36) to $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) \approx \mathbf{R}_{z,\psi}$ and $\mathbf{T}_\Theta(\boldsymbol{\Theta}_{nb}) \approx \boldsymbol{I}_{3\times 3}$. The simplified kinematic equation is given in (2.37a) and can be written once more as

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu},$$

where $\boldsymbol{R}(\psi) := \boldsymbol{R}_{z,\psi}$ with $\boldsymbol{v} = [u, v, r]^\top$ and $\boldsymbol{\eta} = [x^n, y^n, \psi]^\top$.

Low-frequency control design model

The low-frequent control design model is mainly driven by second order mean and slowly varying wave loads (see Section 2.2.3), as well as current and wind loads. Low speed is an important assumption when deriving models for DP systems, and based on this assumption an equation describing the kinetics of the vessel can be greatly simplified. Firstly, the Coriolis and centripretal forces arise due to rotation of the $\{b\}$ about $\{n\}$. For low speed vessels, the rotation of $\{b\}$ is small, and hence these terms can be neglected. The low-speed assumption also indicate that linear damping is a good approximation. As all DOFs are in the horizontal plane and no mooring lines are fastened, all restoring terms can be disregarded as well. The kinetics can then, according to Newton's second law of motion, be described as

$$\mathbf{M}\dot{\boldsymbol{\nu}} = -\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^\top(\psi)\mathbf{b} + \boldsymbol{\tau}_{ctrl}, \tag{2.44}$$

where $\mathbf{b}$ is the bias model that represents second order, slowly varying loads as well as all unmodelled dynamics. $\boldsymbol{\tau}_{ctrl}$ is the applied thrust forces delivered by the propulsion system, and $\mathbf{M}$ is the total inertia matrix consisting of both rigid body and hydrodynamic inertia (added mass). Notice also how the rotation matrix $\mathbf{R}(\psi)$ makes the model nonlinear because of its trigonometric functions. The estimate of the bias term will be nonphysical because several unmeasured components are compensated for simultaneously. This is often referred to as the *DP curent* because of its behaviour as a drift force due to second order wave drift forces, ocean currents and unmodeled dynamics (Fossen, 2022). In the CDM, it can be modelled as a random walk disturbance, displayed mathematically as

$$\dot{\boldsymbol{b}} = \boldsymbol{w}_b. \tag{2.45}$$

Several other bias models also exist. A frequently used model for marine control applications is for instance the first order *Markov process* (Sørensen, 2018, ch.7), which provides great abilities when it comes to stability calculations. Some of these bias models will be elaborated further in Chapter 4. Upcoming sections will also elaborate on how the design of the control law can help to effectively compensate for the bias term.

Wave-frequency control design model

It can be useful to describe the wave-frequency motion by a simple model. It is therefore conventional to model the WF motions as an harmonically damped oscillator. This approach is easy to implement, and makes the performance and robustness of the control system easy to test. The model is driven by white noise, $\boldsymbol{w}_\mathrm{w}$, as input, and works as a shaping filter, shaped according to a desired wave spectrum, which colors the white noise. Mathematically this can be represented on the linear state space form as

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_\mathrm{w}\boldsymbol{\xi}_\mathrm{w} + \mathbf{E}_\mathrm{w}\boldsymbol{w}_\mathrm{w} \tag{2.46a}$$

$$\boldsymbol{\eta}_\mathrm{w} = \mathbf{C}_\mathrm{w}\boldsymbol{\xi}_\mathrm{w}, \tag{2.46b}$$

where $\boldsymbol{\eta}_w$ represent the linear wave response in surge, sway, and yaw, while $\mathbf{C}_w \in \mathbb{R}^{3\times 6}$ is the measurement matrix, and $\boldsymbol{\xi} \in \mathbb{R}^6$ is the state vector. $\mathbf{A}_w \in \mathbb{R}^{6\times 6}$ is the system matrix, and $\mathbf{E}_w \in \mathbb{R}^{6\times 3}$ is

the disturbance matrix. The matrices in (2.46) describes the sea state in accordance to a selected wave spectrum and should be tuned accordingly.

$$\mathbf{A}_{\mathrm{w}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ -\mathbf{\Omega}^2 & -2\Lambda\Omega \end{bmatrix}, \qquad \mathbf{E}_{\mathrm{w}} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{K}_w \end{bmatrix}, \qquad \mathbf{C}_{\mathrm{w}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}$$

Here $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \lambda_3\}$, $\mathbf{\Omega} = \mathrm{diag}\{\omega_1, \omega_2, \omega_3\}$ and $\mathbf{K}_w = \mathrm{diag}\{k_{w_1}, k_{w_2}, k_{w_3}\}$ are tuned to obtain a realistic result of the sea state.

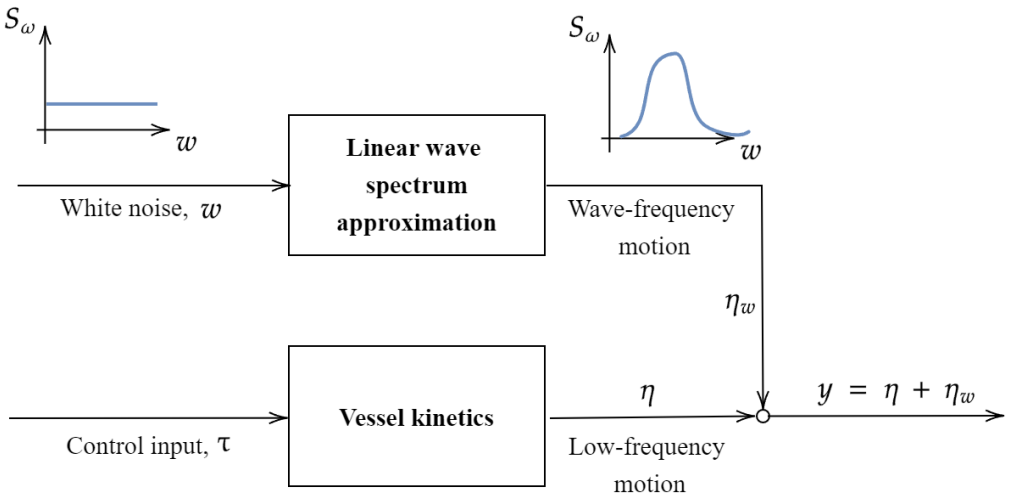


Figure 2.10: WF motion obtained with colored noise.

Total control design model

The complete CDM can then be represented as a combination of the LF and WF models. The total output derived from sensor measurements is expressed using superposition of $\boldsymbol{\eta}$ and $\boldsymbol{\eta}_{\mathrm{w}}$, while adding sensor noise as a zero-mean white noise process $\mathbf{v} \in \mathbb{R}^3$.

$$\mathbf{y} = \boldsymbol{\eta} + \boldsymbol{\eta}_{\mathrm{w}} + \mathbf{v} \tag{2.47}$$

The final model can then be summarized by combining the equations derived so far as

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_{\mathrm{w}}\boldsymbol{\xi}_{\mathrm{w}} + \mathbf{E}_{\mathrm{w}}\boldsymbol{w}_{\mathrm{w}} \tag{2.48a}$$
$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.48b}$$
$$\dot{\boldsymbol{b}} = \boldsymbol{w}_b \tag{2.48c}$$
$$\mathbf{M}\dot{\boldsymbol{\nu}} = -\mathbf{D}\boldsymbol{\nu} + \mathbf{R}^\top(\psi)\mathbf{b} + \boldsymbol{\tau}_{ctrl} \tag{2.48d}$$
$$\mathbf{y} = \boldsymbol{\eta} + \mathbf{C}_{\mathrm{w}}\boldsymbol{\xi}_{\mathrm{w}} + \mathbf{v} \tag{2.48e}$$

2.4.7 Observer design

It was mentioned that a DP system is required to perform wave filtering, noise filtering, and estimation of unmeasured states. This is the work of the DP observer. Its abilities can be summarized in three main bullet points, as stated in (Sørensen, 2018, ch.7):

- **Estimate non-measured states**
  For many systems it can be difficult to apply all states in a feedback loop. The states can be unavailable as the system may be too complicated or sensor measurements may be unattainable. The state observer is then applied to estimate the inaccessible states which then can be used in the state feedback.

- **Perform *Dead Reckoning* during signal loss**
  It is crucial to have reliable and redundant systems. It is well known that at one point during its life cycle. Several regulations and recommended practises regarding redundancy already exist within the maritime industry. However, by applying a model based state observer, the signal may be replaced until the original sensor is back and running. This procedure, relaying solely on previous estimates in the contemporary state calculation, is known as dead reckoning, and will have an increasing cumulative error.
- **Filter out wave-frequency in the feedback loop**
  The vessel dynamic consist of low-frequency and wave frequency motion (WF). The WF motion will have a significantly larger frequency than the LF motion, which the propulsion system often is unable to provide sufficient power for compensation. Thus, to ensure a more sensible fuel consumption, the state observers are designed such that WF motion can be filtered out in the feedback loop.

Some of the most widely recognized observers, and state-of-the-art within marine DP systems, are Kalman Filter-based observers. Early research on applying the Kalman filter in DP systems were conducted by Grimble et al. (1980), as well as Fung and Grimble (1983). The Extended Kalman Filter (EKF) was introduced by J. G. Balchen et al. (1976), and further developed by Saelid et al. (1983). Fossen and Strand (1999) presented a nonlinear (passive) observer (NLO), with extensive experimental results, demonstrating strong evidence of its effectiveness. Remarkably, the NLO exhibits structural similarities to the Linear Time-Varying Kalman Filter (LTV-KF), as their structure can be seen as identical during steady state operations (Værnø, Skjetne et al., 2019).

In this thesis, both the EKF and the LTV-KF will be introduced. The EKF can be described as an nonlinear augmentation of the traditional Kalman Filter, and is an efficient recursive state estimator, capable of predicting the states of a system based on noisy measurements and a system model. It does, however, require that a linearization of the system is done around the current state estimate at each time step. This is not the case for the LTV-KF, which assumes a linear system. The nonlinearities in the rotation matrix is then overcome by considering the matrix as a known, time-varying signal, provided by the measured heading angle. This approach eliminates the need to calculate the Jacobian matrix of the system in the linearization process, leading to a simpler implementation, as well as a more computationally efficient algorithm.

Both algorithms can be divided into two separate blocks, the *predictor* and the *corrector*. The predictor calculates an *a priori* state estimate, $\overline{\boldsymbol{x}}$, purely based on the internal system model. The *a priori* estimate can be thought of as a best guess of the acutal state, prior to incorporation of any measurements. This is fed into the corrector, which generates the final *a posterior* estimate, $\widehat{\boldsymbol{x}}$, as a blending between the *a priori* estimate and the measurements. The two contributions are weighted by the Kalman gain $\mathbf{K}$, which balance the reliability of the sensors and the reliability of the internal model. The gain is designed to minimize the mean-square error of the estimate, and depends on the two tuning matrices $\mathbf{Q}$ and $\mathbf{R}$.

The discrete-time Extended Kalman Filter algorithm can be described with the following equations:

Nonlinear system:

$$\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{2.49a}$$
$$\mathbf{y} = \mathbf{C}\boldsymbol{x} + \boldsymbol{\epsilon} \tag{2.49b}$$

Initial Values:

$$\overline{\boldsymbol{x}}_{k=0} = \boldsymbol{x}_0 \tag{2.50a}$$
$$\overline{\mathbf{P}}_{k=0} = \mathrm{E}[(\boldsymbol{x}(0) - \widehat{\boldsymbol{x}}(0))(\boldsymbol{x}(0) - \widehat{\boldsymbol{x}}(0))^T] = \mathbf{P}_0 \tag{2.50b}$$

Corrector:

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}^T \left[\mathbf{H}\overline{\mathbf{P}}_k \mathbf{H}^T + \mathbf{R}\right]^{-1} \tag{2.51a}$$

$$\widehat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\,\overline{\mathbf{P}}_k\,(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \tag{2.51b}$$

$$\widehat{\boldsymbol{x}}_k = \overline{\boldsymbol{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\overline{\boldsymbol{x}}_k) \tag{2.51c}$$

Predictor:

$$\overline{\mathbf{P}}_{k+1} = \boldsymbol{\Phi}_k \widehat{\mathbf{P}}_k \boldsymbol{\Phi}_k^T + \boldsymbol{\Gamma}_k \mathbf{Q} \boldsymbol{\Gamma}_k^T \tag{2.52a}$$

$$\overline{\boldsymbol{x}}_{k+1} = \widehat{\boldsymbol{x}}_k + h\mathbf{f}(\widehat{\boldsymbol{x}}_k, \mathbf{u}_k, \mathbf{0}) \tag{2.52b}$$

The discrete system matrices $\boldsymbol{\Phi}$, $\boldsymbol{\Gamma}$ and $\mathbf{H}$ are as mentioned defined with the Jacobian matrices based on (2.49). The following linearization must thus be performed for every iteration.

$$\boldsymbol{\Phi} = \mathbf{I} + h\frac{\partial \mathbf{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k, \boldsymbol{w}=\mathbf{0}} \qquad \boldsymbol{\Gamma} = h\frac{\partial \mathbf{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k, \boldsymbol{w}=\mathbf{0}}, \qquad \mathbf{H} = \frac{\partial \mathbf{y}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k} \tag{2.53}$$

For the LTV-KF, however, (2.49a) is modified according to $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{w}$. The discrete system matrices in (2.53) are, consequently, simplified as

$$\boldsymbol{\Phi} = \mathbf{I} + h\boldsymbol{A}(t_k), \qquad \boldsymbol{\Gamma} = h\boldsymbol{E}, \qquad \mathbf{H} = \boldsymbol{C}. \tag{2.54}$$

It is important to emphasize that the EKF algorithm presented above is a state observer that utilize linearization. This results in the loss of several advantages associated with the LTV-KF regarding stability and optimality. *The separation principle* states that different parts of a control system can be analyzed independently regarding stability in order to obtain global asymptotic stability. One of the assumptions in the derivation of this principle is however that the system is considered linear and time-independent, which not is the case in the EKF described above. Nonetheless, as illustrated by Loria et al. (2000), the separation principle can be applied for nonlinear ship models, as the trigonometric functions in the rotation matrix only introduce small nonlinearities. Yet another disadvantage with Kalman Filters is that the estimated covariance matrix $\widehat{\mathbf{P}}$, often tend to underestimate the actual covariance matrix. Care must therefore be taken regarding covariance blow-up and instability (Fossen, 2022). That being said, both the EKF and the LTV-KF are widely popular in maritime navigation systems and is, as mentioned, considered state-of-the-art, yielding great performance.

2.4.8 DP control law

The control law naturally plays a critical part of the DP system in (2.9). The selection of control algorithm will therefore be decisive for the performance of the complete DP system. DP systems are often considered as *multiple-input and multiple-output* (MIMO), and generate a desired control force in all 3 horizontal DOFs to fulfill the DP objective, presented in subsection 2.4.1.

The control law in a DP system often consists of a feedback part, and a feedforward part, that is

$$\boldsymbol{\tau}_{ctrl} = \boldsymbol{\tau}_{FB} + \boldsymbol{\tau}_{FF}. \tag{2.55}$$

The feedforward term in (2.55) is responsible of suppressing wind loads, based on wind speed measurements and wind models. This term can be said to be optional in a DP controller, as a bias rejection term in the feedback control can compensate for slowly varying wind forces as well. That being said, the feedforward component responds quicker, as an integrator in the feedback loop can require several minutes to remove large wind components (Fossen, 2021).

The feedback component in (2.55) can be set to a nominal PD controller, augmented with a bias compensation term, as

$$\boldsymbol{\tau}_{FB} = -\boldsymbol{\tau}_{nPD} - \boldsymbol{\tau}_{BC}. \tag{2.56}$$

The last term in (2.56) serves to compensate for slowly-varying and constant external loads, along with unmodelled dynamics. Its inclusion is therefore essential for achieving the control objective, as a steady state error will be seen if these loads are not compensated for properly. One of the most frequently used compensation techniques is by the use of integral action. By assuming no feedforward components are implemented, the DP control law can then be written as

$$\boldsymbol{\tau}_{ctrl} = -\mathbf{K}_p \mathbf{R}^T(\psi)\widetilde{\boldsymbol{\eta}} - \mathbf{K}_d \widetilde{\boldsymbol{\nu}} - \mathbf{K}_i \mathbf{R}^T(\psi) \int_0^t \widetilde{\boldsymbol{\eta}}(\tau)\mathrm{d}\tau. \tag{2.57}$$

$\mathbf{K}_p$, $\mathbf{K}_d$ and $\mathbf{K}_i \in \mathbb{R}^{3x3}$ are positive tuning gains, normally set to diagonal matrices, which for instance can be tuned according to the MIMO nonlinear PID pole-placement algorithm in Fossen (2022). Alternatively, trial-and-error can also be used, although this can be a complicated process when dealing with MIMO systems. Fossen (2022) also proves the stability of the control law in (2.57) by a Lyapunov stability analysis.

An alternative strategy for handling disturbances, as a substitute to the integral action in (2.57), can be to include them in an internal model while aiming for asymptotic output tracking. If the unknown disturbance can be generated using a predefined model, the CDM can incorporate this model to improve both output tracking and disturbance rejection. This concept is known as the *internal model principle* (Skjetne, 2022).

The PID controller in (2.57) can be considered as one of the simpler control laws. That being said, numerous other control techniques that are more complex in nature can also be implemented in the control law construction. An robust controller can, for instance, be constructed by the inclusion of an adaptive control law, capable of adapting its parameters to variations in the process dynamics. Addintionally, J. Balchen et al. (1976) proposed a model-based control concept based on stochastic optimal control theory, while Agostinho et al. (2009) proposed nonlinear sliding mode control for the DP contol law. The reader is referred to Sørensen (2011), and the references therein, for a deeper explanation.

2.5 Low-speed simulation model

In order to perform credible and reliable numerical simulations of a vessel, it is crucial to employ a high-fidelity model. In this context, the simplified model in Section 2.4.6 can be augmented to more efficiently capture the realistic behavior of a vessel. This model is referred to as the simulation model, and provides a more detailed mathematical description of the real dynamics of the system. The required complexity of a simulation model depends on the purpose of the model. Different models must be developed for different operations and domains. The different domains can, according to Sørensen (2018), be separated into speed regimes as shown in Figure 2.11. Stationkeeping (DP) models can make assumptions such as zero speed and little nonlinear damping, while high-speed models often demand greater implementation of nonlinearities.

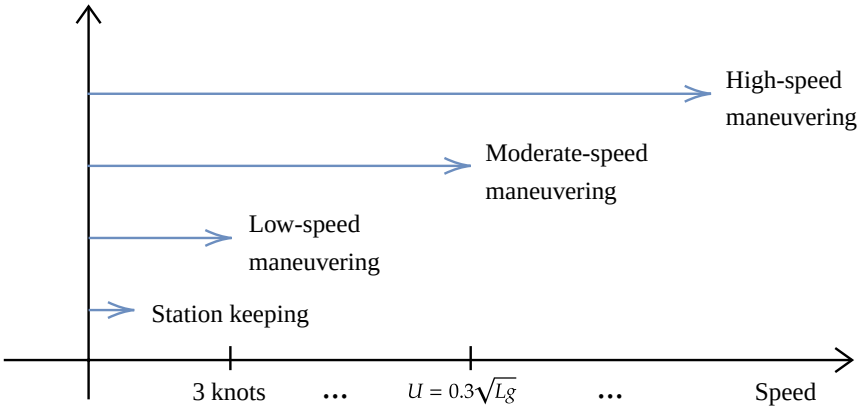


Figure 2.11: Speed regimes for vessels.

A general 6 DOFs simulation model can, according to Fossen (2021), be expressed as shown in (2.58). The kinematics of the simulation model will be the same as in Section 2.4.2, which for 6 DOFs could be described as in (2.58b), where $\mathbf{J}_\Theta(\boldsymbol{\eta})$ is given by Equation 2.36. The kinetics in (2.58a) arise from a generalized and augmented CDM from (2.48d). This equation introduce nonlinear damping, and Coriolis and centripetal forces due to rotation and restoring forces in the vertical plane. External forces from waves and wind are included instead of letting a bias term represent these.

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}_v(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}_p\boldsymbol{\nu}_r + \mathbf{G}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{wind} \tag{2.58a}$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.58b}$$

where

- $\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{A} \in \mathbb{R}^{6x6}$, are the rigid body and added mass inertia matrix.
- $\boldsymbol{C} = \boldsymbol{C}_{RB} + \boldsymbol{C}_A \in \mathbb{R}^{6x6}$ are the Coriolis and Centripetal matrix of the vessel due to its rigid body and added mass. The matrices occur as a consequence of evaluating Newton's law of motion in the rotating body-fixed frame.
- $\mathbf{D}_p, \mathbf{D}_v \in \mathbb{R}^{6x6}$ are the potential and viscous damping matrices of the vessel, and consists of both a linear and a nonlinear term.
- $\mathbf{G}(\boldsymbol{\eta}) \in \mathbb{R}^{6x6}$ is the hydrostatic restoring matrix and is a consequence of the vessel's point of equilibrium in the water.
- $\boldsymbol{\tau}_{ctrl}, \boldsymbol{\tau}_{wave}, \boldsymbol{\tau}_{wind} \in \mathbb{R}^6$ consist of thruster forces and external forces due to waves and wind acting on the vessel. Current forces are interpreted in the relative velocity term $\boldsymbol{\nu}_r$ and therefore not included on the right hand side of the equation. The wave forces consist of both first and second order wave loads, while the sum-frequency loads are neglected as they yield little impact on DP vessels.

The damping is in general one of the most complex and challenging terms in the mathematical simulation model in (2.58a). Hydrodynamic damping is said mainly to be caused by effects due to potential damping and viscous damping. The potential damping is often calculated by the means of a hydrodynamic program built on potential theory, and consequently, viscous damping forces must be added separately. The viscous damping is mainly caused by skin friction, wave drift damping, lift forces and vortex shedding (Fossen, 2021), even though other viscous effects also may be significant. Common for all these forces is that they can be challenging to model. The influence of the different effects are also difficult to separate, and it can therefore be convenient to divide the total damping term into a linear and a nonlinear damping matrix, as seen in (2.59). The linear damping matrix will in general be important for low-speed maneuvering and station keeping, while the nonlinear damping term will

dominate at higher speed operations, as seen in Figure 2.12. It is often necessary to include both terms in the vessel model to obtain satisfactory results.

$$\mathbf{D}_{tot}(\boldsymbol{\nu}_r) = \mathbf{D}_l + \mathbf{D}_{nl}(\boldsymbol{\nu}_r) \tag{2.59}$$

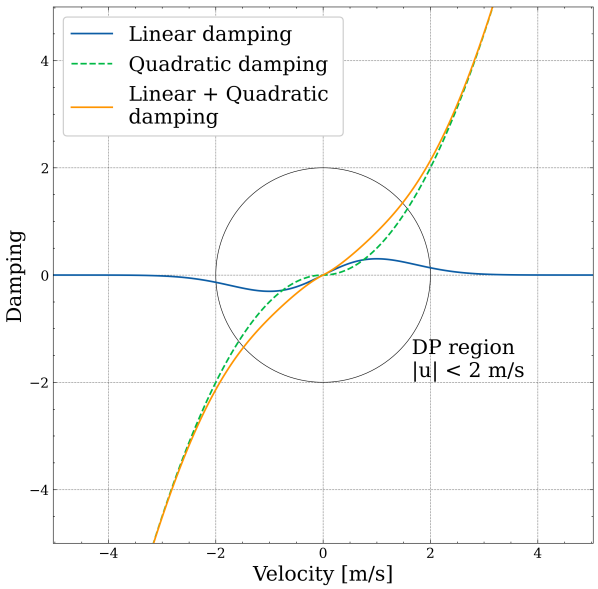


Figure 2.12: The influence of linear and nonlinear damping depending on the speed of the vessel. Courtesy of Fossen (2022).

As this thesis focuses on DP station keeping operations ($U \approx 0$), assumptions can be made to simplify (2.58a). First, the total damping force, consisting of linear potential damping and nonlinear viscous damping, can be combined and linearized around $U = 0$. The Coriolis and centripetal forces are highly velocity dependent and can thus be neglected for zero-speed operations. By assuming small values in the generalized position $\boldsymbol{\eta}$, the restoring force can also be represented by a linear matrix, even though the position must be rotated by the nonlinear rotation matrix. As mentioned in the introduction, wind loads will also be disregarded throughout the scope of this thesis. The simplified simulation model designed for zero-speed DP operations can then be expressed as

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\Theta)\boldsymbol{\nu} \tag{2.60a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} = -\mathbf{D}\boldsymbol{\nu}_r - \mathbf{G}\mathbf{J}(\Theta)^{-1}\boldsymbol{\eta} + \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{wave}. \tag{2.60b}$$

2.6 X-in-the-loop testing

Developing and testing complex systems such as DP control systems can be a time consuming process. X-in-the-loop testing is a set of different testing techniques which are used at different stages of the development cycle of the system which aims to step by step validate and ensure safety for the system. X-in-the loop tests are performed to verify the behavior and performance of the system and to validate the design and functionality of the system in a controlled and repeatable environment. It is much used in the automotive industry. There exist different traditional methods of X-in-the-loop tests. Model-in-the-loop, (MIL), simulator-in-the-loop, (SIL), and hardware-in-the-loop, (HIL), are three standard methods for simulation-based testing and verification which are often performed successively in the listed order (Smogeli, 2022). The idea of performing X-in-the-loop tests is to test the system for all steps in the development process to detecting failures early on. This will consequently reduce the time and cost of developing the final product. By resolving issues early on which potentially could be hazardous, it also increases safety.

Model-in-the-loop

MIL testing is useful to quickly verify a complex system in the early stages of the system development (Park et al., 2020). In MIL testing, the full system, being the control system, the environment and the physical equipment is mathematically modeled and simulated. As the software is emulated, there is no relation to the real hardware (Smogeli, 2022). As the tests are executed completely virtually, the can easily be repeated for a range of different conditions.

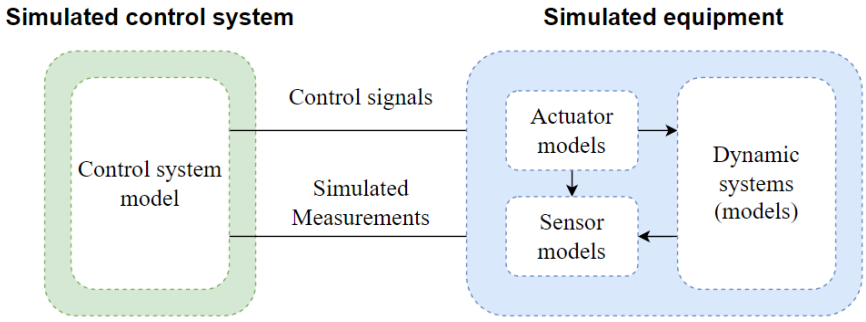


Figure 2.13: Concept of model-in-the-loop. In courtesy of Smogeli (2022).

Software-in-the-loop

In SIL testing, the real software is tested with virtualized hardware in a simulated environment. These software components should be fit to be embedded in the physical control system (Smogeli, 2022). SIL allows developers to isolate and test the software in a controlled environment to detect errors related to system functioning.

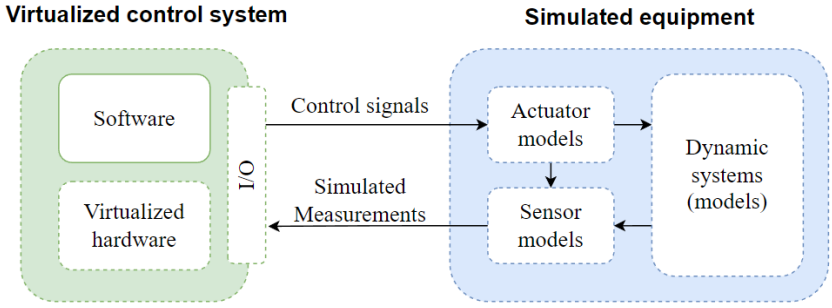


Figure 2.14: Concept of simualtor-in-the-loop. In courtesy of Smogeli (2022).

Hardware-in-the-loop

In HIL testing, the controller on implemented in a physical calculator running on a real-time simulation platform emulating the environment. This means that both the hardware and software are tested. HIL is typically used in the later stages of the system development to validate the integration of the control system and to efficiently test the control system functionality and performance (Skjetne & Egeland, 2006).

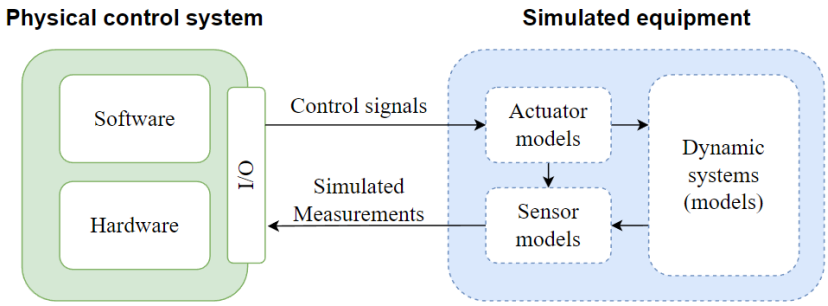


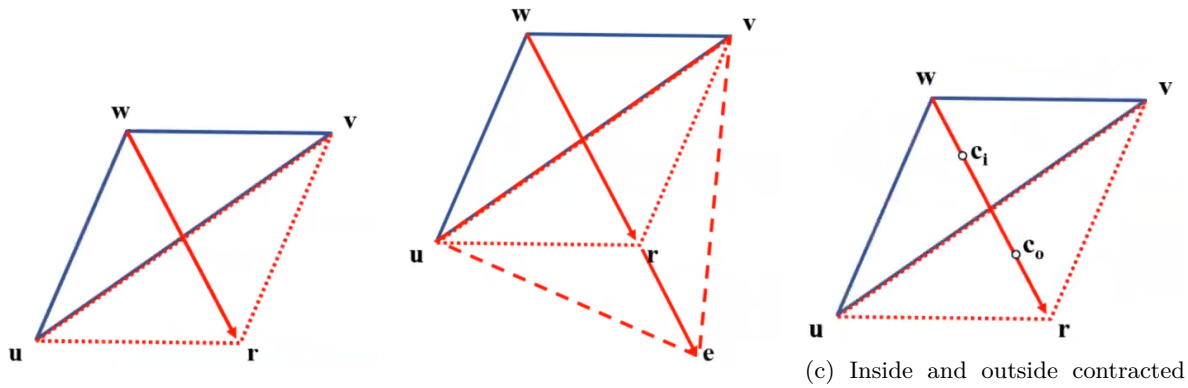
Figure 2.15: Concept of hardware-in-the-loop. In courtesy of Smogeli (2022).

2.7 Derivative-free optimizion techniques

Derivative-free optimization, also called zeroth-order optimization, is a meta-heuristic optimization technique which can be used to find the maximum or minimum of a function without calculating its gradients. This is a particularly useful approach when the objective function only is available as an output from a set of complex equations and little to no information about the system, or its derivatives, are known. This can be due to the system being highly nonlinear, or if several local minima exist. DFO is also practical when the calculation of the function derivatives are becoming too time consuming. Such problems may occur both in experimental methods and stochastic simulation, and DFO can then be a highly advantageous optimization technique (Audet & Hare, 2017).

The development of the modern, gradient-based methods can, in many ways, be said to have come further than the DFO methods. Thus, given that gradient information for an optimization problem is available, reliable, and obtainable, DFO will almost never outperform gradient-based algorithms (Audet & Hare, 2017). Nonetheless, DFO methods are often considered more robust and can deal with a wider range of optimization problems. They can be particularly useful when not expecting the method to find the optimal solution, but rather a solution that is acceptable within a given computational time limit and tolerance level. Consequently, DFO methods are widely used in several separate field of studies and continue to be an area of research within optimization and computer science (Nocedal & Wright, 2006).

Countless different derivative-free optimization algorithms exist, several of which are presented and investigated in the master's thesis of Løvås (2019). The downhill simplex method of Nelder and Mead is one of the simpler examples, which tracks $n+1$ points for each step in the algorithm, where n is the dimension of the objective. The convex set forms a simplex, hence the name of the algorithm (Nocedal & Wright, 2006). For a two-dimensional objective, the simplex will be a triangle. Given the simplex, the optimization algorithm aims to remove the point with the worst objective function value, with a point with a better value. This is done by reflecting or expanding the simplex through the centroid of the remaining points. If no point within the simplex yields a better cost, the point with the best value is chosen and the algorithm proceeds by shrinking the simplex. These algorithm steps is presented in Figure 2.16, according to Chen (2021). Here, $\mathbf{w}$ is the point with the worst function value. The Nelder-Mead method have a vast amount of applications and is used in several different field of studies, including engineering design, finance, and machine learning.



(a) Reflectinged point $\mathbf{r}$. (b) Extending point $\mathbf{e}$. (c) Inside and outside contracted points $\mathbf{c}_i$ and $\boldsymbol{c}_\mathbf{0}$.

Figure 2.16: Steps in the Nelder-Mead optimization algorithm. In courtesy of Chen (2021).

2.7.1 Derivative-free optimization for controller tuning

DFO algorithms are particularly effective for controller tuning purposes. These optimization methods have the potential to outperform manual tuning performed by a human operator, and offer a significantly more time-efficient approach compared to extensive trial-and-error techniques. For these cases, the tuning parameters, being the controller gains, are chosen as optimization variables. A suitable cost function is selected, and utilized to evaluate the performance of a full simulation, where an initial guess for gains is specified. The DFO algorithm is then constructed to select values close to the initial guess of gains, and evaluate if they provide a lower cost function.

An appropriate cost function is critical for achieving a well-performing regulator. In the context of a DP controller tuning process, factors such as trajectory tracking accuracy, wear and tear of actuators, and computational time, should all be taken into consideration in the selection process. In Løvås (2019), several cost functions were presented and discussed. Some of which are presented underneath, and will be addressed later on in this thesis. Common for most of these objective functions is the desire to minimize the pose error, $(\boldsymbol{\eta}(t) - \boldsymbol{\eta}_d(t))$. Velocities, $\boldsymbol{\nu}(t)$, and command forces, $\boldsymbol{\tau}(t)$, are also included to ensure realistic outputs of the controller. As all the mentioned parameters have different magnitudes and units, normalization or weighting techniques are often applied.

Integral of absolute error	IAE	$\int_0^T \lvert(\boldsymbol{\eta} - \boldsymbol{\eta}_d)\rvert \mathrm{d}t$
Integral of absolute error times work	IAEW	$\int_0^T \lvert(\boldsymbol{\eta} - \boldsymbol{\eta}_d)\rvert \mathrm{d}t \int_0^T \lvert\boldsymbol{\nu}^\top \boldsymbol{\tau}\rvert \mathrm{d}t$
Integral of absolute error and control	IAEC	$\int_0^T \lvert(\boldsymbol{\eta} - \boldsymbol{\eta}_d)\rvert \mathrm{d}t + \rho \int_0^T \lvert\boldsymbol{\tau}\rvert \mathrm{d}t$

When analyzing and comparing different controllers, care and considerations must be taken in regards to how well they are tuned. If manual tuning is performed, some might be tuned more optimal than others, and conclusions regarding performance can be drawn wrongfully. Using DFO in the tuning process offers the advantage of enabling a fairer comparison, as they are all tuned in accordance with the same optimization criteria. However, different controllers and observers have different tuning parameters, which can present a challenge related to comparison, even when using DFO tuning.

Another advantage with DFO tuning compared to manual tuning is that it can be included in an autotuning loop to enable autonomous tuning. This can be done for simulations, scaled experiments, and full-scale applications. Consequently, the operator is excluded from the tuning process altogether, which could enhance performance, and aid in a more efficient operation (Løvås, 2019).

Similarly to a DP controller tuning process, DFO can also be utilized in a DP observer tuning process. The cost functions presented above must then be modified to better fit the objectives of the observer,

related to the estimated states of the vessel. However, comparison of different observers is not a part of the scope of this thesis. Consequently, any DFO tuning process of the implemented observers will not be presented in detail.

2.8 MC-lab

Experimental platforms play a crucial role in validating simulation models and testing the response of a system. Although simulation models provide an approximation of real dynamics, it cannot be ensured that the simulated response accurately matches the actual system response. Therefore, the use of experimental platforms and testing is necessary to obtain valuable insights during the design phase of a DP vessel, and to confirm that the implemented system will successfully achieve its intended objective.

The marine cybernetics laboratory is a facility operated by the Department of Marine Technology, mainly used by students and employees at NTNU for teaching and research purposes. The lab has a relatively small basin equipped with a wave maker and a positioning system, making it well-suited for testing motion control systems. The dimension of the wave basin are $\mathrm{L} \times \mathrm{B} \times \mathrm{D} = 40\mathrm{m} \times 6.45\mathrm{m} \times 1.5\mathrm{m}$. The basin has a shallow end with a beach, and a deeper end equipped with a 6 m wide flap capable of generating both regular and irregular waves (NTNU, 2023). The capacity of the wave maker is listed in Table 2.1.

Table 2.1: Capacity of wave maker in the MC-lab.

