Master's thesis

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

Eirik Midtun

Performance analysis of dynamic positioning observer algorithms:

Application to the cybership models in MC-Lab

Master's thesis in Engineering and ICT
Supervisor: Roger Skjetne
Co-supervisor: Emir Cem Gezer
June 2023

NTNU
Norwegian University of
Science and Technology

Eirik Midtun

Performance analysis of dynamic positioning observer algorithms:

Application to the cybership models in MC-Lab

Master's thesis in Engineering and ICT
Supervisor: Roger Skjetne
Co-supervisor: Emir Cem Gezer
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:	Midtun, Eirik
Field of study:	Marine cybernetics
Thesis title (Norwegian):	Analyse av tilstandsestimatorer for dynamisk posisjonering med anvendelse for cyberskipsmodellene i MC-Lab
Thesis title (English):	Performance analysis of dynamic positioning observer algorithms: Application to the cybership models in MC-Lab

Background

In MC-Lab we have a rather large fleet of cybership vessels. These serve us well in teaching, master theses, and PhD research. Recently, they have been updated with a uBuntu/ROS platform running on Raspberry Pi 4 controllers. However, the cybership models would benefit from implementation of improved DP observer (DPO) algorithms implemented in Python – to include also wave filtering capability and robust bias estimation as well as practical tuning procedures. The goal of this project is therefore to study and develop DPO algorithms relevant for the cybership models. A particular focus is on the observer algorithms based on the small-gain condition for stability. The study should also consider good tuning methods, such as auto-tuning based on derivative free optimization (DFO). The project aims to use the model called C/S Arctic Drillship (CSAD) for the implementation and testing. Tentatively, testing should be done on several models.

Scope of Work

1. Perform a background and literature review to provide information and relevant references on:
   - Cybership test platforms in MC-Lab.
   - ROS architecture and ROS-based control on Ubuntu.
   - Observer algorithms for DP with bias estimation and with/without wave filtering, incl. tuning methods/procedures.
   - Derivative-free optimization method(s) for observer tuning purposes.

   Write a list with abbreviations and definitions of terms and symbols, relevant to the literature study and project report.
2. Control system development:
   - Adapt and fit the observers to the ROS control system of CSE1 and/or CSAD for CSJ.
   - Prepare and discuss HMI functions and layout, from the perspective of DPO tuning, monitoring, and datalogging.
   - Propose and establish in ROS a Python-based DP model for Model-in-the-Loop (MiL) testing.
3. Study the observer designs incl. stability properties, establish tuning “rules of thumb”, and manually tune a set of proposed DPO algorithms, considering:
   - DPOs based on *small-gain condition* and *Lyapunov transformation* reported by Skjetne (2022).
   - Kalman filter(s) – discrete EKF and/or KF for LTV systems.
   - Nonlinear passive observer as presented by Fossen (2021).
4. Study DFO algorithms and conclude with a justification on one to use to autotune DPOs.
5. Perform a preliminary numerical analysis using the MiL simulations to get experienced with the DPO algorithms and their tuning. Propose also measures to use for evaluation of performance.
6. Decide on a test program and perform model-scale experiments at MC-Lab with and without waves to generate relevant datasets to run the observers on – using a default DP control law and baseline DPO. Testing the observers can then be done afterwards in a desktop study on each dataset. Compare and contrast the observers, considering:
   - Complexity of implementation and configuration.
   - Ease of tuning.
   - Performance of the observer using your proposed performance measures.
   - Dead-reckoning capability.

7. After making conclusions from Item 6., perform new experiments in MC-Lab, now with your selected DPOs in the control loop. Decide on some test cases, perform testing, compare performance, and discuss the results.

Specifications

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

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

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

The report shall be organized in a logical structure to give a clear exposition of background, problem/research statement, design/method, analysis, and results. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is not expected to be longer than 70 A4-pages, 100 B5-pages, from introduction to conclusion, unless otherwise agreed. It shall be written in English (preferably US) and contain the elements: Title page, project definition, preface (incl. description of help, resources, and internal and external factors that have affected the project process), acknowledgement, abstract, list of symbols and acronyms, table of contents, introduction (project background/motivation, objectives, scope and delimitations, and contributions), technical background and literature review, problem formulation or research question(s), method/design/development, results and analysis, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The contribution of the candidate shall be clearly and explicitly described, and material taken from other sources shall be clearly identified. Chapters/sections written together with other students shall be explicitly stated at the start of the chapter/section. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. natbib Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct, which is taken very seriously by the university and will result in consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed.

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

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

Supervisor: Roger Skjetne
Co-advisor(s): Emir Cem Gezer

Signatures: Roger Skjetne

Digitally signed by rskjetne
Date: 2023.03.21 15:27:00 +01'00'

Preface

This master's thesis is written as part of the study program Engineering and ICT at the Norwegian University of Science and Technology (NTNU). The thesis aims to analyze different dynamic positioning observer designs utilizing derivative free optimization for autotuning. During offshore operations, the DP systems are critical for maintaining the position and heading of marine vessels, and a reliable observer is needed for the safe and efficient operation of these.

All model-scale tests have been performed using the Department of Marine Technologies testing facility, the Marine Cybernetics Laboratory. This thesis presents a literature review of observers and derivative free optimization, a comparative study of different observer designs, and a methodology for autotuning observers. The observers are tested using the model scale vessel C/S Arctic Drillship, C/S Enterprise, and C/S Jonny, located in the marine cybernetics laboratory, demonstrating the practical application of the observers.

Abstract

The Marine Cybernetics Laboratory (MC-Lab) has a growing fleet of cybership vessels used in masters thesis and Ph.D. research. The thesis compares Dynamic Positioning Observer (DPO) algorithms relevant to the cybership models, including robust bias estimation and wave filtering capability. The method is implemented in the MC-Lab, focusing on the experimental model vessel C/S Arctic Drillship.

The thesis considers the Linear Time-Varying Kalman Filter (LTV-KF), the nonlinear passive observer, an observer design based on the small-gain principle, and an observer design based on the Lyapunov transformation. A particular focus on the observer algorithm based on the small-gain principle. Furthermore, the wave filtering properties are the only component of the nonlinear passive observer that is used. The observer algorithms' stability properties and manual tuning guidelines are investigated.

A Graphical User Interface (GUI) is added as an addition to the Human Machine Interface (HMI) of CSAD. The new GUI allows the operator to monitor and interact with the system. It includes realtime plotting of the actual, estimated, and desired position and heading. It also contains functionality to tune observers in realtime and is fully integrated with the CSAD ROS stack for use in Hardware-in-the-Loop (HIL) and model-scale testing.

The observers are tuned using Particle Swarm Optimization (PSO), not relying on the derivative of the objective function. The deterministic observers use the injection gains as optimization parameters, and the LTV-KF uses the process-noise covariance matrix Q. The integral of absolute error (IAE) is selected as the Performance Indicator (PI) used as the objective function.

The observers are tested using a 4-corner test and a 2-step 3 DOF transient maneuver defined as a combined motion test. The tests were implemented in Python and run using the Robotic Operating System (ROS) to function in realtime. Position and heading signal dropouts were used in combination with the combined motion test to test the deadreckoning capability of the observers.

When operating in calm waters, the performance of all the observers was similar. However, the thesis concludes that the LTV-KF is the best performing observer for CSAD regarding the accuracy and deadreckoning capability when operating in rough and very rough sea states. The observer design based on the small-gain principle was implemented for CSE and CSJ with a good position and heading estimation but was only tested in calm water.

Sammendrag

Det Marin Kybernetikk Laboratorium (MC-Lab) har en voksende flåte av kyberskip som brukes til masteroppgaver og doktorgradsforskning. Denne master oppgaven sammenligner Dynamisk Posisjonering Observer (DPO) algoritmer som er aktuelle for kyberskip-modellene, med egenskaper som robust estimering av bias og evne til bølgefiltrering. Metoden er implementert I MC-Lab, med spesielt fokus på modellbåten C/S Arctic Drillship.

Oppgaven vurderer Lineær Tids-Varierende Kalman Filter (LTV-KF), den ikke lineære passive observeren, et observer design basert på small-gain prinsippet og et observer design basert på Lyapunov transformasjon. Det legges spesielt vekt på observer algoritmen basert på small-gain prinsippet. Videre brukes bare bølgefiltreringsegenskapene til den ikke-lineære passive observeren. Stabilitetsegenskapene til observer algoritmene og retningslinjer for manuell justering av observer parameter undersøkes.

Et Grafisk Brukergrensesnitt (GUI) er lagt til i Menneske Maskin Grensesnittet (HMI) til CSAD. Det nye GUI-et lar operatøren overvåke og samhandle med systemet. Det inkluderer sanntidsplotting av faktisk, estimert og ønsket posisjon og retning. Det inneholder også funksjonalitet for å justere observere i sanntid og er fullt innlemmet med CSAD sin ROS-stakk for bruk i Hardware-in-the-Loop (HIL) og modell tester.

Observer parameterne blir justert automatisk ved hjelp av Partikkel Sverm Optimalisering (PSO), uten å være avhengig av den deriverte til målfunksjonen. De deterministiske observerne benytter justerings parameteren som optimaliseringsparametere, mens LTV-KF bruker kovariansmatrisen for prosess støy Q. Integral til Absolutt Feil (IAE) er valgt som Ytelsesindikator (PI) og brukes som målfunksjon.

Observerne blir testet ved hjelp av en 4-hjørne-test og en 2-trinns transiente manøver i 3 frihetsgrader som er definert som en kombinert bevegelse test. Testene ble implementert i Python og kjørt ved hjelp av Robot Operativ System (ROS) for å fungere i sanntid. Tap av posisjon og retnings signal ble brukt i kombinasjon med den kombinerte bevegelses testen for å teste observerne sin deadreckoning-evne.

Ved operasjon i rolig farvann, var prestasjonen til alle observatørene relativt lik. Imidlertid konkluderer oppgaven med at LTV-KF er observeren med best ytelse for CSAD med hensyn til nøyaktighet og deadreckoning-evne når man opererer i grov sjø. Observer designet basert på small-gain prinsippet ble implementert for CSE og CSJ med god posisjons- og retningsestimering, men ble bare testet i rolig vann.

Acknowledgement

There have been multiple contributors helping in different ways throughout the project. Firstly, I would like to give special thanks to my supervisor, Roger Skjetne, for his valuable guidance throughout the year.

I would also like to thank Senior Engineer at IMT, Robert Opland, and my co-advisor Emir Cem Gezer, for helpful advice regarding both software and hardware issues. Robert has spent multiple hours helping make C/S Arctic Drillship and the Qualisys system ready for testing in MC-Lab.

Lastly, I would like to thank the other students for their collaboration on CSAD, Marie Kongshaug, Harald Mo, and Stian Elvenes.

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviation

CO	*Center of Origin*
CSAD	*C/S Arctic Drillship*
CSE	*C/S Enterprise*
CSJ	*C/S Jonny*
DFO	*Derivative Free Optimization*
DOF	*Degree Of Freedom*
DP	*Dynamic Positioning*
DPO	*Dynamic Positioning Observer*
EKF	*Extended Kalman Filter*
FIR	*Finite Impulse Response*
GES	*Globally Exponentially Stable*
GUI	*Graphic User Interface*
HIL	*Hardware in the Loop*
HMI	*Human Machine Interface*
IADC	*Integral of Absolute Derivative Control*
IAE	*Integral of Absolute Error*
IAEC	*Integral of Absolute Error and Control*
IAEW	*Integral of Absolute Error multiplied by Work*
IMU	*Inertial Measurement Unit*
ISE	*Integral of Square Error*
ISS	*Input-to-State Stability*
ITAE	*Integral of Absolute Error multiplied by Time*
JONSWAP	*Joint North Sea Wave Project*
LF	*Low Frequency*
LTV	*Linear Time Varying*
LTV-KF	*Linear Time Varying Kalman Filter*
MC-Lab	*Marine Cybernetics Laboratory*
MoCap	*Motion Capture*
MSS	*Marine Systems Simulator*
NED	*North East Down*
ODM	*Observer Design Model*
PI	*Performance Indicator*

PS4	*Play Station 4*
PSO	*Particle Swarm Optimisation*
QTM	*Qualisys Track Manager*
ROS	*Robot Operating System*
SGO	*Surrogate Model Optimisation*
SVM	*Simulation Verification Model*
UGAS	*Uniformly Globally Asymptotically Stable*
UGES	*Uniformly Globally Exponentially Stable*
UGS	*Uniformly Globally Stable*
WF	*Wave Frequency*

Chapter 1

Introduction

1.1 Motivation

The field of autonomy has made significant progress in the last decade. Great advances in artificial intelligence and machine learning in parallel with computational power continue to grow rapidly. It has made fully autonomous vehicles a feasible concept, not just in the automotive industry but also in the maritime sector. Projects such as the fully autonomous container vessel Yara Birkeland, and autonomous ferries in Stockholm orchestrated by Zeabus, demonstrate the potential of autonomous vessels.

Although significant progress has been made, an autonomous marine system is highly reliant on a solid Dynamic Positioning (DP) system to perform its tasks. The Dynamic Positioning Observer (DPO) plays a central role in the DP system, providing state estimations that are accurate and reliable based on knowledge of the system and measurements of its input and output.

Observers are well-known in control theory, and there exist a number of papers presenting different observer algorithms and designs tested for DP. The Kalman filter was presented by Grimble et al. (1980) and the Extended Kalman Filter (EKF) by Balchen et al. (1976) and Saelid et al. (1983). Afterward, experimental results of a DP system using an EKF were presented by Tannuri and Morishita (2006). The nonlinear passive observer was presented by T. I. Fossen and Strand (1999) and tested in a full-scale experiment on a supply vessel. However, there are few papers with a careful comparison of the different observers used in DP. T. I. Fossen and Strand (1999) argued how the nonlinear passive observer had some benefits compared to the EKF, including having fewer tuning parameters and being proven Uniformly Globally Exponentially Stable (UGES).

This thesis aims to conduct a comparative study of different DP observers, evaluating their strengths and weaknesses using both numerical experiments and model-scale tests in the Marine Cybernetics Laboratory (MC-Lab). Specifically, the thesis will examine the performance of the four DPO: the Kalman filter, a nonlinear passive observer, an observer design based on the small gain principle, and an observer design based on a Lyapunov proof, not including a wave filter. The performance of the observers is measured

through different Performance Indicators (PI) concerning the observers' accuracy, ease of implementation, ease of tuning, and stability properties. In order to tune each observer as fairly as possible, a Derivative Free Optimization (DFO) is used. Manual tuning is only used to find initial conditions for the DFO.

1.2 Objectives and scope

The objective of the thesis is to study and compare observers and find a baseline observer that can be used in the MC-Lab for the model-scale vessels C/S Arctic Drillship (CSAD), C/S Enterprise (CSE), and C/S Jonny (CSJ). CSAD is prioritized in the comparison of the observer, while only minor resources is put to test CSJ and CSE. In addition, the objective includes developing an autotuning module to compare and tune the observers efficiently. The high-fidelity simulator package MCSimPython is used for experiments.

The scope of the thesis has been to

- Perform a background and literature review to provide information and relevant references on the MC-Lab, Robot Operating System (ROS), observer algorithms and DFO.
- Further development of the simulation environment MCSimPython. MCSimPython has been integrated with ROS using ROS wrappers and is improved to be able to perform experiments in realtime. Furthermore, a Graphic User Interface (GUI) is implemented for the Human Machine Interface (HMI) of the system with a focus on DPO tuning and monitoring.
- The observer designs stability properties are studied, and manual tuning guidelines are established for the following DPO designs: DPO design based on small-gain condition and Lyapunov transformation reported by R. Skjetne (2022), a Linear Time Varying Kalman Filter (LTV-KF), Nonlinear passive observer as presented by T. I. Fossen (2021). The two observers proposed by R. Skjetne (2022) is more carefully examined in this thesis than the LTV-KF and the nonlinear passive observer.
- The Particle Swarm Optimisation (PSO) algorithm is used for autotuning observers. It has been chosen based on Løvås (2019) work. It has been tweaked to be able to autotune DPOs.
- Measures to evaluate the performance of the DPOs are proposed. model-scale experiments at MC-Lab are performed with and without waves to generate relevant datasets to run the observers on – using a default DP control law and baseline DPO.
- Observers are compared and contrasted, considering a cost function, ease of tuning, complexity of implementation, and their deadreckoning capability.
- After making conclusions from the comparison, new experiments are performed in MC-Lab, with the best performing DPO in the control loop. In addition, the performance of the small-gain principle based observer design is tested on CSE and CSJ.

Limitations of the thesis:

1. model-scale tests were performed in the MC-Lab with CSAD, a 1:90 scale model.
   (a) Qualisys, the Motion Capture (MoCap) system, is very precise and has little noise. However, the cameras of the system cover only a small part of the basin, resulting in a small tracking area.
   (b) The duration of the test for the full-scale vessel would be 10 times longer.
   (c) Environmental forces such as wind and current were not present in the simulations or model-scale tests.
   (d) The thrusters of CSAD do not have a proper thrust to rpm mapping, and their performance is dependent on the voltage of the batteries. This affects the accuracy of the actual thruster load produced, given as input to the observer.
2. CSJ was not ready for model-scale experiments until the beginning of June due to a lack of parts, which resulted in little time to perform model-scale tests.
3. The fairness of the DFO tuning of stochastic versus deterministic observers is hard to determine due to structural differences. Resulting in an autotuning that might favor one over the other.

1.3 Contributions

- An autotuning module has been created in Python for offline tuning of both deterministic and stochastic observers.
- A modular motion control system has been implemented for CSAD in ROS Melodic using ROS wrappers. The observers deliver accurate estimates with a simple PID controller, a third-order reference filter with feasible velocity constraints, and a pseudo-inverse thrust allocation algorithm with magnitude constraints.
- A GUI based on QT for the HMI of the DP system has been implemented. It makes tuning of observers easier and gives the operator ability to interact, monitor and log the operation. It plots actual and estimated position and attitude, actual and command thruster loads, and desired position from the guidance system. There is also made visualization of each thruster's angle, position, and load magnitude.
- An observer has been tuned and adapted to the ROS control system of CSE and the new vessel CSJ, which was added to the fleet of MC-Lab in 2023.

1.4 Outline

Chapter 1 introduces the thesis to the reader and describes its motivation, objective, scope, and contributions.

Chapter 2 provides relevant background information regarding observers, DFO, ROS, and MC-Lab.

Chapter 3 presents the experimental setup, including an Simulation Verification Model (SVM) for the simulator and the Observer Design Model (ODM) used for observer design. Afterward, a description of the tests conducted, sea states used, and an overview of the software is presented.

Chapter 4 presents the problem formulation of the thesis.

Chapter 5 presents the DP system used for testing. This includes a third-order reference model, a PID controller, a pseudo-inverse thrust allocation algorithm, and HMI.

Chapter 6 presents all the DPOs and suggests tuning rules that provide boundaries for the gains that make the system stable. Observers included are the LTV Kalman filter, the nonlinear passive observer, a small-gain principle observer design, and a Lyapunov based observer design.

Chapter 6 details how DFO is used for autotuning of the observers. The optimization problem is formulated, boundaries are set, and PI is chosen.

Chapter 7 presents the results from the desktop study with autotuned observers performed on real data collected from MC-Lab using the LTV-KF as baseline observer. The study compares their accuracy in calm waters, in rough seas, and in very rough seas. Results from autotuning of both deterministic and stochastic observers are presented.

Chapter 8 presents results from further model-scale tests in the laboratory with the autotuned LTV-KF on CSAD and the small-gain based observer design on CSE and CSJ.

Chapter 9 presents the conclusion and further work.

Chapter 2

Background and literature review

The content of this chapter is reproduced from the author's project report (Midtun et al. 2022). It contains relevant background information for DPO, DFO, MC-Lab, and ROS.

2.1 Dynamic positioning observers

Filtering and state estimation are important features of a DP system due to the nature of the measurement signal. All states of the system are not necessarily measured, and the signals might be noisy, impacting the controller performance if no precautions are taken. The observer is a state estimator, both reconstructing non-measured states and filtering out noise.

This thesis focuses on conventional observer design for free-floating surface vessels operating at zero and low-speed tracking. It is assumed that the vessels are metacentric stable, which implies that there exist restoring forces in heave, roll, and pitch. The motions in these Degrees of Freedom (DOF) can be modeled as damped harmonic oscillators with a small amplitude and a zero mean. The observer design only models the motions in surge, sway, and yaw. The primary purposes of the observer in the positioning system are:

- *Position and velocity estimation.* Assuming that only the position and heading of the vessel are measured, the observer needs to estimate the velocity for the feedback controller. In addition, it should remove the measurement noise from the position and heading signals.
- *Bias estimation.* The observer model does not perfectly model all the effects on the vessel, resulting in a steady-state offset in the velocity estimates. The bias term is used to counteract the unmodeled effects and slowly varying environmental forces removing the offset. In addition, the bias can be included in a feedforward term in the controller.
- *Dead Reckoning.* The DP system gathers information about its position and heading with sensors. However, all equipment will fail in relation to a failure rate. A sudden

dropout of the control system might lead to dangerous situations when in operation if no substitute signal replaces it. The DP system enters dead reckoning when the position and heading measurements are no longer available, and the observer estimates a new signal by applying model-based filters. The signal will work as a replacement until the measured signal is restored.

- *Wave filtering.* The input signal to the feedback controller should not include the Wave Frequency (WF) motion. The observer excludes this from the position and heading signals by including a wave-induced motion model, obtaining a wave filter. A. Sørensen (2018), defines wave filtering as:

Definition 2.2.1: *Wave filtering can be defined as the reconstruction of the Low Frequency (LF) motion components from noisy measurements of position and heading by means of analog or digital filters. In addition to this, if an observer (state estimator) is used, noise-free estimates of the non-measured LF velocities can be produced. This is crucial in ship motion control systems since the WF part of the motion should* ***not*** *be compensated for by the positioning system. If the WF part of the motion enters the feedback loop, this will cause unnecessary wear and tear of the actuators and increase the fuel consumption.*

2.2 Derivative free optimisation

According to Audet and Kokkolaras (2016), BlackBox or DFO are often the only two viable tools for the engineer working on a simulation-based design. DFO refers to mathematical optimization methods that do not utilize the derivative of the objective function. These methods are used for problems where the partial derivatives are not defined or not available (Audet and Kokkolaras 2016). Gradient approximation is another method, but it may not be worth the effort due to the amount of computing power needed to approximate the gradient. Despite that, gradient-based methods often outperform DFO if the gradient is available. In the last 20 years, DFO has gained traction in the engineering design community and can surpass heuristics in some cases, with solutions rarely being characterized in terms of quality. Even though DFO requires a basic understanding of theory and knowledge of use cases, engineers can create great opportunities by taking advantage of these methods.

Løvås (2019) investigates several methods for DFO-based autotuning of DP controllers in his M.Sc. thesis. It includes parameterization of the control law, comparisons, and evaluation of different DFO algorithms such as PSO, Surrogate Model Optimization (SGO), and the Nelder-Mead simplex method.

