Caspar Michal Brodtkorb Eide

Software-in-the-loop testing of a wave filter

With application to cyber-physical testing of a floating
wind turbine

Master's thesis in Marine Technology
Supervisor: Thomas Michel Sauder
June 2023

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Caspar Michal Brodtkorb Eide

Software-in-the-loop testing of a wave filter

With application to cyber-physical testing of a floating wind turbine

Master's thesis in Marine Technology
Supervisor: Thomas Michel Sauder
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:	Caspar Michal Brodkorb Eide
Field of study:	Marine Cybernetics
Thesis title (English):	Software in the loop testing of a wave filter, with application to cyber-physical testing of a floating wind turbine

Background

The project is part of CYBERLAB, a research project involving Equinor, Aker Offshore Wind, Sevan, NOV/APL, Delmar Systems, NTNU, SINTEF (project manager), and Aarhus University.

The goal of CYBERLAB is to study offshore wind parks composed of interconnected floating wind turbines. In this context, model testing of a floating wind park will be performed where one turbine is modelled physically in the hydrodynamic laboratory, while the other ones are simulated numerically. The low-frequency motions and resulting forces will be exchanged in real-time between the numerical and the physical substructures (cyber-physical testing).

The objective of the Master study is to apply Fossen and Strand's (1999) nonlinear observer to extract online low-frequency motions of a floating wind turbine. This wave filter has been widely used in full-scale dynamic positioning systems of ships and rigs. In CYBERLAB, it will be used in hydrodynamic laboratories as part of a cyber-physical empirical setup, see Sauder (2021).

The present project aims validating the use of the nonlinear observer, by applying a software-in-the-loop approach involving the SIMA workbench and HLCC, which is SINTEF Ocean's Hydrodynamic Lab Centralized Control software.

Scope of work

1) Perform a thorough review of the nonlinear observer: formulation, proof of convergence, performance criteria, meaning of the tuning parameters involved, and applications.

2) Define meaningful comparison criteria to assess the performance of the observer, compare several tuning parameters.

3) Validate the performance of the observer using a software-in-the loop approach. The objects of study will be a semi-submersible equipped with a 12 MW wind turbine.

4) Propose a streamlined tuning procedure depending on the floater properties.

5) Summarize results in a report (see guidelines below), and in a presentation.

Execution of the work

The student shall at startup provide a week plan of work for the project period, with main activities and milestones.

Every week 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 during the recent week, and 2) work planned to be done next week. The student should flag needs for additional meetings with his/her supervisor, others than those already planned.

The scope of work may prove different from 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 various steps in the deduction.

The candidate might have to sign a non-disclosure agreement to access relevant data for its project.

Project report

The report shall be organized in a logical structure to give a clear exposition of background, problem, design, results, and critical analysis. 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. It shall be written in English and contain the following elements: title page, abstract, project definition, 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, method, results and analysis, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. LaTeX typesetting is preferred.

The original contribution of the candidate and material taken from other sources shall be clearly identified. 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.

NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

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

Start date: August 31st , 2022

Due date - fall project: December 16th 2022

Due date - MSc thesis: June 2023

Supervisor: Thomas Sauder

Abstract

This master thesis explores the application and performance of a state observer algorithm within a software-in-the-loop (SIL) setup. Software-in-the-loop testing is a method used in model testing where control systems are tested in a simulated environment. The control system algorithm is a Nonlinear Passive Observer, proposed by Thor I. Fossen. SIMA is used for simulating the virtual environment, explicitly focusing on replicating a physical experiment for motions of the substructure on the model of a floating offshore wind turbine design known as INO WINDMOOR 12 MW.

Physical experiments are usually conducted in an ocean basin with a model-scale version of the turbine. Wind turbine testing is known for being difficult due to the scaling effects of forces from both wind and waves. Adaptation of a cyber-physical experimental setup aims to make these experiments more realistic by simulating the dynamics that are difficult to encapsulate in a physical testing environment. In this case, the mooring system is simulated and correct forces are applied by the use of tension lines controlled using Cable Driven Parallel Robots (CDPR).

The research aims to verify the feasibility and effectiveness of the NPO algorithm as a wave filter for the structure. Using HLCC, a software developed by SINTEF, it is possible to simulate the observer as a functional mock-up unit (FMU) and connect it to SIMA. Wave filtering is the process of removing the high-frequent component in the measured position of a floating body exposed to waves. The observer algorithm enables the reconstruction of low-frequent motions based on measured signals. By providing accurate estimates of the low-frequency response for the floater one is able to better estimate nonlinear wave loads and hence optimize turbine and mooring system design.

The findings indicate that the NPO algorithm can be used within this SIL setup, where both SIMA and HLCC prove to be valuable tools. It is crucial to possess a thorough understanding of the model to be used in the model-based estimator, as insufficient knowledge introduces uncertainty, hindering optimal performance and algorithm tuning. This research contributes to the broader objective of developing more efficient and streamlined solutions for the mooring system of offshore wind turbines, making it a competitive and favorable option for clean energy generation, aligned with the green shift initiative.

Sammendrag

Denne masteroppgaven utforsker anvendelsen og ytelsen til en tilstandsestimator i en “programvare-i-sløyfe” (SIL) konfigurasjon. “Programvare-i-sløyfe” er en metode som brukes i utvikling av modeller der kontrollsystemer kan testes i et simulert miljø. Kontrollsystemalgoritmen er en ikke-lineær passiv tilstandsestimator (NPO), utledet av Thor I. Fossen. SIMA brukes til å simulere det virtuelle miljøet, med fokus på å gjenskape et fysisk eksperiment for bevegelser av flyterstrukturen til modellen av en flytende offshore vindturbin kalt INO WINDMOOR 12 MW.

Fysiske eksperimenter gjennomføres som regel i et havbasseng med en modellskala versjon av turbinen. Slike eksperimenter er kjent for å være utfordrende på grunn av skalaeffektene forårsaket av krefter fra både vind og bølger. Ved å tilpasse oppsettet etter en hybrid løsning, kan eksperimenter gjøres mer realistiske ved å simulere dynamikken som er vanskelig å fange opp i det fysiske testmiljøet. I dette tilfellet simuleres forankringssystemet til turbinen, og dets krefter påføres ved bruk av spenningslinjer kontrollert av kabeldrevne parallelle roboter.

Forskningen har som mål å verifisere tilpasningen og effektiviteten til NPO-algoritmen anvendt som et bølgefilter for strukturen. Ved hjelp av HLCC, en programvare utviklet av SINTEF, er det mulig å simulere tilstandsestimatoren som en funksjonell mock-up-enhet (FMU) og koble den til SIMA. Bølgefiltrering omfatter prosessen med å fjerne den høyfrekvente komponenten i den målte posisjonen til et flytende legeme som er eksponert for bølger. NPO-algoritmen gjør det også mulig å rekonstruere lavfrekvente bevegelser basert på målte signaler. Ved å gi nøyaktige estimater av den lavfrekvente responsen til flyteren, er man i stand til å bedre estimere ikke-lineære bølgelaster på strukturen og dermed optimalisere vindturbin- og forankringssystemdesignet.

Resultatene indikerer at NPO-algoritmen kan brukes innenfor dette SIL-oppsettet, hvor både SIMA og HLCC viser seg å være nyttige verktøy. Det er avgjørende å ha en grundig forståelse av modellen som skal brukes, da utilstrekkelig kunnskap fører til usikkerheter som kan hindre optimal ytelse og justering av algoritmen. Denne forskningen bidrar til målet med å utvikle mer effektive og lønnsomme løsninger for forankringssystemet til offshore vindturbiner. Dette vil igjen bidra til å gjøre offshore flytende vind til et konkurransedyktig og gunstig alternativ for produksjon av ren energi, i tråd med den grønne skiftet.

Preface

This thesis presents the work done in the final part of the Master of Science degree in Marine Technology at the Norwegian University of Science and Technology(NTNU), with a specialization in Marine Cybernetics. The thesis was written during the spring of 2023 and is a continuation of the work done in a project thesis during the fall of 2022. The overall workload of the master thesis is equivalent to 30 ECTS and has been carried out solely by the author.

The objective of this thesis was to examine the feasibility and effectiveness of the NPO algorithm used for wave filtering in a simulated experimental environment. This investigation was conducted using the simulation tools HLCC and SIMA. An initial model of the observer was provided for the pre-project which further was expanded upon in this thesis. To facilitate a comprehensive evaluation of the observer's performance, various experiments were conducted which provided valuable insights into the behavior of the observer and demonstrated the system's response.

I would like to extend my sincere gratitude to my supervisor, Thomas Michel Sauder, for his invaluable assistance, patience, and guidance throughout the past year. I am very grateful for him going the extra mile to help me overcome the challenges I encountered.

Thanks to Vishnu Ramachandran Nair Rajasree for providing the model of the wind turbine floater used in SIMA for this project, which greatly contributed to its success. I would also like to acknowledge my friends at office B362 for their engaging discussions, constant encouragement, and the enjoyable moments we shared. Their support and camaraderie have been instrumental in making this journey memorable.

I present this master's thesis, hoping that it contributes to the advancement of knowledge in this exciting field. May it inspire further research, spark new ideas, and pave the way for innovative solutions that will shape the future of marine technology.

Trondheim, 11th of June 2023

Caspar MB Eide

Table of Contents

Project Description i

Abstract i

Sammendrag ii

Preface iii

Contents viii

List of Figures viii

List of Figures ix

List of Tables xi

List of Tables xii

Abbreviations xviii

1 Introduction 1

1.1 Motivation 1

1.2 Background 1

1.2.1 Floating offshore wind turbine technology 1

1.2.2 Historical development of physical testing 2

1.2.3 The reality today 2

1.2.4 Cyber-physical testing and principles 4
1.2.5 Case-study: INO 12MW Floating Wind Turbine 6
1.3 Objective and scope 8
1.4 Main contribution 9
1.5 Structure and organization of the thesis 10

2 Nonlinear Passive Observer theory 11
2.0.1 Summary of literature review 11
2.1 Wave theory 12
2.2 Observer theory 13
2.2.1 Dynamic system definition 13
2.2.2 Nonlinear systems 14
2.2.3 Observability 15
2.3 Wave filtering 15
2.3.1 Why a NPO 16
2.4 Passivity 17
2.5 Stability 19
2.5.1 Lyapunov theory 21
2.6 Formulation of NPO 22
2.6.1 System model 23
2.6.2 Observer equations 24
2.6.3 Convergence of the NPO 27
2.7 Involved tuning parameters and their meaning 29
2.7.1 Bias modelling 31

3 Simulation-based testing and verification study 32
3.0.1 Verification studies 32
3.1 Software-in-the-loop testing and method 34
3.2 SIMA 34
3.2.1 Simulation model description 35

3.2.2 Deviations between project report and SIMA model 36

3.2.3 Linear and quadratic damping 36

3.2.4 Wave simulation 37

3.2.5 SIMA in SIL setup 38

3.3 HLCC 39

3.3.1 Using HLCC for simulations 40

3.3.2 SIMA-Hybrid 42

3.3.3 Observer 45

3.3.4 Mooring 45

3.3.5 Allocation 46

3.3.6 Winch control 46

3.4 How is data processed - experimental method 47

3.4.1 Post-processing in MATLAB 49

3.4.2 Performance criteria 49

3.4.3 Workflow in post-processing 50

3.4.4 Collection of simulation data 53

3.5 Model verification 53

3.6 Tuning of observer algorithm 55

4 Results 56

4.1 Model verification 56

4.1.1 Discussion of model verification 58

4.2 Updated model parameters 59

4.3 Tuning 60

4.3.1 Bias tuning, K3 60

4.3.2 Velocity injection gain, K4 62

4.3.3 Position injection gain, K2 64

4.3.4 Final tuning in surge 65

4.3.5 Wave filter tuning, K1 66

4.3.6 Bias smoothing, Tb 67

4.4 Wave peak period assumed by observer, $T_{p,obs}$ 68

4.5 Response to different angles 69

4.6 Tuning yaw 71

4.6.1 Discussion 72

4.7 Performance when exposed to different sea-states 73

4.7.1 Discussion of changing wave periods 75

4.8 Increasing H_s and T_p 76

4.9 Discussion 77

4.9.1 General remarks 77

4.10 Sources of error 78

5 Conclusions 80

5.1 Concluding remarks 80

5.2 Further work 81

References 83

Appendix i

A Appendix A: Relevant Theory ii

A.1 Added mass ii

A.2 Damping ii

A.3 Reference frame iii

A.4 Model scaling v

A.5 Controllability vii

A.6 Sampling and the Nyquist frequency vii

A.7 Lipschitz continuity vii

A.8 Small-gain theorem viii

A.9 Derivation of quadratic damping force in surge ix

B Appendix B: Plots x

B.1 Model verification: decay x

B.2 K3 tuning xi

B.3 K4 tuning xii

B.4 K2 tuning xiii

B.5 K2 tuning with updated K3 xiii

B.6 Damping factor xiv

B.7 Cutoff frequency xv

B.8 Varying T_b xvi

B.9 Varying T_b for $K_3 = 10$ xvii

B.10 Response to different angles xviii

A 30 degrees environment xviii

B 60 degrees environment xix

C Sway test - 90 degrees environment xix

D Surge test - 180 degrees environment xx

B.11 Yaw tuning xxi

A K4 xxi

B K2 xxii

B.12 Different sea-state simulations xxiv

A $Hs = 3.75$ and $T_p = 7$ xxiv

B $H_s = 14.97$ and $T_p = 14$ xxv

B.13 Comparison of signals in HLCC xxvi

B.14 Initial simulation parameters xxviii

B.15 Updated model parameters xxix

B.16 WP4 model xxx

B.17 Tuned model xxxi

C Appendix C: Code and models **xxxii**

C.1 Simulink model of observer xxxiv

List of Figures

1.1 Four most common concepts for floating offshore wind turbines. . . . 2

1.2 Capital expenditure of installing offshore wind turbines 3

1.3 Cyber-physical substructures for marine testing 5

1.4 Control system perspective on cyber-physical setup for testing 5

1.5 Concept figure of INO WINDMOOR 12MW FWT 7

2.1 Sum and Difference frequencies 12

2.2 Low- and wave-frequency motions 12

2.3 Passive negative feedback system 18

2.4 Lyapunov level surfaces 22

2.5 Block diagram of the NPO 25

2.6 Block diagram of error dynamics to NPO 26

2.7 Passive block diagram configuration of the NPO 26

2.8 Bode plot of wave filtering 30

3.1 V-model for model-based design 32

3.2 Software-in-the-loop system overview 34

3.3 Overview of example model of INO12 MW SIMA 35

3.4 Detailed floater model in SIMA 35

3.5 Full SIMO model overview with actuation lines 35

3.6 JONSWAP wave spectrum from SIMA for $H_s = 6$ and $T_p = 7.55$ . . . 37

3.7 Detailed SIMA model for software-in-the-loop system 39

3.8 HLCC divided into FMUs 40

3.9 HLCC initialization 41

3.10 Overview of a DLL in HLCC 41

3.11 Overview of a DLL in HLCC 42

3.12 UDP settings in SIMA-HYBRID, HLCC 43

3.13 Interface of SIMA-HYBRID in HLCC 44

3.14 SIMA-HYBRID block with subroutines 44

4.1 ODE fit vs SIMA response in surge 57

4.2 Simulations with updated model parameters 60

4.3 Bias plotted for various values of K_3 61

4.4 ν plotted for various values of K_4 63

4.5 Indicator values for $K_3 = 3$ and varying K_2 in surge only 65

4.6 Simulation with tuned parameters in head seas 66

4.7 Surge indicator values for change in T_c. 67

4.8 Surge indicator values for change in $T_{p,obs}$. 69

4.9 Simulation with the environment from 60 degrees. 70

4.10 Sliced plot of $\hat{\eta}$ for $H_s = 6.19$ and $T_p = 12$. 75

4.11 Platform cancellation period simulation 76

A.1 Degrees of freedom for a ship iv

A.2 Visual Lipschitz continuity viii

A.3 Derivation of quadratic damping coefficient in surge. ix

B.1 Decay response for sway and yaw. x

B.2 Bias comparison with K3=1 and K3=0.5. xi

B.3 Sliced plot of η for various values of K_4. xii

B.4 Bias plotted for various values of K_4. xii

B.5 Simulation with varying values of K_2 xiii

B.6 ν plotted for various values of K_2 with updated K_3. xiii

B.7 Sliced plot of $\hat{\eta}$ for $\zeta \in [0.1, 0.3, 0.5, 0.7]$. xiv

B.8 $\hat{\nu}$ for $\zeta \in [0.1, 0.3, 0.5, 0.7]$. xiv
B.9 Velocity estimate for cutoff frequencies= $[0.5, 1, 1.5, 2]$ xv
B.10 Sliced position estimate for cutoff frequencies= $[0.5, 1, 1.5, 2]$ xv
B.11 Bias force for $T_b = [0.5, 1, 1.5, 2]$ xvi
B.12 Velocity estimate for $T_b = [0.5, 1, 1.5, 2]$ xvi
B.13 Bias for $K_3 = 10$ and $T_b \in [0.5, 1, 1.5, 2]$. xvii
B.14 Simulation with the environment from 30 degrees. xviii
B.15 Yaw response for environment from 30 degrees. xviii
B.16 Yaw response for environment from 60 degrees. xix
B.17 Simulation with the environment from 90 degrees. xix
B.18 Yaw response for environment from 90 degrees. xx
B.19 Simulation with the environment from 180 degrees. xx
B.20 Bias in yaw for various values of K_4 xxi
B.21 Velocity estimate in yaw for various values of K_4 xxii
B.22 Bias in yaw for various values of K_2 xxii
B.23 Position estimate in yaw for various values of K_2 xxiii
B.24 $\hat{\nu}$ for $H_s = 3.75$ and $T_p = 7$. xxiv
B.25 Sliced $\hat{\eta}_{LF}$ for $H_s = 3.75$ and $T_p = 7$. xxv
B.26 Bias for $H_s = 14.97$ and $T_p = 14$. xxv
B.27 Sliced plot of $\hat{\eta}$ for $H_s = 14.97$ and $T_p = 14$. xxvi
B.28 Superviewer signal comparison xxvi

C.1 Simulink model of NPO. xxxiv

List of Tables

1.1 Advantages and disadvantages of cyper-physical approach 6

1.2 Substructure parameters 7

1.3 Turbine parameters 8

3.1 Differences between values in Souza and SIMA 36

3.2 Linear damping matrix values from SIMA 36

3.3 First set of sea-states with similar steepness 38

3.4 Second set of sea-states with similar steepness 38

3.5 Positional references for decay tests 54

4.1 Mooring stiffness for three conditions 58

4.2 Fitted model parameters 58

4.3 Indicator values for varying bias gain 62

4.4 Simulation values for K_4 63

4.5 Indicator values in surge for varying values of K_3 and K_4 64

4.6 Indicator values in surge for combined tuning parameters 65

4.7 Indicator values in surge change in T_b 68

4.8 Indicator values for varying direction of environment 70

4.9 Indicator values in yaw for varying values of $K_{2,3}$ and $K_{4,3}$. 72

4.10 Indicator values in surge for different sea-states. 74

A.1 Scaling conversion factors. vi

B.1 Initial simulation parameters. xxviii

B.2 Updated model parameters. xxix

B.3 WP4, offline experiment model parameters. xxx

B.4 Tuned model parameters. xxxi

Nomenclature

α_m	Main axis
$\boldsymbol{A}$	System matrix
$\boldsymbol{B}$	System matrix
$\boldsymbol{C}$	System matrix
$\boldsymbol{D}$	System matrix
$\boldsymbol{K}$	System stiffness matrix
$\boldsymbol{M, m}$	Mass matrix
ϵ	Phase angle, performance criteria
η	NED positional vector
γ	Gain constant, peak enhancement factor, steepness factor
$\hat{\eta}$	Estimated low-frequent position vector
$\hat{\nu}$	Estimated velocity
$\hat{f}$	Online filtered signal
$\hat{g}$	LTI transfer function
λ	Wave length, scaling factor
$\mathcal{C}$	Controllability matrix
$\mathcal{H}_1, \mathcal{H}_2$	Transfer function
$\mathcal{KL}$	Class operator
$\mathcal{K}$	Class operator, gain notation
$\mathcal{L}$	Laplace operator, norm space, stability operator
$\mathcal{O}$	Observability matrix

ν	Body-frame velocity vector, viscosity
ω	Circular frequency
ω_0	Cutoff frequency
ω_c	Cutoff frequency
ω_i	Incident wave frequency
ω_n	Natural frequency
$\overline{F_i}$	Mean reference force
ϕ	Wave potential, roll angle
ψ	Yaw angle, low-pass filtered position in yaw
ρ	Density
τ_m	Mooring load vector
τ_w	Wave force vector
τ_{tot}	External force vector
θ	Pitch angle, winch angle
ξ	Wave frequency earth-fixed position
ζ	Wave amplitude, damping ratio
a	Added-mass matrix
A_j	Wave amplitude
A_w	Waterplane area
$a_{11,0}$	Surge zero frequency added mass
$a_{22,0}$	Sway zero frequency added mass
$a_{66,0}$	Yaw zero frequency added mass
b	Bias vector
C_d	Drag coefficient
C_w	Waterplane stiffness
$c_{critical}$	Critical damping
D_H	Hub diameter
d_h	Hydrodynamic damping matrix
D_l	Linear damping coefficient

d_m	Mooring damping matrix
D_q	Quadratic damping coefficient
D_R	Rotor diameter
E	Low-pass filtered position in sway
f^*	Offline filtered signal
g	Gravitational constant
H_s	Significant wave height
I_{xx}	Inertia along longitudinal axis
I_{yy}	Inertia along transversal axis
I_{zz}	Inertia along vertical axis
J	Combined linear and angular transformation matrix
k	Wave number
K_i	Gain matrix
k_i	Inline mooring stiffness
k_m	Mooring stiffness matrix
k_t	Transversal mooring stiffness
k_{rot}	Rotational mooring stiffness
L	Lipschitz constant, Length
M_{blade}	Single blade mass
M_{hub}	Hub mass
$M_{nacelle}$	Nacelle mass
MTS	Maximum Tip Speed
N	Low-pass filtered position in surge
P	Pressure
R	Linear transformation matrix
r	Low-pass filtered velocity in yaw
S	Spectrum function
s	Laplace operator
T	Draft, Period, Angular transformation matrix, Time

t	Time variable
T_c	Cutoff period
T_n	Natural period
T_p	Peak wave period
U	Incoming velocity
u	Input variable, low-pass filtered velocity in surge
V	Velocity vector, Lyapunov candidate function
v	Low-pass filtered velocity in sway
V_{in}	Cut-in wind speed
V_{out}	Cut-out wind speed
V_{rated}	Rated wind speed
x	State variable
X_{CoG}	Longitudinal centre of gravity
y	Output value
Y_{CoG}	Transversal centre of gravity
Z_{CoG}	Vertical centre of gravity

Abbreviations

List of all abbreviations in alphabetic order:

- **AR** Augmented Reality
- **CAPEX** Capital Expenditures
- **CDPR** Cable-Driven-Parallel-Robot
- **CFD** Computational fluid dynamics
- **CP** Cyber-physical
- **DLL** Dynamically Link Library
- **DNV** Det Norske Veritas
- **DOF** Degree of freedom
- **ECTS** European Credit Transfer and Accumulation System
- **FMU** Functional Mockup Unit
- **FPSO** Floating Production Storage and Offloading
- **FWT** Floating Wind Turbine
- **GES** Globally Exponentially Stable
- **HIL** Hardware-in-the-loop
- **HLCC** HydroLab Centralized Control
- **HW** Hardware
- **JONSWAP** Joint North Sea Wave Project
- **KPN** Kompetansebyggende prosjekt for næringslivet
- **KYP** Kalman–Yakubovich–Popov

- **LabVIEW** Laboratory Virtual Instrument Engineering Workbench
- **LF** Low-frequent
- **LP** Low-pass
- **MIL** Model-in-the-loop
- **MW** Megawatts
- **NED** North-East-Down
- **NI** National Instruments
- **NPO** Nonlinear Passive Observer
- **NTNU** Norwegian University of Science and Technology
- **ODE** Ordinary Differential Equation
- **OWT** Offshore wind turbine
- **PIL** Processor-in-the-loop
- **RX** Receive
- **SIMA** Simulation Workbench for Marine Applications
- **SIL** Software-in-the-loop
- **SPR** Strictly Positive Real
- **SW** Software
- **TX** Transmit
- **UDP** User Datagram Protocol
- **UN** United Nations
- **WF** Wave-frequent

Chapter 1

Introduction

1.1 Motivation

The idea for this project originates from a joint effort “competence project for business, (KPN)” initiative called CYBERLAB, carried out by SINTEF, NTNU, and Aarhus University. A joint effort project is a project in which key actors in the industry join forces to enhance overall understanding in a specific field. The purpose of the CYBERLAB project is to raise competence in methods to study the dynamical behavior of a lattice of floating structures. The project aims to optimize mooring systems for offshore floating wind turbines to be more cost- and energy efficient.

The first step toward making more efficient systems is to increase understanding of their behavior and be able to model them precisely. CYBERLAB believes that this can be obtained by combining physical and numerical experimentation in a cyber-physical approach. Data and models for the tests conducted in this report are provided by SINTEF and CYBERLAB.

1.2 Background

1.2.1 Floating offshore wind turbine technology

Moving wind turbines out into the ocean offers great advantages in terms of stronger and more consistent wind resources. More available space offers the possibility of building larger turbines which generate more power. Most offshore turbines today are mounted to the seabed by monopiles, 81.2% of the installed OWT were monopiles in 2020(WindEurope 2020). This design is favored for water depths up to 60 meters. For depths beyond 60 meters, dynamics of the monopile structure will be compromised by wave interactions (Kallehave et al. 2015). Figure 1.1 shows some concepts of turbines with floating substructures.

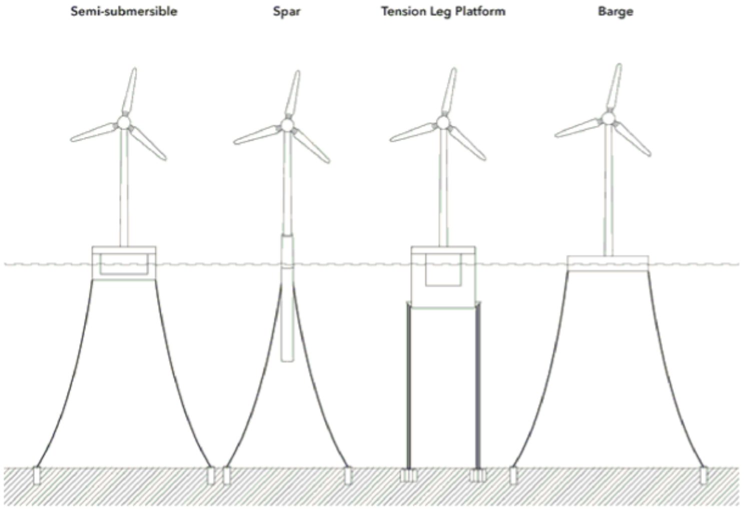


Figure 1-1 Four common floater types for wind turbines

Figure 1.1: Four most common concepts for floating offshore wind turbines. INO WINDMOOR 12MW is a semi-submersible design(DNV 2021).

1.2.2 Historical development of physical testing

The history of offshore wind turbine concept testing and marine model testing has roots in the early development of renewable energy and naval architecture, respectively. Offshore wind turbines emerged in the 1970s as a response to the oil crisis leading to an increased global interest in renewable energy sources. In 1991, the first large-scale offshore wind farm was commissioned in Vindeby, Denmark with a total capacity of 5MW, providing 2200 Danish households with power (Orsted n.d.). Since then, lower costs and technological advancements have led to increasingly larger turbines. Currently, the largest offshore wind farm project in the world is named “Dogger Bank” which aims to have a combined installed capacity of 3.6GW, providing electricity for up to 6 million homes (Farm n.d.).

Marine model testing, originating from William Froude’s experiments, evolved and expanded in the 20th century to include offshore structures. Today, it remains vital for ship and offshore structure optimization, utilizing advanced technologies like Computational Fluid Dynamics (CFD) and Augmented Reality (AR).

The intersection of these two fields can be seen in the development and testing of offshore wind turbines. Marine model testing is used to simulate the behavior of turbines in various sea conditions and to optimize the design of support structures.

1.2.3 The reality today

In 2021, it was estimated that the global average temperature had increased 1.1 degree Celsius since pre-industrialization and that the period 2015-2021 was the warmest ever recorded. The UN has set the goal of limiting the increase in average temperature to a maximum of 1.5 degree Celsius by 2050, (Nations n.d.). To reach the goal of the UN while feeding a constantly growing population, humanity needs more clean, reliable, and affordable sources of energy.

The DNV Energy Transition Outlook estimates that, due to higher and more reliable wind speeds and fewer constraints on hub heights and site locations, offshore wind will show an average annual growth of 13 % from 2020 to mid-century (Ch. 2.1 in DNV 2022, page 61). Furthermore, it expects the offshore wind capacity to increase by 56-fold from today's level and toward 2050 (Ch. 3 in DNV 2022, page 85).

Currently, offshore wind turbines are mainly individually fixed to the seabed. As the turbines are moved further offshore, floating solutions become more attractive. One of the most limiting factors for available floating wind turbines is that mooring far offshore is expensive. A recent estimate suggests that the combined costs of mooring and substructure of an offshore FWT will make up 35% of the total investment cost (Figure 1.2). Additionally, relatively large spacing between the turbines is required to minimize wake effects from the wind (Eliassen n.d.). When several turbines are moored as an interconnected grid structure, the cost and impact on the seabed of the mooring system could be significantly reduced. This raises, however, an optimization problem with distance and wake effects. A grid structure can enable a more cost- and material-effective solution for offshore wind, which might be the key to making this the preferable design for offshore floating wind turbines (T. M. Sauder n.d.).

Before connecting the wind turbines together and assessing the interactions between several turbines, it is crucial to have more information on the loads and motions that affect each individual turbine. This thesis is part of a project aiming to streamline the testing and estimation of these loads on floating turbines by applying principles from cyber-physical testing.

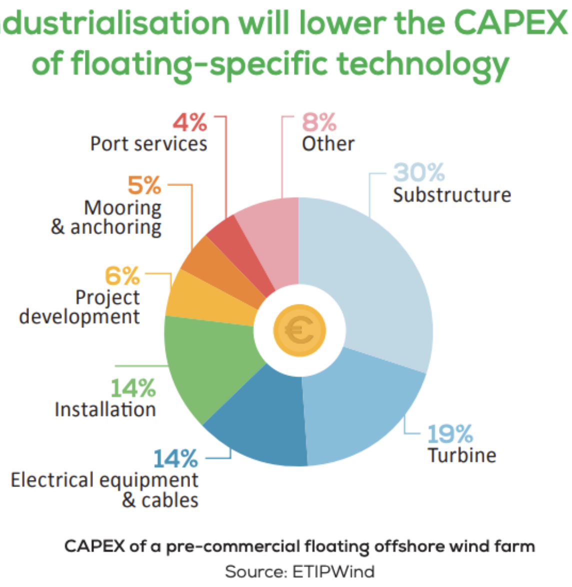


Figure 1.2: Capital expenditure of installing offshore wind turbines(Committee 2020).

1.2.4 Cyber-physical testing and principles

Cyber-physical testing is a method that combines physical experiments with numerical simulations. An experiment using this technique is performed in real-time and controlled using a control system. The main advantage of this method is that it enables the testing of coupled systems that can't be tested using either physical experiments or numerical models alone. The cyber-physical approach is attractive for problems with infrastructure issues, scaling incompatibilities, low efficiency, or optimization problems as the setup allows an efficient change in boundary conditions.

The three main purposes of conducting experimental testing for marine operations defined by Steen (2014) are:

- To achieve relevant design data to verify the performance of actual concepts for ships and other marine structures
- Verification and calibration of theoretical methods and numerical codes
- To obtain a better understanding of physical problems.

Humans have performed physical and empirical experiments for as long as they have existed, the early historical developments of cyber-physical experiments are described in Nakashima (2020) and the first experiment was likely conducted for buildings and infrastructures in 1975 by Takanashi (Takanashi 1975). It is only in the recent 10 years that this method has been applied to marine experiments (Abbiati 2022).

For testing objects in a marine environment, three main facilities have been developed for various testing. These are towing tanks, cavity tanks, and ocean basins. All of these have been designed to imitate the real world as much as possible.

In recent years and with the development of new technology, turbines have grown larger and been designed for use farther offshore. This also implies new challenges for testing, where conventional facilities have become too small to ensure sufficient similarity between the model and full scale. As a result of this, cyber-physical testing has become more relevant. The method is also known as hybrid testing or Hardware-in-the-loop (HIL) testing.

The principle of cyber-physical testing is to divide the setup into two substructures, one physical with the model and one numerical which simulates the dynamics that either are difficult to model or won't fit in the traditional ocean basin. The two systems are then connected using an active truncation system, controlled by a control system responsible for balancing the actuated forces and velocities. This will ensure compatibility and equilibrium between the substructures.

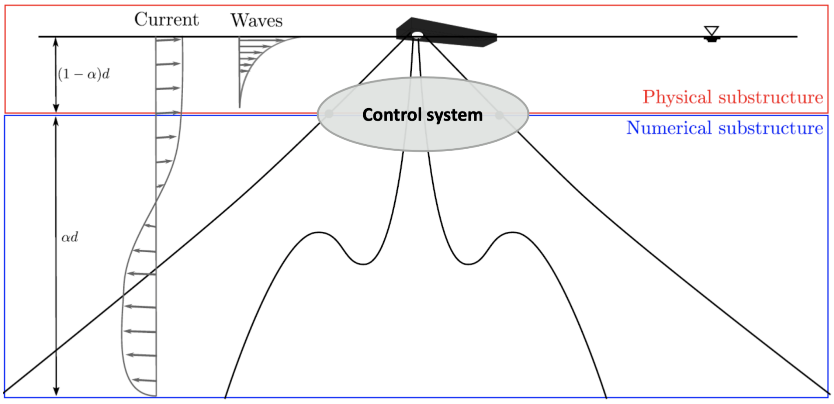


Figure 1.3: Cyber-physical substructures for marine testing(T. Sauder 2022c).

The figure above portrays a typical setup for maritime experimentation using cyber-physical principles. The physical substructure is the model located in an ocean basin equipped with wave-makers capable of producing realistic sea states on a model scale. The numerical substructure here represents the mooring system of the model for testing. This is helpful as most ocean basins are not deep enough to fit a fully scaled mooring system, simply due to their depth and difficulties in modeling weight, forces, and deflections in acceptable time. The mooring in this case will be modeled based on catenary equations. Experiments with wind and deep water operations are other examples where dynamics can be simulated for a better dynamical representation. A visual representation of the various software and hardware components and their connection in a cyber-physical control system is given in the figure below.

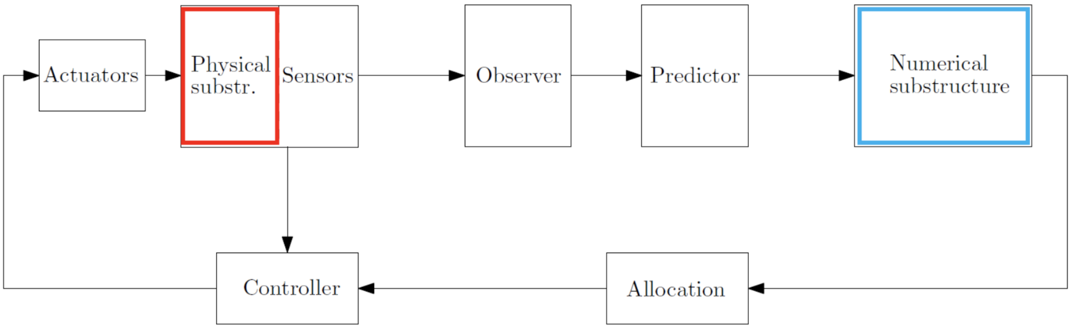


Figure 1.4: Control system perspective on cyber-physical setup for testing(T. Sauder 2022c).

From the two figures above, it becomes obvious that this kind of experimental setup yields multiple layers of complexity. The two most apparent are perhaps stability and delay. These two phenomena are excited by so-called artefacts, a collective term for calibration error, actuator dynamics, noise, jitter, loss of signal, etc. Further, this leads to the study of fidelity for the system which is a critical part as the loss of fidelity often occurs before the loss of stability (T. Sauder 2022a).

The concept of cyber-physical testing is still under development and continuous improvements are being made to ensure correct and accurate results. One of the biggest weaknesses of this method today is that it requires a system that responds fast enough. Computational delay, actuator, and sensor dynamics are limiting factors.

Summarizing advantages and disadvantages of cyber-physical testing

Advantages	**Disadvantages**
Cost-effective: Simulations have lower cost than physical materials and equipment	*Complexity*: Requires expertise in physical and computational modeling
Flexibility: Modifiable and adjustable model, iterative design and optimization	*Fidelity*: Not fully capturing the complexity and nuance of real-world systems
Efficiency: Faster simulations and dynamics	*Accuracy*: Accuracy of model limited to quality of inputs and outputs
Safety: Conduct experiments in virtual environment eliminating dangerous and hazardous conditions	*Hardware/software issues*: Easily impacted by failure and inaccuracies in both software and hardware

Table 1.1: Summarizing some important advantages and disadvantages of a cyber-physical approach.

1.2.5 Case-study: INO 12MW Floating Wind Turbine

The model used as a case study in this thesis is the INO WINDMOOR 12MW floating wind turbine, a concept delivered by Inocean and a part of WINDMOOR, a KPN project funded by the Research Council of Norway and offshore wind industry. The project spans over 4 years, commencing in 2019 (SINTEF n.d.). The project's primary objective is to enhance the load analysis methods to facilitate the creation of even more efficient designs for floating wind turbine farms (Inocean n.d.). This objective aligns with the goal of CYBERLAB.

The turbines are to be installed at a water depth of 150 meters and the design is based on the IEA 10MW design which is an updated version of the DTU 10MW. The design has further been scaled up to meet the rated electrical power supply of 12MW. Some relevant characteristics are presented in Figure 1.5 and tables 1.2 and 1.3.

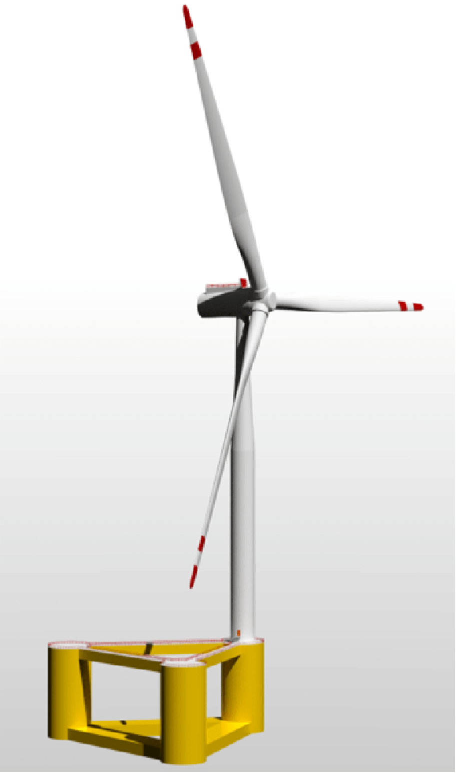


Figure 1.5: Concept figure of INO WINDMOOR 12MW FWT(Inocean n.d.)