Description	**Capacity**
Regular waves	H $<$ 0.25 m, T = 0.3 – 3 s
Irregular waves	Hs $<$ 0.15 m, T = 0.6 – 1.5 s
Available Spectrums	JONSWAP, PM, Bretschneider, ISSC, ITTC
Wave controller update rate	10 Hz
No. wave gauge on paddle	4
Stroke length on actuator	590 mm
Speed limit	1.2 m/s

The motion caption system in the MC-lab is called Qualisys. This can provide real-time measurements of the model vessels position and attitude by using Oqus cameras together with the Qualisys Track Manager software (NTNU, 2023). Qualisys works by detecting and calculating the position of markers installed on the model vessel. These markers are reflecting spheres installed at different positions with different heights on the model vessel. For Qualisys to be able to provide the 6DOF measurements of the model vessel, at least three of the spheres needs to be visible to the Ocus cameras. Consequently, the camera setup limits the operating area for testing motion control systems.

2.8.1 CyberShip Arctic Drillship

CyberShip Arctic Drillship (CSAD) is one of the model vessels hosted by the MC-lab and has been used for experimental testing for this thesis. CSAD is a model-scaled replica of the Inocean Cat I Arctic Drillship built in 2016. The purpose of developing the model was to facilitate for research on thruster-assisted position mooring. A summary of CSAD's main data is presented in Table 2.2.

Table 2.2: Main data for CSAD.

Parameter	**Value**	**Unit**
Length over all (LOA)	2.578	m
Beam	0.440	m
Depth	0.211	m
Design draft	0.133	m
Weight	127.92	kg
Scale	1:90	-

CSAD is equipped with six azimuth thrusters powered by six 12V 12Ah batteries, connected in parallel. The thrust is controlled by a set of electronic speed controllers connected to DC motors, while the thruster angles are controlled servos. Both the motors and the servos are controlled by pulse width modulation signals. For measurement purposes, CSAD is be equipped with multiple inertial measurement units (IMUs) containing both accelerometers and gyroscopes. Additionally, CSAD is equipped with a Raspberry Pi, which is a single-board computer responsible for controlling the hardware on the model vessel.

Figure 2.17: CSAD in the wave basin in the MC-lab.

2.8.2 ROS

Robot Operating System is an open source software used to interact with hardware in the MC-lab. It provides a framework for developing robots and offers powerful capabilities when combined with a Raspberry Pi. A great advantage when using ROS to develop robots is its ability to facilitate for seamless communication between different components in a robotic system, such as actuators, sensors and control systems. This communication is achieved by the use of *topics* and *messages*. The general architecture of ROS, presented in Figure 2.18, involves a ROS master and multiple ROS nodes (Open Robotics, 2022). The ROS master is the facilitator for a ROS process to run sufficiently. Its job is to register all the nodes, the executables, and set up peer-to-peer communication between them. Nodes can exchange messages by publishing to and subscribing from topics. The messages can be stored in bag files subscribing to the topic. Notably, the messages do not need to pass the ROS master. This decentralised architecture is a great strength with ROS as it make it possible for nodes to be run on separate networked computers. In the MC-lab, this multi-machine functionality is an advantageous approach if heavy computations are required, as the Raspberry Pi may face limitations in processing power (The Robotics Back-End, 2022).

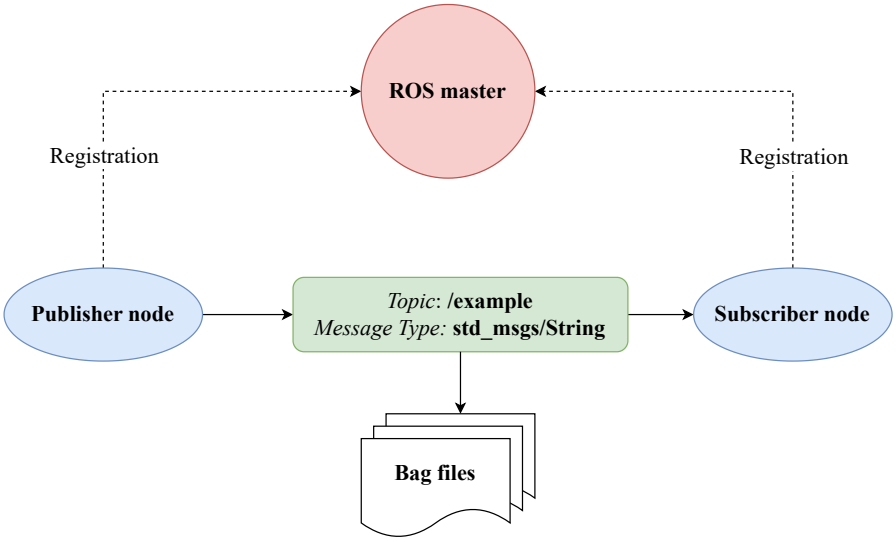


Figure 2.18: Graph model of ROS architecture.

As the nodes are isolated and the messages passed between them are standardized, software in the ROS ecosystem is language-independent. This design allows nodes built up by code written by different programming languages, such as Python and C, to run in conjunction. This feature makes ROS adaptable for software based on different languages and makes it accessible and user-friendly for people with diverse coding experience.

Chapter 3

Development of simulation platform

Parts of the following chapter is developed in collaboration with fellow M.Sc. student Jan-Erik Hygen, and is also presented in (Hygen, 2023). The same applies for all source code in the simulation model and its module.

During the work related to this Master's thesis, a comprehensive simulation platform has been developed, together with fellow student J. Hygen. The simulation platform includes a high fidelity simulation model, building upon work done in the preceding project thesis of Kongshaug and Mo (2022), based on previous work done by Brørby (2022). The model has further been improved during the spring of 2023. The model provides a comprehensive description of the vessel kinetics and kinematics, as well as its surroundings, and can be used for high-fidelity simulations when accurate and robust numerical performance is required. Additionally, the platform includes control system functionality, making it possible to carefully examine the functionality and performance of different DP control systems. This master's thesis is primarily based around the control system aspects of the simulation model, while Hygen (2023) has had more focus on wave-induced response. The simulation platform is made compatible with ROS, facilitating for real model tests of the implemented control systems.

The upcoming chapter describes the design and implementation of the complete simulation platform. The choices made during the design process will be discussed and the modules implemented will be thoroughly described. All implementations are done by the authors, together with Hygen (2023), if not specified otherwise.

The software is created with the intention of being installed and used, not only by the authors, but also by other external individuals. An open-source Python package, called *MCSimPython*, has been created and uploaded to the Python Package Index, PyPI (Hygen et al., 2023). It can be installed from PyPI using the package installer for Python, *pip*. It has been focused on writing general and user-friendly code, well-documented using Sphinx and ReadTheDocs. A detailed step-by-step guide on how to download the package and its documentation is provided in Appendix A as well as in the documentation.

3.1 Design overview

The implemented simulation platform can said to be module-based, using *modular programming*. This is a software design technique that decompose different parts of the code into independent blocks, such that each block includes all necessities to execute and test all its code locally and without interference from other parts of the code. Each module will then have a required input and given output, and is capable of executing only one aspect of the desired functionality of the model.

The source code is written using *object-oriented programming* (OOP), which is closely associated with

modular programming. OOP is based on the concepts of centralizing the code around objects and classes, rather than functions and logical statements. All important information is then privately contained within an object, restricting external access and modification. This approach is well suited for larger and complex programs that are actively updated and/or maintained. The latter state is highly applicable for NTNU-MSC repositories, and also one of the main reasons behind the selected implementation. The technique facilitate for collaborative development, enabling a cooperative design process. Additional benefits of OOP include code reusability, scalability and efficiency (Gillis, 2021).

The simulation model is also implemented using *data abstraction*. This is a design technique which involve implementing abstract classes, from which subclasses can inherit properties from. The abstract classes define general characteristics of an object as well as implementation methods, and works as a template for all its subclasses. This can be seen in the implemented code, where for instance an abstract class is defined for a general base wave spectrum, and a subclass is created for every different variant of the wave spectrum (PM, JONSWAP...). Functions and attributes which are equal for all wave spectra, such as spectral moments and wave realization, can then be defined in the abstract base spectrum class, which all subclasses then inherit. This make the code more perspicuous, and ensures that modifications to current spectra and implementation of new spectra will be easier to organize.

In the master's thesis by Brørby (2022), implementation issues of the simulation model, related to time-consumption was stated. An objective for the design process of the simulation model for this master's thesis has therefore been heavily influenced by optimizing runtime and generating efficient code. This is an important matter as the model is dependent on substantial calculations for every timestep. For Brørby (2022), this limited the fidelity of the model, as it restricted the number of wave components used in the simulations. As the objective was to generate a model with as high fidelity as possible, the runtime have been given considerable awareness throughout the entire programming process.

An overview of the structure of the simulation platform is displayed in Figure 3.1. The platform is as illustrated divided into four modules; the Sea state module, the Wave load module, the Vessel module, and the GNC module. The simulation platform requires some preprocessing before the online functionality can be run continuously during simulations. Each of the four modules presented in Figure 3.1, their input and output parameters, their implementation, as well as all communication amongst them, will be rigorously discussed in the upcoming sections.

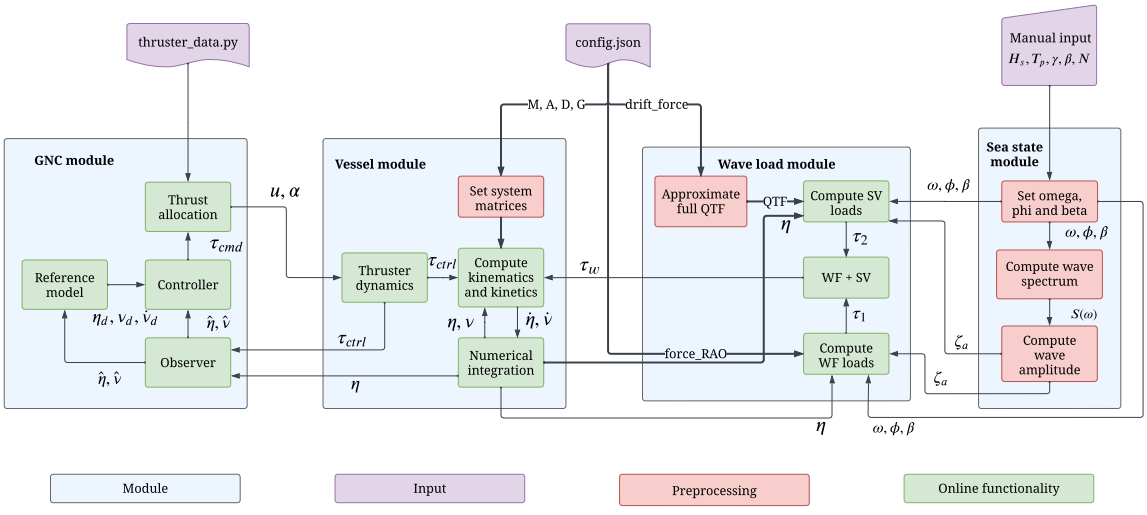


Figure 3.1: The structure of the simulation platform and its modules.

3.2 Implementation of modules

In this section, the implementation of each of the four modules presented in Figure 3.1, will be presented. The simulation platform is developed such that it requires some input to be initialized. These inputs are marked in purple in Figure 3.1. The external configuration file, used as input both in the Vessel module, and in the Wave load module, contains vessel model parameters and hydrodynamic transfer functions. In this Master's thesis, these parameters are extracted from the Veres' hydrodynamic vessel response program, ShipX, and have been used in the simulation model for CSAD. ShipX utilizes 2D potential strip theory to approximate the hydrodynamic terms and transfer functions. For most of the DOFs, linear potential theory can be a sufficient assumption. However, the roll motions are to a large degree determined by nonlinear effects, and hence, care should be taken when using the hydrodynamic data from Veres.

Veres produces four distinct result files:

- *.re1: Response transfer functions (RAOs).
- *.re2: Added resistance (mean drift forces in surge, sway and yaw).
- *.re7: Hydrodynamic coefficients.
- *.re8: Wave excitation force transfer functions.

Veres use a slightly different convention than as presented in this thesis. The coordinate system does not follow the standard body-frame representation, presented in Section 2.4.2, and the force RAOs are defined with a phase lead, rather than with a phase lag, as seen in Section 2.2. All hydrodynamic coefficients are also provided in the center of gravity, while the presented simulation model require the coefficients to be defined in the center of origin. A relevant rotation, according to Fossen (2021), ch.3, was, consequently, also necessary.

The simulation model is developed in Python. It was thus decided that the result files were to be converted to JSON-format configuration files. Given Python's built-in features to process JSON files, this was considered a preferred format.

3.2.1 Sea state module

The sea state module is responsible for defining the wave spectrum and the individual wave components in the simulated environment based on a set of user-defined input parameters. The functionality of the sea state module is summarized in Figure 3.2. The module offers the possibility to construct various wave spectra, including both 1D and 2D spectra. To ensure code reuse and modularity, an abstract base class called *BaseSpectrum* is created as the parent class for all 1D spectra. Each wave spectrum presented in Section 2.2.2 have been implemented as its own subclass inheriting from the corresponding parent class. For example, the MPM spectrum is a modified version of the PM spectrum, hence, it is defined as a subclass of the PM class. Furthermore, since the JONSWAP spectrum coincides with the MPM spectrum for $\gamma = 1$, the JONSWAP class is set to inherit from the MPM class.

The sea state module does not include current and wind loads. However, as will be demonstrated in Chapter 5, current is included in the simulations. This is due to the fact that it is included in the relative velocity state of the vessel model, and thus is defined separately for every simulation. The load could, however, be defined within the sea state module if more complex current models were to be implemented.

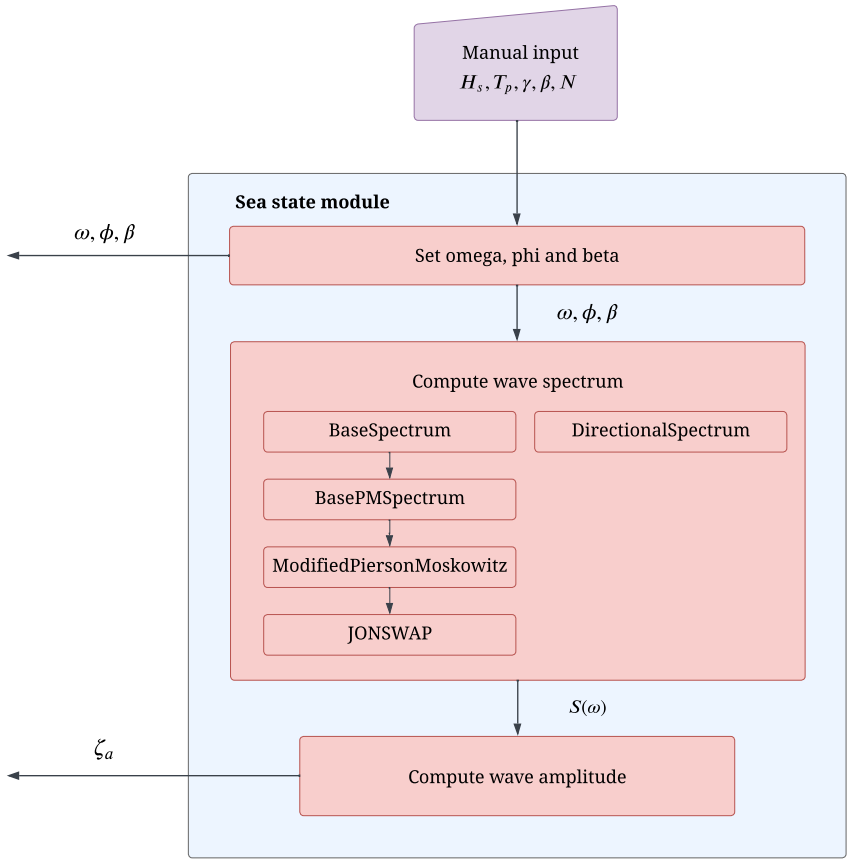


Figure 3.2: Sea state module.

The module also incorporates a directional spectra functionality through the *DirectionalSpectrum* base class. This class uses a predefined 1D spectrum and a spreading function to construct a 2D spectrum. By implementing a directional spectrum, the sea state module provides the possibility to generate multidirectional wavefields. A directional wave spectrum and its belonging multidirectional wave field generated from the sea state module can be seen in Figure 3.3.

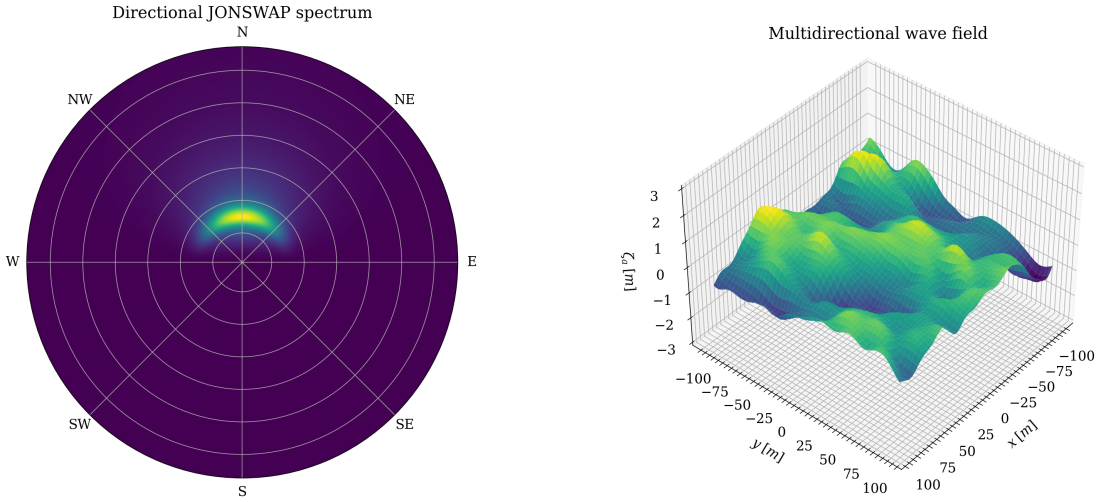


Figure 3.3: 2D wave spectrum and multidirectional wave field with $\theta_p = 0$ [°], $H_s = 2.0$ [m], and $T_p = 9.0$ [s].

As indicated by the color scheme in Figure 3.1, the sea state module only performs preprocessing tasks. Based on the defined wave spectrum, the module computes and outputs wave amplitudes, wave frequencies, random phases, and wave directions. These computed parameters are passed on to the wave load module where they are used in the calculations of first and second order wave loads.

3.2.2 Wave load module

The main objective of the wave loads module is to calculate the vessel-specific first and second order wave loads at each time-step t. The module takes the wave components, provided by the sea state module, as input, in addition to a vessel configuration file. The configuration file contains first order force RAOs and mean drift loads, calculated for a set of wave frequencies and wave angles in Veres. In the preprocessing stage of the simulation model, the wave load module extracts the necessary transfer functions from the Veres configuration file, based on the wave components defined in the sea state module. Figure 3.4 presents a overview of the module.

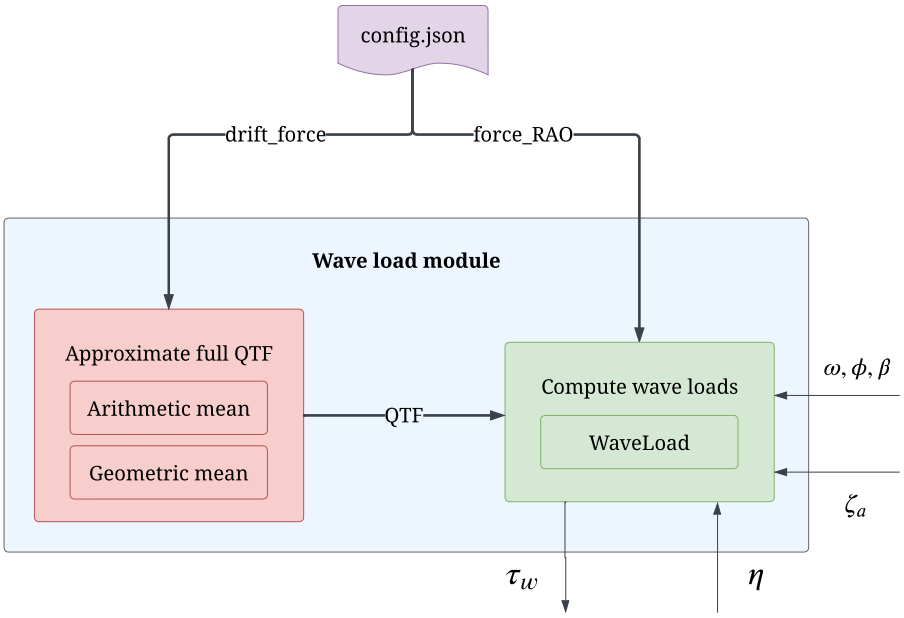


Figure 3.4: Wave load module.

First order wave loads, $\boldsymbol{\tau}^{(1\text{st})} \in \mathbb{R}^6$, are calculated at the instantaneous vessel position using a matrix formulation of (2.20), such that (3.1).

$$\boldsymbol{\tau}^{(1\text{st})} = \mathbf{F}(\boldsymbol{\omega}, \boldsymbol{\beta})\boldsymbol{\zeta}_a. \tag{3.1}$$

Here, $\boldsymbol{\zeta}_a \in \mathbb{R}^{N\times 1}$, is the vector of wave amplitudes computed in the sea state module, and $\mathbf{F} \in \mathbb{R}^{6\times N}$ is the force transfer function for each wave component. This is selected for the wave frequencies and incident angle of the waves. The elements in $\mathbf{F}(\boldsymbol{\omega}, \boldsymbol{\beta})$ are computed at each time step, in DOF i, and wave component j, according to

$$F_{ij} = \rho g \left|F_i(\omega_j, \beta_j)\right| \cos\Big(\omega_j t - k_j x \cos(\beta_{rel,j}) - k_j y \sin(\beta_{rel,j}) - \phi_j - \angle F_i(\omega_j, \beta_{rel,j})\Big), \tag{3.2}$$

where $\left|F_i(\omega_j, \beta_j)\right|$, and $\angle F_i(\omega_j, \beta_{rel,j})$, are the amplitude and phase of the force RAO for wave component j, respectively. k_j denotes the wave number for wave component j, found with either by the

infinite or finite depth dispersion relation, depending on the specified water depth. $\beta_{rel,j}$ indicates the relative incident wave angle for wave component j, calculated based on the heading of the vessel at the given timestep. The model is designed specifically for stationkeeping and low-speed maneuvering applications, which can justify the utilization of the wave frequency ω_j, rather than of the encounter frequency ω_e.

As only a limited set of force RAOs are provided by Veres, these are selected either by their closest index, or by a linear interpolation scheme. Both are implemented into the simulation model, although the last option is recommended to more accurately estimate the RAO amplitudes and phases. As mentioned, the sea state is assumed constant during simulations. Hence, the force RAOs for the different wave frequencies are selected in the preprocessing stage of the simulation. For wave frequencies outside the range of Veres, extrapolation is used to extract suitable RAOs. The force RAO then functions as a lookup table, from which $\left|F(\omega_j, \beta_j)\right|$, and $\angle F(\omega_j, \beta_{rel,j})$, are extracted. Although the wave frequencies are assumed constant during the simulation, this does not hold for the relative incident wave angles. A linear interpolation in the predefined force RAO matrix is therefore performed at every time timestep, making the wave module capable of modeling both uni- and multidirectional wave loads.

Second order drift and slowly-varying loads are computed based on the QTFs provided by Veres. The QTFs are calculated for every DOF ($i = 1, 2, ..., 6$), for M relative wave angles, and for $N \times N$ wave pair components, resulting in a tensor, $\mathbf{Q} \in \mathbb{R}^{6\times M\times N\times N}$. This is done in the preprocessing step of the simulator, to generate a lookup table for the QTFs, as illustrated in Figure 3.5. As all QTFs needed by the simulator are pre-calculated, the computational time is reduced, compared to if the QTFs were computed in real-time.

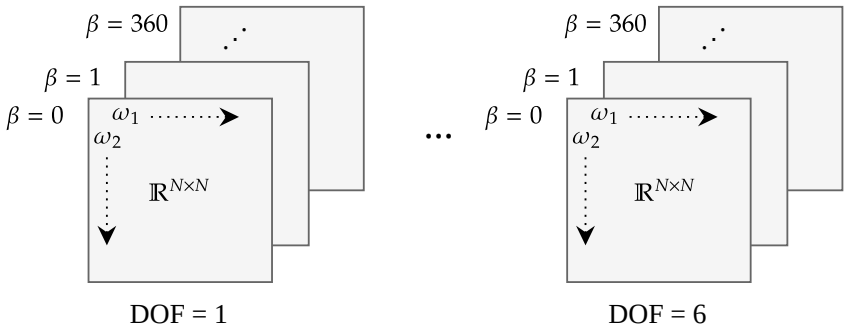
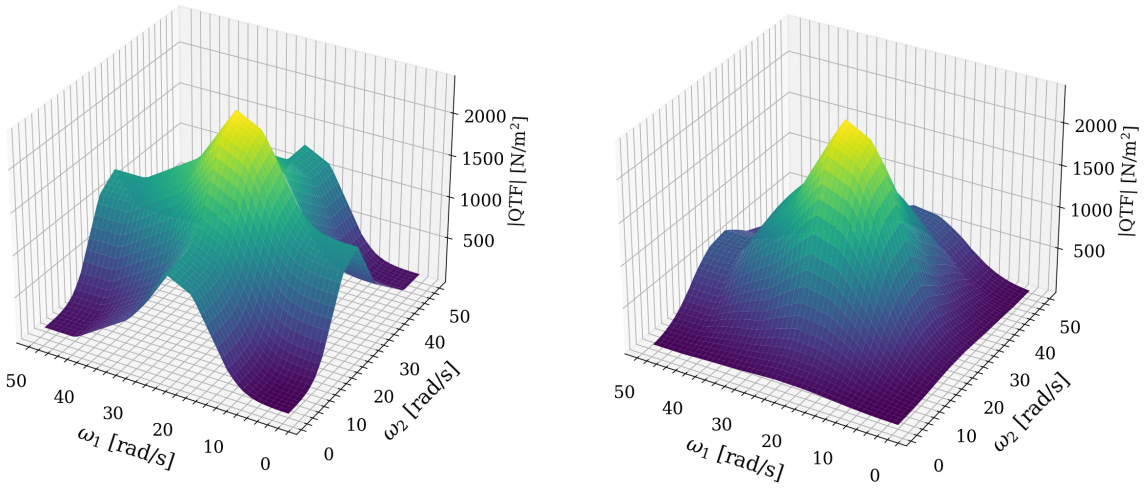


Figure 3.5: Generation of full QTF matrices in preprocessing.

As Veres only computes the mean drift load coefficients, the full QTF matrix, as presented in (2.23), must be approximated. This is done according to the theory in Section 2.2.4, using Newman's approximation. The default method of the module is to approximate the QTFs utilizing the arithmetic mean. However, the user can specify to rather use the geometric mean as an alternative approximation method. The difference is clearly illustrated in Figure 3.6. One can observe that the geometric mean method result in lower QTF values for the far-from diagonal terms, as should be expected. The user should be aware of the differences between the two methods when using the module, as they will influence the wave loads generated.

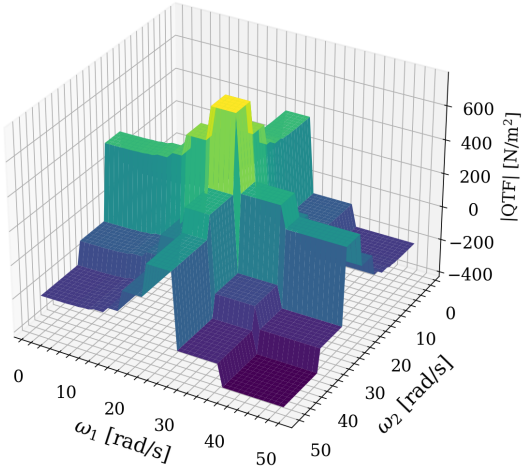


(a) Arithmetic mean approximation.

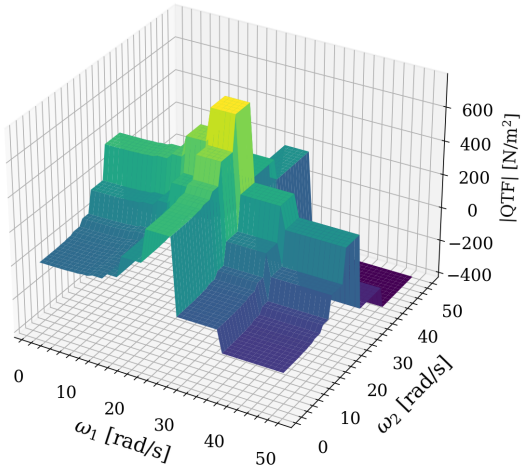
(b) Geometric mean approximation.

Figure 3.6: QTFs generated from the wave load module. Hs = 0.03 m, Tp = 1.5 s, N = 50, $\beta = 0$.

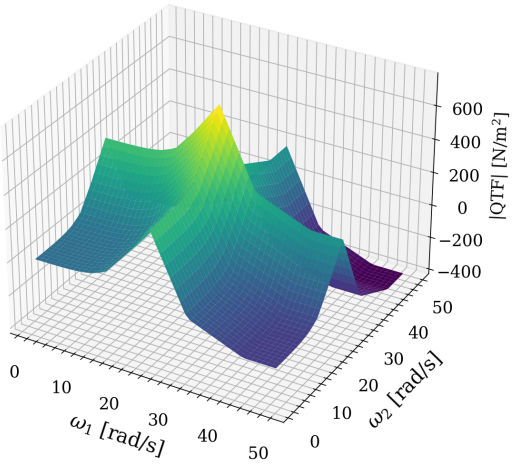
Veres only provides drift coefficients for a set of specific frequencies and headings. 47 frequencies indexes, given in the interval of [0.2094, 9.364] rad/s, as well as 36 headings, given in the range of [0, 2π] rad, are provided by Veres. To retrieve the correct QTFs during simulation, the wave load module offers the user to either interpolate over frequencies and/or headings, or to use the closest value. Figure 3.7 illustrates the difference between these methods. The QTFs presented are defined for CSAD's motions in surge with a heading of $\beta = 8°$, in a sea state with 50 wave components. One can observe significant differences between the different plots. The smoothness, especially, increases when interpolating with respect to frequency. It is recommended to at least use interpolation over the frequencies to generate the QTFs.



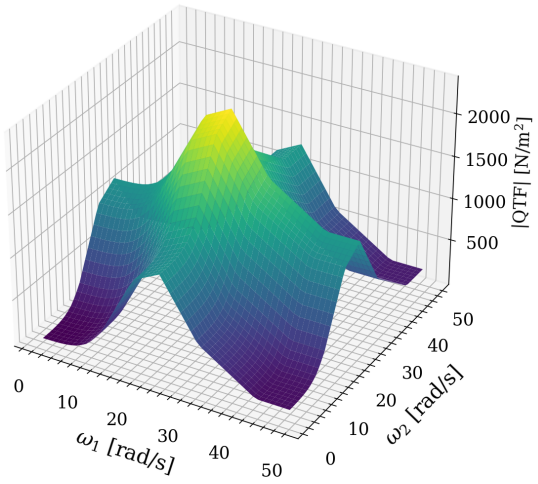
(a) Closest value for both headings and frequencies.



(b) Interpolated for headings, closest value for frequencies.



(c) Interpolated for frequencies, closest value for headings.



(d) Interpolated for both headings and frequencies.

Figure 3.7: QTF matrices retrieved with different methods from the simulation platform.

Second order wave loads are calculated at the origin of the NED-frame, $(N, E) = (0, 0)$. Only unidirectional waves are considered, as it is assumed that the slowly varying wave loads act in the mean wave direction of all wave components, if a multidirectional wave field is simulated. This assumption may not hold for all multidirectional sea states, but, as all simulations in this thesis are done in a unidirectional wave field, it will not be granted much more attention.

The slowly varying wave loads are given, based on (2.22). The double summation can easily be implemented using a double for loop in Python. However, the computational time of a double for loop increases by $\mathcal{O}(N^2)$, making it challenging to simulate in real-time, using a large number of wave components. Thus, a matrix formulation is introduced, in order to use the computationally more efficient Python library, `Numpy`, built in optimized, pre-compiled C code (Numpy, 2022). The final equation, in DOF m, and for a given relative wave angle β_{rel}, is then given as

$$\boldsymbol{\tau}_m^{(2\text{nd})} = \text{Re}\left\{\boldsymbol{\zeta}_a^\top (\mathbf{Q_m} \circ \exp[j(\boldsymbol{\Omega} t + \boldsymbol{\Phi})]\boldsymbol{\zeta}_a\right\}, \tag{3.3}$$

where $\mathbf{Q_m} \in \mathbb{R}^{N\times N}$ is the full QTF matrix in DOF m, for the interpolated relative wave angle β_{rel}.

$\boldsymbol{\zeta}_a \in \mathbb{R}^{N\times 1}$ consists of all wave amplitudes in the sea state, and $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$, and $\boldsymbol{\Phi} \in \mathbb{R}^{N\times N}$ are defined by two precalculated matrices as

$$\boldsymbol{\Omega} = \begin{bmatrix} \omega_1 - \omega_1 & \dots & \omega_1 - \omega_n \\ \vdots & \ddots & \vdots \\ \omega_n - \omega_1 & \dots & \omega_n - \omega_n \end{bmatrix} \tag{3.4a}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1 - \phi_1 & \dots & \phi_1 - \phi_n \\ \vdots & \ddots & \vdots \\ \phi_n - \phi_1 & \dots & \phi_n - \phi_n \end{bmatrix}. \tag{3.4b}$$

When the first and second order wave loads have been calculated, they are passed to the vessel simulation module as input to the equations of motion.

3.2.3 Vessel module

The vessel module is responsible for simulating the kinetics and kinematics of the vessel. An overview of the model's functionality is presented in Figure 3.8. In Section 2.5, it was discussed that different speed regimes require different models based on the operational context of the vessel. Some of the functionality is, however, common to all simulation models, such as the state vector. To accommodate this, the vessel module contains an abstract base class, *Vessel*, which provides the main functionality of the module. This base class can be inherited from, and extended, to add specific functionality for different types of vessels. Separate sub-classes have been created for each vessel, keeping the source code tidy and easy to extend. The source code in Hygen et al. (2023), include both 6 DOF DP models and 3 DOF maneuvering models for both CSAD and NTNU's research vessel, R/V Gunnerus (RVG). However, according to the scope of this master's thesis, the focus has been to develop the DP model for CSAD, and further examine its performance.

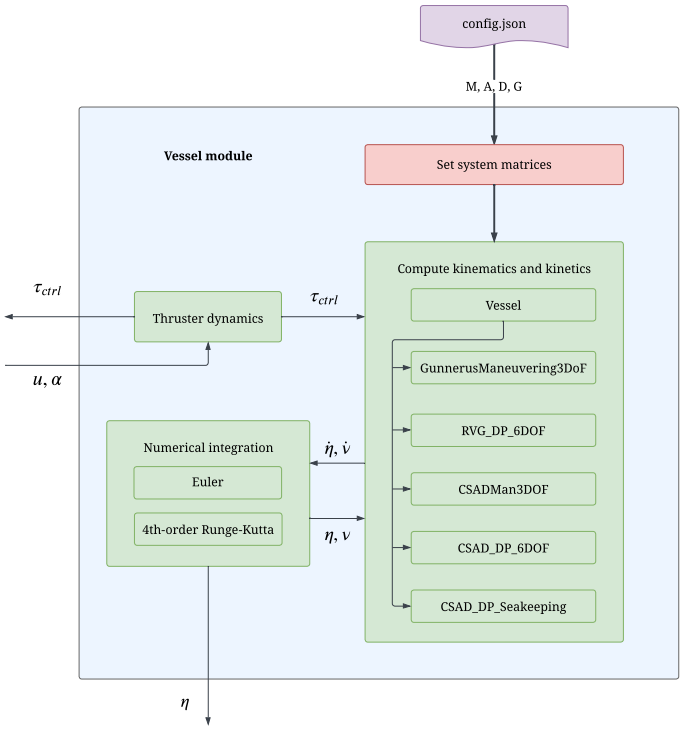


Figure 3.8: Vessel module.

Some assumptions regarding the low-speed 6DOF DP simulation model has already been stated in Section 2.5. It is assumed that the Coriolis and centripetal loads can be neglected for low-speed applications, that the damping is linear, and that the encounter frequency is equal to the wave frequency. These assumptions simplify the kinetics of the vessel. The final simplified DP simulation model is expressed in (2.60), and is restated as

$$\begin{aligned}\dot{\boldsymbol{\eta}} &= \mathbf{J}(\Theta)\boldsymbol{\nu} \\ \mathbf{M}\dot{\boldsymbol{\nu}} &= -\mathbf{D}\boldsymbol{\nu}_r - \mathbf{G}\mathbf{J}(\Theta)^{-1}\boldsymbol{\eta} + \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}^{(1\text{st})} + \boldsymbol{\tau}^{(2\text{nd})}.\end{aligned}$$

The hydrodynamic vessel parameters are defined in the configuration file and set as constant throughout the simulations. The frequency-dependent terms can then be set according to the desire of the user. As default, the coefficients use zero-frequency in the horizontal plane (DOF 1,2,6), and the peak-frequency in the vertical plane (DOF 3,4,5), as proposed by Fossen (2021).

