NTNU – Trondheim
Norwegian University of
Science and Technology

Remote Control and Automatic Path-following for C/S Enterprise I and ROV Neptunus.

Fredrik Sandved

Marine Technology
Submission date: June 2015
Supervisor: Roger Skjetne, IMT

Norwegian University of Science and Technology
Department of Marine Technology

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

PROJECT DESCRIPTION SHEET

Name of the candidate:	Fredrik Sandved
Field of study:	Marine control engineering
Thesis title (Norwegian):	Fjernstyring og automatisk banefølging for C/S Enterprise I and ROV Neptunus.
Thesis title (English):	Remote control and automatic path-following for C/S Enterprise I and ROV Neptunus.

Background
There has been an increased interest in researching and developing technical solutions for underwater vehicles. We now possess the necessary knowledge and technology to perform complex underwater operations with high precision, such as seabed mapping, online underwater monitoring, subsea installations, and maintenance on pipes. There are many types of underwater vehicles on the commercial market: Remotely Operated Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs), and Unmanned Underwater Vehicles (UUVs) to mention the most common types. In most cases, underwater vehicles are expensive and not affordable to private individuals. Lately, a focus has been to develop low-cost solutions. Contributions to this development were carried out fall 2014, as a project thesis (Follestad, Sandved and Valle (2014)). The result was a low-cost ROV prototype, called Neptunus.

In this MSc thesis the targets are more advanced control functions for ROV Neptunus. Correspondingly, detailed system identification shall be conducted, where full-scale tests and computer analysis should be considered. The results shall be compared with estimated parameters from previous work. Moreover, a guidance system shall be designed, and a path-following control mode shall be developed. Simulations, analysis, and comparison of the path-following algorithm shall be carried out. Since the marine vessels will be operating in an environment with ocean currents, an adaptive controller shall be considered to compensate this load.

Since full lab-functionality for ROV Neptunus is still not available, the candidate will use the cybership CS Enterprise I (CSE1) as a development and test platform. This will also include a remote monitoring and control functions for CSE1, with iPad control functionality.

Work description

1) Perform a literature review, providing relevant references on:
    - Remote control of robotic vehicles, relevant software and hardware solutions, including safety issues and methods for allowing vessel command given to or taken from a remote control location (Give command; Take command; etc.).
    - Mathematical modelling of surface and underwater vehicles.
    - Relevant guidance methods for marine vessels, and path-following algorithms for underactuated vessels.
    - Adaptive control in presence of environmental disturbances.

    Write a list with abbreviations and definitions of terms and concepts, explaining relevant concepts.

2) Present the CSE1 vessel, control system, and mathematical model, including all relevant model parameters with numerical values. Present the numerical models for CSE1 (both HIL and non-HIL).

3) Implement and test a remote control and monitoring function for CSE1, considering:
    - Monitoring vessel responses from a remote location.
    - Controlling the vessel from remote location, including fail-safe if communication is lost.
    - Safe method for giving command to remote station, and taking command.
    - Remote monitoring and control by iPad or similar portable platform.

4) Develop a mathematical model of the ROV Neptunus, and find the model parameters, considering:
    - Full-scale tests.
    - Sesam software package.
    - Empirical formulas and expressions from literature.

5) Set up a Simulink-based simulation model of ROV Neptunus, and perform a set of benchmark tests for testing the response. Compare the resulting responses with the estimated simulation model from the project thesis by Follestad, Sandved and Valle (2014).

6) Develop guidance and control algorithms to achieve a path-following control mode in the horizontal plane for a marine vessel, including some integral action to compensate ocean currents. Moreover:
   - Implement the path-following control mode to the simulation model of CSE1.
   - Implement the path-following control mode to the simulation model of ROV Neptunus.

   For both vessels, simulate to test the algorithms with a set of relevant test scenarios. Analyze and discuss the resulting responses.

7) For CSE1, implement the path-following control mode in the cRIO-based control system using Matlab Simulink and NI VeriStand, including an adequate user interface (UI) to specify path.
   - Perform HIL simulations to verify the control algorithm, UI, and signal interfaces.
   - Perform testing in MC Lab, only if HIL simulation is successfully concluded.

Guidelines

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

The report shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. The report shall be written in English (preferably US) and contain the following elements: Abstract, acknowledgements, table of contents, main body, conclusions with recommendations for further work, list of symbols and acronyms, references, and optional appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with a printed and electronic copy to 1) the main supervisor and 2) the external examiner, each copy signed by the candidate. The final revised version of this thesis description must be included. The report must appear in a bound volume or a binder according to the NTNU standard template. Computer code, pictures, videos, data series, and a PDF version of the report shall be included electronically.

Start date: 14 January 2015 **Due date:** 10 June 2015

Supervisor: Prof. Roger Skjetne
Co-advisor(s): PhD candidates Andreas Reason Dahl and Mauro Candeloro

____

Roger Skjetne
Supervisor

Abstract

Remote control and monitoring of marine operations is in many cases desirable, due to distance, harsh environment, and dangers related to human life, and safety. Marine control systems in interaction with remote control give an accurate and reliable operation, where the aim is e.g. motion control, signal processing, maintenance, and installation. The system can perform its mission with a certain level of involvement from the operator, located remotely.

The topic of this thesis is remote control and monitoring of two marine vehicles performing path following operations. The marine vehicles are the remotely operated vehicle *Neptunus*, a low-cost underwater vehicle, and *Cybership Enterprise I* (CSEI), a model-scale vessel in *Marine Cybernetics laboratory* (MC lab) at NTNU.

Mathematical models for both vessels are established, based on Fossen's robot-like vectorial model. For Neptunus, system identification is performed, and the parameters are found using full scale towing tests, software simulations and expressions from the literature. For CSEI, these parameters have been calculated in previous works, and are only briefly presented.

Simulation models for Neptunus and CSEI are set up, and a set of benchmark tests are conducted for validation. A comparing study against an estimated model of Neptunus is performed. Simulations revealed that the estimation model are in compliance with the identified model obtained in this thesis.

Path following algorithms working in the horizontal plane (surge, sway, and yaw) for Neptunus and CSEI are developed and tested. The control systems vary in complexity and performance related to their control objectives. In all the control systems, integral action to account for current is implemented. Neptunus is underactuated, and CSEI is made underactuated by disabling the bow thruster, and mapping the Voith Schneider propellers to a rudder-propeller system. This is done using the first order Nomoto model of the yaw dynamics. Simulations rendered satisfying and stable behavior for the two vehicles. However, for more advanced motion control systems, full and accurate state feedback is needed for optimal performance, together with a robust mathematical model of the system.

Neptunus is not lab-functional. Thus, CSEI is working as test platform. For testing of the performance, hardware in the loop (HIL) is conducted. Successful HIL opened for model-scale testing in the MC lab. The establishment of a interface that allowed communication between CSEI and a tablet opened for remote control and monitoring. The tablet is used to tune controller gains online and to investigate the control objective real time.

Norwegian summary

Fjernstyring og monitorering av marine operasjoner er i mange tilfeller ønskelig, pga. avstand, omgivelser og farene involvert for menneskets liv og sikkerhet. Fjernstyring i interaksjon med marine kontrollsystemer gir en nøyaktig og sikker operasjon, hvor målet kan være regulering av bevegelse, signalprosessering, vedlikehold og installasjon. Operasjonen blir utført helt eller delvis i interaksjon med en operatør.

Emnet for denne oppgaven er fjernstyring og monitorering av to marine fartøy som utfører banefølgingsoperasjoner. Disse fartøyene er ROV *Neptunus*, en lavkost-ROV, og *Cybership Enterprise I* (CSEI), et modellskip i *Marine-Cybernetics laboratory* (MC-lab) på NTNU.

En matematisk modell for begge fartøyene er satt opp, basert på Fossen (1991) sin robotmodell. Modellparameterene til Neptunus er funnet ved hjelp av fullskala tauetester, softwaresimuleringer, og metoder fra litteraturen. Modellparameterene til CSEI har blitt funnet i tidligere arbeid, og er kun kort gjengitt i denne oppgaven.

Simuleringsmodeller for Neptunus og CSEI er satt opp, og et par referansetester er utført for å sjekke ytelsen. En sammenlikningsstudie av Neptunus opp mot en tidligere estimert model er gjennomført. Simuleringene viser at det er likhet med estimeringsmodellen og den identifiserte modellen i denne oppgaven.

Banefølgingssystemene er utviklet og testet i det horisontale (jag, svai og gir bevegelsene er regulert) for Neptunus og CSEI. Systemene varierer i kompleksitet og hvordan kontrollobjektivene blir løst. I alle systemene er det implementert integralvirkning, som motvirker effekten av strømninger. Neptunus er underaktuert, og videre er CSEI antatt underaktuert ved å slå av baugpropellen. De to Voith Schneider propellene på CSEI blir så antatt å virke som et ror-propell system, gjennom å bruke en første ordens Nomotomodell som girdynamikk. Simuleringer gjengir en godkjent og stabil oppførsel. Avanserte kontrollsystemer er imidlertid avhengig av full tilstandstilgjengelighet for optimal oppførsel, i tillegg til en robust matematisk modell.