The DFO objective is to minimize the objective function f(x), subject to some constraints g(x), which uses data collected from the real world for testing. The goal is to minimize the objective function by tuning the control parameters, x. Tuning as an optimization problem has these characteristics:

- **Nonlinear**: Gain parameterization is nonlinear. Hence, the objective function may be nonlinear as well.

- **Non-smooth**: The signal is contaminated by stochastic random noise, such as measurement noise, waves, jitter, etc., which might make the objective function non-smooth.

Furthermore, MATLAB recommends PSO and SGO to solve non-smooth and nonlinear problems (MATLAB 2022).

PSO is the superior algorithm for autotuning a controller in the MC-Lab and seems feasible for autotuning observers (Løvås 2019). In addition, PSO has a better convergence rate than SGO, is suitable for use with historical data, and its implementation is of lower complexity.

Table 2.1: Selection matrix for choosing optimization method. Courtesy: Løvås (2019).

	PSO	SGO
Convergence rate	Good/Very good	Good
Convergence	Good/Very good	Good
Generality	Good	Good
Global vs Local	Global/Local	Global
Complexity of Implementation	Low	Medium
Complexity of Parameterization	Medium	Low
Bounds	Yes	Yes
Ability to use historic data	Suitable	Very suitable

2.2.1 Particle swarm optimization

PSO is an efficient and accurate algorithm well suited to solve nonlinear, non-convex, continuous, discrete integer variable problems. This makes it a good fit to solve the optimization problem. According to Sahib and Ahmed (2016), PSO yielded the best performance when tasked to optimize control parameters for a PID controller. Kaliappan and Thathan (2014) make this same discovery regarding the performance of PSO.

The algorithm is inspired by how schools of fish and swarms of birds collectively search an area together. The swarm contains many particles, which all use their locally best-known position, the best-known position of the swarm, and its velocity to calculate how to move around. The number of particles in the swarm should be equal to the dimension of the optimization problem (Løvås 2019). The algorithm works as follows:

1. **Initialization**: The swarm, $S = \{x_1, x_2, ..., x_n\}$ contains all the different candidate solutions called particles. All the particles are usually initialized with random positions in search space and random velocities. However, in some cases, the position of the initial particles might be set to specific values to decrease the convergence time.

2. **Evaluation and updating**: The objective function is then used to calculate the fitness of each particle in the swarm. The local and global best-known positions are calculated, and the fitness of each particle is updated based on these positions and their velocities.

3. **Iteration**: Step 2 is repeated until a particle reaches the desired threshold and satisfactory convergence is reached.

Based on Løvås (2019) research regarding DFO, PSO is chosen as the autotuning optimization algorithm in this thesis, and the optimization problem is further described in chapter 7.

2.3 Robot operating system

The system's architecture is based on ROS. ROS is an open-source framework with tools and libraries that help developers and researchers build and reuse code to create robotic applications. This includes drivers, advanced algorithms, and developing applications to simplify the process of making a robotic system. In addition, it has a global community of engineers, computer scientists, and hobbyists who make robotics accessible and available for everyone.

ROS is a middleware. It handles services such as hardware abstraction, package management, and message-passing between processes. Furthermore, it provides a publish-subscribe messaging infrastructure supporting construction of distributed computing system. It also provides supporting functionality to build and maintain the application.

ROS-based processes are represented as nodes in the graph architecture. The nodes connect through topics, which they use to interact through publish-subscribe messages-passing. The nodes communicate through service calls and parameter services connected to the ROS master. The ROS master is a parameter server that keeps track of all active nodes and the topic to which each node is connected. It establishes a peer-to-peer communication network between the nodes. Figure 2.1 illustrates a system containing two nodes communicating through a topic. In this case, NODE1 publishes messages to the TOPIC, and NODE2 subscribes to the same topic. The publish-subscribe communication follows the structure of event-driven programming, where the subscribing node will act when a message is published on the topic.

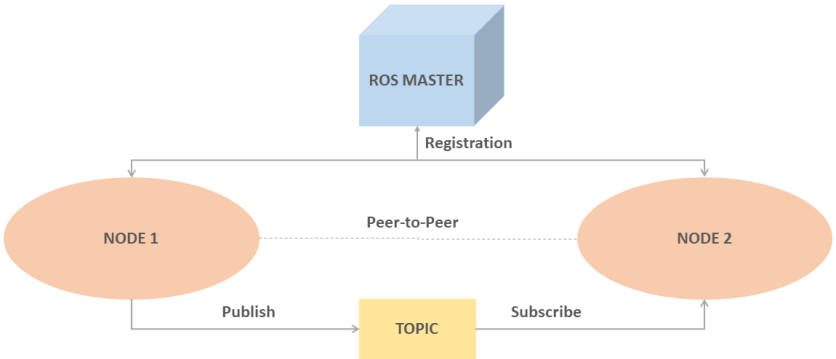


Figure 2.1: ROS architecture concept.

2.4 Marine cybernetics laboratory

MC-Lab is a small wave basin. It was created from the old storage basin for the towing tank, back when models were made of paraffin wax and needed wet storage. Models are no longer made of paraffin wax, yielding an empty basin to be repurposed into the MC-Lab in the 1990s. The basin has a length of 40m, a breadth of 6.45m, and a depth of 1.5 m (NTNU 2022). Today the lab is mainly used to perform model tests by M.Sc. and Ph.D. candidates.

The laboratory has a suite of sensors and equipment to perform several types of tests. This includes tools to generate environmental forces, towing carriage, and a MoCap system.

Wave maker

The MC-Lab is equipped with a 6m width single paddle wave maker that is located at the short end of the basin. It has the ability to produce both regular and irregular waves with height and period (NTNU 2022):

- Regular waves H $<$ 0.25, T = 0.3 – 3 s.
- Irregular waves Hs $<$ 0.15 m, T = 0.6 – 1.5 s.
- Speed limit: 1.2 m/s.

It can be used to create different wave spectra such as Joint North Sea Wave Project (JONSWAP) and Pierson-Moskowitz.

Qualisys motion-capture

To capture movement in 6DOF, the lab is equipped with the Qualisys MoCap system, which consists of two parts. Firstly, the Oqus infrared cameras. They track infrared reflective spheres positioned on the models. Today the lab has four cameras. However, it is currently being fitted with additional cameras to increase its effective tracking area. Secondly, the Qualisys Track Manager (QTM) is run on a dedicated computer. It performs triangulation and broadcasts the position of the model on the local network.

2.5 C/S Arctic Drillship

In 2015 Jon Bjørnø created the CSAD (J. Bjørnø 2016). CSAD is a 1:90 scale model of the Arctic drillship created by Inocean for Statoil. It was developed to perform tests on thruster-assisted position mooring systems. The dimensions are displayed in Table 2.2.

Table 2.2: Dimensions CSAD. Courtesy: J. Bjørnø (2016).

Description	**Data**
Length	2.578$[m]$
Breadth	0.440$[m]$
Depth (moulded)	0.211$[m]$
Draft (design)	0.133$[m]$

The model is equipped with six azimuth thrusters and a rotatable circular turret. Figure 2.2 illustrates the configuration of the thrusters. They are rotated by continuous servos that have shafts that spin continuously. The Center of Origin (CO) is placed at the center of the turret in the waterline when the vessel is fully ballasted. The turret enables it to connect four mooring lines and a riser. The model is primarily used for stationkeeping and low-speed operation, as its size makes it impractical for medium and high-speed maneuvering operations in the MC-Lab.

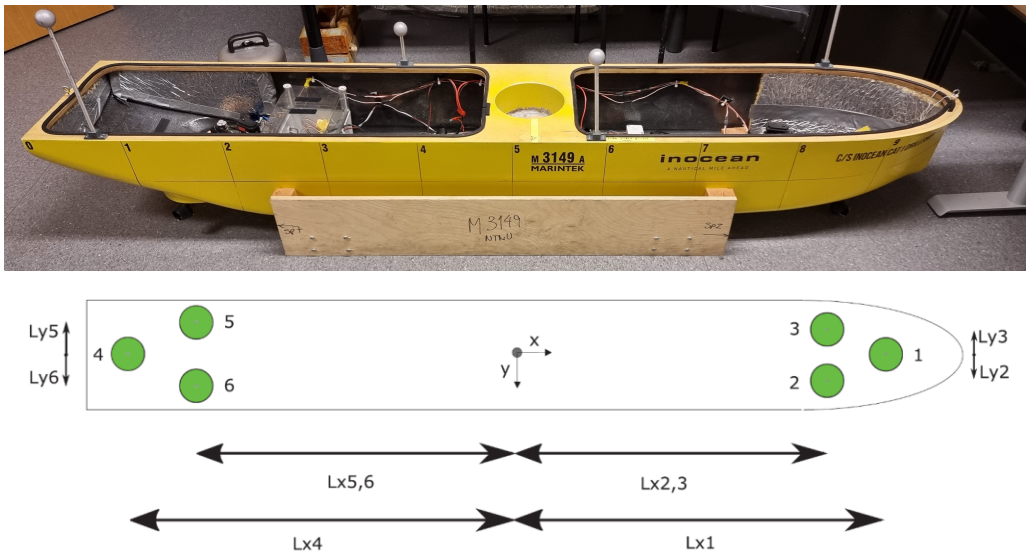


Figure 2.2: CSAD thruster configuration. Courtesy: Jon Bjørnø et al. (2017).

Chapter 3

Experimental setup

The experimental surface vessel, CSAD, is used both in simulation and in model-scale tests in the MC-Lab. Both CSAD and its operational environment, MC-Lab, are described in section 2.4. Significant effort has been put into creating a GUI for the vessel and creating ROS wrappers for all the modules in CSADs DP system, which previously has been implemented in the MCSimPython package and not in ROS. The content of this section is reproduced from the author's project report (Midtun et al. 2022).

3.1 Vessel models

This section contains the mathematical models for a low-speed vessel, including a SVM used to simulate the vessel and an ODM.

The mathematical modeling is based on the assumption that roll- and pitch angles are small, which is a good approximation for most conventional ships (T. I. Fossen 2021) operating at low speed.

The reference frames used in this chapter are the Basin frame and Body-fixed frame. The Basin frame is an inertial frame that is local earth fixed where x, y, and z points, respectively, along the long side of the basin, along the short side of the basin, and down directions. The basin frame replaces the North East Down (NED) as a local Earth-fixed reference frame which is frequently used in DP systems, to give the axes more intuitive directions when performing tests in MC-Lab. In the body-fixed frame, the x-axis is directed aft-to-fore, the y-axis is directed port-to-starboard, and the z-axis points down.

3.1.1 Simulation Verification Model

The SVM is a high-fidelity model that precisely describes the motions of the vessel. The SVM takes in the thruster command angles α and thruster input u and produces the ship's pose η. The SVM includes vessel dynamics, wave loads, thruster dynamics, and a measurement model.

Vessel dynamics

The vessel dynamics are derived from the nonlinear 3DOF ship model in (Lyngstadaas 2018). As described in further work, the numerical values for the model parameters are not yet found for CSJ. Since neither simulations nor experiments will use current, $\nu_c = 0$ assumes no current. The SVM

$$\dot{\eta} = R(\psi)\nu \tag{3.1}$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu = \tau_{wave} + \tau, \tag{3.2}$$

where $\eta = (x, y, \psi)$ is the pose vector, $\nu = (u, v, r)$ is the body-fixed velocity vector, $\tau = (F_{surge}, F_{sway}, M_{yaw})$ is the command forces, $\tau_{wave} = (F_{surge,wave}, F_{sway,wave}, M_{yaw,wave})$ is the wave forces,

$$R(\psi) := \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3x3} \tag{3.3}$$

is the rotational matrix, and $M := M_{RB} + M_A \in \mathbb{R}^{3x3}$ is the mass matrix with,

$$M_{RB} := \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \in \mathbb{R}^{3x3}, \quad M_A := \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{\nu}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{\nu}} & -N_{\dot{r}} \end{bmatrix} \in \mathbb{R}^{3x3} \tag{3.4}$$

where x_g is the distance from the origin to the center of mass, and I_z is the inertia about the z-axis.

$C := C_{RB} + C_A \in \mathbb{R}^{3x3}$ is the Coriolis and centripetal matrix, defined by

$$C_{RB} := \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ -m(x_g r + v) & -mu & 0 \end{bmatrix} \in \mathbb{R}^{3x3}, \tag{3.5}$$

$$C_A := \begin{bmatrix} 0 & 0 & -c_{A,13}(\nu) \\ 0 & 0 & c_{A,23}(\nu) \\ c_{A,13}(\nu) & -c_{A,23}(\nu) & 0 \end{bmatrix} \in \mathbb{R}^{3x3}. \tag{3.6}$$

$D := D_L + D_{NL} \in \mathbb{R}^{3x3}$ which is positive definite and should preferably be symmetric, that is,

$$D_L := \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \in \mathbb{R}^{3x3}, \quad D_{NL} := \begin{bmatrix} d_{NL,11} & 0 & 0 \\ 0 & d_{NL,22} & d_{NL,23} \\ 0 & d_{NL,32} & d_{NL,33} \end{bmatrix} \in \mathbb{R}^{3x3}, \tag{3.7}$$

where

$$d_{NL,11} = -X_{|u|u|}|u| - X_{uuu}u^2 \tag{3.8}$$
$$d_{NL,22} = -Y_{|v|v|}|v| - Y_{|r|v|}|r| - Y_{vvv}v^2 \tag{3.9}$$
$$d_{NL,23} = -Y_{|r|r|}|r| - Y_{|v|r|}|r| - Y_{vvv}r^2 - Y_{ur}u \tag{3.10}$$
$$d_{NL,32} = -N_{|v|v|}|v| - N_{|r|v|}|v| - N_{vvv}v^2 - N_{uv}u \tag{3.11}$$
$$d_{NL,33} = -N_{|r|r|}|r| - N_{|v|r|}|v| - N_{rrr}r^2 - N_{ur}u \tag{3.12}$$

To accommodate the Munk moment, the damping terms,

$$Y_{ur} = X_{\dot{u}} \in \mathbb{R}, \quad N_{uv} = -(Y_{\dot{v}} - X\dot{u}) \in \mathbb{R}, \quad N_{ur} = Y_{\dot{r}} \in \mathbb{R} \tag{3.13}$$

are added to the damping matrix.

Wave loads

In the simulator, waves are generated using the python package MCSimPython created by Hyggen et al. (2023). It is an adaptation of parts of the Marine Systems Simulator (MSS) toolbox available in MATLAB. It produces waves from the JONSWAP and Modified Pierson Moskowitz wave spectra.

Thruster dynamics

The thruster dynamics takes in the actuator input $u \in \mathbb{R}^n$ and the thruster angle $\alpha \in \mathbb{R}^n$ from the thruster allocation and calculates the forces applied to the vessel,

$$\tau(\alpha, u) = B(\alpha)K_T u \tag{3.14}$$

where $K_T := diag(k_1, k_2, \ldots, k_n) \in \mathbb{R}^{nxn}$ is the thrust coefficient matrix, n is the number of thrusters and the thrust configuration matrix is

$$B(\alpha) = \begin{bmatrix} cos(\alpha_1) & cos(\alpha_2) & \ldots & cos(\alpha_n) \\ sin(\alpha_1) & sin(\alpha_2) & \ldots & sin(\alpha_n) \\ N_1 & N_2 & \ldots & N_n \end{bmatrix}, \tag{3.15}$$

$$N_i = L_{x,i}sin(\alpha_i) - L_{y,i}cos(\alpha_i), \qquad i \in 1, 2, \ldots, n, \tag{3.16}$$

where $L_{x,i}$ and $L_{y,i}$ describe the distance from the CO and the thruster i. The rate limitations of the azimuth thrusters are not taken into account.

Measurement model

A simple measurement model is used by adding zero-mean Gaussian white noise.

$$\eta_m = \eta + v \tag{3.17}$$

where $v \in \mathbb{R}^3$ is the Gaussian white noise. The measurements from QTM are precise and the error is in the magnitude of 1mm. Hence, the white noise is set to $v := [0.001, 0.001, 0.001]^T$ (Løvås 2019).

3.1.2 Observer Design Model

The ODM is used for LF controller- and observer design. It captures the main dynamics of the system. Low-speed maneuvers are assumed, in which Coriolis and centripetal terms and the nonlinear damping effects are omitted (Værnø et al. 2019). The dynamics of the model are described by

$$\dot{\eta} = R(t)\nu \tag{3.18}$$

$$M\dot{\nu} = -D_L\nu + R(t)^T b + \tau, \tag{3.19}$$

where $\eta = (x, y, \psi)$ is the LF pose vector, $\nu = (u, v, r)$ is the LF body-fixed velocity vector, $b \in \mathbb{R}^3$ the bias vector, $M \in \mathbb{R}^{3x3}$ is the mass matrix, $D_L \in \mathbb{R}^{3x3}$ is the linear damping matrix, $\tau = (F_{surge}, F_{sway}, M_{yaw})$ is the command forces, and the rotational matrix

$$R(t) := \begin{bmatrix} cos\psi_y(t) & -sin\psi_y(t) & 0 \\ sin\psi_y(t) & cos\psi_y(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3x3}, \tag{3.20}$$

where $\psi_y(t) \approx \psi(t)$ assuming $\psi_\omega(t)$ is negligible (Værnø et al. 2019).

Bias model

The bias model is slowly varying and body-fixed. It accounts for 2. order wave loads, slowly varying forces and moments due to the wind- and current loads, and modeling errors. It is modeled as a 1. order Gaussian-Markov process (T. I. Fossen 2021), by

$$\dot{b} = -T_b^{-1} b + E_b \omega_b, \tag{3.21}$$

where $T_b := diag(t_{b,x}, t_{b,y}, t_{b,\psi})$ is the bias time constants, $E_b \in \mathbb{R}^{3x3}$ is the scaling matrix, and $\omega \in \mathbb{R}^3$ is the zero-mean Gaussian white noise vector.

Wave frequency control design model

The WF motions are modeled as the damped harmonic oscillator,

$$\dot{\xi}_\omega = A_\omega \xi_\omega + E_b \omega_\omega \tag{3.22}$$

$$\eta_\omega = C_\omega \xi \tag{3.23}$$

$$C_\omega := \begin{bmatrix} 0_{3x3} & I_{3x3} \end{bmatrix}, \quad A_\omega := \begin{bmatrix} 0_{3x3} & I_{3x3} \\ -\Omega^2 & -2\Lambda\Omega \end{bmatrix} \in \mathbb{R}^{6x6}, \quad E_\omega := \begin{bmatrix} 0_{3x3} \\ K_\omega \end{bmatrix}, \tag{3.24}$$

where $\Omega := diag(\omega_1, \omega_2, \omega_3)$, $\Lambda := diag(\lambda_1, \lambda_2, \lambda_3)$ and $K_\omega := diag(k_{\omega 1}, k_{\omega 2}, k_{\omega 3})$. The pair (A_ω, C_ω) is observable (R. Skjetne 2022).

Measurement model

The measurement is modeled as

$$y = \eta + C_\omega \xi + \epsilon_y \tag{3.25}$$

where $\epsilon_y \in \mathbb{R}^3$ is the zero-mean Gaussian white noise vector.

Observer design models

By combining (3.19) - (3.25), we get the Linear Time Varying (LTV) ODM. It applies the internal model principle, including the WF output disturbance as a part of the plant modeled as a harmonic oscillator. The ODM is given by LF kinematics, kinetics, bias, and WF, that is

$$\begin{aligned} \dot{\xi}_\omega &= A_\omega \xi_\omega + E_b \omega_\omega \\ \dot{\eta} &= R(t)\nu \\ \dot{b} &= -T_b^{-1} b + E_b \omega_b \\ M\dot{\nu} &= -D_L \nu + R(t)^T b + \tau \\ y &= \eta + C_\omega \xi. \end{aligned} \tag{3.26}$$

3.2 Model-scale tests

Three types of tests are performed in this thesis. Firstly, a baseline observer and control system is used in the MC-Lab to collect data to be used in the desktop comparison study. After making conclusions in the desktop study, a DPO is chosen for final model-scale tests. The final tests consist of a 4-corner test with one 10 to 20-second continuous dropout of position signal between each setpoint. In the third test, the signal dropouts are frequent but with a low duration of 1-5 seconds. The tests performed on CSE and CSJ are not described in this chapter.

Combined motion test

A combined surge-sway-yaw motion test is performed in MC-Lab when collecting data to compare the performance of all DPO. The test is chosen both for its simplicity and to capture the behavior of the vessel in surge, sway, and yaw. The vessel starts the test with a heading of 30 degrees. The guidance system sets the following setpoints:

1. Position and heading change: 1m in the negative Y direction in basin frame and a heading change of 90° counterclockwise, testing combined surge-sway-yaw movement.
2. Position and heading change: 1m in the positive Y direction in basin frame and a heading change of 90° clockwise, testing combined surge-sway-yaw movement.

When the test is complete. the vessel returned to its original position, ready for another test. The test will be performed in all sea states discussed in section 3.3. Figure 3.1 illustrates the the test

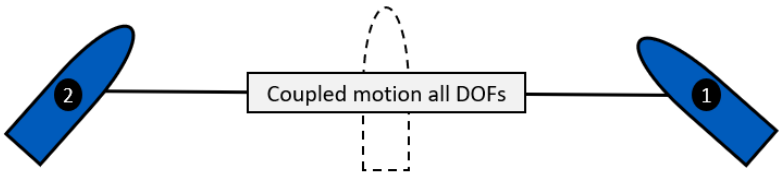


Figure 3.1: Combined motion test, used to collect data for the desktop study. The vessels start position is identical to setpoint 2.

4-corner test

A 4-corner test, based on Roger Skjetne et al. (2017), is used in order to test the performance of the best performing observer in the final tests. First, the vessel is initialized in DP, maintaining its position and heading of 0°, testing the steady state of the vessel with no motion. The start point of the test is the same as setpoint 5. The following setpoints are given to the guidance system:

1. Position change: 1m in the positive X direction in the basin frame, testing a pure surge ahead vessel movement.
2. Position change: 1m in the positive Y direction in the basin frame, testing a pure sway movement in the starboard direction.
3. Heading change: 30° clockwise, testing a pure yaw movement while maintaining position.
4. Position change: 1m in the negative X direction in the basin frame, testing a combined surge-sway movement while maintaining the heading.
5. Position and heading change: 1m in the negative X direction in basin frame and a heading change of 30° counterclockwise, testing combined surge-sway-yaw movement.

After completing a test, the vessel returns to its original position and heading, ready for another test at the same spot along the same track. The setpoints are placed quite close to each other in order to get a larger portion of the test with a transient response from the system. It is chosen to see if the observer is able to track the position of the vessel in the transient state when the vessel is moving in pure and combined surge, sway, and yaw movements. The test is visualized in Figure 3.2.