Fossen also discuss using both fixed and frequency-dependent hydrodynamic coefficients, while introducing the concept of fluid memory effects and *Cummins equation*. The fluid memory effects are estimated from a state space identification of a retardation function. This is implemented in the wave load module, but has not been utilized in any vessel models in this master's thesis. It will, thus, not be discussed any further, and the reader is referred to (Hygen, 2023, ch.2) for a more in-depth derivation.

Two integration methods are implemented to the simulation package, for integrating the state vectors, $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$. These are the forward Euler integration and the 4th-order Runge-Kutta integration, and the user can state his or hers preferred method when initializing the vessel object.

Thruster dynamics

The actuator's dynamics directly affect the performance of a DP system. The systems ability to respond to environmental forces depends on the speed at which thrusters can change their thrust level and direction. Thrusters with a faster response time, naturally provides better control and stability, allowing the DP system to maintain its position more effectively. For a high-fidelity simulation model, it is important to take the thruster dynamics into consideration as it can limit the calculated command loads from the controller.

Thruster dynamics are implemented to the vessel module as a base class. The class object is initialized with input data from a vessel specific configuration file containing information about the thruster configuration, coefficients and specifics. The thruster configuration for CSAD can be seen presented in Figure 3.9, where all six thrusters are the same type of azimuth thrusters and the thruster spesificcs for each thruster are summarized in Table 3.1.

Table 3.1: Thruster specifics for CSAD. In courtesy of Bjørnø et al. (2017)

Parameter	**Value**	**Unit**
Max shaft speed	94.9	RPS
Power	0.868	W
Propeller diameter	0.03	m
Steering speed	114	deg/s
Max torque	0.0015	Nm
Max thrust	1.5	N

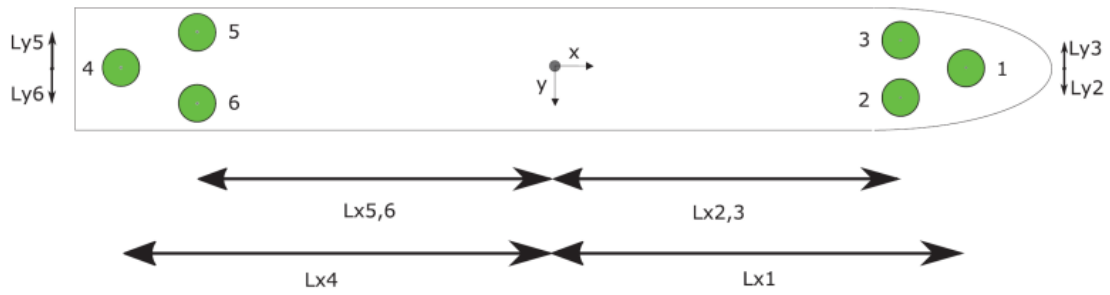


Figure 3.9: Thruster configuration for CSAD. In courtesy of Bjørnø et al. (2017).

The thruster dynamics gets its input commands, $\boldsymbol{\alpha}$ and $\boldsymbol{u}$, from the thrust allocation block. Its objective is then to calculate the actual thrust force, $\boldsymbol{\tau}_{ctrl}$, based on the natural restrictions of the thruster configuration. These restrictions include the likes of saturation limits and rate limits of the azimuth thrusters, ensuring that the output is equal to the resulting loads produced by the thrusters, as they would be for the real vessel.

In (2.38) in Section 2.4.4, a model stating the relation between the the actuator loads, $\boldsymbol{\tau}$, and the control signals $\boldsymbol{u}$, $\boldsymbol{\alpha}$, was stated. To be able to represent realistic thruster dynamics, it is also necessary to state the relation between the control signals and the shaft speed $\boldsymbol{n}$, of the thrusters, as this often needs to be rate-limited. This can be done, according to Fossen (2021), as

$$\boldsymbol{u} = \boldsymbol{n}|\boldsymbol{n}| \quad \Rightarrow \quad \boldsymbol{n} = \text{sgn}(\boldsymbol{u})\sqrt{|\boldsymbol{u}|}, \tag{3.5}$$

which, if inserted into (2.38), provides

$$\boldsymbol{\tau} = \mathbf{T}(\boldsymbol{\alpha})\mathbf{K}^{*}\boldsymbol{n}|\boldsymbol{n}|, \tag{3.6}$$

where $\mathbf{T}(\boldsymbol{\alpha})$ was described as the actual thrust configuration matrix, and $\mathbf{K}^{*}$ is the actual force coefficient matrix. In simulation aspects, the latter matrix can be set equal to the one in Section 2.4.4. For real-world vessels, however, these will only be given as an estimation, based on the specifications of the thrusters.

3.2.4 GNC Module

To expand the simulation platform, developed in Kongshaug and Mo (2022) to a complete DP simulator, the GNC module was added. The module made it possible to perform full end-to-end tests of the simulated system, and included reference models, observers, controllers, and thrust allocation. The objective of the module was to provide the vessel module with the required control input to satisfy the control objective. The theory behind the GNC module has been presented in Section 2.4.3, and this section will only briefly presents the different functionalities implemented. An overview of the GNC modul is presented in Figure 3.10.

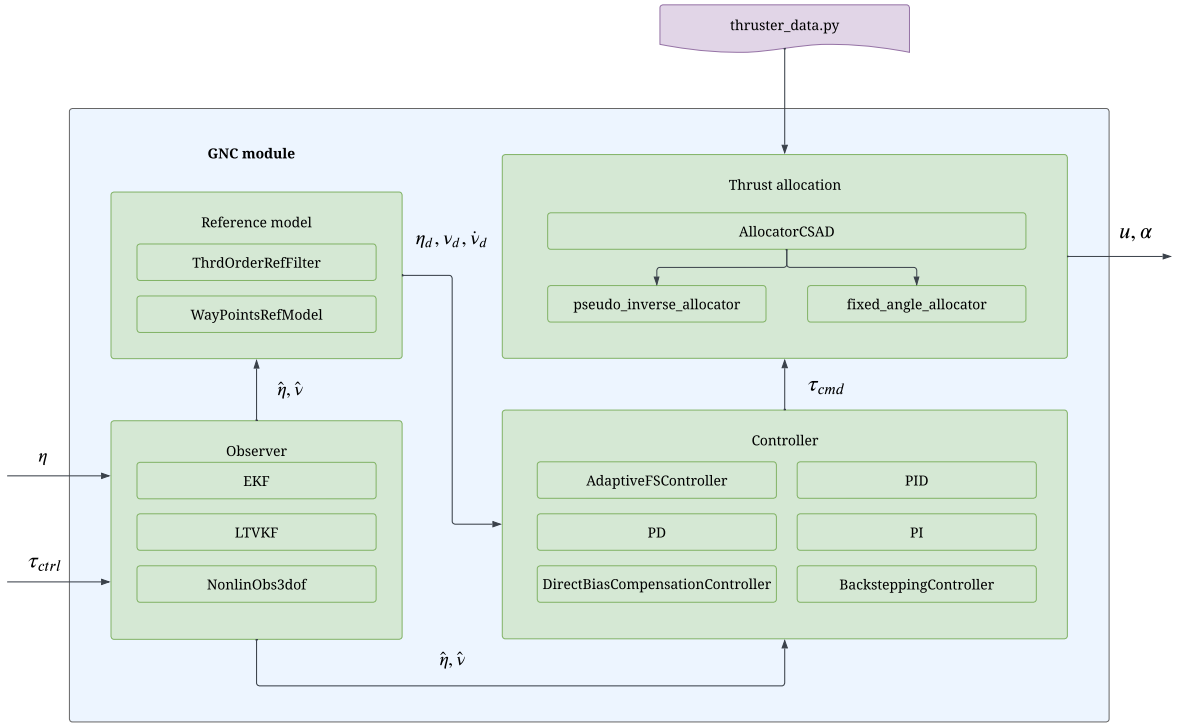


Figure 3.10: GNC module.

Navigation

The navigation receives the input to the GNC module ($\boldsymbol{\eta}$, $\boldsymbol{\tau}_{ctrl}$), and determines the full state vector of the vessel. Three different observers have been implemented, each as their own class. Two of which are variants of the Kalman filter from subsection 2.4.7, being the EKF and LTV-KF, and one nonlinear observer. The observer pass the estimated states to the control and the guidance system.

Guidance

The guidance system is represented by the *Reference model* block in Figure 3.10 which takes the estimated position and velocity from the observer and computes an optimal setpoint for the vessel. The desired position, velocity and acceleration are then sent to the controller. The simulation platform includes the implementation of both a third order reference filter and a hybrid path-parameterization model. Both are implemented as their own class with their own private functionality.

Control

The control block receives observer estimate and reference states as input, and calculates the necessary command forces and moments for the thrusters in order to achieve the control objective. A total of six controllers are present in the created toolbox, where four of which can be considered as well established and familiar controllers. These are the PD, PI, PID, and direct bias compensation controllers. Additionally, an adaptive controller and a backstepping controller also exist. The adaptive controller will, together with some of the more basic controllers, be utilized considerably in the upcoming simulation study, and are presented more in depth in Chapter 4. All controllers are implemented as their own class. A neural network has also been developed as an augmentation of the controller module. This is utilized in an internal neural network bias model, to be presented in Chapter 4

The control-part of the GNC module also consist of thrust allocation. The trust allocation block takes the command loads computed by the controller and maps them to actuator commands. There exist many different thrust allocation algorithms, which all share some common functionality. Hence, a base class has been implemented for CSAD (*AllocatorCSAD*). The base class defines abstract methods for the allocation problem and the allocation algorithm. These are defined in the subclasses, inheriting from the base class. One of the allocation algorithms implemented as a sub class is the pseudo inverse algorithm presented in Section 2.4.4. In addition, a fixed-angle allocator has been implemented, which shares the same logic as the pseudoinverse, with the additional implementation of constant azimuth angles. Saturation of the thrust loads is the only implemented constraint to the allocation algorithm. Ideally, aspects such as angular rates, and forbidden zones for the azimuth thrusters, should also have been implemented. This has, however, not been fully implemented in the toolbox, and can be regarded as an area that requires improvement. The thrust allocation block finally returns the actuator commands, which eventually serves as output of the GNC module.

To facilitate the use of the thruster allocation class, the module also includes implementation of a thruser base class. This defines the common features for a typicall thruster. Two sub-classes, inheriting from this, has been implements, one for creating tunnel thrusters and one for creating azimuth thrusters.

3.3 ROS interface

To enable testing and validation of the simulation model through physical experiments, it was necessary to ensure compatibility between the MCSimPython software and the hardware setup. This involved making the simulation platform ROS-compatible, which was achieved by incorporating a ROS wrapper. This section will present the implementation of the ROS wrapper and the architecture of the nodes and topics in the ROS environment.

The concept of wrapping in programming often refers to creating a module within another module. This can add extra functionalities, or as in this case, create a brand new interface. A ROS wrapper will create such a ROS interface by wrapping non-ROS code, as in MCSimPython, inside a ROS node. In this way, the original source code can be modified without changing the ROS interface. Additionally, this means that anyone familiar with ROS can take the program in use without having any knowledge about the functionality in the source code. Keeping the source code completely independent from ROS, also makes it easier to locate potential errors when troubleshooting. It will also make it possible for developers without knowledge of ROS to understand and modify the source code and use it for other purposes (The Robotics Back-End, 2023).

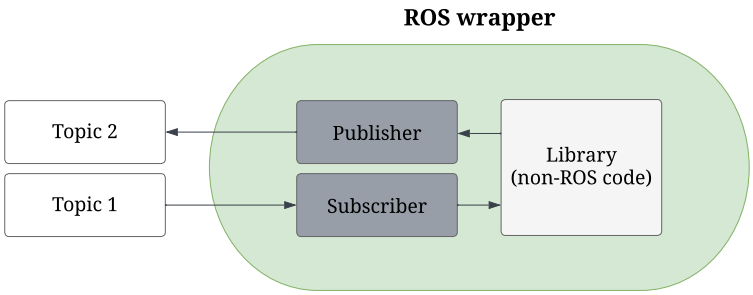
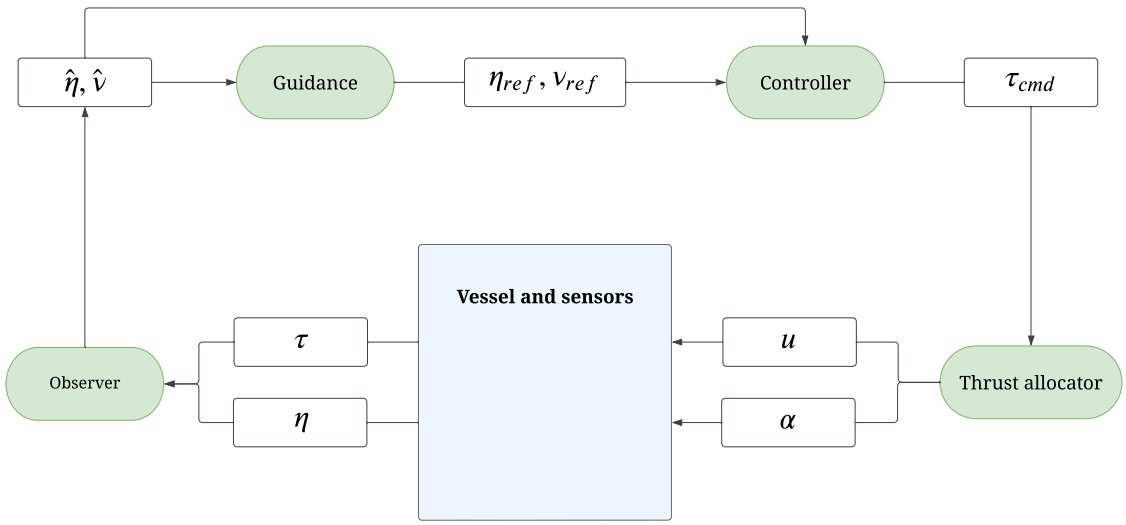


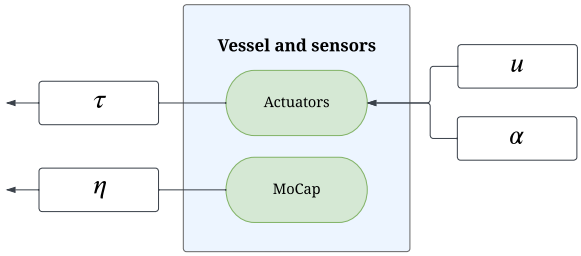
Figure 3.11: ROS wrapper

Figure 3.11 illustrates how a ROS wrapper is created around the functionality of the source code. As the figure illustrates, ROS features are added to enable communication. A total overview of the ROS environment implemented is illustrated in Figure 3.12. The nodes can publish and/or subscribe to topics, indicated by the arrows. As illustrated, the ROS environment can be used both by the simulator

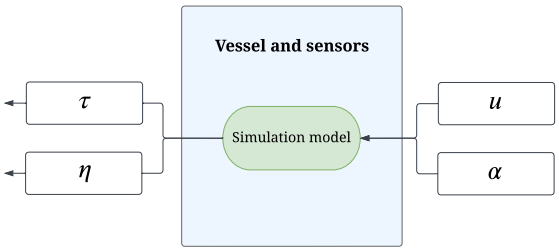
and by the hardware, depending on which nodes are defined in the ”Vessel and sensors” box.



(a) Overview of full ROS architecture.



(b) Vessel and sensor block - hardware.



(c) Vessel and sensor block - simulator.

Figure 3.12: ROS architecture.

3.4 Validation and result

An important aspect of creating a simulation model is to validate its performance. This ensures the model's performance and accuracy, and evaluates the simulation's capability of replicating a credible behavior. According to the test procedures in Section 2.6, this was initially done by conducting model-in-the-loop simulations. This aids in model validation, while using a emulated control system, as can be seen in Appendix B. 1. Further testing of the control system, utilizing both software-in-the-loop (Appendix B. 2), and hardware-in-the-loop (Appendix B. 3), was also conducted before any software could be said to be ready for experimental testing (Appendix B. 4).

For a single regular wave, the second order wave loads are expected constant and no slowly-varying component shall be observed. Figure 3.13 shows the second order wave loads in surge, sway, and yaw, acting on CSAD in head sea. It can be seen that the sway and yaw loads are zero, while a small constant drift load appear in surge. Furthermore, in Figure 3.14, two wave components with distinct frequencies are simulated. The slowly-varying wave loads then provide an oscillating force in surge, while still yielding no loads in sway and yaw. The force in surge can be seen to exhibits oscillations with a nonzero mean and a period of approximately 65 seconds. Finally, in Figure 3.15, three distinct wave components are included in the simulation. The second order wave loads acting on the vessel can now be seen oscillating with three frequencies.

The observations above strengthens the credibility of the implemented wave load module, as all findings

are according to what can be expected by the theory. A single wave component does only provide a mean load. Two components, however, introduce an oscillation in addition to the mean load. This oscillation should act with a frequency of $\omega_1 - \omega_2 = 0.1$ rad/s ≈ 0.016 Hz, which equals a period of 62.5 seconds, well within range of the initial observation. As for three wave components, two more frequencies are introduced into the oscillations. It is worth noticing that if the initial wave frequencies would have been evenly spaced, only two frequencies should be distinguished in the slowly varying loads, in addition to the mean load. It can be concluded that the number of wave components present in a system directly impacts the nature and complexity of the oscillations observed. It can also be noted that the wave load magnitude correlates with the total wave amplitude, being greater when the wave amplitude is large.

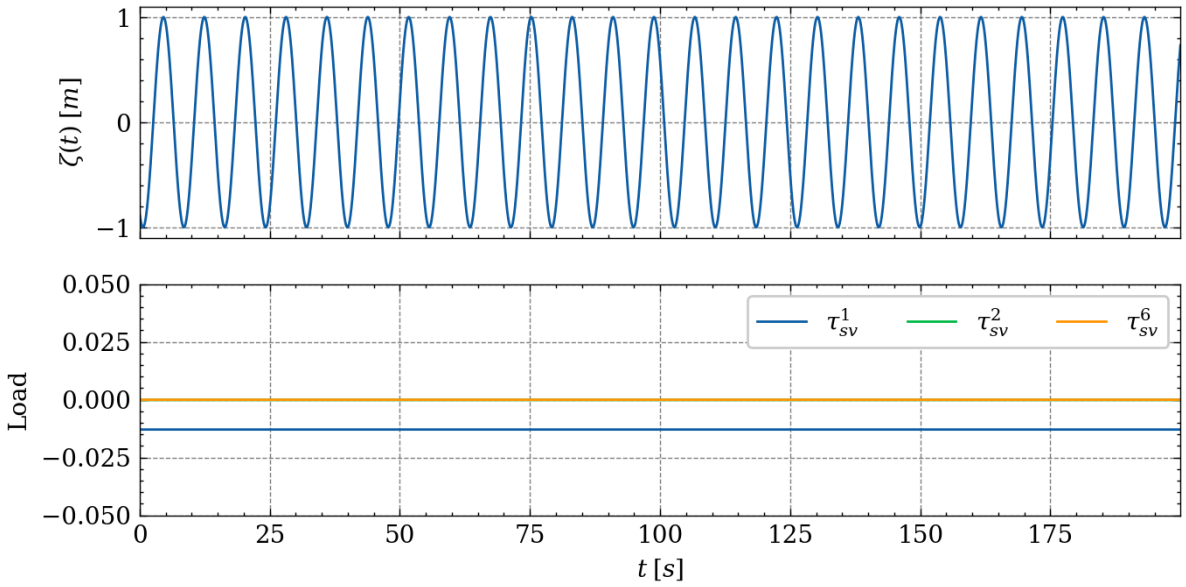


Figure 3.13: Second order wave loads for a single wave component with amplitude $\zeta = 1$ m and frequency $\omega = 0.8$ rad/s.

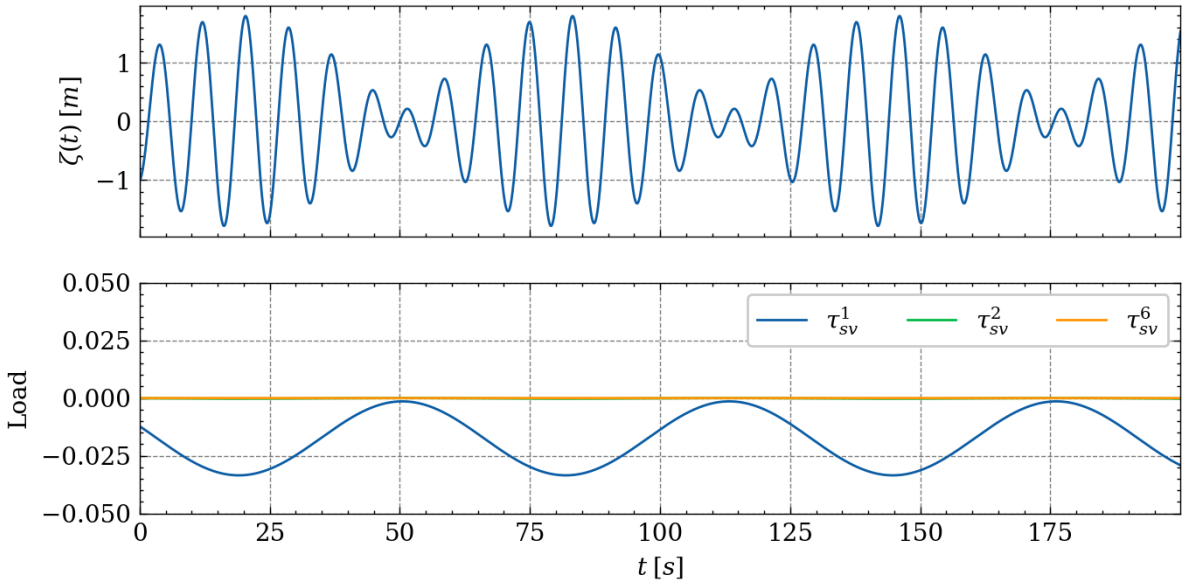


Figure 3.14: Second order wave loads for $N = 2$ wave components with amplitudes $[\zeta_1, \zeta_2] = [1,\ 0.8]$ m and frequency $[\omega_1, \omega_2] = [0.8,\ 0.7]$ rad/s.

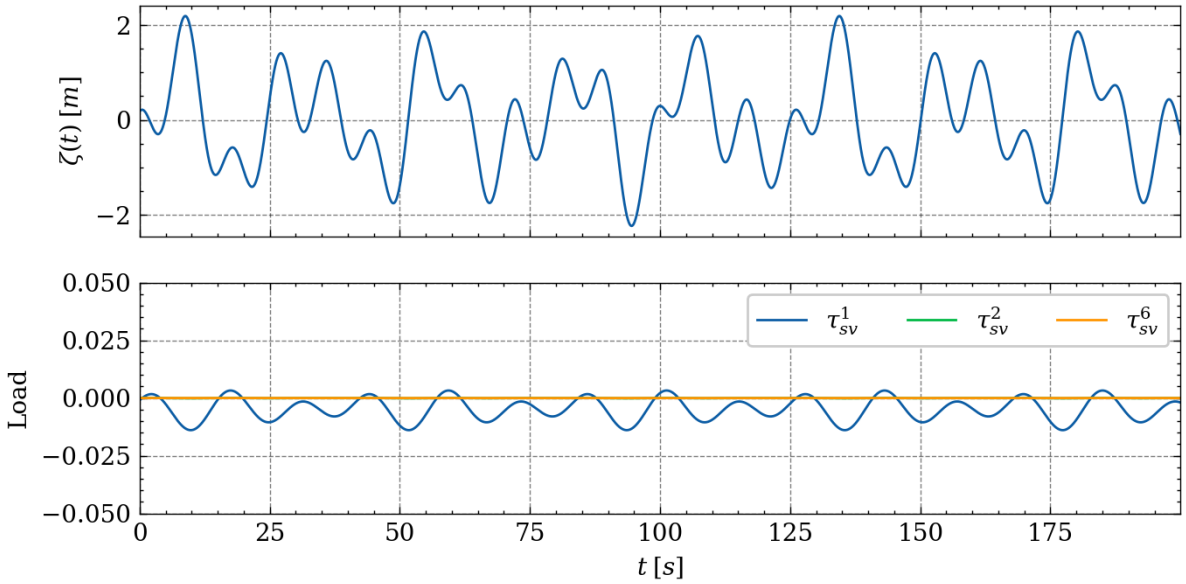
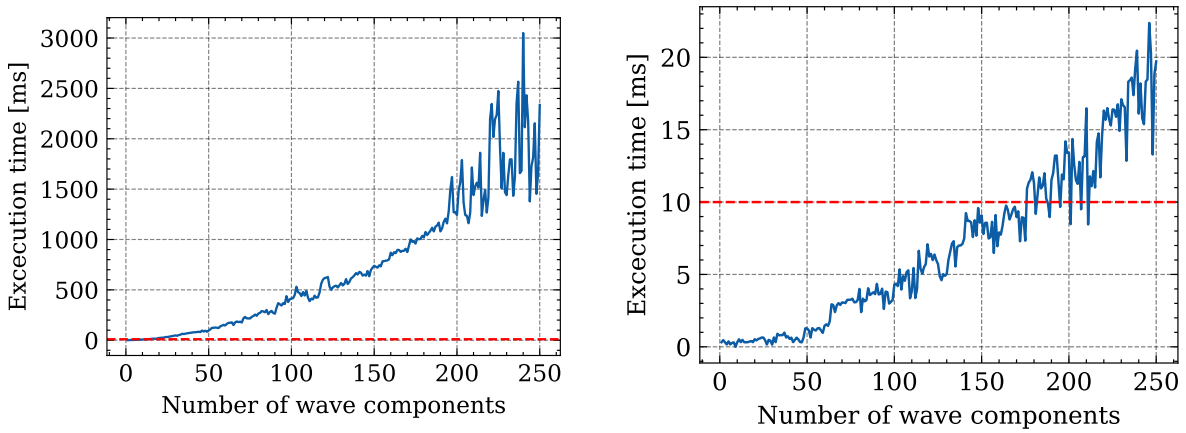


Figure 3.15: Second order wave loads for $N = 3$ wave components with amplitudes $[\zeta_1, \zeta_2, \zeta_3] = [0.5,\ 1.0,\ 0.8]$ m and frequency $[\omega_1, \omega_2, \omega_3] = [0.4,\ 0.25,\ 0.7]$ rad/s.

Given the significant amount of time and resources allocated to reduce the computational time of the proposed algorithm, it became imperative to validate the runtime performance of the wave load algorithm. To determine if the proposed algorithm ran faster than the double for-loop used by Brørby (2022), a computational speed test was performed. The test measured the average computational time of N = (1, 2, ..., n) wave components on a AMD Ryzen 7 3700U CPU with 16 GB RAM. For each distinct number of wave components, the computational time was averaged over 50 computations. The results of the test, shown in Figure 3.16, demonstrate that the matrix formulation in (3.3), runs significantly faster than the double for-loop. If defining the simulation time step to 0.01 seconds (10 ms), the double for-loop algorithm was restricted to running with 20 wave components. In contrast, the matrix formulation allowed for almost 200 wave components, yielding far more realistic simulations of the sea states.



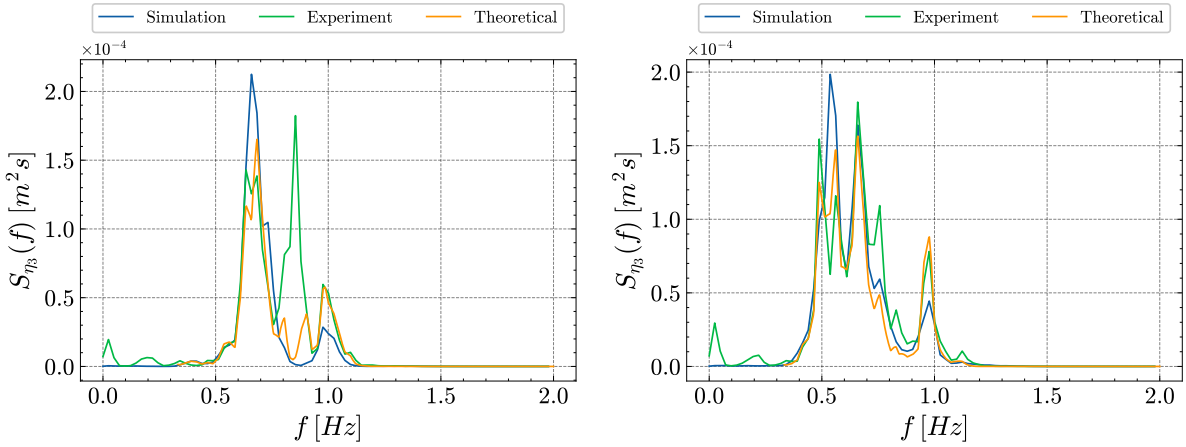
(a) Double for-loop of Brørby (2022).

(b) Matrix formulation of Hygen et al. (2023).

Figure 3.16: Comparison of the computational time for second order wave loads using the matrix formulation in (3.3), and a double for-loop. The dotted lines indicate the global time step used in the simulator (0.01 s).

In the period between Oct.-Nov. 2022, numerous tests with CSAD, were carried out in the MC-lab by PhD candidate Raphael Mounet. The sea states were defined with the JONSWAP spectrum, and

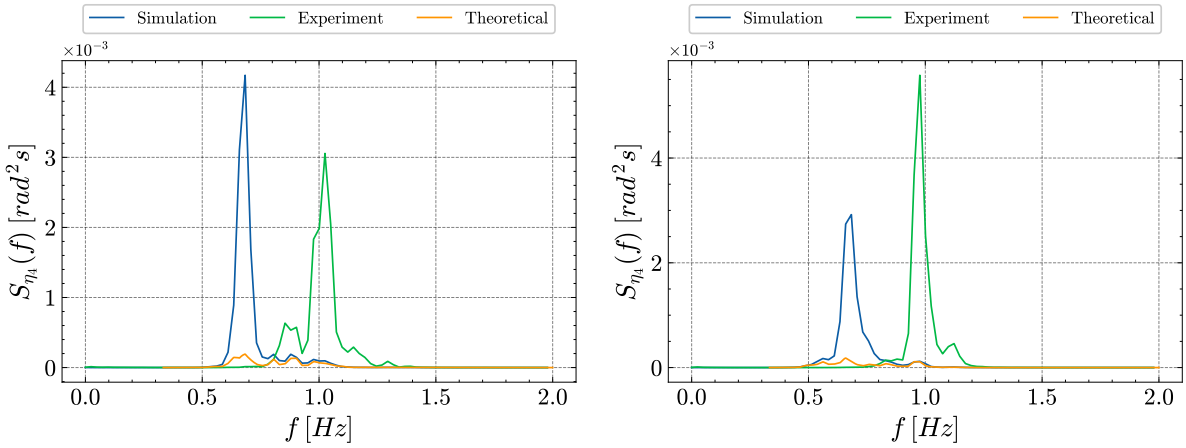
used a total of 500 wave components for the wave realizations. The data generated by Mounet was from an early development stage used in validation of the presented simulation model. The 6DOF simulation model of CSAD was then simulated with the same amount of wave components as those defined in the lab experiments. As no thrust allocation logic or thruster dynamics were included in the vessel simulation model at this point, only the vertical motions were examined. A simple controller was implemented to control the simulated vessel and keep it at a fixed heading in beam sea, such that the relative wave angle remained unchanged at $\beta_{rel} = 180°$. The response spectra for two separate sea states are presented in Figure 3.17 - Figure 3.19. The figures include both the experimental and simulated response, in addition to the theoretical calculated directly from the response amplitude operators of the vessel.



(a) Hs = 0.06 m and Tp = 1.3 s.

(b) Hs = 0.06 m and Tp = 1.5 s.

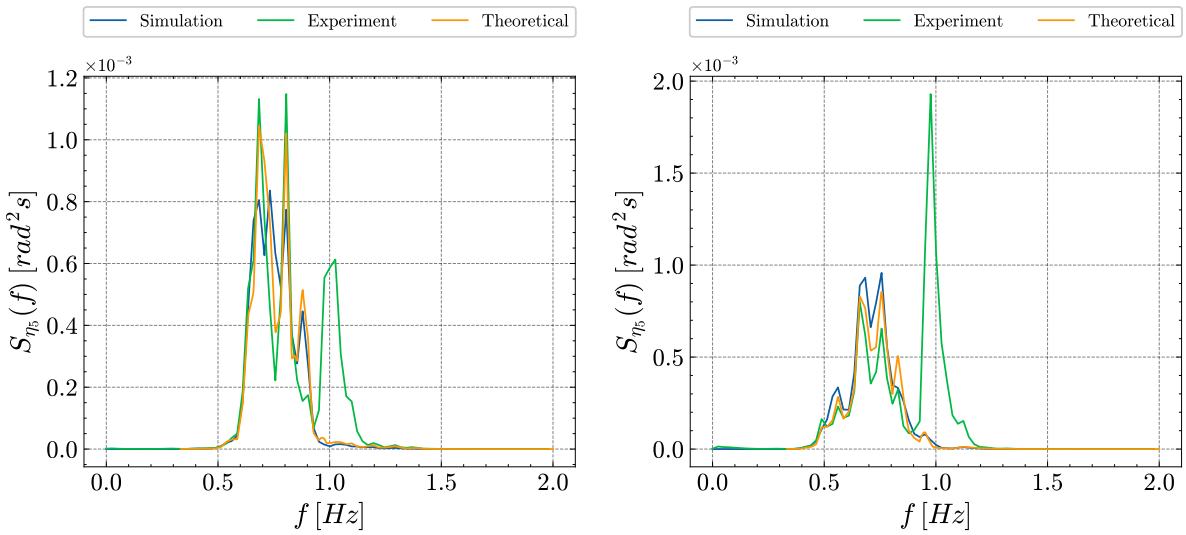
Figure 3.17: Validation of response spectra in heave for simulated, experimental, and theoretical response.



(a) Hs = 0.06 m and Tp = 1.3 s.

(b) Hs = 0.06 m and Tp = 1.5 s.

Figure 3.18: Validation of response spectra in roll for simulated, experimental, and theoretical response.



(a) Hs = 0.06 m and Tp = 1.3 s.

(b) Hs = 0.06 m and Tp = 1.5 s.

Figure 3.19: Validation of response spectra in pitch for simulated, experimental, and theoretical response.

The response spectra presented yield a very promising result. This is particularly the case for the heave and pitch response, in Figure 3.17 and Figure 3.19, where the simulated response coincide well with both the experimental and theoretical results. It can be observed that an unmodeled peak is evident in the experimental pitch response spectrum, for both sea states. The absence of this peak in the theoretical response spectra suggests the need for an improvement in the RAOs of the vessel. It is assumed that the RAOs would require a higher resemblance of the actual dynamics in order to capture these frequencies in the simulation model.

As the tests were conducted in head sea, it could be assumed that no roll motion should be present in the displayed spectra. This is also seen in the theoretical response in Figure 3.18. However, this is not the case for neither the simulated nor the experimental spectra. This could be due to the controller struggling to obtain β_{rel} at exactly 180°, resulting in roll motion. The experimental and simulated peak do also differ in frequency, but as previously stated, caution should be exercised when evaluating roll motion generated by the simulation model. This is because both the RAOs and force RAOs are derived from linear potential theory, whereas roll motion often consist of highly nonlinear phenomena (Faltinsen, 1990).

Chapter 4

Problem formulation and methods

4.1 Problem formulation

This master's thesis addresses the compensation of second order, low-frequency wave loads in dynamic positioning vessels. These loads contain a slowly varying component, and are commonly compensated for using integral action in the feedback loop. However, this method is susceptible to transients originating from low-frequency contributions, which can manifest as offsets in the vessel's position. That being the case, the thesis aims to explore alternative control law algorithms that could more effectively compensate for these loads, and thereby mitigate such offsets. The main objective was given in the introduction and is repeated in the bullet point underneath.

- Explore the effectiveness of alternative compensation techniques, in comparison to traditional integral action, for mitigating response from low-frequency (not necessarily zero-frequency) wave load components.

A DP vessel is exposed to slowly varying loads, arising from second order wave loads, wind, and current, as well as from uncertain factors in the control design model, such as inaccuracies in hydrodynamic parameters. However, the DP system faces difficulty in identifying and distinguishing these forces. Hence, they are usually grouped together as a single vector, termed the residual load vector or bias load vector, $\boldsymbol{b}$. This bias vector is frequently modeled in the NED frame (Fossen, 2022), due to the assumed predominance of environmental loads, which more often than not are represented in a NED frame.

In an ideal case, these residual loads loads should be perfectly counteracted for in a DP controller. This can, however, prove to be a challenging task. The inherent complexity and nonlinearity present in real-world systems, makes the control system struggle to react accordingly, especially during highly transient load conditions. Ongoing research is being performed on the subject, dedicated to address these challenges, and striving towards the attainment of a *perfect bias compensation*.

The bias term is typically modelled to be constant. This can result in inaccuracies during transient behavior. These transient load changes are typically not accounted for in deterministic mathematical models, and are, thus, considered as part of the unmodeled dynamics. Consequently, they should be compensated for by the bias term. A constant bias term may therefore struggle to cope with these transient loads, leading to poor DP system performance. This is often referred to as the *transient load challenge* (Værnø, 2020). The mentioned transient events may be operation specific, such as crane procedures or gangway connections, or they may be operator induced, such as mode changes and heading changes.

The aforementioned challenge is best illustrated by an example. A similar case is presented in Værnø (2020).