Neptunus er enda ikke kompatibel i laboratoriesammenheng, grunnet manglende sensorinstrumentering, og et undervanns-posisjoneringssystem. Derfor er CSEI testplattformen for kontrollsystemene i real-time. I denne oppgaven, er HIL-tester vist og gjennomført. Videre har kontrollsystemene blitt testet på CSEI i en modellskalatest. En applikasjon som åpner for kommunikasjon mellom CSEI og nettbrett har blitt satt opp. Nettbrettet er i testene brukt til å sette kontrollparametere, og til å monitorere oppførselen til skipet på avstand.

Preface

This report is the result of my master thesis within marine cybernetics at NTNU. Working with the thesis has been both exciting and educative, and taught me how extensive marine cybernetics can be.

I would like to thank my supervisor Roger Skjetne for his introduction to the topic, and also for the guidance along the way. Moreover I like to offer my gratitude to PhD candidates Andreas Reason Dahl and Mauro Candeloro for all the help, suggestions and discussions. You are a true inspiration, and I have deeply appreciated all your support and enthusiasm.

A thank goes to Ole Eidsvik for the collaboration during the towings tests, and to senior engineer Torgeir Wahl and Tor Kvestad Idland for the support in the *LabView*/ *VeriStand* framework.

Special thanks are served to the Badgers at A2.023 for the valuable discussions, trouble shooting, and late hours during the semester.

Last, on the personal side, I would like to express my gratitude to the ones closest to me for all the encouragement and support throughout the years of study.

Fredrik Sandved
Trondheim, June 9, 2015

Contents

Nomenclature

AP	Aft Perpendicular
AUV	Autonomous Underwater Vehicle
CAPEX	Capital Expenditure
CB	Center of Buoyancy
CG	Center of Gravity
CLF	Control Lyapunov Function
CO	Center of Origin
CPM	Control Plant Model
cRIO	Compact RIO
CSEI	Cybership Enterprise I
DOF	Degree Of Freedom
DP	Dynamic Positioning
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
ESC	Electronic Speed Controllers
FPGA	Field-Programmable Gate Array
FP	Forward Perpendicular
HD	High-Definition
HIL	Hardware-In-the-Loop
IMU	Inertial Measurement Unit

IO	Input-Output
ISS	Input-to-State Stability
ITTC	International Towing Tank Conference
LARS	Launch And Recovery System
LOS	Line Of Sight
MC	Marine Cybernetics
MSROV	Mid-Size Remotely Operated Vehicle
NACA	National Advisory Committee for Aeronautics
NED	North East Down
NI	National Instruments
NOAA	National Oceanic and Atmospheric Administration
NTH	Norges Tekniske Høgskole
NTNU	Norwegian University of Science and Technology
OCROV	Observation Class Remotely Operated Vehicle
OPEX	Operational Expenditure
OS	Operation System
PID	Proportional-Integral-Derivative
PPM	Process Plant Model
PWM	Pulse-Width Modulation
ROV	Remotely Operated Vehicle
RPM	Rounds Per Minute
SNAME	Society of Naval Architects and Marine Engineers
TMS	Tether Management System
UGAS	Uniformly Global Asymptotic Stability
UGS	Uniformly Global Stability
UI	User Interface
USV	Unmanned Surface Vehicle
US	Uniformly Stable

UUV	Unmanned Underwater Vehicle
VI	Virtual Instrument
WCROV	Work-Class Remotely Operated Vehicle
WiFi	Wireless Fidelity
WP	Waypoint

List of symbols

m	Mass
L	Length
B	Width
H	Height
λ	Model-scale ratio
Z	Number of propeller blades
D	Diameter
P	Propeller pitch
$\frac{A_e}{A_0}$	Blade area ratio
x	North position
y	East position
z	Down position
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
$\boldsymbol{\eta}$	Pose of the vehicle
u	Velocity in surge
v	Velocity in sway
w	Velocity in heave
p	Roll rate
q	Pitch rate
r	Yaw rate
$\boldsymbol{\nu}$	Velocity vector
X	Force in surge
Y	Force in sway
Z	Force in heave
K	Moment in roll
M	Moment in pitch
N	Moment in yaw

$\boldsymbol{J}_\Theta$	Transformation matrix in 6 DOF
$\boldsymbol{R}(\boldsymbol{\Theta})$	Translational transformation matrix
$\boldsymbol{T}(\boldsymbol{\Theta})$	Angular transformation matrix
χ	Course angle
β	Sideslip angle
α_{att}	Angle of attack
U	Vehicle's speed
$\boldsymbol{r}_g$	Vector from CO to CG in x,y,z direction
$\boldsymbol{r}_b$	Vector from CO to CB in x,y,z direction
μ_c	Gauss-Markov process constant
V_c	Ocean current speed
β_c	Ocean current direction
u_c	Ocean current velocity in surge
v_c	Ocean current velocity in sway
$\boldsymbol{\nu}_c$	Ocean current velocity vector
$\boldsymbol{\nu}_r$	Relative velocity vector
$\boldsymbol{M}_{RB}$	Rigid body mass matrix
$\boldsymbol{C}_{RB}$	Rigid body Coriolis matrix
$\boldsymbol{\tau}_{RB}$	Rigid body forces
$\boldsymbol{\tau}_{\mathrm{env}}$	Environmental forces
$\boldsymbol{\tau}_{\mathrm{hyd}}$	Hydrodynamic forces
$\boldsymbol{\tau}_{\mathrm{hs}}$	Hydrostatic forces
$\boldsymbol{\tau}$	Thruster forces
$\boldsymbol{M}_A$	Added mass matrix
$\boldsymbol{C}_A$	Added mass Coriolis forces
$\boldsymbol{D}_L$	Linear damping matrix
$\boldsymbol{D}_q(\boldsymbol{\nu}_r)$	Quadratic damping matrix
ρ	Density of saltwater
g	Gravity constant
W	Weight of vehicle
B	Buoyancy of vehicle
A_{wp}	Waterplane area
$\overline{GM}_T$	Transverse metacentric height
$\overline{GM}_L$	Lateral metacentric height
$N_{(\cdot)}$	Hydrodynamic derivatives (N_r, $N_{\dot{r}}$, $N_{\lvert r\rvert r}$, and so on)
$\boldsymbol{K}$	Thrust coefficient matrix
$\boldsymbol{u}$	Thruster inputs
$\boldsymbol{T}(\boldsymbol{\alpha})$	Thruster configuration matrix
n	Revolutions per second

J_a	Advance ratio
R_N	Reynolds number
η_0	Open water efficiency
V_a	Inflow velocity to propeller
K_T	Thrust coefficient
K_Q	Torque coefficient
C_D	Drag coefficient
$\boldsymbol{W}$	Weighting matrix
δ	Rudder angle
$\tilde{\psi}$, $\tilde{u}$	Error in heading and surge velocity
K_p, K_i	Proportional and integrator gain
$\boldsymbol{\eta}_0$	Initial position pose
ω_n	Natural frequency
ζ	Damping ratio
u_r, u_d	Commanded and filtered reference signal
R_{acc}	Circle of acceptance
Δ	Lookahead distance
χ_p	Path tangential angle
χ_r	Velocity-path relative angle
e	Cross track error
h	Step size
t	Run time
$\boldsymbol{b}$	Bias vector
s	Path parameter
$\boldsymbol{\alpha}$	Virtual control law
$\boldsymbol{\xi}$	Integrator state in maneuvering control design
$\boldsymbol{z}_i$	Error state
V_i	Control Lyapunov function
$\mathsf{B}(\cdot)$	Set containing $(\cdot)$
$\alpha_{1,2}$	Low pass filter time constants

Chapter 1

Introduction

1.1 Background and motivation

According to the *National Oceanic and Atmospheric Administation (2015)* (NOAA), the ocean covers around 71 percent of the earth's surface, and contains 97 percent of the planet's water. Consequently, operations related to resources, trading, and travel are occurring frequently on the oceans.

Today, gained experience and technology has enabled operations far below the sea surface in distant and harsh environments. The use of *unmanned underwater vehicles* (UUV), *unmanned surface vessels* (USV), and remote systems can facilitate exploration, path following, installation and maintenance, where the control and monitoring are being done from the surface, or from operation centers ashore.

Among others, the disciplines regarding marine technology and control theory have been a important promoter for this development. A factor, according to Breivik (2010, Ch. 4), is to use the control theory in order to utilize USVs and UUVs to perform *dirty*, *dull*, *distant* and *dangerous* operations. In this way, the hazard is reduced for the humans participating in the operations. In addition, automatically controlled vehicles can in many cases perform certain operations with higher accuracy and availability.

At the *Norwegian University of Science and Technology* (NTNU), extensive research related to motion control of marine vessel is conducted. The Department of Marine Technology houses the *Marine Cybernetics* laboratory (MC lab), where model-scale vehicles and state of the art systems are being tested, pushing the technology forwards.