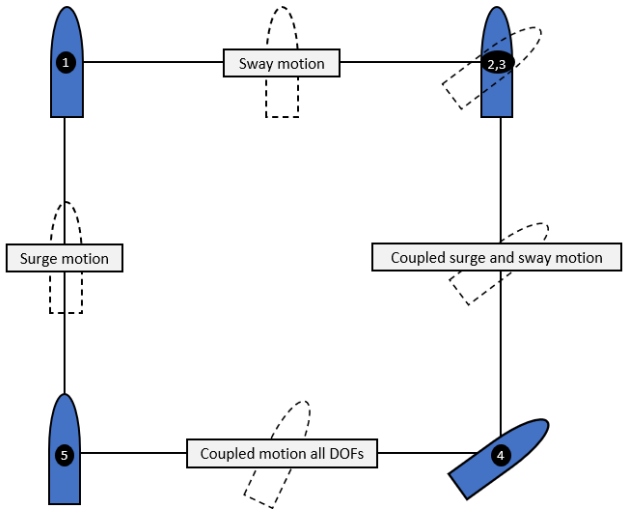


Figure 3.2: 4-corner test used in the final test by the best performing observers. The vessels start position is the same as setpoint 5.

3.3 Sea states

All tests are performed without any current or wind in both calm, rough, and very rough sea states. The wave characteristics are chosen to fit Perrault (2021) definition of sea states.

The tests performed in rough sea state have a full-scale significant wave height, $H_S = 2.7m$, and a peak wave period, $T_p = 6.6s$. The very rough sea state has a full-scale significant wave height, $H_S = 4.5m$, and a peak wave period, $T_p = 10.4s$. Table 3.1 illustrates the model-scale sea states, with Froude scaling, used for Hardware in the Loop (HIL) and model-scale testing.

Table 3.1: Sea states scaled down for model vessel CSAD, used for testing in both HIL and model-scale tests.

Sea State nr.	Description of sea	Significant wave height (H_S)[m]	Peak wave period (T_p)[s]
0	Calm water	0	0
1	Rough sea	0.03	0.7
2	Very rough sea	0.05	1.1

Both of these sea states are within the capabilities of the wavemaker in MC-Lab and the wave component of the MCSimPython package.

3.4 Software

When conducting tests in MC-Lab, there are two computers in use. Computer one is an embedded computer onboard the vessel running the drivers for the actuators and the Qualisys node. The second computer on land runs the control system and GUI. The computer on land runs the remaining modules and provides the operator with an interface showing the position and heading of the vessel. Furthermore, it provides tools to help the operator keep an overview of tests performed and the system's state. The operator is able to listen to all messages sent between nodes onboard the vessel from the second computer on land. All functionalities of the GUI are described later in this section.

All HIL and model-scale tests are performed using ROS. In order to be able to perform offline tuning and simulations, the backend of the system is programmed in Python, and the ROS nodes are created using a ROS wrapper. This enables users to run simulations in simulation time without ROS on their system if needed. Furthermore, with the same ROS wrapper interface, it is easy to switch between various modules underneath. Figure 3.3 illustrates how the MCSimPython package is integrated into CSADs ROS stack using a ROS wrapper interface(Hyggen et al. 2023).

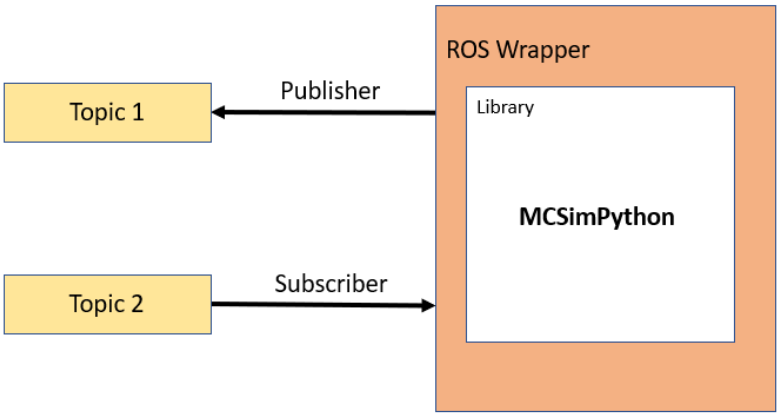


Figure 3.3: ROS wrapper used to integrate features from the MCSimPython package to CSADs ROS stack.

It is possible to run the system in simulation time instead of realtime, performing simulations much faster. This is done using the MCSimPython package without the ROS wrapper. However, these tests have no messages sent between nodes and are run on a single computer, therefore, problems regarding lost messages and latency are not included. However, these tests are helpful to rapidly test each separate module and find baseline parameters for the system.

HIL ROS architecture

Figure 3.4 illustrates a graph of the system's ROS architecture. It depicts all nodes as circles and topics as rectangles. The HIL node, marked in green, runs on a separate computer, simulating the vessel, when performing HIL tests. All other nodes run on the

same computer that is used when performing model-scale tests in MC-Lab. The arrows between nodes and topics refer to their relation. An arrow from node to topic refers to the node publishing on the topic. Likewise, an arrow from a topic to a node refers to the node subscribing to the topic. The message types used are all standard ROS message types.

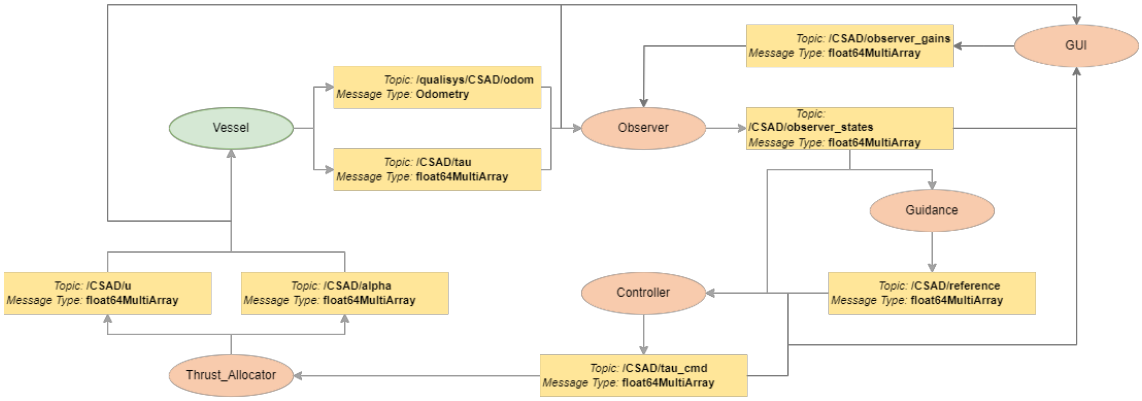


Figure 3.4: Graph of the ROS architecture of the software used in HIL tests.

The system depicted in Figure 3.4 is run in realtime and contains the following nodes:

1. *Vessel.* It contains a model of CSAD based on the SVM proposed in subsection 3.1.1. It stimulates the vessel and takes thruster forces as input giving the vessel's updated position and actual thrust load produced as output. It can be activated to perform various HIL tests without using the MC-Lab, testing the rest of the system on the hardware used in MC-Lab. This node is not active when performing model-scale tests at MC-Lab, where the Qualisys MoCap system and thruster actuator replace it.

2. *Observer.* The observer takes in the vessel's position and command forces. It estimates the unmeasured state velocity based on the measured position and removes noise from the position measurements. Furthermore, it outputs the estimated position, speed, observer bias, and WF motions. In the future, it is possible to use the measured acceleration using Inertial Measurement Units (IMU), as well as position. Note that a fault tolerance node that takes in the measurements from the sensors and detects and handles signal errors is not implemented. The only signal fault that is implemented is a frozen signal which will activate dead reckoning.

3. *Guidance.* The guidance node takes in the observer states and calculates the desired position and velocity of the vessel, which is sent to the controller. A third-order reference filter is implemented.

4. *Controller.* Both a joystick and PID controller is implemented. The controller takes in the estimated and desired position and velocity, calculating the command forces of the vessel.

5. *Thrust_alloc.* The thrust allocator takes in the command forces and calculates the force and angle each thruster should produce. The thrust allocator currently uses a Moore-Penrose pseudoinverse to calculate the thrust to the six azimuth thrusters.

6. *GUI.* The GUI node subscribes to all topics in the system. It is able to publish gains to the observer for tuning purposes. The functionality is easily transferred to realtime tuning of controllers as well.

MC-Lab ROS architecture

When performing model-scale tests in MC-Lab, small modifications are made to the system compared to HIL tests. The changes in the ROS architecture between the HIL tests and MC-Lab tests are listed below.

1. Vessel node replaced with Qualisys and Thruster_actuator nodes.
2. *Qualisys.* The Qualisys node tracks the position of the vessel with cameras. It does not subscribe to any topics and sends the pose and attitude of the vessel to the observer.
3. *Thrust_actuators.* Thrust_actuators takes in the actuator input u and actuator angle α from the thrust allocation node. It then converts the signals to pulse width modulation signals and sends them to the servo and azimuth motors. In addition, it sends the actual thrust produced by the thrusters to the observer.

The embedded computer onboard the vessel runs the two nodes added, marked with green in Figure 3.5. All other nodes run on the laptop as in HIL tests. The laptop runs the same nodes in HIL and model-scale tests. The only difference is that the embedded computer running the MoCap and thruster actuators is switched with a PC that simulates the vessel in HIL tests. All messages are still transferred from the laptop to the embedded computer or the computer simulating the vessel depending on the test being performed in MC-Lab or as a HIL test. The ROS architecture used for model-scale tests at the MC-Lab is illustrated in Figure 3.5.

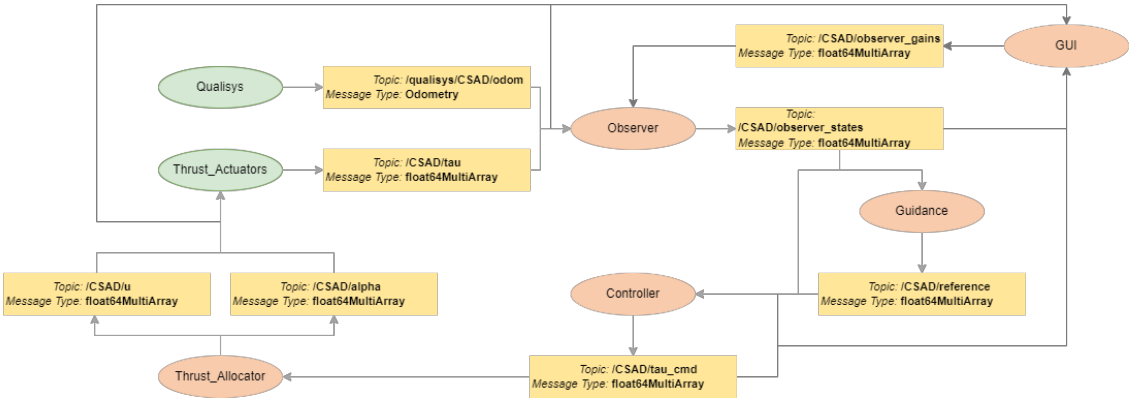


Figure 3.5: Graph of the ROS architecture for the software used in model-scale.

Chapter 4

Problem formulation

The overall goal of this thesis is to compare different DPO and find a baseline observer that can be used in the MC-Lab for the model-scale vessel CSAD. Furthermore, an autotuning module is implemented in order to compare and tune the observers efficiently. The small-gain observer design should also be tested on CSE and CSJ.

4.1 Performance indicators

In order to compare the different observers, PI need to be selected. Both quantitative and qualitative PI are considered in this study after completing a preliminary numerical analysis in the simulator. In addition, a quantitative PI is needed as a cost function when autotuning the observers. It needs to consider both the accuracy of the position and velocity estimates. The Observers will be evaluated by the following PIs.

1. Performance of the accuracy of the observer using a quantitative performance measure.
2. Complexity of implementation and configuration.
3. Ease of tuning.
4. Dead-reckoning capability.
5. Having a WF filter.

4.2 Stability of observers and manual tuning guidelines

The observer designs, including stability properties, should be studied in order to find some rule-of-thumb tuning guidelines to be used for manual tuning. The gains that are found through manual tuning will be used as the initial condition for the autotuning algorithm. This should consider a discrete Kalman filter for LTV systems, a Nonlinear

passive observer as presented by T. I. Fossen (2021), an observer based on the small-gain principle, and one based on a Lyapunov proof reported by R. Skjetne (2023). Stability boundaries for the observer based on small-gain principle should also be compared to the gains found with the autotuning algorithm.

4.3 Baseline control system

A baseline control system needs to be implemented in ROS for CSAD. CSAD already has a Python package called MCSimPython that includes most modules needed, but the modules are not implemented in ROS. ROS wrappers should be used to enhance the modules of the MCSimPython package to be usable in ROS. In addition, to adding observers, a ROS wrapper for a controller, guidance system, and thrust allocator needs to be implemented before CSAD can run in realtime on ROS.

4.4 Autotuning

In order to achieve a fair comparison of the observers, an autotuning module is needed. The DFO algorithm PSO is chosen based on Løvås (2019) work with autotuning of controllers. The algorithm needs to be customized in order to tune observers. An objective function needs to be chosen, and the observer gains need to be parameterized in order to fit the algorithm. The DFO should be able to tune both the deterministic observers and the stochastic Kalman filter. The autotuning algorithm should have rather loose bounds compared to the bounds found from the small-gain principle stability proof. Afterward, the behavior of the convergence of the autotuner can be compared to the limits of the gains found by the stability analyses.

Multiple tests need to be performed in the MC-Lab with and without waves in order to generate relevant datasets. A default DP control law and baseline DPO are needed to collect this data. Afterward, the observers should be tested in a desktop study to compare and contrast them considering all PI. Furthermore, data collected in MC-Lab needs post-processing in order to produce the "actual" velocity, which is not measured by the MoCap system during tests.

4.5 Graphical user interface

CSAD's HMI lacks a proper GUI, which is needed to monitor, log, and interact with the system. Furthermore, functionality to improve the tuning of observers should be prioritized. This includes realtime tuning tools and realtime plotting of measured, desired, and actual position and heading of the vessel. The GUI needs to be integrated into the CSAD ROS stack in order to be used in the realtime system.

Chapter 5

Control system

This chapter describes the different parts of the DP system of CSAD. It included a guidance system, a PID controller, and a thrust allocator. Note that the observers are described in chapter 6 and therefore not included in this chapter.

5.1 Guidance system

Guidance is the action or the system that continuously computes the reference (desired) position, velocity, and attitude of a marine craft to be used by the motion control system. These data are usually provided to the human operator and the navigation system (T. I. Fossen 2021).

The guidance system is a trajectory generator, producing a smooth trajectory from one setpoint to the next one, as well as acceleration and velocity. The reference model is of third order, consisting of a lowpass filter cascaded with a mass-damper-springer system (T. I. Fossen 2021). The author considers the choice of the reference model to not be conclusive for the scope of this master thesis. More details can be seen in T. I. Fossen (2021, Chapter 12).

However, it should be noted that CSAD is able to operate within a feasible set of velocities illustrated in Figure 5.1 (Nørgaard Sørensen et al. 2018). Speed limits for the reference model chosen as surge speed limit, $|u_{max}| < 0.25$ m/s, sway speed limit, $|v_{max}| < 0.075$ m/s, and yaw rate limit, $|r_{max}| < 3$ deg/s. It is a simplification of the findings in Nørgaard Sørensen et al. (2018) and can in some rare cases lead to unfeasible velocities for CSAD, resulting in the vessel struggling to keep up with the reference model.

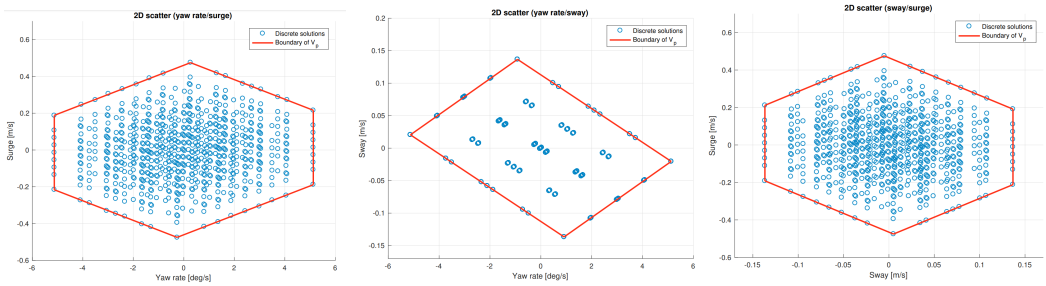


Figure 5.1: Possible combinations of surge speed, sway speed, and yaw rate, with respect to the actuator magnitude saturation limits. Courtesy: Nørgaard Sørensen et al. (2018).

5.2 PID controller

The control objective is a tracking problem, controlling the vessel's position to the desired time-varying trajectory $\eta_d(t)$ with the desired velocity $\nu_d(t)$ produced by the guidance system. This thesis uses a simple PID control law

$$\tau_{PID} = -K_p\tilde{\eta} - K_i\Sigma(\tilde{\eta}) - K_d\tilde{\nu} \tag{5.1}$$

where $\tilde{\eta} = R^T(\psi_d)(\hat{\eta} - \eta_d)$ is the three dimensional positional error vector and $\tilde{\nu} = \hat{\nu} - R^T(\psi_d)\dot{\eta}_d$ is the error velocity vector, $\Sigma(\tilde{\eta})$ is the positional error integrated using the Euler method and $K_p, K_i, K_d \in \mathbb{R}^{3\times3}$ are the control gains matrices.

The gains of the controller used in MC-Lab were found using the standard PID tuning rules presented in T. I. Fossen (2021, Chapter 12), implemented in Python and included in the ROS system with a custom ROS wrapper. Note that no integral windup is implemented, which might result in the accumulation of a significant error when the desired position is far from the estimated position.

5.3 Thrust allocation

CSAD is equipped with six azimuth thrusters, distributed symmetrically onto the x-axis illustrated in Figure 5.2. The position of each thruster is relative to the vessel's CO is shown in Table 5.1. The azimuth thrusters have the capability to rotate freely and produce force in any direction in the XY-plane, making CSAD over-actuated.

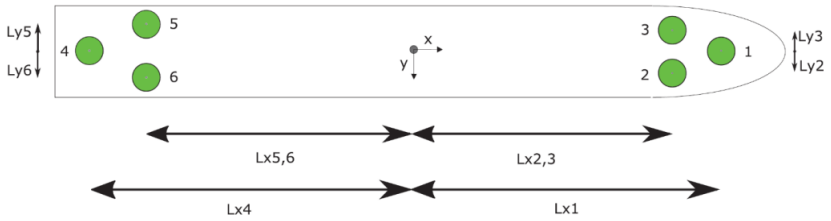


Figure 5.2: Thruster configuration of CSAD. Courtesy: Jon Bjørnø et al. (2017).

Table 5.1: CSAD fixed thruster position and angle in body frame.

Thruster	l_{xi}	l_{yi}	α_i
Azimuth 1	1.0678	0.0	0
Azimuth 2	0.9344	0.11	$\frac{\pi}{4}$
Azimuth 3	0.9344	-0.11	$-\frac{\pi}{4}$
Azimuth 4	-1.1644	0.0	π
Azimuth 5	-0.9911	-0.1644	$-\frac{3\pi}{4}$
Azimuth 6	-0.9911	0.1644	$-\frac{3\pi}{4}$

For simplified and predictable thrust allocation, in previous experimental setups, the orientations of the thrusters have been fixed, hence, creating a constant relationship between the command forces, $\tau \in \mathbb{R}^3$, and the actuator input $u \in \mathbb{R}^6$ (Lyngstadaas 2018). The vessel suffered thruster malfunctions on Azimuth 3 and 5. In order to improve the efficiency of the remaining thrusters, a thrust allocator with freely rotating thrusters was implemented. The relationship between the trust load vector τ and the actuator inputs u, shown in Equation 3.14, is used when solving the thrust allocation problem. Note that τ is the commanded thrust load given by the controller and not the actual thrust load produced, which is used in (3.14). The Moore-Penrose pseudoinverse, from Thor I. Fossen and Sagatun (1991), is used to solve the (3.14) with respect to u, yielding

$$u = B^{\dagger} K^{\dagger} \tau, \tag{5.2}$$

with

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ l_{x1} & -l_{y1} & l_{x2} & -l_{y2} & l_{x4} & -l_{y4} & l_{x6} & -l_{y6} \end{bmatrix} \in \mathbb{R}^{3x8}, \tag{5.3}$$

where B is the extended configuration matrix, and $K = diag(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8) \in \mathbb{R}^{8\times 8}$ is the extended thrust gain matrix. Note that the thrust allocation algorithm does not respect the rate saturation of the thrusters or consider forbidden sectors connected to thrust losses.

The thrusters input is saturated at 0.5,

$$u_i = sat(\sqrt{u_{xi}^2 + u_{yi}^2}) \tag{5.4}$$

and the thruster angles

$$\alpha_i = arctan(\frac{u_{yi}}{u_{xi}}) \tag{5.5}$$

are calculated from the relationship between thruster input in the x and y direction of each individual thruster.

5.4 Human machine interface

The HMI connects the operator to the DP system and lets it monitor, log and interact with the system. The system has both a GUI and a separate joystick control, letting

the operator control the vessel with a Play Station 4 (PS4) controller. This section puts emphasis on the GUI of the HMI, which was built from scratch for this thesis.

GUI node

The GUI node runs on the computer on land together with the Qualisys node. The GUI node is compatible with both HIL and model-scale tests. The GUI node subscribes to all topics in the system.

It is implemented using QT (QT 2022), which is easily integrated with ROS. Utilizing a Model-View-Controller architectural pattern, the application upholds good scalability and modifiability properties (Bass et al. 2013). Figure 5.3 illustrates the layout of the GUI. The window gives the operator an overview of the system in realtime. Furthermore, it provides information to the user, such as the vessel's actual, desired, and estimated position and heading in realtime, and thruster information. Currently, Plotjuggler (Plotjuggler 2022) is used to make detailed plots of data from topics in realtime, and Rosbag (ROS 2022) is used to save the data for later usage. Therefore, this functionality is not implemented in the GUI as well. The GUI gives the operator an overview of the environment the vessel operates in and shows the performance of the observer, guidance system, thrust allocator, and controller.