Example 4.1: The transient load challenge

Given the high-fidelity DP simulation model derived in Chapter 3, presented mathematically once again in (4.1). All external forces from waves and wind are lumped into the nonlinear expression $\boldsymbol{f}$. Current forces are still modelled in the relative velocity term, $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$. It is assumed that the current is irrotational and constant. A simplified low-frequency control design model is suggested in (4.2), according to Section 2.4.6, where the bias is modelled in the NED frame. $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{D}}$ are the estimated mass and damping matrices, while $\hat{\boldsymbol{\tau}}_{ctrl}$ is the estimated thruster dynamics.

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r = -\boldsymbol{D}\boldsymbol{\nu}_r + \boldsymbol{\tau}_{ctrl} + \boldsymbol{f} \tag{4.1}$$

$$\hat{\boldsymbol{M}}\dot{\boldsymbol{\nu}}_{cdm} = -\hat{\boldsymbol{D}}\boldsymbol{\nu}_{cdm} + \hat{\boldsymbol{\tau}}_{ctrl} + \mathbf{R}^\top \boldsymbol{b}^{\{n\}} \tag{4.2}$$

If the residual loads are captured perfectly in the bias term, the two models should coincide. However, this assumption presumes that the velocity in the simulation model only contains low-frequency motion, which is not generally true. This can be handled in a simulation platform, by removing all wave-frequency loads from the vector $\boldsymbol{f}$, so that only slowly varying loads excite the vessel, resulting in a slowly varying response. The simulation platform also provides access to the remaining loads, so that it becomes possible to derive an expression for the optimal bias term. (4.2) is subtracted from (4.1), and solved for the bias term.

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r - \hat{\boldsymbol{M}}\dot{\boldsymbol{\nu}}_{cdm} = -\boldsymbol{D}\boldsymbol{\nu}_r + \hat{\boldsymbol{D}}\boldsymbol{\nu}_{cdm} + \widetilde{\boldsymbol{\tau}}_{ctrl} + \boldsymbol{f} - \mathbf{R}^\top \boldsymbol{b}^{\{n\}} \tag{4.3a}$$

$$\mathbf{R}^\top \boldsymbol{b}^{\{n\}} = -\widetilde{\boldsymbol{M}}\dot{\boldsymbol{\nu}} + \underbrace{\boldsymbol{M}\dot{\boldsymbol{\nu}}_c}_{\mathbf{0}} - \widetilde{\boldsymbol{D}}\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu}_c + \widetilde{\boldsymbol{\tau}}_{ctrl} + \boldsymbol{f} \tag{4.3b}$$

$$\mathbf{R}^\top \boldsymbol{b}^{\{n\}} = -\widetilde{\boldsymbol{M}}\dot{\boldsymbol{\nu}} + \widetilde{\boldsymbol{D}}\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu}_c + \widetilde{\boldsymbol{\tau}}_{ctrl} + \boldsymbol{f} \tag{4.3c}$$

$\widetilde{\boldsymbol{M}}, \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{\tau}}$ represent the error in the estimated mass matrix, damping matrix and thruster configuration respectively. The optimal bias calculated above will also be referred to as the *true* bias or *actual* bias in the following sections.

The expression in (4.3c) can be even further simplified by assuming that the control design model perfectly models these hydrodynamic coefficients, as well as the assumption of perfect thrust allocation, so that $\widetilde{\boldsymbol{M}}, \widetilde{\boldsymbol{D}}, \widetilde{\boldsymbol{\tau}} = \mathbf{0}$. This will never be the case in a real system, but can, nonetheless, be obtained in a simulation platform. The bias will then only be a sum of damping due to current, and external forces represented in $\boldsymbol{f}$.

$$\boldsymbol{b}^{\{n\}} = \mathbf{R}(\psi)\Big[\boldsymbol{D}\boldsymbol{\nu}_c + \boldsymbol{f}\Big] \tag{4.4a}$$

$$\boldsymbol{b}^{\{b\}} = \boldsymbol{D}\boldsymbol{\nu}_c + \boldsymbol{f} \tag{4.4b}$$

This residual term can then be calculated as the vessel is forced to make a complete rotation (360°), while being stationary at the origin. No wave or wind loads are present, but a constant current, $U_c = 0.2$ m/s, from $\beta_c = 45°$, acts on the vessel. The bias term, both in the BODY and NED frame, is presented in Figure 4.1.

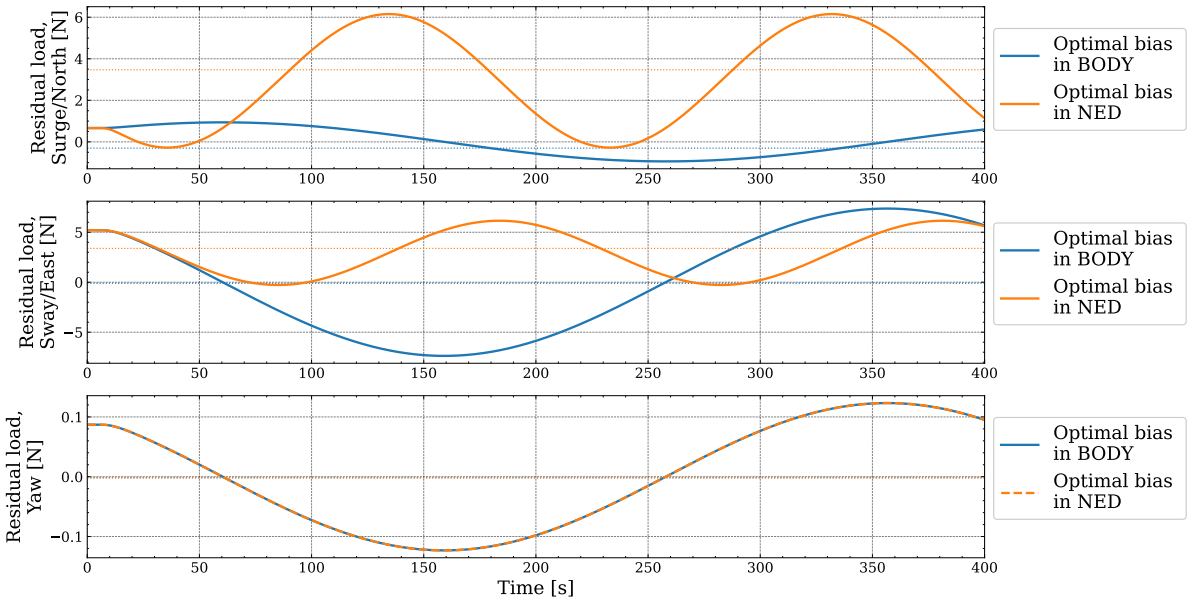


Figure 4.1: Simulated residual loads, both in BODY and NED, during a 360° turn. The dotted lines represent the mean values.

Figure 4.1 quite clearly illustrates that the assumption of a constant bias load can be invalid during large transients. The mean values are also shown as dotted lines. In this case, it can be seen that when expressed in the BODY frame, the bias load oscillates around zero. It can also be observed that for yaw rotation, the two reference frames align.

To more accurately capture the transient variations, demonstrated in the example, one may seek to improve the fidelity of the control design model. However, this would demand more complex hydrodynamic calculations, which could increase the computational time and render stability assessments more difficult. Even so, it would be impossible to perfectly identify the actual hydrodynamic parameters. Thus, representing the bias loads as a constant load has been demonstrated to yield good performance in both simulations and experimental testing, as reported by Fossen and Strand (1999) and Loría and Panteley (1999). However, it should be noted that these results are related to operations where the residual loads act at low-frequency (or zero-frequency), which can not be said to be the case for the example above (Værnø, 2020). Hence, more robust and adaptive controller algorithms, capable of more precisely estimating these residual loads, will be considered.

4.1.1 Disturbance rejection methods

This thesis presents four distinct control algorithms for disturbance rejection. Two of which, can be considered standard within commercial DP systems, involving integral action, and feedforward of observer-estimated bias. The remaining algorithms have not, to the best of the authors' knowledge, been employed in marine DP operations thus far. They entail an adaptive controller using a truncated Fourier series disturbance model, and a PI controller augmented with a neural network disturbance prediction model. All control algorithms are listed below, and thoroughly presented in Section 4.2 - Section 4.5.

Method 1 Bias compensation by integral action.

Method 2 Bias estimation from observer.

Method 3 Adaptive control using a Fourier series disturbance model

Method 4 Bias estimation from a neural network model

Although this master's thesis endeavors to analyze high-fidelity cases, certain simplifications were necessary due to various reasons. These simplifications are explicitly mentioned whenever they occur and their potential impacts on the results are discussed.

4.1.2 Setup and implementation

The control algorithms have been tested on the high-fidelity simulation platform, as previously introduced by the authors (see Chapter 3). The majority of the results have been obtained from this simulation platform, with the aim of validating some of the outcomes obtained from this simulation study through experimental testing. The simulation study is presented in Chapter 5, while the experimental validations are presented in Chapter 6.

For the purpose of conducting simulations, two distinct test scenarios were defined. The first of which, is a general stationkeeping test, aiming to evaluate the performance of the controllers after a stationary condition has been reached. The second scenario consists of a 30°-30° heading change, initiated from $\boldsymbol{\eta}_{ref} = [0, 0, 0]^\top$. The setpoint is then slowly changed to $\boldsymbol{\eta}_{ref} = [0, 0, 30°]^\top$, back to $\boldsymbol{\eta}_{ref} = [0, 0, -30°]^\top$, before returning to $\boldsymbol{\eta}_{ref} = [0, 0, 0]^\top$. It is emphasized that this maneuver assumes that a steady state performance is reached before the heading change is initiated, in order to only assess transients that occur as a consequence of the maneuver. The scenarios are, unless otherwise specified, performed in the same, predefined, environmental conditions. These are summarized as Seastate 1 and Seastate 2 in Table 4.1. Both consist of an irregular sea state, with similar wave parameters. Seastate 1, however, also include current effects. All simulations presented in Chapter 5 are conducted in Seastate 1, while Seastate 2 is intended for the experimental testing in Chapter 6.

Table 4.1: Overview of all sea states used both simulations and experimental testing.

Environmental condition	Hs [m]	Tp [s]	γ [-]	β_w [°]	U_c [m/s]	β_c [°]
Seastate 1	0.03	1.5	1.0	180	0.01	180
Seastate 2	0.03	1.5	1.0	180	0	-

In order to allow for a fair comparison between the distinct controllers, DFO optimization techniques, outlined in Section 2.7, are incorporated in the tuning process. The optimization process is performed, both for the nominal PID/PD controller, as well as for the adaptive controller. The translational and rotational gains were tuned separately, in addition to treating surge and sway gains as equal. This was introduced in order to reduce computational time, which was discovered to be a major problem in the inital tuning phase. The optimization process used the IAEW cost function, as presented in Section 2.7, and multiple runs with varying initial conditions were performed to avoid being stuck at any local minima. The tuning process defined a separate setpoint change from the one described previously, to circumvent the consequence of overfitting to a specific maneuver. A presentation of the tuning results are appended after the implementation of each controller. Even though the optimization technique should provide a set of optimal gains, some manual tuning, through trial and error, had to be performed subsequently, to achieve the desired performance.

As presented, the primary objective of this master's thesis is to evaluate whether or not the indicated control algorithms better cope and compensate for wave loads with low- and zero-frequency components.

The simulation study will involve both qualitative and quantitative evaluations of the controllers, as well as a sensitivity analysis of certain controller parameters. The PID controller will serve as a baseline for comparison with the successive algorithms. Various other hypothesis are also to be defined for each controller, and will be established in the subsequent sections.

The quantitative performance of the controllers can be measured by a given set of key performance indexes (KPIs). The different set of KPIs utilized in the upcoming chapters are

$$J_\eta = \int_{t_0}^{t_f} \Big[|\eta_N - \eta_{d,N}| + |\eta_E - \eta_{d,E}| + \frac{180}{\pi} |\psi - \psi_d| \Big] \mathrm{d}t \tag{4.5a}$$

$$J_{\tau,uv} = \int_{t_0}^{t_f} \Big[|\tau_{\text{surge}}| + |\tau_{\text{sway}}| \Big] \mathrm{d}t \tag{4.5b}$$

$$J_{\tau,r} = \int_{t_0}^{t_f} |\tau_{\text{yaw}}| \mathrm{d}t \tag{4.5c}$$

$$J_b = \int_{t_0}^{t_f} \Big[|\hat{b}_N - b_N| + |\hat{b}_E - b_E| + \frac{180}{\pi} |\hat{b}_\psi - b_\psi| \Big] \mathrm{d}t, \tag{4.5d}$$

where the final KPI, in (4.5d), is an evaluation of the controllers bias rejection term, evaluating the bias compensation term, $\hat{\boldsymbol{b}} \in \mathbb{R}^3$, against the extracted actual bias $\boldsymbol{b} \in \mathbb{R}^3$. Certain analyses will also use evaluations based on a weighted sum of some of these performance indicators.

4.2 Bias compensation by integral action

One of the most frequently used control methods for DP operations is the nominal PID controller. The controller utilizes integral action for disturbance rejection, and is a widely adopted method in control algorithms for DP operations. The integral term accumulates the calculated error over time and efficiently compensates for any constant disturbances. This provides a long-term control action, while ensuring that the control output converges to the desired setpoint. However, an integral term introduces a negative 90° phase shift to the response at low frequencies, which can limit the controller's stability margins (J. Balchen et al., 2016, ch.9). Nonetheless, the PID-controller remains a highly robust, simple and widely trusted controller in the industry. Its integral action will, consequently, serve as a baseline against other less frequently used disturbance rejection techniques in this thesis.

The control law for a nominal PID controller is presented in (4.6). The integral term is represented by a new, augmented state, $\boldsymbol{z}$, such that the control force, $\boldsymbol{\tau}_{ctrl}$, will include the integral of the cumulative positional error over time. The controller will, thus, add a corrective action to drive the system towards the setpoint, even for small or fluctuating errors.

$$\boldsymbol{\tau}_{ctrl} = -\mathbf{K}_p \mathbf{R}^\top(\psi)(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d) - \mathbf{K}_d(\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}_d) - \mathbf{K}_i \mathbf{R}^\top(\psi)\boldsymbol{z} \tag{4.6a}$$

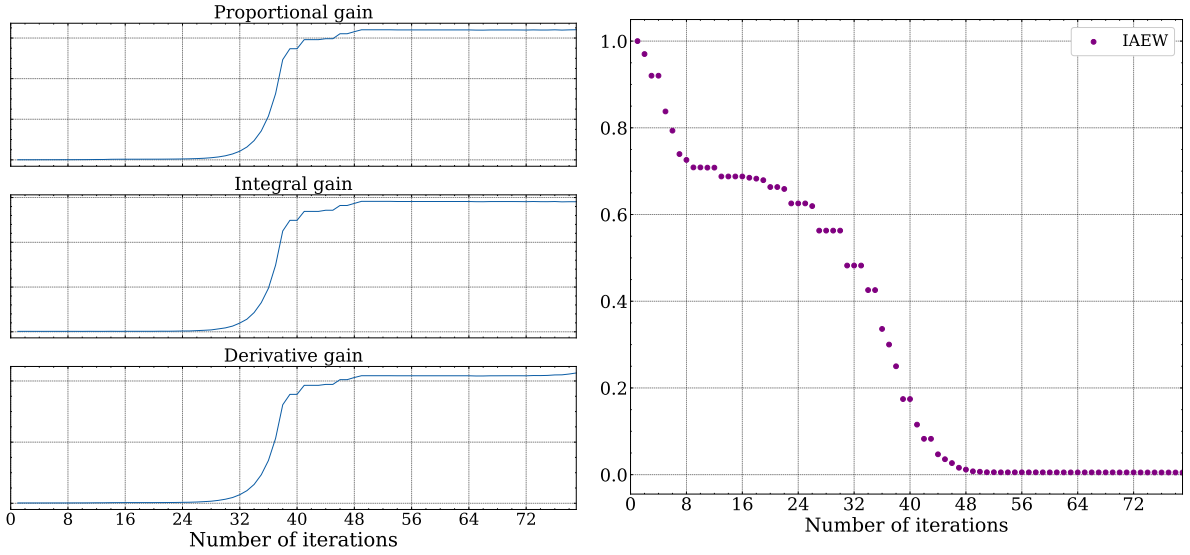
$$\dot{\boldsymbol{z}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d \tag{4.6b}$$

That being said, the integral term may also introduce challenges if not implemented properly. Integral windup is a well-known concept that should be addressed in all PID controllers. The phenomenon occurs when the integrator term continues to accumulate error even when the control output not effectively can compensate for such error. This can lead to undesirable consequences, as both the performance and stability of the controller can be affected. An anti-windup scheme has therefore been implemented simultaneously to the controller, to avoid non-physical actuator input.

The integral term in (4.6) is often tuned relatively slow for DP systems (Værnø, 2020). Fossen, 2021 presents a rule of thumb stating that the integral gain, at least, should inherit a value less than a tenth of the system's natural frequency. This is to prevent any potential overshoot, due to the substantial inertia exhibited by vessels. The large inertia will, in general, result in a slow response, which can cause the integral term to accumulate too much positional error before reaching a desired setpoint. It may, consequently, result in oscillatory performance, and give rise to errors during transient operations with fluctuating disturbances.

4.2.1 Tuning

The tuning of the PID controller was done using a DFO technique. Figure 4.2 displays the process, showing the development of the translational gains (K_p, K_i and K_d), as well as the overall cost function. The initial value for the tuning gains were deliberately set relatively low, so that the initial cost function described a stable system. Figure 4.3 presents the response and control load, τ_{cmd}, in sway, for a simple setpoint change, both before and after DFO tuning was implemented. As can be observed from the figure, the response of the vessel after tuning, closely aligns with the desired response.



(a) Controller gains.

(b) Normalized cost function.

Figure 4.2: Derivative free optimization procedure for the PID controller.

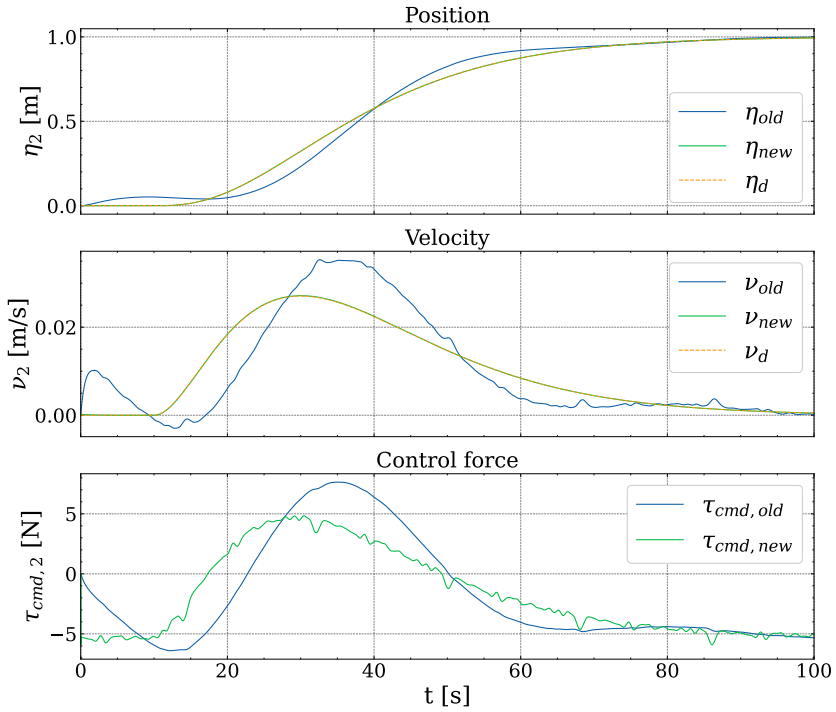


Figure 4.3: Before and after DFO tuning of the PID controller for a simple setpoint change in surge.

4.3 Bias estimation from observer

Alternatively to the conventional integral term, the disturbances can also be compensated for by directly feeding the bias estimate, provided by the observer, into the control law. This method, shown mathematically in (4.7), is proposed by Loría and Panteley (1999), where the separation principle is utilized to ensure global asymptotic stability. A bias estimate, extracted from a model-based observer, such as the one presented in (2.48), is then included in the feedback loop together with position and velocity estimates, and utilized in the control law.

$$\boldsymbol{\tau}_{ctrl} = -\mathbf{K}_p\mathbf{R}^\top(\psi)(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d) - \mathbf{K}_d(\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}_d) - \mathbf{R}^\top(\psi)\hat{\boldsymbol{b}} \tag{4.7}$$

The use of an estimated bias instead of the traditional integral action do lead to some benefits. Firstly, the tuning process is simpler for an offline data process such as in the observer. Optimization methods can also be used to find acceptable gains. Integral windup will also not be a problem when implementing the bias estimate, as no integrator is present to accumulate error. Nonetheless, integral action can be tuned independently of the bias response time, enabling the possibility to account for steady state offsets (Værnø, Brodtkorb et al., 2019). Bias estimates from the observer may also exhibit a more oscillatory behavior, which can lead to undesirable response when employed in a feedback loop.

The bias used in the control law can be represented by several different bias models. Different bias models will have different characteristics depending on traits, such as linearity, stochasticity and variations with time. The bias model should therefore, carefully, be selected according to the dynamics it is desired to capture. The previously presented CDM in (2.48), presented a random walk bias model, $\dot{\boldsymbol{b}} = \boldsymbol{w}_b$, where the bias is represented with random fluctuations over time. This model captures the stochastic nature of the bias, which often is to be found in marine environments, due to the natural stochasticity of waves. The stochastic, first order Markov process was also briefly mentioned as an alternative, as presented in Sørensen (2018), ch.7. This model introduces temporal correlation in the bias estimation, accounting for memory effects in the bias dynamics.

Værnø, Brodtkorb et al. (2019), also looked into how different bias models, utilized as in (4.7), affected the performance of the DP control system. The paper presented three distinct bias models and concluded that bias compensation from a separate observer, which was fine-tuned to accurately estimate the true low-frequency bias load, yielded the most favorable results. Even though these bias models not will be implemented any further in this thesis, a different bias model candidate will be presented in Section 4.5, based on an artificial neural network.

The combination of bias estimation from an observer, and bias compensation through integral action, will serve as a baseline for the upcoming disturbance rejection methods, and are both considered as well established within commercial DP systems.

4.3.1 Tuning

For the augmented PD controller in (4.7), the same tuning parameters as for the proportional and derivative gain in the PID controller have been implemented. Even though the final term in the control law may exhibit some differences to the integral term in (4.6), it is assumed that the PD-part of the controllers can be tuned independently of the bias rejection term. Again, minor changes were done, though trial and error, to obtain an ideal performance.

4.4 Adaptive control using a Fourier series disturbance model

The adaptive control using a Fourier series disturbance model mainly involve estimating the bias term in the control design model as a truncated Fourier series. The Fourier series will have a set of Fourier coefficients $\boldsymbol{\Theta}$, as well as a *regressor*, $\boldsymbol{\Phi}(t)$, consisting of harmonic functions with N pre-defined frequencies. By estimating the Fourier coefficients in an adaptive law, the bias, $\mathbf{b}(t) \in \mathbb{R}^3$, can then be

modeled as

$$\mathbf{b}(t) = \mathbf{\Phi}^\top(t)\mathbf{\Theta}, \tag{4.8}$$

where

$$\mathbf{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^\top & \boldsymbol{\theta}_2^\top & \boldsymbol{\theta}_3^\top \end{bmatrix}^\top, \tag{4.9}$$

and

$$\mathbf{\Phi}(t) = \begin{bmatrix} \boldsymbol{\phi}_1 & \mathbf{0}_{2N+1} & \mathbf{0}_{2N+1} \\ \mathbf{0}_{2N+1} & \boldsymbol{\phi}_2 & \mathbf{0}_{2N+1} \\ \mathbf{0}_{2N+1} & \mathbf{0}_{2N+1} & \boldsymbol{\phi}_3 \end{bmatrix}. \tag{4.10}$$

In (4.9) and (4.10), $\boldsymbol{\theta}_i \in \mathbb{R}^{2N+1}$ and $\boldsymbol{\phi}_i \in \mathbb{R}^{2N+1}$, represent the Fourier coefficients and set of harmonic functions, respectively, for DOF i. Evidently, considering N harmonic frequencies, the resulting dimensions for the full matrices are $\mathbf{\Theta} \in \mathbb{R}^{6N+3}$, and $\mathbf{\Phi}(t) \in \mathbb{R}^{(6N+3)\times(6N+3)}$. For every degree of freedom, the bias is then modelled as

$$\mathrm{b}_i(t) = \boldsymbol{\phi}_i^\top(t)\,\boldsymbol{\theta}_i = \begin{bmatrix} 1 & \cos(\omega_1 t) & \sin(\omega_1 t) & \dots & \cos(\omega_N t) & \sin(\omega_N t) \end{bmatrix} \begin{bmatrix} a_{i0} \\ a_{i1} \\ b_{i1} \\ \vdots \\ a_{iN} \\ b_{iN} \end{bmatrix}, \tag{4.11}$$

or represented as a sum,

$$\mathrm{b}_i(t) = a_{i0} + \sum_{j=1}^{N} a_{ij}\cos(\omega_j t) + b_{ij}\sin(\omega_j t). \tag{4.12}$$

In the forthcoming calculations, $\mathbf{\Theta}$ will be assumed constant, that is $\dot{\mathbf{\Theta}} = \mathbf{0}$, which is considered plausible for a fully developed sea state. It should be noted that the first term in the regressor is constant, constructed to represent the mean drift loads.

The selection of Fourier coefficients in $\mathbf{\Theta}$ can be found with various techniques. As mentioned previously, this section will construct an adaptive law, where stability of the controller can be guaranteed according to its control Lyapunov function (CLF). The coefficients can, however, also be estimated by other means, such as an additional state in the observer model, according to the internal model principle.

The calculated bias is directly fed into an adaptive control law, where the bias estimate is designed to suppress the actual disturbances working on the vessel. A block diagram of the control system is shown in Figure 4.4.

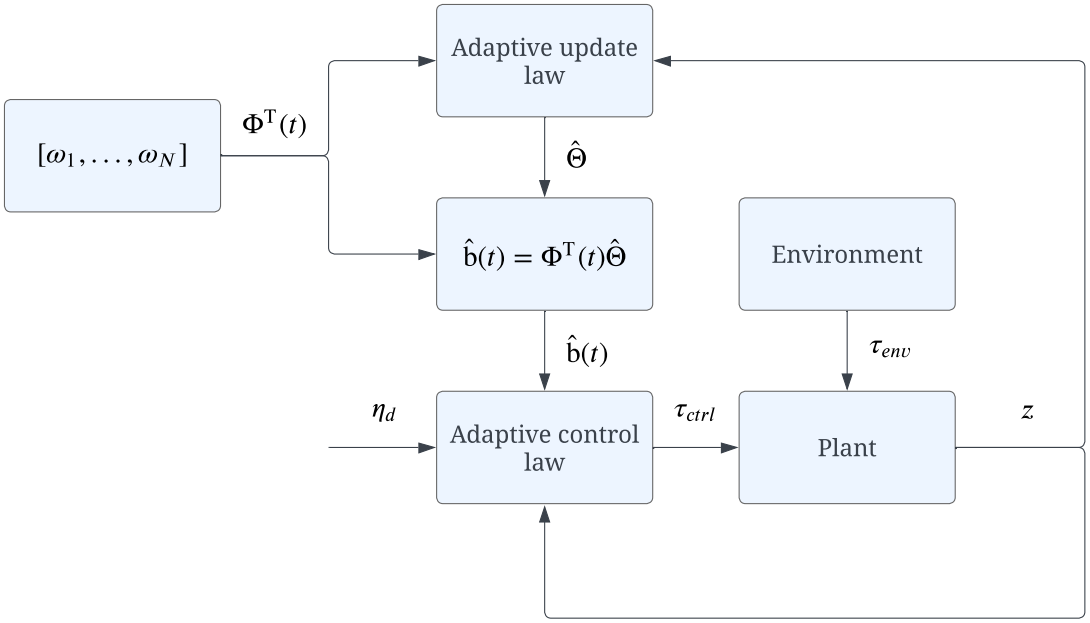


Figure 4.4: Block diagram of the adaptive controller.

The control law is determined by using LgV-backstepping, (Arcak & Kokotović, 2001), calculated through a recursive construction of a CLF. This is a robust design technique ensuring global stability. The procedure introduces a change of coordinates, where a new state vector, $\mathbf{z}$, is generated. The given approach is inspired by the former M.Sc. thesis by Brørby (2022), solving a similar problem, as well as the procedure described in appendix A in Fossen (2021).

The transformation of coordinates is displayed in (4.13). The control objective, in accordance with Fossen (2021), is to drive the first state, $\mathbf{z}_1$, to zero. The second state, $\mathbf{z}_2$, will be used as a virtual control input to stabilize the first subsystem.

$$\mathbf{z}_1 = \mathbf{R}^\top(\psi)(\boldsymbol{\eta} - \boldsymbol{\eta}_d(t)) \tag{4.13a}$$
$$\mathbf{z}_2 = \boldsymbol{\nu} - \boldsymbol{\alpha} \tag{4.13b}$$

Here, $\boldsymbol{\alpha}$, is another virtual control input, and will be specified later.

Step 1: (4.13a) can to begin with be differentiated.

$$\begin{aligned}\dot{\mathbf{z}}_1 &= \dot{\mathbf{R}}^\top(\psi)(\boldsymbol{\eta} - \boldsymbol{\eta}_d(t)) + \mathbf{R}^\top(\psi)(\mathbf{R}(\psi)\boldsymbol{\nu} - \dot{\boldsymbol{\eta}}_d(t)) \\ &= -\mathbf{S}(r)\mathbf{z}_1 + \mathbf{z}_2 + \boldsymbol{\alpha} - \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t)\end{aligned} \tag{4.14}$$

A CLF candidate is then proposed as

$$\mathbf{V}_1(\boldsymbol{z}_1) = \frac{1}{2}\boldsymbol{z}_1^\top \boldsymbol{z}_1, \tag{4.15}$$

which can be differentiated as

$$\begin{aligned}\dot{\mathbf{V}}_1 =& \boldsymbol{z}_1^\top \dot{\boldsymbol{z}}_1 \\ =& \boldsymbol{z}_1^\top \left[-\mathbf{S}(r)\boldsymbol{z}_1 + \mathbf{z}_2 + \boldsymbol{\alpha} - \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t)\right] \\ =& \boldsymbol{z}_1^\top \left[\mathbf{z}_2 + \boldsymbol{\alpha} - \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t)\right].\end{aligned} \tag{4.16}$$

After inspecting (4.16), a proposal can be made for the virtual control input,

$$\boldsymbol{\alpha} = -\mathbf{K}_1 \boldsymbol{z}_1 + \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t) + \boldsymbol{\alpha}_0, \tag{4.17}$$

where $\mathbf{K}_1 \in \mathbb{R}^{3x3}$ is defined as a positive tuning matrix, and $\boldsymbol{\alpha}_0$ as a new virtual control law for making (4.16) negative. This will be further discussed later in the derivation. Inserting (4.17) into (4.16) yield

$$\begin{aligned}\dot{\mathbf{V}}_1 =& \boldsymbol{z}_1^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_1^\top \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t) + \boldsymbol{z}_1^\top \boldsymbol{\alpha}_0 - \boldsymbol{z}_1^\top \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t) \\ =& \boldsymbol{z}_1^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_1^\top \boldsymbol{\alpha}_0 \\ \leq& \kappa \boldsymbol{z}_1^\top \boldsymbol{z}_1 + \frac{1}{4\kappa}\boldsymbol{z}_2^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_1^\top \boldsymbol{\alpha}_0,\end{aligned} \tag{4.18}$$

where the final line in (4.18) has used *Young's inequality*, defined as $\boldsymbol{z}_1^\top \boldsymbol{z}_2 \leq \kappa \boldsymbol{z}_1^\top \boldsymbol{z}_1 + \frac{1}{4\kappa}\boldsymbol{z}_2^\top \boldsymbol{z}_2$ for a constant scalar $\kappa > 0$. The virtual control law can then be set to remove the first term in (4.18) by selecting

$$\boldsymbol{\alpha}_0 = -\kappa \mathbf{I}_3 \boldsymbol{z}_1. \tag{4.19}$$

This results in the following property for the CLF candidate,

$$\dot{\mathbf{V}}_1 \leq \frac{1}{4\kappa}\boldsymbol{z}_2^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1. \tag{4.20}$$

To finish off the first step, (4.17) can be inserted into (4.14), providing

$$\dot{\boldsymbol{z}}_1 = -\mathbf{S}(r)\mathbf{z}_1 + \mathbf{z}_2 - (K_1 + \kappa \mathbf{I}_3)\boldsymbol{z}_1. \tag{4.21}$$

Step 2: For step 2, the dynamics in (4.13b) is considered. Again, the first step is to find the time derivative. $\mathbf{M}$ is multiplied into the equation to easier introduce the kinetics of the CDM.

$$\begin{aligned}\mathbf{M}\dot{\boldsymbol{z}}_2 &= \mathbf{M}\dot{\boldsymbol{\nu}} - \mathbf{M}\dot{\boldsymbol{\alpha}} \\ &= -\mathbf{D}\boldsymbol{\nu} + \mathbf{b} + \boldsymbol{\tau}_{ctrl} - \mathbf{M}\dot{\boldsymbol{\alpha}} \\ &= -\mathbf{D}\boldsymbol{\nu} + \boldsymbol{\Phi}^\top(t)\boldsymbol{\Theta} + \boldsymbol{\tau}_{ctrl} - \mathbf{M}\dot{\boldsymbol{\alpha}}\end{aligned} \tag{4.22}$$

Again, a new CLF candidate is assigned. This time the previous CLF candidate in (4.15) is included, that is

$$\mathbf{V}_2(\boldsymbol{z}_1, \boldsymbol{z}_2) = \mathbf{V}_1(\boldsymbol{z}_1) + \frac{1}{2}\boldsymbol{z}_2^\top \mathbf{M}\boldsymbol{z}_2. \tag{4.23}$$

Its time derivative is given as

$$\begin{aligned}\dot{\mathbf{V}}_2 &= \dot{\mathbf{V}}_1 + \boldsymbol{z}_2^\top \mathbf{M}\dot{\boldsymbol{z}}_2 \\ &\leq \underbrace{\frac{1}{4\kappa}\boldsymbol{z}_2^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1}_{\dot{\mathbf{V}}_1} + \boldsymbol{z}_2^\top \left[-\mathbf{D}\boldsymbol{\nu} + \boldsymbol{\Phi}^\top(t)\boldsymbol{\Theta} + \boldsymbol{\tau}_{ctrl} - \mathbf{M}\dot{\boldsymbol{\alpha}}\right] \\ &\leq \frac{1}{4\kappa}\boldsymbol{z}_2^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_2^\top \left[-\mathbf{D}(\boldsymbol{z}_2 + \boldsymbol{\alpha}) + \boldsymbol{\Phi}^\top(t)\boldsymbol{\Theta} + \boldsymbol{\tau}_{ctrl} - \mathbf{M}\dot{\boldsymbol{\alpha}}\right].\end{aligned} \tag{4.24}$$

$\hat{\boldsymbol{\Theta}}$ is then introduced as an estimate of $\boldsymbol{\Theta}$, with the estimation error $\tilde{\boldsymbol{\Theta}} = \boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}$. The error dynamic can then be described by $\dot{\tilde{\boldsymbol{\Theta}}} = \dot{\boldsymbol{\Theta}} - \dot{\hat{\boldsymbol{\Theta}}} = -\dot{\hat{\boldsymbol{\Theta}}}$, where the final equality is due to the assumption that $\boldsymbol{\Theta}$ is constant.