The motivation for this master thesis is to contribute further within development of remote control and monitoring of marine vehicles. Using the resources from the de-

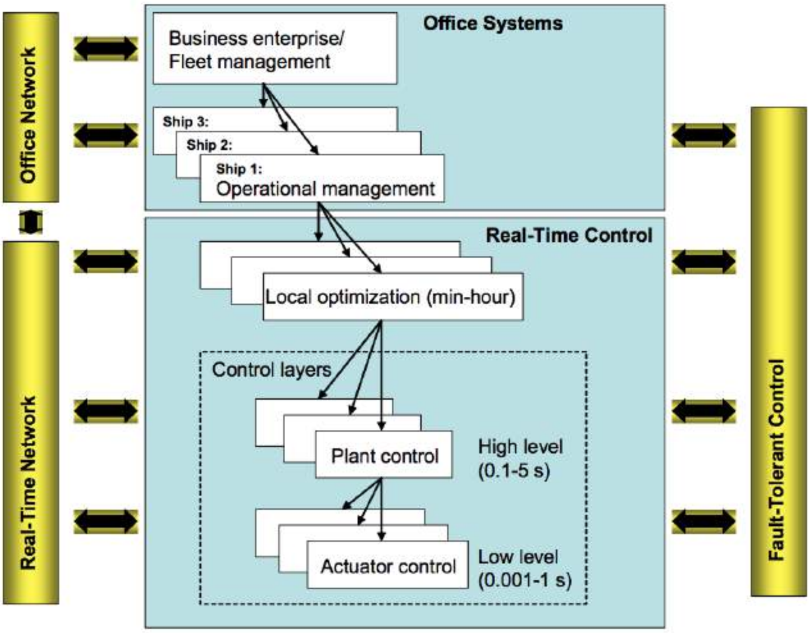


Figure 1.1: The structure of marine control systems. Courtesy of Sørensen (2013).

partment, motion control systems can be established and tested. Furthermore, a motivation is to develop a interface that opens for remote control and monitoring of the vehicles in the MC lab.

1.2 Marine control systems

Since the first automatic ship control system was reported in 1911 (a proportional-integral (PI) controller for ship steering, by Elmer Sperry), marine control systems have developed in a significant manner (Bennett, 1996). Today, the systems can be used in a wide area of operations, from sea keeping and path following control, thruster control, to signal processing and estimation. Systems vary in accuracy and complexity based on the operation of consideration.

According to Sørensen (2013, Ch. 1), it is common to divide marine control systems in two categories; *real time control and monitoring* and *operational and business management*, see Figure 1.1.

Office systems

Operational management is related to life cycle and operation optimization, while business management is concerning fleet allocations and supply chain logistics. This categories are not further elaborated in this thesis, and it is referred to Rensvik et al. (2003) and Sørensen (2013, Sec. 2.8) for further information.

Real time control

Real time control for marine systems is a vast topic. They comprehend engines, propulsion, and power systems. Based on the topic of this thesis, real time control related to motion control is used as an example within this category.

The local optimization block deals with assigning suitable reference points and paths for the vehicle to follow. For example, if the vehicle is to follow a straight-line path, this path is generated within this block.

The *plant control* and *actuator control* are located within the control layers, The plant control calculates the necessary thrust to maintain position, while the actuator control considers the speed, pitch, and torque of the propellers. In this master thesis, the main focus will be within the local optimization and the plant control block.

A typical overview of the plant control for marine applications is given in Figure 1.2. To calculate the desired thruster input to maintain position, a mathematical model is established. The aim of the model is to be a complete representation of the real system. The response is further measured by sensors onboard and processed in terms of filtering and removal of faulty measurements.

If some states or dynamics can not be directly measured by the sensors on board, an *observer* can be used to estimate this behavior. Several types of observers can be utilized, for instance, the *Kalman filter* (see Kalman, 1960), the *non-linear passive observer* (Fossen, 2011, Ch. 11), and *particle filter* (Carpenter et al., 1999).

The processed and estimated measurements are then served as input to the controller. The controller calculates the desired thrust to maintain track or position. A typical controller used in the industry has a *proportional*, a *derivative*, and a *integral* term. While the proportional term multiplies a constant with the error at each step, the derivative term deals with the differentiated error. This term helps to achieve faster convergence. The integral action removes any steady state offset. Together, the three terms forms the *proportional-integral-derivative* (PID) controller, which converges the state to the desired value over time.

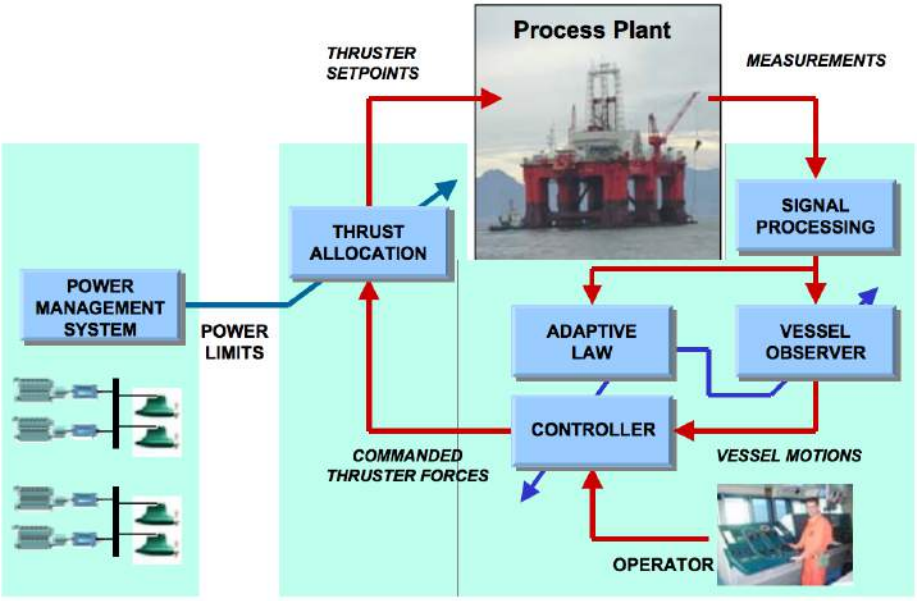


Figure 1.2: Overview of a control system for a marine vehicle. Courtesy of Sørensen (2013).

The operator gives input to the controller block. As a consequence, also the guidance block, which translates the input from the operator into a suitable reference for the vehicle to follow, is connected to this block.

The controller calculates the desired thrust in the different translational and rotational *degrees of freedom* (DOF). The thruster allocation block decompose and distributes the necessary thrust to the correct propellers. As the controller produces transients or too high thrust demand into the allocation block, the power management system is present to monitor the input. Moreover, the aim of the power management system is to match the produced power with the demand. If the thrust request is too high, the power management system will saturate or deny the input (Sørensen, 2013, Ch. 3). The described loop is running as long as the motion control system is activated.

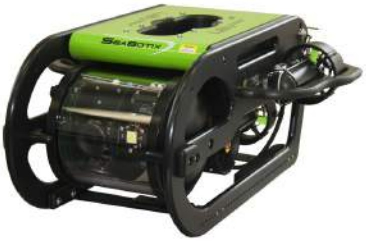


Figure 1.3: The OCROV Seabotics LBV300-5. Courtesy of Seabotics (2015).

1.3 Unmanned marine vehicles

Exploration, maintenance and installation related operations within the marine segment are often performed using unmanned vehicles. The operator can be located on board a mother vessel or at shore. In this section, some unmanned vehicles within the marine segment will be presented briefly.

1.3.1 Underwater vehicles

This section is based on information from Follestad et al. (2014, Ch. 1). For an underwater vehicle operating in the industry, it is common with a operator, who performs remote control and monitoring. The communication is transferred via an umbilical or by acoustic systems.

Remotely operated vehicles

One of the most common type is the *remotely operated vehicle* (ROV). It is usually referred to as the workhorse of the underwater industry, and is used in various tasks related to maintenance and installation. ROVs have a box-shaped appearance, and are often equipped with manipulators in order to perform their tasks. Furthermore, ROVs are controlled partly or fully by a human operator.

Based on the operation type and environment, different types of ROVs are used. For underwater exploration, *observation class ROVs* (OCROV) are often used. These are small vehicles weighing less than 100 kg. Within this classification, the vehicles are limited to a depth rating around 300 m, and generally act as backup to divers, or as a diver substitution, for shallow water inspection tasks (Christ and Wernli Sr., 2014, Sec. 1.1.2). They are normally deployed in water by hand. In Figure 1.3, a typical OCROV is shown.

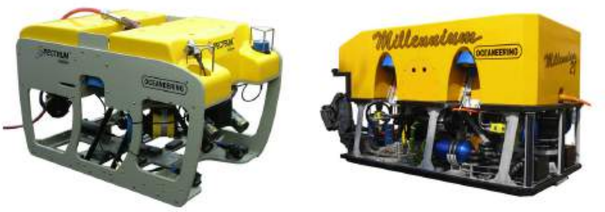


Figure 1.4: The Spectrum ROV, located to the left, and the Millennium Plus ROV, on the right side. Courtesy of Oceaneering (2015).