INO WINDMOOR 12MW Substructure Parameters			
Parameter	*Explanation*	*Symbol*	*Unit*
M	Mass	14176	t
T	Draft	15.5	m
$T_{n,1}$	Natural period in surge	97.3	s
$T_{n,2}$	Natural period in sway	98	s
$T_{n,6}$	Natural period in yaw	88	s
L_p	Pontoon length	46	m
$C_{dy,p}$	Pontoon drag coefficient	4817.5	[-]
L_c	Column length	15.5	m
$C_{dy,c}$	Column drag coefficient	7687.5	[-]
X_{CoG}	Longitudinal centre of gravity	0.38468	m
Y_{CoG}	Transversal centre of gravity	0.0	m
Z_{CoG}	Vertical centre of gravity	4.3178	m
I_{xx}	Inertia along longitudinal axis	$2.7328 \cdot 10^{10}$	$kg \cdot m^2$
I_{yy}	Inertia along transversal axis	$2.7064 \cdot 10^{10}$	$kg \cdot m^2$
I_{zz}	Inertia along vertical axis	$1.2968 \cdot 10^{10}$	$kg \cdot m^2$
$a_{11,0}$	Surge zero frequency added mass	$8.2 \cdot 10^6$	kg
$a_{22,0}$	Sway zero frequency added mass	$8.2 \cdot 10^6$	kg
$a_{66,0}$	Yaw zero frequency added mass	$8.3 \cdot 10^9$	$kg \cdot m^2$

Table 1.2: INO WINDMOOR 12MW Substructure parameters(Souza 2021)

Turbine Parameters			
Parameter	*Explanation*	*Symbol*	*Unit*
D_R	Rotor diameter	216.9	m
D_H	Hub diameter	5.0	m
L_B	Blade length	105.4	m
V_{in}	Cut-in wind speed	4.0	m/s
V_{rated}	Rated wind speed	10.6	m/s
V_{out}	Cut-out wind speed	25.0	m/s
MTS	Maximum tip speed	88.6	m/s
M_{blade}	Single blade mass	60 000	kg
M_{hub}	Hub mass	63 024	kg
$M_{nacelle}$	Nacelle mass	600 000	kg

Table 1.3: INO WINDMOOR 12MW Turbine parameters(Souza 2021)

Not all of the above characteristics are directly relevant to this thesis, but they serve as a reference of magnitude and sizes, see Souza 2021 for more details.

1.3 Objective and scope

The main objective of this thesis is to:

verify the wave filtering ability of a nonlinear passive observer for wind turbine application before physical basins tests by using cyber-physical principles applied to software-in-the-loop testing

The observer should be able to completely separate the wave-frequent and low-frequent components of the motion of the floater based on its simulated position. It should also provide an estimate of the velocity which is within the magnitude of the velocity which can be derived from the low-pass filtered position. The bias estimate of the observer should converge towards a constant value without too much high-frequent signal content.

Ideally, the observer should also be tuned such that it fulfills all the above-mentioned criteria for all three degrees of freedom and for varying environmental conditions. Another desired outcome from this testing is to propose measures that will streamline marine software-in-the-loop testing and make simulations more efficient and reliable.

Both the algorithm and the provided simulation model are complex systems. There is put some emphasis on understanding how the systems work separately and together in a common simulation environment. The tools used to perform simulations are not publicly available and not known to the writer at the beginning of this project.

This report builds on an earlier project which studied wave filtering using the nonlinear passive observer design proposed by Fossen and Strand (Fossen 1998). Relevant literature on the observer will be presented and can be used as a basis for later experiments on floating wind turbine parks using shared mooring. The observer extracts low-frequency (LF) motions and loads. Its performance and ease of tuning will be demonstrated through numerical experiments.

Research is continued by testing wave filtering with a Nonlinear Passive Observer and measuring its performance in a simulated real-world environment. The previous project tested the isolated observer algorithm's effectiveness with synthetic input signals.

To achieve this, a software-in-the-loop setup will be used, and the wave loads will be connected to the observer through the HLCC software program, which is specifically designed for hydrodynamic lab tests.

1.4 Main contribution

The main contributions of this thesis can be summarized as follows:

- Literature study of the nonlinear passive observer algorithm, derivation and areas of use. All relevant theory presented in Chapter 2.
- Adjusting an algorithm mainly used for station-keeping purposes of vessels to fit a floating wind turbine design.
  - Practical study of the nonlinear passive observer using synthetic signals in Matlab and Simulink and adjusting the observer to handle input from simulations in SIMA.
  - Proposing a streamlined method of tuning the observer based on its input.
- Building and verifying the model of floater by simulating decay tests. Displaying the importance of correct modeling for observer performance.
- Displaying the use of FMUs in HLCC and SIMA as simulation tools.
- Proposing a simulation setup for cyber-physical experiments with allocation, observer and mooring.
- Simulations of scenarios with different disturbances to explore the performance of the model within the validity range.

1.5 Structure and organization of the thesis

This project is organized into 5 main chapters. A brief description of the contents follows below:

Chapter 1: Introduction Introducing the project including background, motivation, and contributions.

Chapter 2: Nonlinear Passive Observer A review of the nonlinear passive observer algorithm proposed by Thor I. Fossen and relevant theory.

Chapter 3: Verification study Presentation of method used in verification testing. Explaining the setup for software-in-the-loop testing and its components. This includes the simulation tools SIMA, HLCC, and their coupling.

Chapter 4: Results Presentation of the most relevant results from the study and discussion of their significance.

Chapter 5: Conclusion Concluding remarks and suggestions for further work

Bibliography Presenting all references used in the development of this project

Appendices

- Appendix A: left out, but relevant theory for the interested reader.
- Appendix B: Figures, tables, and plots.
- Appendix C: Code and models

Chapter 2

Nonlinear Passive Observer theory

This chapter presents the relevant theory for the development of a *Nonlinear Passive Observer* for wave-filtering applications on ship structures. Further work will apply the theories described here on floating offshore wind turbines. Most of the theory presented in this chapter is based on the work done in the preliminary report written the past fall as a build-up to this thesis. The chapter will first present the required background theory before deriving the Nonlinear Passive Observer Algorithm.

2.0.1 Summary of literature review

Nonlinear wave loads and motions are frequently studied and several numerical methods have been created to calculate loads on floating structures. WAMIT-S and Hydrostar, based on potential flow theory, are currently the most popular methods. The paper of T. Sauder 2021 suggests an additional method for estimating nonlinear wave loads through a cyber-physical setup.

Regarding previous work done on estimation techniques and in particular design of a Nonlinear Passive Observer (NPO), academic work done by Thor I. Fossen has been reviewed. This includes different methods and applications of the algorithm for usage on ship structures (Fossen 1991, Fossen 1994, Fossen 1998, Fossen 2021).

As theoretical background and support, the works of Khalil, Chen, Faltinsen, and Sørensen have been reviewed. Keywords include non-linear control, passivity, Lyapunov, Kalman filtering, hydrodynamics, and linear control.

2.1 Wave theory

A sea state is composed of several harmonic waves with different frequencies. Two incident waves with different frequencies will cause a nonlinear pressure load on the structure at both the sum, $(\omega_2 + \omega_1)$ and difference frequency $(\omega_2 - \omega_1)$, in addition to zero-frequency and mean loads. This is caused by the multiplication of two harmonic functions at different frequencies (T. Sauder 2022b). Visually, this is shown as:

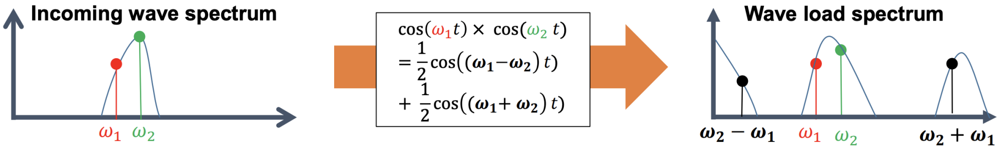

Incoming wave spectrum
ω_1 ω_2
$\cos(\omega_1 t) \times \cos(\omega_2 t)$
$= \frac{1}{2}\cos((\omega_1 - \omega_2)t)$
$+ \frac{1}{2}\cos((\omega_1 + \omega_2)t)$
Wave load spectrum
$\omega_2 - \omega_1$
ω_1 ω_2
$\omega_2 + \omega_1$

Figure 2.1: Visual representation of the creation of sum- and difference frequencies(T. Sauder 2022b).

As seen in the figure, this phenomenon will also imply a wider frequency range of loads. For moored structures, one is most concerned with the forces caused by different frequencies as these are likely to cause resonant motions in the horizontal plane (Faltinsen 1990). Figure 2.1 also shows why different frequencies are associated with low-frequency loads and motions. Figure 2.2 shows the separated frequency components of a single wave.

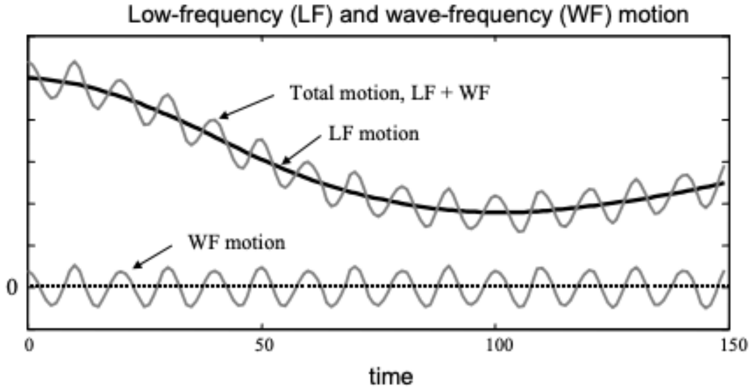


Figure 2.2: Low- and wave-frequency motions(Sorensen 2018).

A wave can be divided into a low-frequent and a high-frequent component. The main goal of this thesis is to use an observer algorithm to split an incoming wave into these two components. Further, the low-frequent component is of particular interest when performing load analysis on the body in waves.

The motion of a floating object in waves can be described using the concept of added mass, which is the apparent increase in the mass of the object caused by its motion through the fluid. The added mass depends on the frequency of the motion of the object through the fluid and is typically proportional to the mass of the fluid displaced by the object. This is described in more detail in Section A.1.

2.2 Observer theory

An observer, also known as a state estimator, is a mathematical algorithm used to estimate an immeasurable internal state of a system based on its input and output signals. By comparing the estimated state with the actual measurements, the observer can filter out noise and disturbances in the signals, enhancing the accuracy and reliability of the system's output. The observer essentially acts as a virtual sensor, filling in the gaps where direct measurements are unavailable or unreliable. There are mainly two types of observers:

- *Model-based estimator*: an observer algorithm based on a dynamic model of the system and measured signals
- *Signal-based estimator*: an observer algorithm based on measured signals and statistical principles

2.2.1 Dynamic system definition

Dynamic systems refer to mathematical models used to describe the behavior and evolution of systems over time. A dynamic system typically consists of a set of variables representing a state that changes over time, based on mathematical equations or rules. The system is often categorized based on the properties of its state variable.

A linear system is a system where all equations are proportional and can be described as a linear combination of its state variables. Linear systems will also satisfy the *superposition principle.* If the system produces the same response regardless of when it is excited by an input or impulse, it is a Linear Time-Invariant (LTI) system and can be presented using the following state-space equations:

$$\dot{x} = \boldsymbol{A}x(t) + \boldsymbol{B}u(t) \tag{2.1a}$$

$$y = \boldsymbol{C}x(t) + \boldsymbol{D}u(t) \tag{2.1b}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are real matrices, *x(t)* is the system state at time t, *u(t)* is the input to the system at time t, and y(t) is the observed output of the system at time t. The system matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ hold values for the general dynamics of the system, i.e. the mass, stiffness, and damping coefficients, while the input *u(t)* for example represents a voltage or a force (Chen 2013).

Most real-world systems are time-varying and will produce different responses dependent on when the impulse is applied. This implies that the system dynamics will then change so that the system matrices will also be time-dependent(Chen 2013).

For some applications, it is convenient to consider a mapping from the input to the output for a given impulse response. This is usually done by means of a transfer function. For a one-degree-of-freedom Linear Time-Invariant (LTI) system, this is calculated by dividing the Laplace transform of the output by the Laplace transform of the input.

$$\hat{g} = \frac{\hat{y}}{\hat{u}} = \frac{\mathcal{L}[output]}{\mathcal{L}[input]} \tag{2.2}$$

2.2.2 Nonlinear systems

A system that cannot be described by linear equations, is nonlinear. Such systems often require more sophisticated tools and methods to analyze and understand their behavior. One of the key challenges in nonlinear systems theory is to identify and classify the various types of nonlinearities that can arise in a system and to develop methods for studying the behavior of these systems in different regimes. For a nonlinear system, the state-space representation is:

$$\dot{x} = f(t, x, u) \tag{2.3}$$

$$y = h(t, x, u) \tag{2.4}$$

These two relations are generalized from the equations of a linear system (2.1), as unspecified functions. In some cases, it is convenient to state the equations without a specific presence of input $u(t)$:

$$\dot{x} = f(t, x) \tag{2.5}$$

This is called an unforced state equation, where input is not necessarily zero, but can be expressed as a specific function of time, a given feedback function, or a combination denoted $u = \gamma(t, x)$. By replacing $u(t)$ with γ in the general state equation (2.3), it becomes unforced. Another special case is when the state equation doesn't depend explicitly on time:

$$\dot{x} = f(x) \tag{2.6}$$

These are called time-invariant or autonomous systems, which are independent of time shifts (Khalil 2015b). In proofs involving non-linear state-space representations, it is often required that the function be piecewise continuous and locally Lipschitz over the relevant time domain. For practical applications, it is enough to consider that the derivative of a function is continuous and bounded. This is explained in more detail in Appendix A, Section A.7.

2.2.3 Observability

In order to develop an observer algorithm for a dynamical system, observability is necessary. An observable system enables the estimation of an initial state variable in finite time when the inputs and outputs are known. A linear time-invariant system is observable if the following matrix has full column rank, and where n denotes the number of state variables(Chen 2013):

$$\mathcal{O} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \dots \\ \boldsymbol{CA}^{n-1} \end{bmatrix} \tag{2.7}$$

2.3 Wave filtering

Signal filtering is the process of manipulating a signal to enhance its quality, accuracy, or interpretability while reducing unwanted components such as noise or interference. It involves applying a mathematical algorithm or filter to the input signal to alter its frequency content or temporal characteristics. Different types of filters, such as low-pass, high-pass, band-pass, or notch filters, are employed depending on the specific requirements of the application. The goal of wave filtering is to remove the wave-frequent motions shown in figure 2.2 and further reconstruct an estimate of the low-frequent motions based on noisy measurements of heading and position (Fossen 2021).

Under normal operating conditions, the control of a surface body is concentrated around the low-frequency motions, avoiding the correction of oscillatory motions caused by first-order wave forces. This also has practical application, as adjustment for wave-frequency motions would lead to wear and tear of thrusters, high fuel consumption, and unacceptable operational conditions (Sorensen 2018).

When it comes to floating offshore wind turbines, low-frequency motions are of interest when designing mooring systems since low-frequent motions can occur at resonance frequencies and therefore trigger large motions and loads (Faltinsen 1990, Chap.5). The first step of wave filtering is to remove the wave content with a high frequency which is typically done using a low-pass filter. The Laplace transfer function of a low-pass filter is given as:

$$H(s) = \frac{\text{Output}}{\text{Input}} = K\frac{1}{\tau s + 1} \tag{2.8}$$

where τ is the time constant and K is the filter gain. The frequency response of the filter can alternatively be written as:

$$H(s) = \frac{\omega_0}{s + \omega_0} \tag{2.9}$$

For irregular waves, the filtering procedure is done by dividing the wave into individual sinusoidal components that act together. A notch filter can then be used to extract a specific range of frequencies from the range of sinusoidal components. When filtering the wave, this attenuation range will be based on a wave model, which will be derived later and can for example be expressed as:

$$h(s) = \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{(s + \omega_n)^2} \tag{2.10}$$

$\zeta \in (0, 1)$ controls the magnitude of the notch, while the notching frequency, ω_n is chosen close or equal to the peak frequency of ω_0. If the single-notch filter does not cover the whole wave frequency range, adding more notch filters in series will solve this (Fossen 2021).

2.3.1 Why a NPO

NPO is an abbreviation for “Nonlinear Passive Observer”. The purpose of an observer is to estimate immeasurable inputs which then can be used by a control system. The NPO in this project is a model-based observer, which implies that for it to function correctly it needs an accurate mathematical model of the system with precise estimates of relevant characteristics such as mass, stiffness, and damping. The algorithm is favored over a low-pass filter such as a Butterworth filter due to its estimation capabilities with less phase distortion.

The state-of-the-art and most widely used observer is the Kalman filter design. One of the drawbacks of this method and the motivation factor for the development of the NPO is that the Kalman filter has a relatively large number of parameters. These parameters must be tuned through trial and error, and the process is often time-consuming. The NPO guarantees global convergence of all estimation errors to zero, contrary to the Kalman filter.

For this study, an NPO proves to be very convenient, as it can also be used for wave filtering. The filtering is obtained by using a cutoff frequency equal to the wave spectrum peak frequency, resulting in a notch effect. When the dominating wave frequency is known and hence the motion frequency to be filtered, the observer should produce a notch filter effect in the frequency range of disturbances caused by incident waves (Fossen 1998).

As with similar algorithms, the NPO can also be used for general state estimation based on observer characteristics. If the model is adapted, it can be used in other types of processes. The filtering effect can also be used for station-keeping and dynamic positioning of marine structures such as ships.

The nonlinearity of the observer is introduced by the rotation of the matrices and due to the avoidance of linearization in the algorithm. Passivity theory ensures the convergence of the observer. Only one set of observer gains is needed to cover the entire state space which significantly simplifies the tuning parameters. Passivity also implies that the phase of the error dynamics is limited by 90 degrees, which gives excellent stability properties (Fossen 2021). For the NPO, passivity is also used as a tool for determining the filter cutoff frequency; see page 325 in (Fossen 2021) for more.

2.4 Passivity

Passivity refers to the ability to *not* be able to produce energy. A passive device is therefore only able to dissipate energy. For an input-output system, the system will be passive if an increase in the output requires an increase in the input. Most physical systems are passive. Passivity can be compromised by phase lags from sensors, actuators, and communication delays.

A linear system is passive if all input-to-output trajectories satisfy the following inequality for a given amount of mechanical energy represented by an arbitrary scalar function, $V(t)$ at time t with a continuous first derivative:

$$V(t) \leq V(0) + \int_0^T \boldsymbol{y}^\top(\tau)\boldsymbol{u}(\tau)\, d\tau \text{ for all } T > 0 \tag{2.11}$$

$V(0)$ represents the system at rest, and the integral represents the external energy input, $\boldsymbol{y}^\top\boldsymbol{u}$. The integral will represent the energy going into the system; hence the system will be passive for as long as it only consumes energy. This can also be related to the rate of change for the energy in the system by this inequality(Fossen 1994):

$$\boldsymbol{u}^\top(t)\boldsymbol{y}(t) \geq \dot{V}(t) \tag{2.12}$$

Alternatively, one can state that the system...

- is a passive mapping if $u^\top y \geq 0$, implying a lower bound for, $V(t)$.
- will have no energy dissipation, hence be lossless if $u^\top y = 0$.
- is input strictly passive if $u^\top y \geq \alpha \cdot u^\top + \beta > 0, \forall u \neq 0$. Where $\alpha > 0$ and β is a constant.
- is output strictly passive if $u^\top y \geq \alpha \cdot y^\top + \beta > 0, \forall y \neq 0$.

From this, we can also say that the system is *intrinsically stable.* In the frequency domain, we relate to passivity using the positive real condition, and for SISO-systems; $Re(G(j\omega)) > 0$ at all frequencies.

Relating this to the properties of control systems, one can say that

- The inverse of a passive system is passive

- The parallel interconnection of passive systems is passive
- The feedback interconnection of passive systems is passive

These properties will be useful later and show that, for a system with uncertain or variable characteristics, it is desirable to use the passive feedback law to ensure closed-loop stability (Mathworks n.d.).

2.4.0.1 The Passivity Theorem

Passivity can be used to demonstrate the stability of a plant, and the passivity theorem states that the feedback connection of two passive subsystems is passive. Applying more assumptions, the same plant can also be shown to have an asymptotically stable equilibrium at its origin and that the input-output mapping is finite-gain $\mathcal{L}_2$ stable. Passivity further yields that the phase of a positive real transfer function cannot exceed 90, For a system with one transfer function that is strictly positive real, the upper limit of the overall loop phase will be equal to 180 degrees.

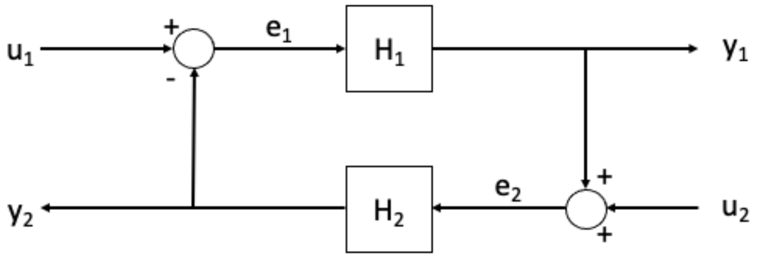


Figure 2.3: Example of a negative feedback system

Figure 2.3 presents a negative feedback system consisting of the two transfer functions, $\mathcal{H}_1$ and $\mathcal{H}_2$. Stability of the loop can be defined using either the Nyquist criterion, requiring a loop phase less than 180 degrees or the small-gain theorem(see Appendix A: Section A.8 for definition). Passivity applies to the same system if the feedback connection of the two systems, $\mathcal{H}_1$ and $\mathcal{H}_2$, is passive; the overall system will also be passive (Khalil 2015b).

From the passivity argument of the whole loop one can further state that the origin is asymptotically stable when u is zero and the following conditions are met:

- $\mathcal{H}_1$ and $\mathcal{H}_2$ are strictly passive (the energy absorbed by the system is equal to the change in stored energy when x(t) = 0)
- $\mathcal{H}_1$ and $\mathcal{H}_2$ are strictly passive output and zero-state observable (no solutions can stay in a point other than the zero solution, x(t) = 0).
- One of $\mathcal{H}_1$ or $\mathcal{H}_2$ is strictly passive and the other is output strictly passive and zero-state observable. A combination of the two conditions above

If the storage functions V for both components are radially unbounded, the origin will be *globally* asymptotically stable (Khalil 2015b).

2.5 Stability

The stability of a linear system is a property of its equilibrium point. An equilibrium of a dynamical system is stable if all solutions starting at nearby points stay nearby; otherwise, it is unstable. Instability occurs when an initial state of the system produces an unbounded response. Asymptotic stability adds to the definition of stability the notion that the solution also approaches the equilibrium as time approaches infinity. This section leaves out stability theory for time-invariant systems and focuses mainly on two relevant terms. The theory in this section is based upon definitions from the book "*Nonlinear Control*" by Hassan K. Khalil (Khalil 2015b).

Input-output stable (IO): For some systems, it is convenient to treat the dynamics as a black box and only consider their input and output, this is especially useful for systems with time delays or other elements which are difficult to physically model or represent in state-space. When describing such systems, it is often more convenient to use norms.

The norm of a signal, $||u||$ is used to measure and describe the magnitude of a signal. A norm should satisfy the following properties(Khalil 2015b):

- The norm of a signal is zero iff the signal is identically zero and is strictly positive otherwise
- Scaling a signal results in a corresponding scaling of the norm. $||au|| = |a| \cdot ||u||$.
- The norm satisfies the triangle inequality, which states that the norm of a sum is less than or equal to the sum of the two norms; $||u_1 + u_2|| \leq ||u_1|| + ||u_2||$.

Norms can be defined in terms of larger spaces of continuous functions, such as $\mathcal{L}_p^m$. The subscript p is used to define the type of p-norm and its space, generally, a value given as $1 < p < \infty$. For $p = 2$, the space will represent the inner product space of the norm, $u^\top u$. The superscript m denotes the dimension of the norm signal u.

$\mathcal{L}_p^m$ consists of all measurable functions $u : R \rightarrow R^m$ that satisfies:

$$||u||_{\mathcal{L}_p} = \left(\int_0^\infty ||u(t)||^p \, dt \right)^{1/p} < \infty \tag{2.13}$$

Using this definition, input-output stability can then be defined by the general definition of $\mathcal{L}$ stability, which is given by introducing a mapping operator, H that specifies y in terms of u.

The mapping H, is $\mathcal{L}$ stable if there exists a gain function, g defined for all positive x values and a non-negative constant β such that the following inequality is satisfied:

$$||(Hu)_\tau||_\mathcal{L} \leq g(||u_\tau||_\mathcal{L}) + \beta \tag{2.14}$$

for all values of u belonging to a $\mathcal{L}$ space and a τ defined for positive values of x. It can further be said to be finite-gain $\mathcal{L}$ stable for two non-negative constants γ and β such that:

$$||(Hu)_\tau||_\mathcal{L} \leq \gamma(||u_\tau||_\mathcal{L}) + \beta \tag{2.15}$$

for the same conditions. β is the bias term for the system with a gain less than or equal to the value of γ. From definition 6.2 in (Khalil 2015b). If the system can be said to be $\mathcal{L}_\infty$ then for every bounded input, $u(t)$, the output $Hu(t)$ is also bounded.

By further studying stability theory, it becomes apparent that many of the properties are complementary, which will be useful when later defining Lyapunov stability for a nonlinear system.

Input-to-state stability (ISS): limits unbounded growth and instability for systems affected by both internal dynamics and external inputs. In order for a system to be ISS-stable, there should exist a class of functions such that for any given time and initial state, a bounded input will produce a solution that satisfies the following inequality:

$$||x(t)|| \leq \beta(||x(t_0)||, t - t_0) + \gamma(\sup_{t_0 \leq \tau \leq t}||u(\tau)||), \forall t \geq t_0 \tag{2.16}$$

This inequality guarantees that the state will be bounded for any bounded input. In order for the system response, $x(t)$ to be smaller than the function β and such that the function γ, defined for an upper bounded input, u two new requirements are introduced. The function β should be of class $\mathcal{KL}$ and γ should be of class $\mathcal{K}$.

- A function belongs to the $\mathcal{K}$-class if it is a scalar continuous function that is strictly increasing and defined as 0 in the origin. One can also define a class $\mathcal{K}$ as a gain function for use in control design.
- A continuous scalar function $\beta(r, s)$ belongs to the $\mathcal{KL}$ class if the two dependent variables have the following relation. For a fixed s, the function maps to the class $\mathcal{K}$ when r varies. For a fixed r, the function decreases with respect to s so that $\beta(r, s) \to 0$ is as $s \to \infty$.

This follows the definition of time-varying systems by (Khalil 2015b). These two classes of functions can also be used to define the other types of stability for time-varying systems. Input-to-state stability will make the origin of an unforced system globally uniformly asymptotically stable. The Lyapunov theorem leads to input-to-state stability in a system, as stated in (Khalil 2015b).

It can be challenging to understand how the stability classifications listed above apply to real-world data. However, there are techniques that enable graphical representation and assessment of a system's stability. The Nyquist criterion for stability can be used for this, explained in Section A.6.

2.5.1 Lyapunov theory

The Lyapunov theory is an extended combination of input-output stability and passivity. The theory is applicable to both autonomous and non-autonomous systems. The main advantage of Lyapunov is that it enables the study of stability without having to solve the state equations, which is very useful for the stability assessment of nonlinear systems.

Lyapunov theory is applied by defining a Lyapunov function, which usually represents the energy of the system. This function can also be used for plotting the trajectories of the systems, which enables visual interpretation of the theory.

There are two methods to solve a Lyapunov equation, a direct method meant for time-independent autonomous systems and an indirect method to cover the general case (Fossen 1994). The Lyapunov direct method for autonomous systems is defined given a local Lipschitz function $\dot{x} = f(x)$, and a scalar Lyapunov function $V(x)$ defined for the region Ω which also satisfies the following conditions:

1. $V(x) = 0$ when $x = x_{eq}$
2. $V(x) > 0$ for all x in Ω except $x = x_{eq}$
3. $\nabla V(x) \cdot f(x) \leq 0$ for all x in $\Omega \implies x_{eq}$ is stable
4. If also: $\dot{V}(x) < 0$ for all x in Ω (except x_{eq}) $\implies x_{eq}$ is Locally Asymptotically Stable (LAS)
5. If also: $\Omega = R^n$ (whole state space) and $V(x)$ is radially unbounded, $||x|| \to \infty \implies V(x) \to \infty$, then x_{eq} is Gloablly Asymptotically Stable (GAS)

Condition 1 and 2 imply that the Lyapunov function is positive definite and condition 3 states that the derivative is negative definite.

For a more intuitive and visual representation of the theorem, one can use Lyapunov surfaces to describe the stability. An example of such surfaces is shown for an arbitrary Lyapunov function in figure 2.4.

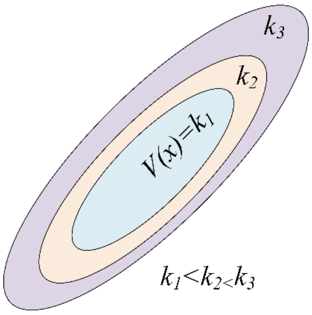


Figure 2.4: Level surfaces of a Lyapunov function (Hossain et al. 2017)

If a trajectory crosses a Lyapunov surface, given as $V(x) = k_i$ for $i = \{1, 2, 3\}$, it will be entrapped by the set belonging to the Lyapunov function according to the condition; $\dot{V} \leq 0$. For most cases, this surface will converge towards the origin, and in all cases, it is able to contain the origin within a ball (Khalil 2015b).

The convenience of this theorem is that it can be applied without having to solve the differential equation. The main drawback is that it can be difficult to find the Lyapunov functions as there is no systematic approach to deriving them (Khalil 2015b).

2.5.1.1 Lyapunov Stability Theorem for non-autonomous systems

The main difference when assessing the Lyapunov stability of a nonautonomous system instead of an autonomous system is that, in addition to the previously stated conditions for the Lyapunov function, it is now also demanded that the functions must be decrescent due to the added time dependence:

- $V(x,t) \leq V_0(x)\ \forall t \geq 0$ where $V_0(x) > 0$

If the function fulfills all these conditions, an equilibrium of the non-autonomous system satisfying $\boldsymbol{f}(\boldsymbol{x}^*, t) = \boldsymbol{0}$ will be globally asymptotically stable (GAS) (Fossen 1994).

2.6 Formulation of NPO

Based on the above-mentioned theory, the NPO can be derived considering the nonlinear low-frequency six-degree of freedom (6-DOF) body-fixed coupled equations model for a marine craft as defined by Fossen (Chapter 2, Fossen 2021)

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.17a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.17b}$$

where both $\boldsymbol{\eta}$, $\boldsymbol{\nu}$ and their relationship have been defined in Section A.3. $\boldsymbol{M}$ is the rigid-body mass matrix which also includes the added mass of the structure. $\boldsymbol{C}$ is the added mass, rigid-body Coriolis, and centripetal matrix caused by rotation of the body position with respect to the inertial frame. $\boldsymbol{D}$ is the damping matrix of the system. $\boldsymbol{g(\eta)}$ and $\boldsymbol{g}_0$ are to the hydrostatic forces and restoring forces acting on the body. The terms on the right side of the equation are the control inputs, $\boldsymbol{\tau}$ and environmental forces from wind, $\boldsymbol{\tau}_{wind}$ and waves, $\boldsymbol{\tau}_{wave}$.

2.6.1 System model

From the system equations, a model can be derived by applying the following assumptions:

- Structural symmetries for port and starboard of the object: $\boldsymbol{M} = \boldsymbol{M}^\top > \mathbf{0}$ and $\boldsymbol{D} > \mathbf{0}$, where the added mass terms are independent of wave frequency such that $\dot{\boldsymbol{M}} = \mathbf{0}$. A stationary mass matrix is valid due to the fact that motions are assumed small and therefore not frequency-dependent.
- Zero-mean Gaussian white noise driven bias $\mathbf{w} = \mathbf{0}$ and measurements $\mathbf{v} = \mathbf{0}$.
- Small wave-induced yaw disturbance yields, less than 5 degrees: $\mathbf{R}(\psi(t)) = \mathbf{R}(\psi) = \mathbf{R}(y_3)$, $y_3 = \psi + \psi_w \approx \psi$

This yields a simplified 3-DOF system model which can be written with respect to an earth-fixed frame using the following equations.

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}_w \boldsymbol{\xi} \tag{2.18a}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.18b}$$

$$\dot{\boldsymbol{b}} = -\boldsymbol{T}^{-1}\boldsymbol{b} \tag{2.18c}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} = -\boldsymbol{D}\boldsymbol{\nu} + \boldsymbol{R}^\top(\psi)\boldsymbol{b} + \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} \tag{2.18d}$$

$$\boldsymbol{y} = \boldsymbol{\eta} + \boldsymbol{\eta}_w = \boldsymbol{\eta} + \boldsymbol{C}_w \boldsymbol{\xi} \tag{2.18e}$$

In the above equations, the notation, $\boldsymbol{\xi}$ is introduced. This is a state vector containing perturbations with respect to the equilibrium state of the system such that $\boldsymbol{\eta}_w = \boldsymbol{C}_w \boldsymbol{\xi}$ models the wave-frequent response. $\boldsymbol{b}$ is the bias term, and $\boldsymbol{T}$ is a matrix that holds the positive bias time constants. Alternately, equations 2.18a, 2.18b and 2.18e can be represented using the state-space form:

$$\dot{\boldsymbol{\eta}}_0 = \boldsymbol{A}_0\boldsymbol{\eta}_0 + \boldsymbol{B}_0\boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.19a}$$

$$\boldsymbol{y} = \boldsymbol{C}_0\boldsymbol{\eta}_0 \tag{2.19b}$$

where the $\boldsymbol{A}_0$ matrix holds the value for $\boldsymbol{A}_w$ and the $\boldsymbol{C}_0$ matrix holds $\boldsymbol{C}_w$ (Fossen 2021). The values of $\boldsymbol{A}_w$ and $\boldsymbol{C}_w$ are selected based on resonance frequencies, damping ratios, and the peak period of the sea state (T. Sauder 2021). $\boldsymbol{y}$ is the measured position and the position vector, $\boldsymbol{\eta}_0$ is defined as

$$\hat{\boldsymbol{\eta}}_0 = \begin{bmatrix} \hat{\boldsymbol{\xi}}^\top \\ \hat{\boldsymbol{\eta}}^\top \end{bmatrix} \tag{2.20}$$

2.6.2 Observer equations

The observer equations are designed to copy the dynamics of the system and result in 15 ordinary differential equations.

$$\dot{\hat{\boldsymbol{\xi}}} = \boldsymbol{A}_w\hat{\boldsymbol{\xi}} + \boldsymbol{K}_1(\omega_0)\tilde{\boldsymbol{y}} \tag{2.21a}$$

$$\dot{\hat{\boldsymbol{\eta}}} = \boldsymbol{R}(\psi)\hat{\boldsymbol{\nu}} + \boldsymbol{K}_2\tilde{\boldsymbol{y}} \tag{2.21b}$$

$$\dot{\hat{\boldsymbol{b}}} = -\boldsymbol{T}^{-1}\hat{\boldsymbol{b}} + \boldsymbol{K}_3\tilde{\boldsymbol{y}} \tag{2.21c}$$

$$\boldsymbol{M}\dot{\hat{\boldsymbol{\nu}}} = -\boldsymbol{D}\hat{\boldsymbol{\nu}} + \boldsymbol{R}^\top(\psi)\hat{\boldsymbol{b}} + \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{R}^\top(\psi)\boldsymbol{K}_4\tilde{\boldsymbol{y}} \tag{2.21d}$$

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\eta}} + \boldsymbol{C}_w\hat{\boldsymbol{\xi}} \tag{2.21e}$$

Where $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ is the estimation error. Note that some literature uses another notation for $\boldsymbol{K}_3$ and $\boldsymbol{K}_4$ to put emphasis on its relation to the Lyapunov analysis. $\boldsymbol{K}_3 = \frac{1}{\gamma}\boldsymbol{\Lambda}$ and $\boldsymbol{K}_4 = \frac{1}{\gamma}\boldsymbol{\mathcal{K}}$. γ is a gain scaling parameter and $\boldsymbol{\Lambda}$, $\boldsymbol{\mathcal{K}}$ are gain matrices (Fossen 1998).

Alternately, the equations listed above can be expressed in a state-space form:

$$\dot{\hat{\boldsymbol{\eta}}}_0 = \boldsymbol{A}_0\hat{\boldsymbol{\eta}}_0 + \boldsymbol{B}_0\boldsymbol{R}(\psi)\hat{\boldsymbol{\nu}} + \boldsymbol{K}_0(\boldsymbol{\omega}_0)\tilde{\boldsymbol{y}} \tag{2.22a}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}_0\hat{\boldsymbol{\eta}}_0 \tag{2.22b}$$

In this case, the gain vector is expressed as:

$$\boldsymbol{K}_0(\boldsymbol{\omega}_0) = \begin{bmatrix} \boldsymbol{K}_1(\omega_0) \\ \boldsymbol{K}_2 \end{bmatrix} \tag{2.23}$$

The system is represented in a block diagram in Figure 2.5.

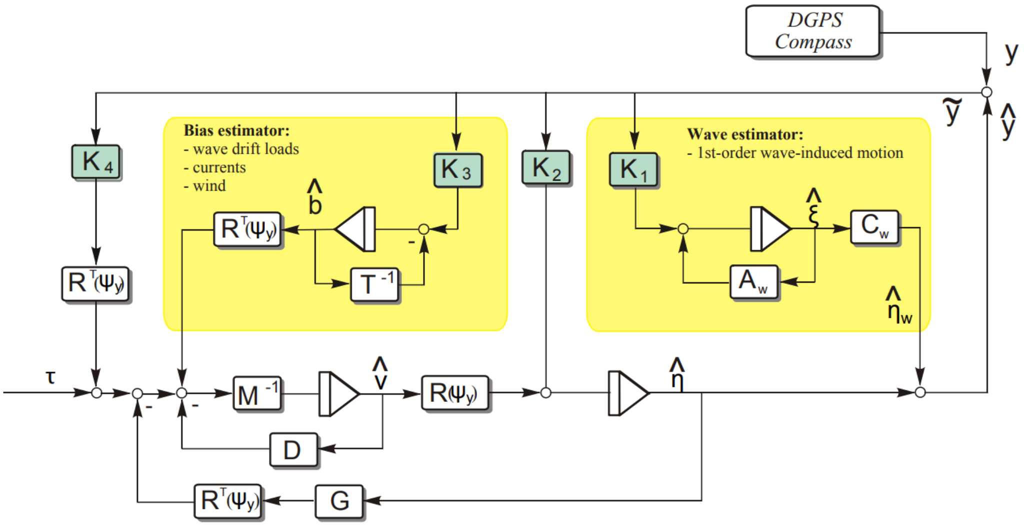


Figure 2.5: Block diagram of the NPO (Fossen 2021).