The control law, $\boldsymbol{\tau}_{ctrl}$, in (4.24), can then be chosen to counteract the undesired terms in the CLF. Note that it is desirable to keep the damping term $-\mathbf{D}\boldsymbol{z}_2$ because of the stability property it naturally comes with. This is in general an advantage with vectorial backstepping, allowing the designer to select which terms to keep and remove due to their stability properties. The control law is proposed as

$$\boldsymbol{\tau}_{ctrl} = -\mathbf{K}_2 \boldsymbol{z}_2 + \mathbf{D}\boldsymbol{\alpha} + \mathbf{M}\dot{\boldsymbol{\alpha}} - \boldsymbol{\Phi}^\top(t)\hat{\boldsymbol{\Theta}}, \tag{4.25}$$

where $\mathbf{K}_2 \in \mathbb{R}^{3x3}$, similarly to $\mathbf{K}_1$, is a positive tuning matrix. Inserting (4.25) into the CLF candidate in (4.24) provides

$$\begin{aligned}\dot{\mathbf{V}}_2 \leq & \frac{1}{4\kappa}\boldsymbol{z}_2^\top \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 - \boldsymbol{z}_2^\top \mathbf{D}\boldsymbol{z}_2 - \boldsymbol{z}_2^\top \mathbf{K}_2 \boldsymbol{z}_2 + \boldsymbol{z}_2^\top \boldsymbol{\Phi}^\top(t)\tilde{\boldsymbol{\Theta}} \\ = & -\boldsymbol{z}_2^\top \underbrace{\left[-\frac{1}{4\kappa}\mathbf{I}_3 + \mathbf{D} + \mathbf{K}_2\right]}_{\mathbf{K}_3} \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_2^\top \boldsymbol{\Phi}^\top(t)\tilde{\boldsymbol{\Theta}}\end{aligned} \tag{4.26}$$

where $\mathbf{K}_3$ is defined for simplicity. The state equation in (4.22) can now also be written as

$$\mathbf{M}\dot{\boldsymbol{z}}_2 = -\mathbf{D}\boldsymbol{z}_2 - \mathbf{K}_2\boldsymbol{z}_2 + \boldsymbol{\Phi}^\top(t)\tilde{\boldsymbol{\Theta}} \tag{4.27}$$

Step 3: Finally, we introduce the previously defined state $\dot{\tilde{\boldsymbol{\Theta}}} = -\dot{\hat{\boldsymbol{\Theta}}}$. The final CLF candidate can then be selected as

$$\mathbf{V}(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Theta}}) = \mathbf{V}_2 + \frac{1}{2}\tilde{\boldsymbol{\Theta}}^\top \boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\Theta}}. \tag{4.28}$$

$\boldsymbol{\Gamma} \in \mathbb{R}^{(6N+3)\times(6N+3)}$ is defined as the final tuning matrix, determining how rapidly the coefficients in $\boldsymbol{\Theta}$ will be updated. For the final time the time derivative of the CLF is derived,

$$\begin{aligned}\dot{\mathbf{V}} = & \dot{\mathbf{V}}_2 + \tilde{\boldsymbol{\Theta}}^\top \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\Theta}}} \\ \leq & -\boldsymbol{z}_2^\top \mathbf{K}_3 \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \boldsymbol{z}_2^\top \boldsymbol{\Phi}^\top(t)\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\Theta}}^\top \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\Theta}}} \\ = & -\boldsymbol{z}_2^\top \mathbf{K}_3 \boldsymbol{z}_2 - \boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 + \tilde{\boldsymbol{\Theta}}^\top \left[\boldsymbol{\Phi}(t)\boldsymbol{z}_2 + \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\Theta}}}\right],\end{aligned} \tag{4.29}$$

where the final equation is obtained by exploiting $\boldsymbol{z}_2^\top \boldsymbol{\Phi}^\top(t)\tilde{\boldsymbol{\Theta}} = \tilde{\boldsymbol{\Theta}}^\top \boldsymbol{\Phi}(t)\boldsymbol{z}_2$. The error state $\dot{\tilde{\boldsymbol{\Theta}}}$, can then be chosen such that the CLF candidate will be negative, that is

$$\dot{\tilde{\boldsymbol{\Theta}}} = -\boldsymbol{\Gamma}\boldsymbol{\Phi}(t)\boldsymbol{z}_2, \tag{4.30}$$

which gives

$$\dot{\hat{\boldsymbol{\Theta}}} = -\dot{\tilde{\boldsymbol{\Theta}}} = \boldsymbol{\Gamma}\boldsymbol{\Phi}(t)\boldsymbol{z}_2. \tag{4.31}$$

This choice will evidently lead to the time derivative of the CLF being negative semi-definite

$$\dot{\mathbf{V}} \leq -\boldsymbol{z}_1^\top \mathbf{K}_1 \boldsymbol{z}_1 - \boldsymbol{z}_2^\top \mathbf{K}_3 \boldsymbol{z}_2 \leq 0 \quad \forall\, (\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Theta}}) \neq \mathbf{0}. \tag{4.32}$$

A conclusion stating that the equilibrium point $(\boldsymbol{z}_1, \boldsymbol{z}_2, \tilde{\boldsymbol{\Theta}}) = \mathbf{0}$ will be uniformly globally stable (UGS), can, as a consequence of Lyapunov's direct method, thus be drawn. *LaSalle-Yoshizawa*'s theorem for non-autonomous systems also proves that $(\boldsymbol{z}_1, \boldsymbol{z}_2)$ will converge towards $\mathbf{0}$ (Fossen, 2022). Note that the convergence of $\tilde{\boldsymbol{\Theta}}$ towards $\mathbf{0}$ not is guaranteed. All conclusions will only hold if the already made assumptions prove to be valid. Thus, it is required that the CDM accurately describes the system dynamics, and that the proper bias term persistently excite the regressor, $\boldsymbol{\Phi}(t)$.

The calculations in this section can be summerized with the following equations.

Change of coordinates:

$$\begin{aligned}\boldsymbol{z}_1 &= \mathbf{R}^\top(\psi)(\boldsymbol{\eta} - \boldsymbol{\eta}_d(t)) \\ \boldsymbol{z}_2 &= \boldsymbol{\nu} - \boldsymbol{\alpha}\end{aligned}$$

Virtual control:

$$\boldsymbol{\alpha} = -\mathbf{K}_1\boldsymbol{z}_1 + \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}}_d(t) - \kappa\mathbf{I}_3\boldsymbol{z}_1$$

Control law

$$\boldsymbol{\tau}_{ctrl} = -\mathbf{K}_2\boldsymbol{z}_2 + \mathbf{D}\boldsymbol{\alpha} + \mathbf{M}\dot{\boldsymbol{\alpha}} - \boldsymbol{\Phi}^\top(t)\hat{\boldsymbol{\Theta}}$$

Adaptive update law

$$\dot{\hat{\boldsymbol{\Theta}}} = \boldsymbol{\Gamma}\boldsymbol{\Phi}(t)\boldsymbol{z}_2$$

4.4.1 Frequency-selection for the adaptive update law

An important aspect using the derived adaptive controller lays in the selection of, and quantity of, frequencies employed in the regressor. A higher number of frequencies is expected to give a more accurate disturbance estimate, but also increase the required computational time by enlarging the vectors $\mathbf{\Phi}$ and $\mathbf{\Theta}$. An appropriate interval, $[\omega_1, \ldots, \omega_N]$, consisting of N components must therefore be selected with care.

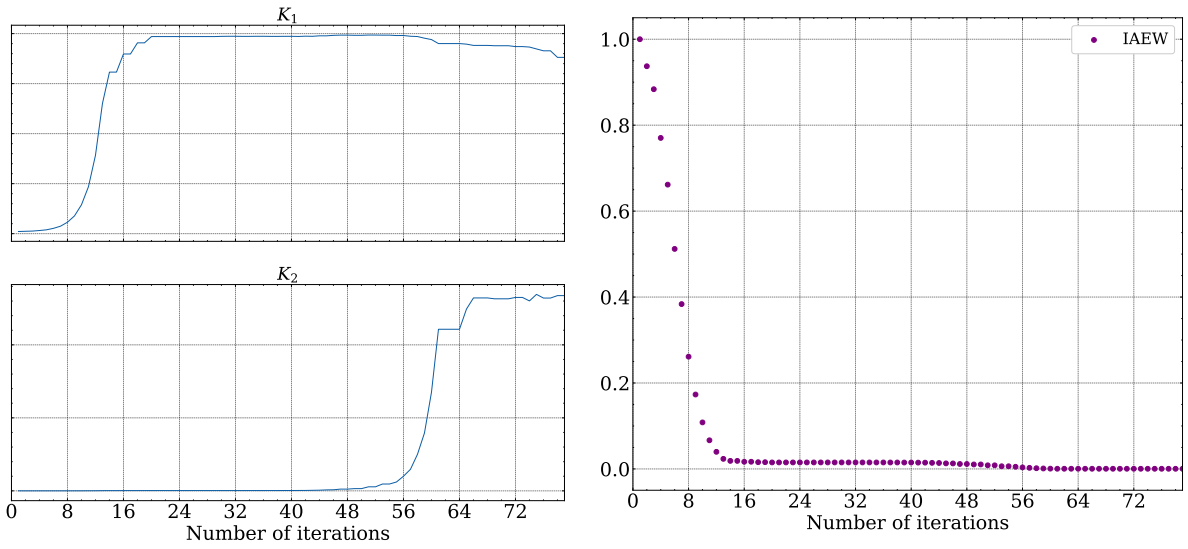
An extensive analysis was conducted by the authors in the preceding project thesis. The thesis introduced a frequency analysis of the model vessel CSAD, from Section 2.8.1, with the aim of identifying the most important, low-frequent frequencies in the load, and response spectrum. The analysis yielded valuable information regarding which discrete frequency components that should be represented in the truncated Fourier series, for sufficiently accurate estimation and compensation. Findings found in the analysis have been used in the implementation of the adaptive controller (Kongshaug & Mo, 2022).

The number of frequencies, N, represented in the truncated Fourier series, may affect the performance of the estimate. A higher number of frequencies would, intuitively, be expected to enhance the accuracy of the estimate. Nevertheless, as the residual loads often only span over a limited frequency range, the frequency components of interest may be closely spaced. To evaluate the effects this may produce, an objective for the adaptive controller was formulated:

- Evaluate how varying number of frequency components in the adaptive bias model impact the performance of the estimated bias.

4.4.2 Tuning

The adaptive control law in (4.25) can, in general, be considered as less intuitive to tune, compared to the previously presented PID and PD controllers. This enhance the practicality of DFO techniques, ensuring that optimization algorithms will handle the process. DFO is performed for the tuning matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$, again separating the procedure into translational and rotational motion. Figure 4.5 display the development of the tuning gains and cost function during the optimization method, while Figure 4.6 presents the response in sway for a simple setpoint change, both before and after DFO tuning was implemented.



(a) Controller gains.

(b) Normalized cost function.

Figure 4.5: Derivative free optimization process for the adaptive controller.

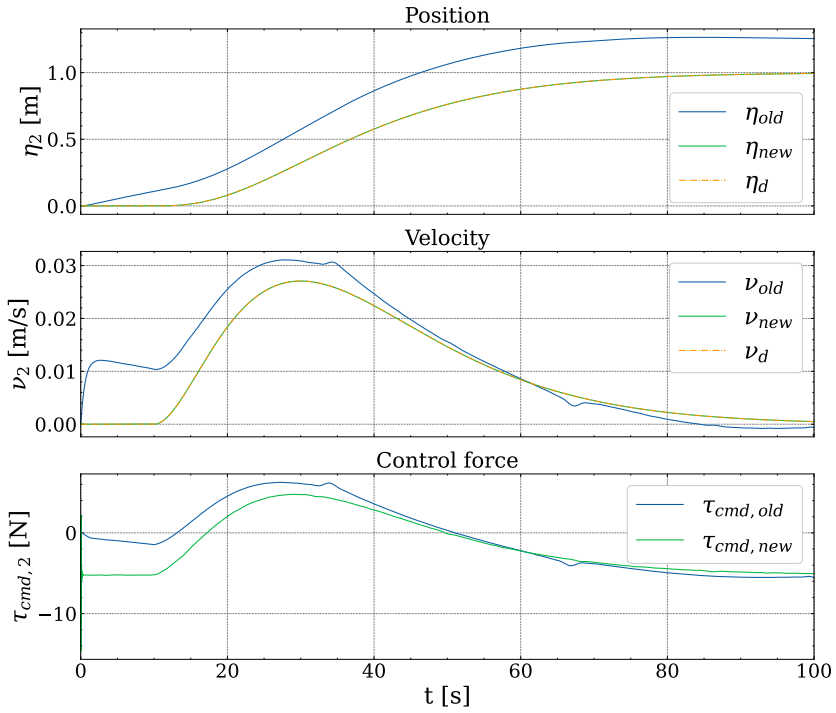


Figure 4.6: Before and after DFO tuning of adaptive controller for a simple setpoint change in sway.

The adaptive update law in (4.31), also include a tuning matrix, $\mathbf{\Gamma}$, which regulates the rate at which the adaptive coefficients are updated. This is adjusted manually. The choice of $\mathbf{\Gamma}$ has a significant impact on the convergence behavior of the bias estimate. It can be expected that a larger value of $\mathbf{\Gamma}$ should lead to a more aggressive update of the adaptive coefficients, potentially driving the estimate closer to the true bias value within a shorter period of time. This improvement may, however, come at the expense of introducing overshoot and instability. This trade-off will be evaluated in Chapter 5 with the following objective.

- Evaluate how the performance of the adaptive bias model is affected by its tuning matrix, $\mathbf{\Gamma}$, and how the magnitude of the gain works as a trade-off between a rapid estimation and introduction of oscillations.

4.5 Bias estimation with a neural network

As presented in Chapter 2, artificial neural networks are in general well suited for nonlinear function approximation. Consequently, they can be effectively utilized to approximate highly nonlinear characteristics of a marine vessels. By implementing neural networks for this purpose, the nonlinear and complex behavior of the real-time residual loads may be captured and approximated. The residual loads, or bias loads, $\boldsymbol{b} \in \mathbb{R}^3$, can then be directly integrated into, and compensated for, in a nominal control law, such as the augmented PD controller in (4.7). Determined by the accuracy of which the neural network is able to approximate the actual bias, this approach can further enhance the controller's ability to adapt and respond to various disturbance, and, hence, improve overall control performance in a dynamic marine environment.

In order to implement the aforementioned controller, a neural network has been implemented. The inputs for the network have been carefully selected according to their assumed relevance to the actual residual bias loads. The availability of measurements onboard the vessel also played a role in determining the selection of input parameters, which will be further discussed in Section 4.5.1. By considering the mathematical equation in the already established simulation model, (2.58), it can be suspected that certain variables, such as $\boldsymbol{\nu}$, and $\boldsymbol{\tau}_{ctrl}$, heavily influence the residual loads on the vessel. As

slowly varying wave loads and current also are lumped into the bias term, it is reasonable to assume that significant wave height, Hs, wave period, Tp, current velocity, ν_c, and wave and current angles, $\beta_{w,rel}, \beta_{c,rel}$, also impact the residual loads. For the network to be dependent on the relative wave, and current, angle, the heading of the vessel, ψ, should consequently be taken as input. Incorporating recently estimated bias output as input when training the network can also prove to be advantageous. This implementation, often referred to as a *recurrent neural network* (RNN), introduce an internal memory effect into the network, and can deal with the concept of autocorrelation, indicating that past values may influence the current bias. The fully developed neural network in this thesis then end up consisting of a total of 18 inputs and 3 outputs, which is selected to propagated through four hidden layers of dimension N, mathematically represented as a function $f : \mathbb{R}^{18} \to \mathbb{R}^3$, and visualized in Figure 4.7. The Leaky ReLU function is used as activation function in all layers, based on a simple study, evaluating the performance of some of the previously mentioned functions. This study is seen in Table 4.2 and Figure 4.8.

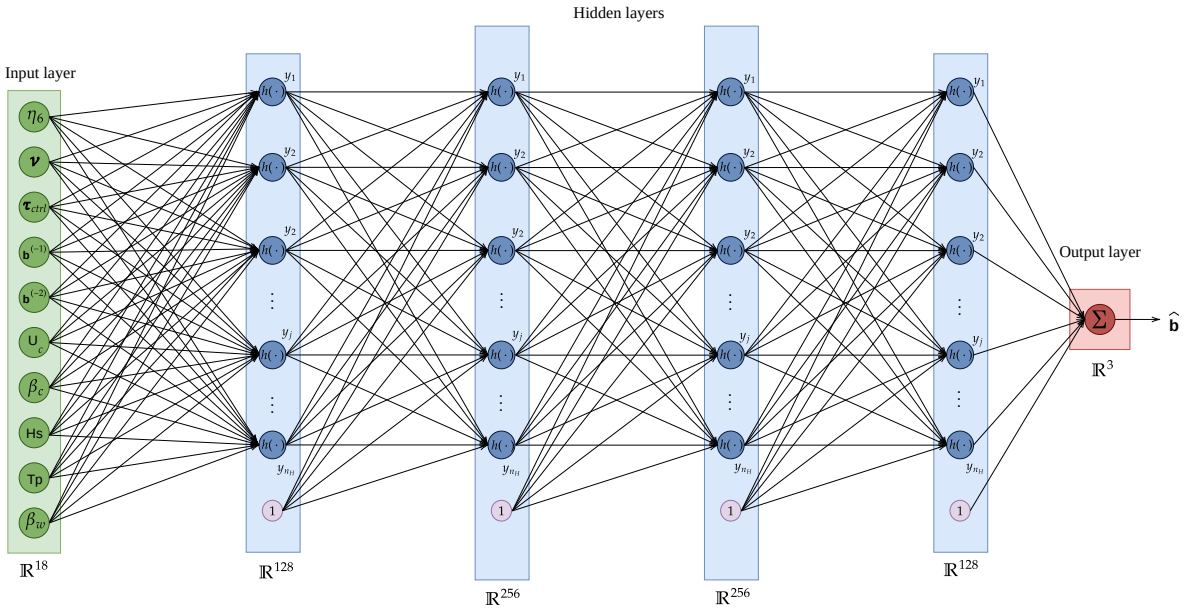


Figure 4.7: The fully developed neural network.

The bias model in Figure 4.7 can also be presented mathematically. The bias in the CDM in (2.48), which the network desire to predict, can be replaced with $\boldsymbol{W}^{*\top}\boldsymbol{H}(x)$. $\boldsymbol{H}(x) : \mathbb{R}^N \to \mathbb{R}^N$, is then the set of activation functions in the final hidden layer of the network, and $\boldsymbol{W}^* \in \mathbb{R}^{N\times 3}$ is the ideal set of weights in the final layer that ensures that the network perfectly predicts the actual bias. This representation assume that the bias signal can be perfectly reconstructed with N set of a given activation function.

An estimate for the bias is then introduced as

$$\hat{\mathbf{b}}(t) = \boldsymbol{W}^\top \boldsymbol{H}, \tag{4.33}$$

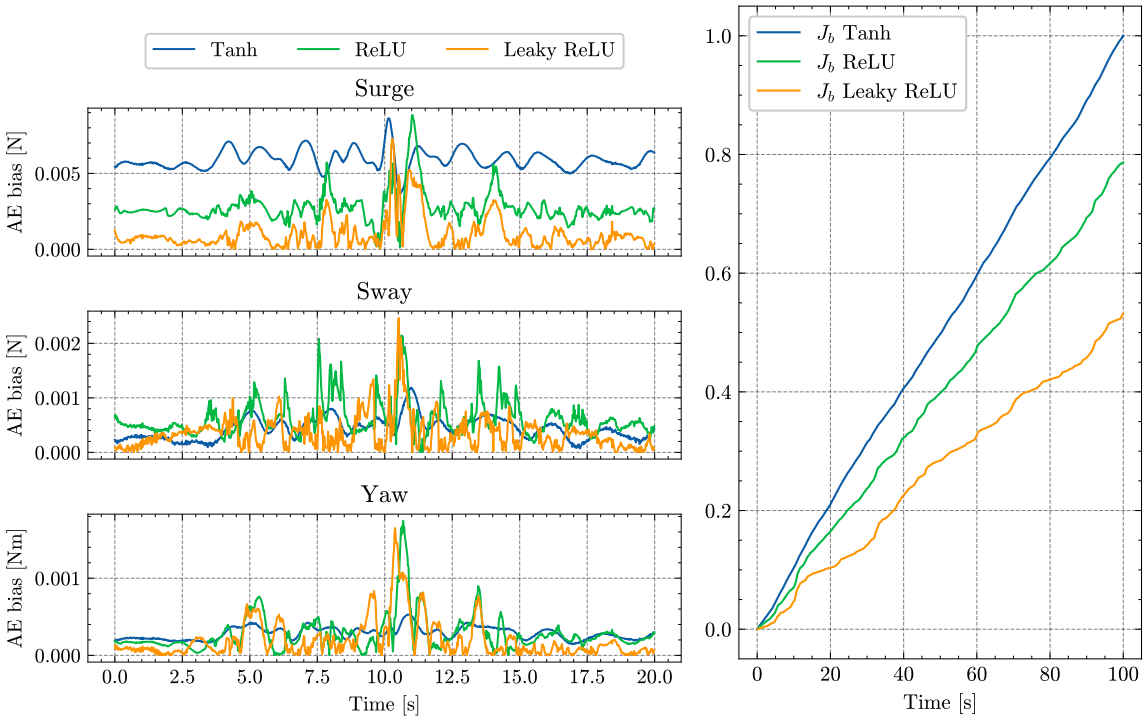
where $\boldsymbol{W} \in \mathbb{R}^{N\times 3}$ is the trained set of weights in the network, such that the true bias can be represented as

$$\mathbf{b}(t) = \boldsymbol{W}^\top \boldsymbol{H} + \boldsymbol{\epsilon} = \hat{\mathbf{b}}(t) + \boldsymbol{\epsilon}. \tag{4.34}$$

$\boldsymbol{\epsilon} \in \mathbb{R}^3$, denotes the approximation error of the estimate. Through sufficient adjustment of the weights, it is then desirable that the trained set of weights, $\boldsymbol{W}$, approach the ideal value, $\boldsymbol{W}^*$, such that the estimation error, $\boldsymbol{\epsilon}$, tend towards zero, and estimated bias, $\hat{\mathbf{b}}(t)$, converges towards the true bias, $\mathbf{b}(t)$.

One can notice the resemblances in the shape of the mentioned bias estimate in (4.33), and the previously introduced bias model in (4.8) from Section 4.4. By treating the activation functions as the regressor, $\boldsymbol{\Phi}(t)$, and the weights as adaptive coefficients, $\boldsymbol{\Theta}$, an alternative adaptive model is formulated. Utilizing, for instance, the beneficial traits of radial basis functions, as explained in Section 2.3.2, a new radial

basis function adaptive law can be employed. Similar studies, involving radial basis function neural networks in marine control system, is performed by Li et al. (2021), and Du et al. (2013). Both papers present a robust adaptive neural network control scheme for DP operations in the present of both model and disturbance uncertainties, displaying promising results.



(a) Absolute error in the bias estimate in surge, sway, and yaw, for the activation functions Tanh, ReLU, and Leaky ReLU.

(b) Normalized bias estimation KPI from (4.5d) for the activation functions Tanh, ReLU, and Leaky ReLU.

Figure 4.8: Study evaluating how different activation functions in the hidden layers of the neural network estimate the bias loads.

Table 4.2: Bias estimation KPI for different activation functions. The indicators are normalized such that the worst performing function has a value of 100.

Activation function	J_b
Tanh	100.0
ReLU	78.7
Leaky ReLU	53.2

The neural network has been implemented with the same modular approach as the simulation platform only in a separate source code. The network defines a base class for the layers, named *Layer*. From this, both dense layers, a class called *Dense*, and activation layers, a class named *Activation*, inherit functionality. Further, the activation class defines common functionality for all activation functions and serves as a parent lass for the different activation functions implemented. The neural network itself is defined by a separate class, named *NeuralNetwork*. This class is built up by layers from the layer-classes and defines the total network and its general functionality. Utility functionality such as loss functions has been implemented. Additionally, functionality from the python package PyTorch, supplemented our neural network analysis.

4.5.1 Further enhancement of the bias estimate

The performance of the neural network is intricately tied to its training process, and the accuracy of the training data. As demonstrated in (4.3c), in the initial example at the very start of this chapter, if given a well-established, high-fidelity simulation model, it is possible to accurately extract the residual bias load from simulations. This load can subsequently serve as training data, when fitting the weights in the network. If trained appropriately, the network will be capable of estimating the bias loads impeccably, thereby achieving an optimal bias feedforward mechanism. This approach, however, may be deemed as somewhat artificial from a real-world perspective. Inaccuracies in the simulation model, compared to the actual vessel, particularly in the hydrodynamic parameters, would most likely result in highly unreliable and unrepresentative bias loads. The accurate acquisition of environmental states, such as significant wave height and current velocity, is also a demanding task. In general, this represents one of the most difficult challenges related to disturbance estimation, as attaining a credible bias signal is complex and highly elusive.

As the acquisition of a precise bias signal is vital for the performance of the neural network, inaccuracies in this signal would inevitably lead to reduced estimation accuracy and performance. In light of these challenges, the implementation of the controller will, in this paper, be limited to a simulation stage. Nevertheless, to generate credible simulation results, these inaccuracies can be simulated by introducing subtle offsets in hydrodynamic parameters in the simulation model, as well as marginally imprecise estimates of environmental conditions. To evaluate the importance of accuracy in these parameters, two more objectives are presented, to be evaluated in the upcoming simulation study in Chapter 5.

- Evaluate how inaccuracies in the environmental conditions affect the performance and robustness of the neural network bias estimation.
- Evaluate how inaccuracies in the training model, as well as in hydrodynamic parameters, affect the performance and robustness of the neural network bias estimation.

That being said, the bias estimation could also be further enhanced by introducing acceleration measurements in the training process. By utilizing acceleration measurements, the neural network would enable the calculation of the specific forces acting on the vessel through the application of Newton's second law of motion. In the event that these force measurements are obtained, filtered, and processed, all residual loads in the CDM could be predicted more precisely. These loads can be incorporated into the bias term, and presents a valuable opportunity to enhance the training data for a neural network, thereby improving the provided estimates from the network. Acceleration measurements are, however, seldom used in marine control theory due to the not so trivial measuring, filtering and decoupling process. Nonetheless, Skjetne and Kjerstad (2016) introduce these specific force measurements in a control system design, by the use of four distinct inertial measurement units. The main contribution of the paper is the novel method of integrating the dynamic acceleration as an acceleration feedforward mechanism in the control law, providing promising results. Fossen (2021) also introduce the concept of specific force measurements into marine control system approaches, demonstrating that the inclusion of acceleration measurements can be a viable and valuable addition to marine control systems.

Another possibility to enhance the feasibility of incorporating the controller within an actual vessel can be to further improve the available models. Recent studies, conducted by both Wang et al. (2022) and Kanazawa et al. (2022), at NTNU in Ålesund, Norway, propose a physics-data cooperative modeling approach, to further improve the vessel models. These papers introduce a hybrid model, constructed by a deterministic model derived from hydrodynamic principles, merged with a data-driven, black-box model, learned from the observed vessel states. The prior knowledge given by the numerical model of a vessel, is integrated into a neural network as informative input, so that the informed neural network calibrates the bias between model outcomes and actual states in principle. This is in general a positional bias, but synergies may be drawn to the bias load already discussed in this thesis, as both represent a bias between a mathematical model and real-world observations. The papers showed promising results, concluding that a hybrid model yielded a more accurate model with relaxed data requirements and less learning consumption. A similar approach could also be made for bias loads. For instance through implementing a neural network to estimate the bias in a rough simulation model, given real-time specific

force measurements from sensors onboard an operating vessel. This has, nonetheless, not been a part of this thesis, due to temporal constraints.

4.5.2 Tuning

The controller utilizes the same control law as in (4.7). Consequently, the same procedure as described in Section 4.3.1, also apply for this controller.

Chapter 5

Simulation results and discussion

The following chapter will present an overview of the simulation results obtained for the four control techniques derived in the previous chapter. The controllers are evaluated, both in a steady state, stationkeeping operation, and in a 30°-30° heading test. The stationkeeping test is performed to evaluate the controllers' ability to maintain a predetermined setpoint under given environmental load conditions, serving as the fundamental test before the subsequent maneuver is examined. The 30°-30° heading maneuver is more related towards the main objective of the thesis, and focuses on evaluating the controllers' performance during transient operations. Both tests are mainly performed in the predefined sea state, as presented in Table 4.1.

The test procedures in the this chapter is in accordance with the strategies outlined in Section 2.6. Firstly, the simulation model and the simulated environment were tested with MIL tests to validate the response of the vessel's response to environmental forces. An overview of the test design for the MIL tests are illustrated in Appendix B. 1.

The proposed control algorithms were then tested with a SIL approach, as seen presented in Appendix B. 2, yielding valuable insight into the disturbance rejection characteristics. These insights guided the selection of controllers for further investigation in an experimental lab setup. The most promising controllers were eventually tested in the MC-lab.

Section 5.1 presents the result obtained from the stationkeeping test. The $30° - 30°$ heading change scenario is presented in Section 5.2. Ultimately, in Section 5.3, a more in-depth analysis and discussion, related to all presented objectives in the problem formulation, is provided. This section also includes a sensitivity analysis of key parameters, both for the adaptive, and neural network controller parameters.

5.1 Test scenario 1 - Stationkeeping

The following section displays the result of the stationkeeping test scenario. The test was conducted for 1750 seconds, and precautions were taken to ensure that the vessel had attained a steady state before commencing the test. Figure 5.1 presents the surge position of the vessel for the four different control methods. As it can be observed, very little separate the performance of the four methods. All controllers were capable of obtaining the setpoint, although some low-frequency motion might be detected. Figure 5.2a and Figure 5.2b present the corresponding bias rejection term in surge, for all the controllers in a 100 seconds interval, as well as their absolute error compared to the true bias term. The neural network bias estimate, visible as control method 4 in Figure 5.2a, is trained on the assumption of a perfect model, and can thus be considered very close to the true bias, as evidenced by the low absolute error.

Table 5.1 exhibit the previously presented KPIs during the test scenario in a tabular format. All KPIs have been normalized such that the worst performing control method had a value of 100. With the

exception of the bias estimation KPIs, it is evident that there is minimal difference in the performance indications among the four control methods. Method 3, being the adaptive controller, displayed the worst overall performance in all performance indicators, which also marginally can be seen in the response, in Figure 5.1. That being said, very little separateed the performance of the four controllers after a steady state had been reached. For the bias estimation KPI, nonetheless, the neural network control estimate outperformed the other controllers significantly.

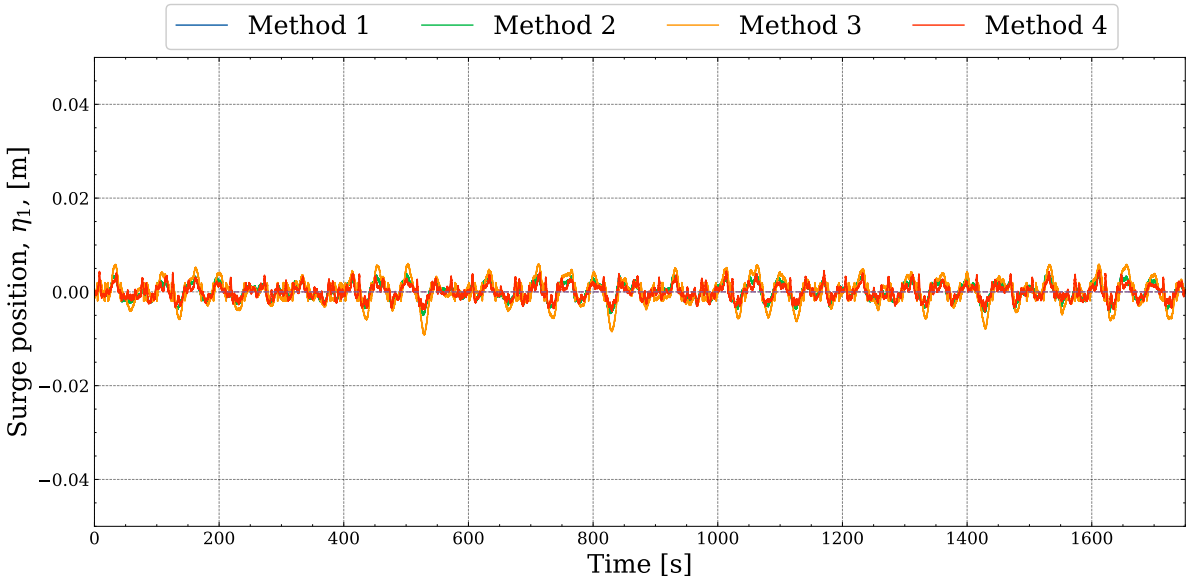
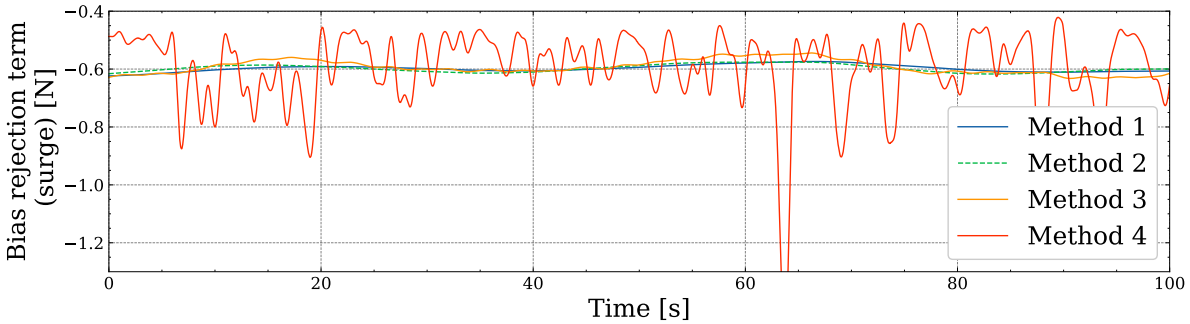
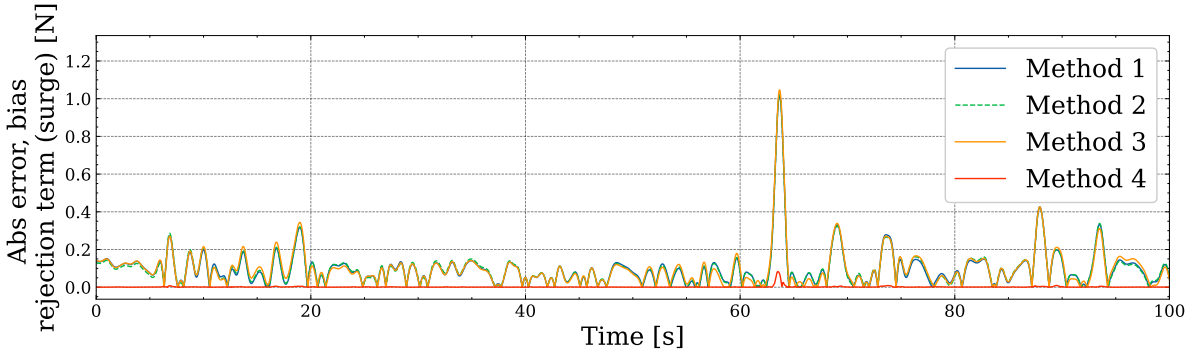


Figure 5.1: Vessel response for all control methods during stationkeeping test scenario.



(a) Bias rejection term.



(b) Absolute error between the estimated bias and true bias.

Figure 5.2: Bias rejection in surge for all controllers, in a 100 seconds interval, during stationkeeping.

Table 5.1: Key performance indicators for steady state stationkeeping. The KPIs are normalized for every cost function, such that the worst performing control method has a value of 100.0

Controller	J_η	$J_{\tau,uv}$	$J_{\tau,r}$	$J_{b,xy}$	$J_{b,\psi}$
Method 1	99.7	94.3	93.3	98.8	93.4
Method 2	99.3	94.3	95.8	98.9	89.2
Method 3	100.0	100.0	100.0	100.0	100.0
Method 4	98.9	95.1	95.6	1.8	48.3

These KPIs can also be visualized graphically, as in Figure 5.3. The indicators are then normalized such that the worst performing controller has a maximum of 1 for the whole test scenario. Using a weighted sum of the previously defined KPIs, the indicators are combined into three separate KPIs, evaluating performance, effort, and bias estimation, respectively. The KPIs coincide well with the previously displayed performance of the controllers, showing minimal difference, both in performance and effort. As stated, control method 3 exhibited the poorest performance, while control method 4 vastly outperformed the other methods in estimating the bias loads.

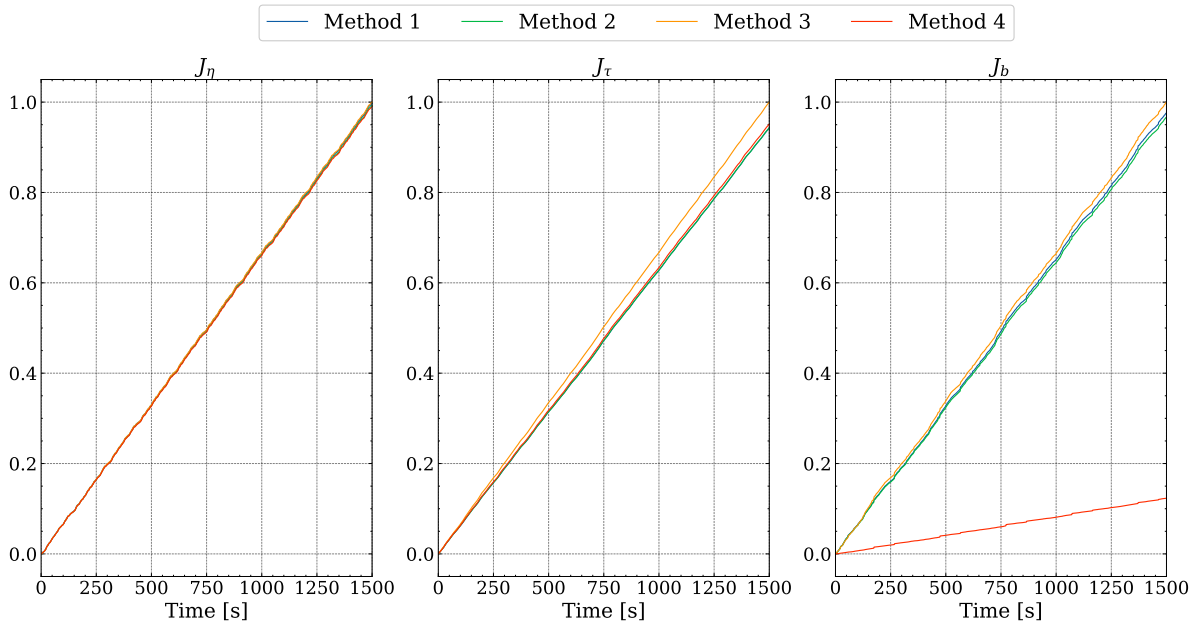


Figure 5.3: Key performance indicators for test scenario 1. The KPIs are normalized such that the worst performing controller has a maximum of 1 for the whole test scenario.