During maintenance and installations operations, or when the environmental conditions are harsher, it is more common to use bigger and more robust ROVs. The most common types here are the *Mid-Sized ROVs* (MSROV), ranging from 100-1000 kg, or *work-class ROVs* (WCROV) weighing more than 1000 kg, see Figure 1.4. It is also necessary for the MSROV and WCROV to have a *launch and recovery system* (LARS), in addition to a *tether management system* (TMS) (Christ and Wernli Sr., 2014, Sec. 1.1.2).

Autonomous underwater vehicles

The other main class of unmanned underwater vehicles, is the *Autonomous Underwater Vehicle* (AUV). As opposed to the ROVs, AUVs do not a umbilical connected to the body. Instead, the AUV is pre programmed; the commands for the mission are given in advance. Compared to a ROV, the commanded signals from the operator to the vehicle is limited during operation. The AUV is torpedo shaped, and thus more convenient to use in exploration operations. The AUV is mainly working in transit, so in many cases, the AUV is equipped with a main thruster and control surfaces (Antonelli, 2014).

1.3.2 Unmanned surface vessels

USVs have existed at least since World War II, but mainly for military use (Bertram, 2008). However, due to limitations in technology, well working USVs did not exist before the 90s. Today, unmanned surface vessel are used for inspection, coast protection, and exploitation of resources from the surface.

According to Dragland (2014), 75 percent of the accidents on sea are caused by humans. In addition, due to the technology, the need for onboard crew at sea is falling. Hence, a solution is to use USVs for freight-related operations.

It has been speculated in several method for carrying freight with unmanned vessels. One methods is to have a single unmanned ship, which maneuvers from port to port.

Figure 1.5: A swarm of unmanned ships. Courtesy of Royce (2015).

The monitoring and any control of the ship is done from a remote location. An other solution is to have a mother ship with crew, and several unmanned ships which follow the mother ship. Moreover, this procedure can also be done without a mother ship, and let all monitoring and control happen remotely (see Figure 1.5).

The use of unmanned surface vessels will reduce the fuel and crew cost, in addition to increase the efficiency and availability (Rolls Royce, 2015).

1.4 Methodology

Vehicle modeling: to obtain a mathematical model of the marine vehicles, the robot-like vectorial model from Fossen (1991) is used.

System identification: for the underwater vehicle, system identification is conducted. This includes *full scale towing tests*, *software analysis*, and *estimation formulations* from the literature. The towing tests are conducted in the Marine Cybernetics laboratory.

Mathematical analysis: the vehicles models with the accompanying motion control system are depended on proper stability properties for optimal performance. Using mathematical analysis, the closed loop systems can be proven to show certain behavior with respect to the stability.

Simulations: to test the performance of the obtained mathematical models and control systems, simulations in *MATLAB*/*Simulink* are performed.

Hardware-in-the-loop testing (HIL): for additional, and high fidelity simulations, HIL tests has been conducted. In a HIL test, a comprehensive simulation model of the vehicle is connected to the control system. The goal is to test the performance and reduce the risk before sea trials.

Marine cybernetics laboratory: is a small test basin at NTNU[1]. Due to the small basin size and advanced instrument package, it is suitable for testing motion control system for marine vehicles. It is operated by the department of marine technology, and is used by master and PhD students, as well as the industry.

Model-scale tests: upon completing the HIL-test, model-scale tests on CSEI in the MC-lab are performed. Since Neptunus is not full lab functional, only CSEI is considered.

Remote control: since operations may be *dirty*, *dull*, *dangerous*, and *distant* for humans, remote control is considered. In this thesis, a tablet application is used to remotely control and monitor CSEI in the MC lab.

[1] For further information regarding the laboratory, see ntnu.no/imt/lab/cybernetics.

1.5 Contributions

The main contributions of this thesis are:

- a mathematical model of Neptunus, where the system identification has been conducted with towing tests in the MC lab[2], software analysis in HydroD, and expressions from the literature.
- a comparison of the mathematical model with the results from Follestad et al. (2014), where all the system identification was performed using estimation methods from the literature.
- thruster characteristics for Neptunus that are in compliance with real propulsion tests, conducted by OpenROV.
- a transformation of the propellers on CSEI (Voith Schneider) into a rudder-propeller system. The yaw dynamics is defined using a first order Nomoto model.
- a mathematical stability proof of the maneuvering by backstepping control design with integral action.
- the implementation of heading on waypoints, lookahead-based line of sight, and maneuvering by backstepping control algorithms for Neptunus and CSEI, with integral action for accounting the effect of current. The control algorithms have been tested in offline simulations, hardware-in-the-loop, and in model scale experiments.
- a communication interface between CSEI and iPads and Android-supported devices.

[2] The towing tests are done in collaboration with Eidsvik (2015).

1.6 Outline of thesis

In this section, the contents of each chapter are briefly presented. The use *italic* notation in the thesis represents key words.

- **Chapter 2** presents the marine vehicles of concern in this master thesis, the ROV Neptunus and the CSEI. This chapter deals with the structure, sensors and propulsion system for each of the vessels.
- **Chapter 3** deals with mathematical modeling of a marine vehicle. System identification for Neptunus through computer analysis, towing tests and estimation methods are presented. The responses of the vehicles are simulated using a set of benchmark tests. For Neptunus, the response is compared against the results from Follestad et al. (2014, Ch. 6).
- **Chapter 4** develops the path following control system which is used to control Neptunus and CSEI along a pre defined path. The control algorithms are *heading on waypoints*, *lookahead-based line of sight*, and *maneuvering control designs by backstepping*. Simulations through a set of defined paths are performed.
- **Chapter 5** develops a solution for remote control of the vehicles. Using software and hardware systems available on the MC lab, remote control and monitoring can be done via iOS and Android devices. Force and thruster control of CSEI from an iPad are presented.
- **Chapter 6** presents the HIL testing the various control systems for CSEI. The HIL tests are conducted within a real-time simulation environment, where the vessel model is a comprehensive representation of the real system.
- **Chapter 7** shows the results related to the model-scale tests on CSEI. The control algorithms make CSEI follow a pre defined path in the MC lab. The iPad application allows remote control and monitoring of the performance.

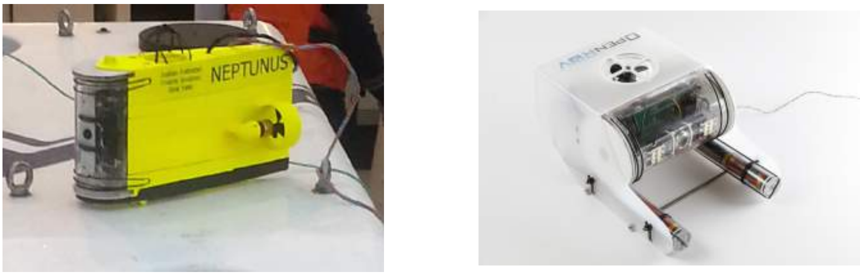
Chapter 2

Experimental platforms

2.1 Neptunus

The underwater vehicle of concern in this master thesis is the ROV Neptunus, developed by Follestad, Sandved and Valle in their project thesis (Follestad et al., 2014). In this section, a brief presentation of the structure and functionality of Neptunus is given.

Neptunus is a small, low-cost ROV prototype, development autumn 2014. The design is based on *OpenROV*[1], and for the current design of Neptunus, the instrumentation is directly adapted thence. The main improvement is the hydrodynamic capability. In addition, Neptunus is supplied with land power, whereas OpenROV is driven by batteries. Figure 2.1 is showing the appearance of Neptunus and OpenROV.



(a) Neptunus

(b) OpenROV v2.6. Courtesy of OpenROV (2015).

Figure 2.1: Modified and original ROV.

[1] Information of the ROV at openrov.com

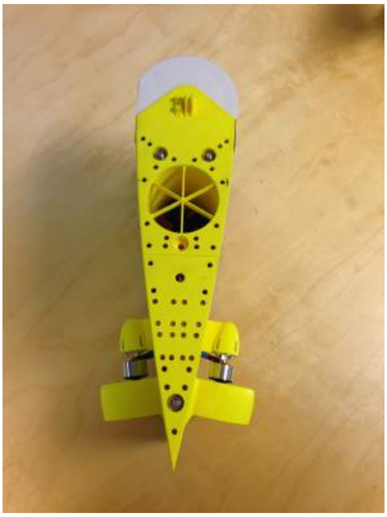


Figure 2.2: Neptunus, seen from the top.

2.1.1 Body and design

Neptunus is designed with a foil shaped body, in order to induce low drag forces in the longitudinal direction. The foil design is based on the NACA-foil formula, proposed by the *National Advisory Committee for Aeronautics* (Jacobs et al., 1933), and is stated as

$$\pm y = a_0\sqrt{x} + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4, \tag{2.1}$$

where the cord length is defined along the x-axis. A photo showing Neptunus from the top is provided in Figure 2.2 to confirm the NACA-foil from of the structure.

To model the new design, the program *Inventor*[2], by *AutoDESK* was used. Inventor is a 3D-design modelling program, which provides excellent tools for designing a structure like Neptunus'. In Figure 2.3, the design model for Neptunus is shown.