2.6.2.1 Observer estimation errors

Further, the error dynamics can be defined by establishing the following estimation errors:

$$Velocity : \tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \hat{\boldsymbol{\nu}} \tag{2.24a}$$

$$Bias : \tilde{\boldsymbol{b}} = \boldsymbol{b} - \hat{\boldsymbol{b}} \tag{2.24b}$$

$$Position : \tilde{\boldsymbol{\eta}}_{\boldsymbol{0}} = \boldsymbol{\eta}_{\boldsymbol{0}} - \hat{\boldsymbol{\eta}}_{\boldsymbol{0}} \tag{2.24c}$$

Using these relations, some of the observer equations can be restated as follows:

$$\dot{\tilde{\boldsymbol{\eta}}}_0 = [\boldsymbol{A}_0 - \boldsymbol{K}_0(\omega_0)\boldsymbol{C}_0]\tilde{\boldsymbol{\eta}}_0 + \boldsymbol{B}_0\boldsymbol{R}(\psi)\tilde{\boldsymbol{\nu}} \tag{2.25a}$$

$$\dot{\tilde{\boldsymbol{b}}} = -\boldsymbol{T}^{-1}\tilde{\boldsymbol{b}} - \boldsymbol{K}_3\tilde{\boldsymbol{y}} \tag{2.25b}$$

The dynamics of the velocity estimation error:

$$\boldsymbol{M}\dot{\tilde{\boldsymbol{\nu}}} = -\boldsymbol{D}\tilde{\boldsymbol{\nu}} + \boldsymbol{R}^{\top}(\psi)\tilde{\boldsymbol{b}} - \boldsymbol{R}^{\top}(\psi)\boldsymbol{K}_4\tilde{\boldsymbol{y}} \tag{2.26}$$

By further defining a variable, $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{\eta}}_0\ ,\ \tilde{\boldsymbol{b}}]^\top$ and with: $\tilde{\boldsymbol{z}} = \boldsymbol{K}_4\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}$ the error dynamics can be restated as follows:

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}\tilde{x} + \boldsymbol{B}\boldsymbol{R}(\psi)\tilde{\boldsymbol{\nu}} \tag{2.27a}$$

$$\tilde{\boldsymbol{z}} = \boldsymbol{C}\tilde{\boldsymbol{x}} \tag{2.27b}$$

Using the terms above, two new error terms can be introduced as $\boldsymbol{\epsilon}_z = -\boldsymbol{R}^\top(t)\tilde{\boldsymbol{z}}$ and $\boldsymbol{\epsilon}_z = \boldsymbol{R}(t)\tilde{\boldsymbol{\nu}}$. Consult chapter 11 in Fossen 2021 for the formulation of the matrices. The advantage of this formulation is that one can now rewrite the block diagram in figure 2.5 which allows a more intuitive understanding of the system. This is first done by rescheduling and grouping the block diagram as follows in figure 2.6.

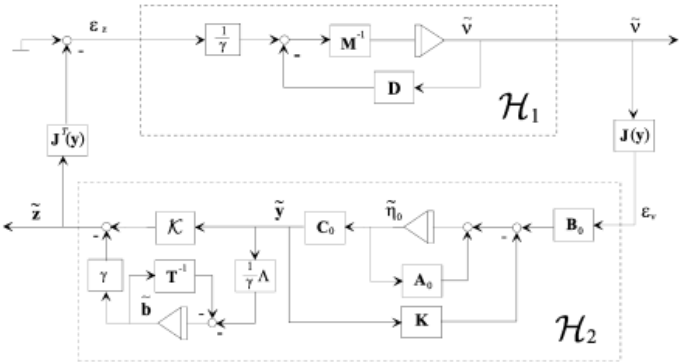


Figure 2.6: Block diagram of error dynamics to NPO(Fig.4 in Fossen 1998)

which is further simplified in the following figure 2.7:

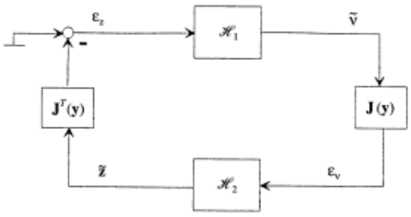


Figure 2.7: Passive block diagram configuration of the NPO(Fig. 7 in Fossen 1998)

Introducing linear relations and a new state vector enables another formulation of the observer error system consisting of two linear blocks, $\mathcal{H}_1$ and $\mathcal{H}_2$ (Fossen 2021).

$$\mathcal{H}_1 : \boldsymbol{M}\dot{\tilde{\boldsymbol{\nu}}} = -\boldsymbol{D}\tilde{\boldsymbol{\nu}} + \boldsymbol{\epsilon_z} \tag{2.28a}$$

$$\mathcal{H}_{21} : \dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{\epsilon_\nu} \tag{2.28b}$$

$$\mathcal{H}_{22} : \dot{\tilde{\boldsymbol{z}}} = \boldsymbol{C}\tilde{\boldsymbol{x}} \tag{2.28c}$$

Given appropriate terms of $\mathcal{H}_1$ and $\mathcal{H}_2$, one can establish that the observer fulfills requirements for both passivity and stability. This will be explained in more detail in the following sections. The derivations of the above equations were all derived by Thor I. Fossen in his papers: (Fossen 2021, Fossen 1998 and Fossen 1994).

2.6.3 Convergence of the NPO

The word convergence in the context of control system theory relates to stability analysis and the fact that for a dynamical system it is desired that the error *converges* to zero. The stability theory derived earlier in the report will now be applied to the system model to prove the convergence of the observer.

The first step is to show that the velocity error dynamics given by $\mathcal{H}_1$ in Equation 2.28 is strictly passive, this can be done by defining a positive definite storage function, S. The physical explanation of this is that energy is dissipated due to hydrodynamic damping.

$$S = \frac{1}{2}\tilde{\boldsymbol{\nu}}^\top \boldsymbol{M}\tilde{\boldsymbol{\nu}} \tag{2.29}$$

Differentiating this storage function with respect to $\tilde{\boldsymbol{\nu}}$ and substituting for the error signal, $\boldsymbol{\epsilon}_z$ yields:

$$\boldsymbol{\epsilon}_z^\top \tilde{\boldsymbol{\nu}} = \dot{S} + \frac{1}{2}\tilde{\boldsymbol{\nu}}^\top(\boldsymbol{D} + \boldsymbol{D}^\top)\tilde{\boldsymbol{\nu}} \tag{2.30}$$

This equation states that the block $\mathcal{H}_1$ is strictly passive for the mapping from $\boldsymbol{\epsilon}_z$ to $\tilde{\boldsymbol{\nu}}$. In addition, the block $\mathcal{H}_2$ can be shown to be passive by applying the Kalman-Yakubovich-Popov (KYP) lemma. This lemma states that the square transfer function matrix, $\boldsymbol{Z}(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}$ is strictly positive real, given the following equations.

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{P} = -\boldsymbol{Q} \tag{2.31}$$

$$\boldsymbol{B}^\top \boldsymbol{P} = \boldsymbol{C} \tag{2.32}$$

These relations will hold under the assumption that the $\boldsymbol{A}$ matrix is Hurwitz i.e. the eigenvalues should all have negative real parts. In addition to the pair of matrices $(\boldsymbol{A}, \boldsymbol{B})$ being controllable and $(\boldsymbol{A}, \boldsymbol{C})$ being observable. Then the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ should be defined such that they are positive definite, satisfying the relations:

$$\boldsymbol{P} = \boldsymbol{P}^{\top} \tag{2.33a}$$

$$\boldsymbol{Q} = \boldsymbol{Q}^{\top} \tag{2.33b}$$

This will ensure that the transfer function $\mathcal{H}_2$ and its mapping, $\boldsymbol{\epsilon}_{\nu} \rightarrow \tilde{\boldsymbol{z}}$ is passive given that the set of matrices ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) can be defined as strictly positive real (SPR).

2.6.3.1 Observer stability properties

After having established that block $\mathcal{H}_1$ can be strictly passive and $\mathcal{H}_2$ can be made passive, it is now time to establish that the plant is stable. This is done by defining the Lyapunov function of the observer as follows:

$$V_{obs} = \tilde{\boldsymbol{\nu}}^{\top} \boldsymbol{M} \tilde{\boldsymbol{\nu}} + \tilde{\boldsymbol{x}}_0^{\top} \boldsymbol{P} \tilde{\boldsymbol{x}}_0 > 0, \forall \tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{x}}_0 \neq \mathbf{0} \tag{2.34}$$

which is a passive function for the positive constant matrix, $\boldsymbol{P} = \boldsymbol{P}^{\top} > \mathbf{0}$. Applying a time differentiation of the Lyapunov candidate along the trajectories of $\tilde{\boldsymbol{\nu}}$ and $\tilde{\boldsymbol{x}_0}$ yields the following:

$$\begin{aligned} \dot{V}_{obs} = -\tilde{\boldsymbol{\nu}}^{\top}(\boldsymbol{D}^{\top} + \boldsymbol{D})\tilde{\boldsymbol{\nu}} + \tilde{\boldsymbol{x}}_0^{\top}(\boldsymbol{P}\boldsymbol{A}_0 + \boldsymbol{A}_0^{\top}\boldsymbol{P})\tilde{\boldsymbol{x}}_0, \\ + 2\tilde{\boldsymbol{\nu}}^{\top}\boldsymbol{R}^{\top}(\psi)\boldsymbol{B}_0^{\top}\boldsymbol{P}\tilde{\boldsymbol{x}}_0 - 2\tilde{\boldsymbol{\nu}}^{\top}\boldsymbol{R}^{\top}(\psi)\tilde{\boldsymbol{z}}_0 \end{aligned} \tag{2.35}$$

Defining the variable, $\boldsymbol{Q}_1$ as:

$$\boldsymbol{D}^{\top} + \boldsymbol{D} = \boldsymbol{Q}_1 > \mathbf{0} \tag{2.36}$$

And demanding that P satisfies the Lyapunov equation:

$$\boldsymbol{P}\boldsymbol{A}_0 + \boldsymbol{A}_0^{\top}\boldsymbol{P} = -\boldsymbol{Q}_2 < \mathbf{0} \tag{2.37}$$

Where both matrices, $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ are positive definite. The two last terms in the equation of the derivative (2.35) will be canceled by stating that a new constant, $\boldsymbol{C}_0$ is given as:

$$\boldsymbol{C}_0 = \boldsymbol{B}_0^{\top}\boldsymbol{P} \tag{2.38}$$

These definitions result in a restatement of the derivative:

$$\dot{V}_{obs} = -\tilde{\boldsymbol{\nu}}^{\top}\boldsymbol{Q}_1\tilde{\boldsymbol{\nu}} + \tilde{\boldsymbol{x}}_0^{\top}\boldsymbol{Q}_2 < 0, \forall \tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{x}}_0 \neq \mathbf{0} \tag{2.39}$$

which satisfies all assumptions made for a function to be Lyapunov stable and both $\tilde{\boldsymbol{\nu}}$ and $\tilde{\boldsymbol{x}}_0 = [\tilde{\boldsymbol{\xi}}^{\top}, \tilde{\boldsymbol{\nu}}^{\top}, \tilde{\boldsymbol{b}}^{\top}]^{\top}$ will converge exponentially towards zero, making the function Globally Exponentially Stable (GES) (Marius F. Aarset 1998).

2.7 Involved tuning parameters and their meaning

Previously in this report, the blocks $\mathcal{H}_1$ and $\mathcal{H}_2$ have been defined using the KYP lemma and choosing a storage function that solves the Lyapunov equations. In order for these blocks to satisfy the established assumptions and ensure passivity as well as stability for the observer, proper gains have to be chosen. The full demonstration and derivation of decoupled transfer functions are found in Fossen (2021). This thesis focuses on the main results of this derivation.

The observer gain matrices are defined in such a way that the error mapping is strictly positive real (SPR). Fossen suggests the following gains based on the decoupled system in surge, sway and yaw implying that the gain matrices should have a diagonal structure:

$$\boldsymbol{K}_1(\omega_0) = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \\ k_4 & 0 & 0 \\ 0 & k_5 & 0 \\ 0 & 0 & k_6 \end{bmatrix} \tag{2.40}$$

with the corresponding values for the coefficients:

$$k_{1,2,3} = -2(\zeta_{ni} - \zeta_i)\frac{\omega_{ci}}{\omega_i} \tag{2.41a}$$

$$k_{4,5,6} = 2\omega_i(\zeta_{ni} - \zeta_i) \tag{2.41b}$$

As seen from the equations above, $\boldsymbol{K}_1$ is dependent on a cutoff frequency, ω_{ci} and a damping value, ζ_i. This gain controls the wave-estimator part of the observer and produces the estimate of the wave position, η_w. Choosing the wrong values of these two will cause the observer to be unable to separate the wave frequency from the rest of the signal.

$$\boldsymbol{K}_2 = \begin{bmatrix} k_7 & 0 & 0 \\ 0 & k_8 & 0 \\ 0 & 0 & k_9 \end{bmatrix} \tag{2.42}$$

$$k_{7,8,9} = \omega_{ci}$$

From the equation above, one can see that $\boldsymbol{K}_2$ also relates to the filter properties of the observer, as ω_{ci} denotes the cutoff frequency. K_2 is also referred to as “position injection” indicating that adjusting this parameter will affect the position estimates. This can also be seen from Figure 2.5 which also shows the relation between $\boldsymbol{K}_3$ and the bias estimate in the observer.

$$\boldsymbol{K}_3 = 0.1 \cdot \boldsymbol{K}_4 \tag{2.43}$$

$$\boldsymbol{K}_4 = \mathrm{diag}(\boldsymbol{M}) = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \tag{2.44}$$

$\boldsymbol{K}_4$, also called “velocity injection” can be chosen based on the masses of the system and relates to the velocity estimates of the system. In all cases, increasing the gain puts more emphasis on the related characteristics of the system.

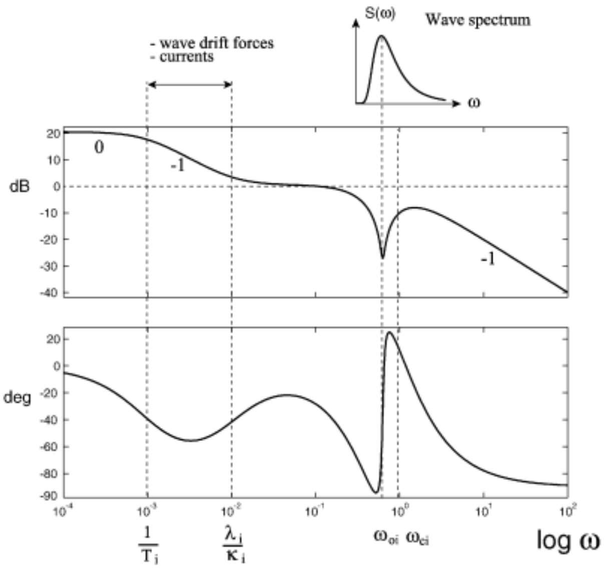


Fig. 6. Bode plot showing the transfer functions $h^i(s)$ when $1/T_i \ll \lambda_i/\kappa_i < \omega_{oi}\omega_{ci}$ (i = 1–3).

Figure 2.8: Bode plot of wave filtering(Fossen 1998)

Figure 2.8 shows a bode plot of the transfer functions for a decoupled marine system in surge with an incoming wave spectrum. The figure also displays the relative difference between the bias time constants, T_i, the relation $K3/K4 = \lambda_i/K_i$, the dominating wave frequency in the given wave spectrum, ω_{oi} and finally the cutoff frequency, ω_{ci}. Fossen (2021) states that if the relation;

$$\frac{1}{T_i} << \frac{K_{3i}}{K_{4i}} < \omega_{oi} < \omega_{ci}, i \in (1,3) \tag{2.45}$$

between the tuning parameters is fulfilled, the Kalman-Yakubovich-Popov (KYP) lemma and Strictly Positive Real (SPR) requirement will also be satisfied.

2.7.1 Bias modelling

The bias is caused by the slowly-varying environmental disturbances, i.e. second order wave drift loads, currents, wind, and unmodeled dynamics. These are, in turn, usually modeled as a Wiener process (Fossen 2021). For this application, the bias forces in surge, sway, and yaw are assumed to be slowly varying (Fossen 1998).

By using a simplified environmental model, wave frequency motions and bias states can be neglected (Fossen 1998). Marius F. Aarset (1998) further extends this to the general case by including a dynamic model for wave filtering and bias state estimation. The NPO will, contrary to a linearized or extended Kalman filter, guarantee that the bias state estimation will converge toward zero in addition to all estimation errors (Fossen 2021). The bias model from the NPO is given by:

$$\dot{\hat{\boldsymbol{b}}} = -\hat{\boldsymbol{T}}^{-1}\hat{\boldsymbol{b}} + \boldsymbol{K}_3\tilde{\boldsymbol{y}} \tag{2.46}$$

where the first term includes low-pass filtering for $\boldsymbol{T} > 0$ instead of just integrating the white noise term, $\boldsymbol{K}_3\tilde{\boldsymbol{y}}$. This ensures exponential stability, whereas the model without a low-pass term only will give asymptotic stability (Fossen 2021). In a DP system, an integral action is used to counteract the biased terms.

Chapter 3

Simulation-based testing and verification study

3.0.1 Verification studies

To ensure good estimates and correct calculations, modeling should be combined with experiments. The common practice when designing and testing complex systems is to use a model-based design approach or a v-model design. One key benefit of this method is its ability to enable abstraction, automation, collaboration, simulation, and analysis on multiple levels. The figure below tries to encapsulate the key elements of both methods.

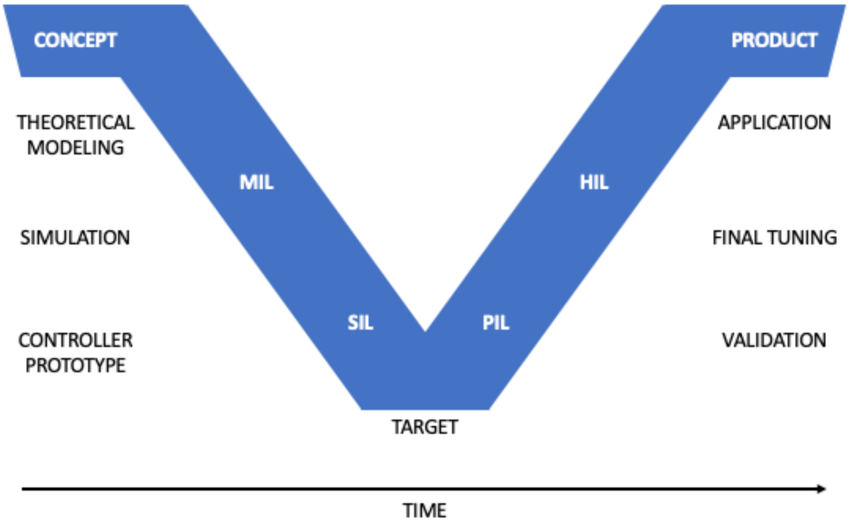


Figure 3.1: V-model for model-based design

In a step-wise approach to testing and verification, the process usually begins with the conception of a desired product; in this case an experimental setup of an offshore wind turbine. The development of this concept can be done using model-in-the-loop testing (MIL), where controller logic is tested on a simulated model of the plant. This was done in the preliminary project, where the NPO was tested and verified in SIMULINK using synthetic signals.

The next step is software-in-the-loop (SIL) testing, the main focus of this thesis. Controller logic is replaced by a controller block to verify hardware conversion, in this case, done by extracting the NPO as an FMU. This increases the level of fidelity compared to MIL. Further, the system behavior is evaluated in the simulated environment from SIMA while assessing functionality and performance without physical hardware.

Before converting to the use of hardware in the testing, a possible intermediate step can be taken, this is known as processor-in-the-loop (PIL), and can then be introduced to test whether the processor is capable of running the controller logic.

Hardware-in-the-loop (HIL) testing is conducted by creating a physical control system and running simulations with it. The HIL simulator operates in real-time and remains in closed-loop synchronization with the control computer system's hardware and software. Its purpose is to facilitate efficient and realistic testing of the control system's performance, functionality, and ability to handle failures (Skjetne and Egeland 2005). Communication routines, like UDP, can be utilized to verify communication channels and the I/O interface. Once the HIL testing has given satisfying results, the system is ready to be tested using only hardware components. For this thesis, the complete hardware test conducted in an ocean basin will also be the final product in Figure 3.1.

The level of process development can be tracked using a software-to-hardware (SW-to-HW) ratio. In this case, the SW-to-HW ratio represents the extreme where the system is a pure simulation study, in the SIL phase. This eliminates the requirement for the system to run in real-time, limiting the scope of operation. However since the whole process is simulated, it is important to consider simulator functionality, such as failure modes and accuracy (Skjetne and Egeland 2005). Another advantage of SIL comes from the absence of physical hardware components, making it more more accessible and allowing testing at an earlier stage in the design and development process of a system.

For further discussions, it is helpful to differentiate between the three terms validation, verification, and calibration. Validation refers to the study of whether the model at hand solves the correct equations for its intended purpose. Verification seeks to discover if the model is solving its respective equations correctly. Calibration is about finding the parameters that fit best in a theoretical model with a given structure. The rest of this thesis is focused on verification and calibration.

3.1 Software-in-the-loop testing and method

As already mentioned, SIL testing evaluates software systems without physical hardware. It simulates inputs and outputs, ensuring software performs as expected. This enables early testing and saves time and resources as it removes the risk of hardware damage. It detects defects, verifies behavior, and optimizes performance. A thorough understanding of system components and careful design of the simulation environment is crucial for SIL testing in hybrid systems.

In this project, the controller will be tested in simulation with data simulated from the virtual environment in SIMA (Simulation Workbench for Marine Applications). An overview of the configuration of the two different simulation software is displayed below. For the physical experimental setup, HLCC will send tension commands in model scale instead of winch angles to SIMA.

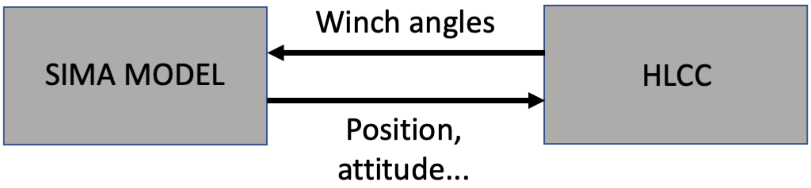


Figure 3.2: Software-in-the-loop system overview

3.2 SIMA

SIMA, developed by SINTEF, is a dynamic simulation platform for the modeling, optimization, and analysis of marine systems. SIMA imports files from hydrodynamical software such as WAMIT, RIFLEX, SIMO, and SESAM. RIFLEX is focused on load, response, fatigue, and probabilistic analysis of flexible elements and optimization design for risers in offshore oil and gas production. SIMO is a tool for the analysis of static and dynamic responses for offshore wind turbines, it is also useful for structural integrity and reliability tests.

In this project, SIMA is used to simulate the response of the substructure to the INO WINDMOOR 12MW wind turbine, showcased in Figure 3.3. The output from this simulation is sent to HLCC with the model of the NPO. As this testing is preliminary to physical experiments and intended as verification studies, the focus of the loading conditions will be around the intended conditions for physical tests.

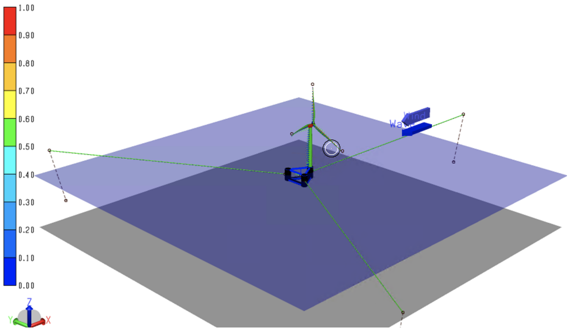


Figure 3.3: Overview of example model of INO12 MW SIMA

3.2.1 Simulation model description

Figure 3.3 is an example model of the INO WINDMOOR 12MW turbine that can be found in SIMA. The model consists of a turbine, tower, blades, and mooring system modeled using RIFLEX elements while the floater has been modeled in SIMO. Dynamic analysis of RIFLEX elements is often more resource-demanding, than SIMO elements and is likely to cause longer run time. To ensure that the simulation is both feasible and efficient, it was deemed necessary to simplify the model.

This thesis is mainly interested in the hydrodynamic response of the floating substructure and not the turret effects. Experimentation with the model and gradual removal of the dynamic RIFLEX components deemed faster simulations with fewer dynamic elements. Comparisons between the simplified and full model have been verified in decay tests and modifications to compensate for the removed tower and blade bending moments and stiffness were applied. This work was carried out by Vishnu as a part of his doctoral research. With little difference in dynamic response and significantly faster simulation times, it was decided to continue simulations with only the SIMO body of the floater. A close-up visualization and an overview of the full model are shown in figures 3.4 and 3.5.

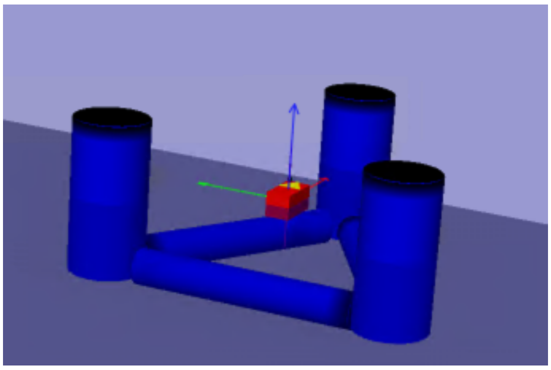


Figure 3.4: Detailed floater model in SIMA

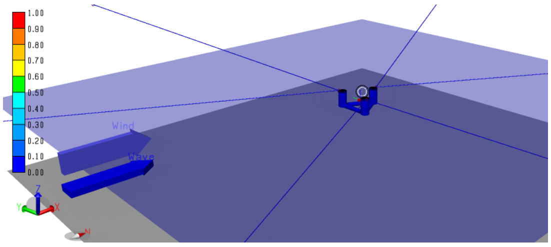


Figure 3.5: Full SIMO model overview with actuation lines

3.2.2 Deviations between project report and SIMA model

There are some differences between the project report describing the full model of INO WINDMOOR 12MW Souza (2021) and the model used for simulations in SIMA. The most noticeable are the masses of the substructure.

Souza (2021) presents a mass of $14.176 \cdot 10^6$ kg, whereas the model in SIMA has a mass of $13.99 \cdot 10^6$ kg. This gives a total difference of 186 kg. Not the biggest difference, but worth noting. Other relevant deviations are listed in the table below.

Unit	Souza 2021	SIMA	Difference	Percentage of report
Mass	$14.176 \cdot 10^6$	$13.99 \cdot 10^6$	$186 \cdot 10^3$	98.69%
Ixx	$2.7292 \cdot 10^{10}$	$2.7328 \cdot 10^{10}$	$-3.6 \cdot 10^7$	100.13%
Iyy	$2.7205 \cdot 10^{10}$	$2.7064 \cdot 10^{10}$	$141 \cdot 10^6$	99.48%
Izz	$1.2985 \cdot 10^{10}$	$1.2968 \cdot 10^{10}$	$17 \cdot 10^6$	99.87%

Table 3.1: Differences between values in Souza and SIMA

3.2.3 Linear and quadratic damping

An important difference with this experimental setup compared to the one in the offline simulations is the introduction of time-varying forces in the model. The offline simulations were conducted using sinusoidal synthetic inputs and without any quadratic damping. The introduction of this force is important as it is relatively dominant in terms of magnitude when compared to linear damping. Linear damping in the model comes from the linear damping matrix and viscous effects whereas the nonlinear and quadratic damping is caused by the terms from Morison's equation, more detailed descriptions and derivations of these terms are given in chapter A, Section A.2.

Degree of freedom	Linear damping coefficient	Unit
Surge, Dl_1	1442.2	$[Ns/m]$
Sway, Dl_2	1442.2	$[Ns/m]$
Yaw, Dl_6	$2.61 \cdot 10^6$	$[Nms/rad]$

Table 3.2: Linear damping matrix values from SIMA

Since the NPO is a model-based observer algorithm, it will be affected by the introduction of this new force. The result will be a different behavior as it will experience more forces that are not directly explained by the model. This will be observable as an increase in the bias force calculated by the observer. The first goal of the simulations is therefore to identify this effect on the results and then adjust the model with respect to the new input. This thesis neglects interactions between columns in the model and assumes that the force in surge and sway then will be equal.

3.2.4 Wave simulation

The main reason why SIMA is chosen as the simulation environment for this type of testing is its ability to generate realistic wave series and the following response of a body exposed to these waves. One goal of this report is to simulate the behavior of a model in an experimental setup before conducting physical experiments. It is therefore advantageous to be able to replicate the wave conditions which will be applied in the physical setup.

The generation of wave series in this thesis is done using a three-parameter JONSWAP (Joint North Sea Wave Project) spectrum in SIMA. The JONSWAP spectrum is selected based on its ability to provide a simple, yet effective description of the statistical properties of ocean waves. This is essential for predicting wave behavior and designing structures that can withstand the forces of waves. Listed below are the three parameters and their physical definition.

- Significant wave height (H_s): The average height of the one-third highest waves in a wave record.
- Peak wave period (T_p): The wave period at which the energy density is highest in the wave spectrum.
- Peak enhancement factor (γ): A dimensionless parameter that describes the ratio of the energy density at the peak frequency to the energy density at the spectral tail.

Unless stated otherwise, the simulations in this thesis have been conducted with a significant wave height of 6 meters, a peak period of 7.55 seconds, and an enhancement factor of 3.3. Figure 3.6 shows the resulting JONSWAP spectrum for these conditions.

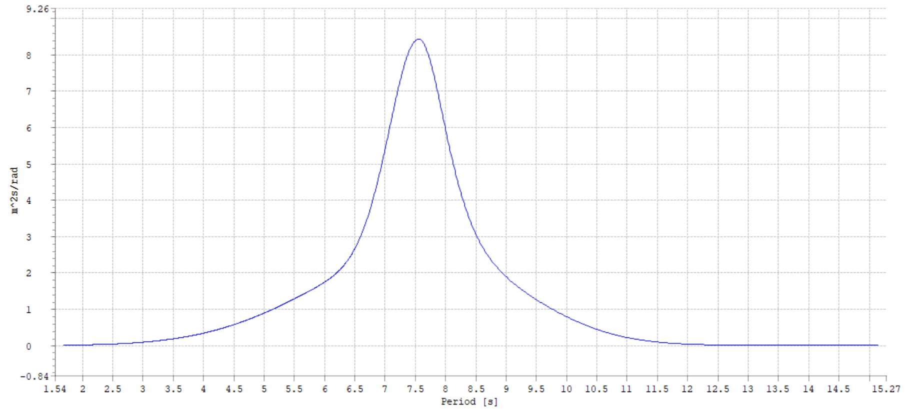


Figure 3.6: JONSWAP wave spectrum from SIMA for $H_s = 6$ and $T_p = 7.55$

When conducting physical tests on floating structures like this turbine it is common practice to categorize environmental loads for comparison of simulation scenarios with varying waves. This can be done by establishing an analogy to wave steepness of regular waves, which refers to the ratio between the height and length of the wave. In general, waves with higher steepness are more powerful and can cause more damage than waves with lower steepness. It also describes how much energy the wave contains and its behaviour as it propagates through the water.

We define steepness by the following relation:

$$\gamma = \frac{H_s}{T_p^2} \tag{3.1}$$

The initial simulation state has a steepness factor of 0.11. Additionally, seven wave conditions have been divided into two groups of constant steepness and will be used in simulation and experimentation. The parameters of the first group are given:

Condition [-]	**Significant wave height** (Hs) [m]	**Peak period** (Tp) [s]	**Steepness** (γ) [-]
1	3.75	7	0.08
2	6.19	9	0.08
3	11.00	12	0.08
4	14.97	14	0.08

Table 3.3: First set of sea-states with similar steepness

The second group:

Condition [-]	**Significant wave height** (Hs) [m]	**Peak period** (Tp) [s]	**Steepness** (γ) [-]
1	2.00	7	0.04
2	3.74	9	0.05
3	6.19	12	0.04

Table 3.4: Second set of sea-states with similar steepness

3.2.5 SIMA in SIL setup

Figure 3.7 illustrates the details of the “SIMA Model block” previously shown in in Figure 3.2. This figure shows that the SIMA model receives a command in terms of winch angles from HLCC and provides measurements of position and applied forces among others to HLCC. The signal received by the HLCC component is sent through a JAVA-control block which translates the signal into readable input for SIMA. The winch angles are sent to the modeled winches in SIMA, each responsible for actuating the desired force to the fixed elongation couplings and further onto the floater model.

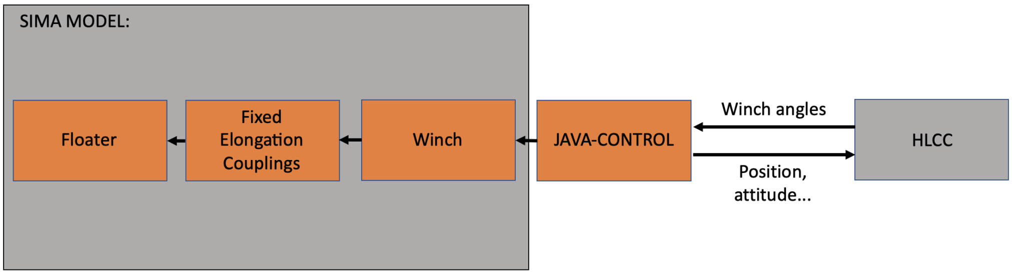


Figure 3.7: Detailed SIMA model for software-in-the-loop system

SIMA then measures the positional data, containing information on the position and orientation of the floater, and sends it as output through the JAVA-control to the HLCC software.

The winches in SIMA have been modeled using a “generic external controller system”. This system receives signals from the JAVA control and controls the winches by measuring the position of the floater and lines.

3.3 HLCC

HydroLab Centralized Control (HLCC) is a system designed for remote monitoring and control of various aspects in a hydrodynamic laboratory using a centralized controller. This typically includes real-time monitoring of experiments, data acquisition and analysis, instrument control, and system administration tasks. Having a single point of access for all users reduces the need for manual intervention, which can increase the efficiency and accuracy of experiments. The software builds on the National Instruments (NI) framework called LabVIEW (Laboratory Virtual Instrument Engineering Workbench), a graphical programming language widely used in measurement and automation applications. HLCC is the other module in Figure 3.2. The figure resembles Figure 1.4 and portrays the subroutines of HLCC which will be explained in further detail later.

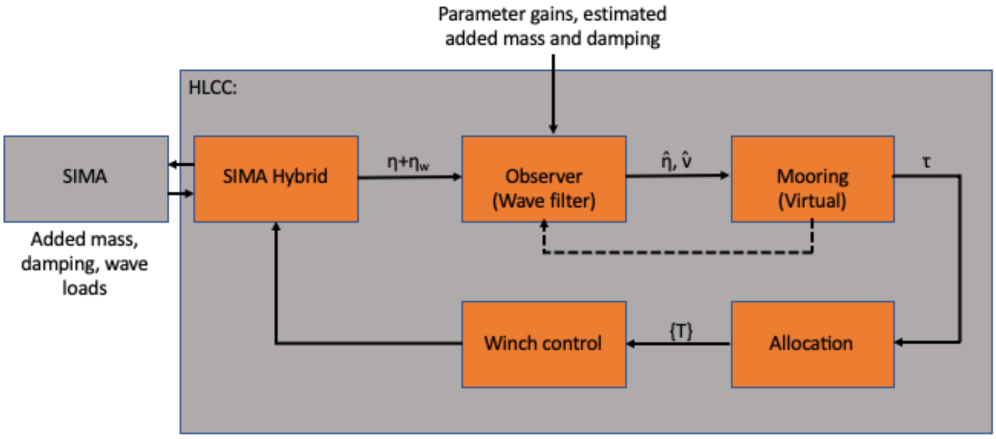


Figure 3.8: HLCC divided into FMUs

Modules for monitoring and controlling aspects of the experiments are loaded into HLCC using "Functional Mockup Units" (FMUs). This is a standardized format based on the "Functional Mock-up Interface" (FMI), supporting both Model Exchange and Co-Simulation (*Functional Mock-up Interface (FMI) Standard website* n.d.). The compatibility of FMUs enables them to be used in a wide range of applications and streamlines the process of model exchange and integration.

HLCC utilizes DLL subroutines to run the FMUs. DLL stands for "Dynamic Link Library". It is a type of file that contains code and data that can be used by multiple programs at the same time. When an application needs to use a DLL, it can dynamically load the DLL into its memory space at runtime. This allows the application to access the functions and data provided by the DLL, without having to include the entire code of the DLL within its own executable file. This makes the application more efficient and easier to maintain. Another advantage of DLLs is that they can be updated independently of the applications that use them. This means that improvements can be made to a DLL, without requiring the application to be updated or recompiled (*Microsoft Learn - Troubleshoot Windows Client: Dynamic Link Library* n.d.).

3.3.1 Using HLCC for simulations

When first launching the application, the user is asked to appoint a project number and what type of input is desired to use for the experiment. The options, in this case, include input as DLLs, a signal-stream enabling the reading of data from a .txt file as input, and SIMA-simulated signals by the use of SIMA-Hybrid. In Figure 3.9, all the relevant modules for this project are shown. Each DLL is denoted with DLL in addition to the name of the FMU which is loaded.

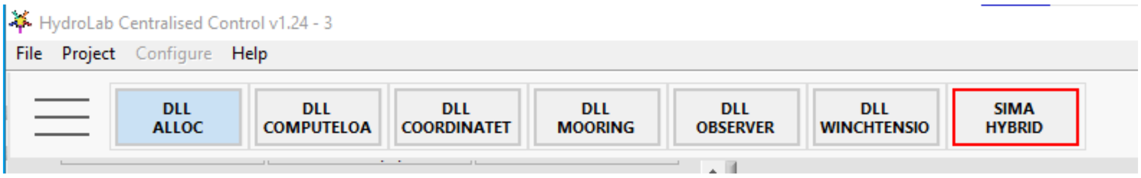
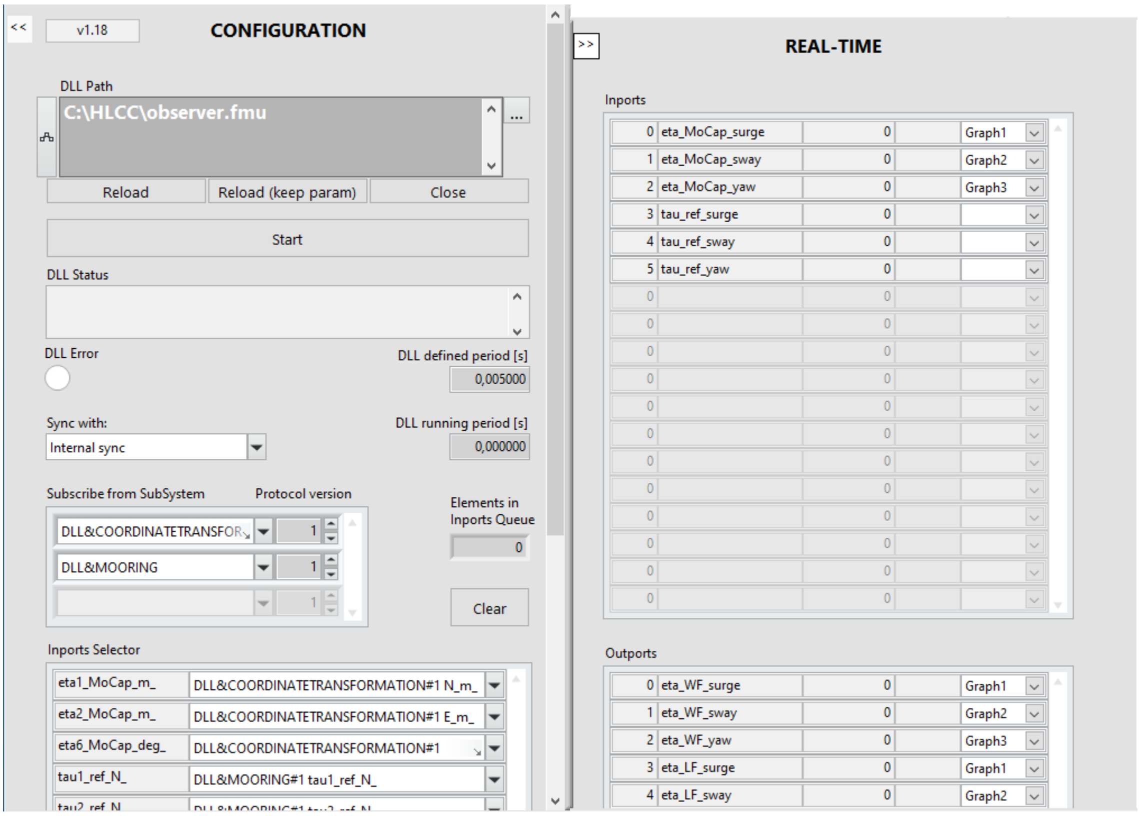


Figure 3.9: HLCC initialization

In the pictures below, the experimental setup in HLCC for this thesis is showcased. Figure 3.10a shows the configuration of the observer algorithm as a DLL in HLCC. The top field shows the path from which the file is imported, and below there are four buttons that make it possible to start the simulation and reload it with or without changes made to the input parameters. The close button removes the FMU from the DLL.