Even though some variations may be observed in the presented figures, it is reasonable to assume that all controllers demonstrate sufficient capability for stationkeeping in head sea. This assumption is fundamental for the upcoming section, and serves as the basis for further analysis and comparison of the controller performances.

5.2 Test scenario 2 - 30°-30° heading change

Having evaluated the controllers' ability to maintain a given setpoint, it can be proceeded to the second test scenario. This section presents the results from the 30°-30° heading change maneuver for all four control methods. The test consisted of three setpoint changes which introduced transients to the test. It started with the vessel at zero heading, followed by a change to 30° and -30° heading respectively,

before finally returning to the initial zero heading setpoint. The test lasted for a total of 2500 seconds, and a setpoint change was initialized at $t = 250$ s, $t = 1000$ s, and $t = 1750$ s. Again, precautions were taken to ensure that the vessel had attained a stationary performance before commencing the test.

Figure 5.4 presents the heading, and bias rejection term in sway, for all four controllers, during the entire maneuver. It can be observed that all controllers, to a certain degree, were capable of following the desired trajectory. Large similarities can also be seen in the compensated bias, even though some controllers displayed a more rapid, or noisy, behavior.

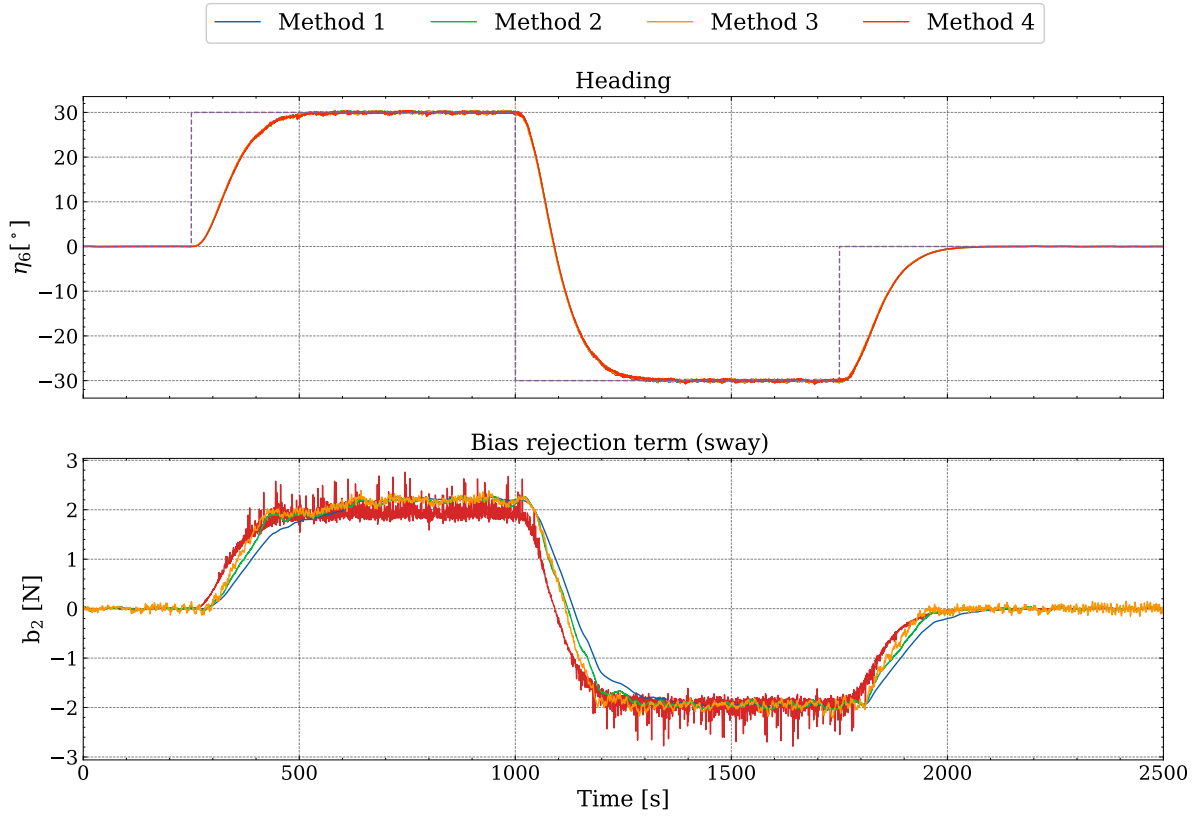


Figure 5.4: Heading response, and bias rejection term in sway, for all control methods during a 30°-30° test scenario.

Figure 5.5 shows the same bias rejection plot in sway, as seen in Figure 5.4, but zoomed in on the final transient phase, occurring from approximately $t = 1750$ s to $t = 2100$ s. The absolute error, and normalized cumulative error are also indicated on the right. This illustrates how the four controllers performed during more rapid changes in the residual loads. It can be seen that control method 1, being integral action, displayed the slowest performance. Yet again it is worth noticing that the neural network bias estimate (Method 4) provided close to perfect bias compensation.

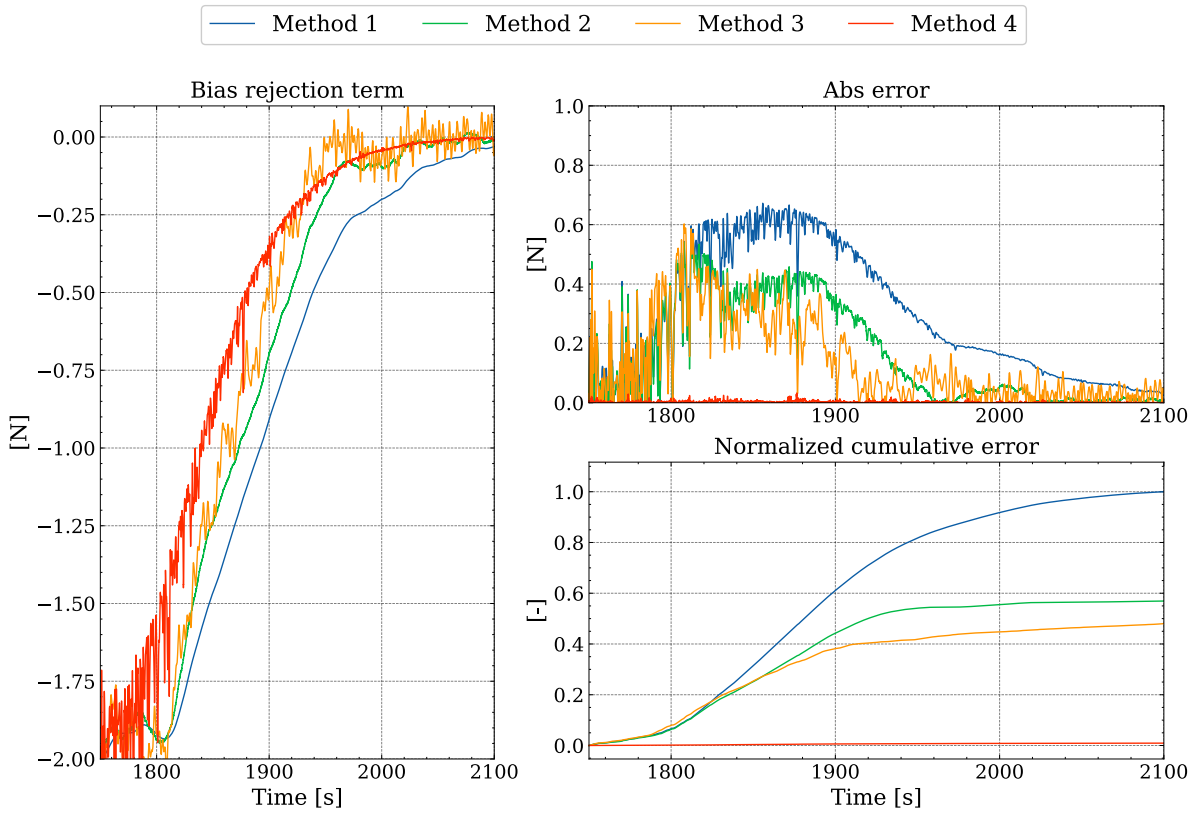


Figure 5.5: The left plot display the bias rejection terms in sway, from Figure 5.4, in the time interval $t = 1750$ s to $t = 2100$ s. The right hand plots display the absolute error and cumulative error, when comparing the bias estimates with the true bias.

Similarly to the stationkeeping test scenario, the controllers behavior, this time only for the transient interval presented above, can be summarized in a set of KPIs. This is shown graphically in Figure 5.6.

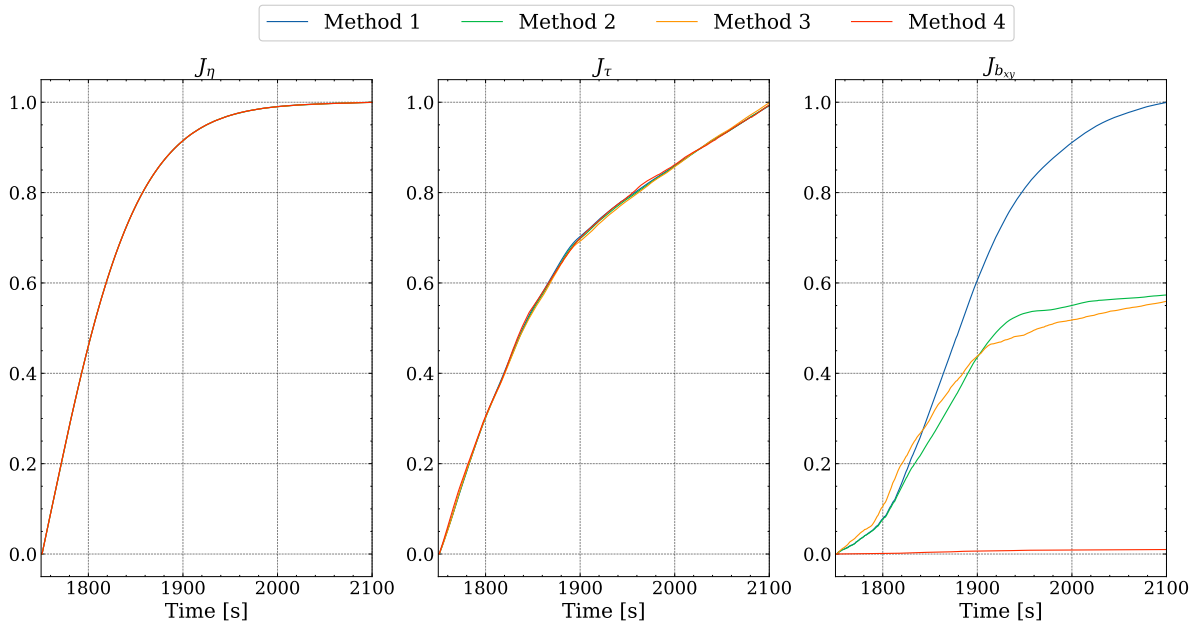


Figure 5.6: Key performance indicators for test scenario 2 in the interval $t = 1750$ s to $t = 2100$ s. The KPIs are normalized such that the worst performing controller has a maximum of 1 for the whole test scenario.

5.3 Analysis and discussion

With the presentation of the simulation study results, the focus now shifts to their analysis and in-depth discussion in the following sections. As a reminder for the reader, the main objective of the analysis is restated.

- Explore the effectiveness of alternative compensation techniques, in comparison to traditional integral action, for mitigating response from low-frequency (not necessarily zero-frequency) wave load components.

The findings presented in Section 5.1 justified that all control methods were capable of maintaining a predetermined setpoint under the influence of environmental loads. This served as a basis, and could be considered a minimum requirement before any further analyses were conducted. According to the performance indicators from the stationkeeping trial, control method 3, being the adaptive Fourier series controller, showed a slightly worse performance, compared to the others, although differences were minimal. As expected, control method 4, being the neural network controller, completely outperformed the others in regards to bias estimation. As the controller was trained on the exact bias calculations from the same simulation model, these results can be considered somewhat artificial, and would be challenging to obtain in a more realistic, real-life scenario. The controller's robustness to such errors are discussed further in the upcoming sensitivity analysis. The calculated bias signal may also be considered as too noisy to be implemented into a DP control system, and precautions should be taken to overcome this challenge.

Test scenario 2 introduced transients into the maneuver. It was, thus, well-suited for evaluating the effectiveness of the alternative bias compensation techniques, as the residual loads changed considerably during the trial. Again, very little distinguished the controllers with regards to performance and effort. This can be seen, both in the heading response, in Figure 5.4, and in the KPIs, in Figure 5.6. That being said, dissimilarities aroused in the bias rejection term for the controllers. Figure 5.5 clearly illustrates these differences through an observable error from the true bias. The integral action-based bias compensation exhibited the slowest reaction, and did, thus, have the largest cumulative error for the setpoint change. This can be seen as a consequence of the inertia in the integral term in the control law, which require that an error must occur, and be accumulated over time. As mentioned in the implementation, the integral term is also often tuned relatively slow in DP systems, to avoid overshoot and instability. Bias estimates from the observer outperformed the integral action in transient behavior, and did, in this case, provide a better bias rejection.

Furthermore, the Fourier series based bias model, demonstrated a more rapid bias rejection than both the previous methods. However, it exhibited a significantly noisier behavior. This could possibly be explained through the adaptive update law, in (4.31), stating that $\dot{\hat{\boldsymbol{\Theta}}} = \boldsymbol{\Gamma}\boldsymbol{\Phi}(t)\boldsymbol{z}_2$, as derived in Section 4.4. If either $\boldsymbol{\Gamma}$ is tuned too rapidly, or the velocity estimates in $\boldsymbol{z}_2$ include wave-frequency components, a noisy behavior can be expected, as the coefficients in $\hat{\boldsymbol{\theta}}$ would struggle to achieve their optimal value. It can be reasonable to assume that the latter is the cause of the noisy behavior observed in Figure 5.5, as the bias rejection term inhibited some wave-frequent oscillations. An even better tuned observer could potentially eliminate such oscillations. A sensitivity analysis of $\boldsymbol{\Gamma}$ is performed below, and its impact on the response will be addressed there.

The neural network controller had close to zero absolute error, and could thus be considered as almost identical to the true bias. This control method could, consequently, be looked upon as close to perfect bias rejection. This is visible in the bias KPI, displaying close to zero error. Nonetheless, the overall performance of the controller did not differ much from the other rejection techniques.

5.3.1 Sensitivity analysis

Sensitivity analyses were conducted for parameters both in the adaptive controller and neural network controller. This was to evaluate their robustness, with the aim of achieving the presented objectives,

stated in Chapter 4. The analyses provided a broader understanding of the significance of certain parameters, by systematically varying their value, while keeping all remaining factors constant.

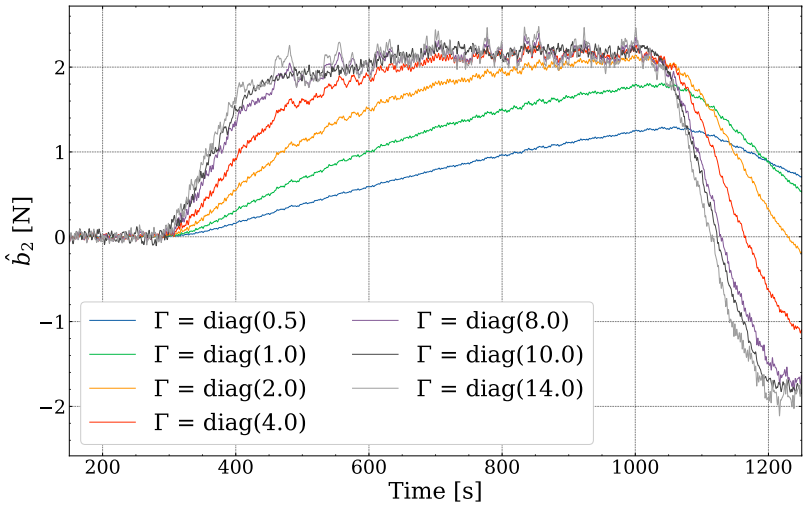
Adaptive controller

The objectives for the sensitivity analysis for the adaptive controller were, as previously stated in Section 4.4:

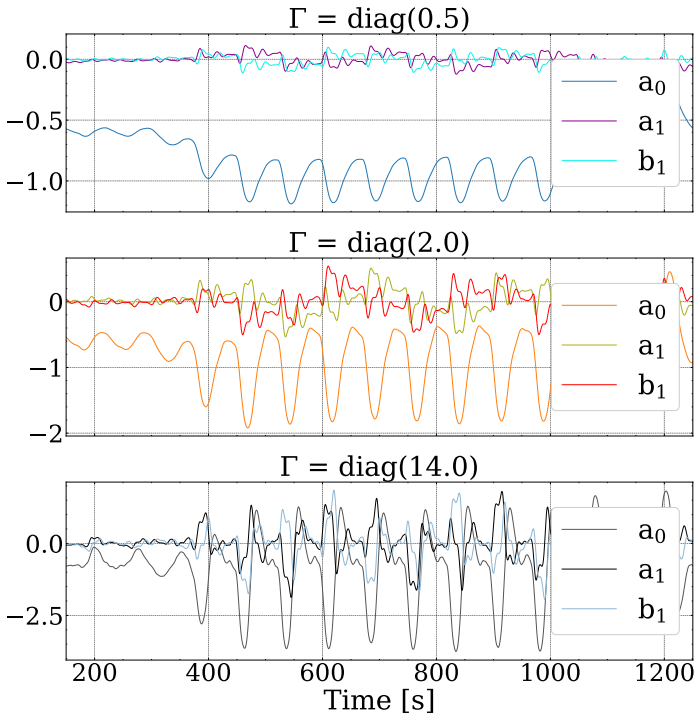
- Evaluate how the performance of the adaptive bias model is affected by its tuning matrix, $\mathbf{\Gamma}$, and how the magnitude of the gain works as a trade-off between a rapid estimation and introduction of oscillations.
- Evaluate how varying number of frequency components in the adaptive bias model impact the performance of the estimated bias.

The gain factor, $\mathbf{\Gamma}$, is crucial in the adaptive update law, as it governs the rate of which the adaptive parameter, $\hat{\mathbf{\Theta}}$, is updated. It is an important parameter that influences speed, stability, convergence, and tracking performance of the adaptive control system.

Figure 5.7a illustrates the bias rejection term in test scenario 2, performed for multiple values of $\mathbf{\Gamma}$. A clear correlation between the magnitude of the gain factor and the rate of the bias rejection term can be observed. As expected, it can also be seen that an increase in bias compensation rate led to an intensified signal variance. This is a direct trade-off between speed and stability, and can be compared to the gain parameter in a conventional integral action controller. The underlying cause of the increased noise in the bias rejection term is evident in Figure 5.7b. As $\mathbf{\Gamma}$ increased, the adaptive parameter, $\mathbf{\Theta}$, struggled to obtain a steady state value, resulting in large oscillations. As previously stated, the transformed vessel state was also included in the adaptive update law. Noisy signals from the observer could, thus, influence the limit of how rapidly $\mathbf{\Gamma}$ could be tuned, and decrease the rate of which the residual loads were compensated for.



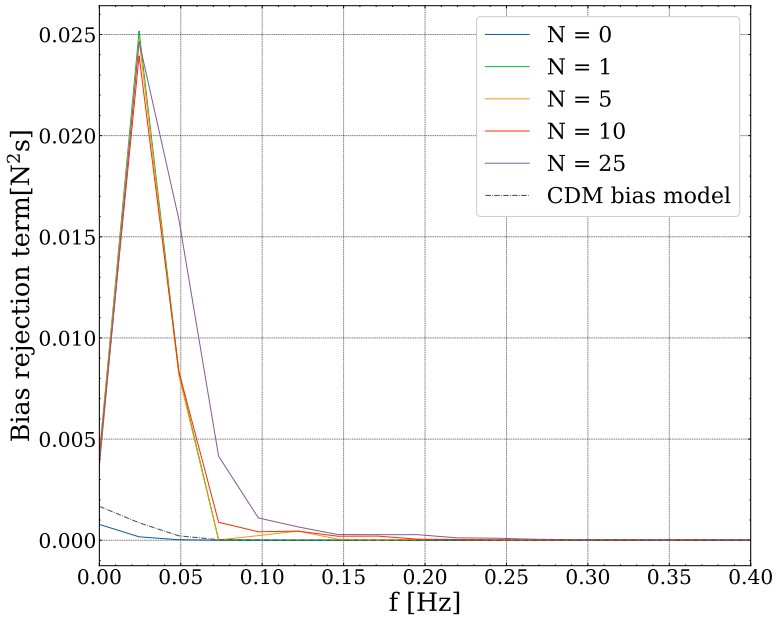
(a) Bias rejection term (sway).



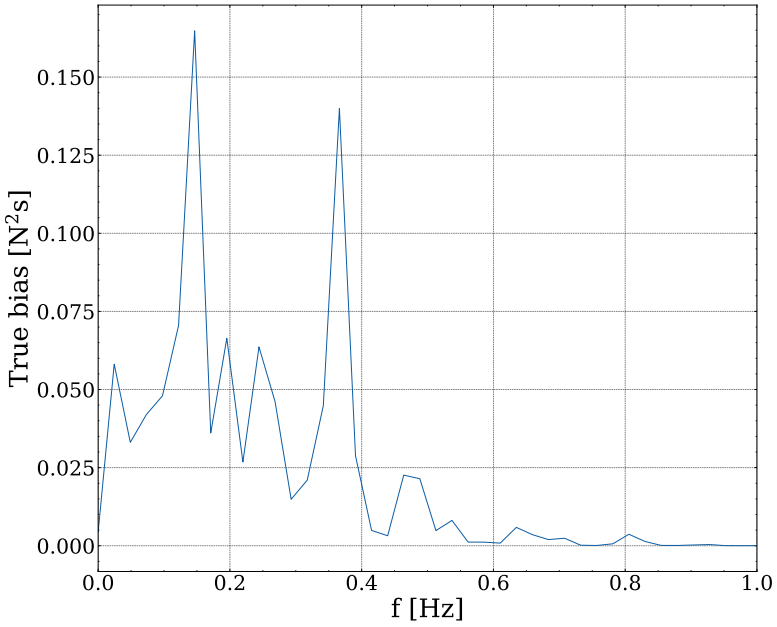
(b) The first three components of the adaptive parameter, $\boldsymbol{\Theta}$.

Figure 5.7: Sensitivity plot for test scenario 2, plotted for different values of $\boldsymbol{\Gamma}$.

The internal bias model implemented in the adaptive controller also required a finite number of frequencies, N. Figure 5.8a displays the power spectral density of the bias rejection term for a various selection of N. Very little separated the different graphs, and it can be seen that close to no frequencies above $f = 0.15$ Hz were excited. Figure 5.9 presents how the adaptive coefficients behaved for N equal to 0, 1, and 5. For $N = 5$, it can be seen that all higher-frequency coefficients stabilized at zero, diminishing their associated frequency. This, despite the fact that Figure 5.8b proved that these frequencies in fact were present in the residual loads. This can be said to be a trend for all simulations, as the adaptive update law in general struggled to maintain a persistent level of excitation for its upper bound of frequencies. This could be due to various underlying reasons. The bias term was, for the given environmental conditions, dominated by the current load. As this load, being a strong and consistent disturbance, was present during the entire excitation period, it might have dominate the vessel's response and limited the exploration of other dynamics. Further investigations should, consequently, be conducted.



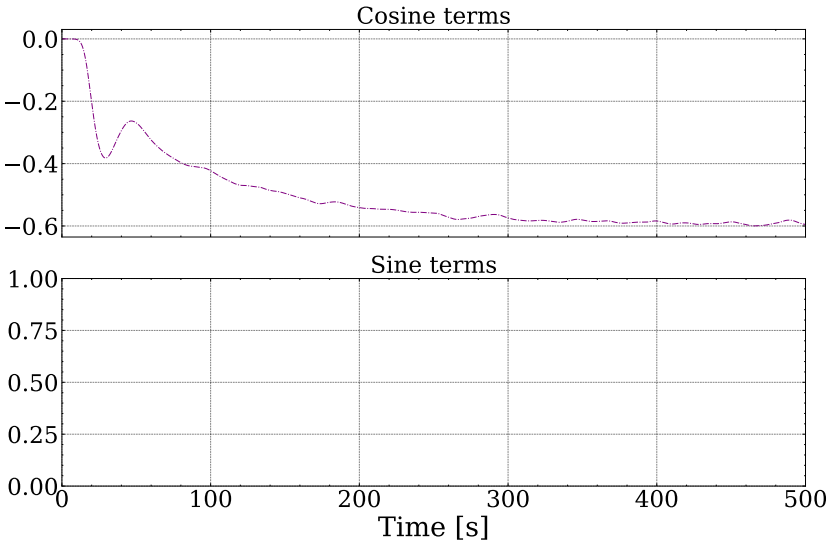
(a) $N = 0$ to $N = 25$.



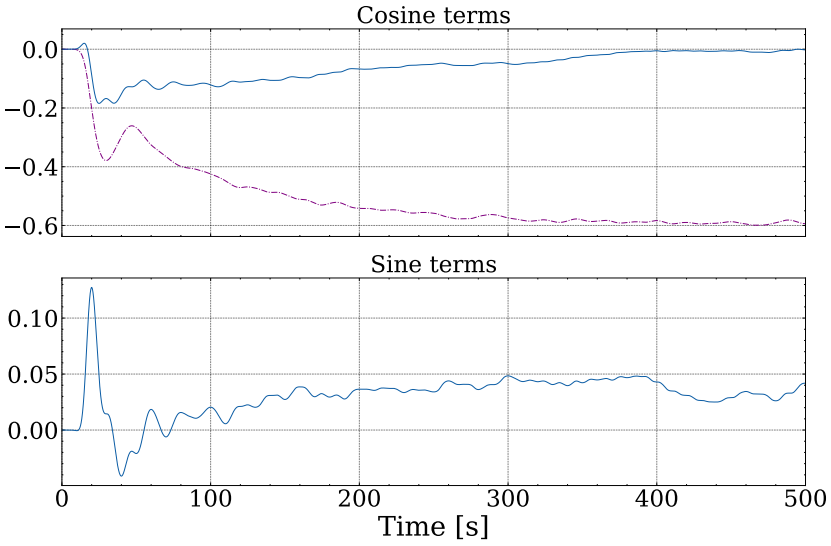
(b) True bias term.

Figure 5.8: Power spectral density of the bias rejection terms for various values of N.

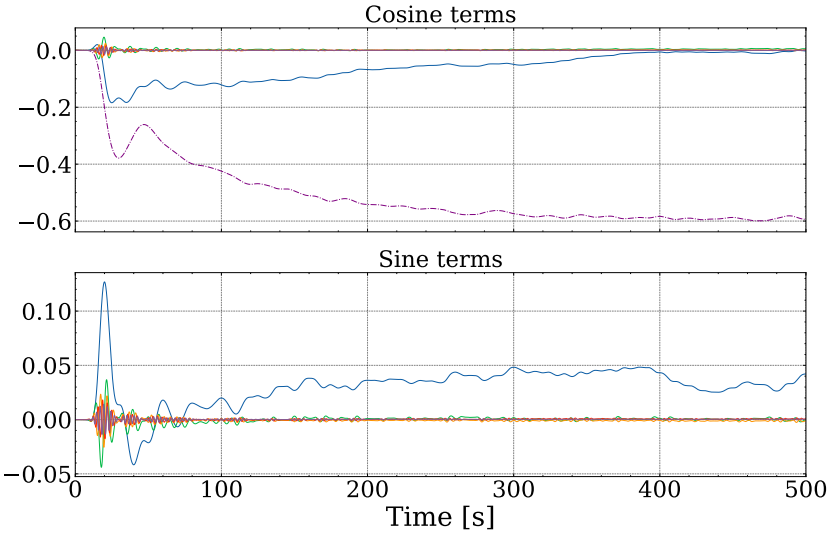
Although little separated the graphs in Figure 5.8a, a difference can be seen for $N = 0$. This case should, theoretically, only be considered as a constant bias model. Resemblances can be therefore be drawn to the bias model utilized in the observer CDM, in (2.48), which also is modelled as constant. This is evident in Figure 5.8a, where the spectral density of the observer bias, employed as a bias feedforward term in control method 2, also is plotted. The similarities to the constant bias model for $N = 0$ are apparent, with both graphs distinctly differing from the remaining spectra.



(a) $N = 0$



(b) $N = 1$



(c) $N = 5$

Figure 5.9: Adaptive coefficients in $\boldsymbol{\Theta}$, for $N = 0, 1$ and 5.

Neural network controller

In Section 4.5 two objectives for the analysis of the neural network controller were formulated. As for the adaptive controller, these are restated.

- Evaluate how inaccuracies in the training model, as well as in hydrodynamic parameters, affect the performance and robustness of the neural network bias estimation.
- Evaluate how inaccuracies in the environmental conditions affect the performance and robustness of the neural network bias estimation.

Section 5.1 and Section 5.2 presented results of a neural network controller under *perfect* conditions. This controller was trained on data obtained in the same conditions as in the test it was evaluated for. However, in a more realistic scenario, such ideal conditions will never be met. Hence, it is of interest to assess the bias prediction in more realistic scenarios. The subsequent paragraphs present two such scenarios. The first involves evaluating the controller's sensitivity to various sea states, and the other considers imperfections in the CDM.

As mentioned in Section 4.5, neural network bias estimation relies on sea state parameters as input. Therefore, if the neural network controller designed in this thesis were to be implemented on a real system, the vessel would require measurement systems to precisely estimate these input parameters. An example of such a measurement system is an X-band radar. Understandably, the availability and accuracy of the sea state input parameter estimates may vary significantly depending on the quality of the vessel sensors. To evaluate the robustness of the bias prediction, it was therefore important to assess how robust the controller was for inaccuracies in the sea state estimation.

To evaluate the sensitivity to sea state changes, the variation of two relevant input parameters, Hs and Tp, was examined. The network used in the sensitivity analysis was pre-trained on the Seastate 2 presented in Table 4.1. The test was then performed by evaluating the bias prediction for simulations with a range of different values for Hs and Tp. However, the network was, for each prediction, fed the same values, Hs = 0.03 m and Tp = 1.5 s, to mimic an estimation error.

The KPIs for the bias prediction for the different simulations are presented in a contour plot in Figure 5.10. The sea state for which the network has been trained for is marked with a black marker. This region in the plot corresponds to a relative low KPI value, indicating that the prediction for these values of Hs and Tp is relatively good. However, other combinations of Hs and Tp values results in even better KPIs, such as Hs = 0.00 m and Tp = 1.0 s, and Hs = 0.01 m and Tp = 1.6 s. The worst KPI value is found for Hs = 0.01 m and Tp = 1.1 s. It is challenging to observe any trends from Figure 5.10, and additional combinations of Hs and Tp should be included. That being said, as the contour plot displays distinct regions with a larger prediction error, it is clear that the estimated sea state parameters, utilized as input in the network, play a significant role in the overall bias compensation.

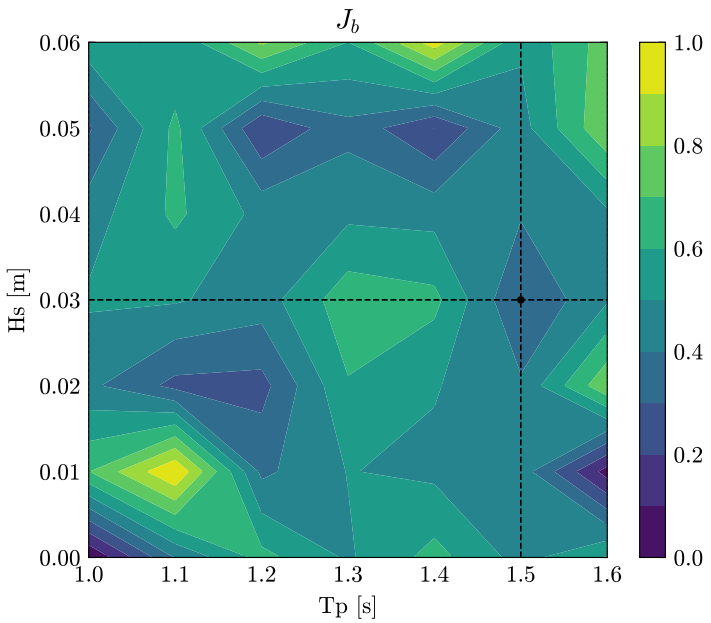


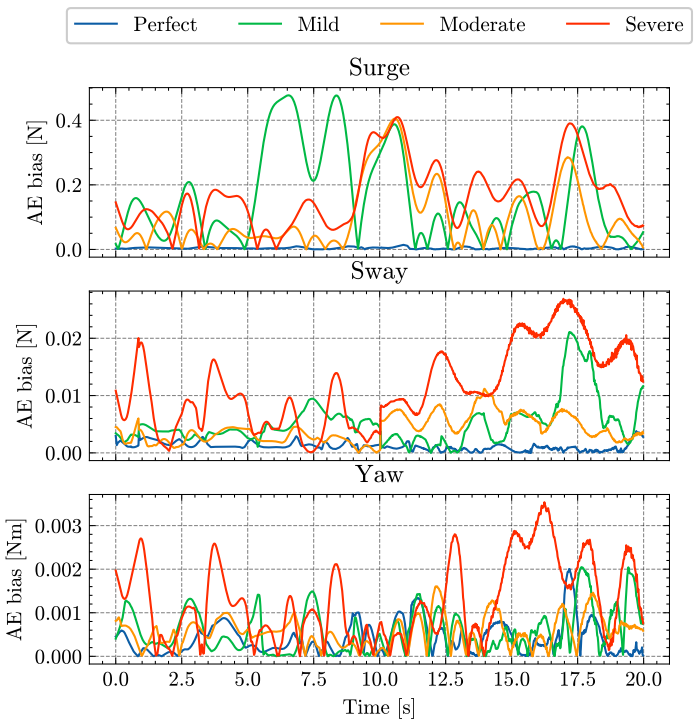
Figure 5.10: Normalized KPIs for bias prediction from the neural network model, performed for a range of different Hs and Tp.

When predicting the bias term with the inputs stated in Section 4.5, it is assumed that the CDM is ideal. This assumption implies that the hydrodynamic terms, such as damping and added mass, as well as the thrust allocation algorithms represent the true dynamics. The structure of the CDM itself is also assumed to accurately mimic the real world dynamics. This ideal scenario does not hold, and errors will occur in the CDM. These imperfections will cause variations in the input parameters for the network. To investigate how this will impact the network's predicted output, three levels of CDM imperfections, presented in Table 5.2, were introduced and tested on the network. These were all compared to the predictions of the bias when using a perfect CDM. It should be noted that this is only a small robustness test, and that more extensive testing should be performed to further validate the neural network algorithm.

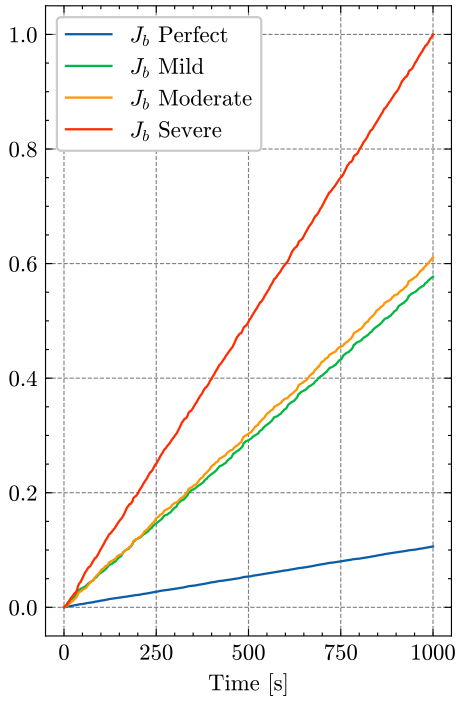
Table 5.2: Defined levels of errors in the control design model.

Error source	Mild error	Moderate error	Severe error
Hydrodynamic damping, $\boldsymbol{D}$	5 %	20 %	40 %
Hydrodynamic added mass, $\boldsymbol{M}$	5 %	20 %	40 %
Thrust allocation	5 %	15 %	25 %

The absolute error in the prediction of the bias in surge, sway and yaw is presented in Figure 5.11a. From the plot, it is clear that the test where a perfect CDM was used predicted the bias term the best and that the tests with severe modeling errors in the CDM provided the worst predictions. However, it is difficult to distinguish between the mild and moderate error predictions. This can be observed more clearly when plotting the KPIs for the different tests, as seen in Figure 5.11b. As the plot clearly illustrate, the bias prediction performed by the neural network got worse with increasing errors in the CDM. The KPIs for all tests after 1000 seconds are summarized in Table 5.3.



(a) Bias rejection term.



(b) Absolute error between the estimated bias and true bias.

Figure 5.11: Bias rejection performance in the neural network controller for a given set of errors in the CDM.

Table 5.3: KPIs for different levels of errors in the CDM.

CDM error	J_b
Perfect	10.6
Mild	57.7
Moderate	61.1
Rough	100.0

5.4 Qualitative assessment of the control methods

So far, the analysis of the control methods has primarily been performed using quantitative measures, through the aid of key performance indicators. However, a qualitative evaluation has also been performed for the four controllers, considering factors that may be more difficult to measure quantitatively. The analysis yielded more insights into practical aspects, and feasibility of implementing the controllers in real-world applications. Criteria include factors such as ease of implementation, tuning process, robustness, and commercial feasibility. The evaluation is fully based on the procedures and use-cases experienced by the authors in this thesis, and can, therefore, be considered as subjective.