The designed prototype consists of several blocks, made of acrylonitrile butadiene styrene (ABS) - plastic, 3D printed at NTNU. Moreover, the hull has ventilation holes at the top and bottom to facilitate the water or air flow in or out when moving the ROV into or out from the water. This helps to counteract the pressure differences, and to easier make the ROV neutrally buoyant. Further buoyancy adjustments are made by placing foam and lead into the top and bottom parts of Neptunus, respectively. This setup also increases stability of the ROV. In Table 2.1 some size and weight parameters are provided

[2] Product information at autodesk.com/products/inventor/overview

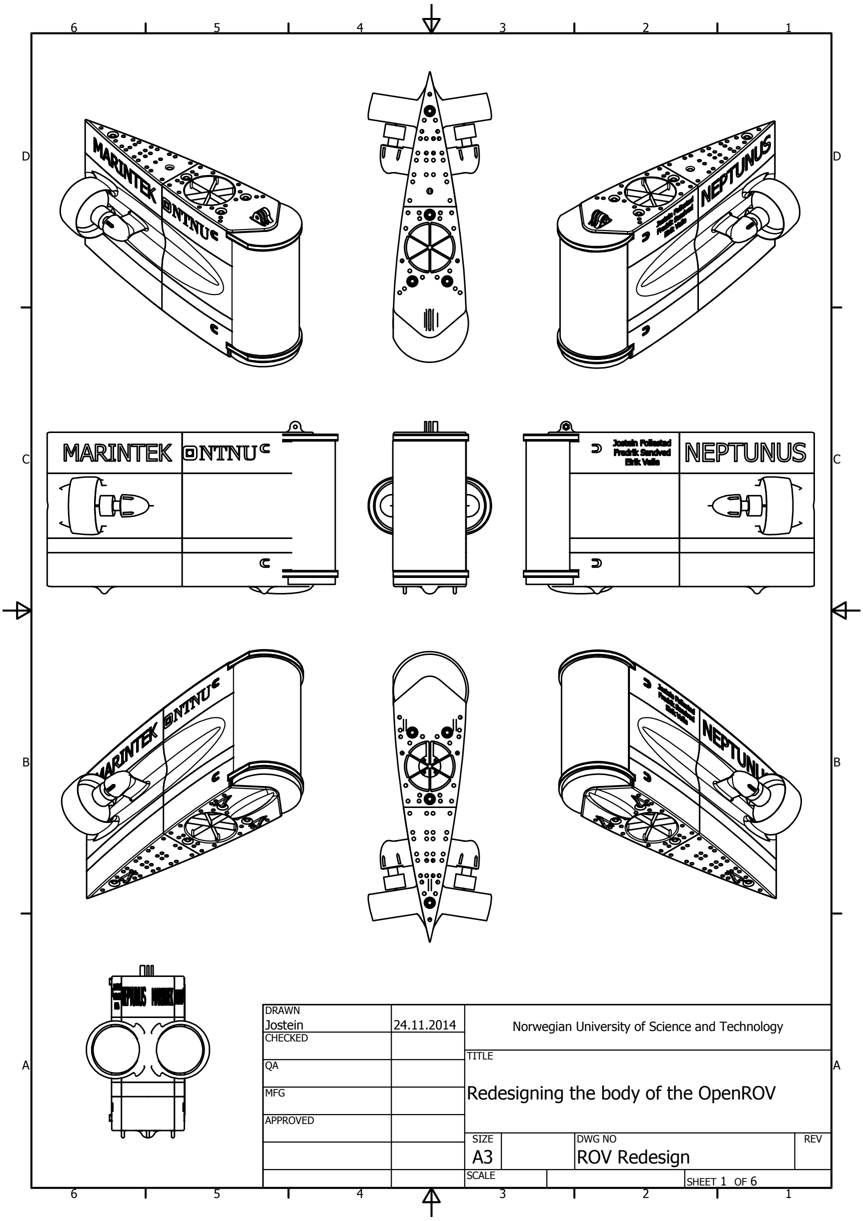


Figure 2.3: Neptunus Inventor drawing. Courtesy of Follestad et al. (2014).

Parameter	Symbol	Value
Weight	m	3.43 kg
Length	L	0.40 m
Width	B	0.10 m
Height	H	0.21 m

Table 2.1: Size and weight parameters.

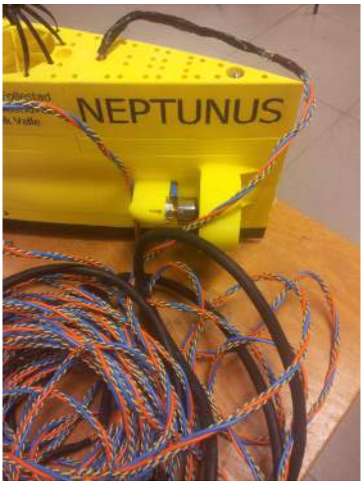


Figure 2.4: Neptunus umbilical.

2.1.2 Sensors and communication

Neptunus is a low-cost ROV, and thus the onboard sensors are quite limited, both in number and in quality. It is equipped with an *inertial measurement unit* (IMU), which measures the roll, pitch and yaw angles of the ROV. Neptunus carries a *high definition* (HD) web-camera, for visual inspection. In addition, there are lights installed. Currently, Neptunus has no positioning system.

The communication between the operator and Neptunus is done via an umbilical. The umbilical, which can be seen in Figure 2.4, transmits power, camera stream, lights and IMU signals, as well as control signals to the propulsion system. The main processes are driven by a *BeagleBone* computer, and an *Arduino* control board. The BeagleBone's main task is to run the *user interface* (UI), and process the video feed, while the Arduino handles the *input/output* (I/O) and power management (Follestad et al., 2014, Ch. 2).

Parameter	**Symbol**	**Value**
Max RPM	RPM_{max}	1964
Min RPM	RPM_{min}	-1964
Propeller blades	Z	3
Propeller diameter	D	0.065 m
Propeller pitch	P	0.034 m
Expanded area ratio	$\frac{A_E}{A_0}$	0.934

Table 2.2: Propulsion system parameters. Information from Graupner (2015a) and Hobbyking (2015).

2.1.3 Propulsion system

There are three thrusters on Neptunus: two in the longitudinal direction, and one in the vertical. The motors are connected through a *electronic speed controller* (ESC) to the control board. In Table 2.2 some motor and the propeller parameters are stated. These parameters will later be used to establish the thrust characteristics for Neptunus.

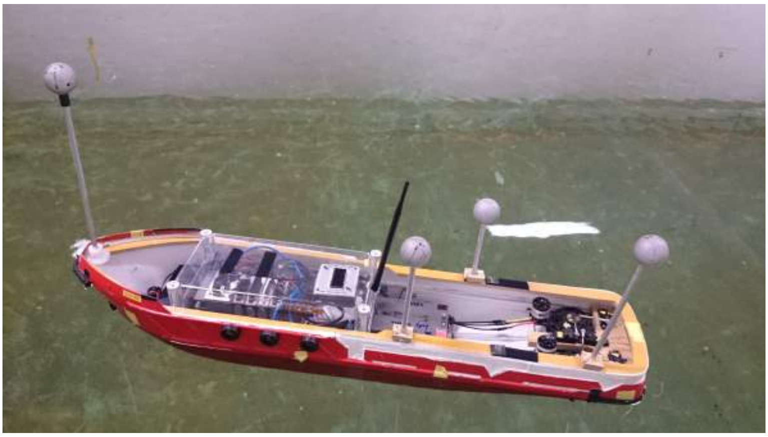


Figure 2.5: Cybership Enterprise I.

Parameter	Symbol	Value
Mass	m	14.790 kg
Length	L	1.105 m
Width	B	0.248 m
Scale	λ	1:50

Table 2.3: Weight and dimension parameters for CSEI.

2.2 Cybership Enterprise I

The surface vessel of concern in this master thesis is the Cybership Enterprise I (CSEI). The CSEI is a model ship located MC-lab at Tyholt. It is used as a test platform for various control systems.

2.2.1 Body and design

The CSEI can be seen in Figure 2.5. Its design is based on an Anchor Handling Tug / Supply Vessel used in offshore operations (Slipway, 2014). The hull is of fiberglass. In Table 2.3, some data regarding CSEI are provided. Extended information regarding the body and design is given by Skåtun (2011).

2.2.2 Sensors and communication

In the MC lab, a positioning system is installed to calculate the position of the surface vessels in the basin, called *Qualisys motion tracker system*.[3] Using the spheres located on the vessel (see Figure 2.5), the system triangulates to obtain the position.

In Figure 2.6 the communication signal flow for CSEI is presented. The control system is located onboard on a *compact RIO* (cRIO), provided by *National Instruments*. The cRIO is a re-configurable embedded control and acquisition system, which handles input and output to the system real time[4]. Moreover, the cRIO sends signals through Ethernet, which is sent over *wireless fidelity* (WiFi) using a router. The I/O from CSEI is handled by *VeriStand* which is a software application for monitoring and controlling real time systems[5]. More information regarding these systems, see Section 5.1.

For manual control of CSEI, a joystick control mode has been implemented. The joystick, a *Play Station* controller, is communicating using Bluetooth. Since the cRIO is not compatible with bluetooth, a *Rasperry Pi* is used to transform the bluetooth signals to Ethernet. Using this setup, a mobile device can communicate with CSEI through VeriStand.