DLL Path: C:\HLCC\observer.fmu

DLL Status:

DLL Error: ○

DLL defined period [s]: 0,005000

Sync with: Internal sync

DLL running period [s]: 0,000000

Subscribe from SubSystem / Protocol version

Subscribe from SubSystem	Protocol version
DLL&COORDINATETRANSFOR	1
DLL&MOORING	1
	1

Elements in Inports Queue: 0

Inports Selector

eta1_MoCap_m_	DLL&COORDINATETRANSFORMATION#1 N_m_
eta2_MoCap_m_	DLL&COORDINATETRANSFORMATION#1 E_m_
eta6_MoCap_deg_	DLL&COORDINATETRANSFORMATION#1
tau1_ref_N_	DLL&MOORING#1 tau1_ref_N_
tau2_ref_N	DLL&MOORING#1 tau2_ref_N

Inports

0	eta_MoCap_surge	0	Graph1
1	eta_MoCap_sway	0	Graph2
2	eta_MoCap_yaw	0	Graph3
3	tau_ref_surge	0	
4	tau_ref_sway	0	
5	tau_ref_yaw	0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	
0		0	

Outports

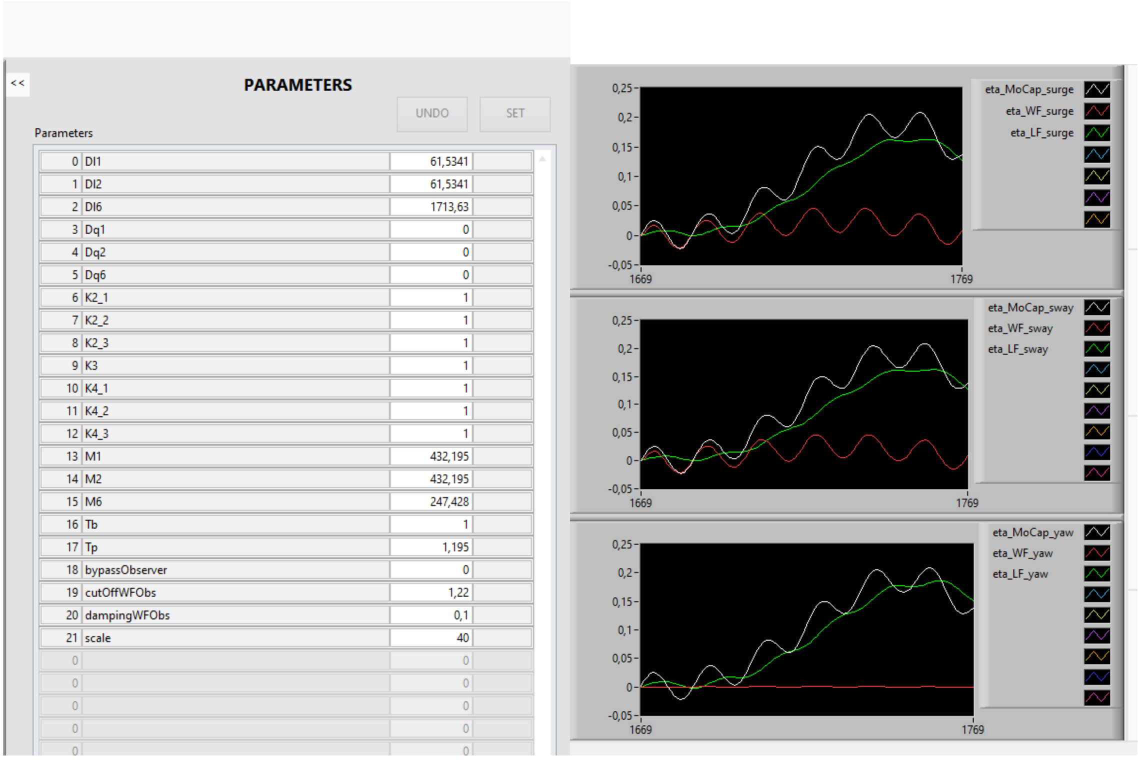
0	eta_WF_surge	0	Graph1
1	eta_WF_sway	0	Graph2
2	eta_WF_yaw	0	Graph3
3	eta_LF_surge	0	Graph1
4	eta_LF_sway	0	Graph2

(a) Configuration window in HLCC (b) Input and output in HLCC

Figure 3.10: Overview of a DLL in HLCC

Further down in the picture, the current status is shown. This field will show an error message if it occurs. The DLL-defined period is the set time step of the simulation and the DLL running period refers to the sampling time for the relevant DLL. HLCC is also equipped with an option for synchronizing the run-time of the respective DLL with the input from another DLL or the internal clock of the computer. This is useful for experiments where the inputs of various DLLs are dependent on each other.

At the bottom of the figure, the required input parameters can be assigned to the given input streams. Below the inports selector, one can find a button for saving the output. This should be pressed for all the modules one wishes to save input and output. After assigning inports, the pane to the right, Figure 3.10b displays all available inputs and outputs for the loaded DLL module, in this case, the wave- and low-frequent components of the incoming signal. Additionally, the real-time section also allows the plotting of the signal in a graph generator implemented in HLCC, which can be seen in Figure 3.11b.



Parameters

0	DI1	61,5341	
1	DI2	61,5341	
2	DI6	1713,63	
3	Dq1	0	
4	Dq2	0	
5	Dq6	0	
6	K2_1	1	
7	K2_2	1	
8	K2_3	1	
9	K3	1	
10	K4_1	1	
11	K4_2	1	
12	K4_3	1	
13	M1	432,195	
14	M2	432,195	
15	M6	247,428	
16	Tb	1	
17	Tp	1,195	
18	bypassObserver	0	
19	cutOffWFObs	1,22	
20	dampingWFObs	0,1	
21	scale	40	
0		0	
0		0	
0		0	
0		0	
0		0	

(a) Parameteres window in HLCC

(b) Plotting function window in HLCC

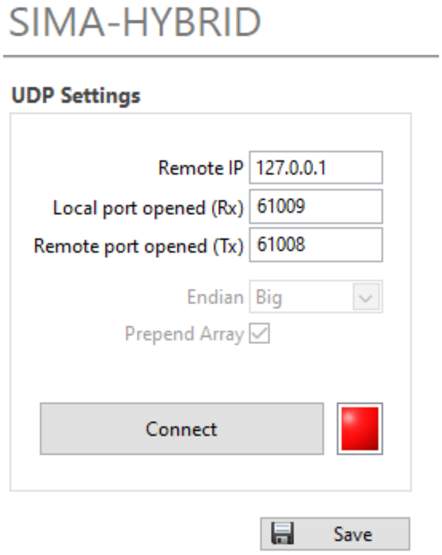
Figure 3.11: Overview of a DLL in HLCC

Figure 3.11a contains all the given parameters for the given model. This section also allows modification of these parameters, which can be reloaded for each simulation or changed back to initial values in the configuration panel. The final section of the DLL object in HLCC is shown in Figure 3.11b, which shows the values selected for plotting in real-time, as mentioned previously.

3.3.2 SIMA-Hybrid

The interface that enables connection from HLCC to SIMA is a DLL called SIMA-Hybrid. To establish a connection between SIMA and HLCC, a local area network is employed, using the standard IPv4 loopback traffic protocol address, 127.0.0.1. Additionally, port numbers for transmitting (TX) and receiving (RX) data packets are assigned values to ensure proper transmission between the two software. The values can be seen implemented in Figure 3.12. Once a valid connection is established, data flow during simulation is controlled by the JAVA control from Figure 3.7.

Information exchange between the two software is enabled through the use of the User Datagram Protocol (UDP), a fundamental protocol within the Internet Protocol (IP) suite. UDP does not guarantee reliable packet delivery and lacks error checking, flow control, or congestion control mechanisms. However, it is a fast and lightweight protocol, making it suitable for real-time communication applications that require high throughput and low latency, such as the one in question.



SIMA-HYBRID

UDP Settings

Remote IP 127.0.0.1

Local port opened (Rx) 61009

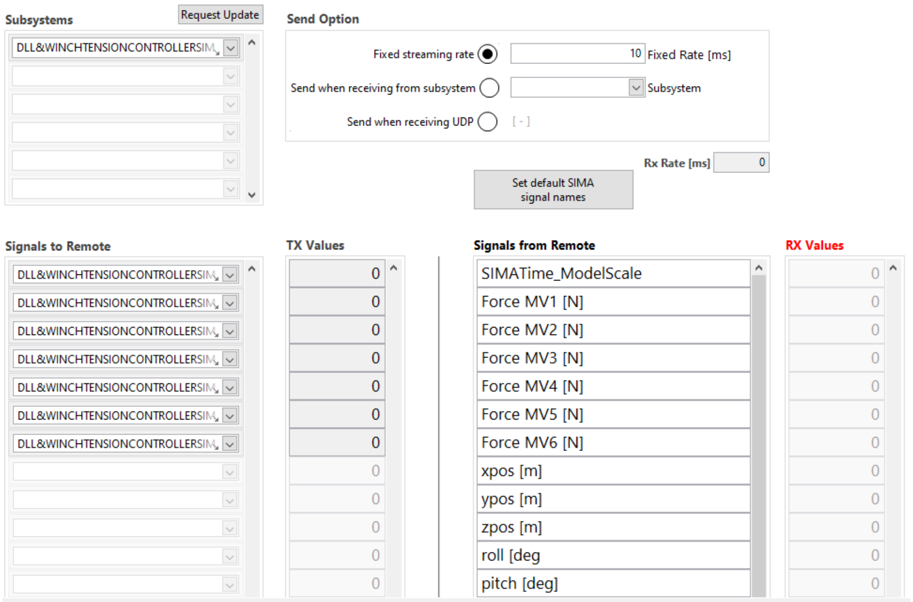
Remote port opened (Tx) 61008

Endian Big

Prepend Array ☑

Figure 3.12: UDP settings in SIMA-HYBRID, HLCC

Figure 3.13 shows the rest of the interface in SIMA-HYBRID. To the top left, one can see that SIMA-Hybrid subscribes values from the winch controller. In the section below, with signals to remote, one can see that the module sends 7 values to SIMA, these include the commanded angles for all six winches as well as the current simulation time in HLCC. The top right has various options for when data should be sent to SIMA, in this configuration it is sent with a fixed rate of 10 ms. In the bottom right, the values which HLCC receives from SIMA are shown with their real-time value during simulation in the RX-values column. As with the signals from the remote, an additional time signal is sent from SIMA. These two-time signals are compared in the JAVA control block to ensure that the simulation times in both environments correspond to each other.



Subsystems [Request Update]

DLL&WINCHTENSIONCONTROLLERSIM...

Send Option

Fixed streaming rate ● 10 Fixed Rate [ms]

Send when receiving from subsystem ○ Subsystem

Send when receiving UDP ○ [-]

Rx Rate [ms] 0

[Set default SIMA signal names]

Signals to Remote	TX Values	Signals from Remote	RX Values
DLL&WINCHTENSIONCONTROLLERSIM...	0	SIMATime_ModelScale	0
DLL&WINCHTENSIONCONTROLLERSIM...	0	Force MV1 [N]	0
DLL&WINCHTENSIONCONTROLLERSIM...	0	Force MV2 [N]	0
DLL&WINCHTENSIONCONTROLLERSIM...	0	Force MV3 [N]	0
DLL&WINCHTENSIONCONTROLLERSIM...	0	Force MV4 [N]	0
DLL&WINCHTENSIONCONTROLLERSIM...	0	Force MV5 [N]	0
DLL&WINCHTENSIONCONTROLLERSIM...	0	Force MV6 [N]	0
	0	xpos [m]	0
	0	ypos [m]	0
	0	zpos [m]	0
	0	roll [deg	0
	0	pitch [deg]	0

Figure 3.13: Interface of SIMA-HYBRID in HLCC

3.3.2.1 Subroutines in SIMA-Hybrid and synchronization

Supporting the process within SIMA-HYBRID are two DLLs; the *CoordinateTransform.fmu* and the *ComputeLoad.fmu*. The *CoordinateTransform.fmu* takes the positional data from SIMA and applies the theory in Section A.3 in order to rotate the positional input to the correct reference frame, in this case, the body frame. *ComputeLoad.fmu* is not essential for the setup to work as it only serves as a quality check that the applied and measured line tension matches. The updated signals are then ready to be sent into the remaining loop of the HLCC, shown in Figure 3.8.

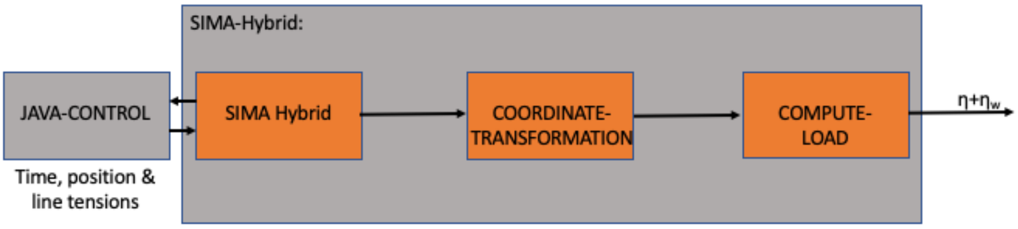


Figure 3.14: SIMA-HYBRID block with subroutines

As mentioned earlier in this section and showcased in Figure 3.10a and 3.13, it is possible to subscribe values from another DLL when running simulations. In order to synchronize the simulation with respect to the schematic block diagram in Figure 3.8 the following synchronization is applied in the following DLLs in HLCC:

- *CoordinateTransform.fmu* is synced with the internal clock of the computer
- *ComputeLoad.fmu* is synced with *CoordinateTransform.fmu*, which can alternatively be synced with the internal clock as it is not as important or a part of

the loop in the figure.

- *Observer.fmu* is also synced with *CoordinateTransform.fmu*
- *Mooring.fmu* is synced after *observer.fmu*
- *Allocation.fmu* synced after *mooring.fmu*
- *WinchTension.fmu* synced after allocation, ensuring the most updated value for the commanded line tension

3.3.3 Observer

The observer is built, based on the theory in chapter 2, and the FMU is generated from the Simulink model used in the project thesis, shown in chapter C, Figure C.1. It receives the input from *ComputeLoad.fmu*, consisting of the measured, but rotated position of the floater from SIMA, $\eta_{MoCap} = \eta + \eta_w$. As output, it returns the estimated low- and wave-frequent position, $\hat{\eta}$, the velocity, $\hat{\nu}$ as well as the bias force. The analysis of this report is performed on the output from the observer. This implies that the data from this block should be saved during the simulation. The interface and adjustable parameters are shown in Figure 3.11a.

3.3.4 Mooring

With reference to Figure 1.4 the mooring represents the numerical substructure that is simulated in the cyber-physical setup. *Mooring.fmu* therefore represents a virtual mooring configuration, meaning that there are no actual mooring lines connected to the floater model. The corresponding forces are instead applied by the winches in SIMA based on the position and velocity data received from the observer block. The calculations for mooring forces are based on the equation below which is derived in T. Sauder (2021), equation 8.

$$\tau_m = -R(\psi)^\top R(\alpha_m)[D_m R(\alpha_m)^\top R(\psi)\nu + K_m R(\alpha_m)^\top (\eta - \eta_{ref})] \tag{3.2}$$

The given variables, which also can be found in the FMU, and their explanation are listed below

- τ_m: desired mooring force vector
- α_m: main axis direction with respect to North
- $K_m = diag(k_i, k_t, k_{rot})$: inline, transverse, rotational stiffness
- $D_m = diag(d_i, d_t, d_{rot})$: inline, transverse, rotational damping
- $\eta_{ref} = (N_0, E_0, \psi_0)^\top$: equilibrium point or setpoint

The equation can further be simplified by stating that the damping term, α, and the equilibrium are zero, yielding the following:

$$\tau_m = -R(\psi)^\top K_m \eta \tag{3.3}$$

The *mooring.fmu* takes estimated position and velocity as inputs and calculates the mooring line loads which are further sent to the allocation block. In order to ensure stability and accurate simulations, the FMU is also equipped with values for the maximum allowable change of rate in heading and position as well as a forced period.

3.3.5 Allocation

The allocation block gets the forces acting on the virtual mooring system and computes the tension that should be exerted on the structure for each of the six actuator lines in the cable-driven parallel robot (CDPR). In other words, its purpose is to calculate and distribute the correct pull forces acting on the object. The change in tension for the controller is found by solving equation 9 in T. Sauder (2021):

$$\Delta T = A_c^\dagger(\tau - A_c T_0) \tag{3.4}$$

τ is the input received from the *mooring.fmu* and the allocation block produces the output: $\{T\} = (T_0 + \Delta T)$. As changeable parameters, the *allocation.fmu* has a variable for the adaptive gain of T_0 and limits for the maximum and minimum desired tension.

For a given T_0, the change in tension, ΔT can be obtained from a Moore-Penrose pseudo-inverse of the configuration matrix A_c. The main advantage of the Moore-Penrose pseudo-inverse is that it guarantees the existence of a unique symmetric solution and minimizes the error; $||Ax - b|| = |\Delta T|^2$. The values are further sent to the winch tension controller which applies the calculated tension to the SIMA model

3.3.6 Winch control

A crucial component of the simulation is the DLL that governs the operation of the winches. As mentioned in the section for SIMA-Hybrid, the block is subscribed to the output of the winch controller and the JAVA control is implemented in a way such that it compares the simulation times from both SIMA and HLCC before opening the data stream between the two.

winchTensionControllerSIMA.fmu receives inputs from the allocation block, the positional vector from the coordinate transformation block, as well as the measured line tension value and simulation time from SIMA-Hybrid. The output of this block is seven signals, consisting of one signal for each of the six winches indicating the commanded change in θ, along with a clock-timer signal. These output signals are sent to the SIMA-Hybrid block for further processing.

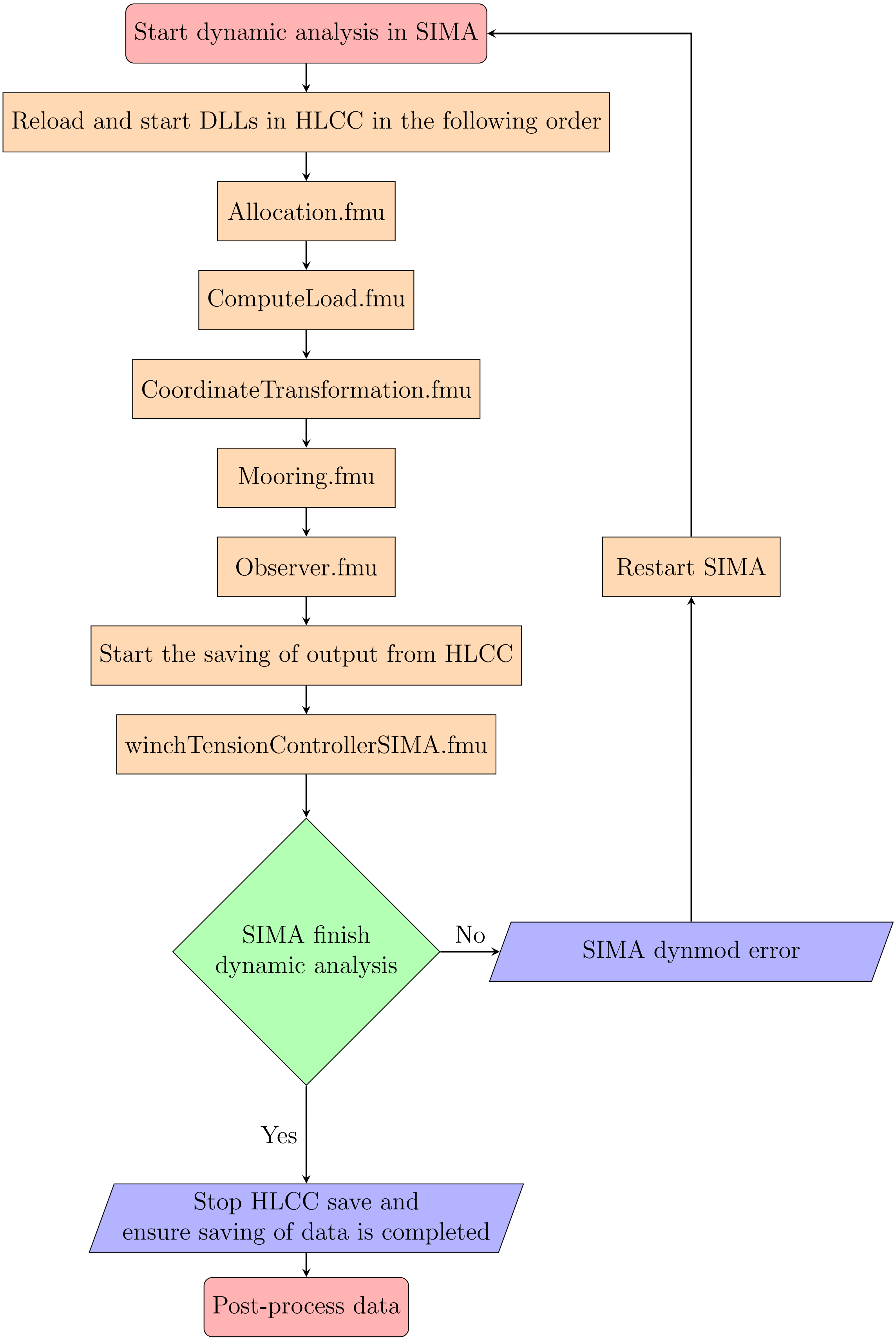
The FMU is fitted with adjustable parameters for controlling the winch radius, line stiffness, an internal gain for tension control as well as a variable for activating the controller.

3.4 How is data processed - experimental method

After loading all the necessary DLL modules in HLCC and establishing a connection with SIMA, the user is prepared to carry out a simulation. Before starting, all parameters should be adjusted in the observer and SIMA based on the problem being investigated.

First start a dynamic analysis in SIMA from the initial condition of the project. The simulation is then designed to stop at a given percentage, in this case, 8% of completion, indicating that the static analysis is complete and that it awaits input from HLCC before starting the dynamic calculations. To proceed with the simulation, the winch tension controller in HLCC must be activated. Proper reset of the internal clock timer of the winch tension controller before starting is critical for the simulation to run correctly, as failure to do so is likely to result in an error and simulation failure.

The procedure for conducting simulations with the current setup is organized in the flow chart below:


Start dynamic analysis in SIMA
Reload and start DLLs in HLCC in the following order
Allocation.fmu
ComputeLoad.fmu
CoordinateTransformation.fmu
Mooring.fmu
Observer.fmu
Start the saving of output from HLCC
winchTensionControllerSIMA.fmu
SIMA finish dynamic analysis
No
SIMA dynmod error
Restart SIMA
Yes
Stop HLCC save and ensure saving of data is completed
Post-process data

3.4.1 Post-processing in MATLAB

The post-processing of data has been done in MATLAB, and the routine was divided into three different files; one for processing data and assigning variables, one for computing the performance, and one for plotting.

The file format in which HLCC stores simulation data is *tdms_defrag* a type of file format used in National Instruments (NI) software and hardware systems. It is a binary file format that is used to store measurement data acquired from NI systems. The file format; *tdms_defrag* is optimized for reading and writing data quickly by organizing the data in contiguous blocks on the hard drive. This helps to improve performance when reading or writing large amounts of data.

3.4.2 Performance criteria

Having collected data for a simulation and wanting to compare its performance with other simulations, a quantitative result and mean of measuring the performance of the system has been developed. As this project seeks to verify that the proposed observer can perform an online estimation and wave filtering it will be benchmarked against offline estimates. For the offline estimates, the whole times series will be available allowing phase-free wave filtering. The performance criterion is denoted, ϵ, and inspired by mathematical theory for norms on $\mathcal{L}_p$-spaces, previously described in Section 2.5.

$$||x||_p = \left[\int_0^\infty |x(t)|^p \, dt\right]^{1/p} \tag{3.5}$$

From the general expression, the $\mathcal{L}_2$ norm with an inner product structure is derived. This corresponds to the square root of the integral of the difference between the two values over the time of the entire series. The root makes the equation homogeneous and the factor $\frac{1}{T}$ will compensate for an otherwise increasing error over time.

$$\epsilon = \sqrt{\frac{1}{T}\int_0^T (\hat{f} - f^*)^2 \, dt} \tag{3.6}$$

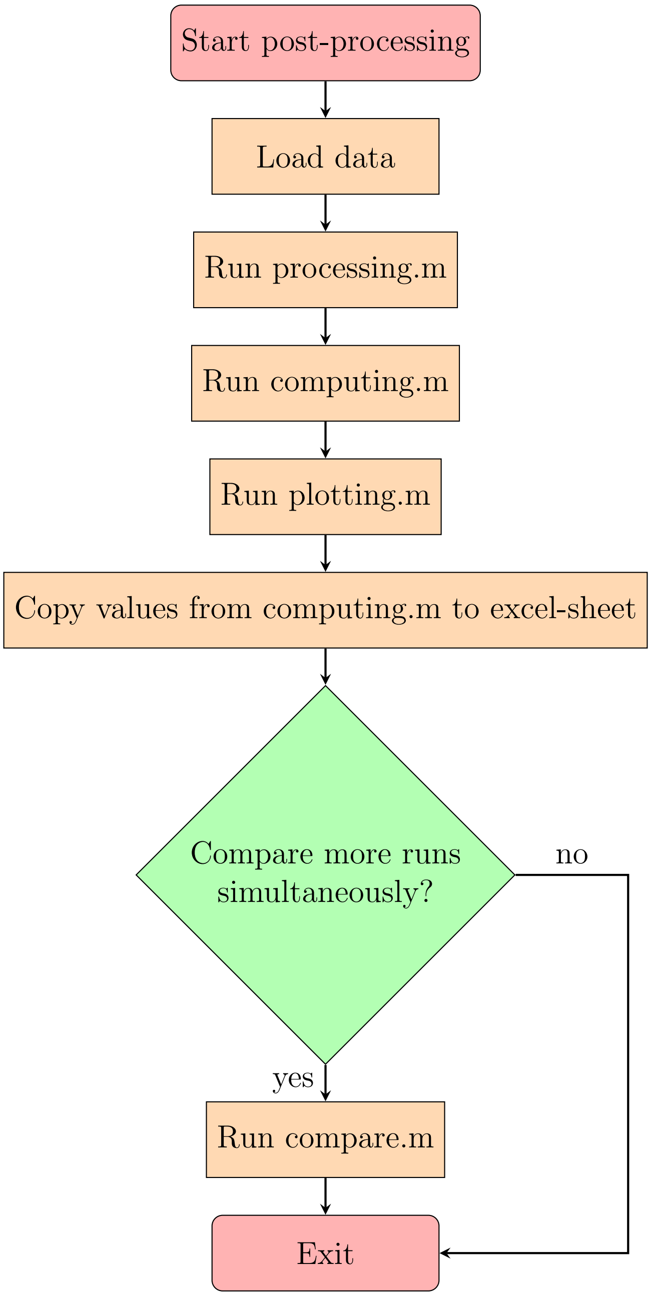
the result can further be divided by the norm of the latter function to normalize the result for any arbitrary input.

$$\epsilon = \sqrt{\frac{\int_0^T (\hat{f} - f^*)^2 \, dt}{\int_0^T {f^*}^2 \, dt}} \tag{3.7}$$

$\hat{f}$ represents the online filtered result while f^* is the “offline” low-pass filtered result, generated by applying a Lanczos filter in MATLAB to the measured position. It is referred to as an offline estimate since the function has access to both past and future values of the time series. The reason why the Lanczos filter is used as the offline measurement is that it is able to provide a low-pass filtered signal of the input function while increasing the sampling frequency when interpolating between values.

3.4.3 Workflow in post-processing

The following flowchart portrays the workflow when processing data after a successful simulation.



3.4.3.1 processing.m

The processing routine is done to ensure that the tdms-file is loaded correctly into MATLAB for further analysis. The file from HLCC in .tdms format is read using the *tdmsread()* function from the Data Acquisition toolbox. Data can then be accessed as a "cell" type containing tables with the project information, inputs, and outputs from HLCC. The order in which data is stored in the cell might vary, so the processing routine also ensures that the correct input and output are found based on the table sizes in the cell.

As the saving of the simulation starts and ends at different times, measures are done to ensure similar responses. All data where input is constant or zero at the beginning and end of the time series is removed as this indicates that the simulation in SIMA has not started yet or is already done.

Ideal filtering is applied using the *.Lanczos* function in MATLAB. This is a low-pass filter set with a cutoff period of 30 seconds for the full-scale model, well above the intended incoming wave periods for these simulations.

During simulations, it was discovered that using a timestep, dt for the lowpass filter, defined by subtracting elements in the time vector from HLCC gave a varying value for dt. To avoid this in further simulations, the time step was therefore set to the mean of the gradient of the time vector, which in most cases corresponded to 5ms seconds. Large deviations from a 5ms mean and 1ms standard deviation should be handled, and the simulation should most likely be performed again. A larger timestep, equal to the one in SIMA of 0.063 was tested but this gave a high-frequent result which will be wrong when trying to display an ideal filtered low-frequent signal.

The filter is applied to the measured positional data, η_{MoCap} in all three degrees of freedom. It is also desirable to have a reference for the velocity estimates produced by the observer algorithm. This has been done by applying ideal filtering to the gradient of the filtered positional data:

$$\frac{\nabla \eta}{dt} \tag{3.8}$$

which resulted in velocities for the global reference frame. In order for it to be compared with the data produced by the observer, the rotational matrix, R_b^n defined in Section A.3 is applied, making it a filtered local velocity estimate denoted as $\{u, v, r\}$ for surge, sway, and yaw.

3.4.3.2 computing.m

This routine computes indicators for the performance of all measurable outputs of the observer in all three degrees of freedom as well as a total sum of performance. The indicators are given as:

b_1^{ind}, b_2^{ind} and b_3^{ind} for bias:

$$\sqrt{\frac{\sum(bias - \overline{F_i})^2}{(\overline{F_i}T)^2}} \tag{3.9}$$

$\overline{F_i}$ is the mean of the reference force along the respective degree of freedom in surge, sway, and yaw. T is the length of the time series over which the indicator is computed, normalizing the result.

u, v, and r for velocities:

$$\sqrt{\frac{\sum(\nu - \nu_{filt})^2}{(\sum \nu_{filt})^2}} \tag{3.10}$$

where ν is the estimated velocity and ν_{filt} is the derivative of the filtered position found in *processing.m*. N, E, and ψ for position:

$$\sqrt{\frac{\sum(\eta - \eta_{filt})^2}{(\sum \eta_{filt})^2}} \tag{3.11}$$

η is the estimated low-frequency position and η_{filt} the low-pass filtered measured position.

3.4.3.3 plotting.m

As proven in the work leading up to this thesis, just looking at the performance indicator computed for each simulation is not enough to fully evaluate whether a simulation was good or not. Small indicator values, should ideally yield good estimates and a successful simulation. However, some cases showed that simulations could have a good indicator but poor response with for example wave-frequency content in the estimated low-frequent response.

The purpose of this routine is to make plots of all the outputs. It generates a total of seven plots; one for the displacement in all three degrees of freedom (surge, sway, and yaw), one plot with a sliced response in surge, one plot with all the velocities, one with the reference force in the mooring system as well as bias forces for surge and yaw.

All these plots prove useful when looking closer at why the observer performs as it does in the given simulation. It also enables visual detection of any irregularities or discrepancies in the simulation.

3.4.3.4 compare.m

A separate script has been made in order to compare the responses from multiple simulations in the same plot. This script implements the functionality of *processing.m* and *plotting.m* in one file and makes it flexible for more simulations at the same time.

One difficulty with this program is that the start and end of the time series have been saved for slightly different times, in addition to having different time steps. In an attempt to avoid these differences, only the middle of the time series have been compared. This is achieved by dividing the series in the middle and then only extracting the leading and following third from this point of the series. Plotting these curves gave an acceptable alignment which can be used for further analysis. It is worth noting that the plots from this code should not be compared with respect to their intermediate phase difference, only in terms of magnitudes and frequency content.

Reference values for bias force, position, and velocity estimates have been calculated as the mean of input for the position, of derived velocity and reference mooring force. This value is not likely to be very accurate but serves as a measure for the comparison of responses in terms of magnitudes and convergence.

3.4.4 Collection of simulation data

As the simulation cases grew in numbers, it became evident that a means to store and compare their performance indicator values was necessary. This has been done by creating a table in Excel containing all relevant information for each simulation. The sheet proved handy for tracking progress, changes, improvements, and group data based on parameters. Some built-in features of Excel have also been used to make visual comparisons, i.e. plots and conditional formatting. Some of these results will be presented as tables in the results chapter later on.

3.5 Model verification

In order to verify the model parameters of the INO WINDMOOR 12MW FWT model for use in the observer algorithm, a decay test using HLCC to provoke movement in SIMA has been conducted. The result from this decay test is then compared to the ordinary differential equation (ODE) solution of a mass-spring-damper system with the same characteristics as the FWT. The ODE has been defined and solved in MATLAB using the ODE45 command.

By changing the "bypassObserver" parameter of the NPO in HLCC from 0 to 1, the observer changes from being a state estimator to just copying the input as output, i.e. bypassing the incoming signal. This ensures that the simulated response is not affected by any dynamics in the observer algorithm. The decay test is then conducted by starting the simulation with no environmental loads and waiting a few seconds for a steady-state response of the substructure. This can be seen during simulation in HLCC by plotting the measured position. Once the transient ends and the structure moves with as small motions as possible, the positional reference in the *mooring.fmu* can be changed. The transient usually ends after 30-50 seconds. Changing the positional reference in *mooring.fmu* results in the system trying to bring the structure to its new reference and maintain the position.

The simulations for decay tests have been run for a longer time of about 1000 seconds compared to the other simulations to ensure that a steady state is reached before and after setting the new positional reference. For analysis and comparison in MATLAB, the signals from ”CoordinateTransform.fmu” was extracted from HLCC, the purpose of this was to get data in the proper reference frame. All positional references used in decay tests are listed below, note that it is the model-scaled reference that is used as all parameters in HLCC are in this scale:

Degree of freedom	Full-scale reference	Model-scale reference
Surge, N_0	10 [m]	0.25 [m]
Sway, E_0	10 [m]	0.25 [m]
Yaw, ψ_0	5 [deg]	5 [deg]

Table 3.5: Positional references for decay tests

The model solved using ODE45 builds on the system equation simplified into an equation of motion representing a mass-spring-damper system:

$$(M + A)\dot{\nu} + D_1\nu + D_2\nu|\nu| + K(\eta - \eta_{ref}) = 0 \tag{3.12}$$

Which is solved for acceleration and given as input to the ODE45 function:

$$\dot{\nu} = (M + A)^{-1}[-K(\eta - \eta_{ref}) - D_1\nu - D_2\nu|\nu|] \tag{3.13}$$

For evaluating and comparing results, the critical damping can be computed using the formula:

$$c_{critical} = 2 \cdot \sqrt{K \cdot (M + A)} \tag{3.14}$$

and then the damping ratio is further found by:

$$\zeta = \frac{D_1}{c_{critical}} \tag{3.15}$$

Model parameters for fitting are found by starting with the values from SIMA and making adjustments until alignment is achieved. An alternative method of verifying a model using decay tests is to compute the period of oscillations from the natural frequency:

$$\omega_n = \sqrt{\frac{K}{M + A}} \tag{3.16}$$

And check whether the response from decay tests has the same period. This method will often have less accuracy but serves useful as an initial check of parameters.

3.6 Tuning of observer algorithm

This section presents the procedure of tuning the nonlinear passive observer algorithm which has been performed in this thesis.

The process of tuning has been done as an iterative parameter study, i.e. changing the input in one parameter while keeping the others constant and then observing the change in output before moving on to the next parameter once satisfied with the obtained result.

The FMU with the observer is loaded with the gain terms normalized as 1. Tuning is then done by multiplying the gain constant by a value in HLLC and assessing the response of the system. Results from simulations are compared based on indicators and plots and the most promising values of the respective parameter are kept while moving on to the next. Repeat for all degrees of freedom, starting with a focus on tuning surge. For the parameters where it is possible to make changes with respect to the degree of freedom, all should be changed equally with respect to the degree of freedom.

The order of what parameters to tune has been predetermined, beginning with the bias term, ensuring convergence with as few oscillations as possible. Moving on to the velocity terms, where it is desired that the estimated velocity approaches the low-pass filtered velocity, derived from the position of the floater. Then the terms related to positional estimates are tuned before finally making adjustments to the wave filtering, bias time constant, and damping to see if some improvements can be made.

Chapter 4

Results

In this chapter, the simulated results will be presented, more than 200 simulations have been carried out and not all will be portrayed. Their average length has been of 1500 seconds in SIMA, corresponding to about 200 seconds when plotted and post-processed. This has proven sufficient, allowing the system to reach a steady state. The first goal of running these simulations is to showcase how the observer can be tuned in order to obtain a balanced result with the best possible performance. Secondly, it aims to show how the proposed software-in-the-loop setup can be used to verify a model concept.

Results will be judged based on calculated performance indicators and generated plots. The results of all simulations have been collected and saved. Generally, the analysis will be focused on velocity and position estimates, as some simulations revealed that some total and bias indicator values provided misleading results. The bias will be commented upon when relevant during the fitting process.

4.1 Model verification

The provided model had default parameters related to another project, these can be seen in Table B.1. In order for the observer to function as desired, its parameters must be adapted to the current simulation model. This was discovered by running an initial simulation with default parameters, yielding a response for which the observer was not able to provide estimates for the velocity. Revising the initial model parameters reveals that the quadratic damping term holds alarmingly large values compared to the other parameters. Additionally, it was discovered that the masses did not agree with the values in Souza (2021).

This section presents the results from this study as figures with the response from SIMA for decay tests in surge, sway, and yaw compared to a solution provided by the ordinary differential equation with respective model parameters in MATLAB. The procedure is based on the method of model verification described earlier in Section 3.5 and the tests have been conducted using the listed positional references, the results are as follows.