The commercial feasibility is a crucial aspect of the analysis. It encompasses factors such as scalability, and cost-effectiveness, and assess whether or not the controllers align well with commercial requirements and industry practices. This is particularly of interest for the newly proposed control methods, and can evaluate the likelihood of successful adoption and implementation in a real-world system. Control method 4, as presented in this thesis, is still far from being implemented in a real vessel. That being said, the integration of machine learning and neural network for residual load compensation can offer

numerous opportunities. Some of which were referenced in Section 4.5.1.

Another aspect that influence the commercial feasibility is the computational requirement of the different controllers. Figure 5.12 display the average computational time per calculated control load from the simulation model. Although none of the controllers demand tremendous computational time, it should be noted that the time of control method 4 will increase as complexity in the neural network increases. The control method would require an exceptionally more intricate network model in real-life scenarios, leading to higher perplexity, and consequently, computational time could become challenging.

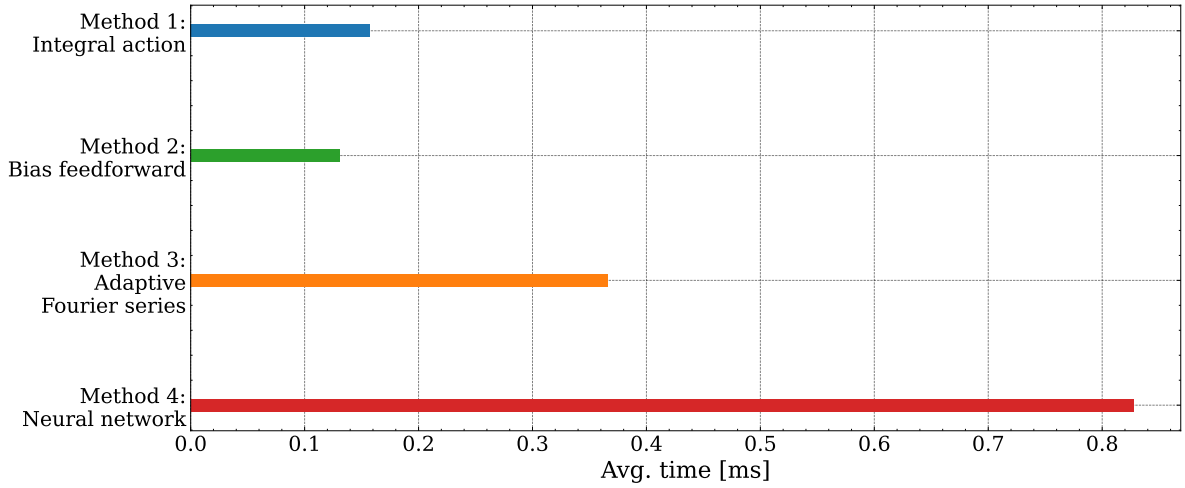


Figure 5.12: Average computational time per calculated control load for the different control methods.

In present DP systems, both method 1 and method 2 are widely known, and simple to implement. They can, thus, be expected to outperform the other proposed control techniques in certain qualitative criteria. Given their establishment in the industry, well-performing heuristic tuning methods have also been developed to simplify the tuning processes significantly. The same also applies to control method 4, which, likewise, utilize the same nominal PD controller in its control law. Method 1 is the only control method that can be regarded as entirely model-free, while method 2 and method 4 incorporate an internal model to estimate the bias. Method 3 fully relies on a model-based control law. A full overview of the performance indicators are given in Table 5.4.

Table 5.4: Qualitative PIs for the four control methods.

Qualitative PI				
Criterion	**Method 1**	**Method 2**	**Method 3**	**Method 4**
Ease of implementation	Excellent	Excellent	Good	Moderate
Tuning process	Excellent	Excellent	Moderate	Excellent
Disturbance rejection robustness	Moderate	Good	Good	Excellent
Model uncertainty robustness	Excellent	Good	Moderate	Moderate
Commercial feasibility	Excellent	Excellent	Good	Moderate

Chapter 6

Experimental testing and validation

The purpose of this chapter is to validate some of the results obtained from the conducted simulation survey. The validation is performed, first by utilizing HIL testing, before experimental model tests are carried out in the MC-lab (see Appendix B. 3 and Appendix B. 4). The tests aimed to verify the integration of the control systems and provide a comprehensive end-to-end assessment of the complete system.

ROS played a crucial role in creating an interface between the simulation platform and the hardware, and to facilitate for seamless communication between different components therein. The ROS interface can be seen in Figure 3.12. In accordance with the methodology presented in section 2.6, the source code was first tested by itself in a ROS environment utilizing the ROS wrapper methodology presented in Figure 3.11, before running the code in real-time on the actual Raspberry Pi hardware. X-in-the-loop procedure had thus been executed, thereby ensuring the vessel's readiness for laboratory testing.

The experimental testing was divided in two separate lab sessions. The first had the primary objective of evaluating the baseline controller, while familiarizing the authors with the lab setup. The second session aimed to validate some of the findings from the simulation study, particularly with focus on one of the newly proposed disturbance rejection approaches. Due to the unavailability of certain input measurements required by the neural network in control method 4, the adaptive controller was selected as the controller of interest.

The primary objective of the experimental laboratory tests was to replicate the observed trends from the simulation tests, thereby validating the simulation results. Specifically, the aim was to determine if the bias compensation in the adaptive controller exhibited faster response compared to the integral action term in the PID controller, as observed in the simulation study. This validation process aimed to enhance the confidence in the simulation results and assess the performance of the controllers in more realistic conditions.

Furthermore, to assess the performance of both controllers, specific tolerance boundaries were established for the stationkeeping offsets. It was determined that for a controller to be considered satisfactory, it should not exceed a tolerance offset of ± 0.05 m in position and $\pm 3.0°$ in heading. These tolerance limits served as benchmarks to evaluate the controllers' effectiveness in maintaining the desired stationkeeping position.

The results obtained from the experimental tests will be presented in the subsequent sections, providing insights into the performance and effectiveness of each controller. Following the presentation of results from a lab session, a discussion will be provided to analyze and interpret the findings. Potential impacts from error sources will also be discussed.

6.1 Lab setup and conventions

This section provides an overview of the laboratory setup and conventions utilized for the experiments conducted in the MC-lab.

Coordinate frames

Careful consideration were paid to coordinate frames within the lab. A fixed coordinate frame, established for the basin in the forthcoming tests, is illustrated in Figure 6.1. This is referred to as the basin-frame, $[X^{\text{basin}}, Y^{\text{basin}}, Z^{\text{basin}}]$. Additionally, the motion capture system also operated with its own local coordinate system, directing the vertical axis upwards. A rotation matrix was thus implemented, which rotated these measurement signals π rad around the horizontal x-axis, ensuring alignment with the defined basin-frame. The vessel naturally employed its own body-frame, as presented in subsection 2.4.2.

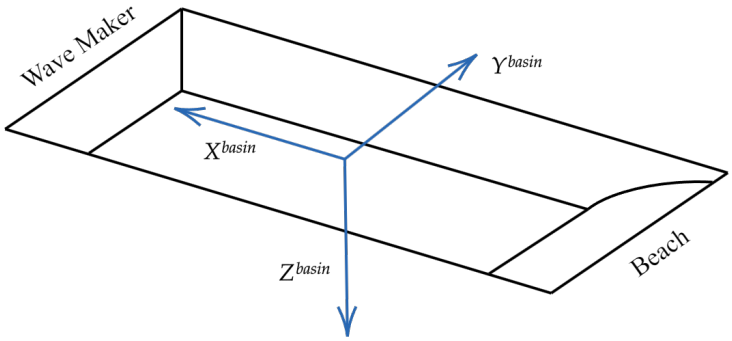


Figure 6.1: The basin frame defined in the MC-lab.

Wave generation

In all experimental tests presented in the upcoming sections, the wave maker was configured to generate waves corresponding to the Sea state 2 described in Table 4.1. This excluded the implementation of current, as the MC-lab does not have the functionality to generate it. Two wave probes were installed and placed both in front of, and behind the operating area of the vessel to capture the wave realization. These wave probes revealed that the wave maker generated a somewhat inconsistent wave field. This discrepancy is important to consider when evaluating the experimental results. Figure 6.2 presents wave spectra obtained from the wave probes for an arbitrary test run, in addition to the associated, theoretical JONSWAP spectrum. Minor fluctuations can be observed. These fluctuations can also be assumed to increase during the duration of the sea state, as error sources such as wall effects, and an imperfect beach, influence the wave realization.

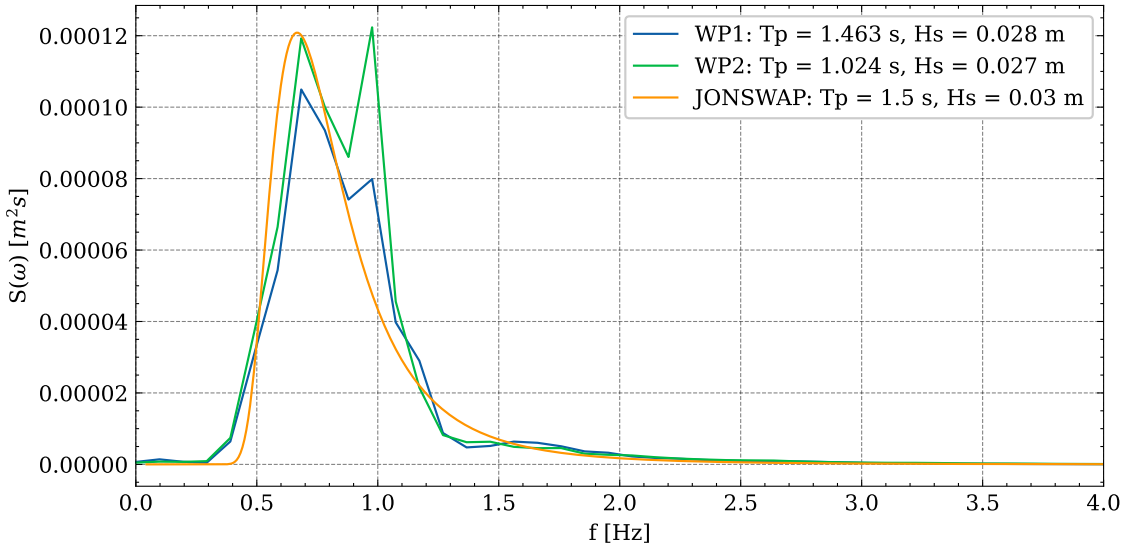


Figure 6.2: Wave spectrum measured in the MC-lab from two distinct wave probes, compared to the theoretical JONSWAP spectrum.

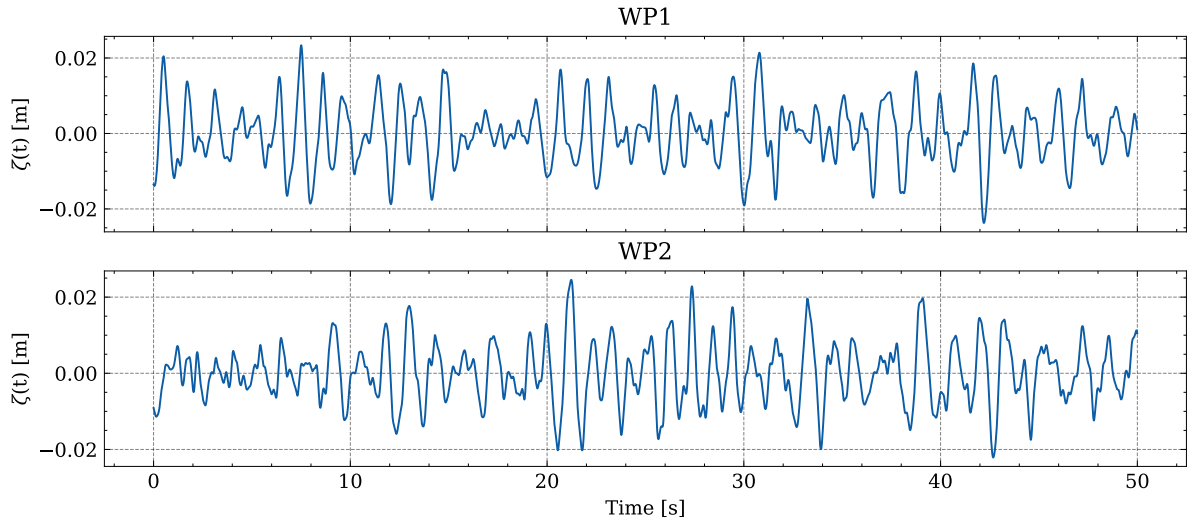


Figure 6.3: Wave realizations for WP1 and WP2.

Thruster configuration

CSAD is equipped with 6 azimuth thrusters. However, throughout the spring of 2023, the model vessel experienced a significantly reduced performance for certain thrusters. Both port side thrusters were consistently non-functional, while the forward thruster also experienced malfunctioning for parts of the spring, particularly during the final lab session. This strongly affected the vessel's maneuverability, especially in bow and beam sea.

It was also noticed that the thruster configuration at CSAD not provided any thrust for the $\sim 10\%$ smallest thrust signals. Achieving satisfactory performance, even for stationkeeping in calm water, would then become challenging. A constant thruster bias was thus introduced, ensuring that every thruster at all time was running on a slightly higher rpm than originally required. Thrust was thus provided, even for small rpms, while also making the vessel more reactive to sudden and significant changes in environmental loads.

6.2 Lab session 1 - Test of baseline controller

The first controller to be tested in the lab was the baseline PID controller. For the PID controller, two specific tests were conducted: a stationkeeping test and a 4-corner test. The stationkeeping test aimed to assess the controller's ability to maintain the vessel at a fixed position, while the 4-corner test involved maneuvering the vessel through a predetermined set of four corners.

6.2.1 Stationkeeping test scenario

The initial test conducted for the PID controller was the stationkeeping test. The resulting position, represented in the basin-frame, is illustrated in Figure 6.4.

From the figure, it is evident that the vessel's position oscillates around the desired setpoint at x = 0. The oscillations indicate that the vessel maintains a relatively close proximity to the setpoint. The observed steady state offset from the setpoint is minimal, with a value of only 0.001 m. Additionally, the maximum deviation of the vessel from the setpoint can be observed to be 0.06 m.

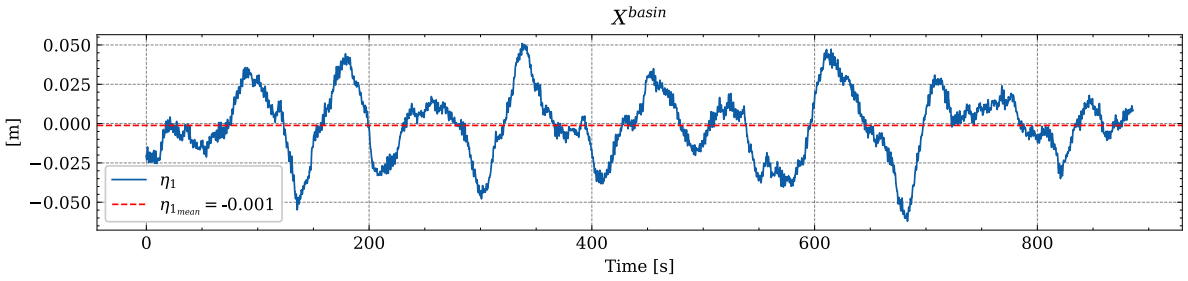


Figure 6.4: Position for a PID controller during a station keeping test in head sea.

The results demonstrate a minimal steady-state offset, suggesting that the integral action implemeted for the PID controller effectively compensates for residual loads. However, it is important to note the presence of oscillatory movement in the graph. It is likely that this oscillatory behavior origins from a slightly aggressive tuning of the derivative term in the controller. This should be properly addressed before any evaluation of the bias compensation performance. Limited time in the lab did, however, restrict the time used for stationkeeping tests.

6.2.2 4-corner test scenario

The final test for the baseline controller was the 4-corner test. The 4-corner test is useful for evaluating the performance of a DP system as it captures both isolated and coupled movements of the vessel. The test is performed in five distinct steps which assess the DP behavior when performing strongly coupled setpoint changes (Skjetne et al., 2017). The test can be described as:

1. Change of position consisting of pure surge motion.
2. Change of position consisting of pure sway motion.
3. Change of heading consisting of pure yaw motion.
4. Change of position consisting of coupled surge and sway motion.
5. Coupled maneuver involving motion in all three DOFs.

For each setpoint, the vessel was directed to maintain its position, in accordance with the predefined tolerance limits of ±0.5 m and ±0.3°, for a duration of 100 seconds before proceeding.

Figure 6.5 and Figure 6.6 present the resulting position and heading from the final 4-corner test. The test was initialized in the origin of the basin-frame. It can be observed that the PID controller managed to follow the reference trajectory in a satisfactory way. The best performance was seen for purely surge motion. For this maneuver, only small deviations from the reference trajectory were observed. Small offsets could be detected, primarily when the vessel started to approach the second setpoint, and could be caused by the vessel decreasing its velocity.

The controller displayed a slightly more oscillatory behavior for the isolated sway maneuver, than during the surge motion. However, this should be expected as the vessel is imposed to greater loads during transverse motion. Additionally, as both thrusters on the port side of the vessel were disfunctional, the maneuverability of the vessel was most likely heavily decreased. This was the case for all maneuvers.

The third setpoint change involved a heading change from 0 to 45°. As can be observed in the top right corner of Figure 6.5, this maneuver led to a positional offset. During a heading change, the vessel is imposed to transient behaviors, which can cause the controller to struggle. As the vessel was ordered to continuously stay within the tolerance for a given duration of time, a few attempts were required. Consequently, it stayed at the setpoint for a longer period of time, compared to other setpoints. This could potentially result in the performance being perceived as even worse in Figure 6.5, than the actual case.

The two lasts steps in the four corner test introduced coupled setpoint changes, and were also executed with a satisfactory result. Some oscillatory behavior can be observed, but overall, the results could be deemed satisfactory.

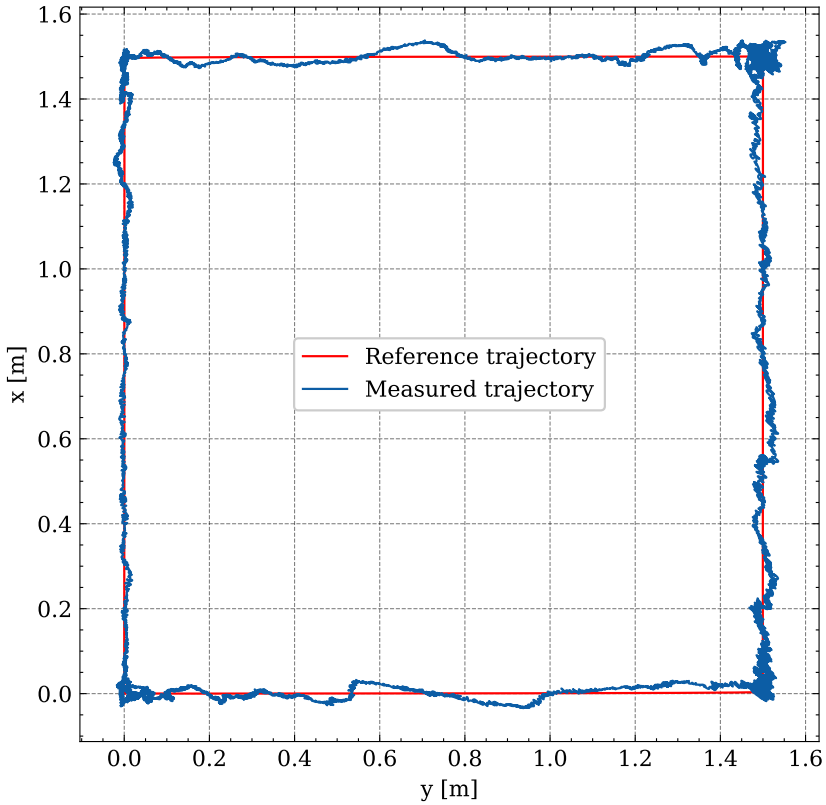


Figure 6.5: xy-plot.

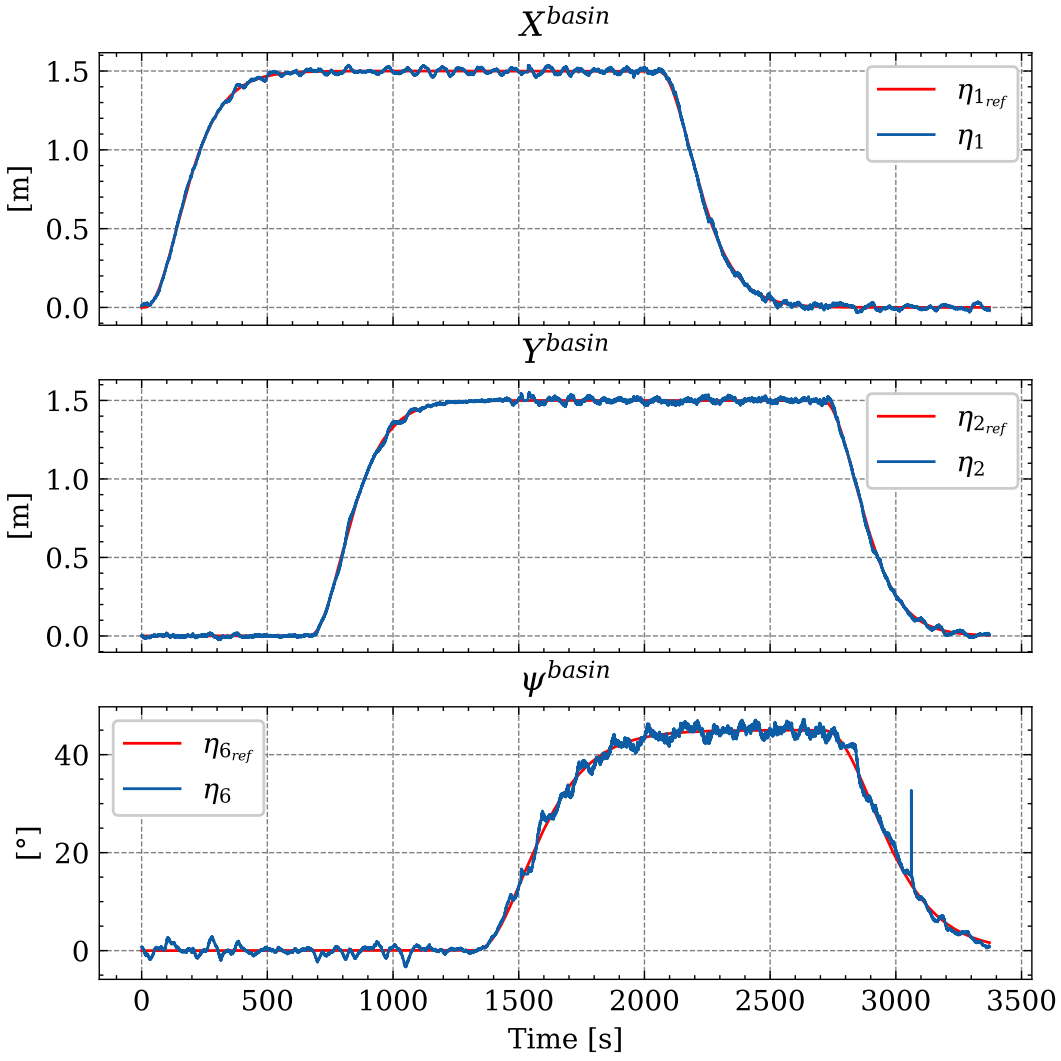


Figure 6.6: Position and heading plots.

6.3 Lab session 2 - Test of the adaptive controller

The adaptive controller was the second and final controller to be tested in the MC lab. Similarly as for the PID controller, a stationkeeping test was first performed, to evaluate the controller's ability to maintain a predefined setpoint. Secondly, a 30°-30° heading test were to be performed. However, due to problems with the thrusters on CSAD during lab session 2, resulting in a strongly reduced maneuverability, a 20°-20° heading test was performed instead. The original intention was to conclude the lab session with a 4-corner test, as for the first session, but due to the time-consuming tuning process and limited available lab time, this final test had to be disregarded.

6.3.1 Stationkeeping test scenario

The initial validation test of the adaptive controller involved the simple stationkeeping test. Figure 6.7 showcases the x-position and the bias rejection term, respectively, during the trial. It can be observed that the adaptive controller exhibited a constant offset of 0.009 m in x-position, surpassing that of the PID controller. However, the maximum deviation from the setpoint, measuring 0.57 m, was less than for the PID controller.

The steady state offset indicates that the bias rejection term was not functioning optimally. Upon further examination of the bias rejection term in surge, from the second plot in Figure 6.7, it became clear that the term had not yet fully reach a stationary condition. Still, as indicated by the red trend line, it was progressively increasing over time. It is worth noticing that the mean value of the x-position demonstrated a slight smaller offset to the setpoint during the second half of the simulation, compared to the first. This can be seen from Table 6.1 where the mean values for the two intervals are presented. This observation suggests that if the bias term would have been given enough time to reach a stationary condition, the positional offset would likely decrease further.

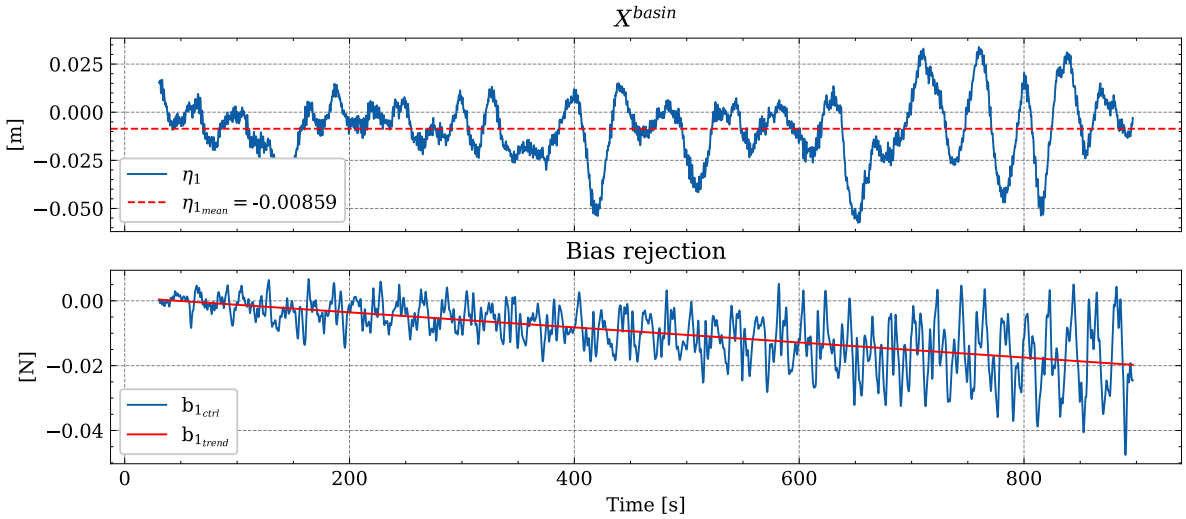


Figure 6.7: Position and bias rejection term for an adaptive controller during a stationkeeping test in head sea.

Table 6.1: The evolution of the average x position of the vessel during stationkeeping.

Time interval [s]	Mean value [mm]
$t = [0 - 450]$	-9.01
$t = [450 - 900]$	-8.17

The challenge related to the slowly growing bias rejection term could be addressed by two potential approaches. Firstly, by allowing the controller more time to reach a stationary conditions. However, as indicated by the small vertical axis in the bias rejection term in Figure 6.7, this would require an extensively large amount of time.

Secondly, it could be attempted to increase the value of the adaptive gain term, $\mathbf{\Gamma}$, to expedite the build-up of the rejection term. The impact of adjusting this gain was discussed in Section 5.3.1, proving as a trade-off between rapid estimation and stability. This was tested in the laboratory, and the results of the test is seen in Figure 6.8. Notably, a more aggressive tuning caused the steady state in x-position to be almost mitigated, as indicated with the dashed mean value. Furthermore, the bias rejection term from the controller aligned well with the estimated bias from the observer. In the case of Figure 6.7, the observer bias was not displayed due to the significant difference in magnitude compared to the bias rejection term, noticeable on the large vertical axis in Figure 6.8. Even though the increased tuning showed promising results, it is crucial to exercise caution when increasing the values of $\mathbf{\Gamma}$, as it can be observed that, after around 200 seconds, the bias rejection term starts exhibiting unstable behavior. This instability indication was consistently observed in all cases when the adaptive gain was tuned to aggressively.

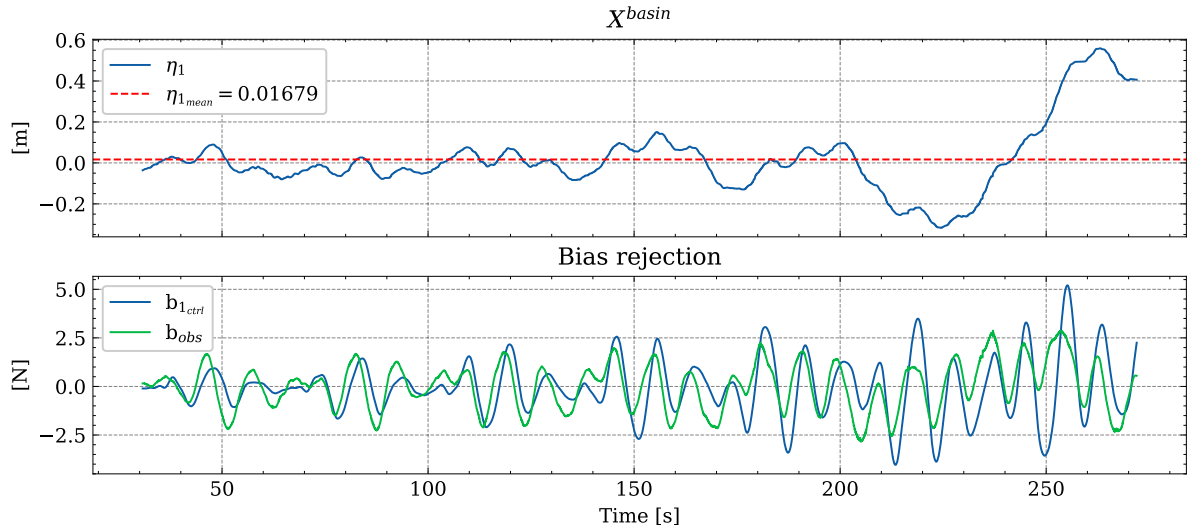


Figure 6.8: Experimental stationkeeping test with an increased adaptive gain. The mean value in the upper plot is calculated in the interval 0-200 seconds.

Section 5.3.1 stated that a potential explanation for these unstable oscillations could be due to the presence of noise in the velocity estimates. This issue would also be more pronounced during experimental testing, as a narrower time window restricted the extent to which the observer could be tuned to perfection. Consequently, the adaptive gain had to be tuned to smaller values, evidently leading to a slower bias compensation. This trade-off was necessary in order to preserve the stability of the vessel, and mitigate noisy signals from entering the controller.

6.3.2 20°-20° heading change test scenario

Secondly, the 20°-20° heading change test was conducted. Figure 6.9 presents the vessel's heading throughout the trial. It can be observed that the heading is maintained within an acceptable range of the setpoint. The vessel encountered greater challenges as the heading angle increased, displaying one of the reasons why the test was constrained to a maximum 20° angle.

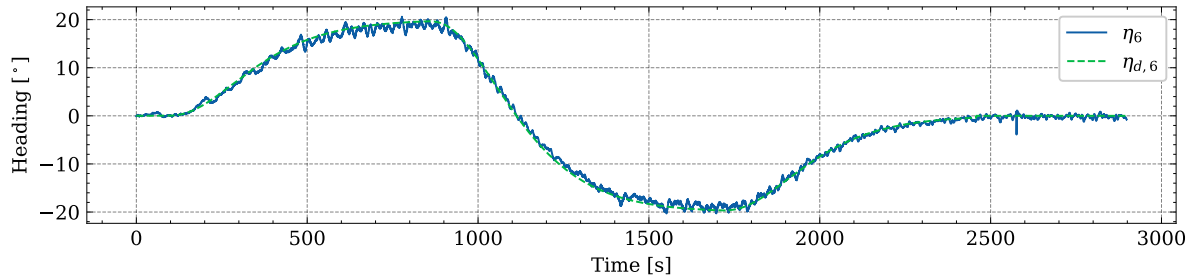


Figure 6.9: Heading for CSAD during a 20°-20° heading test scenario.

Figure 6.10 displays the corresponding bias rejection term for the duration of the test. Red trend lines are drawn to better illustrate the tendency of the estimates. As for the stationkeeping test, the slow development, and presence of noise, are noticeable. That being said, the trend lines disclose the development of the compensation, which can be said to align with the expected results, both from theory and former simulations.

The bias rejection term in surge maintained a negative mean value, as the vessel only operated in head and partially bow sea. All residual loads are thus working in the negative x-axis. Fluctuations in surge would be expected as the vessel turns from 20° to -20°, but these are not distinguishable in the figure,

due to the slow update of the rejection estimate. As for in sway and yaw, the bias rejection term swung around a zero-mean value, in accordance with the heading of the vessel. Stationary conditions were not obtained, neither at 20° nor -20°, as the vessel would require a considerably longer duration for the controller to achieve such conditions.

The corresponding bias rejection term from the same test in the simulation platform is given in Figure 6.11. This figure illustrates a clear synergy between the experimental and theoretical bias rejection term, especially if the simulated bias rejection term is evaluated against the trend lines in the experimental data. The magnitude of the bias compensation is, nonetheless, different. It is reasonable to assume that these magnitudes would have been more comparable if the experimental bias rejection term had been given sufficient time to fully develop and stabilize.

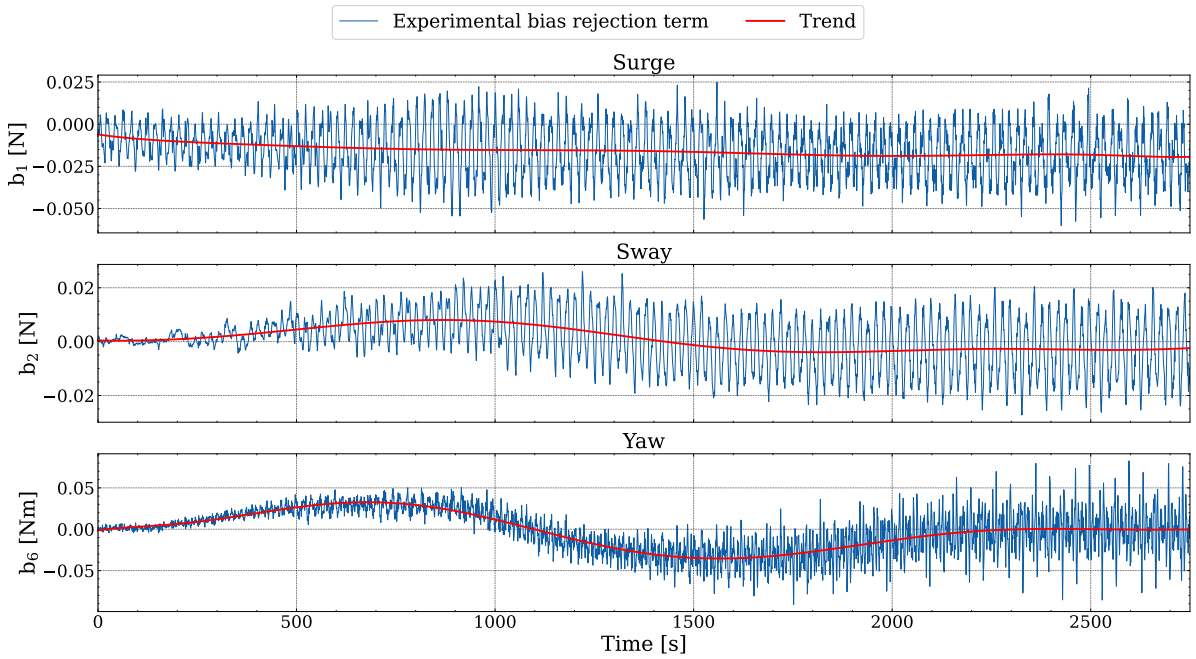


Figure 6.10: The experimental bias rejection term for CSAD during a 20°-20° heading test scenario.