In Figure 2.7, an overview of the control signal flow is given. The figure shows how the signals from the joystick controller, as well as the VeriStand user interface, are transformed to the thrusters. The switch enables for manual control, as well as remote control from VeriStand. The signals from the controller modes are transformed to *pulse-width modulation* (PWM) signals which is delivered to the *field-programmable gate array* (FPGA). The FPGA deals with the I/O to the thrusters.

It is referred to NTNU (2015) for more in-depth information on how the communication structure is set up.

[3] See qualisys.com/products for information

[4] More information can be found at ni.com/compactrio/whatis/

[5] Information at ni.com/veristand/whatis/

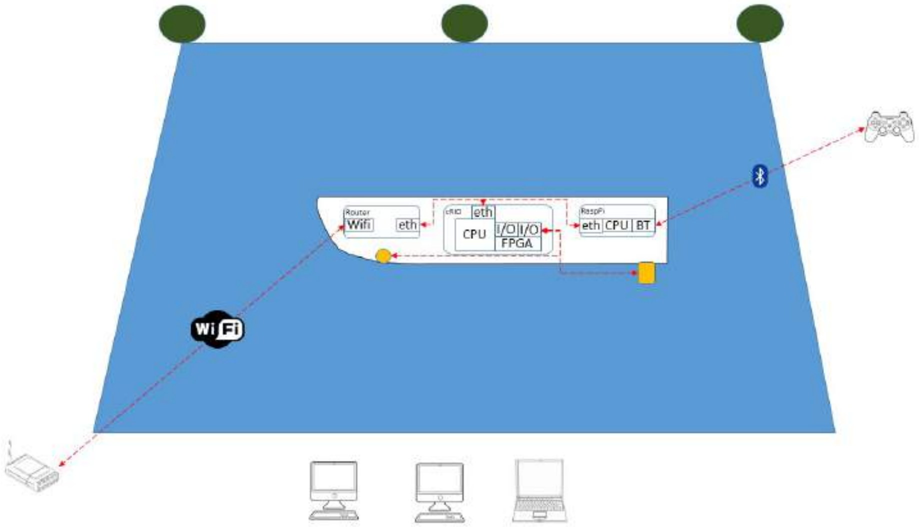


Figure 2.6: The communication signal flow for CSEI in the MC-lab. Courtesy of NTNU (2015).

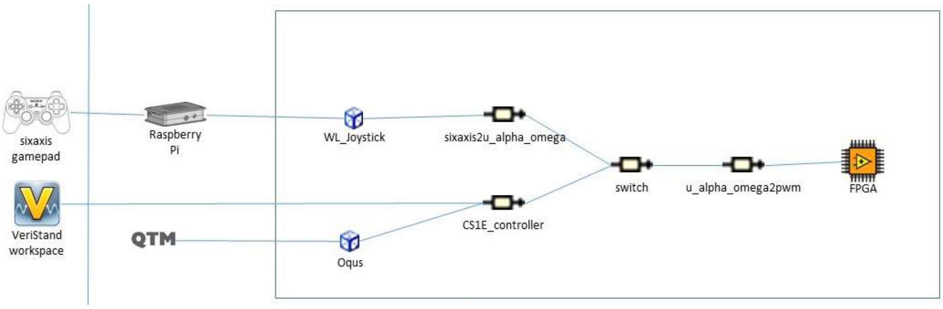


Figure 2.7: Control signal flow for CSEI. Courtesy of NTNU (2015).

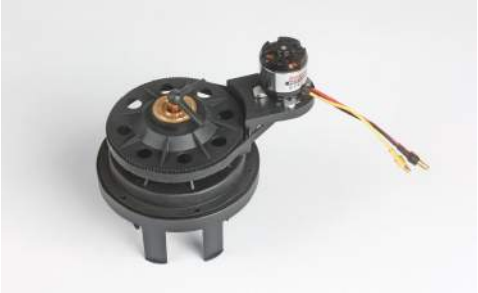


Figure 2.8: The Voith Schneider propeller used on CSEI. Courtesy of Graupner (2015b).

Parameter	**Symbol**	**Value**
x length to VSP	$L_{x,\text{VSP}}$	-0.4575 m
x length to bow thruster	L_{bt}	0.3875 m
y length to VSP	$L_{y,\text{VSP}}$	0.0550 m

Table 2.4: Thruster parameters for CSEI.

2.2.3 Propulsion system

CSEI has a bow thruster, and two *Voith Schneider propellers* (VSPs) at the stern. In Figure 2.8, the VSP used on CSEI is seen. The VSP is consisting of a rotating wheel, with five foils attached to it. In order to produce thrust, the pitch angle on the foils must be changed. This is done by changing the position of a control rod, seen at the top of the figure. Based on the position of the rod, thrust forwards, backwards and sideways can be produced. This implies that even at full rotational speed of the wheel, no thrust will be produced if the foils are standing in neutral position.

A thruster configuration of CSEI is presented in Figure 2.9. Moreover, in Table 2.4, some of the parameters in the thruster overview are identified. These parameters are necessary for the thruster allocation in Section 3.3.1.

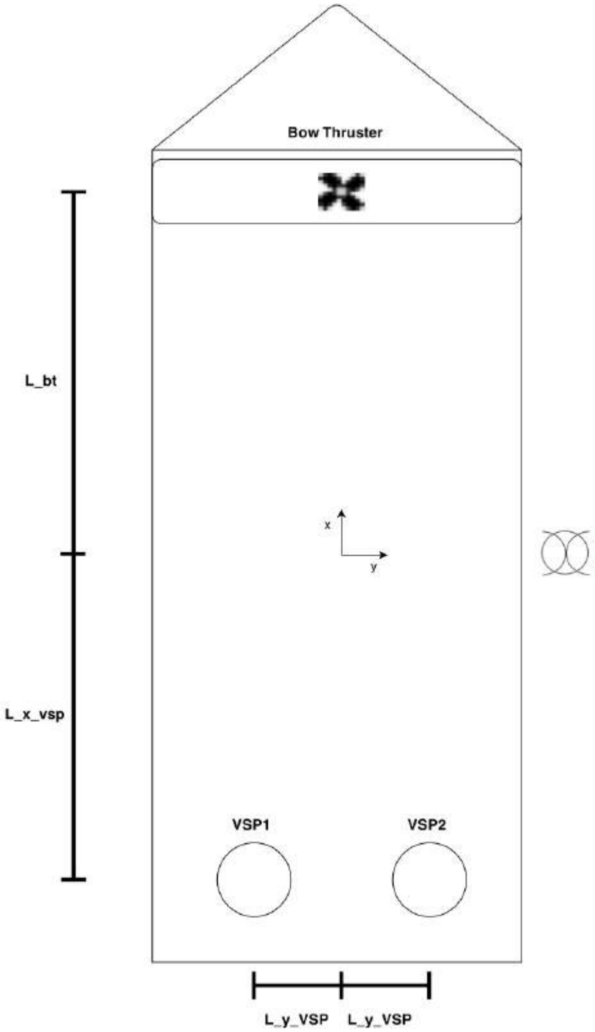


Figure 2.9: Thruster overview of CSEI. Courtesy of Valle (2015).

Chapter 3

Mathematical modeling and system identification

In order to predict and verify the response of a dynamic system, a mathematical model can be established. The model is a mathematical representation of the real system, and can be used to simulate the response and exploit the physical properties in the time domain.

In this chapter, mathematical models for Neptunus and CSEI are developed. The models are based on the *kinematic* and *kinetic* expressions shown in Fossen (2011, Ch. 2 and 6). Both for Neptunus and CSEI a model in six DOF model is established. The models are based on maneuvering theory, which implies frequency independent hydrodynamic forces. A vectorial representation of the equations of motion are used, as by Fossen (1991). Moreover, hydrodynamic forces are described using the theory by Faltinsen (1990). The notation will follow the definitions from the *Society of Naval Architects and Marine Engineers* (SNAME, 1950).

A system identification of the mathematical models is presented. For Neptunus, a 6 DOF system identification is conducted based on towing tests, software analyses, and formulations from the literature. The system identification of CSEI is provided by NTNU (2015). Thus, only a brief presentation of the result will be revealed here.

DOF	Definition	Position [m] & Attitude [rad]	Linear [m/s] & Angular [rad/s] velocities	Forces [N] & Moments [Nm]
1	*surge*	x	u	X
2	*sway*	y	v	Y
3	*heave*	z	w	Z
4	*roll*	ϕ	p	K
5	*pitch*	θ	q	M
6	*yaw*	ψ	r	N

Table 3.1: Motion variables notation from SNAME (1950).

3.1 Mathematical modeling

The overall goal for a mathematical model is to represent the dynamics of a physical system using mathematical set up and notations. The complexity of a mathematical model depends on the comprehensiveness, i.e. how detailed of the properties of the system are described. According to Sørensen (2013, Ch. 7), a mathematical model is divided into two levels of complexity:

- *Process Plant Model* (PPM): a comprehensive description of the actual dynamic system. The PPM includes as much information as possible, with it's purpose of simulation the dynamic system in an accurate way.
- *Control Plant Model* (CPM): a simplified version of the PPM. Here it is normal to include the forces and effects which is most relevant for the application, and the rest is neglected. It is important to mention, that in order to simplify a model, care must be taken. It is essential to know what is neglected, and the effect of it, in relationship with the application.