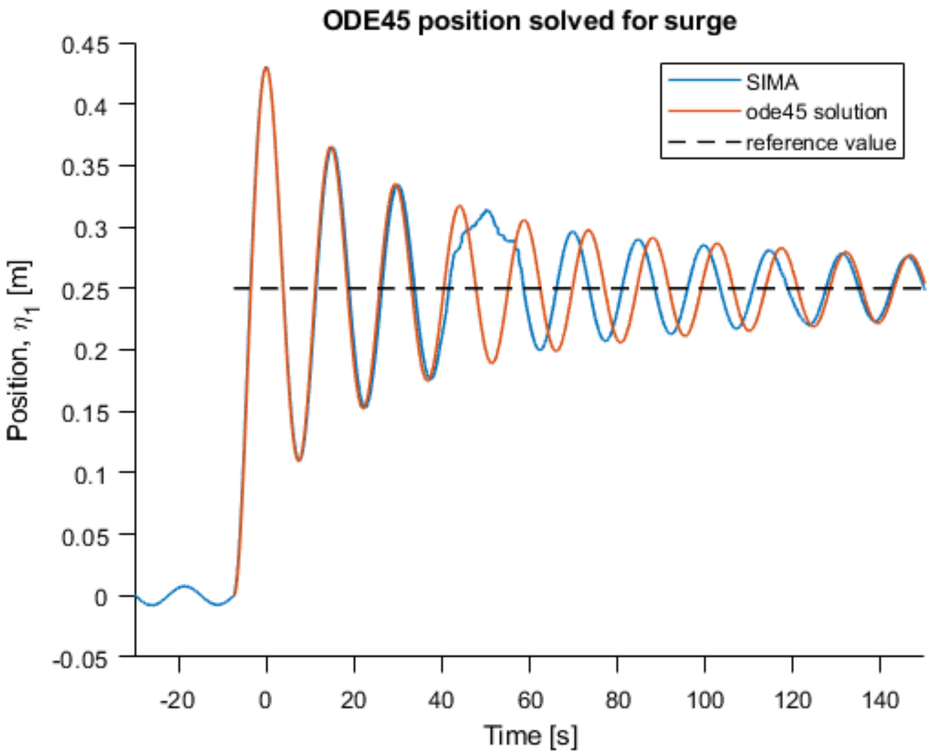


Figure 4.1: ODE fit vs SIMA response in surge

Figure 4.1 shows the surge decay test compared to the simulated response of an Ordinary Differential Equation (ODE) fitted with parameters from SIMA. The two responses demonstrate a satisfactory alignment during the initial three oscillations. However, a phase shift occurs thereafter. Despite this phase shift, it is evident that the level of decay and damping remains relatively consistent between the two curves. The phase shift can be attributed to two factors. Firstly, the response in SIMA does not achieve complete stillness before altering the surge reference, resulting in a minor phase difference between the two responses from the outset. Secondly, at the bottom of the third oscillation, a jitter can be observed in the SIMA signal. This jitter is likely to introduce a delay in the response, which propagates through the signal and contributes to the observed discrepancy. At 120 seconds the two responses align again.

The same test is conducted in sway with results shown in chapter B, Figure B.1a. In this, the reference is changed in E_0 instead of N_0. The resulting response is quite similar to the surge analysis where satisfactory alignment is obtained between SIMA and the ODE simulation. However, there are a few distinctions between the two simulations. The main difference is that the response from SIMA is practically still before changing reference. Despite this, the phase shift still occurs. There are also no obvious signs of jitter in the signal. Given the good fit observed during the initial three oscillations and the acceptable similarity in relative damping, no additional adjustments were made to further refine the response.

The yaw decay simulations did not yield a fit as satisfactory as those obtained for the two previous scenarios. Initial simulations suggested that the system was either too stiff or had excessive mass. Updated stiffness parameters were provided based on those used in the physical experimental setup, where the mooring system was replaced by linear springs. Their values can be seen in the second line in Table 4.1. A third source of values was discovered when studying the setup used for offline simulations of the model which can be seen in Table B.3.

Origin	k_i [N/m]	k_t [N/m]	k_{rot} [Nm/deg]
Initial values	53.36	53.36	5.449
Experimental setup	62.2	59.94	1.983
Offline simulation	58.48	59.95	79.33

Table 4.1: Mooring stiffness for three conditions

The three sets of stiffness show good alignment in terms of inline and transversal stiffness across all three cases. However, there is a noticeable difference in rotational stiffness. This was believed to be the main cause of misalignment between SIMA and the ODE solution in yaw. Further research revealed more sources of error.

It was also discovered that the given mass and linear damping coefficient in yaw were expressed in radians per second. SIMA provides measurements in degrees. Applying a conversion factor of $\frac{180}{\pi} = 57$ yielded a matching response between SIMA and the fitted ODE solution, which can be seen in Figure B.1b. This proved a useful lesson and highlights the significance of correctly scaling units in order to achieve accurate results. It is essential to ensure that such scaling and unit details are adequately documented in both SIMA and HLCC for future reference.

The values of all fitted ODE solutions are given in the table below:

DOF	M+A	K	D_l	D_q	C_{cr}	ζ
Surge	338	62.2	0.139	389	269	$< 1\%$
Sway	338	59.94	0.139	389	269	$< 1\%$
Yaw	3.53	1.983	0.1098	0.06	5.29	2%

Table 4.2: Fitted model parameters

The damping ratio from Table 4.2 might give the misconception that the system is lightly damped. However, the parameter only indicates that the system has little linear damping, and when comparing the values of D_l and D_q it is evident that the system is dominated by quadratic damping, especially for surge and sway. The ratio is included as a mean for comparing the damping of the degrees of freedom.

4.1.1 Discussion of model verification

Model parameters that provided a good fit between SIMA and the ODE solution in MATLAB were found. After about three oscillations a phase difference can be observed for surge and sway. This shift in phase might be caused by a jitter in the incoming signal from SIMA. A closer inspection of the curve reveals a deviation from the smooth sinusoidal signal at the bottom of the oscillation after the third top. This will likely accumulate over time and lead to a propagating difference between the two signals. One reason for the disparity in the signals can also be attributed to the complexity of the SIMA model, which incorporates more intricate forces compared to a basic mass-spring-damper equation.

The simulations and results in this section have been generated without the observer and with the mooring model using the measured position directly from SIMA. This functionality was enabled by setting the "bypassObserver" parameter in the *observer.fmu* to 1. An initial attempt was made with the parameter "bypassObserver" set to 0. This proved critical as the dynamics of the observer affected the response of the substructure. The test will provide a wrong response as the observer affects the mooring with wrong position estimates. Despite this, it is still possible to make a fit, but the ODE parameters will deviate significantly from those in SIMA. Resulting in larger mass and linear damping. Errors like this can cause faulty model verification which will affect the result of the estimates provided by the observer.

The assumption that the quadratic damping coefficient in surge and sway could be set equal is supported and kept based on these simulations. This section also shows the importance of having a good understanding of the physical system when verifying and modeling, wrong units and conversions can cause severe problems and a forced fit might yield an unstable system.

It is important to note that the verification of model parameters for yaw has been done in degrees. The internal dynamics of the observer computes values using radians, while its input and output will be in degrees. SIMA also works with angle measurements in degrees. This explains the relatively small values in yaw when compared to surge and sway for mass and quadratic damping. The observer algorithm will scale the terms of linear and quadratic damping by multiplying with $(\frac{180}{\pi})$ and $(\frac{180}{\pi})^2$. This will give values of $D_{l,6} = 6.2908$ $[\frac{Ns}{rad}]$ and $D_{q,6} = 197$ $[\frac{Nms}{rad}]$. However, the mass is not being scaled in the observer model, so it must be scaled manually before assigning it to the observer. By also multiplying this value by $(\frac{180}{\pi})$, the mass for yaw to be used in the observer will be equal to 202.6 kg. If this is not properly done, the observer will still be stable, but the estimates in yaw will be wrong.

4.2 Updated model parameters

Having verified and updated the model parameters, a new simulation was conducted to see if improvements had been made. Values are shown in Table B.2 and the updated response in surge has been plotted in the figure below.

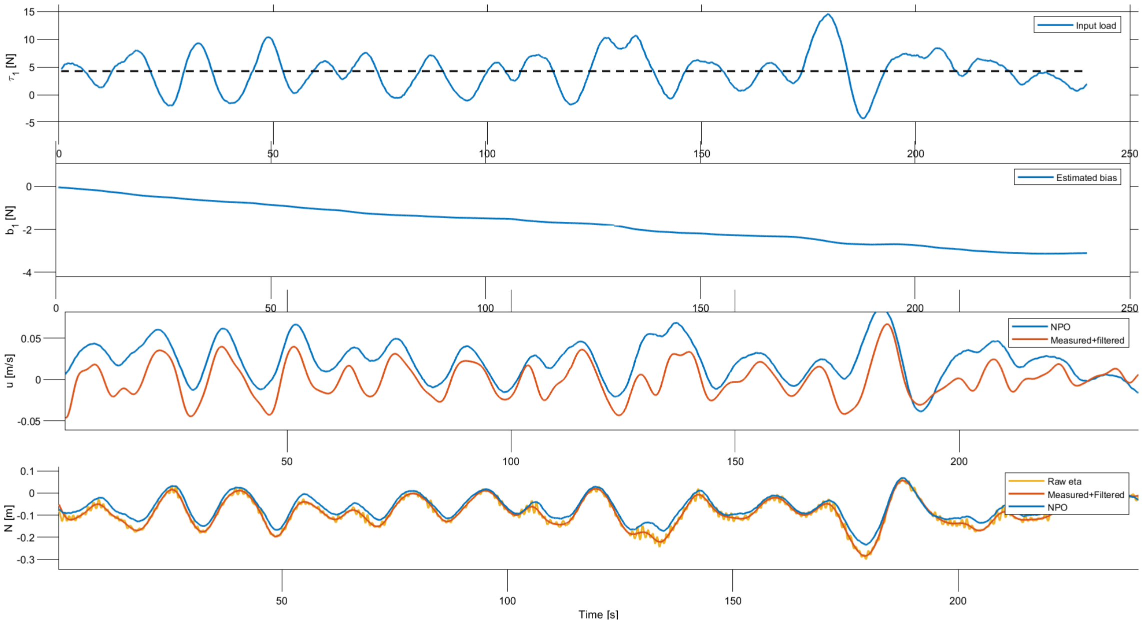


Figure 4.2: Simulations with updated model parameters

The observer is now able to make a velocity estimate and the bias also converges toward the negative mean of the reference force. From this point on it is assumed that the model parameters are correct and remaining discrepancies can be adjusted for by tuning the other model parameters.

4.3 Tuning

Following the establishment of the fitting model parameters, the process of tuning the observer was initiated using the procedure from Section 3.6. This section aims to present the obtained results throughout the observer tuning procedure.

The focus was primarily directed toward the surge direction. However, considering the model's similarity in sway, it was assumed that adjusting the parameters in a similar manner as for surge would also lead to satisfactory tuning in sway. This assumption will be revisited later in this chapter.

4.3.1 Bias tuning, K3

The initial parameter that undergoes tuning is the bias gain, denoted as K_3. This parameter is expected to have minimal impact on other parameters, making it relatively easy to adjust. The desired outcome is for the bias to converge towards the mean wave drift force, equal to the negative mean of the applied mooring force. This value is hereby referred to as the bias reference value. The signal should also be without high-frequency or wave-like components. The presence of high-frequency content in the bias force indicates an excessively large gain, while a slow convergence towards a mean value suggests an insufficient gain. The objective is to strike a balance where the bias gain is appropriately tuned to facilitate convergence towards the reference while maintaining a bias signal that is free from undesired frequency components.

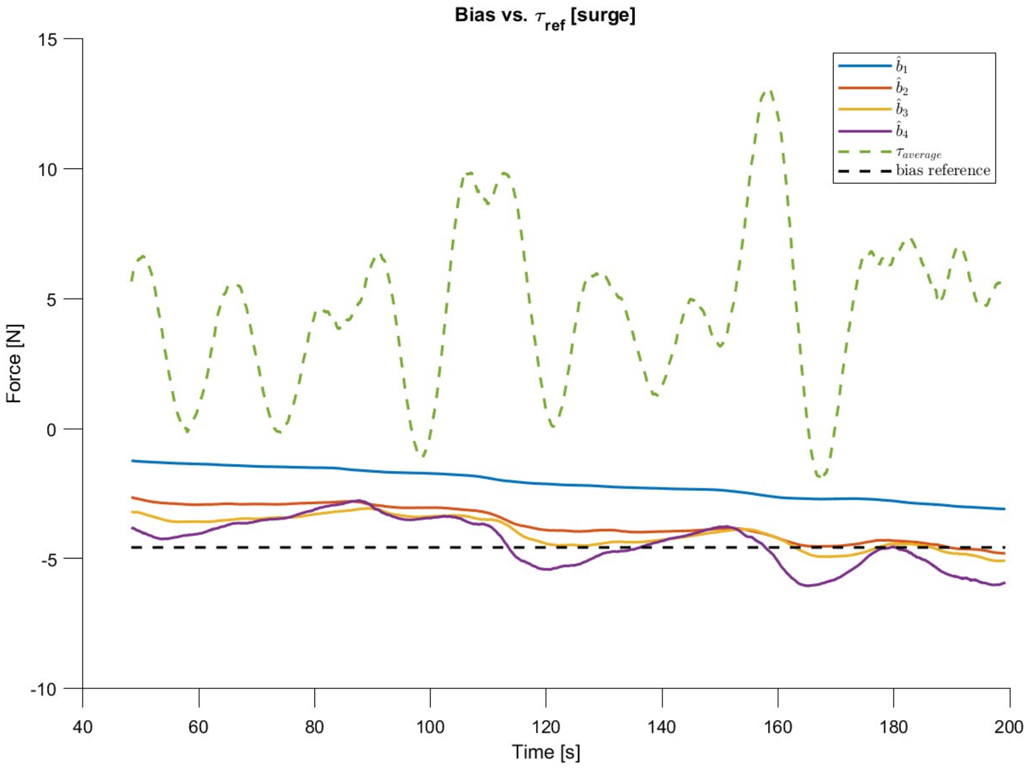


Figure 4.3: Bias plotted for various values of $K_3 = [1, 3, 5, 10$

The above figure presents the bias force for four simulations corresponding to different values of K_3, specifically 1, 3, 5, and 10. The average of the mooring force for all simulations is also plotted as a reference using a green dotted line, while the negative of its mean is depicted by a black dotted line. From the lower four curves, it can be observed that an increase in the K_3 gain leads to a larger bias force with progressively more oscillations in the force, as expected. As previously noted, a bias gain constant of $K_3 = 1$ is insufficient for the bias to converge within the simulation time frame. It is anticipated that by adjusting the bias time constant, denoted as T_b, the curve can be smoothed out. Further investigation into this matter will be conducted. The table below displays the indicator values for the simulations.

Indicator values in surge			
Simulation	**Bias, b_1**	**Velocity, u**	**Position, N**
1,K_3=1	1.30	1.27	0.22
2,K_3=3	1.54	0.92	0.17
3,K_3=5	1.65	0.85	0.16
4,K_3=10	1.72	0.81	0.16
5,K_3=0,5	1.10	1.53	0.25

Table 4.3: Indicator values for varying bias gain with the smallest and largest values highlighted.

To compare the effects of reducing the bias gain constant, an additional simulation was conducted. The corresponding values for this simulation are displayed in the fifth line of Table 4.3.1. Surprisingly, the reduction in the bias gain resulted in the lowest indicator value, while an increase in K_3 led to a higher indicator value for bias. From Table 4.3.1 increasing K_3 also seemingly has a positive effect on the indicator values for velocity and position. Since the emphasis is on bias, the impact of tuning on the other parameters is downplayed, although it still demonstrates their interdependency. The response of the simulation with the reduced gain is also compared to the first simulation, which is provided in the appendix, Figure B.2 that portrays an even slower convergence than for $K_3 = 1$.

The purpose of this simulation case was to establish the gain term, K_3 that would ensure that the bias force converged toward the mean wave drift force. As observed, an increasing gain yielded a faster convergence, but a more oscillating force. In this case, the value from the second simulation with $K_3 = 3$ is chosen as this is the first to ensure convergence. K_3 is different from the other gains in the way that it is only adjustable in one DOF and not all three. It would perhaps be an improvement to allow this gain to be adjusted in all directions, allowing more flexibility. Throughout the simulations, it was observed that in certain scenarios, an increase in gain adversely affected the bias indicator for other degrees of freedom. This phenomenon could potentially lead to misleading results, prompting the decision to downplay the importance of total and bias indicator values. Consequently, only the indicator values for surge are presented in the tables hereafter.

4.3.2 Velocity injection gain, K4

The next parameter for tuning is K_4 which governs the velocity estimates. A total of 12 simulations have been conducted with varying values of K_4 in combination with K_3. Four of the simulations are shown in the same figure, with the parameters from the table below:

Simulation	K_3	$K_{4,i}$
1	3	1
2	3	3
3	3	5
4	3	10

Table 4.4: Simulation values for K_4

The subsequent figure shows the response in velocity estimate when altering the value of K_4 with the updated value for K_3 from the previous section.

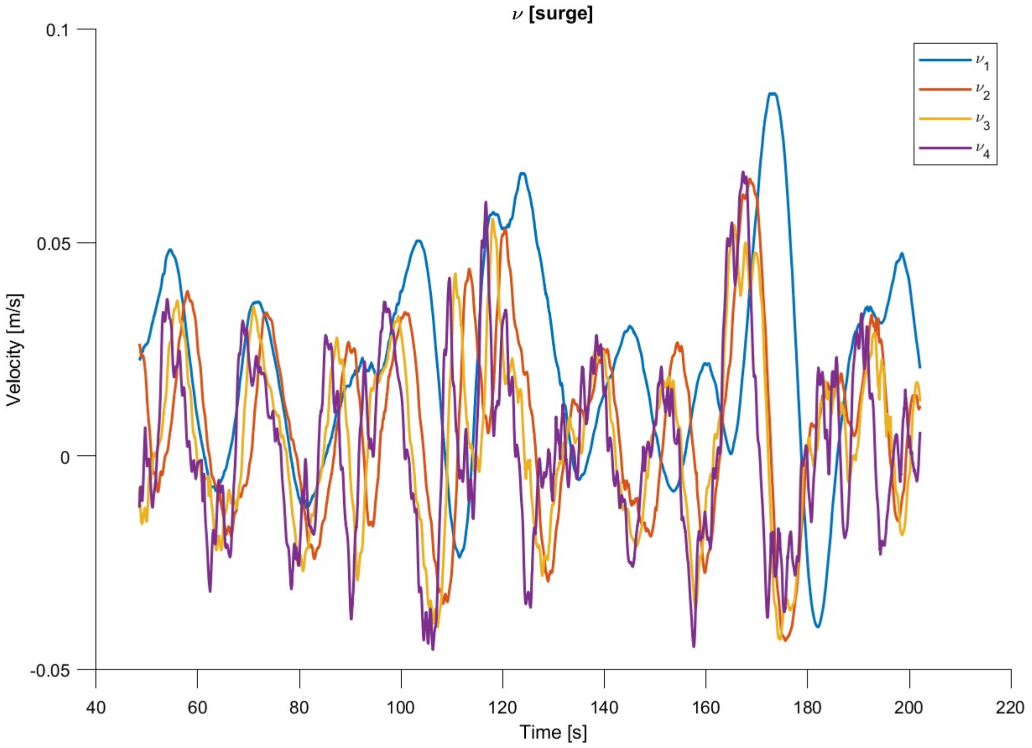


Figure 4.4: ν plotted for various values of K_4

The plot shows that increasing the gain, K_4 gives an estimate with more wave-frequent content, but seemingly less phase delay. This effect can also be observed in Figure B.4 where the bias force becomes more high-frequent for the higher values of K_4.

Equation 2.45 states that the relation $\frac{K_3}{K_4}$ should be less than the frequency of the incoming wave and larger than $\frac{1}{T_i}$, in numerical values: $\frac{1}{T_i} = 1 << \frac{K_3}{K_4} < \omega_0 = 1.195 < \omega_c = 1.22$. By solving for K_4:

$$\frac{K_3}{K_4} = 1.111 < 1.195 \tag{4.1}$$

one obtains a value of 2.7.

Indicator values in surge			
$\mathbf{K_3}$	$\mathbf{K_{4,i}}$	**Velocity,** u	**Position,** N
1	0.5	2.13	0.32
3	0.5	1.51	0.26
1	1	1.35	0.22
3	1	0.92	0.17
1	3	0.55	0.11
3	2.6	0.51	0.10
3	2.7	0.53	0.10
3	3	0.50	0.09
1	5	0.43	0.08
3	5	0.39	0.07
1	10	0.40	0.06
3	10	0.38	0.06

Table 4.5: Indicator values in surge for varying values of K_3 and K_4

Table 4.5 shows that an increase in K_4 yields better estimates for both velocity and position based on the indicators in surge. However, it is important to keep in mind plotted responses from earlier where increasing the gain too much results in a more high-frequent response. Considering this, the optimal solution from this was taken as a middle value, marked in yellow which also fulfills the frequency relation from Equation 2.45.

4.3.3 Position injection gain, K2

Once both the bias and velocity injection were tuned to a satisfactory response, the next quantity for tuning is the position injection gain which is adjusted by the change in K_2. When tuning K_2 it is desired that the estimated position aligns with the offline low-pass filtered position.

Figure B.5a shows that when increasing only K_2 and leaving $K_3 = K_4 = 1$, the positional estimates get less phase delay, lower amplitudes, and becomes more high-frequent, as was seen for K_4. Meanwhile, it can be observed from Figure B.5b that increasing the value of K_2 also leads to larger amplitudes in the velocity estimate.

Further, the system response has been simulated for the updated value of K_3 while changing K_2 to see if the updated bias gain affects the results in any critical way. Figure B.6 exhibits a similar trend as previously observed, wherein an augmentation in K_2 continues to result in an increase in the amplitude of the velocity estimate. Figure 4.5 displays indicator values from the simulations, grouped with respect to K_2. The observations from figures B.5 and B.6 are confirmed by the indicator as larger K_2 values lead to a larger velocity indicator and a smaller indicator for the positional estimate. The bias indicator remains relatively constant.

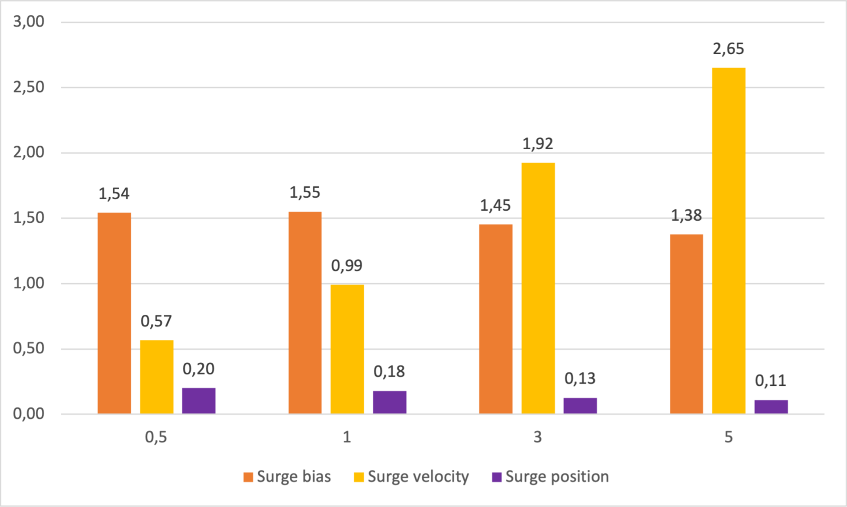


Figure 4.5: Indicator values for $K_3 = 3$ and varying K_2 in surge only

Differentiation between the simulations has been done based on indicators, where the choice falls on $K_{2,i} = 1$, the value that is used further on. This is supported by the observation that a too large increase in the gain will result in the positional estimate being corrupted by wave frequencies.

4.3.4 Final tuning in surge

Having looked at the effect of tuning the various gains, the purpose of this section is to see the effect when applying all adjustments simultaneously. Three sets of values have been chosen for further comparison and study.

Indicator values in surge				
$\mathbf{K_{2,i}}$	$\mathbf{K_3}$	$\mathbf{K_{4,i}}$	**Velocity,** u	**Position,** N
0.5	3	2.7	0.48	0.13
1	3	2.7	0.50	0.10
3	3	2.7	1.06	0.08

Table 4.6: Indicator values in surge for combined tuning parameters

Based on this, it is decided to keep the values of $K_2 = 1$, $K_3 = 3$ & $K_{4,i} = 2.7$ since it yields a balanced result for all indicators when comparing with the other two simulations of $K_{2,i} = 0, 5$ or 1. The following figure shows the response for all degrees of freedom using these values.

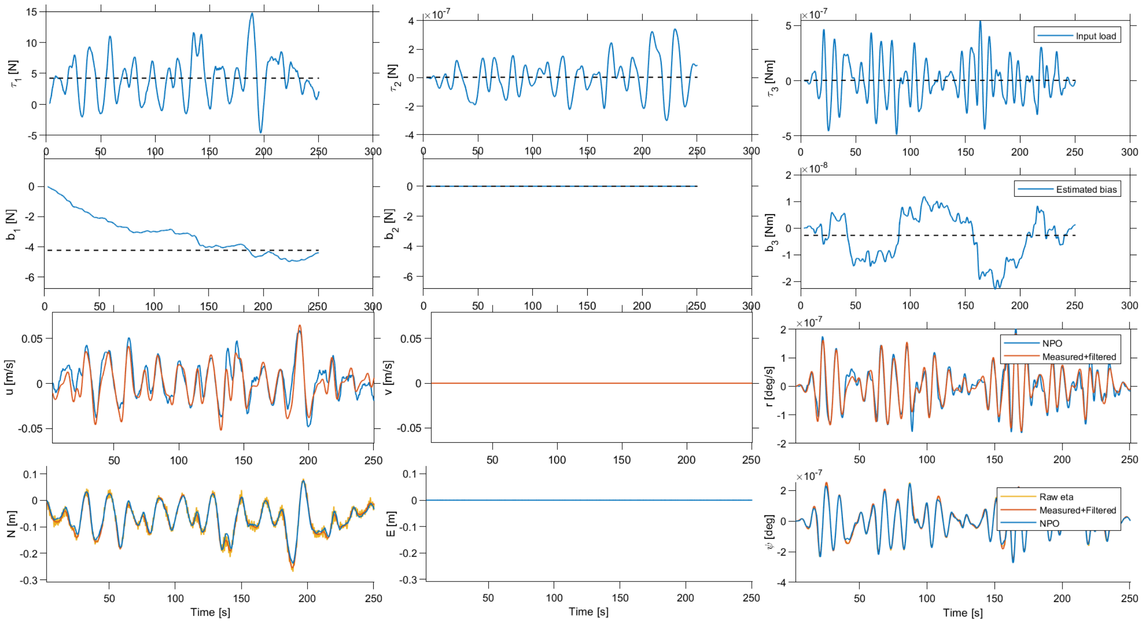


Figure 4.6: Simulation with tuned parameters in head seas. Note that the values for sway and yaw are very small.

Figure 4.6 shows that the observer exhibits satisfactory performance in surge with no significant deviations observed in sway or yaw. It is important to mention that the platform is not subjected to environmental conditions that would induce notable motion in sway or yaw, resulting in a minimal response, and these should not be given undue emphasis.

4.3.5 Wave filter tuning, K1

In contrast to the other three gain terms, K_1 is not adjusted by simple scalar multiplication. As stated in Section 2.7, K_1 influences the wave filtering characteristics of the observer and is governed by the two parameters, ω_0 and ζ_i, representing the cutoff frequency and damping ratio, respectively. For further investigations, the observer gain values: $K_2 = 1$, $K_3 = 3$, and $K_{4,i} = 2.7$ are used. This section will present the outcomes obtained from simulations where the cutoff frequency and damping ratio have been adjusted, highlighting their impact on the system. Their initial values are given as cutoff frequency, $\omega_c = 1.22$, and damping ratio, $\zeta = 0.1$. The incoming wave has a model scale period of 1.195 seconds.

Based on the observations depicted in Figure B.7, it can be concluded that an increase in damping results in a higher presence of high-frequency content in the low-frequency estimated position. Excessive damping, with a factor of 1, renders the observer incapable of estimating the wave frequency. Consequently, the estimated low-frequency signal becomes equivalent to the incoming wave signal or the motion of the floater. This can also be deduced by Equation 2.41 for which the entire equation cancels out, resulting in zero gain. Conversely, a damping factor of 0.5 yields an inadequate velocity estimate characterized by a significant amount of high-frequency content. This suggests that the final value for damping should approach zero. Notably, the initial value of 0.1 proves to be suitable in this context.

When considering the cutoff frequency, it is observed that lower cutoff frequencies lead to improved velocity and positional estimates when only considering the indicators values in Figure 4.7. Conversely, for cutoff frequencies higher than 1.22 seconds, the opposite trend is observed. These observations are confirmed by looking at the figures Figure B.9 and B.10 which shows increasing velocity amplitudes and more high-frequent content in the positional estimates as the cutoff period increases.

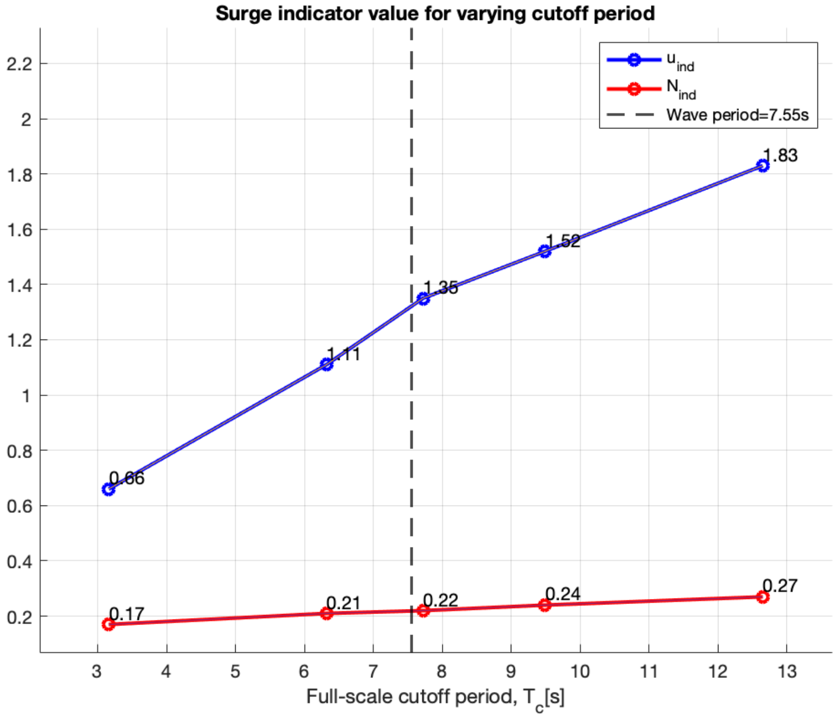


Figure 4.7: Surge indicator values for change in T_c.

Relating these observations to Equation 2.45, a general guideline is established: the cutoff frequency should always be chosen to be greater than the incident wave period. However, this guideline is not followed in the first two simulations, where the cutoff frequency is smaller than the incident wave period.

4.3.6 Bias smoothing, Tb

As mentioned in Section 4.3.1, the bias time constant should contribute towards smoothing the signal of the bias force. Simulations yield that setting the bias time constant to zero renders the observer unable to generate estimates. Selecting values within the range of 0.5 to 2 for the bias time constant does not yield significant differences in simulation results, except for the default value of 1, which significantly improves the performance. This value is the only value that ensures convergence of the bias force which can be seen in Figure B.11. Figure B.12 also provides velocity estimates of smaller amplitudes when compared to the other responses, which is a desired effect. The significant improvement of having $T_b = 1$ is reflected in the Table 4.7 which shows indicator values in surge for various values of T_b.

Indicator values in surge			
$\mathbf{T_b}$	**Bias,** b_1	**Velocity,** u	**Position,** N
0.5	1.32	1.25	0.22
1	1.57	0.50	0.10
1.5	1.25	1.32	0.22
2	1.26	1.33	0.23

Table 4.7: Indicator values in surge change in T_b

To address the oscillations in the bias force caused by $K_3 = 10$ in Figure 4.3, simulations were conducted with varying T_b to assess whether the oscillations could be reduced or smoothed out as expected. However, the response in Figure B.13 shows that adjusting T_b does not bring about substantial changes in this case either. A general tuning guideline can be derived from Equation 2.45, which states that $\frac{1}{T_i}$ should be less than $\frac{K_3}{K_4}$ or, in this case, T_i should be greater than or equal to 1. As of now $T_b = 1$ provides the most promising result and is therefore kept.

4.4 Wave peak period assumed by observer, $T_{p,obs}$

A key feature of the observer is the parameter, $T_{p,obs}$ which is the assumed peak period of the incoming wave. The presented results are obtained from simulations in which the observer's estimated peak period is varied while keeping a constant peak period of the incident wave constant. The full-scale wave period is 7.55 seconds, corresponding to 1.195 seconds in model scale. Notably, setting the observer's peak period to zero renders it incapable of generating accurate estimates. Most of the indicator values remain consistent across the different peak periods, but the velocity estimate demonstrates the most notable improvement with increasing observer peak periods.

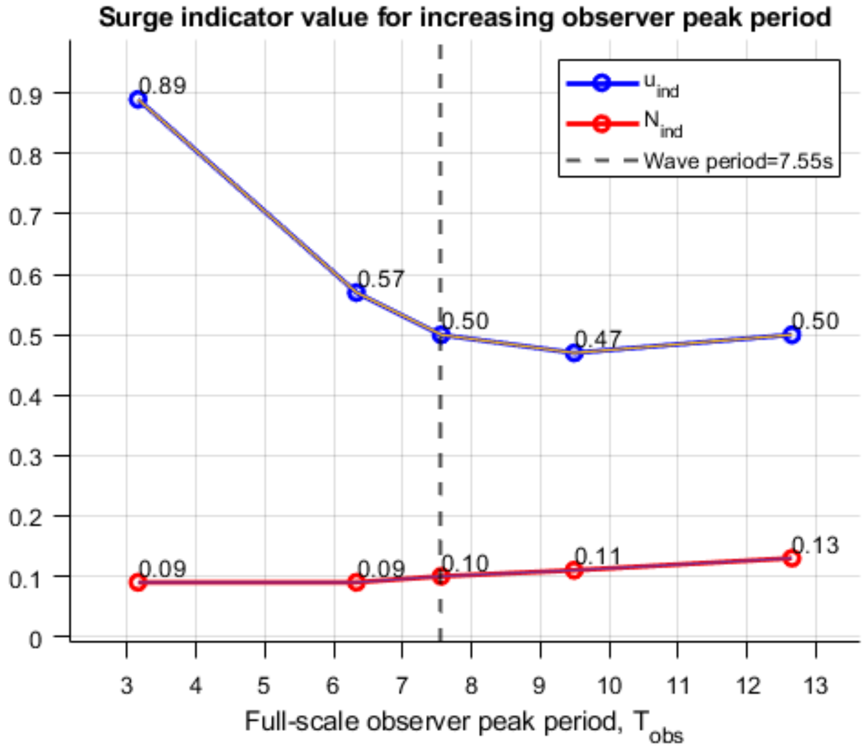


Figure 4.8: Surge indicator values for change in $T_{p,obs}$.

There exists an inverse relationship between the improvement of the velocity estimate and the positional estimate when adjusting the observer's peak period. When the observer's peak period is smaller than the incident wave period, the velocity estimate deteriorates, whereas a larger peak period leads to an improved velocity estimate but a worsened positional estimate. Simulation results indicate that having the observer's peak period close to, yet slightly higher than the incident wave period yields the most favorable response.

4.5 Response to different angles

Having obtained an adequate result for tuning in surge from Section 4.3.4 with the plotted response of Figure 4.6, the studies proceeded with testing the observer algorithm and model response for environmental loads acting from other directions. The following simulations have been conducted with the optimal parameters from the earlier tests. Environmental loads have been simulated for the following directions; 30, 60, 90, and 180 degrees, and will be compared to the tuned example where the environment was set to 0 degrees. All responses for all degrees of freedom can be seen in Section B.10.

The general observation made for these simulations is that the response in all degrees of freedom is satisfactory. The bias reaches its reference and the estimates for position and velocities are seemingly accurate without too much wave frequent content. The assumption that surge and sway are sufficiently similar to be fine-tuned using identical parameter adjustments appears to still be valid. One of the most intriguing findings from these simulations is the response in yaw. It is evident in all simulations that the mooring force applied in yaw exhibits higher frequency compared to the forces in surge and sway. This observation also clarifies the oscillations in the bias force for yaw, which are more pronounced than those in the other two degrees of freedom.

Sum of indicator values in surge, sway and yaw				
Simulation	**Total**	**Bias**	**Velocity**	**Position**
1 (0deg)	9.83	5.60	0.50	0.36
2 (30deg)	5.57	3.16	0.48	0.34
3 (60deg)	5.76	3.10	0.64	0.45
4 (90deg)	5.55	3.14	0.44	0.35
5 (180deg)	5.64	3.23	0.49	0.35

Table 4.8: Sum of indicator values in all degrees of freedom for varying environmental direction.

When considering the performance indicators, the response is generally similar across all degrees of freedom, except for the simulation involving weather from a 60-degree angle. In this particular case, the performance indicator, highlighted in red, increases by approximately 30% for both velocity and positional estimates. Figure 4.9 is the plotted response for the environment with a direction of 60 degrees. This figure shows no obvious sign as to why the response in yaw should be any poorer than for the other simulations.

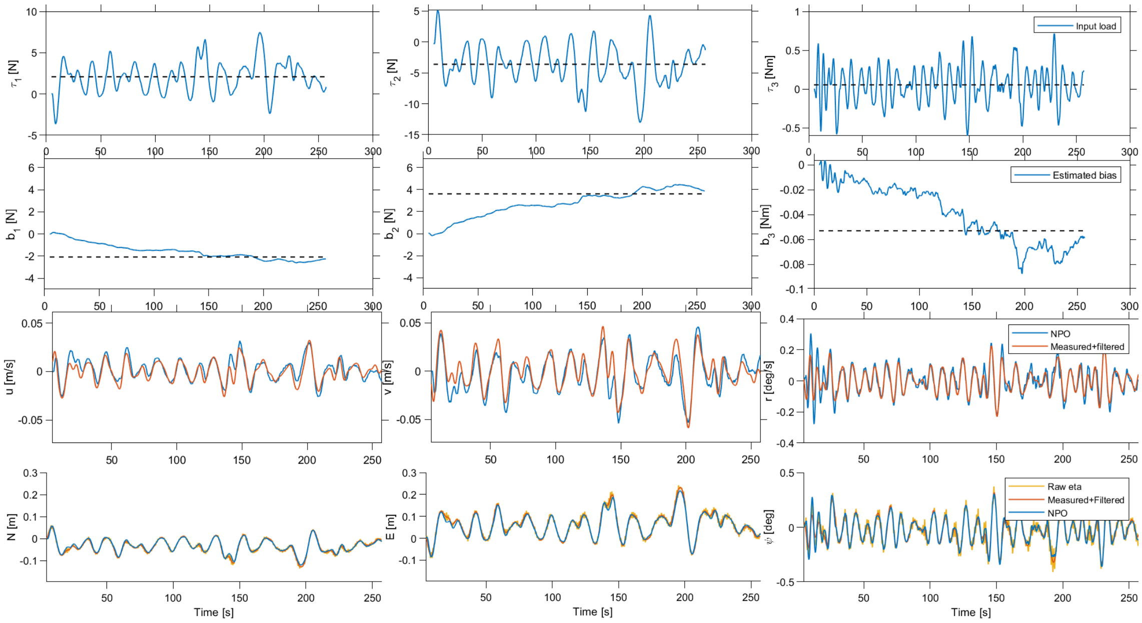


Figure 4.9: Simulation with the environment from 60 degrees.

The values for the total indicator and bias have also been recorded. These values for the initial simulation with weather from 0 degrees are inconsistent with the remaining values which fall within a similar range. It has been commented upon earlier that for some simulations, the bias indicator yields inconsistent values. The claim for reducing its significance is supported by this observation. The deviation in the total value can be attributed to an increase in the bias indicator.

A closer inspection of the values for the various degrees of freedom reveals that the larger positional indicator value for the third simulation is driven by an increase in yaw estimate. The value in yaw is 0.25 compared to the average of 0.15 from the other simulations, an increase of 66%. For the velocity, it is observed that the discrepancy is mainly caused by an increase in sway. All other values are 0.01 and the value for estimated velocity in sway, v for 60 degrees is 0.12.

The response in yaw only has been plotted for the simulations with 30, 60, and 90 degrees to get a closer look, this has been done in Figure B.15, B.16 and B.18. In these figures and especially for the simulations of 30 and 90 degrees, the velocity estimates in yaw are polluted by a high-frequent content.