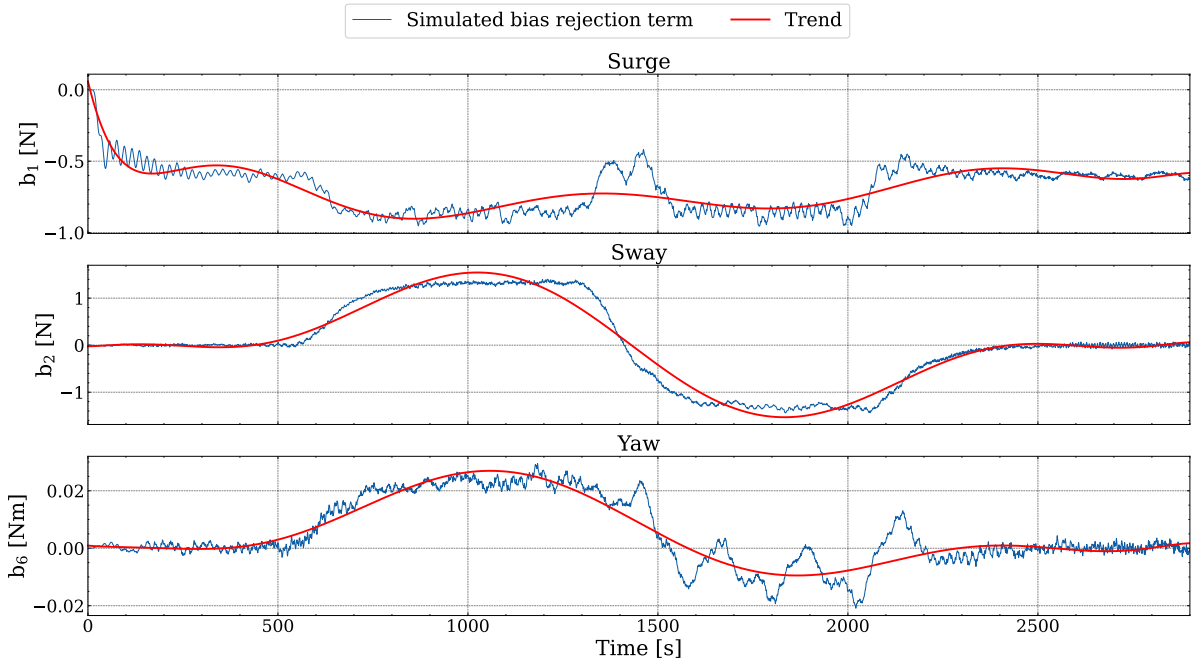


Figure 6.11: The corresponding simulated bias rejection term for CSAD during a 20°-20° heading test scenario

6.4 Sources of error

When conducting experimental tests, potential sources of error are always present, which one will need to take into consideration when evaluating the results. During the test performed with CSAD in the MC-lab, a significant numbers of such error sources were present.

Motion capture system

The control system relies on measurements of the vessel's position obtained from the motion capture system, Qualisys. As presented in Section 2.8, the system calculate these measurements from four distinctly placed cameras. The observable zone is defined as the area in the basin where the cameras have a clear view of, and are capable of tracking, the vessel. Outside this zone, Qualisys is unable to determine the pose of the vessel. The size of this observable zone depends on the setup and location of the cameras. During the lab experiments it was discovered that the observable zone was somewhat restricted. This resulted in a limited area of approximately 2 m $\times$ 2 m where tests could be conducted. In addition to loosing track of the vessel outside this area, the Qualisys system also struggled to maintain its track of the vessel for certain orientations within the observable zone.

When Qualisys lost track of the vessel, an implemented dead reckoning functionality was activated to reduce the negative impact of losing the vessel's position. However, the estimated position quickly turned inaccurate when the vessel remained outside the observable zone, indicating that the internal observer model likely exhibited considerable deficiencies or imperfections.

The quality of the Qualisys observation also varied greatly within the observable zone. This observability was defined by a residual weight, provided by Qualisys. In an ideal scenario, the measurement noise covariance matrix in the Kalman filter, responsible for weighing the uncertainties in the measurements, could be adjusted specifically for each location in the basin, dependent on this residual weight. However, during the experimental tests, such variable tuning was not implemented due to the inconsistent and unpredictable nature of the residual. Due to its complexity, it is unlikely that such observer tuning would have yielded any better results for this master's thesis.

Vessel model

The vessel model, CSAD, did, as previously stated, experience significant errors due to several malfunctioning thrusters. Specifically, two out of the six thrusters were consistently non-functional, and for some test runs, only three out of the six thrusters were activated. The affected thrusters were located on the port side, and in the bow of the vessel. The original idea, prior to the lab experiments, was to run the fixed-angle pseudoinverse thrust allocation from the simulation platform. However, as this yielded poor maneuverability attributes, the original pseudoinverse allocation method had to be used. The fixed-angle allocation was initially intended as it removed any rate constraints in the thruster angles.

In the lab, the thruster angle offset required calibration on a daily basis. These offsets were approximate values, and had to be adjusted by visually observing the angle of the thrusters. This calibration method introduced a high possibility of human error, and required that the offsets were carefully observed prior to each test.

As previously mentioned, during the discussion of the experimental setup, the thrusters installed on CSAD had limitation where they did not generate thrust for the smallest thrust commands. To overcome this limitation, thruster bias was implemented. The lower actuator signal limit, for which the thrusters failed to provide any thrust at, also varied in magnitude. This could potentially suppress the implemented thruster bias, and was found to vary with the amount of voltage supplied by the batteries to the servos. Careful considerations were thus taken, to ensure that CSAD always operated with batteries in a fully charged condition.

Wave basin

Conducting experiments in a wave tank allows for the investigation of a vessel's response under different wave conditions within a controlled environment. However, it is important to note that the sea state generated in a wave tank only is an approximation of real life conditions, and is influenced by tank wall effects. In particular, wave reflection can be a significant in the MC-lab's wave tank, due to its relatively small size. When waves encounter the tank walls, they can bounce back and interfere with the incoming waves, leading to an altered wave pattern.

Additionally, wave diffraction can occur when waves encounter obstacles that causes the waves to change direction. This phenomena was also believed to occur for the lab testing in this project.

Both wave reflection and wave diffraction impacted the accuracy and realism of the wave conditions experienced by the model vessel and should be considered when interpreting the results of the conducted experiments.

Wave maker

The wave generator in the MC-lab has shown to generate wave conditions which does not always match the input to the generator. This was also the case for the lab experimenters presented above, as indicated in Figure 6.2. When generating regular waves, it was also noticed smaller, high-frequent ripples on top of the regular wave. This was assumed to be caused by loose mechanical components, vibrating as the wave flap is moved back and forth by the piston. The piston was also connected to the wave flap at its center, and as the wave flap not necessarily can be considered as completely rigid, this may also lead to errors in the generated waves.

Combined with the previously discussed tank wall effects, it can be presumed that the waves acting on the vessel, not coincide entirely with the predetermined wave parameters. However, as the exact conditions of the sea state not were the primarily focus of the lab sessions, time was not allocated to extensively calibrate the wave generator prior to every test.

6.5 Key takeaways from the lab sessions

The experimental results obtained from these tests served as the basis for validating the results of the simulation study. By comparing the simulation outputs with the experimental data, the researchers could further assess the accuracy and reliability of the simulation platform.

The laboratory tests presented in this chapter have been highly susceptible to various sources of error. Additionally, re-tuning of gains compared to the settings used in the simulation tests was required. This adjustment was necessary to account for the differences between the simulation environment and the physical setup. However, this naturally influenced the basis of comparison between the different sets of tests.

Despite challenges related both to error sources and tuning, the same tendencies and behaviors observed in the simulation studies could be seen replicated for the tests executed on the model vessel. This serves as a strong indicator of validity of the simulation platform. These findings further enhance the assumption that the simulation platform accurately represents the real dynamics of the system. Combined with the simulation results, this contributes to a comprehensive validation process, affirming the credibility of the simulation platform.

Chapter 7

Conclusion

This final chapter will include a concise conclusion, as well as listing some recommendations for further research, based on findings in this thesis.

7.1 Conclusion

This master's thesis have successfully implemented a high-fidelity simulation platform in Python. The simulation platform has been developed in close collaboration with Hygen (2023), and builds upon previous work done by Brørby (2022). The simulation platform provides comprehensive descriptions of vessel kinetics and kinematics, external environmental loads, and DP functionality. This thesis has primarily focused on the developed simulation model of the model-scale vessel CSAD, with particular emphasis on computational efficiency. Through significant preprocessing, as by employing a matrix formulation for second order wave load calculations, the runtime was reduced by a magnitude of 10, compared to the traditional double for-loop implementation. This allowed for a higher number of wave components in real-time simulation, thereby improving the fidelity of the platform. The simulated response of CSAD has been verified against theory and validated by experimental results, yielding highly encouraging results.

The developed simulation model has been made compatible with ROS. This enabled communication between the software of the platform and hardware in the MC-lab, facilitating for experimental testing. Leveraging the widespread usage of ROS, the source code of MCSimPython could then interact effectively with various robotic systems, enabling comprehensive experimental investigations. This integration not only enabled the validation of the simulation study conducted in this thesis, but also enhanced the platform's versatility, opening up additional use cases for external users. This feature could prove to be particularly beneficial for future master's students whom may have limited knowledge of software-hardware communication and wish to conduct laboratory tests. The ROS compatibility made it simple to interface the MCSimPython platform with hardware components, simplifying the setup and execution of experiments in the laboratory environment.

With the help of the implemented simulation platform, an extensive simulation study has been conducted. The study addressed the compensation of second order, low-frequency wave loads in DP vessels, and explored alternative control law algorithms that more efficiently would compensate for such loads. Consequently, four controllers have been implemented and tested. In addition to the conventional integral action and bias estimation in the DP observer, two additional controllers were proposed. These included an adaptive control technique using a truncated Fourier series bias model, and a direct compensation neural network controller.

The adaptive controller displayed sufficient performance for the different tests conducted in the simulation study. However, the specified KPIs indicated that the controller had a slightly worse overall performance than the other three methods presented. Despite this, the Fourier series based bias model

demonstrated a more rapid bias rejection than both baseline methods, in transient behavior. It was discovered that the robustness of the controller's bias compensation depended on the quality of the velocity signals, which in this thesis was believed to be moderately too noisy. A perfectly tuned observer would likely improve the performance of the bias compensation even further. The controller was also tested experimentally, but as the velocity estimates in lab had a more significant amount of noise, the adaptive gain had to be tuned appropriately slow to avoid instability.

Furthermore, a neural network bias model was implemented. This method was based on the nominal PD control law, and compensated for residual loads through an internally trained neural network. The controller demonstrated a satisfactory DP performance, marginally showcasing the highest positional accuracy among the tested controllers. Additionally, the method exhibited a superiority over the other approaches in regard to bias estimation. However, as the neural network was implemented as a direct bias compensation mechanism, based on a perfect vessel model, it was deemed to be somewhat unrealistic. This motivated the conducted sensitivity analysis, highlighting the need for further studies and testing to make the controller more commercially feasible. Studies involving more indirect implementations of neural networks in DP control systems have thus also been explored and referred to in the thesis. While the developed network may not be currently suitable for direct implementation in a real system, the authors acknowledge the promising potential of utilizing neural networks for bias estimation in marine controllers.

Finally, it is crucial to acknowledge the limitations inherent in the test scenarios. As all simulations presented in the simulation study are tested in the same sea state, and limited to two distinct maneuvers, the results might present an artificial degree of adaptation tailored specifically to these scenarios. Hence, the possibility that the controllers may not exhibit the same level of robustness in other test should be considered.

7.2 Further work

This master's thesis is intended to provide a foundation for future research and exploration. The following suggestions are given to guide future researchers and students in further enhancing the impact of the contributions delivered by this thesis.

- **Conduct more extensive testing of the proposed disturbance rejection methods**

  In order to properly analyze and validate the implemented disturbance rejection methods in this master's thesis, it is crucial to conduct tests across a broader range of sea states. Currently, the simulation study is limited to evaluating the methods in only one specific sea state. To assess the robustness of the controllers, it is imperative to perform tests in a variety of environmental conditions, especially including more extreme sea states. To enhance the credibility of the presented sensitivity analyses, additional testing is also recommended. Particularly when considering the sensitivity analysis of the neural network controller, more extensive testing that accounts for model and parameter inaccuracies would be beneficial. By expanding the scope of testing, the findings from this thesis would prove more credible.

- **Further explore the use of neural networks in DP control design**

  An approach for using neural networks in DP control design has been investigated and implemented in this thesis. As stated in the conclusion, this specific implementation face some challenges which would need to be addressed before implementing it to a real system. However, the authors strongly believe that neural networks and machine learning has great potential within the field of marine control systems and highly urge others to look into the use of these methods for bias estimation, as well as for other challenges, such as system modeling and identification.

- **Implement the MCSimPython toolbox as verified NTNU software**

  One of the main contributions of this thesis is the development of the open source and high-fidelity simulation platform, the MCSimPython Python package. From the very start, the objective was to develop a user-friendly and valuable resource for researchers, particularly for future master's

students. It is desirable that the platform can serve a solid foundation for future innovating research, and to obtain this, it is vital that the platform is continuously validated and approved by external experts. A unified base platform would be beneficial for future students as it would eliminate the need for individual development of simulation platforms, thereby potentially expediting their research initiatives.

Bibliography

Aflak, O. (2018). *Neural network from scratch in python.* https://towardsdatascience.com/math-neural-network-from-scratch-in-python-d6da9f29ce65

Agostinho, A. C., Moratelli, L., Tannuri, E. A., & Morishita, H. M. (2009). Sliding mode control applied to offshore dynamic positioning systems [8th IFAC Conference on Manoeuvring and Control of Marine Craft]. *IFAC Proceedings Volumes*, *42*(18), 237–242. https://doi.org/10.3182/20090916-3-BR-3001.0009

Al-Aubidy, K. M. (2023). Lecture (5) neural networks in control systems. https://www.philadelphia.edu.jo/academics/kaubaidy/uploads/IntCon-lec5.pdf

Arcak, M., & Kokotović, P. (2001). Redesign of backstepping for robustness against unmodelled dynamics. *International Journal of Robust and Nonlinear Control*, *11*, 633–643. https://doi.org/10.1002/rnc.620

Audet, C., & Hare, W. (2017). *Derivative-free and blackbox optimization.* https://doi.org/10.1007/978-3-319-68913-5

Baheti, P. (2021). *Activation functions in neural networks [12 types & use cases]* [Extracted 06.2023 from https://www.v7labs.com/blog/neural-networks-activation-functions].

Balchen, J. G., Jenssen, N. A., & Sælid, S. (1976). Dynamic positioning using kalman filtering and optimal control theory [In IFAC/IFIP symposium on automation in offshore oil field operation].

Balchen, J., Andresen, T., & Foss, B. (2016). *Reguleringsteknikk* (Vol. 6). Institutt for teknisk kybernetikk - Norges teknisk-naturvitenskapelige universitet (NTNU).

Balchen, J., Jenssen, N., & Saelid, S. (1976). Dynamic positioning using kalman filtering and optimal control theory. *IFAC/IFIP Symposium on Automation in Offshore Oil Field Operation.*

Bjørnø, J., Heyn, H.-M., Skjetne, R., Dahl, A. R., & Frederich, P. (2017). Modeling, parameter identification and thruster-assisted position mooring of c/s inocean cat i drillship. *ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering.* https://doi.org/10.1115/OMAE2017-61896

Brigham, E. O. (1988). *The fast fourier transform and its applications.* Prentice Hall.

Brørby, B. T. T. (2022). *Wave load compensation in dp control systems* (Master's thesis). Norwegian University of Science and Technology.

Chen, M. (2021). *Lesson 8-2 nelder mead method / algorithm* [UCLA Winter 2021 - Stats 102A - Introduction to Computational Statistics with R]. https://www.youtube.com/watch?v=vOYlVvT3W80

Du, J., Yang, Y., Wang, D., & Guo, C. (2013). *A robust adaptive neural networks controller for maritime dynamic positioning system* (tech. rep.). Elsevier.

Faltinsen, O. M. (1990). *Sea loads on ships and offshore structures.* Cambridge University Press.

Fonseca, N., & Stansberg, C. T. (2017). Wave drift forces and low frequency damping on the exwave fpso.

Fossen, T. I. (2021). *Handbok of marine craft hydrodynamics and motion control*. John Wiley & Sons.

Fossen, T. I. (2022). *Lecture notes ttk 4190 guidance, navigation and control of vehicle: Chapter 3 – rigid-body kinetics*. https://www.fossen.biz/wiley/pdf/Ch3.pdf

Fossen, T. I., & Strand, J. P. (1999). Passive nonlinear observer design for ships using lyapunov methods: Full-scale experiments with a supply vessel. *Automatica*, *35*(1), 3–16. https://doi.org/10.1016/S0005-1098(98)00121-6

Fung, P., & Grimble, M. (1983). Dynamic ship positioning using a self-tuning kalman filter. *IEEE Transactions on Automatic Control*, *28*(3), 339–350. https://doi.org/10.1109/TAC.1983.1103226

Gillis, A. S. (2021). *Object-oriented programming*. https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20(OOP)%20is%20a%20computer%20programming%20model,has%20unique%20attributes%20and%20behavior.

Grimble, M., Patton, R., & Wise, D. (1980). Use of kalman filtering techniques in dynamic ship-positioning systems [In IEE proceedings D-control theory and applications, Vol. 127 (p. 93). IET.].

Hygen, J.-E. (2023). *Deterministic response prediction of wave-induced vessel motions*.

Hygen, J.-E., Konsghaug, M., & Mo, H. (2023). MCSimPython. https://doi.org/10.5281/zenodo.8018718

IBM. (2023). *What is a neural network?* https://www.ibm.com/topics/neural-networks

IMO. (1994). *Guidelines for vessels with dynamic positioning systems*. International Maritime Organization.

International Trade Administration. (2022). *Norway - country commercial guide*. https://www.trade.gov/country-commercial-guides/norway-shipping-maritime-equipment-services

Kaasen, K. E. (2015). Efficient thrust allocation for dp systems using a biased least squares criterion. *ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering*. https://doi.org/10.1115/omae2015-41948

Kanazawa, M., Skulstad, R., Wang, T., Li, G., Hatledal, L. I., & Zhang, H. (2022). A physics-data co-operative ship dynamic model for a docking operation. *IEEE Sensors Journal*, *22*(11), 11173–11183. https://doi.org/10.1109/JSEN.2022.3171036

Kongshaug, M., & Mo, H. (2022). *Pre-study on wave kinematics and real-time compensation of wave loads by a dp control system* [Pre-study as a part of TMR4510 - Marine control systems specialization project.], NTNU.

Kreyszig, E. (2011). *Advanced engineering mathematics* (tenth). John Wiley & Sons, INC.

Larsen, C. M., et al. (2021). *Marine dynamics, tmr 4182 marine dynamics*. Department of Marine Technology, Faculty of Engineering, NTNU.

Li, J., Xiang, X., & Yang, S. (2021). *Robust adaptive neural network control for dynamic positioning of marine vessels with prescribed performance under model uncertainties and input saturation* (tech. rep.). Elsevier.

Liu, J. (2013). *Radial basis funtion (rbf) neural network control for mechanical systems*. Tsinghua University Press,

Loria, A., Fossen, T., & Panteley, E. (2000). A separation principle for dynamic positioning of ships: Theoretical and experimental results. *IEEE Transactions on Control Systems Technology*, *8*. https://doi.org/10.1109/87.826804

Loría, A., & Panteley, E. (1999). A separation principle for a class of euler-lagrange systems. In H. Nijmeijer & T. Fossen (Eds.), *New directions in nonlinear observer design* (pp. 229–247). Springer London.

Løvås, H. S. (2019). *Dp autotuning by use of derivative-free optimization* (Master's thesis). Norwegian University of Science and Technology. Faculty of Engineering Department of Marine Technology.

Newman, J. N. (2017). *Marine hydrodynamics*. The MIT Press.

Newman, J. N. (1974). *Second-order, slowly-varying forces on vessels in irregular waves* (Report Paper 19). Massachusetts Institute of Technology, MIT, USA, Department of Ocean Engineering.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (second). Springer.

NTNU. (2023). *Marine cybernetics teaching laboratory*. https://www.ntnu.edu/imt/lab/cybernetics

Numpy. (2022). *What is numpy?* [Extracted 05.2023 from https://numpy.org/doc/stable/user/whatisnumpy.html].

Open Robotics. (2022). *Ros - robot operating system: What is ros?* https://www.ros.org/

OrcaFlex. (2022). *Vessel theory: Wave drift and sum frequency loads*. https://www.orcina.com/webhelp/OrcaFlex/Content/html/Vesseltypes,WavedriftandsumfrequencyQTFs.htm

Park, C., Chung, S., & Lee, H. (2020). Vehicle-in-the-loop in global coordinates for advanced driver assistance system. *Applied Sciences*, *10*, 2645. https://doi.org/10.3390/app10082645

Pastor-Pellicer, J., Zamora-Martínez, F., España-Boquera, S., & Castro-Bleda, M. J. (2013). F-measure as the error function to train neural networks. In I. Rojas, G. Joya & J. Gabestany (Eds.), *Advances in computational intelligence* (pp. 376–384). Springer Berlin Heidelberg.

Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, *51*(3), 406–413. https://doi.org/10.1017/S0305004100030401

Saelid, S., Jensen, N. A., & Balchen, J. G. (1983). Design and analysis of a dynamic positioning system based on kalman filtering and optimal control. *Automatic Control, IEEE Transactions on*, *28*, 331–339. https://doi.org/10.1109/TAC.1983.1103225

Sharma, S. (2017). *Activation functions in neural networks* [Extracted 06.2023 from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6].

Skjetne, R. (2022). Marine control systems ii lecture 10: Disturbance rejection and integral action [Lecture Notes in TMR4243 - Marine control systems II (2022)].

Skjetne, R., & Egeland, O. (2006). Hardware-in-the-loop testing of marine control systems. *Modeling, Identification and Control*, *27*(4), 239–258.

Skjetne, R., & Kjerstad, Ø. K. (2016). Disturbance rejection by acceleration feedforward for marine surface vessels. *4*.

Skjetne, R., Sørensen, M. E. N., Breivik, M., Værnø, S., Brodtkorb, A. H., Sørensen, A. J., Kjerstad, Ø. K., Calabrò, V., & Vinje, B. O. (2017). *AMOS DP Research Cruise 2016: Academic Full-Scale Testing of Experimental Dynamic Positioning Control Algorithms Onboard R/V Gunnerus* (Vol. Volume 1: Offshore Technology). https://doi.org/10.1115/OMAE2017-62045

Smogeli, Ø. (2022). Module 4: Simulation-based testing and verification [Lecture notes presented in the course TMR06, fall 2022].

Solomon, O. M., Jr. (1991). Psd computations using welch's method. [power spectral density (psd)]. https://doi.org/10.2172/5688766

Sørensen, A. J. (2011). A survey of dynamic positioning control systems. *Annual Reviews in Control*, *35*(1), 123–136. https://doi.org/10.1016/j.arcontrol.2011.03.008

Sørensen, A. J. (2018). *Marine cybernetics: Towards autonomous marine operations and systems, lecture notes*. Department of Marine Technology, NTNU.

Standing, R., Brendling, W., & Wilson, D. (1987). *Recent Developments in the Analysis of Wave Drift Forces, Low-Frequency Damping and Response* (Vol. All Days) [OTC-5456-MS]. https://doi.org/10.4043/5456-MS

The Robotics Back-End. (2022). *Using ros on raspberry pi: Best practices.* https://roboticsbackend.com/using-ros-on-raspberry-pi-best-practices/

The Robotics Back-End. (2023). *Create a ros driver package – introduction & what is a ros wrapper [1/4].* https://roboticsbackend.com/create-a-ros-driver-package-introduction-what-is-a-ros-wrapper-1-4/

Tian, Y., Su, D., Lauria, S., & Liu, X. (2022). Recent advances on loss functions in deep learning for computer vision. *Neurocomputing*, *497*, 129–158. https://doi.org/10.1016/j.neucom.2022.04.127

Torsethaugen, K. (2004). *Simplified double peak spectral model for ocean waves.* https://www.sintef.no/globalassets/upload/fiskeri_og_havbruk/konferanser/2004-jsc-193.pdf

Værnø, S. (2020). *Transient performance in dynamic positioning of ships: Investigation of residual load models and control methods for effective compensation* (Doctoral dissertation). NTNU.

Værnø, S., Brodtkorb, A., & Skjetne, R. (2019). Compensation of bias loads in dynamic positioning of marine surface vessels. *Ocean Engineering*, *178*, 484–492. https://doi.org/10.1016/j.oceaneng.2019.03.010

Værnø, S., Skjetne, R., Kjerstad, Ø. K., & Calabrò, V. (2019). Comparison of control design models and observers for dynamic positioning of surface vessels. *Control Engineering Practice*, *85*, 235–245. https://doi.org/10.1016/j.conengprac.2019.01.015

Wang, T., Skulstad, R., Kanazawa, M., Li, G., Æsøy, V., & Zhang, H. (2022). Physics-informed data-driven approach for ship docking prediction. *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 111–117. https://doi.org/10.1109/RCAR54675.2022.9872179

Appendix A

Guides

1 Guide: How to use MCSimPython

The complete documentation can be found at: https://wave-model.readthedocs.io/en/latest/index.html.

PyPI

The `MCSimPython` package can be installed from `PyPi` using `pip`:

```
pip install MCSimPython
```

Github

Install from GitHub in the following:

1. Clone the GitHub repository to your local computer.
2. Create a virtual environment: `py -m venv name-of-venv`
3. Activate virtual environment: `name-of-venv \scripts\activate`
4. Update pip and setuptools: `py -m pip install --upgrade pip setuptools`
5. Install the python package locally: `pip install .`
   Alternatively you can install as an editable: `pip install -e .`
6. Verify that the python package `MCSimPython` has been installed properly by running a demo script, or simply in the command promt:

   ```
   (venv) C:\path\to\dir> python
   >>> import MCSimPython
   >>>
   ```

Appendix B

X-in-the-loop

1 Model-in-the-loop

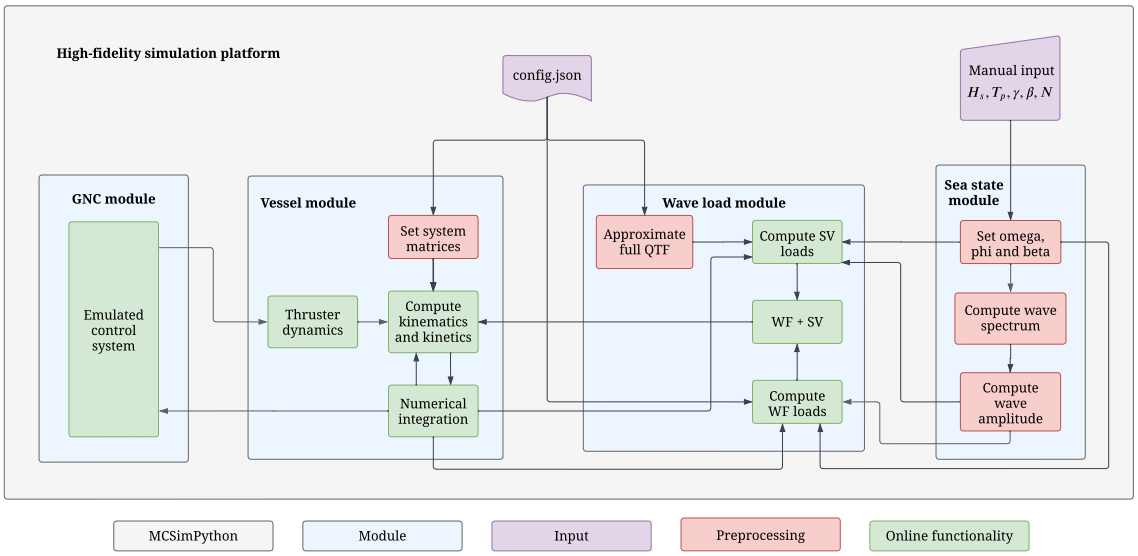


Figure B.1: Structure of MIL testing.

2 Simulator-in-the-loop

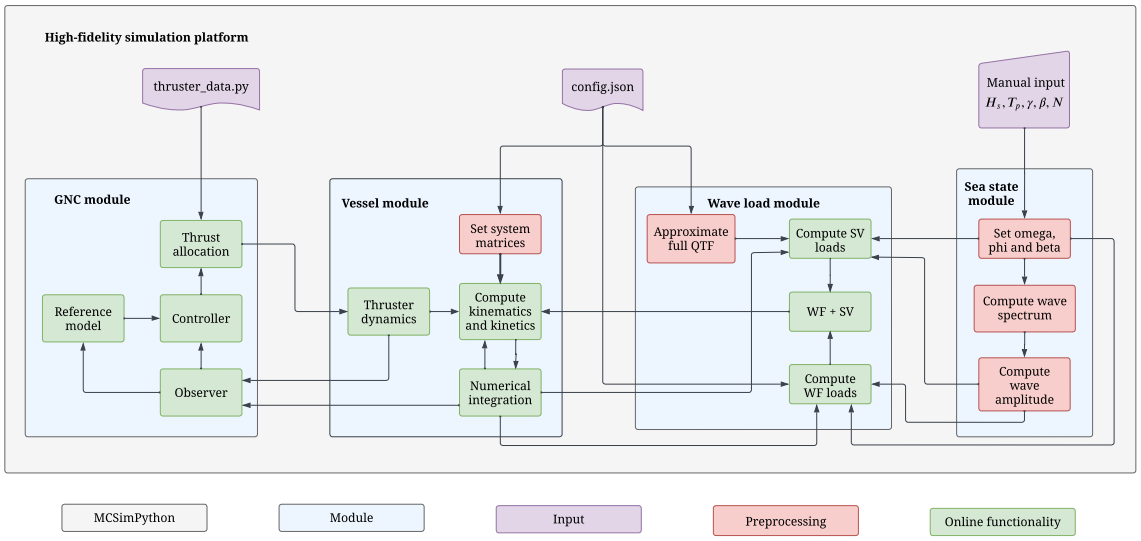


Figure B.2: Structure of SIL testing.

3 Hardware-in-the-loop

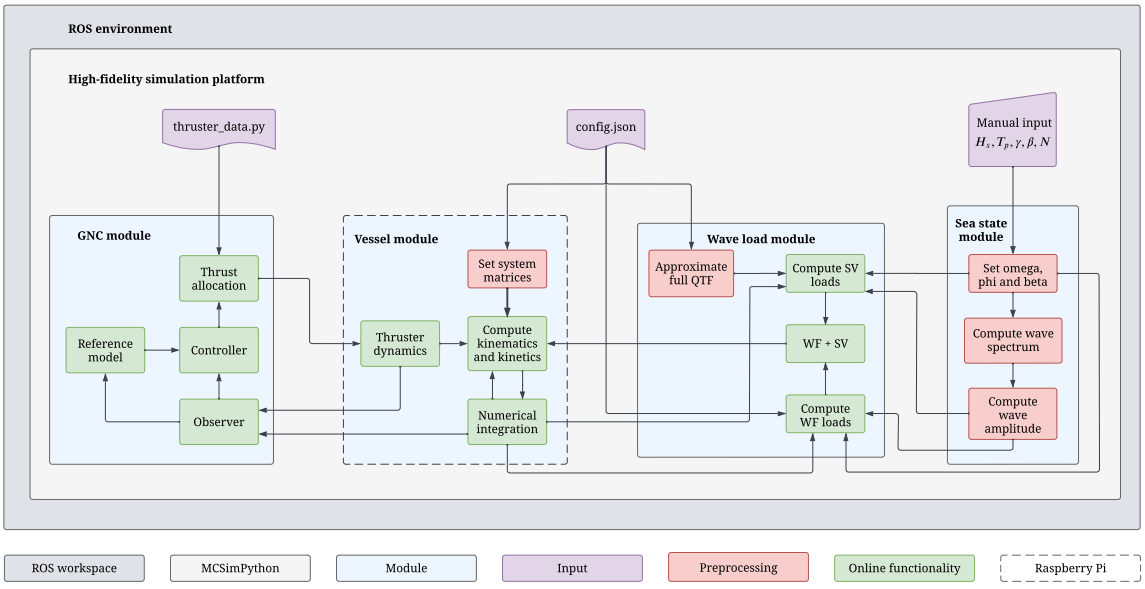


Figure B.3: Structure of HIL testing.

4 Physical system

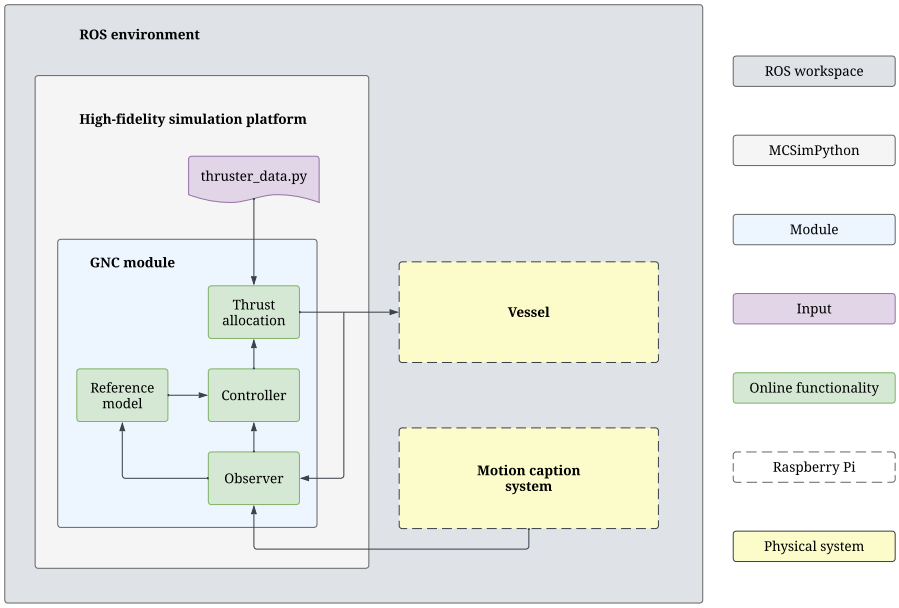
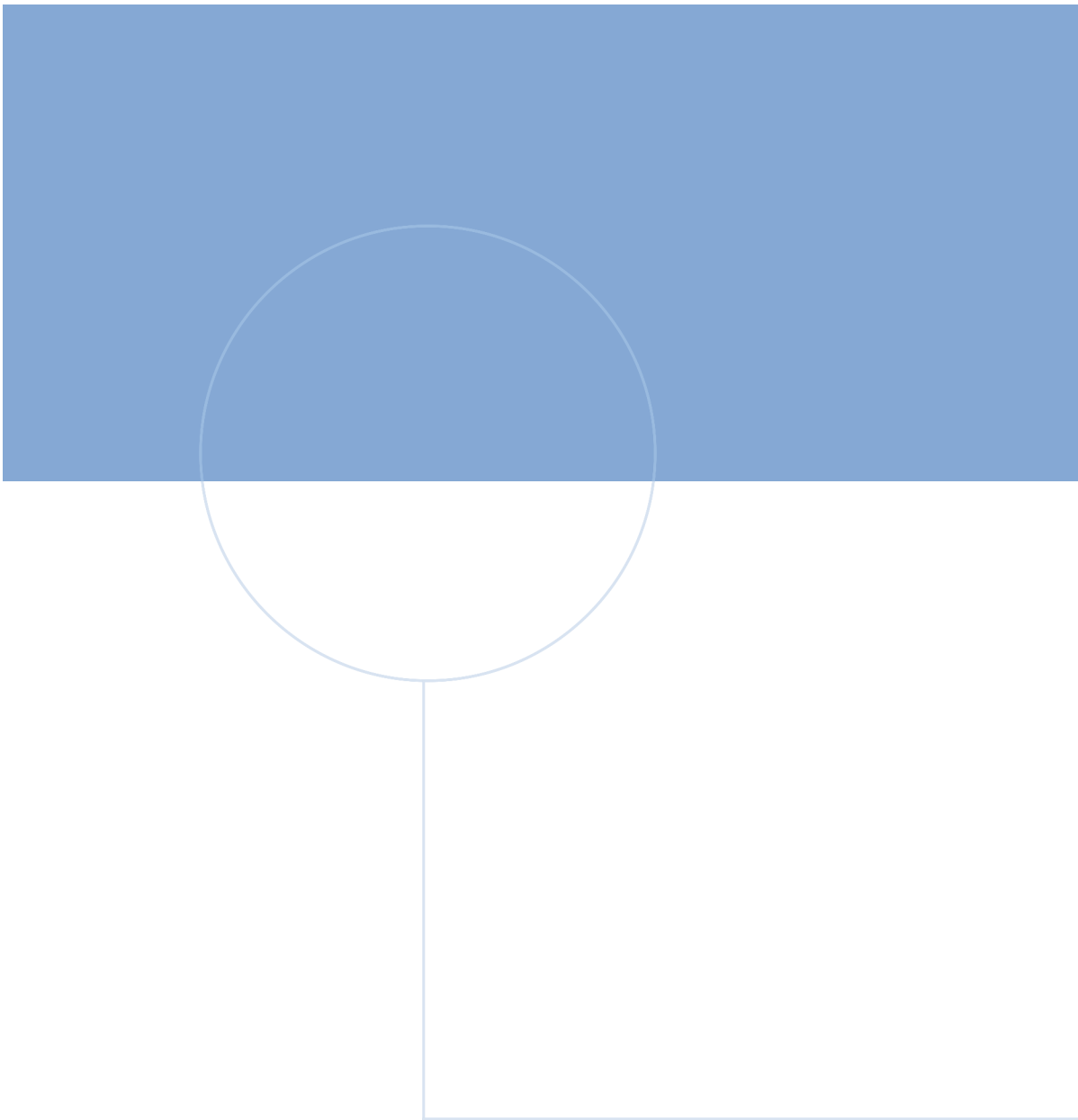


Figure B.4: Structure of physical testing system.



NTNU

Norwegian University of
Science and Technology