From a control point of view, a CPM is often considered. In this model, unnecessary effects are neglected. Before the kinematics and kinetics are defined and presented, an overview of the motion variables used in the mathematical model is given.

3.1.1 Motion variables

For a vehicle moving in every DOF, 18 motion variables are needed to describe the dynamics. In Table 3.1, the SNAME notation for describing the vehicles position and velocities are given. Also, the corresponding forces are stated.

Using the vectorial notation from Fossen (2011), the position is written as

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T, \tag{3.1}$$

and the velocity vector as

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T. \tag{3.2}$$

Further, the position and velocities are divided into

$$\boldsymbol{\eta}_1 = \boldsymbol{P} = \begin{bmatrix} x & y & z \end{bmatrix}^T, \quad \boldsymbol{\eta}_2 = \boldsymbol{\Theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T, \tag{3.3}$$

$$\boldsymbol{\nu}_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T, \quad \boldsymbol{\nu}_2 = \boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T. \tag{3.4}$$

The splitting (3.3)-(3.4) is convenient for describing the kinematics and kinetics in a vectorial manner.

3.1.2 Kinematics

According to the Oxford Dictionary (2015a), the kinematics can be defined as

> *The branch of mechanics concerned with the motion of objects without reference to the forces which cause the motion.*

Hence, the motion of the vehicle is the only concern when dealing with the kinematics. A central part of the kinematics is the concept of reference frames. In order to describe the motion of a system, it is common to express it by using different coordinate frames. Normally, an *inertial* coordinate frame is defined as the area for the vehicle to move within. By additionally defining a coordinate system which is fixed with respect to the vehicle, it's relative movement can be expressed.

There exists several coordinate frames which are used to express the motion of a marine vessel. The coordinate frame fixed to the vehicle, called the *BODY* frame, is always present. An *inertial* coordinate frame is defined to describe the relative movement of the BODY frame. The requirement for an inertial coordinate frame is that it non-accelerating, i.e that Newton's laws of motion can be applied.

Fossen (2011, Ch. 2) points out that if the movement of a vehicle is worldwide, such that the earth motion must be taken into consideration, the *Earth-Centered Inertial* (ECI), or *Earth-Centered Earth Fixed* (ECEF) coordinate frame should be used. However, if the movement of a vessel can be fixed to a certain longitude and latitude, the *North-East-Down* (NED) frame can be used instead.

Since Neptunus and CSEI are assumed to operate within a small geographic area, the NED frame will be used as the inertial frame in the simulations.

Relation between NED and BODY frame

The position of the vehicle is given in the NED-frame, while the forces are applied in the BODY-frame. The relationship between these two frames are given by the *transformation* matrices. In this master thesis, the *Euler-angle* transformation is considered. The advantage of using this method is that it is intuitive and relatively easy to understand. A disadvantage is that the Euler-angles will introduce singularities for $\theta = \pm 90°$. A transformation that deals with this problem is the *unit-quaternions*-transformation. Even though this method is more computationally effective, it is less intuitive (Ickes, 1970).

For Neptunus, the scenario of $\theta = \pm 90°$ can not occur due to it's stability characteristics. Moreover it can not occur for CSEI, unless it is capsizing. Consequently it is assumed that the vehicles never will pitch such that singularities occurs, and hence the Euler Angle transformation is used.

According to Fossen (2011, Eq. (2.40)), the kinematic relationship is expressed as:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{3.5}$$

where the transformation matrix $\boldsymbol{J}_\Theta(\boldsymbol{\eta})$ is given by

$$\boldsymbol{J}_\Theta(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\Theta}) & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{T}(\boldsymbol{\Theta}) \end{bmatrix}. \tag{3.6}$$

The translational transformation matrix $\boldsymbol{R}(\boldsymbol{\Theta})$ is:

$$\boldsymbol{R}(\boldsymbol{\Theta}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi} = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \tag{3.7}$$

and the angular transformation matrix $\boldsymbol{T}(\boldsymbol{\Theta})$ is:

$$\boldsymbol{T}(\boldsymbol{\Theta}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix}. \tag{3.8}$$

Here, the notation is abridged as $c(\cdot) = \cos(\cdot)$, $s(\cdot) = \sin(\cdot)$ and $t(\cdot) = \tan(\cdot)$. As mentioned above, the angular transformation matrix is not defined for $\theta = \pm \frac{\pi}{2}$.

3.1.3 Course, sideslip angle, and angle of attack

When a marine vehicle is maneuvering within a reference frame, certain definitions are often used (Breivik and Fossen, 2009):

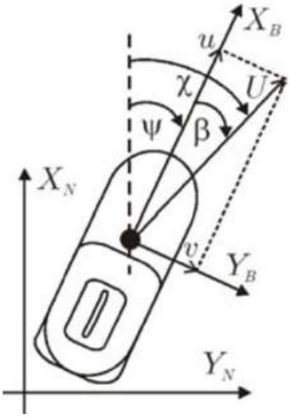


Figure 3.1: Geometrical relationship. Courtesy of Breivik and Fossen (2004).

Course angle χ: The angle from the x-axis of the NED frame to the velocity vector in BODY, positive rotation about the z-axis of the NED frame by the right-hand screw convention.

Side slip angle β: The angle from the x-axis of the BODY frame to the velocity vector of the vehicle, positive rotation about the z-axis of the BODY frame by the right-hand screw convention.

Angle of attack α_{att}: is the angle between the horizontal line in the BODY and the incoming force. Mathematically this can be written as

$$\alpha_{\text{att}} = \tan^{-1}\left(\frac{w}{u}\right). \tag{3.9}$$

From the definitions above, it is follows that

$$\chi = \psi + \beta, \tag{3.10}$$

where

$$\beta = \sin^{-1}\left(\frac{v}{U}\right). \tag{3.11}$$

v is the sway velocity, and U is the speed, i.e $U = \sqrt{u^2 + v^2}$. A geometrical relationship between these definitions is seen in Figure 3.1.

3.1.4 Ocean current

The current will in general vary in magnitude and direction, and is an important effect to take into consideration. According to Fossen (2011, Eq. (8.152)), the ocean current can be represented as a *Gauss-Markov* process

$$\dot{V}_c = -\mu_c V_c + w, \tag{3.12}$$

where V_c is the current velocity and w is Gaussian white noise. μ_c is a constant related to how fast the process converges. A low μ_c results in a pole close to the imaginary axis, and gives slow oscillations with slow convergence. A high μ_c results in a more negative pole, which gives rapid oscillations, and a fast convergence (Balchen et al., 2003, Ch. 4). Moreover, a saturation is introduced, such that

$$V_{\min} \leq V_c(t) \leq V_{\max}. \tag{3.13}$$

Assuming a 2D current with zero attack angle, the current velocities can be expressed in the BODY frame as

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} V_c \cos(\beta_c - \psi) \\ V_c \sin(\beta_c - \psi) \end{bmatrix}. \tag{3.14}$$

The relative velocity $\boldsymbol{\nu}_r$ to the vehicle is defined as

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c = \begin{bmatrix} u_r & v_r & w & p & q & r \end{bmatrix}^T, \tag{3.15}$$

where $\boldsymbol{\nu}_c = [u_c \; v_c \; 0 \; 0 \; 0 \; 0]^T$.

3.1.5 Kinetics

According to the Oxford Dictionary (2015b), the kinetics can be defined as

> *The branch of mechanics concerned with the motion of bodies under the action of forces.*

In contrast of the kinematics, which deals with the motion in a geometrical way, the kinetics considers and analyzes the forces creating this motion. Within the kinetics, both the internal forces such as the mass, damping and restoring forces, as well as the external forces from the environment are considered.

Two types of dynamics can be considered; *seakeeping* or *maneuvering* (Fossen, 2011, Ch. 5 and 6). The seakeeping theory addresses the hydrodynamic forces as frequency depended, i.e. when wave excitation is present. The maneuvering theory on the other hand, assumes frequency independent hydrodynamic forces. This involves the case when the water is calm.

Assuming that Neptunus operates below the sea surface, and that CSEI is operating in calm water, kinetics based on maneuvering theory will be addressed. The main result is to define the kinetics on the form (Fossen, 2011, Eq. (3.1))

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{\text{RB}}, \tag{3.16}$$

where $\boldsymbol{M}_{RB}$ is the rigid body mass matrix, while $\boldsymbol{C}_{RB}(\boldsymbol{\nu})$ is the rigid body Coriolis matrix, and occurs due to the vehicles rotation in the NED reference frame. Further, the rigid body forces $\boldsymbol{\tau}_{\text{RB}}$ can be written as

$$\boldsymbol{\tau}_{\text{RB}} = \boldsymbol{\tau}_{\text{hyd}} + \boldsymbol{\tau}_{\text{hs}} + \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{env}}, \tag{3.17}$$

where $\boldsymbol{\tau}_{\text{hyd}}$ are the hydrodynamic forces, $\boldsymbol{\tau}_{\text{hs}}$ are the hydrostatic forces, $\boldsymbol{\tau}$ are the thruster forces, and $\boldsymbol{\tau}_{\text{env}}$ are the environmental forces.