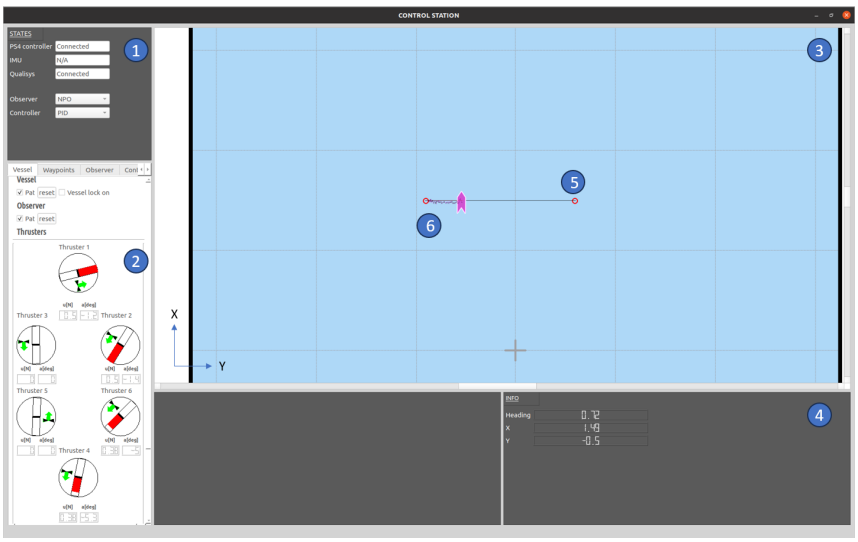


Figure 5.3: Layout of the GUI, implemented in the GUI node.

The GUI has the following functionality:

1. *Status indicators*, showing the status of the PS4 controller and MoCap system Qualisys.
2. *Thruster usage*, showing the angle and power produced by each thruster.
3. *Overview of vessels environment*, Vessel position and heading are shown in 2D. In addition, the 2D map shows the path of the vessel, observers estimated position,

reference trajectory, and the active setpoints. There is functionality added to zoom in and out, lock the camera on the vessel and move the camera freely around the environment.

4. *Vessel info*, showing the position and heading of the vessel.

5. *Realtime plots*, showing the desired position from the guidance system, an estimated position from the observer, and the actual position of the vessel. The functionality to hide and remove each individual plot is added.

6. *Setpoints* are marked, and trajectory between setpoints are displayed.

7. *Realtime tuning* is implemented for observers. It makes the tuning process simpler, and gains are easily set. This functionality is easily transferred to the tuning of controllers, which can't be tuned offline like the observers. This functionality is illustrated in Figure B.3.

More screenshots from the GUI implemented are found in Appendix B. This includes an interactive testing interface and more detailed pictures of plots, and a zoom-in on the thruster information.

Chapter 6

DP observer designs

In this chapter, different DPO designs are considered. Both the stability and limits for gains that make the system stable are analyzed for the different DPO. Afterward, guidelines for manual tuning are presented for each observer.

6.1 Nonlinear passive observer

The DP passive observer is based on the 3 DOF ODM (3.26). Represented in state space, it becomes

$$\begin{aligned}
\dot{\hat{\xi}} &= A_\omega \hat{\xi} + K_1 \tilde{y} \\
\dot{\hat{\eta}} &= R(\psi)\hat{\nu} + K_2 \tilde{y} \\
\dot{\hat{b}} &= -T_b^{-1} b + K_3 \tilde{y} \\
M\dot{\hat{\nu}} &= -D\hat{\nu} + R^T(\psi)\hat{b} + \tau + R^T(\psi) K_4 \tilde{y} \\
\hat{y} &= \hat{\eta} + C_\omega \xi,
\end{aligned} \tag{6.1}$$

with the observer gains

$$K_1 := \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \\ k_4 & 0 & 0 \\ 0 & k_5 & 0 \\ 0 & 0 & k_6 \end{bmatrix}, \quad K_2 := \begin{bmatrix} k_7 & 0 & 0 \\ 0 & k_8 & 0 \\ 0 & 0 & k_9 \end{bmatrix}, \tag{6.2}$$

$$K_3 := \begin{bmatrix} k_{10} & 0 & 0 \\ 0 & k_{11} & 0 \\ 0 & 0 & k_{12} \end{bmatrix}, \quad K_4 := \begin{bmatrix} k_{13} & 0 & 0 \\ 0 & k_{14} & 0 \\ 0 & 0 & k_{15} \end{bmatrix}. \tag{6.3}$$

The matrices are tuned according to T. I. Fossen (2021) rules in order to make the system Globally Exponentially Stable (GES), that is,

$$k_i = -2(\zeta_{ni} - \lambda_i)\frac{\omega_{ci}}{\omega_{oi}} = -2.2059 \tag{6.4}$$

$$k_{3+i} = 2(\zeta_{ni} - \lambda_i)\omega_{oi} = 1.8 \tag{6.5}$$

$$k_{6+i} = \omega_{ci} \tag{6.6}$$

$$k_{9+i} \gg \frac{k_{12+i}}{T_{b,i}}, \tag{6.7}$$

where $i = \{1, 2, 3\}$, $\lambda_i = 0.1$ is the relative damping of the wave spectrum, $\zeta_{ni} = 1 > \lambda_i$ is damping parameters, ω_{oi} is the peak frequency of the wave spectrum, $\omega_{ci} = 1.226\omega_{oi}$ is the filter cut-off frequency, and $T_{b,i}$ are the bias time constants. The bias time constant is typically set to 1000s for full-scale, which scales down to $T_{b,i} = \frac{1000}{\sqrt{90}} = 105.41s$ for CSAD.

6.2 LTV Kalman filter

The LTV-KF algorithm is the same as presented in Simon (2006). It is based on the continuous dynamics of Equation 3.26 written on the form

$$\hat{x} = A(t, x) + Bu + Ew \tag{6.8}$$

$$y = Cx + v \tag{6.9}$$

Værnø et al. (2019) also uses the exact same discretized system, which is as follows

$$x_k = f_{k-1}\left(x_{k-1}, t_{k-1}, u_{k-1}, w_{k-\mathbf{1}}\right) \tag{6.10}$$

$$= \Phi_{k-1}\left(x_{k-1}, t_{k-1}\right) x_{k-1} + \Delta_{k-1}\left(x_{k-1}, t_{k-1}\right) u_{k-1} \tag{6.11}$$

$$+ \Gamma_{k-1}\left(x_{k-1}, t_{k-1}\right) w_{k-1} \tag{6.12}$$

$$y_k = H_k x_k + v_k \tag{6.13}$$

$$w_k \sim (\mathbf{0}, Q_k) \tag{6.14}$$

$$v_k \sim (\mathbf{0}, R_k)\,, \tag{6.15}$$

Where

$$\Phi_k = \mathrm{e}^{A(t_k, x_k)h} \approx I + A\left(t_k, x_k\right) h + \frac{1}{2}\left(A\left(t_k, x_k\right) h\right)^2 \tag{6.16}$$

$$\Delta_k = \left(\int_0^h \mathrm{e}^{Ah} d\sigma\right) B \approx \left(Ih + \frac{1}{2}A\left(t_k, x_k\right) h^2 + \frac{1}{6}A\left(t_k, x_k\right)^2 h^3\right) B \tag{6.17}$$

$$\Gamma_k = \left(\int_0^h \mathrm{e}^{A(t_k, x_k)h} d\sigma\right) E \approx \left(Ih + \frac{1}{2}A\left(t_k, x_k\right) h^2 + \frac{1}{6}A\left(t_k, x_k\right)^2 h^3\right) E \tag{6.18}$$

$$H_k = C. \tag{6.19}$$

It is implemented as an iterative loop, and its stability properties are investigated with linear system theory. The LTV-KF has the following implementation.

Initialization:

$$\hat{x}_0^+ = E\left(x_0\right) \tag{6.20}$$

$$P_0^+ = E\left[\left(x_0 - \hat{x}_0^+\right)\left(x_0 - \hat{x}_0^+\right)^\top\right] \tag{6.21}$$

The Kalman filter has the following state equations:
The Kalman filter gain

$$K_k = P_k^- H_k^\top \left(H_k P_k^- H_k^\top + R_k\right)^{-1}. \tag{6.22}$$

The state corrector

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(y(k) - H_k\hat{x}_k^-\right). \tag{6.23}$$

Covariance corrector:

$$P_k^+ = \left(I - K_k H_k\right) P_k^- \left(I - K_k H_k\right)^\top + K_k R_k K_k^\top. \tag{6.24}$$

State predictor:

$$\hat{x}_k^- = \Phi_{k-\mathbf{1}}\left(t_{k-1}\right)\hat{x}_k^+ + \Delta_{k-1}\left(t_{k-1}\right)u_{k-1} \tag{6.25}$$

Covariance predictor:

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^\top + \Gamma_{k-1}Q_{k-1}\Gamma_{k-1}^\top. \tag{6.26}$$

The LTV-KF is a well known observer design, and the stability properties will not be investigated further in this thesis but are found in Simon (2006).

6.3 DPO design based on Lyapunov transformation

This chapter presents an observer design that estimates η, ν, b, and restrictions for injection gains to ensure the system is stable. The observer disregards the wave motions ξ and is based on the ODM in (3.26), setting $\xi = 0$. Let $\dot{\hat{x}} := col(\hat{\eta}, \hat{\nu}, \hat{b})$ be the observer state. Furthermore, the injection gains are added with the feedback from observer error $\tilde{y} = y - \hat{y}$. The proposed observer model is,

$$\begin{aligned}
\dot{\hat{\eta}} &= R(t)\hat{\nu} + L_1\tilde{y} \\
M\dot{\hat{\nu}} &= -D\hat{\nu} + R(t)^T\hat{b} + \tau + R(t)^T L_2\tilde{y} \\
\dot{\hat{b}} &= -T_b^{-1}\hat{b} + L_3\tilde{y} \\
\hat{y} &= \hat{\eta}.
\end{aligned} \tag{6.27}$$

Inserting $\tilde{y} = \tilde{\eta}$, the error dynamics becomes

$$\dot{\tilde{x}} = A(t)\tilde{x} \tag{6.28}$$

$$A(t) := \begin{bmatrix} -L_1 & R(t) & 0 \\ -M^{-1}R(t)^T L_2 & -M^{-1}D & M^{-1}R^T(t) \\ -L_3 & 0 & -T_b^{-1}\tilde{b} \end{bmatrix} \in \mathbb{R}^{9\times 9}, \quad \tilde{x} := x - \hat{x}. \tag{6.29}$$

R. Skjetne (2022) proves the origin $\tilde{x} = 0$ stable when the following conditions are satisfied:

1. The model matrices satisfy $M = M^T > 0$, $D + D^T > 0$, and $T_b > 0$.
2. The injection gain matrices satisfy $L_1 = L_1^T > 0$, $L_2 = L_2^T > 0$ and $L3 = L_3^T > 0$, where L_1, and L_3 commute.
3. The symmetric matrices $L_1L_2 + L_2L_1 - 2L_3 - T_b^{-1}$, $L_3^{-1}L_1 - L_2^{-1}$, and $2L_3^{-1}L_1 - I$ are positive definite.

If conditions 1-3 are satisfied, the stability of $\tilde{x} = 0$ can be shown to be UGES, using the Lyapunov direct method. This is done by considering a quadratic Lyapunov function,

$$\lambda_{min}(P)|\tilde{x}|^2 \leq V(\tilde{x}) := \tilde{x}P\tilde{x} \leq \lambda_{max}(P)|\tilde{x}|^2, \tag{6.30}$$

where

$$P = \begin{bmatrix} L_2 & 0 & -I \\ 0 & M & 0 \\ -I & 0 & L_3^{-1}L_1 \end{bmatrix}. \tag{6.31}$$

By using the Schur complement (Gallier 2019) as the complement of P, it is shown that $P > 0$ when $L_2 > 0$, $M > 0$, and $L_3^{-1}L_1 - L_1^{-1} > 0$, which is satisfied by the assumptions. Værnø et al. (2019) shows that the time derivative of the Lyapunov function along the trajectories of the error dynamics, $\dot{V} = 2\tilde{x}P\dot{\tilde{x}}$ is bounded by

$$\dot{V}(\tilde{x}) \leq -(L_1L_2 + L_2L_1 - 2L_3 - T_b^{-1})|\hat{\eta}|^2 - (D + D^T)|\hat{\nu}|^2 - (2L_3^{-1}L_1 - I)T_b^{-1}|\hat{b}|^2. \tag{6.32}$$

Given that condition 1 is relaxed to $D + D^T \geq 0$ and $T_b^{-1} = 0$, the system can be proven to be Uniformly Globally Stable (UGS) by the fact that $V(\tilde{x})$ is positive definite, radially bounded, and $\dot{V}(\tilde{x}) \leq -(L_1L_2 + L_2L_1 - 2L_3 - T_b^{-1})|\hat{\eta}|^2 \leq 0$. Furthermore, the equilibrium can be proven Uniformly Globally Asymtotically Stable (UGAS) (R. Skjetne 2021), using Matrosovs Theorem (Loría et al. 2005).

The injection gains should satisfy conditions 2 and 3. R. Skjetne (2022), proposes choosing L_1, L_2, and L_3 as positive diagonal matrices, with L_2 relatively large and L_3 relatively small to satisfy condition 3. This is satisfied by choosing $L_1 > 0$ diagonal, $L_2 = aL_1$, and

$L_3 = \frac{1}{b}L_1$, with $a, b \geq 1$. This yields the new conditions

$$L_1L_2 + L_2L_1 - 2L_3 - T_b^{-1} = 2(aL_1 - \frac{1}{b})L_1 - T_b^{-1} > 0 \implies (aL_1 - \frac{1}{b})L_1 > \frac{1}{2}T_b^{-1} \tag{6.33}$$

$$L_3^{-1}L_1 - L_2^{-1} = bI - \frac{1}{a}L_1^{-1} > 0 \implies abL_1 > I \tag{6.34}$$

$$2L_3^{-1}L_1 - I = 2bI - I > 0 \implies b > \frac{1}{2}, \tag{6.35}$$

which is easily satisfied. Let $L_1 := diag(l_1, l_2, l_3)$. The 1st condition states that each diagonal element l_i is bounded by $(al_1 - \frac{1}{b})l_1 > \frac{1}{2t_{b,(\cdot)}}$, the 2nd condition bounds each diagonal element l_i by $abl_i > 1$ and 3rd condition is satisfied through $b \geq 1$. L_2 and L_3 is then calculated from given L_1 that satisfies these three conditions.

6.4 DPO design based on small-gain principle

6.4.1 Small-gain principle

Before presenting the observer design, the small-gain principle is presented (Khalil 2015). It derive robust stability criteria for the observer and sets sufficient conditions for robust performance. Physically, it corresponds to the fact that every time the external input circulates in the closed loop, it is attenuated (Liu and Yao 2016). Stability characterization based on the small-gain principle complements deterministic design based on Lyapunov stability results in the helpful toolbox for nonlinear observer design (Teel 1996).

Consider the two subsystems,

$$\dot{x}_i = f_i(t, x_i, u_i) \quad f_i(t, 0, 0) = 0 \qquad i = \{1, 2\}, \quad \forall t. \tag{6.36}$$

Suppose there exist some $V_i(t, x_i)$ and $\alpha_{j,i}, \kappa_i \in \kappa_\infty$ for $j = 1, 2, 3, 4$ s.t. $\forall t$,

$$\alpha_{1,i}|x_i|^2 \leq V_i(t, x_i) \leq \alpha_{2,i}|x_i|^2 \tag{6.37}$$

$$\dot{V}_i = V_i^t(t, x_i) + V_i^{x_i}(t, x_i)f_i(t, x_i, u_i) \leq -\alpha_{3,i}|x_i|^2 + \kappa_i|u_i|^2. \tag{6.38}$$

Both subsystems are Input-to-State Stability (ISS) w.r.t input u_i (Sontag 2008). Suppose $u_1 = x_2$ and s.t. we have the cascade,

$$\dot{x}_1 = f_1(t, x_1, x_2), \quad \dot{x}_2 = f_2(t, x_1, u_2), \tag{6.39}$$

that is ISS w.r.t u_2 as input (R. Skjetne 2022). Suppose that we make the feedback interconnection $u_2 = K(t)x_1$, that is

$$\dot{x}_1 = f_1(t, x_1, x_2) \qquad \dot{x}_2 = f_1(t, x_1, K(t)x_1), \tag{6.40}$$

where K(t) is piecewise continuous and $sup_t\|K(t)\| \leq \rho \leq \infty$. The origin $(x_1, x_2) = 0$ of (6.40) is UGES when the small-gain condition,

$$\rho^2 < \frac{\alpha_{3,2}(\alpha_{3,1} - \epsilon_2)}{\kappa_1\kappa_2(1 + \epsilon_1)} < \frac{\alpha_{3,1}\alpha_{3,2}}{\kappa_1\kappa_2} \tag{6.41}$$

where $0 < \epsilon_1, \epsilon_2 \ll 1$.

ISS of (6.39) can be shown by letting $V := V_1 + aV_2$ with $a = (1+\epsilon_1)\frac{\kappa_1}{\alpha_{3,2}}$ for $\epsilon_1 > 0$ resulting in

$$\begin{aligned}\dot{V} &\leq -\alpha_{3,1}|x_1|^2 + \kappa_1|x_2|^2 - a\alpha_{3,2}|x_2|^2 + a\kappa_2|u_2|^2 \\ &= -\alpha_{3,1}|x_1|^2 - \varepsilon_1\kappa_1|x_2|^2 + a\kappa_2|u_2|^2,\end{aligned} \tag{6.42}$$

where V is an ISS-Lyapunov function. Further, (6.40) is shown to be UGES when considering V as a Lyapunov function and $u_2 = K(t)x_1$, resulting in

$$\begin{aligned}\dot{V} &\leq -\alpha_{3,1}|x_1|^2 - \varepsilon_1\kappa_1|x_2|^2 + a\kappa_2\|K(t)\|^2|x_1|^2 \\ &\leq -\alpha_{3,1}|x_1|^2 + a\kappa_2\rho^2|x_1|^2 - \varepsilon_1\kappa_1|x_2|^2 \\ &= -\alpha_{3,1}|x_1|^2 + (1+\varepsilon_1)\frac{\kappa_1\kappa_2}{\alpha_{3,2}}\rho^2|x_1|^2 - \varepsilon_1\kappa_1|x_2|^2 \\ &\leq -\varepsilon_2|x_1|^2 - \varepsilon_1\kappa_1|x_2|^2\end{aligned} \tag{6.43}$$

where the small-gain condition (6.41) is applied. Lyapunov's direct method proves the origin $(x_1, x_2) = 0$ UGES (R. Skjetne 2023).

6.4.2 DPO design

This section presents a DPO design copying the ODM in (3.26), and adding injection gains with feedback from the observer output error $\tilde{y} = y - \hat{y}$, that is,

$$\begin{aligned}\dot{\hat{\eta}} &= R(t)\hat{\nu} + L_1\tilde{y} \\ M\dot{\hat{\nu}} &= -D\hat{\nu} + \hat{b} + \tau + R(t)^T L_2\tilde{y} \\ \dot{\hat{b}} &= -T_b^{-1}\hat{b} + L_3\tilde{y} \qquad (6.44) \\ \dot{\hat{\xi}} &= A_w\hat{\xi} + L_4\tilde{y} \qquad (6.45) \\ \hat{y} &= \hat{\eta} + C_w\hat{\xi},\end{aligned}$$

where $L_1, L_2, L_3 \in \mathbb{R}^{3\times 3}$, and $L_4 \in \mathbb{R}^{6\times 3}$ are injection gains.

Stability condition

The error dynamics of the system,

$$\begin{aligned}\dot{\tilde{\eta}} &= R(t)\tilde{\nu} - L_1\tilde{y} \\ M\dot{\tilde{\nu}} &= -D\tilde{\nu} + \tilde{b} - L_2R(t)^T\tilde{y} \\ \dot{\tilde{b}} &= -T_b^{-1}\tilde{b} - L_3R(t)^T\tilde{y} \qquad (6.46) \\ \dot{\tilde{\xi}} &= A_w\tilde{\xi} - L_4\tilde{y}. \qquad (6.47)\end{aligned}$$

Let $\tilde{z} := col(\tilde{\eta}, \tilde{\xi}) \in \mathbb{R}^9$, $\tilde{x} := col(M\tilde{\nu}, \tilde{b}) \in \mathbb{R}^6$, $\Upsilon(t) := col(R(t), I) \in \mathbb{R}^{9\times 3}$, $\Lambda = row(M^{-1}, 0_{3\times 3}) \in \mathbb{R}^{3\times 6}$, $L_{14} := col(L_1, L_4) \in \mathbb{R}^{9x3}$ and $L_{23} := col(L_2, L_3) \in \mathbb{R}^{6x3}$. Associating Σ_1 to $\tilde{z}_1$ and Σ_2 to $\tilde{x}$ with $u_1 = \tilde{x}$ and $u_2 = -L_{23}R(t)^T\tilde{y}$ gives the interconnected subsystems,

$$\Sigma_1 := \dot{\tilde{z}} = A_1\tilde{z} + \Upsilon(t)\Lambda u_1 \tag{6.48}$$

$$\Sigma_2 := \dot{\tilde{x}} = A_2(t)\tilde{x} + u_2 \tag{6.49}$$

where

$$A_1 := \begin{bmatrix} -L_1 & L_1C_w \\ -L_4 & A_w - L_4C_w \end{bmatrix}, \quad A_2(t) := \begin{bmatrix} -M^{-1}D & I \\ 0 & -T_b^{-1} \end{bmatrix}. \tag{6.50}$$

It is worth noting that $A_1 = A_0 - L_{14}C_0$, where $A_0 := diag(0_{3\times 3}, A_w) \in \mathbb{R}^{9\times 9}$ and $C_0 := row(I, C_w) \in \mathbb{R}^{3\times 9}$. The pair A_0 and C_0 is observable, hence, L_{14} can be designed to render A_1 Hurwitz. Further, let $P_1 = P_1^T$ s.t.

$$P_1A_1 + A_1^TP_1 = -q_1I \tag{6.51}$$

and $V_1(\tilde{z}) := \tilde{z}^TP_1\tilde{z}$ to show that Σ_1 is ISS, that is

$$\lambda_{min}(P_1) \leq V_1(\tilde{z}) \leq \lambda_{max}(P_1), \tag{6.52}$$

$$\begin{aligned} \dot{V}_1(\tilde{z}) &\leq -q_1|\tilde{z}|^2 + 2\lambda_{max}(P_1)|\tilde{z}|\|\Upsilon(t)\|\|\Lambda\||u_1| \\ &\leq -(q_1 - \frac{\lambda_{max}(P_1)^2\|\Lambda\|^2}{\kappa_1})|\tilde{z}|^2 + \kappa_1|u_1|^2 \\ &\leq -c_1|\tilde{z}|^2 + \kappa_1|u_1|^2, \qquad c_1 := q_1 - \frac{\lambda_{max}(P_1)^2\|\Lambda\|^2}{\kappa_1}, \end{aligned} \tag{6.53}$$

where $\kappa_1 > 0$ is chosen s.t. $c_1 > 0$.