4.6 Tuning yaw

In order to make up for the lacking estimates in yaw, an attempt was made in order to tune the observer utterly. For this, the case with weather coming from 60 degrees was chosen as it showed the biggest potential for improvement based on its indicator values. Initial gain values were set with respect to the result obtained for tuning in surge; $K2_i = 1$, $K3 = 3$, $K4_i = 2, 7$, $i \in (1, 2, 3)$. The values for surge and sway will remain constant and the values in yaw will be changed to assess their effect.

The process of tuning was initiated by adjusting the velocity injection term K_4 since K_3 already had been adjusted. The results are portrayed in Figure B.20 which shows that an increase in K_4 gives a faster convergence of the bias force at the cost of a more high-frequent signal. Figure B.21 shows a decrease in amplitude of the velocity estimates when increasing K_4, also in this case an increased presence of high-frequent content in the signal. Decreased amplitudes are in general a desired effect, as the simulations for change in the direction of the environment revealed velocity estimates with too large amplitudes. This effect is also reflected in the following table where the third, fourth, and fifth lines show an increase in velocity estimates for increasing K_4

Indicator values in yaw				
$\mathbf{K_{2,3}}$	$\mathbf{K_{4,3}}$	**Bias,** b_3	**Velocity,** r	**Position,** ψ
1	0.5	1.55	0.69	0.29
1	1	1.55	0.52	0.24
1	2.7	1.56	0.46	0.25
1	5	1.55	0.59	0.24
1	10	1.59	0.63	0.20
0.5	2.7	1.59	0.76	0.39
1	2.7	1.56	0.46	0.25
5	2.7	1.56	0.61	0.19
10	2.7	1.53	0.87	0.23

Table 4.9: Indicator values in yaw for varying values of $K_{2,3}$ and $K_{4,3}$.

Changing the value of K_2 in yaw does not reveal much change in the estimates. The bias force in Figure B.22 is more or less unaffected and estimated velocity in Figure B.23 shows slightly more high-frequent oscillations for increased K_2.Table 4.9 reveals a decrease in position indicator and an increase in velocity estimates, as was seen when tuning in surge.

4.6.1 Discussion

As discussed earlier, increasing a gain is likely to improve the performance and estimates of a state in the observer, up until a point where the signal gets corrupted by higher frequencies which comes from the wave-frequent motions. Increasing K_4 in yaw gives a faster convergence of bias and smaller amplitudes in velocity estimates. These are both desired effects, as the simulations with change in environment revealed that the bias in yaw converged slowly and the amplitudes of velocity estimates were too large. Also, in this case, the biggest effect on overall performance is achieved by adjusting K_4. It should be noted that a change in K_4 also affects the performance in surge and sway, this yields a duality and trade-off in which one has to decide which degree of freedom is of the biggest importance. In this case, surge and sway have been deemed the most important and it has been decided not to change the values of gains in yaw.

One drawback of adjusting K_4, in this case, is that the relation between K_3 and K_4 in yaw will be affected. If one wishes to maintain the same ratio as for surge and sway, the value of $K_{3,3}$ should be adjusted in line with K_4. This will require an update to the observer FMU which enables the change of K_3 with respect to the degree of freedom.

4.7 Performance when exposed to different sea-states

With the tuned observer, it is now desired to test how it responds to a change in wave state and whether this will affect the performance of the algorithm. A central question that should be answered is whether the cutoff frequency and observer peak period should be adjusted with respect to the wave and whether adaptive control is necessary for optimal performance. This is done by testing for various sea states.

Initial simulation parameters give an estimated peak period of the observer set to 1.195 seconds. This corresponds to a full-scale wave period of 7.558 seconds. The incoming wave in the simulation has a peak period of 7.55. The difference is not huge, but the observer peak period is put to be slightly larger than the incoming wave.

$$\frac{Tp_{obs}}{Tp_{wave}} = \frac{7.56}{7.55} = 1.001 \tag{4.2}$$

When it comes to the cutoff frequency given by the observer parameter; "cutOffWFObs" is initially set to 1.22 seconds, corresponding to a period of 7.716 seconds in full scale, denoted T_c. Stating the relation between cutoff and incoming wave period we have a percentage difference of 2.3%.

$$\frac{T_{cutoff}}{Tp_{wave}} = \frac{7.72}{7.55} = 1.023 \tag{4.3}$$

In order to test the observer under various wave conditions, three pairs of significant wave height and peak period have been chosen from Table 3.3 and 3.4 in Section 3.2.4. The first two sets have been tested for two conditions, while the third and fourth have been simulated with four different conditions:

<table>
<tr><th colspan="6"><b>Indicator values:</b> $H_s = 3.75m$, $T_p = 7s$ <b>and</b> $\gamma = 0.076$</th></tr>
<tr><th>Simulation</th><th>T_p [s]</th><th>T_c [s]</th><th>Bias, b_1</th><th>Velocity, u</th><th>Position, N</th></tr>
<tr><td>1</td><td>7.56</td><td>7.72</td><td>1.57</td><td>0.50</td><td>0.10</td></tr>
<tr><td>2</td><td>3.80</td><td>7.72</td><td>1.59</td><td>0.61</td><td>0.09</td></tr>
<tr><td colspan="6"><b>Indicator values:</b> $H_s = 6.19m$, $T_p = 9s$ <b>and</b> $\gamma = 0.076$</td></tr>
<tr><td>1</td><td>7.56</td><td>7.72</td><td>1.59</td><td>0.47</td><td>0.09</td></tr>
<tr><td>2</td><td>7.56</td><td>9.03</td><td>1.58</td><td>0.52</td><td>0.11</td></tr>
<tr><td colspan="6"><b>Indicator values:</b> $H_s = 14.97m$, $T_p = 14s$ <b>and</b> $\gamma = 0.076$</td></tr>
<tr><td>1</td><td>7.56</td><td>7.72</td><td>1.57</td><td>0.96</td><td>0.27</td></tr>
<tr><td>2</td><td>7.56</td><td>14.33</td><td>1.58</td><td>0.90</td><td>0.27</td></tr>
<tr><td>3</td><td>14.99</td><td>14.33</td><td>1.60</td><td>0.72</td><td>0.21</td></tr>
<tr><td>4</td><td>14.99</td><td>7.72</td><td>1.60</td><td>0.80</td><td>0.21</td></tr>
<tr><td colspan="6"><b>Indicator values:</b> $H_s = 6.19m$, $T_p = 12s$ <b>and</b> $\gamma = 0.043$</td></tr>
<tr><td>1</td><td>7.56</td><td>7.72</td><td>1.58</td><td>0.64</td><td>0.21</td></tr>
<tr><td>2</td><td>7.56</td><td>12.23</td><td>1.59</td><td>0.62</td><td>0.20</td></tr>
<tr><td>3</td><td>12.02</td><td>12.23</td><td>1.61</td><td>0.52</td><td>0.18</td></tr>
<tr><td>4</td><td>12.02</td><td>7.72</td><td>1.60</td><td>0.56</td><td>0.18</td></tr>
</table>

Table 4.10: Indicator values in surge for different sea-states.

The first set of simulations has been tested for two different values of the assumed peak period in the observer. The simulated wave has a smaller period than the assumed peak period in the observer, the test aims to investigate the effect of having an estimated peak period which is larger than the incoming wave.

As can be seen in Table 4.10, the observer performs better with an overly large value of estimated peak period compared to the incident wave. The change has a visual effect in the estimate of velocity in Figure B.24 where the second simulation has significantly more high-frequent content. A low estimate of the peak period in the observer gives more wave-frequent content in the estimates.

In the second set, the incoming wave has a larger period than the estimate in the observer, this test was then conducted to see the effect of an increased cutoff period in the observer. From the performance indicator, there is only a small difference in values between the two simulations with a slight increase in the second case.

For the two last simulation sets, a combination of the tests from the two first sets has been conducted. The initial simulation is used as a basis for comparison, the second with an increased peak period, the third with an updated cutoff period, and a combination where both parameters are adjusted with respect to the incoming wave in the fourth simulation.

When only considering the indicator value from these two simulation sets, it is clear that the performance in surge draws an advantage of increasing the cutoff period. Both the velocity and positional estimates get a reduced indicator value for the third and fourth simulations when compared to the first and second. There is also a slight improvement when increasing the estimated peak period in both cases. The most prominent simulation is the one where both the cutoff and peak period have been adjusted.

Both the following figure and Figure B.27 show a sliced plot for the estimated position for all four simulations in respectively the fourth and third set. The figures give a visual impression that in simulations 3 and 4, where the peak period has been adjusted, the amplitudes of the wave frequency content are smaller than for simulations 1 and 2.

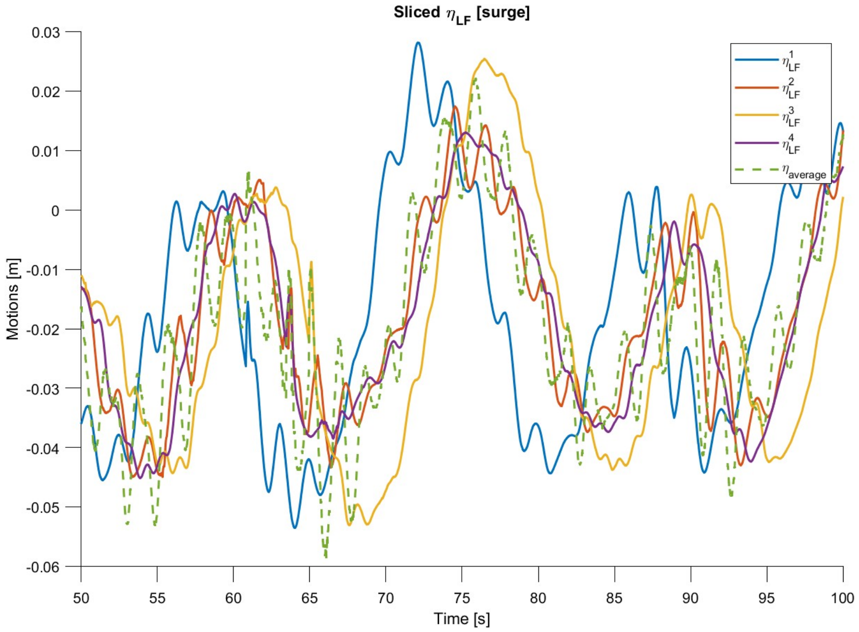


Figure 4.10: Sliced plot of $\hat{\eta}$ for $H_s = 6.19$ and $T_p = 12$.

4.7.1 Discussion of changing wave periods

Based on the analysis of the two larger sea states depicted in figures 4.10 and B.27, it is evident that the yellow and purple lines representing the last two simulations show a reduced presence of wave-frequency content in the estimated low-frequency position compared to the first two simulations. This observation suggests that adjusting both peak- and cutoff periods provides smoother estimates of the low-frequent position with less wave-frequent content. Tuning the observer parameters based on the incoming wave period offers significant benefits to the observer. One advantage when conducting physical experiments is that the wave period will always be known, as it is assigned to the wave-maker for each case and can therefore be appropriately adjusted in the observer.

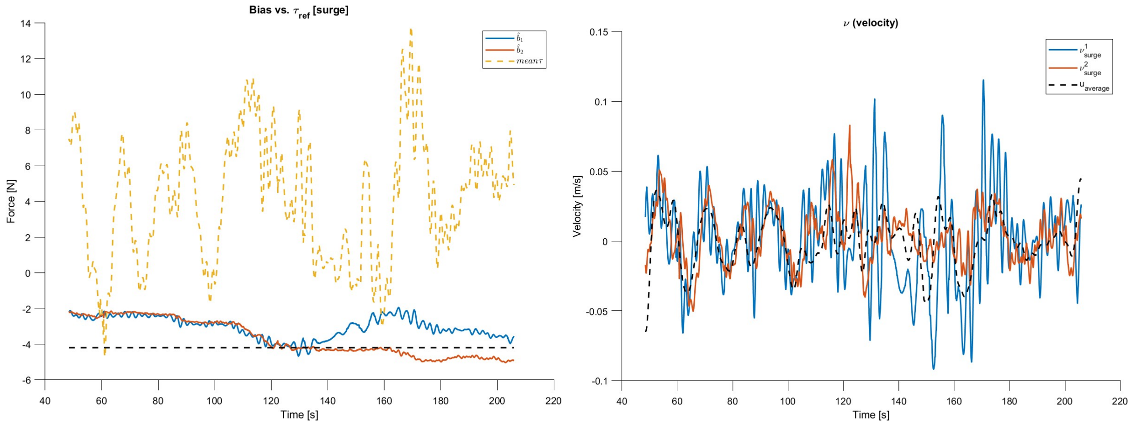
Even though the wave period is known for experiments, a suggested rule of thumb is to err on the side of caution by using conservative estimates for both peak and cutoff periods, rather than risking an estimate that proves to be too low. Implementing a safety factor is recommended as a precautionary measure to ensure this.

4.8 Increasing H_s and T_p

According to the documentation for the model of INO WINDMOOR 12MW, the model does not account for the influence of viscous excitation. This can impact the heave, roll, and pitch responses, particularly during high sea states close to the platform's natural period of oscillation, equal to 15.2 seconds. Consequently, the second-order excitation loads may be affected (Souza 2021).

It is intriguing to examine whether this omission has any discernible effects. To facilitate a comparative analysis with the earlier simulations, the sea state under consideration is adjusted to have a significant wave height that yields the same steepness as the previous simulation with $\gamma = 0.076$, for $T_p = 15.2$ seconds. This adjustment results in a significant wave height, $H_s = 17.56$ meters.

The first simulation is run with the tuned set of observer parameters, not adjusting the estimated peak period in the observer and the cutoff frequency. For the second simulation, both the cutoff period and the estimated peak period have been adjusted to full-scale values of $T_{p,obs} = 15.21$ and $T_c = 15.55$ seconds. The response from these two tests can be seen and compared in the following figures:



(a) Bias plotted for simulations with platform cancellation period.

(b) $\hat{\nu}$ for simulations with platform cancellation period.

Figure 4.11: Bias and velocity estimates plotted for simulations with platform cancellation period.

Figure 4.11a shows that correctly fitted and adapted variables for a new sea state provide the better result, as default parameters (blue line) contain more wave-frequent content and strains further from the reference after first finding it. Figure 4.11b also shows that the velocity estimates are better for updated parameters where the orange curve shows less high-frequency oscillations and in general smaller amplitudes of the estimated velocity.

4.9 Discussion

The title of this thesis is:

Software in the loop testing of a wave filter:
with application to cyber-physical testing of a floating wind turbine

This section will discuss and elaborate on the main findings from this chapter with respect to the title and objective of the thesis. It also aims to answer the feasibility of this setup for experimental usage, in addition to summarizing the key elements before drawing any conclusions.

4.9.1 General remarks

The two previous chapters present the proposed method when adapting an NPO algorithm to a SIL setup using HLCC and SIMA, as well as the results obtained when applying this method and conducting simulation experiments.

Based on the time spent and the attention devoted to this thesis, it becomes evident that much of this experimentation depends on a good model and a correct setup. Uncertainties and modeling errors will cause difficulties when trying to fit a model-based estimator to the experimental setup and the subsequent results will be erroneous.

Caution is advised when fitting model parameters to a simulated response using ODE, as parameters can be customized to the response in a way that will yield a nonphysical result. Despite wrong parameters, the observer will in most cases be stable and provide credible results. This can quickly cause misunderstandings and propagating errors throughout the setup once implemented that will be hard to trace back to their origin.

Once having ensured that all parameters were fitted correctly and held correct physical values with respect to the simulation model, the observer was able to separate the wave frequencies out of the signal and generate a low-frequent estimate in all cases. What separated a good from a poor result was the amount of wave frequencies filtered out and with how much accuracy.

Tuning the observer proves to be a relatively straightforward procedure once the model parameters are estimated correctly. Models with more uncertainty in parameters can also be tuned but will require higher gains which yields less trust in the model. For an acceptable response, the gains do not need much fine-tuning and a few simulations with various values for the respective gain term should be enough to provide an adequate response. The trick of tuning is to strike the balance where gains have been tuned to the extent of providing accurate estimates without any higher frequencies as a larger gain also gives more wave-frequencies in the signal. This is observed for all the simulations. This thesis has not gone through the tedious process of finding the perfect balance as this would require more time and tests.

The proposed method for tuning and its order of changing the different constants in this thesis seems to work. Primarily tuning the bias term to reach its reference without too much wave frequent signal, then the velocity injection term up until it provides estimates which are close enough to the derived velocity before finally adjusting the position injection term, K_2. K_4 is proven to have a larger effect on the system as it is more related to the system equations from Equation 2.6.1 and should therefore be tuned before K_2. This procedure also makes it easier to fulfill the equation proposed by Fossen in Equation 2.45.

For a symmetric construction like this, the tuning in surge should provide a satisfying result for sway as well. However, time should be devoted to tuning the algorithm specifically in yaw in order to obtain better performance in this degree of freedom. Despite not applying more gain in yaw, the observer responds well to changes in the direction of the environment and sea state. It is noted that the performance is worst for nonaxial directions calling for the structure to be moored facing the most probable direction of waves. For an optimal solution, the parameters should be adjusted based on the incoming wave period.

When testing the observer algorithm for extreme conditions, it does not indicate to struggle.

Overall it can be stated that the proposed method and setup works as a platform for evaluating the response of an offshore floating wind turbine preliminary to basin tests. In order to evaluate whether it is effective or not, the data should be evaluated against data from basin tests. As the simulated model is a simplification of reality and a real model is likely to encapsulate more dynamic effects some deviations are expected. For this to become a recommended practice, effective model design and verification is a prerequisite.

4.10 Sources of error

Over the course of this study, some sources of error have been detected, these are listed and briefly discussed in the following:

- Lacking documentation of model parameters and differences between the simulated model and the report Souza (2021) deems it difficult to decide which parameters are correct. This thesis has pursued to be consistent in only using values from the simulated model.
- Restricted computational power sometimes results in delays and signal jitter directly affecting the response of the system.

- Direct comparison of time series from different simulations is advised to be done with care as most are not perfectly aligned and what seems like a big difference in a plot might be caused by for example delay in one simulation.
- Lag in the loop and delay between FMUs. When comparing the time series of a parameter at different intervals in the loop one can see a slight shift in the signal as in Figure B.28.
- Human error. Over the course of performing more than 200 simulations, having a lot of adjustable parameters, it is worth noting that human error can occur which could have affected some results.

Chapter 5

Conclusions

The main goal of this thesis was to:

verify the wave filtering ability of a nonlinear passive observer for wind turbine application before physical basins tests by using cyber-physical principles applied to software-in-the-loop testing

This chapter concludes the work done in this thesis and presents some suggestions for further work.

5.1 Concluding remarks

This thesis focuses on the modeling and application of a wave filter for a floating offshore wind turbine. The aim has been to accurately estimate the low-frequent motions and velocities from measurements. The simulation model is based on the INO WINDMOOR 12MW concept design. The wave filter is a Nonlinear Passive Observer algorithm based on the theory by Thor I. Fossen. In addition, the feasibility of testing models using general software in the loop testing setup using the tools SIMA and HLCC has been investigated.

A software-in-the-loop testing setup was developed using cyber-physical principles, simulating the floating platform model in SIMA as the physical substructure and having virtual mooring as the numerical substructure. Virtual mooring was applied using modeled actuator lines in SIMA, controlled by forces calculated in HLCC. The setup and model have been verified through comparison using simulated decay tests and the modeled response of a mass-spring-damper system solved by ordinary differential equations in MATLAB.

The nonlinear passive observer has been presented and is modeled in Simulink before exporting it as an FMU for use in HLCC. By adjusting the algorithm, it is able to filter the incoming wave signals and estimate the low-frequency position and velocity of the floater accurately. Moreover, the bias force aligns with its reference value. Comparing the output from the observer with a low-pass filtered position and the derived velocity grants a measure of performance for the algorithm. The algorithm has been tested for several scenarios with changing environmental loads. Results show that the system is able to separate the wave frequent content in measured position from the low-frequent component in most cases.

For optimal performance, it is recommended to employ adaptive control that considers the incoming wave characteristics. This allows the system to adapt to changes in the wave conditions, resulting in improved performance. Obtaining good results requires a solid understanding of the system dynamics, familiarity with the mathematical model, and accurate estimation of model parameters.

To conclude, the suggested SIL setup proves suitable for experimental and verification testing of the simplified model for INO WINDMOOR 12MW. The low frequent position and velocity can be estimated using the suggested NPO.

5.2 Further work

Below are some bullet points for suggested further work:

Modeling

- The model used in this thesis only holds the floating platform, it would be interesting to develop a full model equipped with a tower and turbine for testing.
- Develop model with more floating substructures and shared mooring for testing.

Simulation and analysis

- The model can be tested for more varying conditions and even harsher environments from more angles to see if this has an effect. SIMA also allows the introduction of currents and wind in the model.
- Quite a substantial amount of data has been collected from all simulations. It would be interesting to apply statistical analysis with the ordinary least squares (OLS) method to see if any other or a more optimal solution could be found.
- Further research into whether the bias indicator should be updated to provide more accurate and representative performance computations. This should be done to penalize non-convergence towards reference value and for strong oscillations.
- Calculate the nonlinear forces acting on the structure based on the low-frequent estimates, as done in T. Sauder (2021).

Setup and control

- A lot of time has been spent dealing with the magnitudes of signals in HLCC. In order to make it more user-friendly and intuitive, the implementation of signal units in HLCC could contribute to avoiding confusion and errors in the future.

- A functionality which allows start and reload (with or without parameter changes) for all DLLs at the same time would be helpful in order to increase efficiency when testing with multiple modules

References

Abbiati, Giuseppe (2022). *Cyber-physical testing: beyond marine technology.*

Atwood, Bruce (Mar. 2011). *Lipschitz Continuity.* URL: https://demonstrations.wolfram.com/LipschitzContinuity/. (accessed: 06.05.2023).

Balchen, Jens G (2016). *Reguleringsteknikk.* nob. Trondheim.

Chen, Chi-tsong (2013). *Linear system theory and design.* Oxford University Press. ISBN: 978-0-19-996454-3.

Committee, ETIPWind Executive (2020). *FLOATING OFFSHORE WIND DELIVERING CLIMATE NEUTRALITY.*

DNV (June 2021). *ST-0119 floating wind turbine structures.* URL: https://www.dnv.com/energy/standards-guidelines/dnv-st-0119-floating-wind-turbine-structures.html.

— (2022). *Energy Transition Outlook 2022.*

Eliassen, Lene (n.d.). *Arealbruk i havvind.* URL: https://blogg.sintef.no/sintefocean-nb/arealbruk-i-havvind/.

Faltinsen, Odd M (1990). *Sea loads on ships and offshore structures.* eng. Cambridge.

Farm, Dogger Bank Wind (n.d.). *Building the world's largest offshore wind farm.* URL: https://doggerbank.com/. (accessed: 14.04.2023).

Fossen, Thor I. (1991). *Nonlinear modelling and control of underwater vehicles.* Universitetet i Trondheim, Norges tekniske høgskole, Institutt for teknisk kybernetikk. ISBN: 8271192949.

— (1994). *Guidance and Control of Ocean Vehicles.* Wiley. ISBN: 0471941131.

— (1998). 'Passive nonlinear observer design for ships using Lyapunov methods: full-scale experiments with a supply vessel'. In: DOI: http://dx.doi.org/10.1016/S0005-1098(98)00121-6.

— (2021). *Handbook of marine craft hydrodynamics and motion control.* Wiley. ISBN: 9781119575054.

Functional Mock-up Interface (FMI) Standard website (n.d.). https://fmi-standard.org/.

Hossain, Eklas et al. (Nov. 2017). 'Sliding Mode Controller and Lyapunov Redesign Controller to Improve Microgrid Stability: A Comparative Analysis with CPL Power Variation'. In: *Energies* 10, pp. 1–24.

Inocean (n.d.). *INO 12MW Floating Offshore Wind Turbine.* URL: https://www.inocean.no/news/more-efficient-design-of-floating-wind-turbine-farms/. (accessed: 06.05.2023).

Kallehave, Dan et al. (2015). ‘Optimization of monopiles for offshore wind turbines’. eng. In: *Philosophical topics* 373.2035, p. 20140100. ISSN: 1364-503X.

Khalil, Hassan K. (2015a). *Lecture notes in Nonlinear Systems.* URL: https://www.egr.msu.edu/~khalil/NonlinearControl/Slides-Full/Lect_1.pdf.

— (2015b). *Nonlinear Control.* Pearson. ISBN: 1-292-06050-6.

Marius F. Aarset Jann Peter Strand, Thor I. Fossen (1998). ‘Nonlinear vectorial observer backstepping with integral action and wave filtering for ships’. In: DOI: https://doi.org/10.1016/S1474-6670(17)38420-3.

Mathworks (n.d.). *About Passivity and Passivity Indices.* URL: https://se.mathworks.com/help/control/ug/about-passivity-and-passivity-indices.html. (accessed: 06.11.2022).

Microsoft Learn - Troubleshoot Windows Client: Dynamic Link Library (n.d.). https://learn.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library.

Molin, Bernard (2023). *Offshore structure hydrodynamics.* eng. Cambridge.

Nakashima, Masayoshi (2020). ‘Hybrid simulation: An early history’. eng. In: *Earthquake engineering structural dynamics* 49.10, pp. 949–962. ISSN: 0098-8847.

Nations, United (n.d.). *Goal 13: Take urgent action to combat climate change and its impacts.* URL: https://www.un.org/sustainabledevelopment/climate-change/. (accessed: 05.11.2022).

Orsted (n.d.). *1991-2001 The first offshore wind farms.* URL: https://orsted.com/en/insights/white-papers/making-green-energy-affordable/1991-to-2001-the-first-offshore-wind-farms. (accessed: 14.04.2023).

Sauder, Thomas (2021). ‘Empirical estimation of low-frequency nonlinear hydrodynamic loads on moored structures’. In: DOI: http://dx.doi.org/10.1016/j.apor.2021.102895.

— (2022a). *TMR18 - Cyber-physical testing in marine technology: Artefacts, stability and fidelity.*

— (2022b). *TMR18 - Cyber-physical testing in marine technology: Estimation of nonlinear hydrodynamic loads on moored structures.*

— (2022c). *TMR18 - Cyber-physical testing in marine technology: Main concepts in cyber-physical testing.*

Sauder, Thomas Michel (n.d.). *CYBERLAB.* URL: https://www.sintef.no/prosjekter/2022/cyberlab/.

SINTEF (n.d.). *Advanced Wave and Wind Load Models for Floating Wind Turbine Mooring System Design.* URL: https://www.sintef.no/projectweb/windmoor/. (accessed: 20.05.2023).

Skjetne, Roger and Olav Egeland (Oct. 2005). ‘Hardware-in-the-loop testing of marine control systems’. In.

Sorensen, Asgeir J. (2018). ‘Marine Cybernetics: Towards Autonomous Marine Operations and Systems - Lecture Notes’. In.

Souza, Carlos Eduardo Silva de (2021). *Definition of the INO WINDMOOR 12 MW base case floating wind turbine.* URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2772738.

Steen, Sverre (2014). *Lecture Notes TMR7: Experimental Methods in Marine Hydrodynamics.*

Takanashi, Koichi (1975). *Non-linear Earthquake Response Analysis of Structures by a Computer-actuator On-line System (detail of the System).*

WindEurope (2020). *Offshore Wind in Europe: Key Trends and Statistics 2020.* https://windeurope.org/intelligence-platform/product/offshore-wind-in-europe-key-trends-and-statistics-2020/. Accessed: June 4, 2023.

Appendix

Appendix **A**

Appendix A: Relevant Theory

A.1 Added mass

In fluid dynamics, added mass refers to the apparent increase in the mass of an object caused by its acceleration through a fluid. The motion causes disturbances in the surrounding fluid causing a force acting on the floating object that appears as if it is an increase in the mass of the object. This force is referred to as added mass and is dependent on the frequency of the motion of the object through the fluid.

We differentiate between two phenomena called zero- and infinite-added mass. Zero frequency added mass, A_0 occurs for larger periods of motion. For lower periods, close to the infinite frequency, the added mass, A_∞ approaches a constant value. At high frequencies, the fluid is unable to adjust to the motion of the object quickly enough, and the added mass becomes independent of the frequency of the motion. The main difference between zero and infinite frequency added mass is that the former depends on the frequency of the object's motion through the fluid, while the latter does not (Chapter 3 in Faltinsen 1990 or Molin 2023 for more in-depth information). For this application, we are mostly interested in the zero-frequency as it is assumed that the response of the body will be in the larger periods, up towards 100 seconds.

A.2 Damping

Another relevant hydrodynamic phenomenon is damping; the process of reducing the amplitude of motion of a body that is undergoing oscillations in a fluid. Damping is important for controlling the motion of the body and ensuring its stability in the fluid. There are two main categories of damping that can occur for a floating body in waves: linear damping and quadratic damping.

Linear damping is a type of damping where the damping force is proportional to the velocity of the body. The main component of linear damping is radiation damping, caused by waves radiated by the body. Quadratic damping is proportional to the square of the velocity of the body. The main component of quadratic damping for a floating body in waves is viscous-quadratic damping occurring due to the friction between the body and the fluid. Chapter 3 in Faltinsen (1990) or Molin (2023) for more in-depth information.

The quadratic damping terms for a floating body can be calculated based on the equation for effective drag from the Morison equation, given as:

$$F_D = \frac{1}{2} C_D v|v| A \tag{A.1}$$

Where the quadratic damping force is a result of the term: $C_D v|v|$. The equivalent equation for calculating the force in the surge for INO WINDMOOR 12MW is found to be:

$$C_D^{effective} v|v| = C_D^P v|v| L_P + 2(C_D^P v\cos(60°)|v\cos(60°)|\cos(60°))L_P + 3C_D^C L_C \tag{A.2a}$$

$$C_D^{effective} = C_D^P L_P + \frac{1}{4} C_D^P L_P + 3C_D^C L_C \tag{A.2b}$$

$$C_D^{effective} v|v| = \frac{5}{4} C_D^P L_P + 3C_D^C L_C \tag{A.2c}$$

See appendix for illustration (Figure A.3)

A.3 Reference frame

The general dynamics of a system are usually described in two parts; kinematics and kinetics. These are respectively describing the motions and the forcing of the system. In order to describe the kinetics of a system, a kinematic base should be first established.

The orientation of a marine craft is usually described using six independent coordinates, each representing one degree of freedom for the structure. In order to give an accurate description of the orientation, a reference must be established. Different references are usually referred to as reference frames. For marine applications, the two most commonly used are the North-East-Down (NED) frame and the BODY frame (Sorensen 2018).

The north-east-down frame uses a tangent plane to the Earth's reference ellipsoid as the origin, and the position is noted as $\{n\} = (x_n, y_n, z_n)$. The x_n axis point to the true north, the y_n axis towards the east, and the z_n axis are normal to the earth's surface, pointing towards the center of the earth (Fossen 2021).

The body frame is fixed to a virtual center of the marine craft, often represented as the origin of the guidance, navigation, and motion control systems of the craft or coinciding with the point midships along the waterline. The reference frame is described as $\{b\} = (x_b, y_b, z_b)$. where x_n is defined along the longitudinal direction of the craft, y_b towards the transverse axis, and z_b normal to the craft, pointing downward to the sea surface (Fossen 2021).

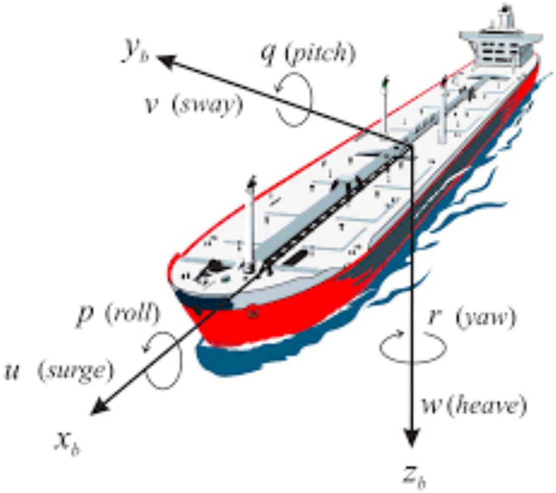


Figure A.1: Degrees of freedom for a ship(Fossen 2021).

The position of a marine craft is often denoted using the NED frame, while the orientation is defined in terms of Euler angles on the body frame with respect to the NED frame. The generalized position vector can be written as:

$$\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^\top \tag{A.3}$$

and the generalized velocity vector in the body frame is then given as:

$$\boldsymbol{\nu} = [u, v, w, p, q, r]^\top \tag{A.4}$$

The motion vector which is the combination of the generalized position and velocity vector is then written as:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{p}^n_{nb} \\ \boldsymbol{\theta}^n_{nb} \end{bmatrix} = \begin{bmatrix} x^n \\ y^n \\ z^n \\ \phi \\ \theta \\ \psi \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{v}^b_{nb} \\ \boldsymbol{w}^b_{nb} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{A.5}$$

It is important to be able to express one set of variables given in one reference frame in terms of another. Transformations between the two should therefore be established. The Euler angle transformation matrix for angles in the body frame can be related to the velocities in the NED frame using the following rotational matrix which is given in (Fossen 2021):

$$\boldsymbol{R}_b^n = \boldsymbol{R}(\theta_{nb}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi} \tag{A.6}$$

the inverse transformation is performed by using the transpose of Equation A.6. For a detailed description and derivation of the matrices, consult chapter 2 in (Fossen 2021).

A similar transformation can be done for the angular velocities, in this case, the transformation equation is stated as the following, also given in (Fossen 2021):

$$\dot{\boldsymbol{\theta}}_{nb} = \boldsymbol{T}(\boldsymbol{\theta}_{nb})\boldsymbol{w}_{nb}^b \tag{A.7}$$

When working with the full 6-DOF kinematic equations, the transformation is usually written in the form:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{A.8}$$

where $\boldsymbol{J}_\theta$ contains both the transformation matrix for linear and angular equations:

$$\boldsymbol{J}_\theta = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\theta}_{nb}) & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{T}(\boldsymbol{\theta}_{nb}) \end{bmatrix} \tag{A.9}$$

Assuming small angles simplifies the 3-DOF model, and allows for the following simplified transformation relation to be stated (Fossen 2021).

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{A.10}$$

A.4 Model scaling

This report will present results generated by simulations for a model of a full-scale floating wind turbine. Therefore, it is relevant to present some basic theories behind hydrodynamic scaling laws.

A challenge when designing and modeling objects for marine applications is the large variation in environmental conditions, i.e., wind and waves. To achieve similarity, by forcing a model and its full-size equivalent, there are three conditions that should be met. The full scale and the testing model should have the following:

- Geometric similarity
- Dynamic similarity

Parameter	Unit	Scaling factor
Length	[m]	λ
Time	[s]	$\sqrt{\lambda}$
Angular displacement	[deg]	$\lambda^0 = 1$
Structural mass	[kg]	$\lambda^3 \rho_f/\rho_m$
Force	[N]	$\lambda^3 \rho_f/\rho_m$
Moment	[Nm]	$\lambda^4 \rho_f/\rho_m$
Damping	[Ns/m]	$\lambda^{2.5} \rho_f/\rho_m$

Table A.1: Scaling conversion factors.

- Kinematic similarity

Having both geometric and dynamic similarity usually implies kinematic similarity as well. The most typical scaling rule applied to marine operations is Froude-scaling which ensures similarity in inertia and gravitational forces. This ensures that most of the characteristics of the wave-related full-scale phenomena are maintained.

The equation for the Froude number can be expressed as

$$F_n = \frac{U}{\sqrt{gL}} \tag{A.11}$$

The scaling factor between a full-scale model and a model can be further derived as:

$$\frac{V_m}{V_F} = \frac{1}{\sqrt{\lambda}} \tag{A.12}$$

Where λ is a scalar.

The fluid surrounding the body also introduces viscous effects; an equal ratio between the inertia and viscous forces acting on a body can be expressed using the Reynolds number.

$$Re = \frac{UL}{\nu} \tag{A.13}$$

Scaling a model based on both the Reynold and Froude numbers of the full-scale system is desirable. Nevertheless, computing the velocities of the model at the scale of $\lambda = 1$ is impractical. Consequently, hydrodynamic scaling has predominantly relied on Froude-scaling as the prevailing approach over the years.

Applying the same scaling rules to other relevant quantities gives the following relations

ρ_f and ρ_m describe the density of the fluid surrounding the full-scale structure and the model. For most marine applications, ρ_f will correspond to the density of salt water and ρ_m to fresh water, such that $\rho_f/\rho_m = 1.025$.

A.5 Controllability

Observability can also be related to another term; Controllability. The theorem of duality states that if the pair of system matrices $\{\boldsymbol{A}, \boldsymbol{B}\}$ is controllable if and only if the pair $\{\boldsymbol{A}^\top, \boldsymbol{B}^\top\}$ is observable (Chen 2013). The physical interpretation of a controllable system is that, for a given input, the system is able to move any state in the state space to another state within a finite amount of time. The magnitude of the input is not bounded and can be as large as desired or possible (Chen 2013).

Mathematically, the controllability matrix can be calculated as:

$$\mathcal{C} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & ... & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \tag{A.14}$$

and the system will be controllable if the matrix, $\mathcal{C}$ has full row rank. A system without full-row rank has redundancy in its inputs (Chen 2013).

A.6 Sampling and the Nyquist frequency

This thesis deals with sampled analog signals that serve as inputs to an algorithm. Some signal processing theory is therefore needed and the sampling theorem derived by Nyquist is fundamental when working with digital systems and signals.

The theorem states that in order to reproduce an input signal digitally, the sampling frequency should be at least twice the highest frequency contained in the original signal in order to keep its main characteristics (Balchen 2016).

Nyquist also derived a theorem for the stability of dynamical systems that can be interpreted visually. Without going into too much detail, the simplified theorem states that a system that is open-loop stable is also stable if the Nyquist curve does not encircle the point $(-1 + j0)$ (Balchen 2016).

A.7 Lipschitz continuity

Lipschitz continuity is used to describe the form and steepness of a curve, its main characteristic being that it limits the derivative of a curve. A function f is locally Lipschitz at a point x_0 if a neighborhood $\boldsymbol{N}(x_0, r) = \{x \in R^n | ||x - x_0|| < r\}$ where f satisfies the Lipschitz condition:

$$||f(t,x) - f(t,y)|| \leq L||x - y||, L > 0 \tag{A.15}$$

The function is globally Lipschitz if it meets the above condition for all $\{x, y\}$ with the same value for the positive constant, L. Visually, this interpretation shows that any straight line joining points on f(x) cannot have a slope with an absolute value greater than L (Khalil 2015a).

$$\frac{|f(y) - f(x)|}{|y - x|} \leq L \tag{A.16}$$

Lipschitz continuity is practical when demonstrating the existence and uniqueness of a solution to general dynamical systems and non-linear processes. The smallest possible L that satisfies the above equation will correspond to the largest magnitude of the slope in the given interval. The property of Lipschitz continuity can be shown geometrically and Figure A.2 shows an example of a visual representation of Lipschitz continuity using a sinus curve.

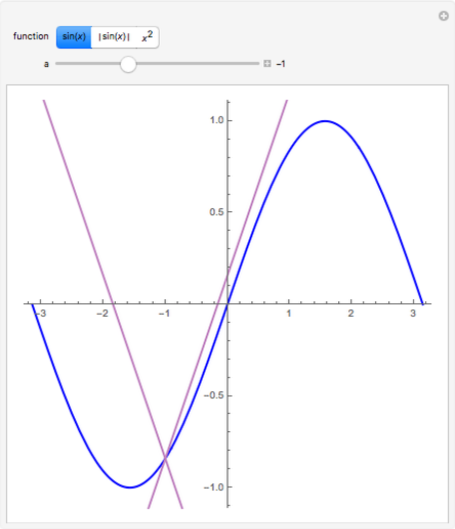


Figure A.2: Figure displaying Lipschitz continuity(Atwood 2011)

The continuity of $f'(x)$ ensures that $|f'(x)|$ is bounded by a constant k in the given neighborhood. If the partial derivatives are continuous for all relevant values of x, $x \in R^n$ the function will be globally Lipschitz for x if all the partial derivatives are also globally bounded and their absolute values are bounded for all values of t (Khalil 2015b).

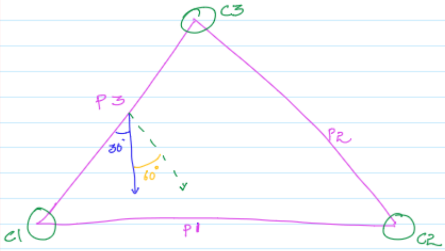
A.8 Small-gain theorem

This theorem applies to a well-defined system with the two transfer functions $\mathcal{H}_1$ and $\mathcal{H}_2$ which are finite-gain $\mathcal{L}$ stable in feedback connection with respective gains, γ_1, γ_2. The small-gain theorem states that the overall feedback connection is also finite-gain $\mathcal{L}$-stable if:

$$\gamma_1 \cdot \gamma_2 < 1 \tag{A.17}$$

This theorem proves to be very useful when studying the robustness of systems with model uncertainties as the small-gain requirement will be met for as long as γ_2 is small enough.

A.9 Derivation of quadratic damping force in surge



$$C_D^{effective}\, V|V| = C_D^P\, V|V|\, L_p + 2*\left(C_D^P\, V\cos 60\, |V\cos 60|\, \cos 60\right) L_p + 3*\left(C_D^C\, V|V|\right) L_c$$

$$C_D^{effective} = C_D^P\, L_p + \frac{C_D^P}{4} L_p + 3\, C_D^C\, L_c$$

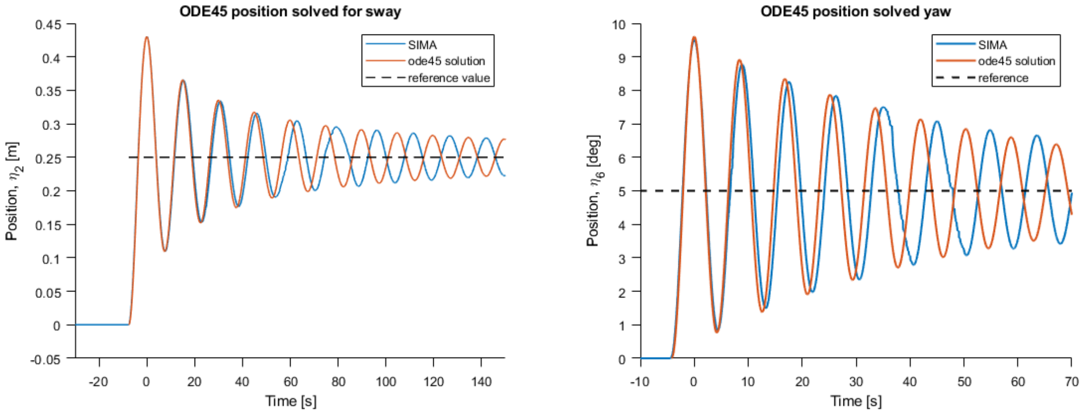
$$\boxed{C_D^{effective} = \frac{5}{4}\, C_D^P\, L_p + 3\, C_D^C\, L_c}$$

Figure A.3: Derivation of quadratic damping coefficient in surge.

Appendix B

Appendix B: Plots

B.1 Model verification: decay



(a) ODE fitted to SIMA response in sway

(b) ODE fitted to SIMA response in yaw

Figure B.1: Decay response for sway and yaw.

B.2 K3 tuning

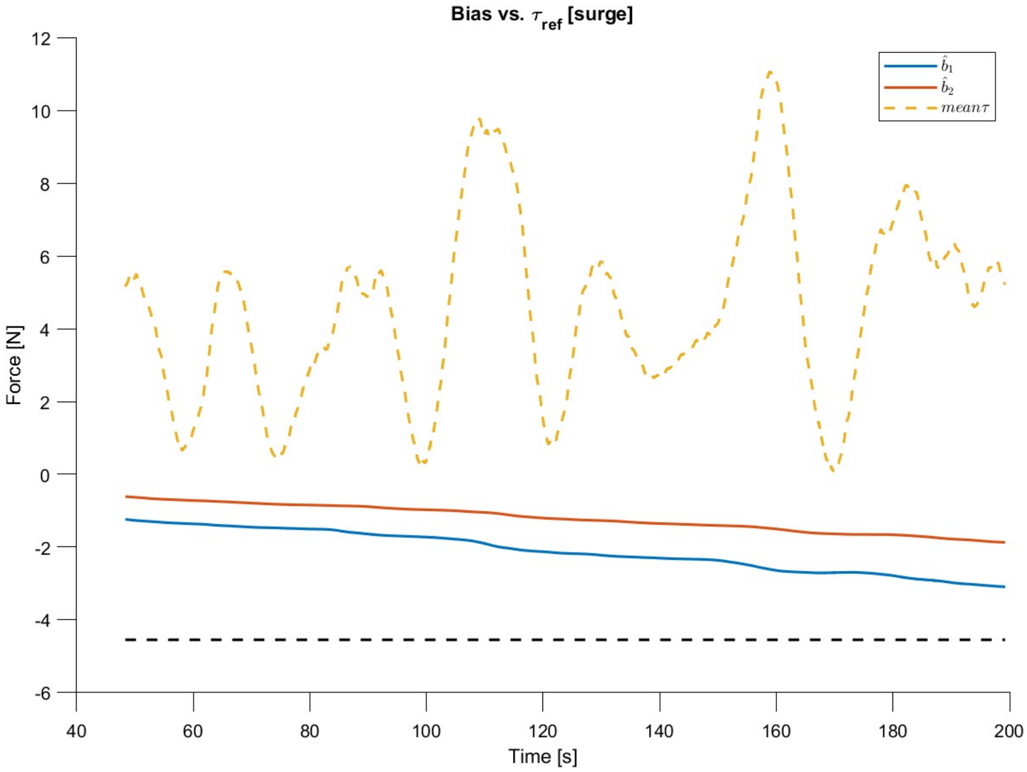


Figure B.2: Bias comparison with K3=1 and K3=0.5.

B.3 K4 tuning

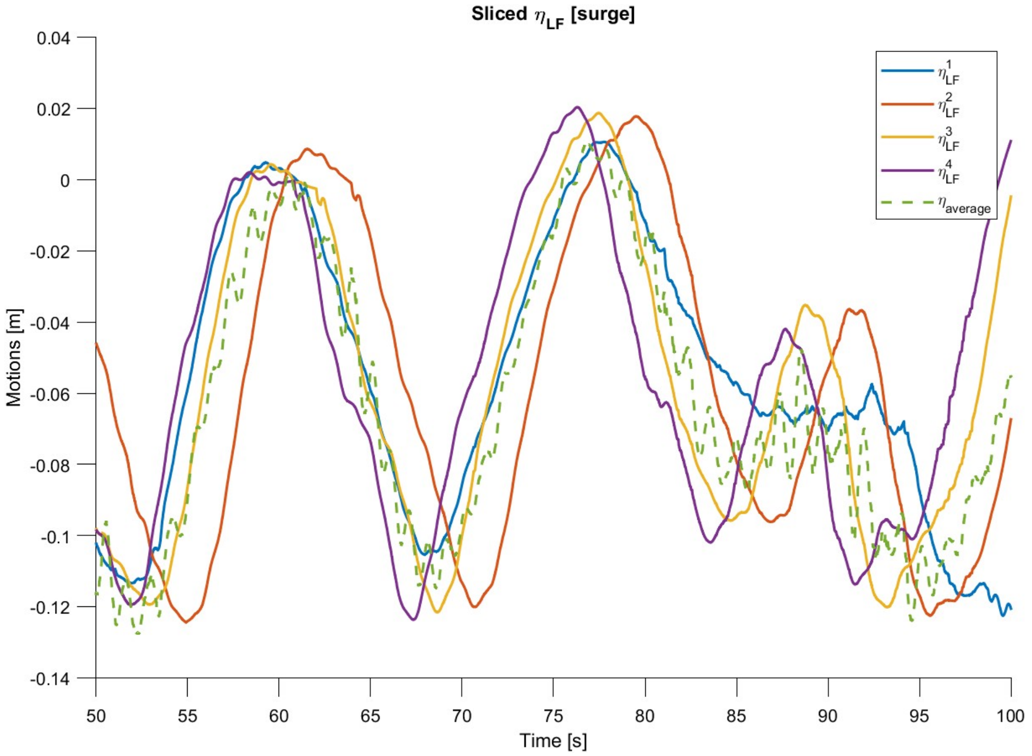


Figure B.3: Sliced plot of η for various values of K_4.

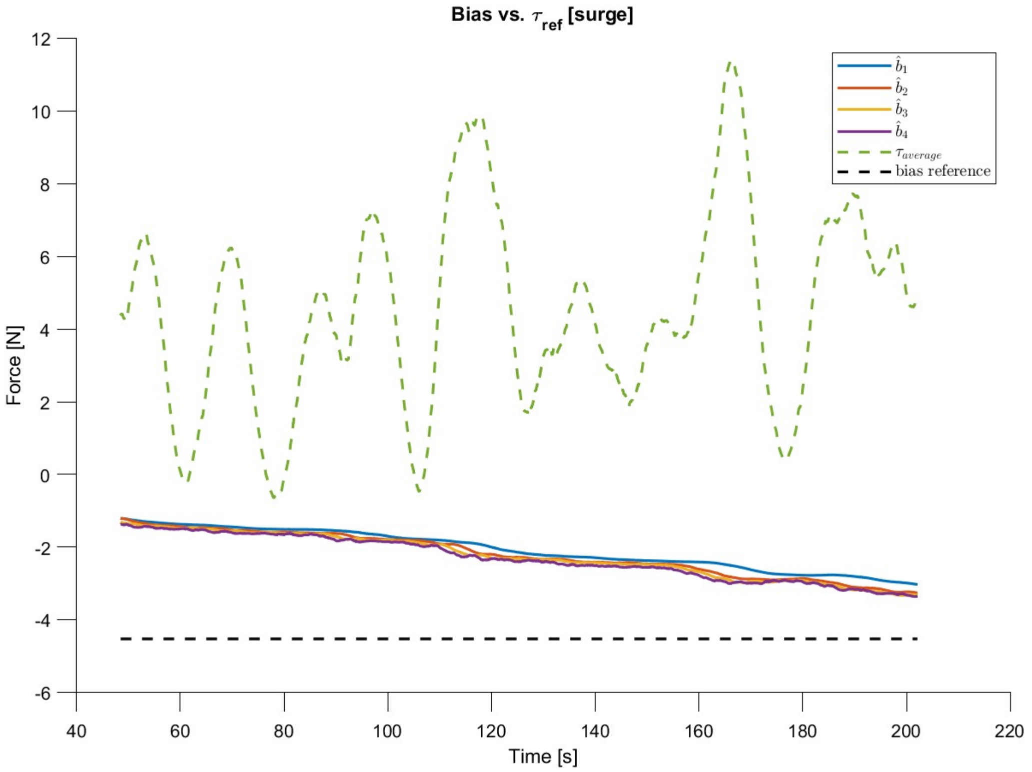
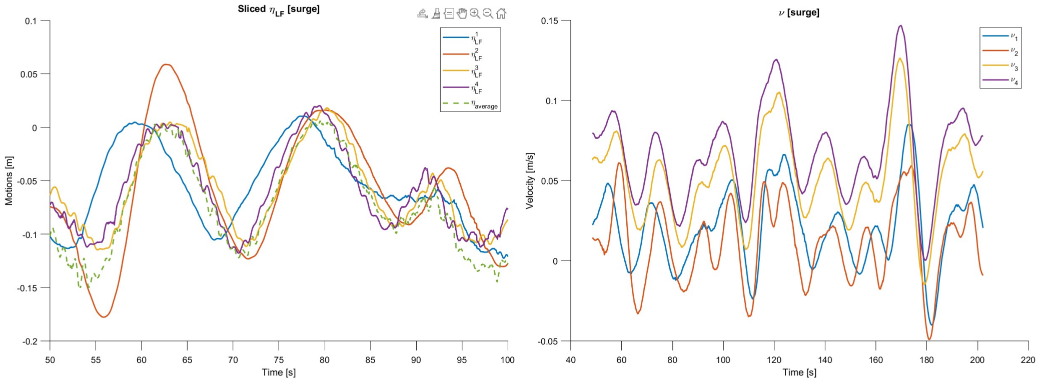


Figure B.4: Bias plotted for various values of K_4.

B.4 K2 tuning



(a) η plotted for various values of K_2.

(b) ν plotted for various values of K_2.

Figure B.5: Simulation with varying values of K_2

B.5 K2 tuning with updated K3

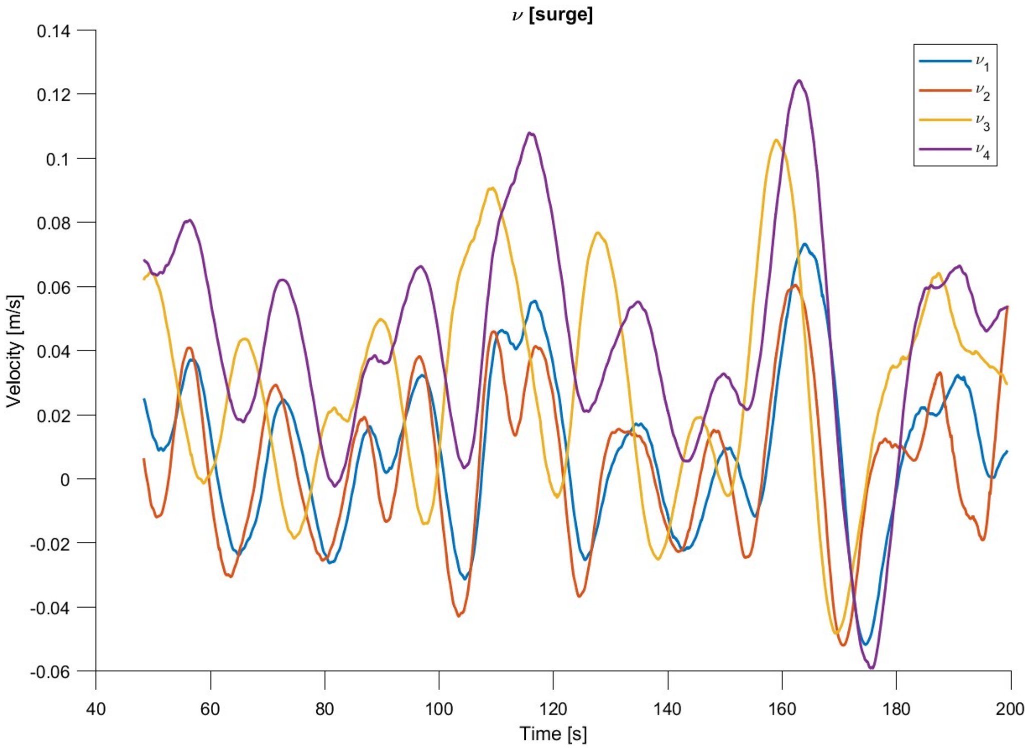


Figure B.6: ν plotted for various values of K_2 with updated K_3.

B.6 Damping factor

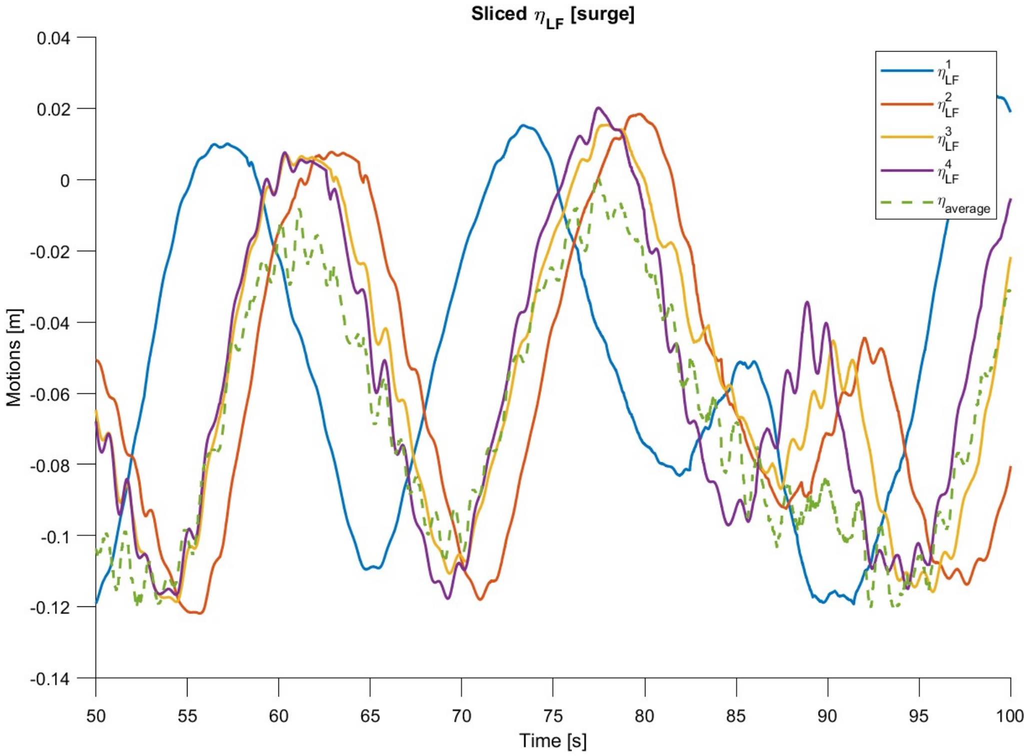


Figure B.7: Sliced plot of $\hat{\eta}$ for $\zeta \in [0.1, 0.3, 0.5, 0.7]$.

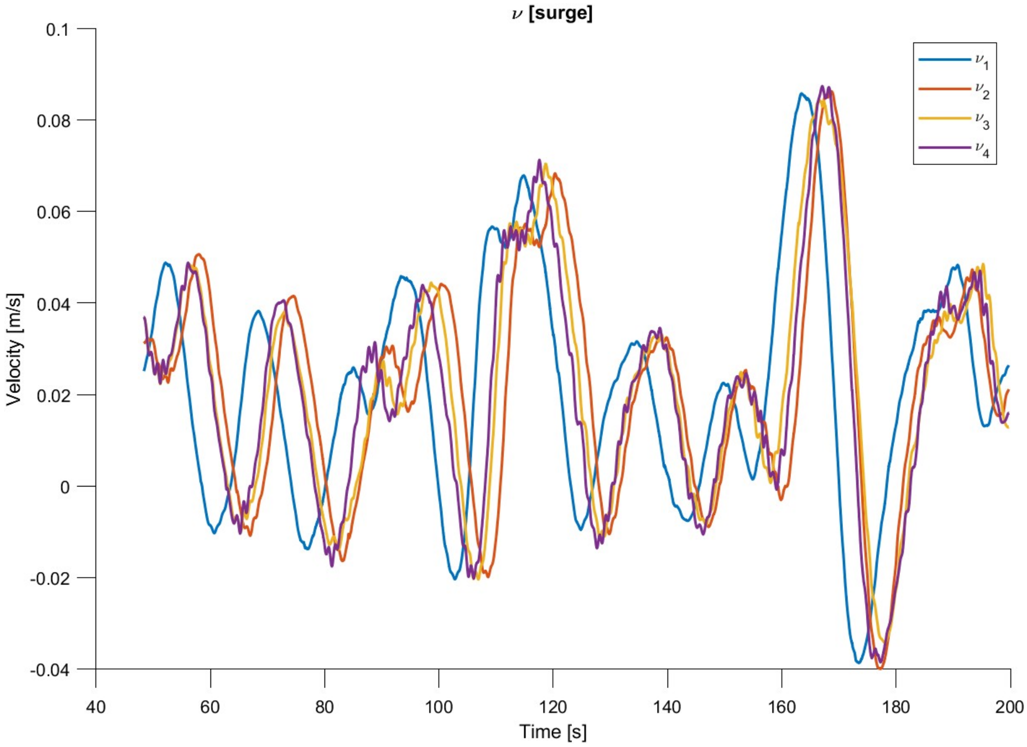


Figure B.8: $\hat{\nu}$ for $\zeta \in [0.1, 0.3, 0.5, 0.7]$.

B.7 Cutoff frequency

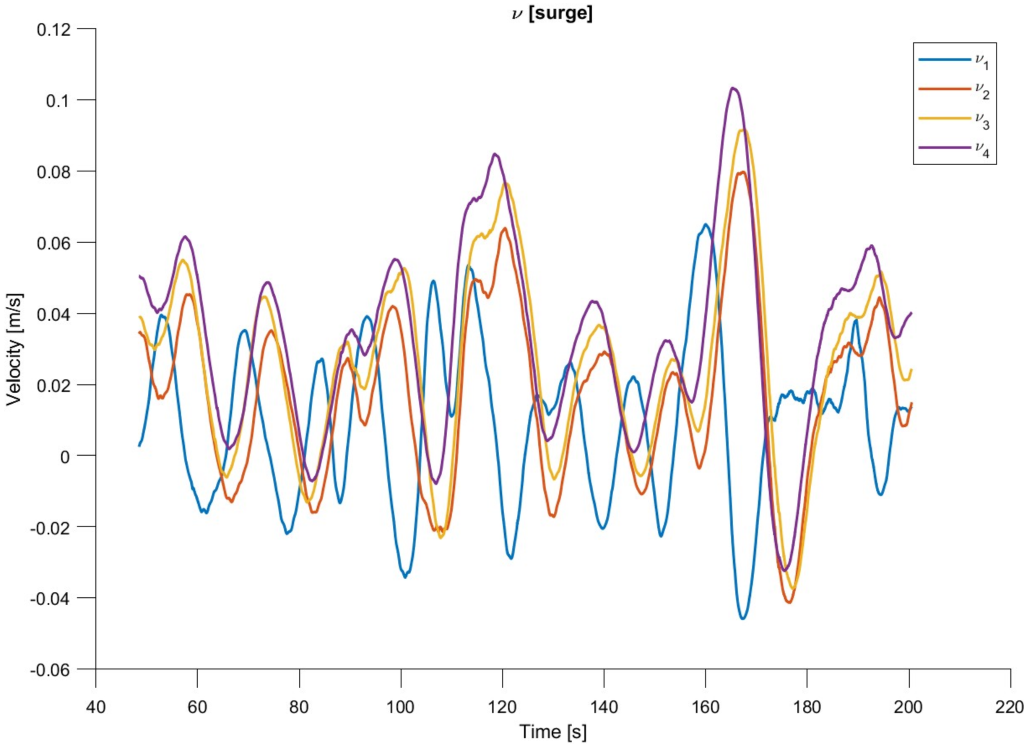


Figure B.9: Velocity estimate for cutoff frequencies= $[0.5, 1, 1.5, 2]$

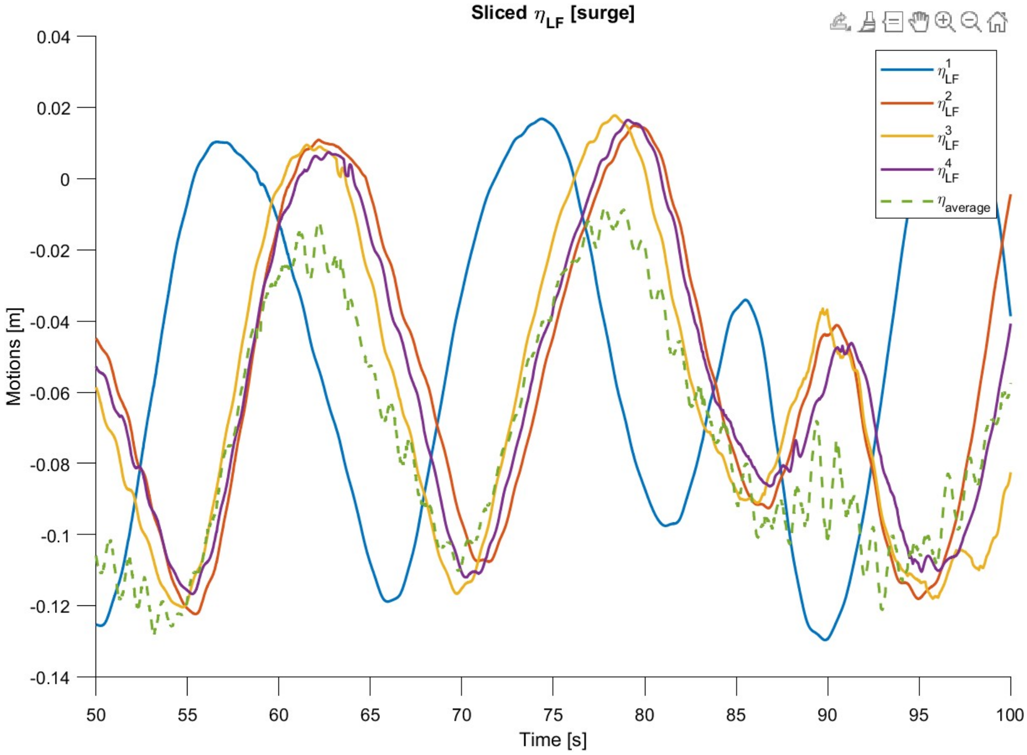


Figure B.10: Sliced position estimate for cutoff frequencies= $[0.5, 1, 1.5, 2]$

B.8 Varying T_b

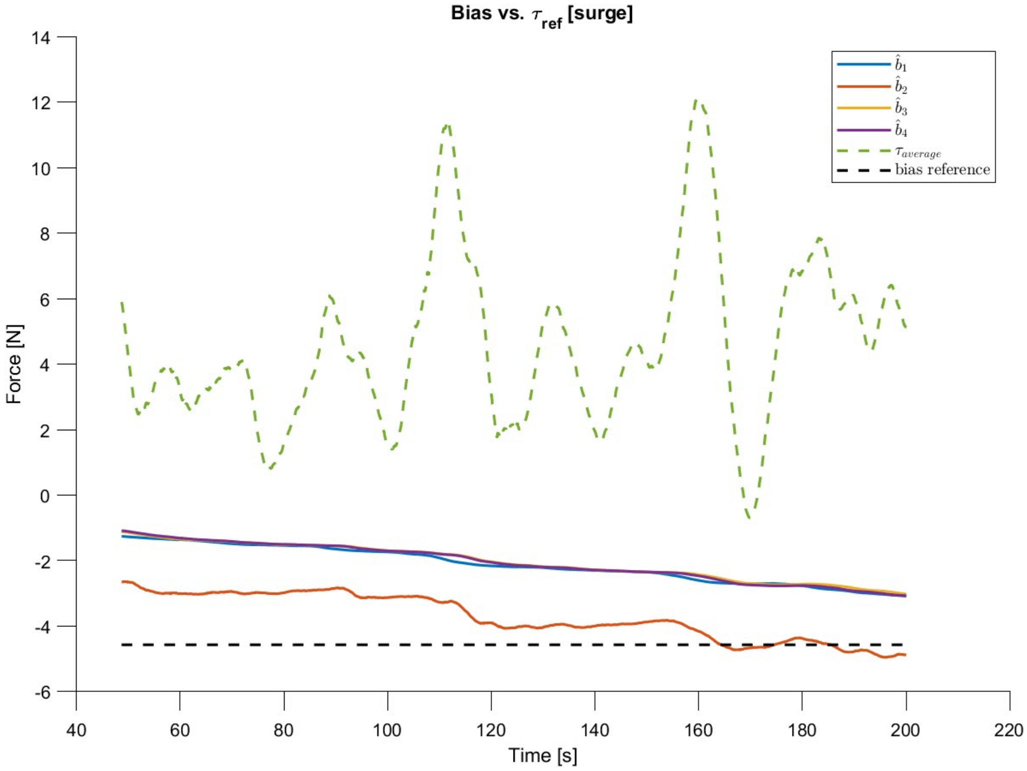


Figure B.11: Bias force for $T_b = [0.5, 1, 1.5, 2]$

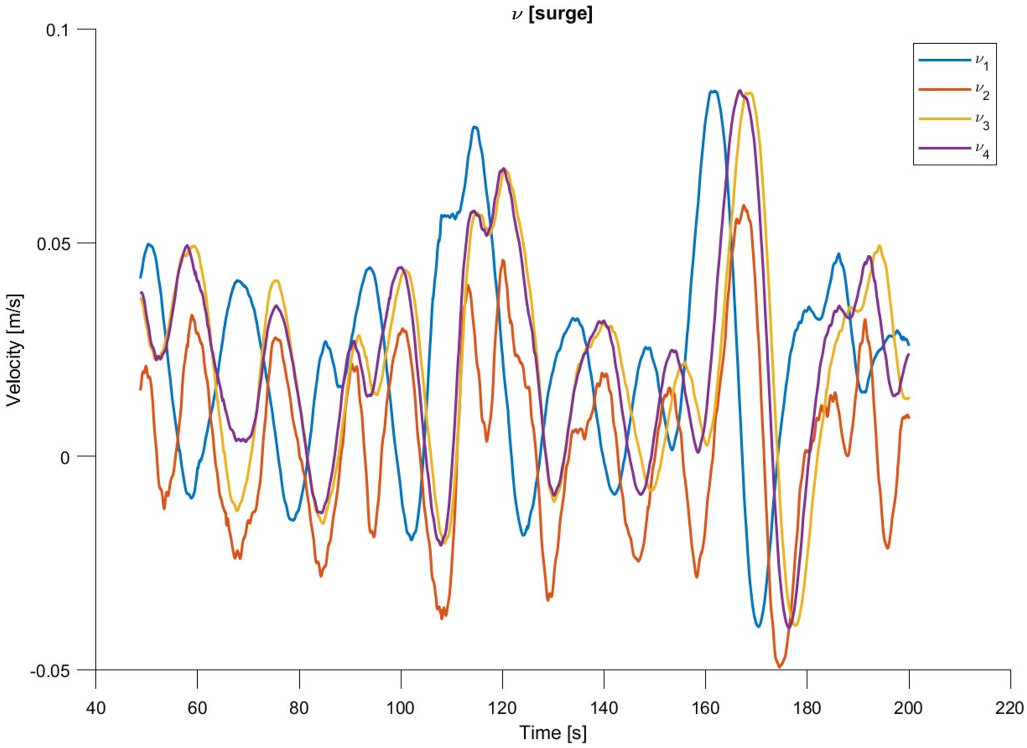


Figure B.12: Velocity estimate for $T_b = [0.5, 1, 1.5, 2]$

B.9 Varying T_b for $K_3 = 10$

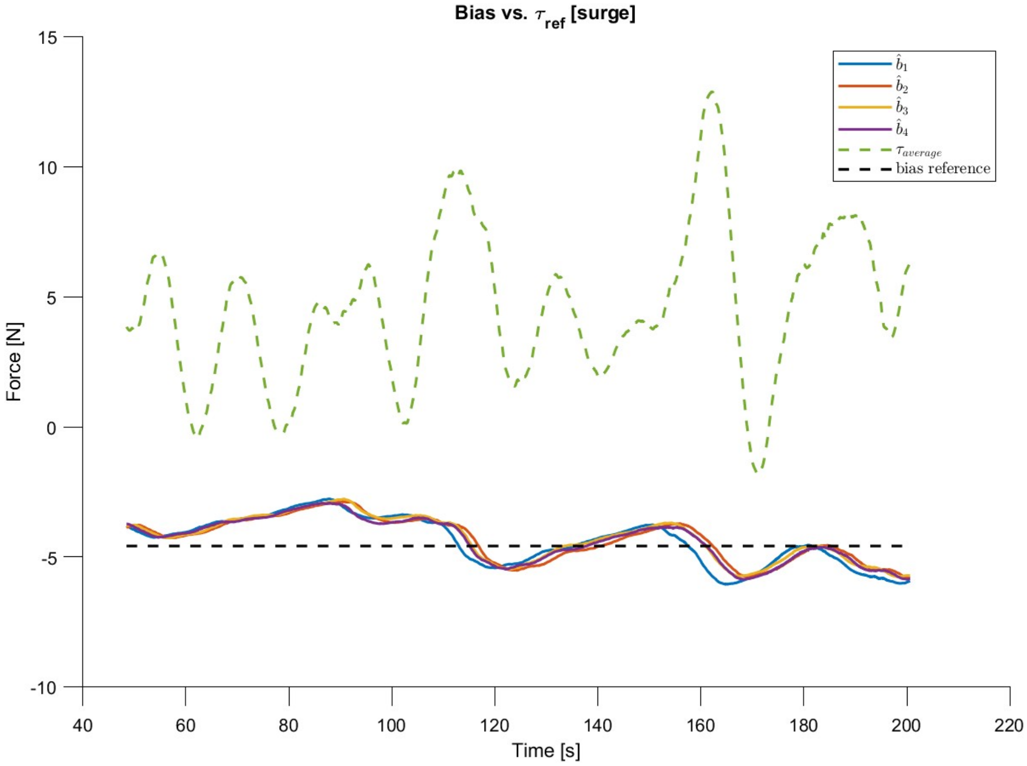


Figure B.13: Bias for $K_3 = 10$ and $T_b \in [0.5, 1, 1.5, 2]$.

B.10 Response to different angles

A 30 degrees environment

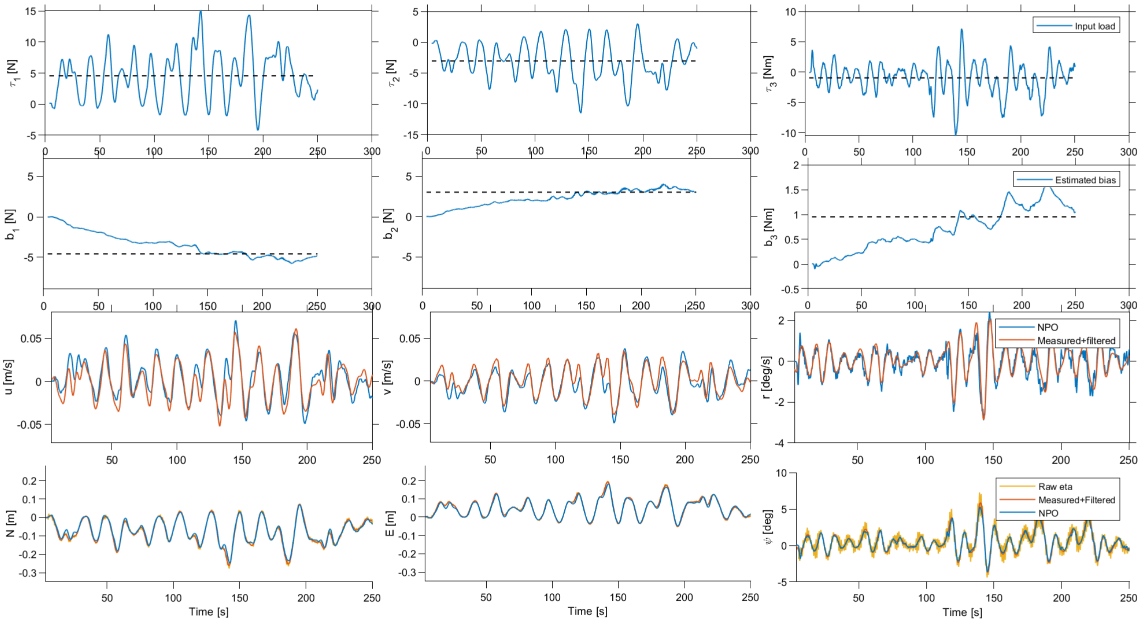


Figure B.14: Simulation with the environment from 30 degrees.

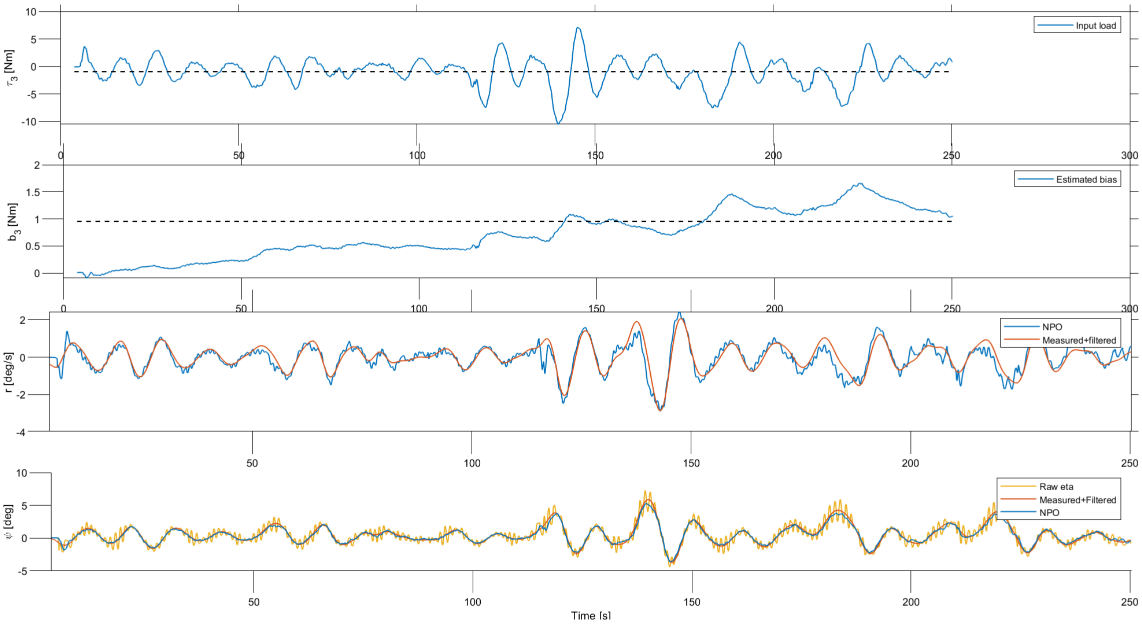


Figure B.15: Yaw response for environment from 30 degrees.

B 60 degrees environment

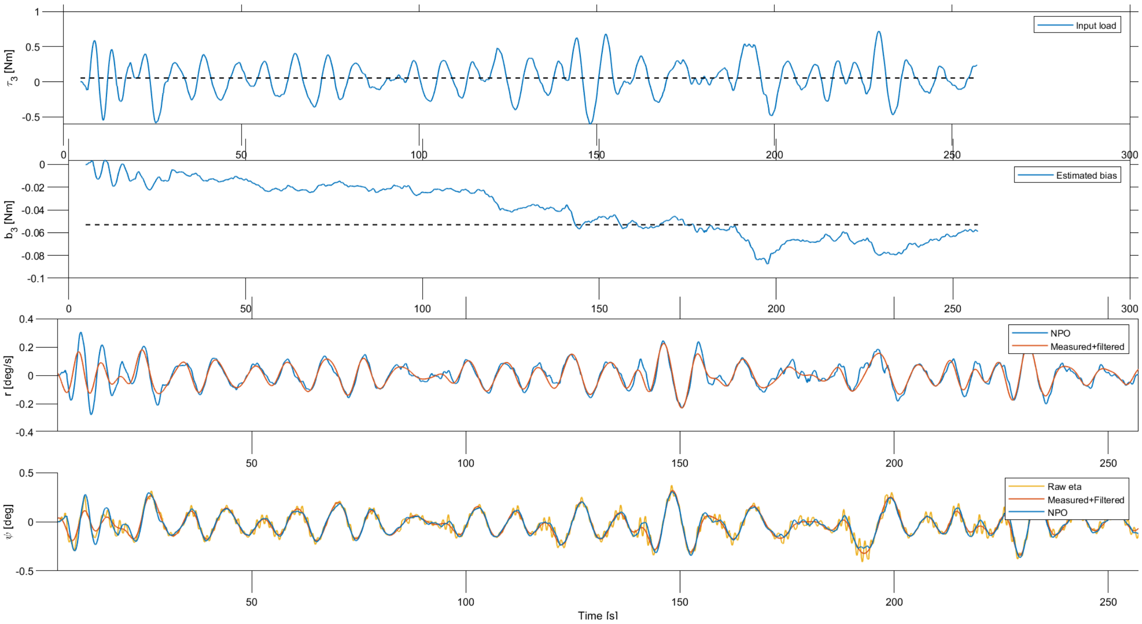


Figure B.16: Yaw response for environment from 60 degrees.