For Σ_2, A_2 is designed Hurwitz with Appropriate pole placement. Correspondingly, let $P_2 = P_2^T$ s.t.

$$P_2A_2 + A_2^TP_2 = -q_2I, \tag{6.54}$$

and $V_2(\tilde{x}) := \tilde{x}^TP_1\tilde{x}$, yielding

$$\lambda_{min}(P_2) \leq V_2(\tilde{x}) \leq \lambda_{max}(P_2), \tag{6.55}$$

$$\begin{aligned} \dot{V}_2(\tilde{x}) &\leq -q_2|\tilde{x}|^2 + 2\lambda_{max}(P_2)|\tilde{x}||u_2| \\ &\leq -(q_2 - \frac{\lambda_{max}(P_2)^2}{\kappa_2})|\tilde{x}|^2 + \kappa_2|u_2|^2 \\ &\leq -c_2|\tilde{x}|^2 + \kappa_2|u_2|^2, \qquad c_2 := q_2 - \frac{\lambda_{max}(P_2)^2}{\kappa_2}, \end{aligned} \tag{6.56}$$

where $\kappa_2 > 0$ is chosen s.t. $c_2 > 0$. Hence, Σ_2 is ISS (Sontag 2008).

The interconnected system with $u_2 = -L_{23}R(t)^T\tilde{y}$, with $\tilde{y} = C_0\tilde{z}$, the $sup_t\|-R(t)^TC_0\| = 2$. By letting $\|L_{23}\| = \|L_{23}\|_2 = \rho$, and $4\|L_{23}\|_2^2 < \frac{c_1c_2}{\kappa_1\kappa_2}$, the interconnected subsystems with $u_1 = \tilde{x}$ and $u_2 = -L_{23}R(t)^TC_0\tilde{z}$, is UGES by the small gain theorem.

Choosing gains

Before choosing the gains κ_1 and κ_2 have to be set in order to render $c_1 > 0$ and $c_2 > 0$. Using design recommendations from R. Skjetne (2023), setting $\kappa_1 = 2\frac{\lambda_{max}(P_1)^2\|\Lambda\|^2}{q_1}$ and $\kappa_2 = 2\frac{\lambda_{max}(P_2)^2}{q_2}$ gives $c_1 = \frac{q_1}{2}$ and $c_2 = \frac{q_2}{2}$. In order to satisfy the small gain condition

$$\|L_{23}\| < \frac{1}{8\|\Lambda\|}\frac{q_1}{\lambda_{max}(P_1)}\frac{q_2}{\lambda_{max}(P_2)} \tag{6.57}$$

has to be satisfied. It sets an upper bound on the gain matrix L_{23} which depends on L_2 and L_3. It can be further simplified by using properties of the matrix 2-norm (Petersen and Pedersen 2012), giving $\|\Lambda\|_2 = \sqrt{\lambda_{max}(\Lambda^T\Lambda)} = \sqrt{\lambda_{max}((M^{-1})^2)} = \|M^{-1}\|_2 = \frac{1}{\lambda_{min}(M)}$ due to $M = M^T > 0$ and $\|L_{23}\|_2 = \sqrt{\lambda_{max}(L_{23}^T L_{23})} = \sqrt{\lambda_{max}(L_2L_2 + L_3L_3)}$, that gives

$$\|L_{23}\| < \frac{\lambda_{min}(M)}{8}\frac{q_1}{\lambda_{max}(P_1)}\frac{q_2}{\lambda_{max}(P_2)}. \tag{6.58}$$

Now let $L_2 = diag(l_{2,1}, l_{2,2}, l_{2,3} > 0$ and $L_3 = diag(l_{3,1}, l_{3,2}, l_{3,3} > 0$ and $q_1 = q_2 = 1$, the the sum of each diagonal pair is bounded by

$$l_{2,i}^2 + l_{3,i}^2 < \left(\frac{\lambda_{min}(M)}{8\lambda_{max}(P_1)\lambda_{max}(P_2)}\right)^2, \qquad \forall i \in 1, 2, 3. \tag{6.59}$$

The wave motion gains are chosen using recommendations of T. I. Fossen (2021). Let $L_4 = row(L_{4,1}, L_{4,2}) \in \mathbb{R}^{3\times 6}$ where

$$L_{4,1} = -2(\zeta_n - \lambda)\frac{\omega_c}{\omega_0}I = -2.2059 \cdot I \tag{6.60}$$

$$L_{4,2} = 2(\zeta_n - \lambda)\omega_0 I = 1.8 \cdot I \tag{6.61}$$

with the filter cutoff frequency $\omega_c = 1.2255\omega_0$, and $\zeta_{ni} > \lambda$ determining the notch is set to $\lambda = 0.1$, and $\zeta = 1.0$.

Chapter 7

Derivative Free Optimisation

To investigate the performance of the model and compare the different observers, optimization is used. Since the Hessian or gradient is not available, DFO is the most viable option. This chapter covers how a PSO is used to tune the observer gains offline. Firstly, the optimization problem is formulated, and then the PI is proposed.

7.1 Optimisation problem

The goal is to calibrate the observer gains, x, to minimize the objective function, $f(x)$, subjected to some constraints, $g_i(x)$, given on standard form as

$$\min\ f(x) \tag{7.1}$$
$$\text{subject to } g_i(x). \tag{7.2}$$

The constraint $g_i(x)$ makes sure the system is stable and is described in chapter 6. For the Kalman filter the Q-matrix (process-noise) is used as tuning parameters and the Q-matrix is restricted to be diagonal. Furthermore, the deterministic observer's gains, $L1, L2, L3$, and $L4$, are also restricted to being diagonal with some simplifications. The

- $L1 = diag(x_1, x_1, x_2)$, $L2 = diag(x_3, x_3, x_4)$, $L3 = diag(x_5, x_5, x_6)$.
- $Q = diag(x_1, x_2, x_3, x_4, x_5, x_6)$.

L4 is set manually according to T. I. Fossen (2021) rules of tuning. Note that these simplifications are not optimal for the performance of the observer, but in practice, it is necessary in order to find the minimum of the DFO in a reasonable time (Værnø et al. 2019). It is a trade-off between the performance of the DFO and the computing resources used. Including all non-diagonal terms would construct a problem too large to solve with the computing resources available.

7.2 Performance indicator functions

The PI relevant to this thesis is a scalar value defining the performance of the observer. The PIs evaluate how well the observers estimate the position of the vessel. A well-tuned observer is used when collecting training data, both in HIL tests and model-scale tests. The following data is collected to be used for DFO tuning: η position and heading of the vessel, ν linear and angular velocity, τ actual thrust load, and Tp peak period when waves are used. In the high-fidelity simulator, all this data is available. The data collected from the model-scale tests require some post-processing,

List of different PI functions used by M. E. N. Sørensen and Morten Breivik (2015) and Eriksen and M. Breivik (2017):

1. Integral of Absolute Error (IAE): It is a standard performance measure due to its simplicity. It describes the overall accuracy, and intuitively, the perfect IAE is zero.
2. Integral of Square Error (ISE): The ISE describes the overall accuracy, and the perfect ISE is intuitively zero. However, compared to the IAE, it weighs significant errors.
3. Integral of Absolute Error multiplied by Work(IAEW): It measures error combined with work, where error velocity and control input are weighted equally. The integral of error can be reduced to zero, and the work reduced to the minimal work that can perform the maneuver.
4. Integral of Absolute Error and Control (IAEC): The measure of both error and control input. The integral of error can be reduced to zero, and the minimum control input will be larger than zero. It measures the wear and tear to some extent.
5. Integral of Absolute Error multiplied by Time (ITAE): The ITAE weights errors that happen later in the simulation.
6. Integral of Absolute Derivative Control (IADC): The IADC measures the change in control input and does not measure the accuracy, which makes it a PI solely for wear and tear. However, it is highly susceptible to noise being derivative and should be paired up with an accuracy measure.

According to Løvås (2019) and Værnø et al. (2019), IAE, IAEW, and IAEC, are suitable PI candidates for controller tuning. However, the observer is not concerned with the work of the system. Therefore, IAE is chosen for the thesis due to its simplicity and lack of required tuning.

The cost function, J(x), used for the DFO in the thesis is

$$J(x, \eta_{test}) = J_\eta(x, \eta_{test}) + cJ_\nu(x, \eta_{test}) \tag{7.3}$$

where the observer is initialized with the gains set by x, then IAE measures the cumulative position and velocity error

$$J_\eta(x, \eta_{test}) = \Sigma_{k=0}^{n}(|\eta_{N,k} - \hat{\eta}_{N,k}| + |\eta_{E,k} - \hat{\eta}_{E,k}| + \frac{180}{\pi}|\psi_k - \hat{\psi}_k|) \tag{7.4a}$$

$$J_\nu(x, \nu_{test}) = \Sigma_{k=0}^{n}(|u_k - \hat{u}_k| + |v_k - \hat{v}_k| + \frac{180}{\pi}|r_k - \hat{r}_k|). \tag{7.4b}$$

Here, η_{test} is the data used to evaluate the cost function, $k = n$ is the final step of the interval, and $c > 0$ is a scaling constant. It is found iteratively and set s.t. the size of the IAE contribution for position and velocity are similar. The weighting of (7.3) is chosen in order to be close to DP operating limits, which are typically 3m and 3 degrees (Veritas 2011). The cost of 1 m error in position is equivalent to 1 deg error in the heading. Likewise, 1 m/s error in linear velocity is equivalent to 1 deg/s error in angular velocity. η_{test} contains the following sets: $Y = \{y_0, y_1, y_2, \ldots, y_n\}$ the measurement data containing the position and heading of the vessel, $T = \{\tau_0, \tau_1, \tau_2, \ldots, \tau_n\}$ is the actual thruster loads produced, $H = \{\nu_0, \nu_1, \nu_2, \ldots, y_n\}$ and $V = \{\nu_0, \nu_1, \nu_2, \ldots, \nu_n\}$ which is the actual position and velocity of the vessel used to evaluate the performance of the observer.

For the high-fidelity simulator, all the datasets are directly available and extracted. Likewise, all data from the model-scale tests are directly extracted, except H, which is post-processed using a low pass Finite Impulse Response (FIR) filter to find actual velocities.

Chapter 8

Desktop study

This chapter presents the results from the desktop study comparing the performance of the DPOs. The results from four different test cases are presented. Model-scale tests with CSAD in MC-Lab were performed with the LTV-KF to collect data for the test cases. For each scenario, the cost function over time is presented. Finally, the results are consolidated in Table 8.1, Table 8.2, and Table 8.3. The cost functions of the DPOs are normalized, giving the worst-performing observer a score of 100. The nonlinear passive observer from T. I. Fossen (2021) was only used to set gains for the wave filter of the small-gain observer design and was not tuned with the autotuner due to the similarities between the structure of the two observers.

8.1 Test 1: Combined surge-sway-yaw motion in calm water

The combined motion test described in section 3.2 was performed in order to collect data for the desktop study. The data was collected in MC-Lab using the manually tuned LTV-KF. The first test was performed in calm water with no waves. Firstly, the vessel stabilizes at $\eta = [1.5\ 0.6\ 30°]$. Afterward, it gets a position change 1m in the negative Y direction and a heading change of 60 degrees counterclockwise to $\eta = [1.5\ \text{-}0.4\ -30°]$. Then the reference setpoint is changed 1m in the positive Y direction and 60 degrees clockwise in heading to return to its original position.

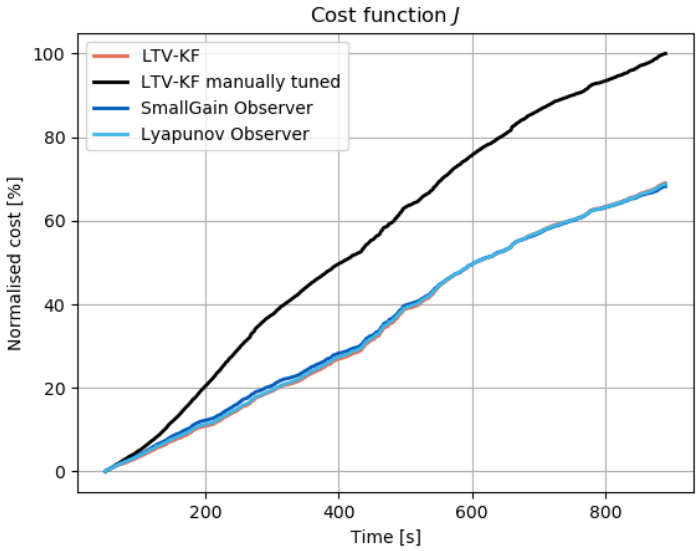


Figure 8.1: Comparison of DPO performance during the combined surge-sway-yaw motion in calm water.

Figure 8.1 illustrates the accuracy of each autotuned observer through their cost function. All the autotuned observers have a relatively equal performance when no waves are present, and they have a lower cost than the manually tuned LTV-KF. There is no significant difference between the performance of these observers when operating at low speed in no waves.

8.2 Test 2: Combined surge-sway-yaw motion in rough sea

The second test was performed in rough seas with a full-scale significant wave height of $H_S = 2.7m$ and a peak wave period of $T_P = 6.64s$. Figure 8.2 illustrates the performance of the autotuned observers relative to each other and the manually tuned LTV-KF. The Lyapunov based designs' performance falls off relative to the observer designs when adding waves, but it still outperforms the manually tuned LTV-KF. The autotuned LTV-KF performs quite well compared to the other designs. The small-gain based design seems to be able to filter out some of the WF motions but not all of them. The manually tuned LTV-KF is performing well compared to the autotuned LTV-KF.

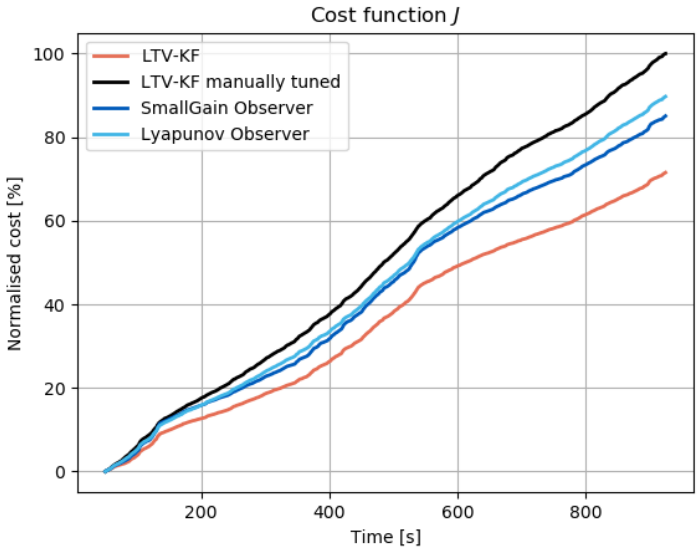


Figure 8.2: Comparison of DP performance during the combined surge-sway-yaw motion in rough seas.

8.3 Test 3: Combined surge-sway-yaw motion in very rough sea

The third test was run in rough seas with a full-scale significant wave height of $H_S = 4.5m$ and a peak wave period of $T_P = 10.4s$. Figure 8.3 illustrates the performance of the autotuned observers relative to each other and the manually tuned LTV-KF. In very rough seas, the manually tuned LTV-KF accuracy is significantly worse relative to the other tests. The Lyapunov based design performance has a bigger problem in very rough seas than in rough seas. This is expected as the wave motions play a bigger part in the very rough seastate. The LTV-KF is still the most accurate.

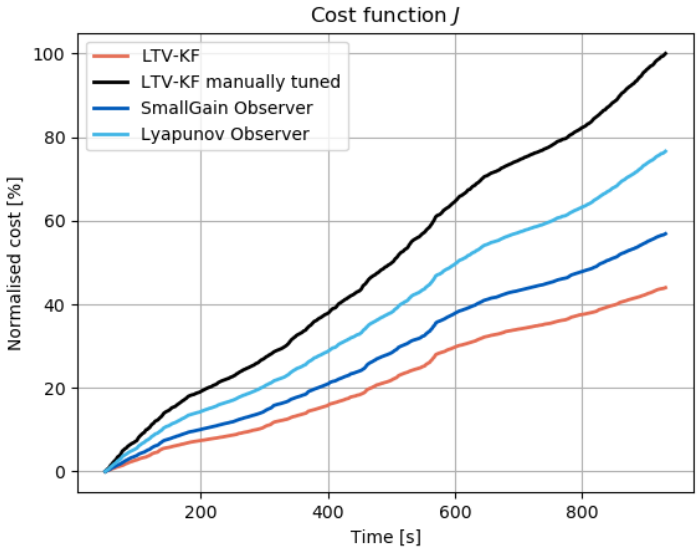
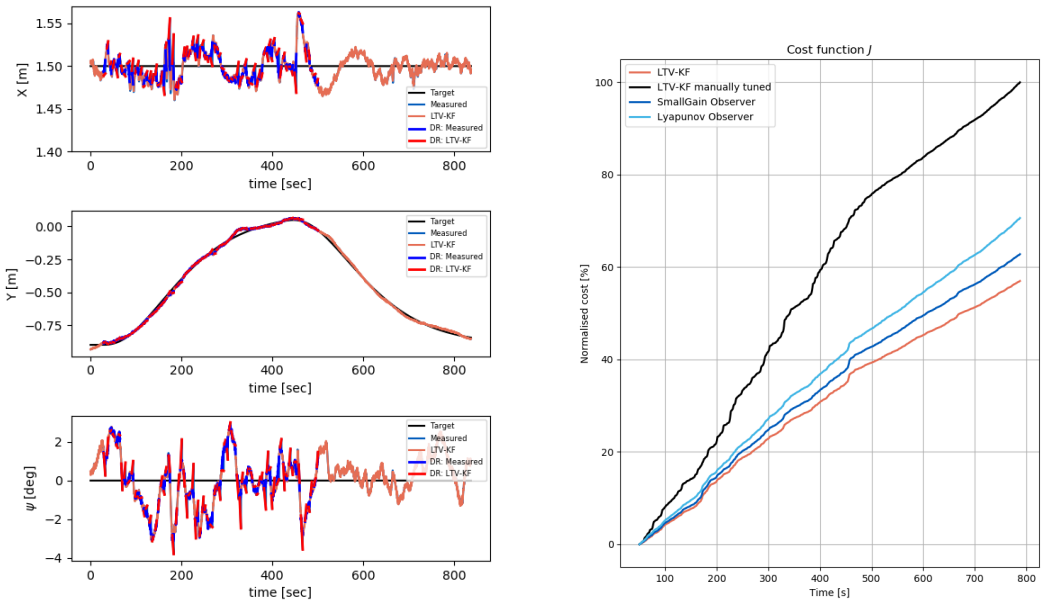


Figure 8.3: Comparison of DP performance during the combined surge-sway-yaw motion in very rough seas.

8.4 Test 4: Combined surge-sway-yaw motion in very rough sea with position loss

The fourth test is similar to the third test but with added position and heading loss. The duration of heading and position signal dropouts in MC-Lab are commonly between 1-5 seconds in model-scale time. The system loses position and heading with intervals of 1-5 seconds followed by 1-5 seconds with the signal regained. This continues until the vessel reaches the first setpoint. The signal is not lost when traveling from setpoint 1 to 2. The test checks if the observers are able to handle shorter frequent signal dropouts. In full-scale time the dropout duration is in the range of 10 - 50 seconds. Figure 8.4 illustrates the performance of the DP system and observers during the deadreckoning test. It can be seen that the DP performance is impacted when the vessel loses position and heading signal frequently. This is likely due to the observer struggling to estimate an accurate position and velocity. Figure 8.4b shows that the cost over time grows more rapidly for the observers when the signal is lost. However, the DP system still manages to follow the track, and the observers converge quickly close to the actual position of the vessel when the position and heading signal is regained. The biggest surprise is how the Lyapunov design improves its performance relative to the other observers when deadreckoning is introduced. It's important to note that this improvement is only relative to the other observers and not compared to its performance in test 3.



(a) Trajectory of estimated and measured position of CSAD during deadreckoning test with manually tuned LTV-KF.

(b) Comparison of DP performance during the combined surge-sway-yaw motion in very rough seas with deadreckoning.

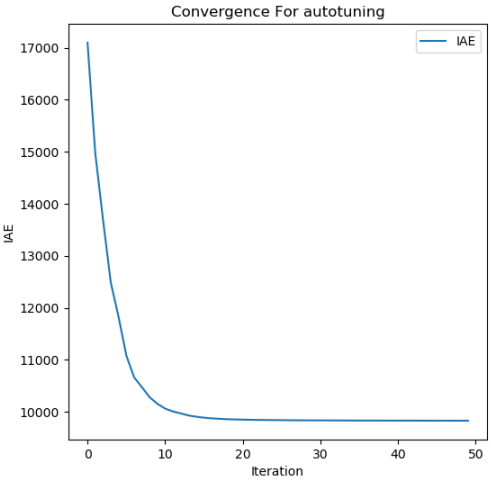
Figure 8.4: DP performance during the combined surge-sway-yaw motion in very rough seas with frequent position and heading signal dropout.

8.5 Autotuning

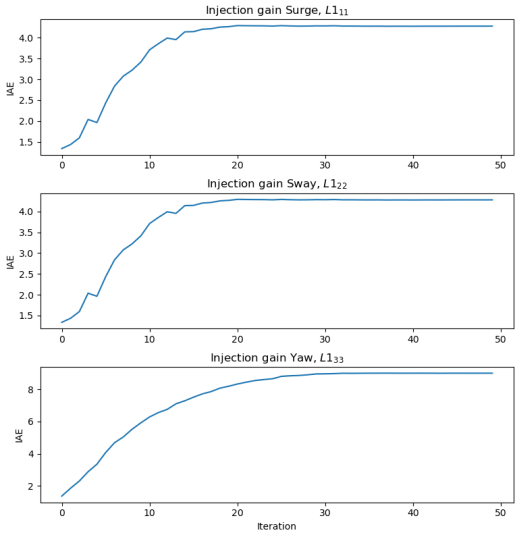
The PSO algorithm was used to autotune all observers. The particles were initiated with values based on the gains found when tuning manually. Noise was added to the initial particles when initialising the PSO to spread the particles throughout the search space. The autotuning result is separated into deterministic and stochastic parts since these two categories have different structures. In this section, the autotuning performed for the very rough sea state is presented.

Deterministic observers

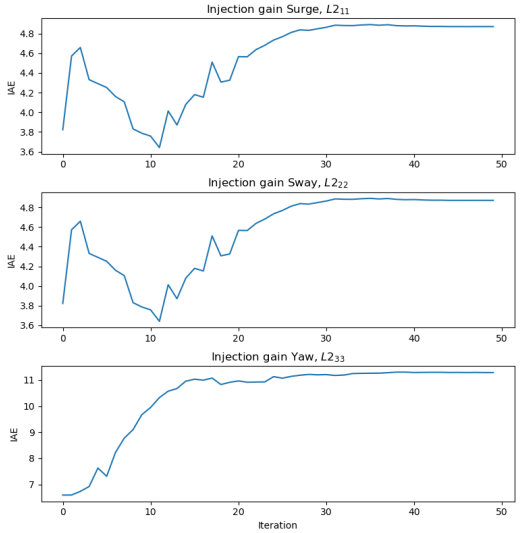
The injection gains L1, L2, and L3 are tuned using PSO, while L4 is set manually. The development of the gains and the convergence of the cost function is illustrated in Figure 8.5.