C Sway test - 90 degrees environment

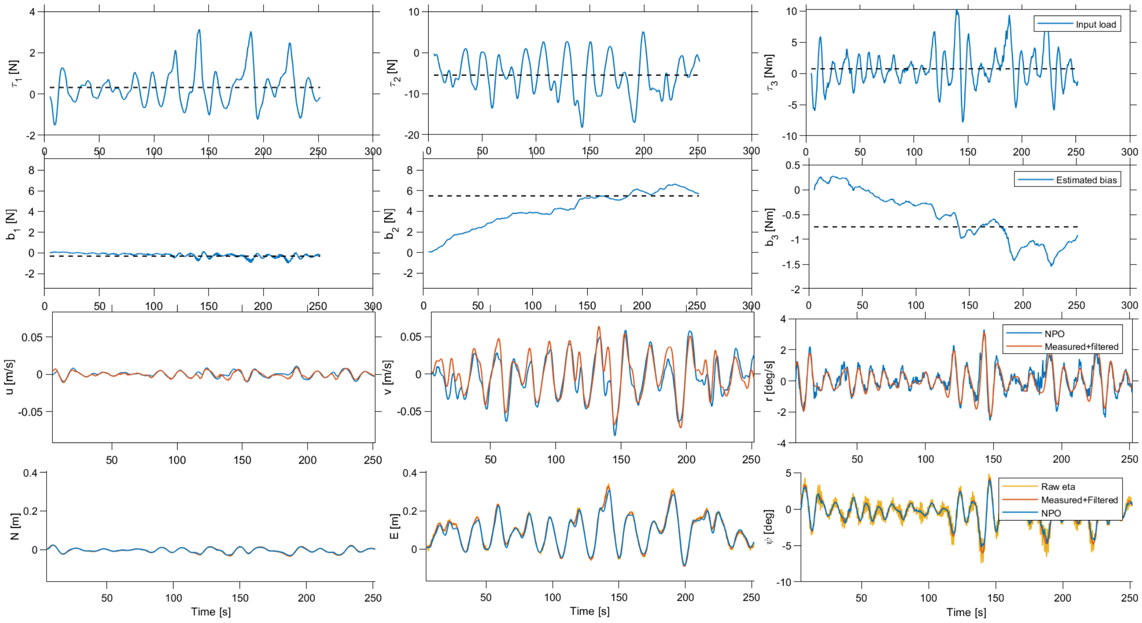


Figure B.17: Simulation with the environment from 90 degrees.

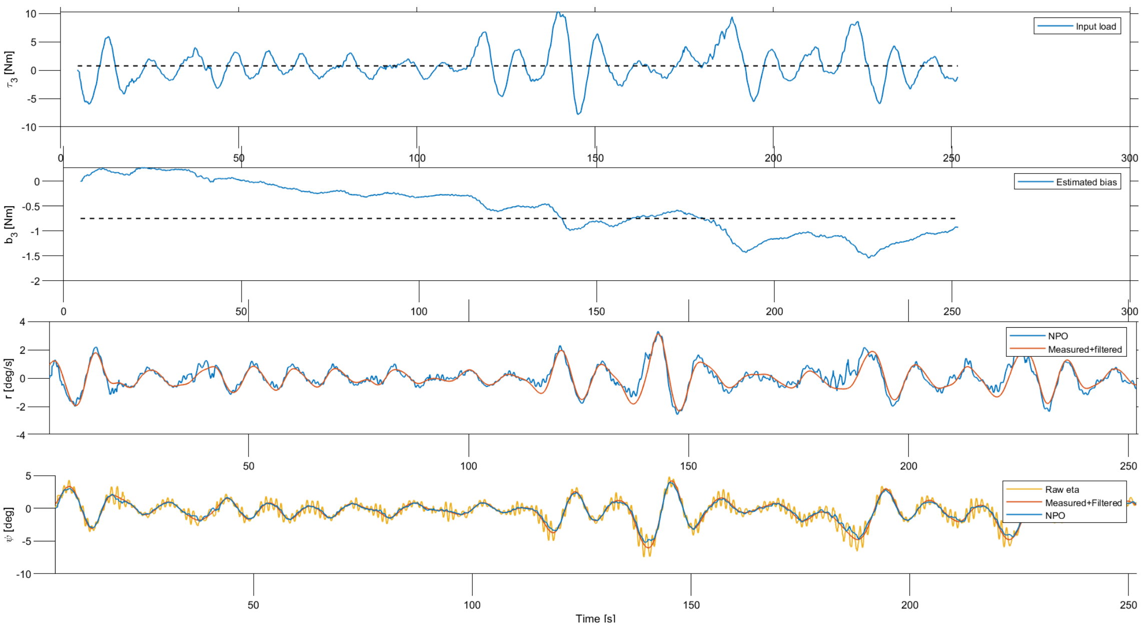


Figure B.18: Yaw response for environment from 90 degrees.

D Surge test - 180 degrees environment

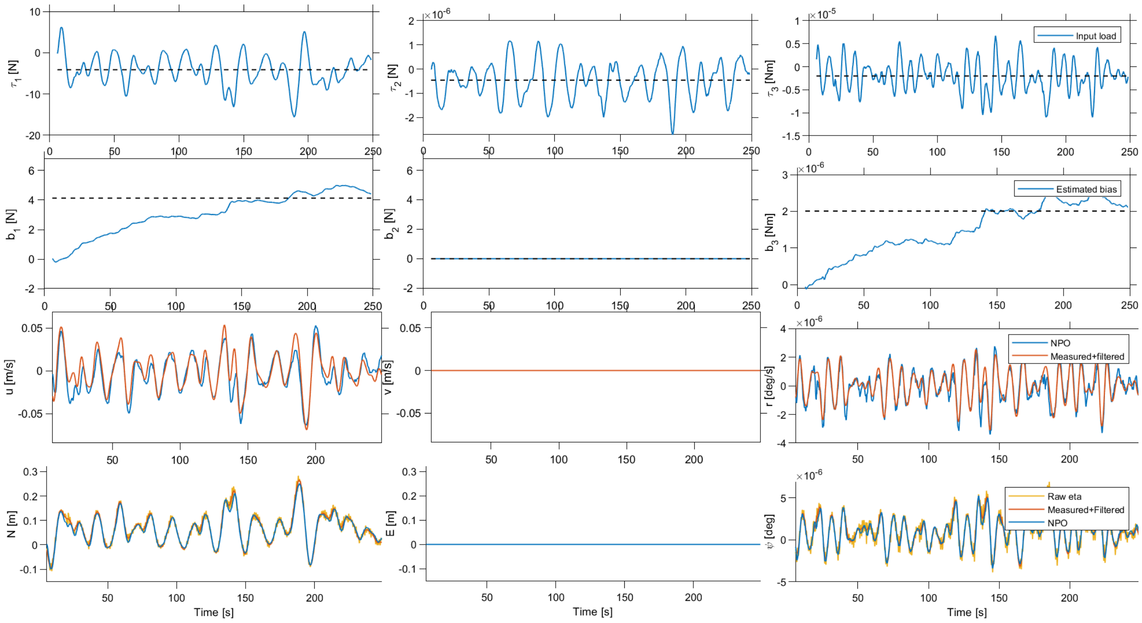


Figure B.19: Simulation with the environment from 180 degrees.

B.11 Yaw tuning

A K4

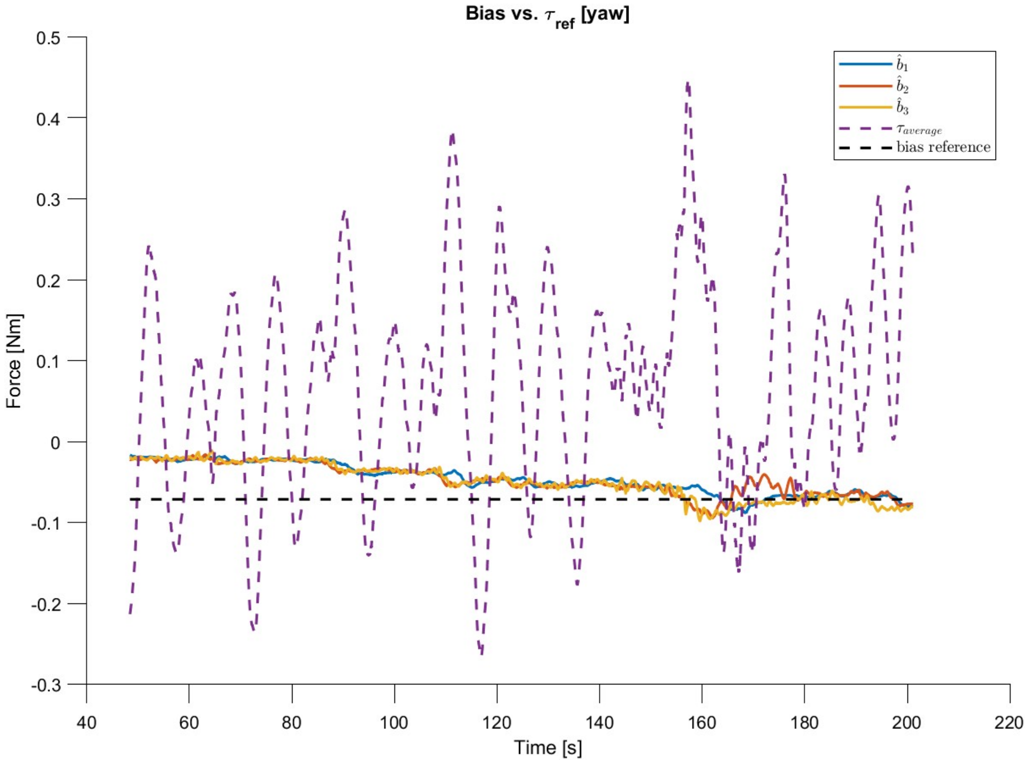


Figure B.20: Bias in yaw for various values of $K_{4,3} = [2.7, 5, 10]$.

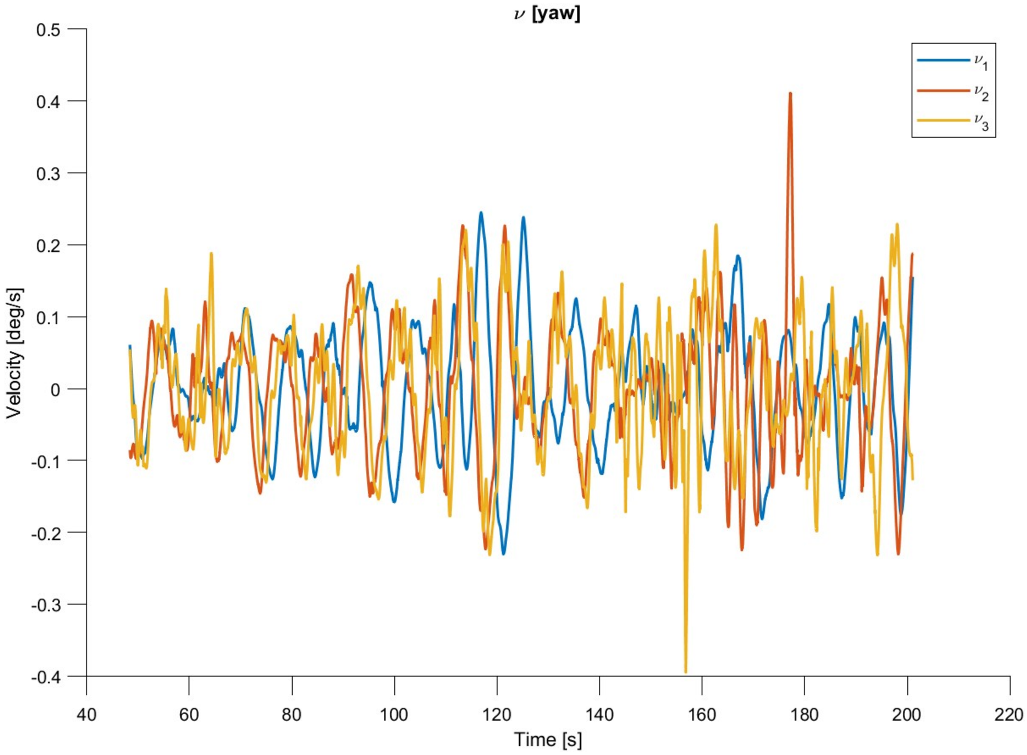


Figure B.21: Velocity estimate in yaw for various values of $K_{4,3} = [2.7, 5, 10]$.

B K2

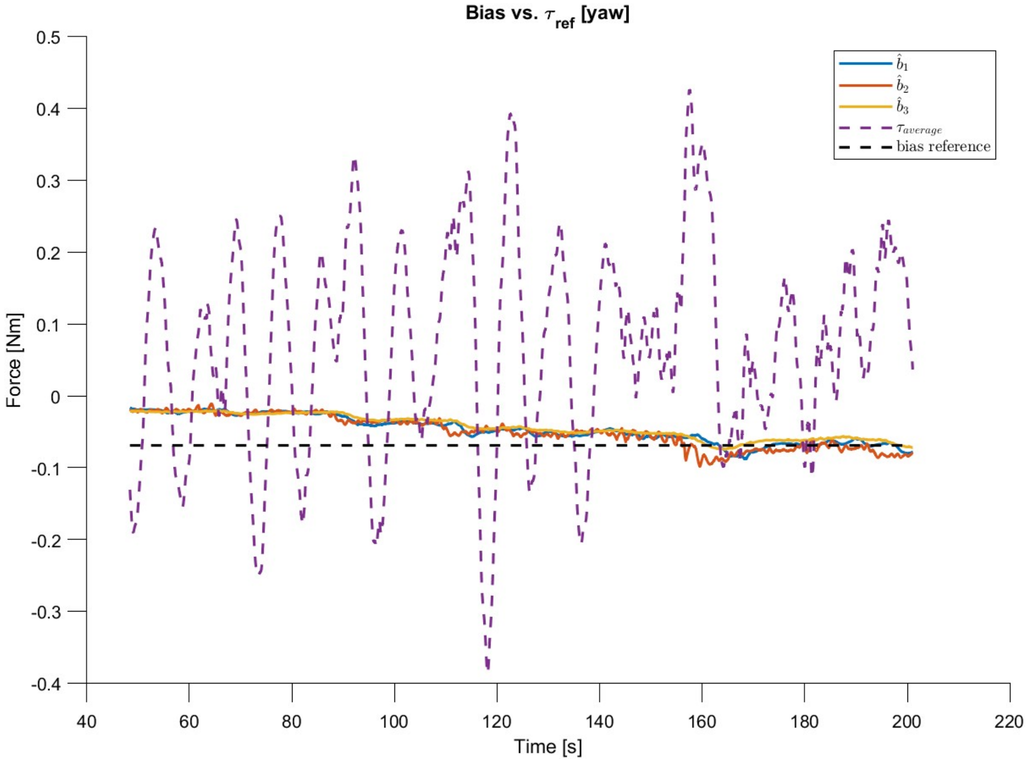


Figure B.22: Bias in yaw for various values of $K_{2,3} = [1, 5, 10]$.

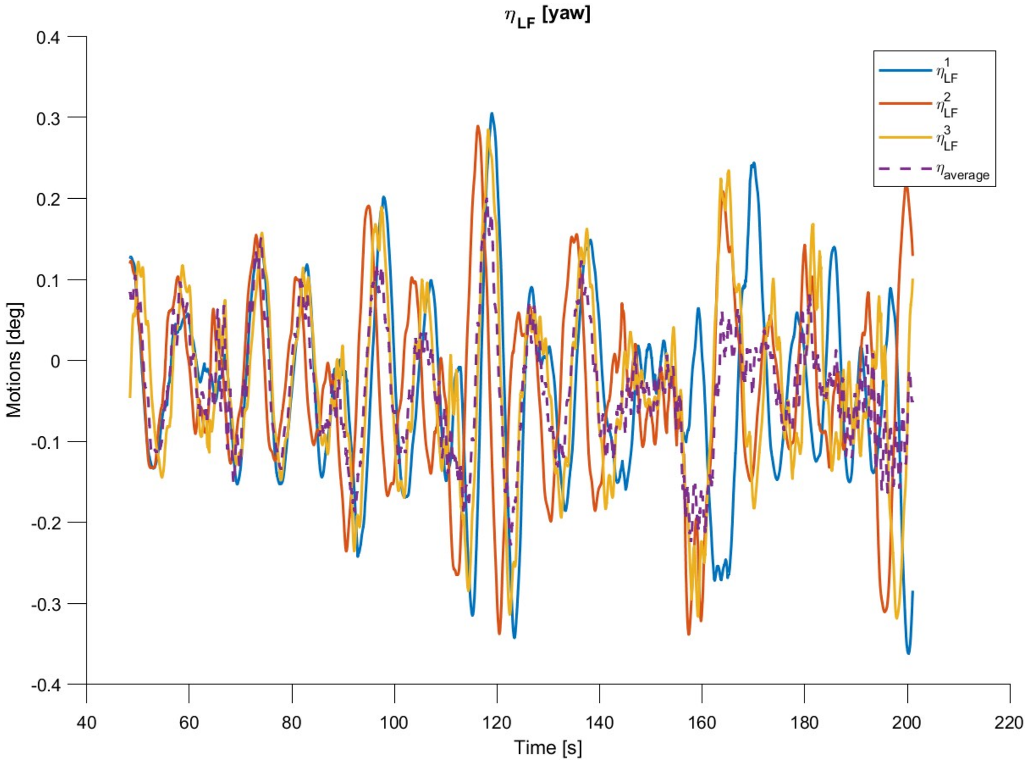


Figure B.23: Position estimate in yaw for various values of $K_{2,3} = [1, 5, 10]$.

B.12 Different sea-state simulations

A $Hs = 3.75$ and $T_p = 7$

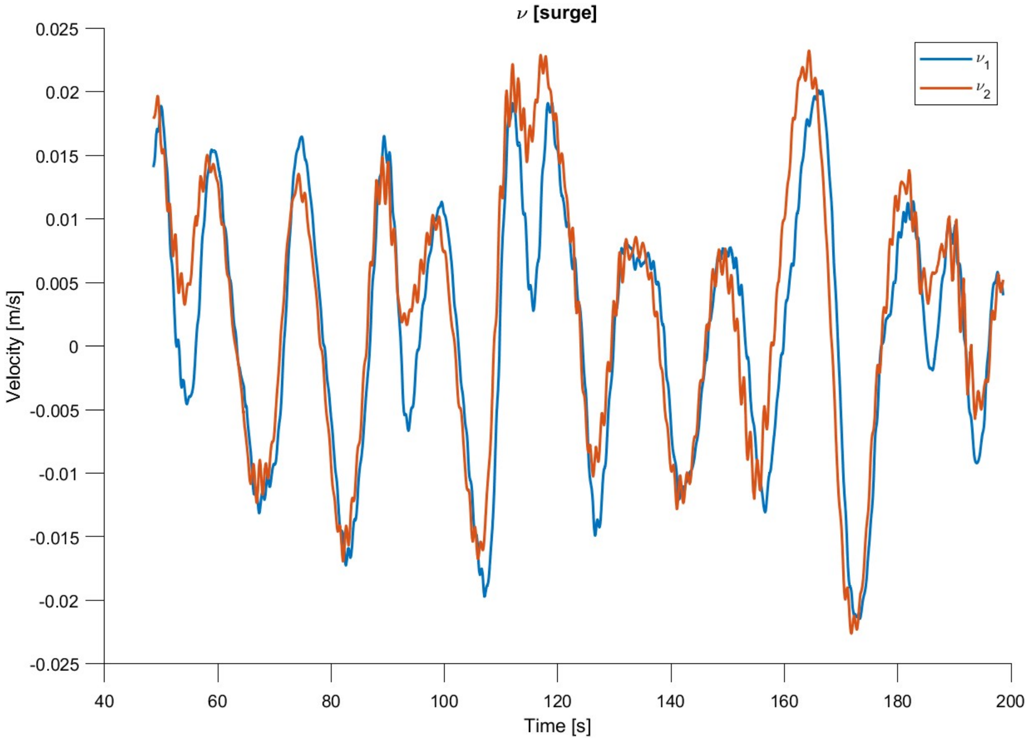


Figure B.24: $\hat{\nu}$ for $H_s = 3.75$ and $T_p = 7$.

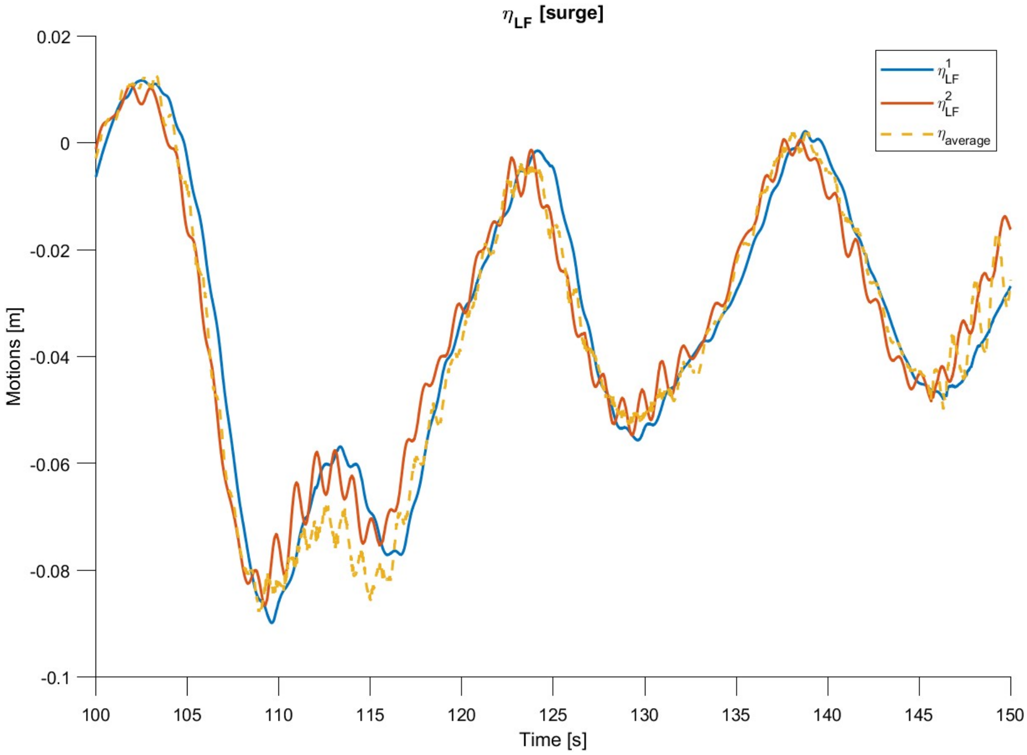


Figure B.25: Sliced $\hat{\eta}_{LF}$ for $H_s = 3.75$ and $T_p = 7$.

B $H_s = 14.97$ and $T_p = 14$

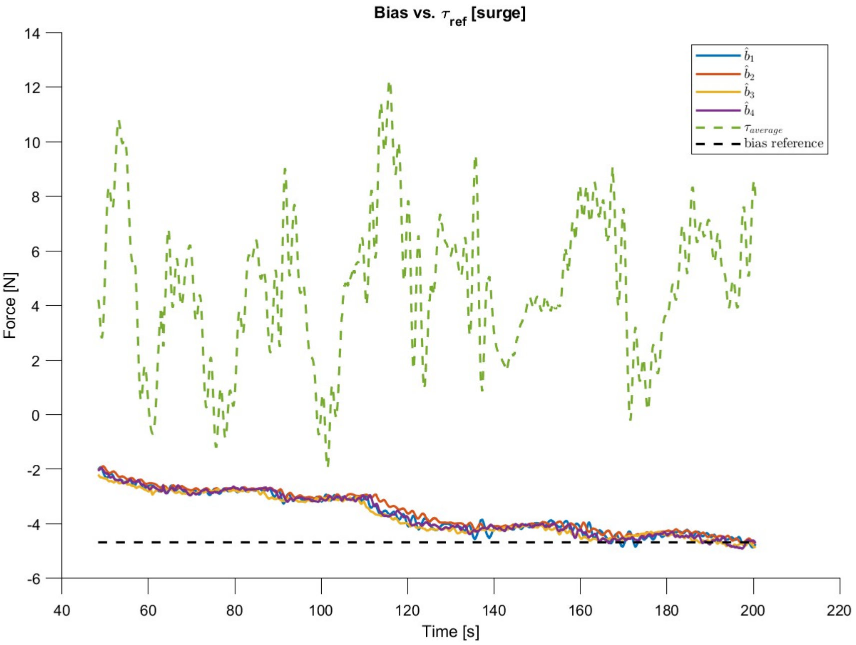


Figure B.26: Bias for $H_s = 14.97$ and $T_p = 14$.

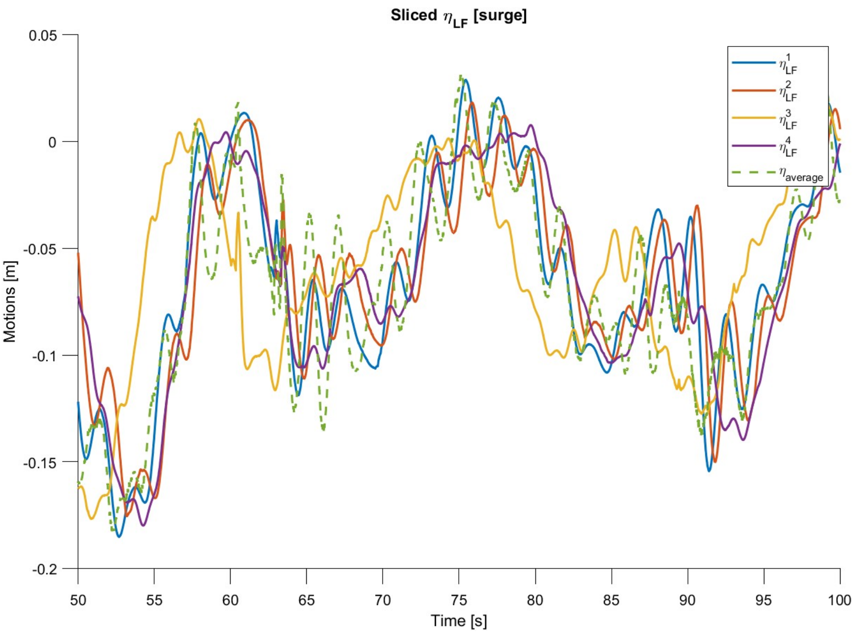
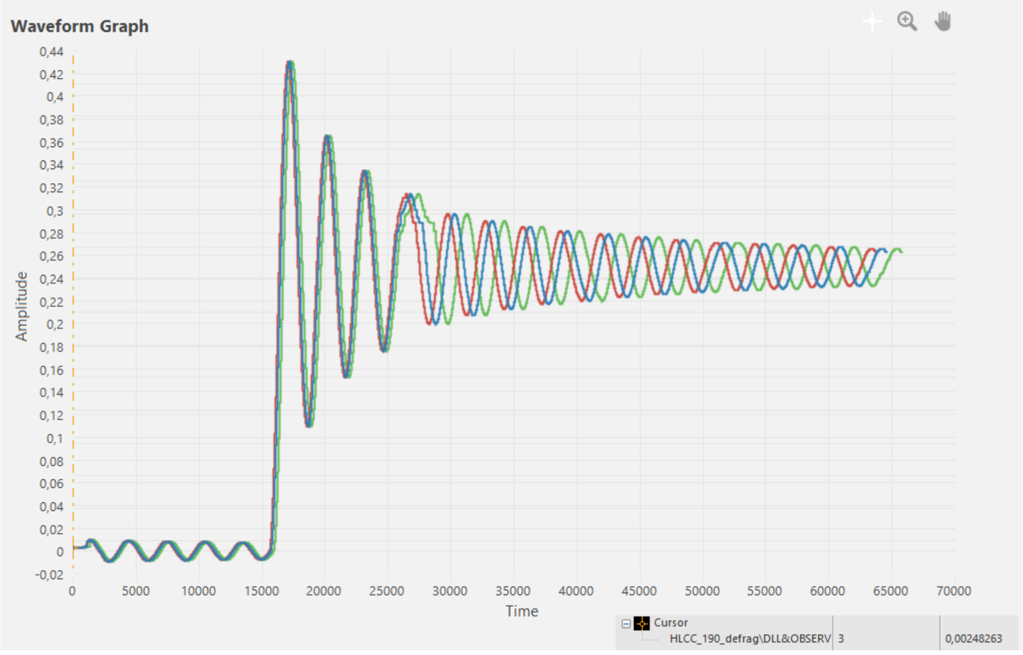


Figure B.27: Sliced plot of $\hat{\eta}$ for $H_s = 14.97$ and $T_p = 14$.

B.13 Comparison of signals in HLCC



HLCC_190_defrag\DLL&OBSERV 3 0,00248263

Figure B.28: Comparison of input to the observer, output from the mooring, and coordinate transform.

B.14 Initial simulation parameters

Simulation Parameters			
Parameter	*Explaination*	*Symbol*	*Unit*
Observer Parameters (HLCC)			
Dl_1	Linear damping in surge	0.139044	$N/(m/s)$
Dl_2	Linear damping in sway	0.139044	$N/(m/s)$
Dl_6	Linear damping in yaw	0.109795	$N/(deg/s)$
Dq_1	Quadratic damping in surge	945204	$N/(m/s)^2$
Dq_2	Quadratic damping in sway	945204	$N/(m/s)^2$
Dq_6	Quadratic damping in yaw	3834.97	$Nm/(deg/s)^2$
$K_{2,1}$	Observer gain	1	[-]
$K_{2,2}$	Observer gain	1	[-]
$K_{2,3}$	Observer gain	1	[-]
K_3	Observer gain	1	[-]
$K_{4,1}$	Observer gain	1	[-]
$K_{4,2}$	Observer gain	1	[-]
$K_{4,3}$	Observer gain	1	[-]
M_1	Mass in surge	295.799	kg
M_2	Mass in sway	295.799	kg
M_6	Mass in yaw	175.611	kg
T_b	Bias time constant	1	s
T_p	Observer peak period	1.195	s
$bypassObs$	Observer bypass switch	0	[-]
$cutoffWFObs$	Observer cutoff frequency	1.22	s
$dampingWFObs$	Observer damping term	0.1	[-]
$scale$	scaling factor	40	[-]
Wind Parameters (SIMA)			
$Direction$	Incoming direction of wind	0	deg
U_0	Mean wind speed	$1 \cdot 10^{-6}$	m/s
Mooring Parameters (HLCC)			
k_i	Inline mooring stiffness	53.56	N/m
k_t	Transverse mooring stiffness	53.36	N/m
k_{rot}	Rotational mooring stiffness	5.449	$Nm/deg)$
Wave Parameters (SIMA)			
Direction	Incoming direction of waves	0	deg
Spread type	[-]	Unidirectional	[-]
Spread exponent	[-]	2	[-]
Number of directions	[-]	11	[-]
H_s	Significant wave height	6	m
T_p	Wave period	7.55	s
γ	Peak enhancement factor, steepness of spectrum	3.3	[-]

Table B.1: Initial simulation parameters.

B.15 Updated model parameters

Simulation Parameters			
Parameter	*Explaination*	*Symbol*	*Unit*
Observer Parameters (HLCC)			
Dl_1	Linear damping in surge	0.139	$N/(m/s)$
Dl_2	Linear damping in sway	0.139	$N/(m/s)$
Dl_6	Linear damping in yaw	**0.1098**	$Nm/(deg/s)$
Dq_1	Quadratic damping in surge	**389**	$N/(m/s)^2$
Dq_2	Quadratic damping in sway	**389**	$N/(m/s)^2$
Dq_6	Quadratic damping in yaw	**0.06**	$Nm/(deg/s)^2$
$K_{2,1}$	Observer gain	1	[-]
$K_{2,2}$	Observer gain	1	[-]
$K_{2,3}$	Observer gain	1	[-]
K_3	Observer gain	1	[-]
$K_{4,1}$	Observer gain	1	[-]
$K_{4,2}$	Observer gain	1	[-]
$K_{4,3}$	Observer gain	1	[-]
M_1	Mass in surge	**338**	kg
M_2	Mass in sway	**338**	kg
M_6	Mass in yaw	**202.6**	kg
T_b	Bias time constant	1	s
T_p	Observer peak period	1.195	s
$bypassObs$	Observer bypass switch	0	[-]
$cutoffWFObs$	Observer cutoff frequency	1.22	s
$dampingWFObs$	Observer damping term	0.1	[-]
$scale$	scaling factor	40	[-]
Mooring Parameters (HLCC)			
k_i	Inline mooring stiffness	**62.2**	N/m
k_t	Transverse mooring stiffness	**59.94**	N/m
k_{rot}	Rotational mooring stiffness	**1.983**	$Nm/deg)$

Table B.2: Updated model parameters.

B.16 WP4 model

Simulation Parameters			
Parameter	*Explaination*	*Symbol*	*Unit*
Observer Parameters (HLCC)			
Dl_1	Linear damping in surge	0.139	$N/(m/s)$
Dl_2	Linear damping in sway	0.139	$N/(m/s)$
Dl_6	Linear damping in yaw	6.29	$Nm/(deg/s)$
Dq_1	Quadratic damping in surge	0	$N/(m/s)^2$
Dq_2	Quadratic damping in sway	0	$N/(m/s)^2$
Dq_6	Quadratic damping in yaw	0	$Nm/(deg/s)^2$
$K_{2,1}$	Observer gain	1	[-]
$K_{2,2}$	Observer gain	1	[-]
$K_{2,3}$	Observer gain	1	[-]
K_3	Observer gain	1	[-]
$K_{4,1}$	Observer gain	1	[-]
$K_{4,2}$	Observer gain	4	[-]
$K_{4,3}$	Observer gain	10	[-]
M_1	Mass in surge	430.61	kg
M_2	Mass in sway	430.61	kg
M_6	Mass in yaw	0.24	kg
T_b	Bias time constant	1	s
T_p	Observer peak period	0.1750	s
$bypassObs$	Observer bypass switch	0	[-]
$cutoffWFObs$	Observer cutoff frequency	1.22	s
$dampingWFObs$	Observer damping term	0.1	[-]
$scale$	scaling factor	40	[-]
Mooring Parameters (HLCC)			
k_i	Inline mooring stiffness	58.48	N/m
k_t	Transverse mooring stiffness	59.94	N/m
k_{rot}	Rotational mooring stiffness	79.33	$Nm/deg)$

Table B.3: WP4, offline experiment model parameters.

B.17 Tuned model

Simulation Parameters			
Parameter	*Explaination*	*Symbol*	*Unit*
Observer Parameters (HLCC)			
Dl_1	Linear damping in surge	0.139	$N/(m/s)$
Dl_2	Linear damping in sway	0.139	$N/(m/s)$
Dl_6	Linear damping in yaw	0.1098	$Nm/(deg/s)$
Dq_1	Quadratic damping in surge	389	$N/(m/s)^2$
Dq_2	Quadratic damping in sway	389	$N/(m/s)^2$
Dq_6	Quadratic damping in yaw	0.06	$Nm/(deg/s)^2$
$K_{2,1}$	Observer gain	**1**	[-]
$K_{2,2}$	Observer gain	**1**	[-]
$K_{2,3}$	Observer gain	**1**	[-]
K_3	Observer gain	**3**	[-]
$K_{4,1}$	Observer gain	**2.7**	[-]
$K_{4,2}$	Observer gain	**2.7**	[-]
$K_{4,3}$	Observer gain	**2.7**	[-]
M_1	Mass in surge	338	kg
M_2	Mass in sway	338	kg
M_6	Mass in yaw	202.6	kg
T_b	Bias time constant	1	s
T_p	Observer peak period	1.195	s
$bypassObs$	Observer bypass switch	0	[-]
$cutoffWFObs$	Observer cutoff frequency	1.22	s
$dampingWFObs$	Observer damping term	0.1	[-]
$scale$	scaling factor	40	[-]

Table B.4: Tuned model parameters.

Appendix C

Appendix C: Code and models

MATLAB file folder:

- compare.m
- computing.m
- INO12MW_values.m
- odeSolver.m
- one_plot.m
- plotting.m
- processing.m
- Rzyx_.m

HLCC file folder:

- HLCC_setup
- alloc.fmu
- computeLoad.fmu
- coordinateTransformation.fmu
- mooring.fmu
- observer.fmu
- winchTensionControllerSIMA.fmu
- externalWinchController.jar
- HLCC_setup.ini

SimulationExample:

- HLCC_225_defrag.tdms
- HLCC_Setup

Simulink file folder:

- observer.slx

SIMA file folder:

- CYBERLAB.stask

C.1 Simulink model of observer

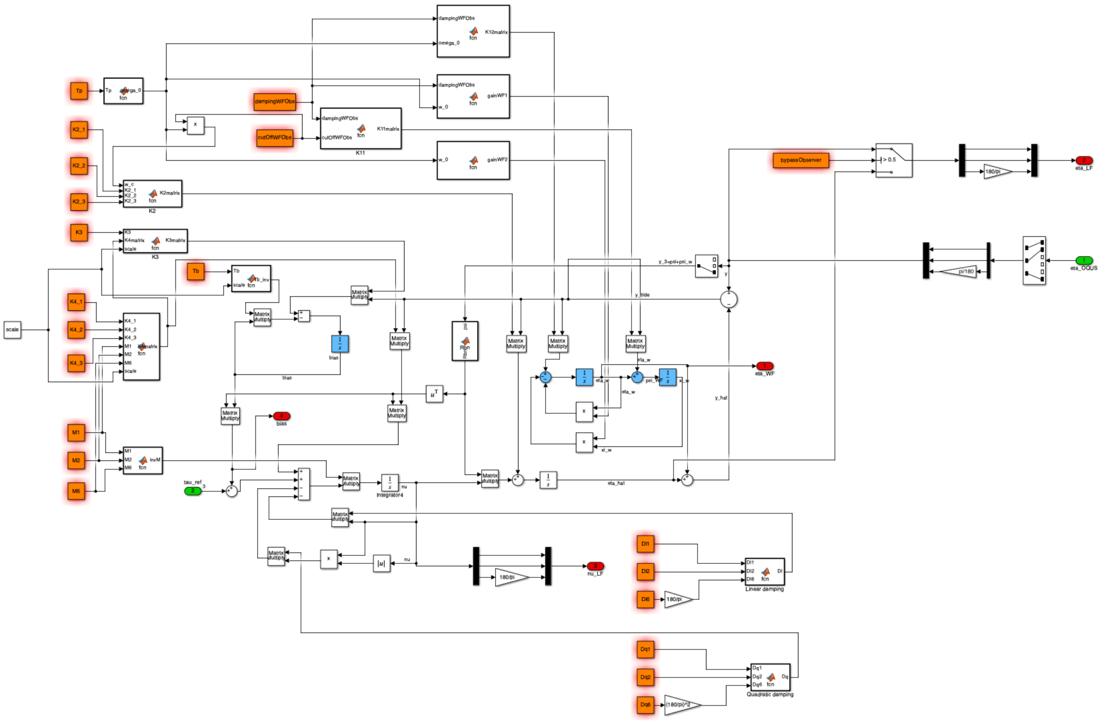


Figure C.1: Simulink model of NPO.

NTNU

Norwegian University of
Science and Technology