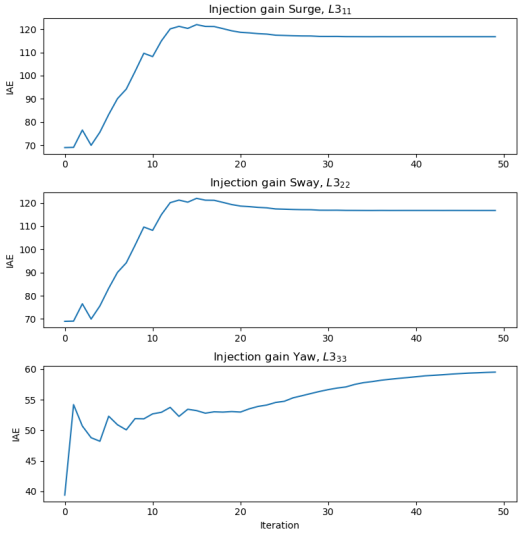
(a) Convergence of objective function, IAE.



(b) Development of L1.



(c) Development of L2.



(d) Development of L3.

Figure 8.5: (a) shows the convergence of IAE, (b)-(d) development of the best injection gains as a function of iteration.

The performance of the small-gain observer design increases significantly when tuned with the PSO algorithm. Figure 8.6 illustrates a comparison of the cost function of the observer design before and after it is tuned with the PSO algorithm. Originally the guidelines presented at chapter 6 were used to tune the observer manually.

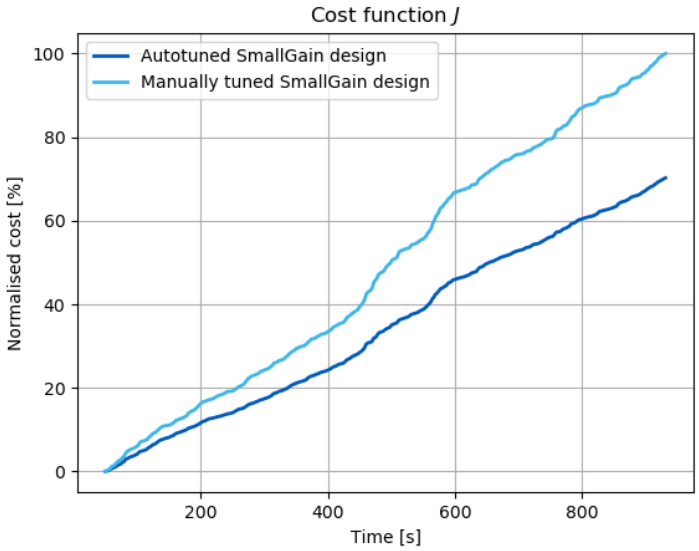


Figure 8.6: Comparison of the small-gain observer design before and after autotuning using the PSO algorithm.

When conducting the autotuning, the PSO algorithm was free to choose gains inside and outside of the stability region of the small-gain design. Afterward, the behavior of the autotuner was compared with the stability limits presented in (6.59). Figure 8.7 illustrates how the stable limits of the gains and the best injection gains changed each iteration when the observer was tuned for the very rough sea state. The autotuner is not bounded to be contained in the stable space but converges into it in this specific case. This relationship was not further investigated but is a promising observation.

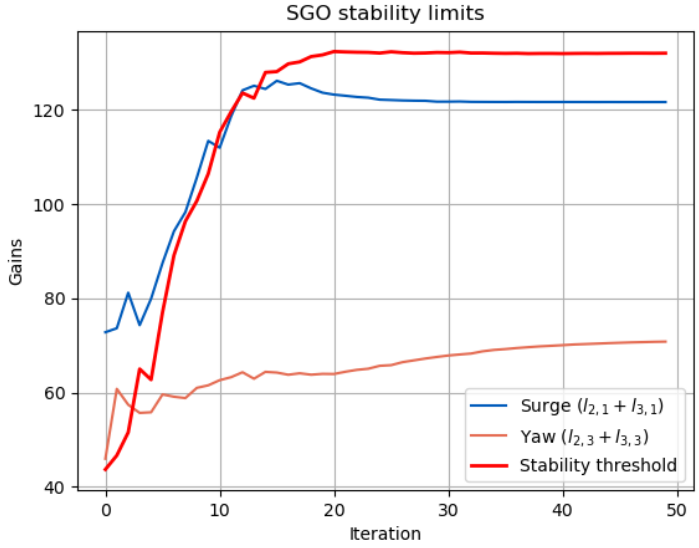
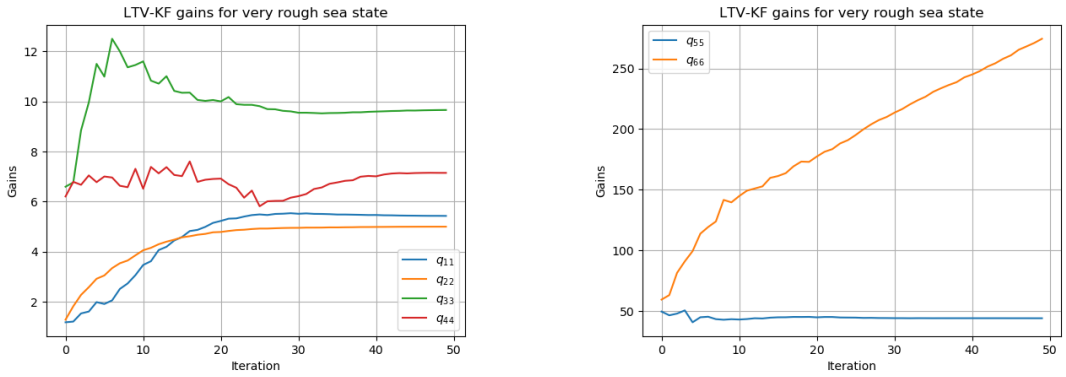


Figure 8.7: The development of the injection gains L2 and L3 with respect to the stability limits set by the small-gain observer design.

Stochastic observers

The measurement covariance matrix Q is autotuned using PSO. The development of the gains and the convergence of the cost function is illustrated in Figure 8.8. The Q matrix is in this case simplified to be diagonal.



(a) Development of $q_{11} - q_{44}$.

(b) Development of q_{55} and q_{66}.

Figure 8.8: Development of the best injection gains by PSO as a function of iteration.

Figure 8.9 illustrates the performance of the manually tuned LTV-KF and the autotuned LTV-KF. The LTV-KF was more challenging to manually tune relative to the small-gain observer design. The autotuned LTV-KF improves more compared to the manually tuned LTV-KF compared to the deterministic observers. This support the hypothesis that the LTV-KF is more challenging to tune manually.

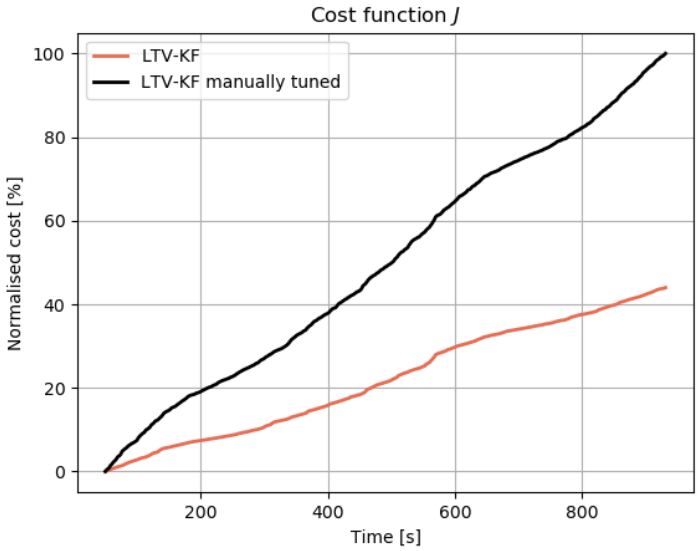


Figure 8.9: Comparison of the LTV-KF before and after autotuning using the PSO algorithm.

8.6 Summary of results

To compare the performance of each observer during the test scenarios, the quantitative position and velocity cost functions in (7.4) were used. The cost values are normalized such that the worst performing observer has a score of 100. The positional accuracy is presented in Table 8.1 and velocity accuracy in Table 8.2. The performance of the Lyapunov design relative to the small-gain design was decreasing when the vessel was subjected to larger waves. This was true both for accuracy of position and velocity.

Table 8.1: Position term of the cost function with normalized values such that the worst performing observer has a score of 100.

	LTV-KF baseline	LO	SG	LTV-KF
Test 1	46.03	22.41	20.93	23.39
Test 2	33.34	39.57	35.86	27.41
Test 3	31.29	32.67	23.57	18.28
Test 4	46.90	31.16	27.16	25.53

Table 8.2: Velocity term of the cost function with normalized values such that the worst performing observer has a score of 100.

	LTV-KF baseline	LO	SG	LTV-KF
Test 1	53.97	46.37	47.28	45.64
Test 2	66.66	50.14	49.19	44.11
Test 3	68.71	43.93	33.31	25.70
Test 4	53.10	39.39	35.23	32.47

During the MC-Lab model-scale tests, the autotuned observers outperformed the manually tuned LTV-KF. Furthermore, the autotuned LTV-KF had the overall best performance. All observer's performances were similar when waves were not introduced. Table 8.3 illustrates the total performance of all observers, including both position and velocity. The manually tuned LTV-KF is better relative to the other observers in the rough sea state compared to calm water. This is likely due to the baseline LTV-KF being tuned more accurately in that sea state. More time was invested in tuning the LTV-KF in waves than in calm waters. The goal of the manual tuning was to have a good enough observer to collect data, not to perfect the performance of the manually tuned LTV-KF in all sea states. Its also worth mentioning that the manually tuned LTV-KF was returned before performing test 4, which is the reason why it performs better relative to the small-gain design and LTV-KF in this test compared to test 3, which is without signal dropouts.

Table 8.3: The total cost function with normalized values such that the worst performing observer has a score of 100.

	LTV-KF baseline	LO	SG	LTV-KF
Test 1	100	68.79	68.21	69.03
Test 2	100	89.71	85.05	71.51
Test 3	100	76.60	56.88	43.98
Test 4	100	70.55	62.39	57.00

To summarize the comparison, all different PIs mentioned in section 4.1 are listed together in Table 8.4. The observers are able to handle short-duration position and heading signal loss, but the estimation error grows rapidly when the duration increases. The deadreckoning capability of the observers would be increased if other internal states were measured. The Lyapunov obsever design actually performed better relative to the other observers when performing deadreckoning, which was surprising. The Lyapunov was the easiest observer to implement and tune, but it has a decreased performance relative to the other observers when waves are introduced. The small-gain design with a wave filter similar to the nonlinear passive observer works well both in waves and calm waters and is easier to implement than the LTV-KF. The best performing observer all over is the LTV-KF. When introducing waves, it performs significantly better than the other observers both in position and velocity accuracy when tuned with the PSO algorithm.

Table 8.4: Comparison of all observers using selected PIs.

	LO	SG	LTV-KF
Accuracy	Good	Good/Great	Great
Manual tuning	Medium	Medium	Hard
Wave filter	No	Yes	Yes
Complexity of Implementation	Low/Medium	Medium	Medium/High
Deadreckoning capability	Medium	Medium	Medium

Chapter 9

Model-scale results

This chapter presents the results from the model-scale experiment conducted at MC-Lab. The tests performed are described in detail in section 3.2. The LTV-KF filter is the chosen observer for the final tests on CSAD and is autotuned using the PSO algorithm. The tests are performed in all sea states presented in section 3.3. In this chapter, experiments are performed in a very rough sea state. The results from other sea states are found in Appendix A. Lastly, the result from model-scale experiments with the small-gain observer on the vessels CSE and CSJ is presented.

9.1 4-corner test in very rough sea

The 4-corner test is described in section 3.2 and was performed in very rough seas with CSAD. Between each setpoint, the vessel was cut off from measurements of position and heading in 10-20 second intervals. All stages of the test will be investigated in the following sections. Figure 9.1 illustrates the measured position and estimated position of the 4-corner performed in very rough seas.

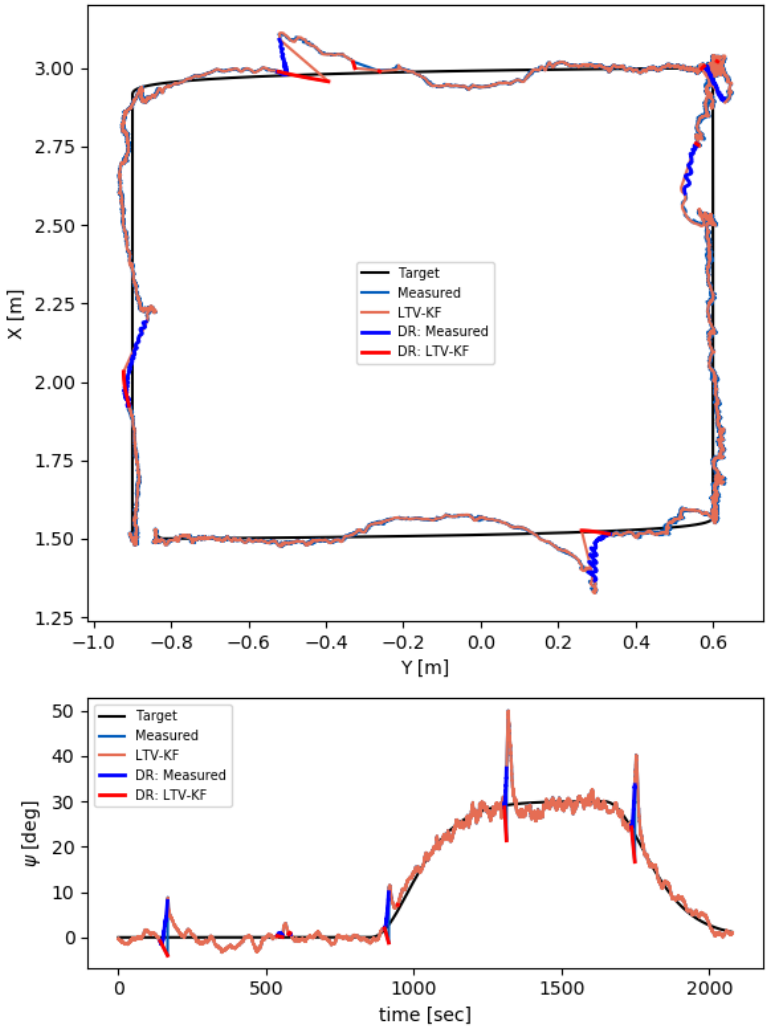
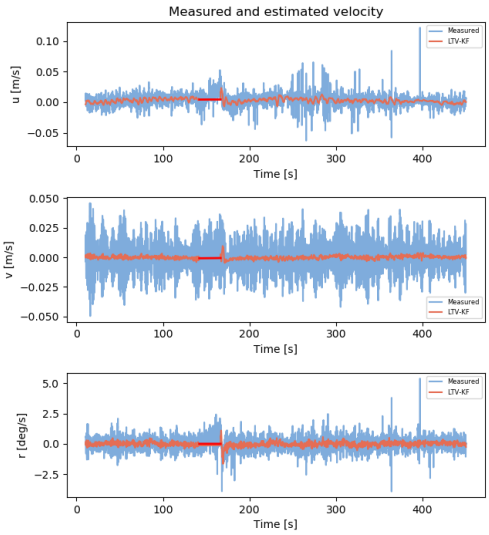


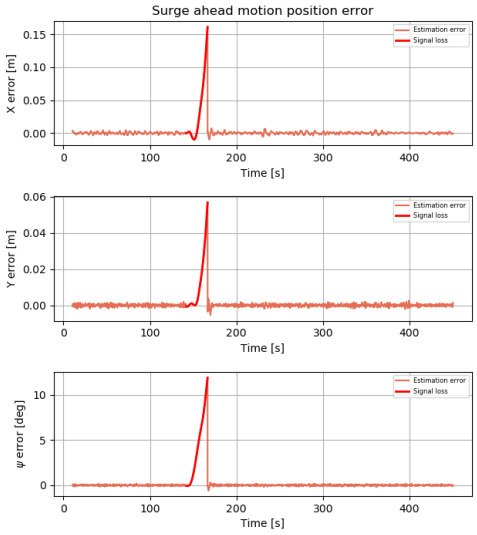
Figure 9.1: Trajectory of the 4-corner tests in very rough seas with deadreckoning intervals of 10-20 seconds.

Setpoint 1: Surge ahead motion

The first section of the 4-corner test investigates the vessel in pure surge ahead motion in head seas. The reference setpoint was changed 1.5m in the positive X direction from $\eta =$ [1.5 -0.9 0°] to $\eta =$ [3.0 -0.9 0°].



(a) LTV-KF velocity estimation error.



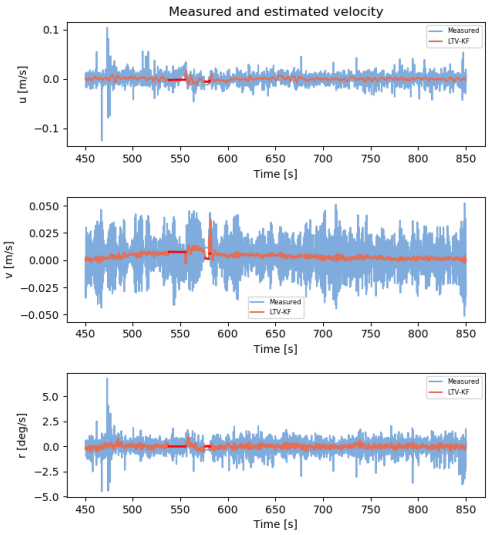
(b) LTV-KF position and heading estimation error.

Figure 9.2: LTV-KF performance during surge ahead motion in very rough seas with position and heading loss.

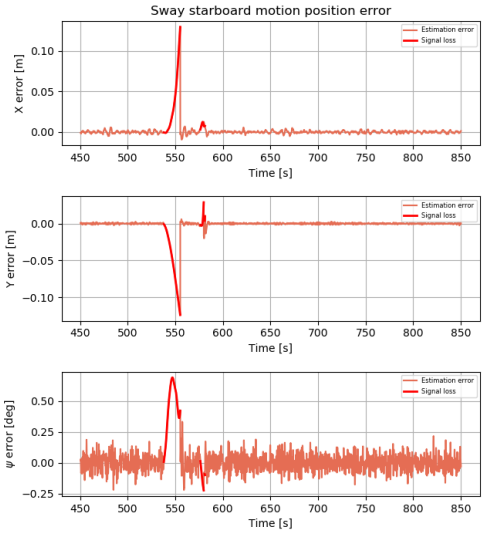
The LTV-KF manages to estimate the true position of the vessel quite well. This is true both when the vessel is standing still in its initial position and when the vessel starts its surge motion. The DP system loses position and heading signal after approximately 150 seconds and initiates deadreckoning as illustrated in Figure 9.2. The signal loss duration corresponds to 3 min in full-scale time. As expected, the observer estimation error grows rapidly in all 3DOF due to only relying on the single input, thruster load. It should be noted that after recovering the position and heading signal, the observer quickly converges close to the true position of the vessel.

Setpoint 2: Sway starboard motion

The second section investigates the vessel in pure sway motion in starboard direction in head seas. The reference setpoint was changed 1.5m in the positive Y direction to $\eta = [3.0\ 0.6\ 0°]$.



(a) LTV-KF velocity estimation error.



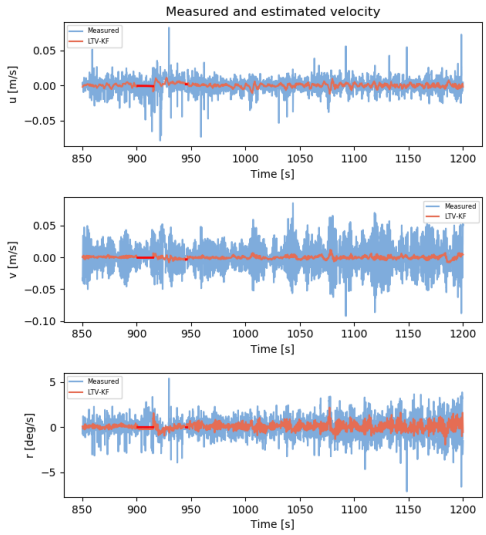
(b) LTV-KF position and heading estimation error.

Figure 9.3: LTV-KF performance during sway starboard motion in very rough seas with position and heading loss.

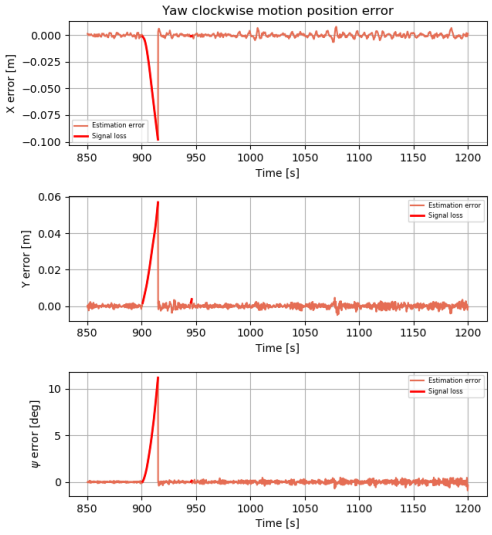
Figure 9.3 illustrates the performance of the observer during the sway starboard motion of the vessel. The vessel still has bow seas and moves in a sway position while being pushed in negative surge motion by the waves. The surge error has some small increases in error throughout the sway motion, which is likely a result of the vessel facing large waves at those moments. The vessel loses position and heading signal after 540 seconds which lasts 2 min and 30 seconds in full-scale time. The observer estimation error grows rapidly in surge and sway, but it's low in the heading. This is likely due to the waves meeting the vessel head on, and the wave load does not apply a moment to the vessel turning it around. The observer recovers quickly after regaining position and heading signal. In addition to the controlled signal loss, the MoCap system loses track of CSAD after 575 seconds. The signal loss is approximately 50 seconds in full-scale time. The shorter signal loss does not seem to face the observer with any problem.

Setpoint 3: Yaw clockwise motion

The third section investigates the vessel in pure yaw motion in head and bow quartering seas. The reference heading setpoint was changed 30◦ in the clockwise direction to $\eta = [3.0\ 0.6\ 30°]$.



(a) LTV-KF velocity estimation error.



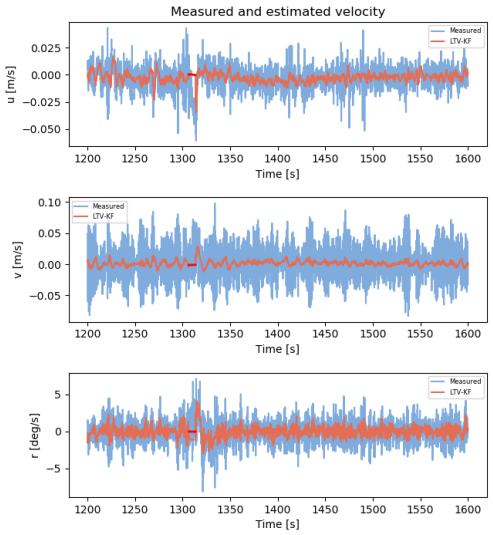
(b) LTV-KF position and heading estimation error.

Figure 9.4: LTV-KF performance during yaw clockwise motion in very rough seas with position and heading loss.

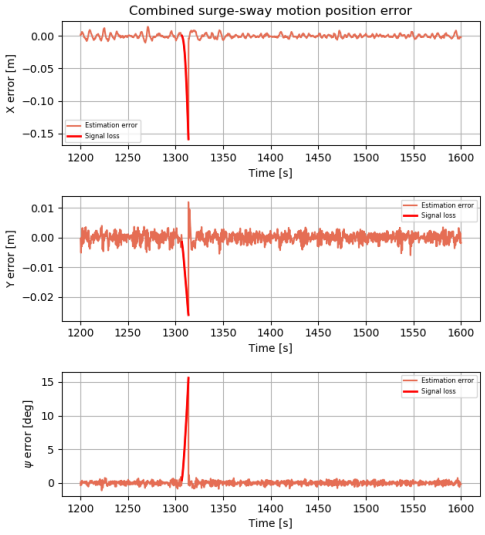
Figure 9.4 illustrates the performance of the observer during the yaw clockwise motion of the vessel. The vessel moves from bow sea towards bow quartering seas. The error of the observer grows when moving to bow-quartering seas. The observer does not manage to filter out all the WF motion when facing waves to the side of the vessel's hull. The vessel loses position and heading signal after 900 seconds which lasts approximately 2 min and 30 seconds in full-scale time. During the signal loss, the X estimation error is quite small, and the sway estimation error is less than in previous tests. However, the heading estimation error grows rapidly before converging quickly when the signal is regained. There is a signal loss due to the MoCap system after 945s which is not notable to the performance of the observer.

Setpoint 4: Combined surge-sway motion

The fourth section investigates the vessel in combined surge sway motion in bow quartering seas. The reference setpoint was changed 1.5m in the negative X direction to $\eta = [1.5\ 0.6\ 30°]$.



(a) LTV-KF velocity estimation error.



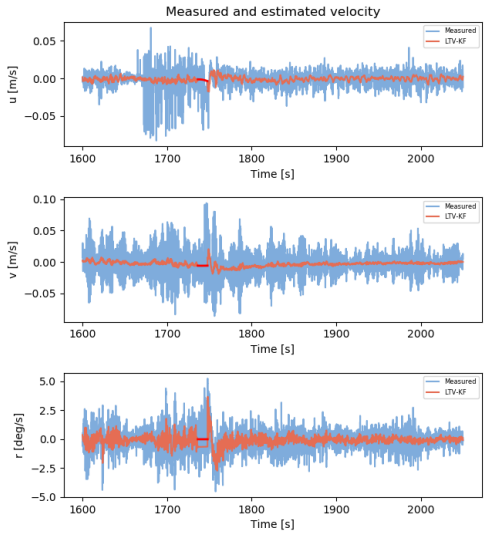
(b) LTV-KF position and heading estimation error.

Figure 9.5: LTV-KF performance during combined surge-sway motion in very rough seas with position and heading loss.

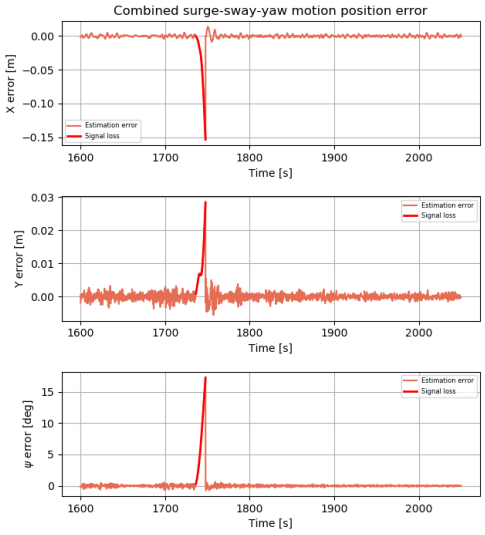
Figure 9.5 illustrates the performance of the observer during the surge-sway motion of the vessel. The vessel losses position and heading signal after 1310 seconds. The signal loss duration corresponds to 1 min and 30 seconds in full-scale time. The observer position error in surge and the heading error grows large quickly. It is expected due to the waves moving in the negative X direction and the vessel being in bow quartering sea, making the wave load apply a moment to the vessel. The sway error is quite low during the signal loss.

Setpoint 5: Combined surge-sway-yaw motion

The fifth section investigates the vessel in combined surge-sway-yaw motion in bow quartering/head seas. The reference setpoint was changed 1.5m in negative Y direction and a heading change of 30∘ counterclockwise to $\eta = [1.5\ -0.9\ 0°]$.



(a) LTV-KF velocity estimation error.



(b) LTV-KF position and heading estimation error.

Figure 9.6: LTV-KF performance during combined surge-sway-yaw motion in very rough seas with position and heading loss.

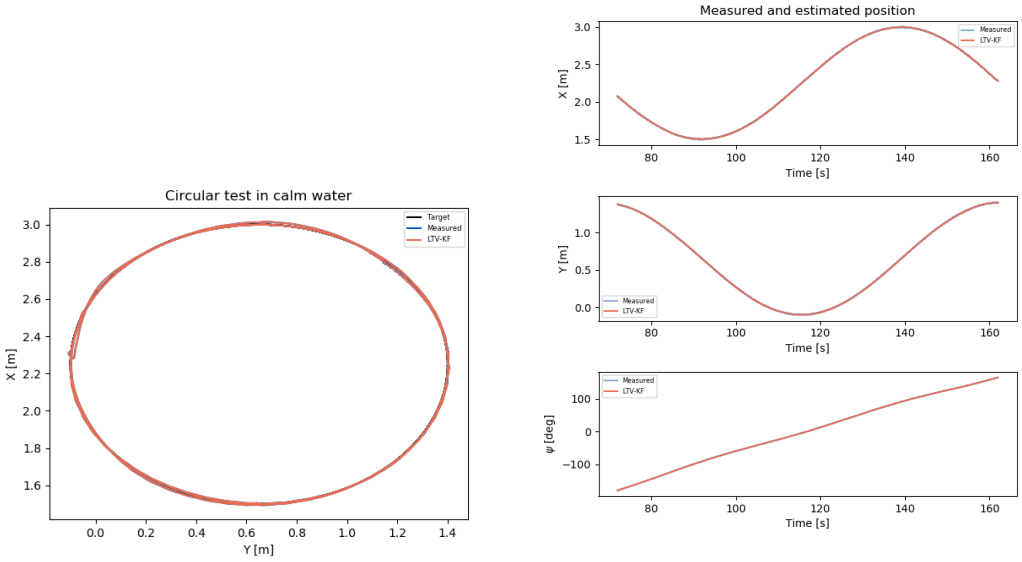
Figure 9.6 illustrates the performance of the observer during the surge-sway-yaw motion of the vessel. The vessel losses position and heading signal after 1740 seconds. The signal loss duration corresponds to approximately 1 min and 45 seconds in full-scale time. Similar to the other tests in bow quartering seas, the observer is struggling in surge and yaw estimation during deadreckoning. Sway error is low throughout the signal loss.

9.2 C/S Enterprise

All tests on CSE were performed in calm water. Firstly a circular test is performed to check the observer's position estimation capability. Afterward, a test where different velocities are given as setpoints are performed to check the accuracy of the velocity estimation. The observer implemented for CSE was the small-gain principle based observer design with new gains, damping matrix, and inertial matrix.

9.2.1 Test 1: Circle

The vessel performed three laps in a circular shape with a diameter of 1.5m. Figure 9.7 illustrates the performance of the observer during the circle test. Note, Figure 9.7b shows the measured and estimated position from the second circle only. The observer is able to estimate the position and heading well throughout the whole circle. The performance is equally good throughout the whole circle.



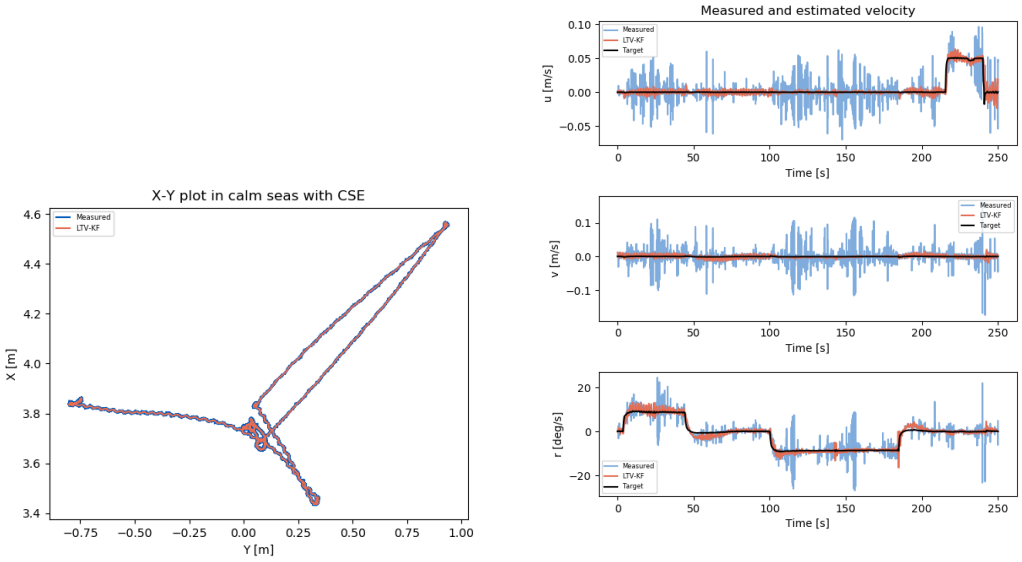
(a) Trajectory of CSE in calm water during circle test.

(b) Position estimation error during circle tests in calm water.

Figure 9.7: LTV-KF performance during circle test performed in calm water with CSE.

9.2.2 Test 2: Velocity setpoints

The velocity test was performed using a joystick controller, which sent out velocity setpoints for the vessel. The test is conducted with lots of turning since the MoCap system in MC-Lab covers a relatively small area. Figure 9.8 illustrates the trajectory of CSE and the velocity accuracy of the observer during the test. The observer is having some problems estimating the velocity of the vessel. When turning at maximum speed, the velocity estimations have a fast oscillatory behavior with small magnitudes. The estimation error also increases when the vessel moves in a surge ahead motion after 200s.



(a) Trajectory of CSE during velocity test in calm water.

(b) Velocity estimation error during velocity tests in calm water.

Figure 9.8: LTV-KF performance during joystick control test performed in calm water with CSE.

9.3 C/S Jonny

The CSJ did not yet have a control system. Therefore, a simple joystick controller was implemented to collect data to evaluate the performance of the observer. The observer implemented for CSJ was the small-gain design with new gains, damping matrix, and inertial matrix.

9.3.1 Test 1: Joystick control

The vessel is driven with a PS4 controller using a simple joystick control. The controller gives command thrust forces directly to the thrust allocator. The vessel is driven in a combined surge-sway-yaw motion testing transient behavior introduced by the operator. CSJ is moving at high speeds compared to CSAD. The observer position estimates are accurate in all three DOFs. The test was conducted to test the performance of the observer in transient movements at higher speeds.

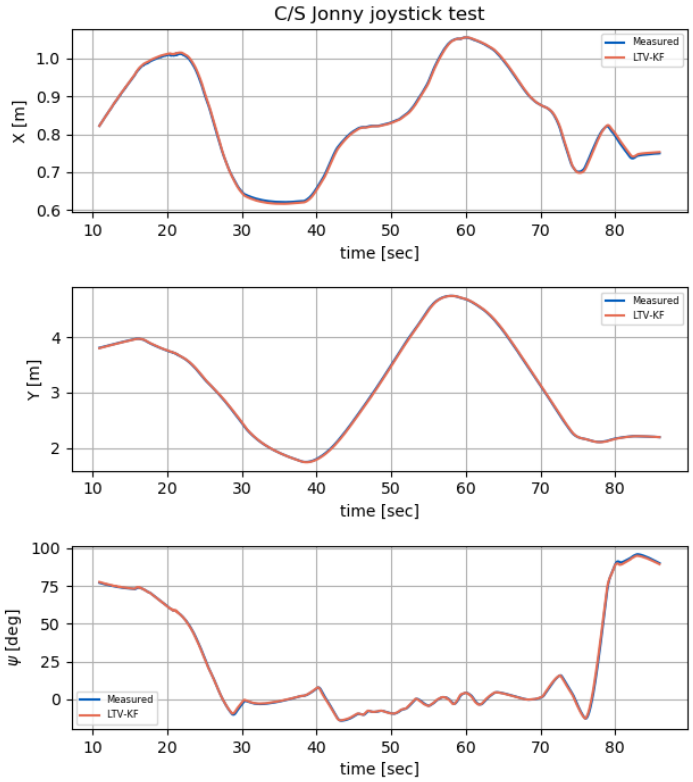


Figure 9.9: Position and heading estimates during joystick control test performed in calm water with CSJ.

9.4 Summary of results

The 4-corner test is used to test the performance of the LTV-KF in the surge, sway, and yaw of motion. Between each setpoint, the system loses the position and heading signal between approximately one and a half to 3 minutes in full-scale time. The LTV-KF estimation error rises quickly when the heading and position signal is lost. Especially heading and X position is difficult for the observer to estimate in deadreckoning. This is likely due to the fact that the vessel is facing head and bow quartering seas throughout the test. However, the estimated position and heading converge fast, close to the actual position, when the position and heading signal is regained. Hence, the observer is robust enough to be used in MC-Lab, where signal loss is normally between 1-5 seconds. The system also lost the position and heading signal for short periods due to the MoCap system. This did not affect the performance of the system considerably. Considering the tests performed, the LTV-KF implemented for CSAD is robust enough to handle shorter signal losses of up to 3 minutes in full-scale time but is not able to perform dead reckoning over longer periods due to not measuring enough internal states.

The small-gain observer design was implemented for both CSE and CSJ. It yielded high

accuracy of the position of both CSE and CSJ. However, the velocity estimation was not as accurate, struggling after fast movements in surge or yaw. The accuracy seems high enough for the small-gain design to be used as a baseline observer for the respective vessels in calm water. Note that tests were only performed in calm water for CSE and CSJ, which for CSAD yielded similar performance for all observers.

Chapter 10

Conclusion

This chapter presents the findings, contributions, and concluding remarks of the thesis. All analyses are conducted with respect to results from a high-fidelity simulator based on the MCSimPython package and the model-scale tests at MC-Lab at the Department of Marine Technology, NTNU.

10.1 Conclusion

The overall goal of this thesis was to compare different observers and find a baseline observer for the model-scale vessel CSAD. Furthermore, an autotuning module has been implemented in order to compare and tune both stochastic and deterministic observers efficiently. Through simulations and laboratory experiments, the results confirm that the LTV-KF has the best performance. Hence, this should be used as a baseline observer for CSAD in further experiments at MC-Lab. The new GUI proved a useful tool for monitoring operations and allows for realtime tuning of observers and plotting of observer states.

The small-gain observer design was implemented for the vessels CSE and CSJ. The observer estimated the position with high accuracy and worked well as a baseline observer for the two vessels. However, the performance of the observer has not yet been tested in waves.

The DFO algorithm PSO was used to autotune both the deterministic observers and the LTV-KF. The autotuning method showed promising results compared to manual tuning but relied on good initial gains in order to not get stuck in a local minimum with poor performance. Using tuning guidelines presented in chapter 6 to find initial gains for the PSO initialization worked well.

The deadreckoning capability of the observers was tested through both complete position and heading signal loss over a longer period of time and multiple short consecutive periods. The observers relied only on the actual thrust load as input during signal loss. When subjected to many short consecutive signal losses, all tested observers managed to estimate

both position and velocity quite well. Resulting in a performance loss on the whole DP system. However, when subjected to longer intervals, the LTV-KF estimation error grew quickly, especially when subjected to waves in quartering bow seas. Overall the LTV-KF deadreckoning capability is robust enough for testing in MC-Lab, where signal loss from the MoCap system ranges from 1-5 seconds in model time. The observer converges close to the actual position quickly after regaining the heading and position signal.

10.2 Further work

- Include more measured states, such as acceleration, to the system. IMU can be used in order to capture the vessel's acceleration. This will further enhance the system's deadreckoning capability.
- Implement a functioning autotuning loop for online tuning of observers in realtime. Functionality for capturing data and tuning the observers based on this data already exists but is currently only used for offline observer tuning.
- A module handling signal faults such as wild points and low variance on the signal should be implemented. The current system only handles frozen signals.
- The thrusters and the electrical system on CSAD need an upgrade. During the model-scale tests, three thruster malfunctions were encountered, with one being replaced. This resulted in only 4 thrusters being used instead of 6.
- The GUI can be further enhanced to allow control over multiple ships at once. Currently, it only supports tracking one ship, but it would be a nice improvement to track multiple ships in MC-Lab.

Bibliography

Audet, C. and M. Kokkolaras (Feb. 2016). 'Blackbox and derivative-free optimization: theory, algorithms and applications'. In: *Optimization and Engineering* 17. DOI: 10.1007/s11081-016-9307-4.

Balchen, J. G., N. A. Jenssen and S. Sælid (1976). 'Dynamic positioning using Kalman filtering and optimal control theory'. In: *symposium on automation in offshore oil field operation* 183, p. 186.

Bass, L., P. Clements and R Kazman (2013). *Software Architecture in Practice. 3rd.* Addison-Wesley.

Bjørnø, J. (2016). *Development of the C/S Inocean Cat I Drillship Model.* MA thesis. NTNU.

Bjørnø, Jon et al. (June 2017). 'Modeling, Parameter Identification and Thruster-Assisted Position Mooring of C/S Inocean Cat I Drillship'. In: V07BT06A019. DOI: 10.1115/OMAE2017-61896.

Eriksen, Bjørn-Olav Holtung and M. Breivik (2017). 'Modeling, Identification and Control of High-Speed ASVs: Theory and Experiments'. In: *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles.* Ed. by Thor I. Fossen, Kristin Y. Pettersen and Henk Nijmeijer. Cham: Springer International Publishing, pp. 407–431. ISBN: 978-3-319-55372-6. DOI: 10.1007/978-3-319-55372-6_19. Available at: https://doi.org/10.1007/978-3-319-55372-6_19.

Fossen, T. I. (2021). *Handbook of Marine Craft Hydrodynamics and Motion Control.* 2nd ed. Wiley-Blackwell. ISBN: 978-1-119-57505-4.

Fossen, T. I. and J. P. Strand (1999). 'Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel'. In: *Automatica* 35.1, pp. 3–16. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(98)00121-6. Available at: https://www.sciencedirect.com/science/article/pii/S0005109898001216.

Fossen, Thor I. and Svein I. Sagatun (1991). 'Adaptive control of nonlinear systems: A case study of underwater robotic systems'. In: *Journal of Robotic Systems* 8.3, pp. 393–412. DOI: https://doi.org/10.1002/rob.4620080307. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.4620080307. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.4620080307.

Gallier, J. (Aug. 2019). 'The Schur Complement and Symmetric Positive Semidefinite (and Definite) Matrices'. In: *Dept. of Comp. and Info. Sci., School of Eng. and Applied Sci. Lecture notes*.

Grimble, M.J., R.J. Patton and D.A. Wise (May 1980). 'Use of Kalman filtering techniques in dynamic ship-positioning systems'. In: *IEE Proceedings D (Control Theory and Applications)* 127 (3), 93–102(9). ISSN: 0143-7054. Available at: https://digital-library.theiet.org/content/journals/10.1049/ip-d.1980.0015.

Hyggen, J. E., H. Mo and M. Konshaug (2023). *Global Optimization Toolbox Solver Characteristics*. Available at: https://wave-model.readthedocs.io/en/latest/ (Accessed: 17th Mar. 2023).

Kaliappan, V. and M. Thathan (Nov. 2014). 'Design of optimum pid controller for non-linear process using evolutionary algorithms'. In: *Journal of Theoretical and Applied Information Technology* 69, pp. 522–529.

Khalil, H. K. (2015). *Nonlinear Control, global edition*. Pearson Education Ltd.

Liu, K. Z. and Y. Yao (2016). *Robust Control: Theory and Applications. (chapter 12)*. John Wiley & Sons (Asia) Pte Ltd.

Loría, A. et al. (2005). 'A nested Matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems'. In: *TAC* 50.2, pp. 183–198. ISSN: 0018-9286. DOI: 10.1109/TAC.2004.841939.

Løvås, H. S. (2019). *DP Autotuning by use of Derivative-free Optimization*. MA thesis. NTNU.

Lyngstadaas, O. N. (2018). *Ship Motion Control Concepts Considering Actuator Constraints*. MA thesis. NTNU.

MATLAB (2022). *Global Optimization Toolbox Solver Characteristics*. Available at: https://se.mathworks.com/help/gads/improving-optimization-by-choosing-another-solver.html (Accessed: 8th Dec. 2022).

Midtun, E, M. L. Kvebaek and S. A. Elvenes (Dec. 2022). *Project Thesis: Develop a new cybership ASV for MC-Lab, with further attention on DP observer designs and tuning.* IMT. Trondheim, Norway: NTNU.

Nørgaard Sørensen, Mikkel Eske et al. (2018). 'A Dynamic Window-Based Controller for Dynamic Positioning Satisfying Actuator Magnitude Constraints'. In: *IFAC-PapersOnLine* 51.29. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018, pp. 140–146. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.09.483. Available at: https://www.sciencedirect.com/science/article/pii/S2405896318321724.

NTNU (2022). *Marine cybernetics teaching laboratory*. Available at: https://www.ntnu.edu/imt/lab/cybernetics (Accessed: 11th Dec. 2022).

Perrault, D. E. (Mar. 2021). 'Probability of Sea Condition for Ship Strength, Stability, and Motion Studies'. In: *Journal of Ship Research* 65.1, pp. 1–14.

Petersen, K. B. and M. S. Pedersen (Nov. 2012). *The Matrix Cookbook*. Available at: https://www.matrixcookbook.com.

Plotjuggler (2022). *Plotjuggler*. Available at: https://www.Plotjuggler.io/ (Accessed: 11th Dec. 2022).

QT (2022). *QT*. Available at: https://www.qt.io/ (Accessed: 11th Dec. 2022).

ROS (2022). *Rosbag documentation*. Available at: http://wiki.ros.org/rosbag (Accessed: 11th Dec. 2022).

Saelid, S., N. Jenssen and J. Balchen (1983). 'Design and analysis of a dynamic positioning system based on Kalman filtering and optimal control'. In: *IEEE Transactions on Automatic Control* 28.3, pp. 331–339. DOI: 10.1109/TAC.1983.1103225.

Sahib, Mouayad A. and Bestoun S. Ahmed (2016). 'A new multiobjective performance criterion used in PID tuning optimization algorithms'. In: *Journal of Advanced Research* 7.1, pp. 125–134. ISSN: 2090-1232. DOI: https://doi.org/10.1016/j.jare.2015.03.004. Available at: https://www.sciencedirect.com/science/article/pii/S209012321500034X.

Simon, D. (2006). *Optimal State Estimation: Kalman, H infinity, and Nonlinear Approaches*. John Wiley & Sons.

Skjetne, R. (Jan. 2021). *Lecture notes of TMR4243 Marine Control Systems II*. IMT. Trondheim, Norway: NTNU.

— (Mar. 2022). *Technical note: DP observer designs*. Trondheim, Norway: NTNU.

— (Feb. 2023). *Lecture 4: DP control design*. IMT. Trondheim, Norway: NTNU.

AMOS DP Research Cruise 2016: Academic Full-Scale Testing of Experimental Dynamic Positioning Control Algorithms Onboard R/V Gunnerus (June 2017). Vol. Volume 1: Offshore Technology. International Conference on Offshore Mechanics and Arctic Engineering. V001T01A080. DOI: 10.1115/OMAE2017-62045. eprint: https://asmedigitalcollection.asme.org/OMAE/proceedings-pdf/OMAE2017/57632/V001T01A080/2532576/v001t01a080-omae2017-62045.pdf. Available at: https://doi.org/10.1115/OMAE2017-62045.

Sontag, E. D. (2008). *Input to state stability: Basic concepts and results. In Nistri, P. and Stefani, G., editors, Nonlinear and Optimal Control Theory, volume 1932.* Springer-Verlag, Berlin, Heidelberg.

Sørensen, A. (2018). *Marine Cybernetics Towards Autonomous Marine Operations and Systems. Lecture notes of TMR4240 Marine Control Systems I*. IMT. Trondheim, Norway: NTNU.

Sørensen, Mikkel Eske Nørgaard and Morten Breivik (2015). 'Comparing Nonlinear Adaptive Motion Controllers for Marine Surface Vessels'. In: *IFAC-PapersOnLine* 48.16. 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC 2015, pp. 291–298. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.10.295. Available at: https://www.sciencedirect.com/science/article/pii/S2405896315021862.

Tannuri, E. A. and H. M. Morishita (2006). 'Experimental and numerical evaluation of a typical dynamic positioning system'. In: *Applied Ocean Research* 28.2, pp. 133–146. ISSN: 0141-1187. DOI: https://doi.org/10.1016/j.apor.2006.05.005. Available at: https://www.sciencedirect.com/science/article/pii/S0141118706000587.

Teel, A.R. (1996). 'A nonlinear small gain theorem for the analysis of control systems with saturation'. In: *IEEE Transactions on Automatic Control* 41.9, pp. 1256–1270. DOI: 10.1109/9.536496.

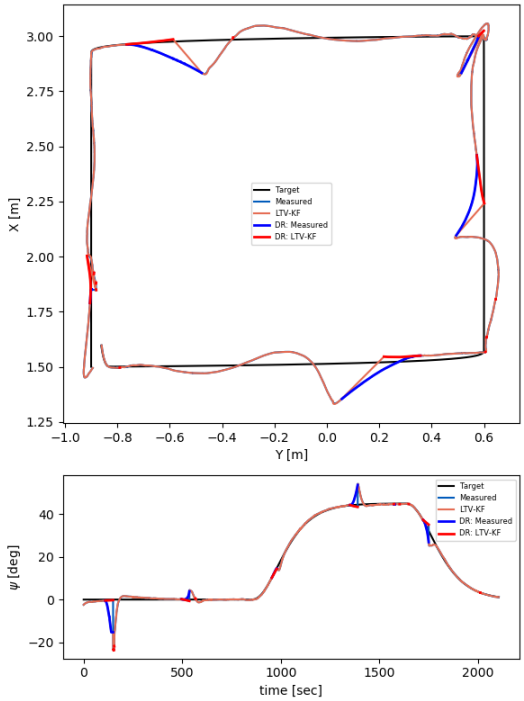
Værnø, S. A. et al. (2019). 'Comparison of control design models and observers for dynamic positioning of surface vessels'. In: *Control Engineering Practice* 85, pp. 235–245. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2019.01.015. Available at: https://www.sciencedirect.com/science/article/pii/S0967066119300085.

Veritas, D. N. (2011). *Dynamic positioning systems - operation guidance*.

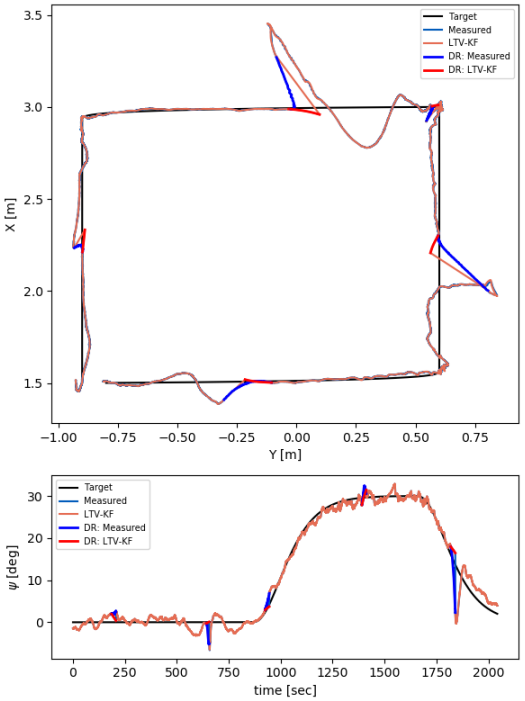
Appendix A

Additional results

4-corner tests performed in all sea states with controlled deadreckoning intervals.



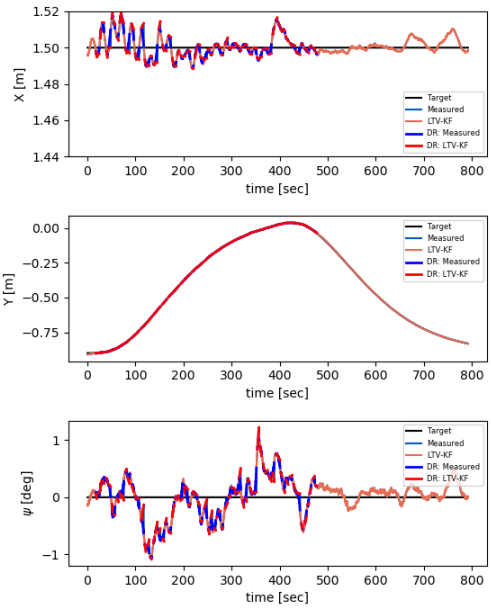
(a) LTV-KF performance in calm waters with deadreckoning intervals of approximately 35-40 seconds.



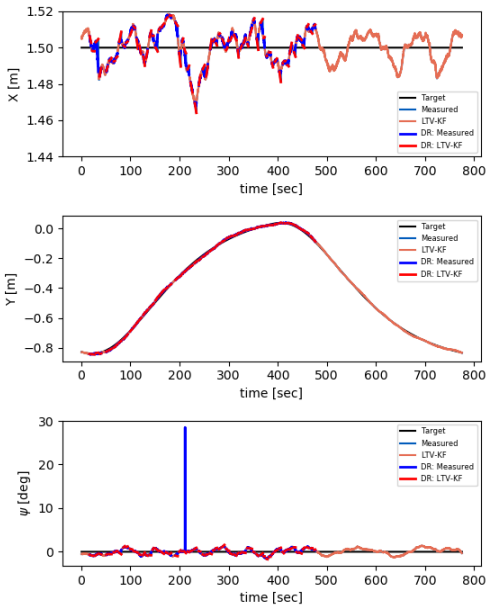
(b) LTV-KF performance in rough seas with dead reckoning intervals of approximately 15-20 seconds.

Figure A.1: Trajectory of the 4-corner tests in very rough seas with deadreckoning intervals of 10-20 seconds.

Deadreckoning test performed with frequent short-duration position and heading dropouts.



(a) LTV-KF performance in calm waters with frequent deadreckoning intervals of approximately 1-5 seconds.



(b) LTV-KF performance in rough seas with frequent dead reckoning intervals of approximately 1-5 seconds.

Figure A.2: Trajectory of the combined motion tests in calm water and the rough sea state with frequent deadreckoning intervals of 1-5 seconds.

Appendix B

GUI

This appendix contains a few figures showing different views of the GUI.

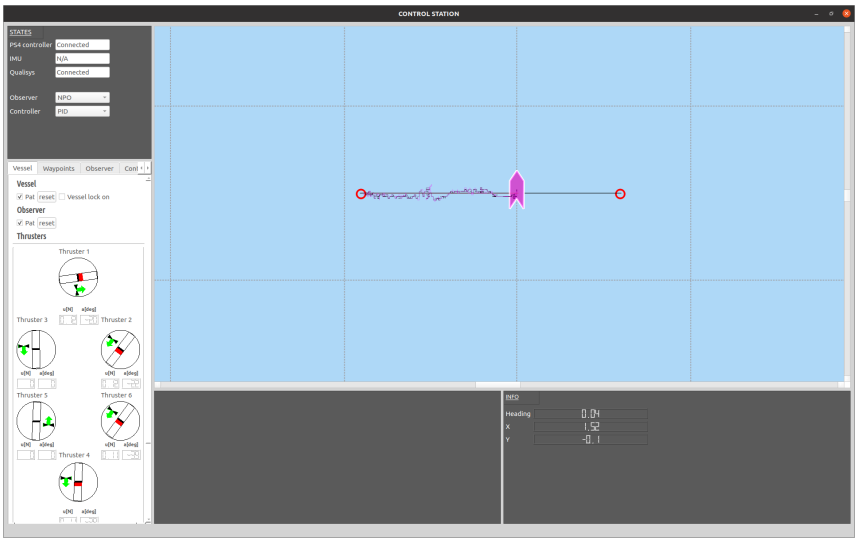


Figure B.1: Layout of the GUI, both the measured and estimated position are plotted.

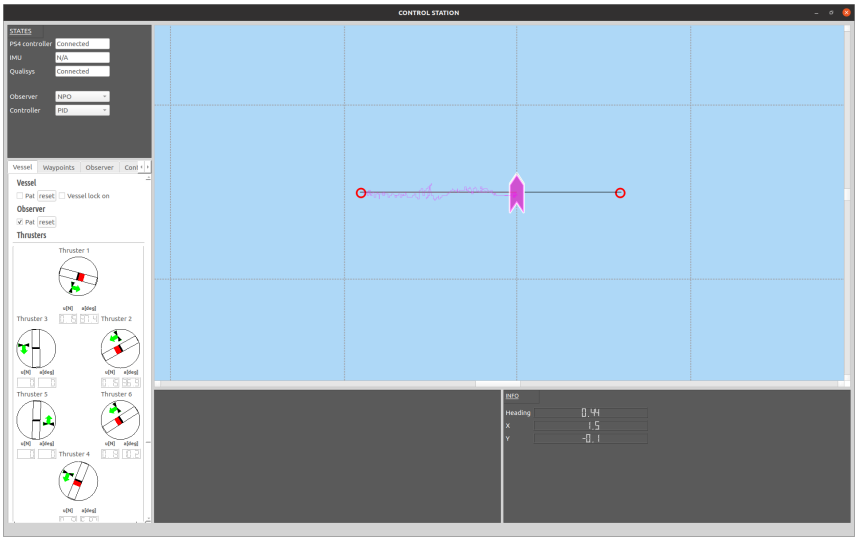


Figure B.2: Layout of the GUI. The estimated position is plotted while the measured position is hidden.

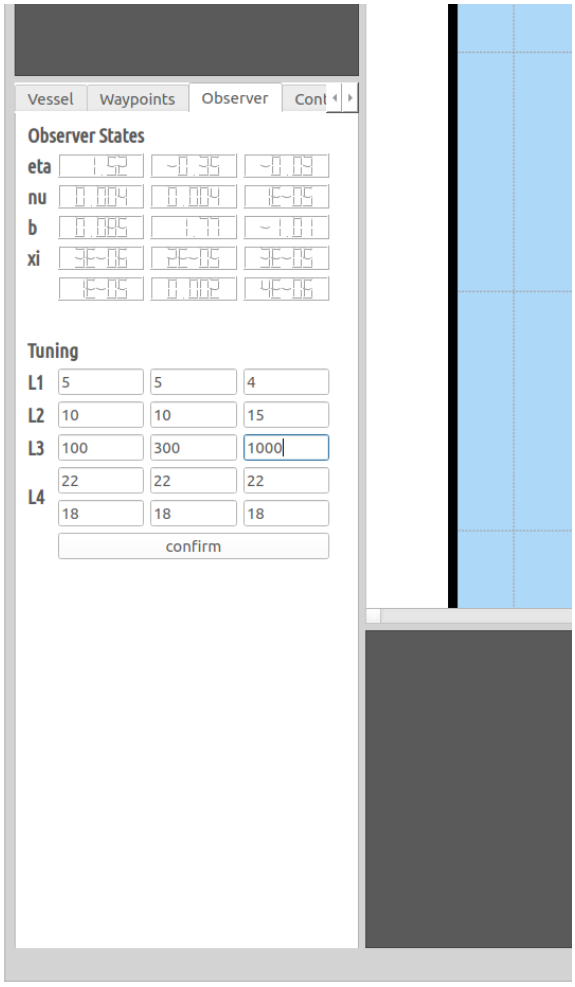


Figure B.3: The observer tab allows for realtime observation of observer states and tuning of gains.

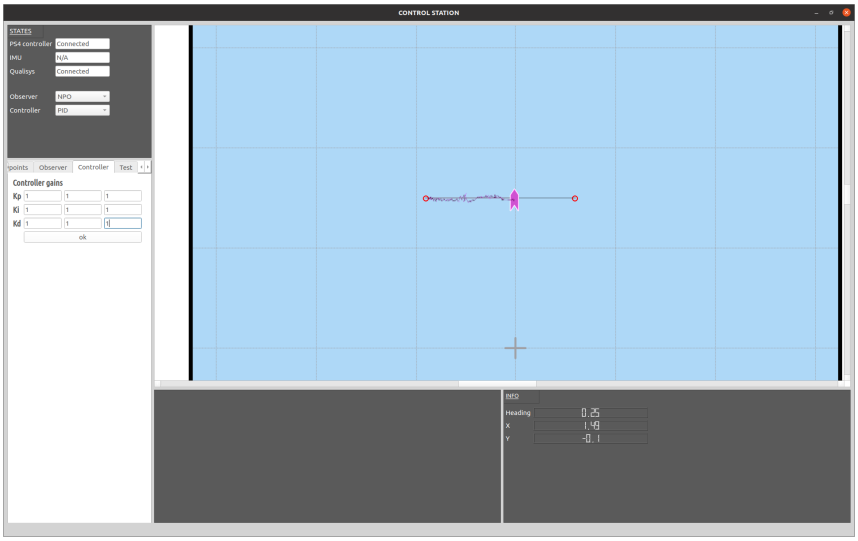


Figure B.4: Tuning of the controller. The controller tab lets the operator directly change the gains of the PID controller.

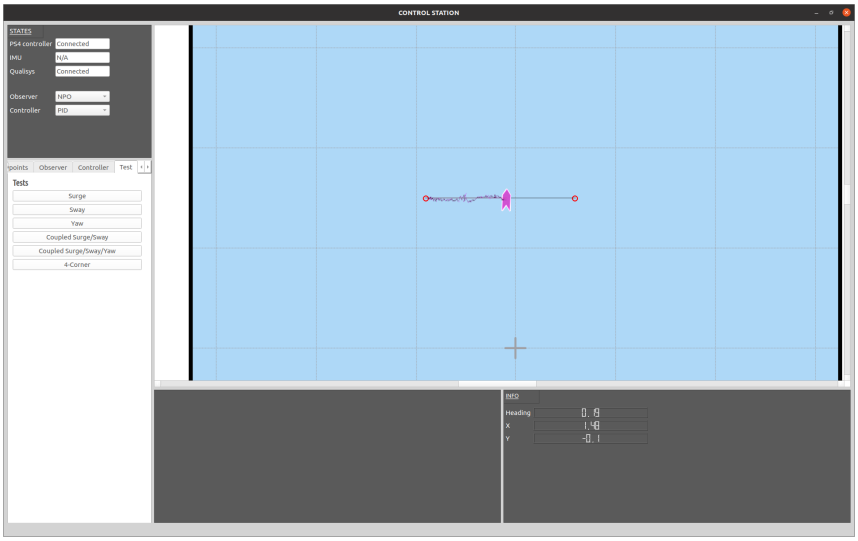


Figure B.5: Interactive test screen. Allows the operator to easily configure tests to be later activated with just one click.

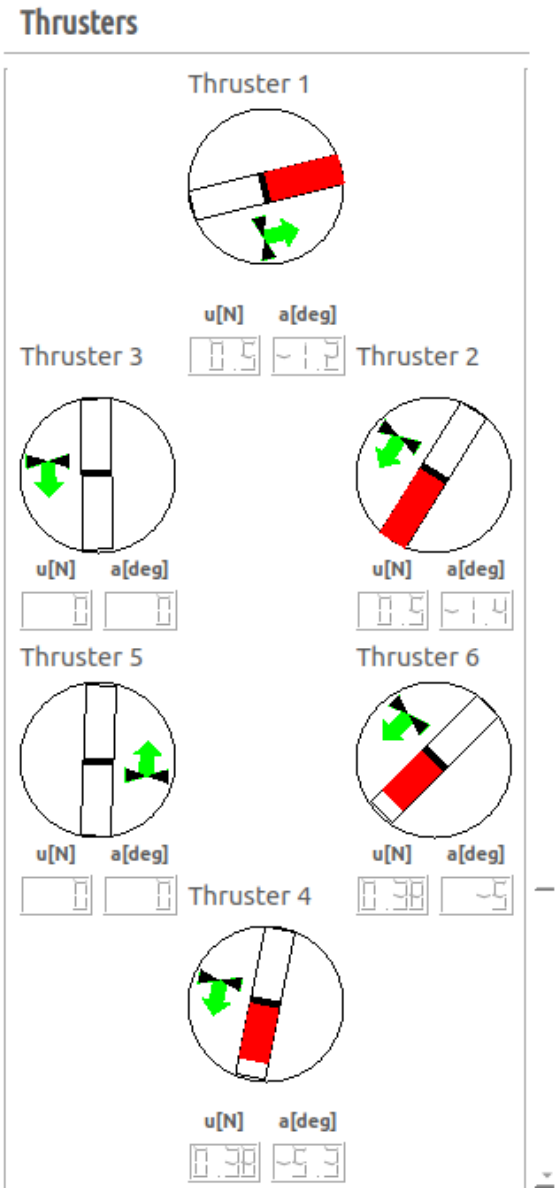
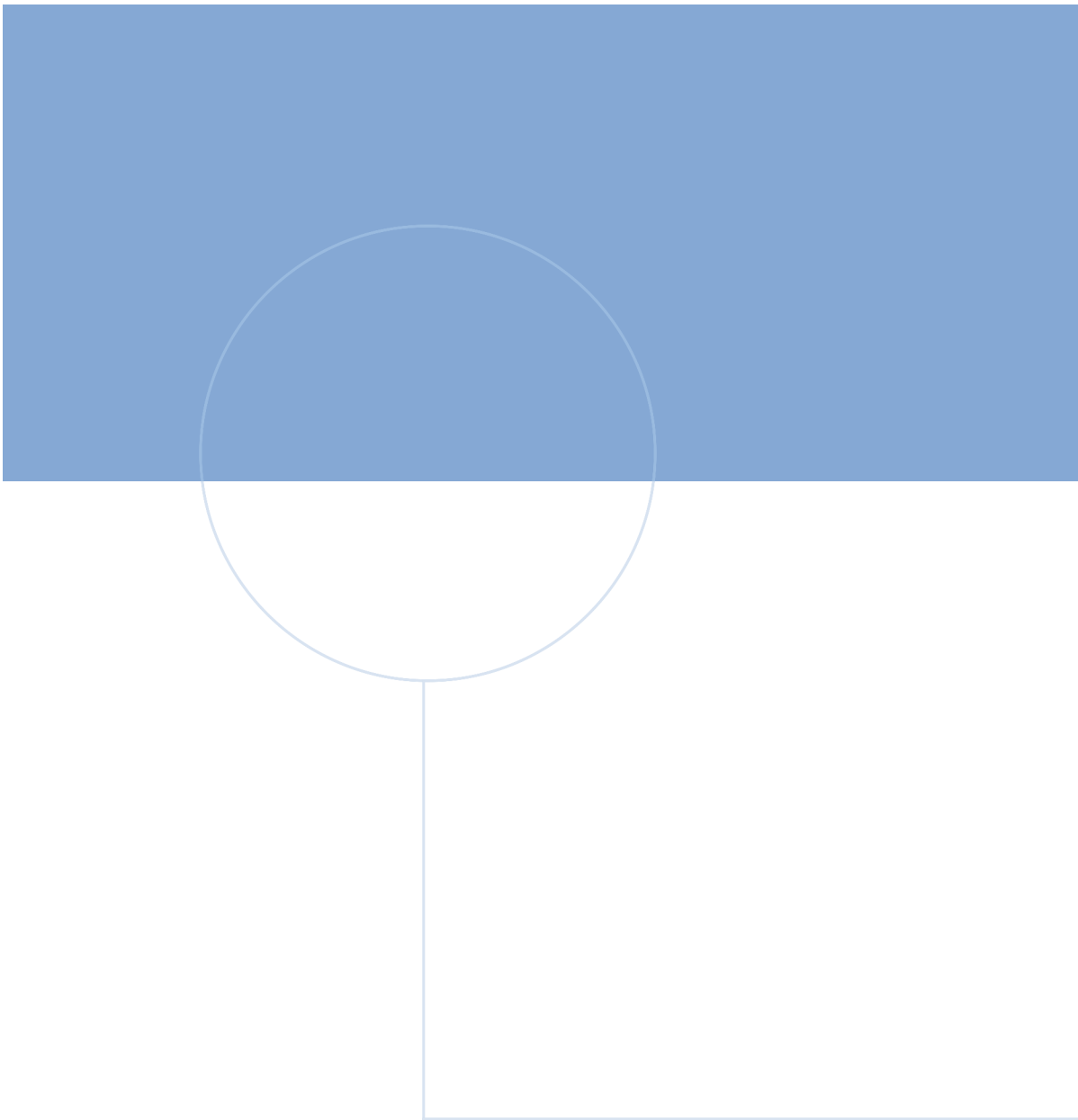


Figure B.6: Zoom in on thruster overview. Positive thrust is marked with red and moves in the positive direction from the center of the circle. Negative thrust marked with blue moves in the opposite direction, with 0 thrusts being marked with just a black line in the center. The green arrow shows the direction of the propeller.



NTNU

Norwegian University of
Science and Technology