Rigid body forces

The rigid body forces working on a vehicles is defined according to (3.16), The rigid body equation of motion is established using *Newton-Euler* formulation and *vectorial mechanics.*

It exists several parameterizations of the mass and Coriolis matrices, which is further stated in Fossen (2011, Ch. 3). The parameterizations is valid as long as $\boldsymbol{M}_{RB} = \boldsymbol{M}_{RB}^T$ is symmetric and positive, while $\boldsymbol{C}_{RB} = -\boldsymbol{C}_{RB}^T$ is skew symmetric. A valid parameterization is

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3x3} & -m\boldsymbol{S}(\boldsymbol{r}_g) \\ m\boldsymbol{S}(\boldsymbol{r}_g) & \boldsymbol{I}_g - m\boldsymbol{S}^2(\boldsymbol{r}_g) \end{bmatrix}, \tag{3.18}$$

and

$$\boldsymbol{C}_{RB} = \begin{bmatrix} m\boldsymbol{S}(\boldsymbol{\omega}) & -m\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{S}(\boldsymbol{r}_g) \\ m\boldsymbol{S}(\boldsymbol{r}_g)\boldsymbol{S}(\boldsymbol{\omega}) & -\boldsymbol{S}((\boldsymbol{I}_g - m\boldsymbol{S}^2(\boldsymbol{r}_g))\boldsymbol{\omega}) \end{bmatrix}, \tag{3.19}$$

where m represents the mass of the vehicle, $\boldsymbol{r}_g = [x_g, y_g, z_g]^T$ is the vector from *center of origin* (CO) to *center of gravity* (CG), $\boldsymbol{I}_g$ is the inertia matrix

$$\boldsymbol{I}_g = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix}, \tag{3.20}$$

and $\boldsymbol{S}(\boldsymbol{\omega})$ and $\boldsymbol{S}(\boldsymbol{r}_g)$ are skew-symmetric matrices, defined as

$$\boldsymbol{S}(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_1 & \lambda_3 \\ \lambda_1 & 0 & -\lambda_2 \\ -\lambda_3 & \lambda_2 & 0 \end{bmatrix}, \quad \forall \boldsymbol{\lambda} \in \mathbb{R}^3. \tag{3.21}$$

Considering symmetry about the different axes reduces the number of cross terms in the rigid body equations (Faltinsen, 1990, Ch. 3).

Hydrodynamic forces

The hydrodynamic forces are the forces which arises due to that the vehicle is operating in a fluid environment. According to Fossen (2011, Sec. 6.5), the hydrodynamic forces can be expressed as a *hydrodynamic mass-spring damper* system

$$\boldsymbol{\tau}_{\text{hyd}} = -\boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r - \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r, \tag{3.22}$$

where $\boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r$ are the forces due to the acceleration of the fluid surrounding the vessel when moving. As for the rigid body forces, $\boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$ is the added mass Coriolis forces. $\boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$ represents the damping forces, and consists of linear and non-linear terms.

The added mass matrix $\boldsymbol{M}_A$ can generally be written as a full six-times-six matrix (Faltinsen, 1990, Eq. (3.10))

$$\boldsymbol{M}_A(k,j) = A_{\text{kj}}, \qquad k = 1,..,6, \;\; j = 1,..,6. \tag{3.23}$$

Introducing the notation from SNAME (1950), the added mass matrix can be expressed as

$$\boldsymbol{M}_A = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} = -\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \quad \boldsymbol{M}_A = \boldsymbol{M}_A^T. \tag{3.24}$$

Moreover, a parameterization of the added mass Coriolis matrix is given as (Fossen, 2011, Eq. (6.43))

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} \boldsymbol{0}_{3x3} & -\boldsymbol{S}(\boldsymbol{A}_{11}\boldsymbol{\nu}_{r1} + \boldsymbol{A}_{12}\boldsymbol{\omega}) \\ -\boldsymbol{S}(\boldsymbol{A}_{11}\boldsymbol{\nu}_{r1} + \boldsymbol{A}_{12}\boldsymbol{\omega}) & -\boldsymbol{S}(\boldsymbol{A}_{21}\boldsymbol{\nu}_{r1} + \boldsymbol{A}_{22}\boldsymbol{\omega}) \end{bmatrix} = -\boldsymbol{C}_A(\boldsymbol{\nu}_r)^T. \tag{3.25}$$

Finally, the damping forces can be written as the sum of the linear and quadratic contributions

$$\boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{D}_L\boldsymbol{\nu}_r + \boldsymbol{D}_q(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r, \qquad \boldsymbol{D}(\boldsymbol{\nu}_r) > 0. \tag{3.26}$$

Again by using the SNAME notation, the linear damping matrix is written as

$$\boldsymbol{D}_L = -\begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ K_u & K_v & K_w & K_p & K_q & K_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix}, \tag{3.27}$$

while the non-linear quadratic damping matrix generally is expressed as

$$\boldsymbol{D}_q(\boldsymbol{\nu}_r) = -\begin{bmatrix} X_{|u|u}|u| & X_{|u|v}|u| & X_{|u|w}|u| & X_{|u|p}|u| & X_{|u|q}|u| & X_{|u|r}|u| \\ Y_{|v|u}|v| & Y_{|v|v}|v| & Y_{|v|w}|v| & Y_{|v|p}|v| & Y_{|v|q}|v| & Y_{|v|r}|v| \\ Z_{|w|u}|w| & Z_{|w|v}|w| & Z_{|w|w}|w| & Z_{|w|p}|w| & Z_{|w|q}|w| & Z_{|w|r}|w| \\ K_{|p|u}|p| & K_{|p|v}|p| & K_{|p|w}|p| & K_{|p|p}|p| & K_{|p|q}|p| & K_{|p|r}|p| \\ M_{|q|u}|q| & M_{|q|v}|q| & M_{|q|w}|q| & M_{|q|p}|q| & M_{|q|q}|q| & M_{|q|r}|q| \\ N_{|r|u}|r| & N_{|r|v}|r| & N_{|r|w}|r| & N_{|r|p}|r| & N_{|r|q}|r| & N_{|r|r}|r| \end{bmatrix}. \tag{3.28}$$

It should be noted that other higher order damping terms can be included in the mathematical model. This included third order terms or higher. However, these terms are often of low contribution when considering low velocities.

Hydrostatic forces

According to Faltinsen (1990, p. 58), a body will create restoring forces when it floats freely on the sea surface, or completely submerged. As for a typical spring in a mass-spring-damper system, the restoring forces depend on the mass and stiffness. For a completely submerged vehicle, such as Neptunus, the stiffness will depend on the location of the *center of buoyancy* (CB). On the other side, for a surface vessel, such as CSEI, the stiffness will depend on the *metacentric* height and the water plane area.

Utilizing Archimedes' law (the buoyancy force B is equal to the submerged weight W), the restoring forces can be established. Rotating these forces from NED to BODY frame in the translational and angular DOFs, the restoring forces for a underwater vehicle can be stated as (Fossen, 2011, Eq. (4.6))

$$\boldsymbol{\tau}_{\text{hs}} = -\boldsymbol{g}_{\text{uv}}(\boldsymbol{\eta}) = -\begin{bmatrix} (W-B)\sin(\theta) \\ -(W-B)\cos(\theta)\sin(\phi) \\ -(W-B)\cos(\theta)\cos(\phi) \\ -(y_gW-y_bB)\cos(\theta)\cos(\phi)+(z_gW-z_bB)\cos(\theta)\sin(\phi) \\ (z_gW-z_bB)\sin(\theta)+(x_gW-x_bB)\cos(\theta)\cos(\phi) \\ -(x_gW-x_bB)\cos(\theta)\sin(\phi)-(y_gW-y_bB)\sin(\theta) \end{bmatrix}, \tag{3.29}$$

where $W = mg$ is the submerged weight, $B = \rho g \nabla$ is the buoyancy and $\boldsymbol{r}_b = [x_b, y_b, z_b]^T$ is the distance from CO to CB.

Moreover, the restoring forces for a surface vessel is investigated. From Faltinsen (1990, p. 58), it is argued that for a freely floating body, the restoring coefficient will only be non-zero in C_{33}, C_{44}, $C_{35} = C_{53}$, and C_{55} when considering a body with x-z symmetry. The reason is that no restoring force will act in surge, sway, or yaw direction without the presence of mooring lines or other constraints. It is assumed that $C_{35} = C_{53}$ will be of low contribution, and hence neglected. In addition, it is assumed that the vessel will the heave, roll and pitch motion are small. Hence, a linear approximation, in accordance to Faltinsen (1990, Pg. 58), can be used to express the restoring forces

$$C_{33} = \rho g A_{\text{wp}}, \quad C_{44} = \rho g V \overline{GM}_T, \quad C_{55} = \rho g V \overline{GM}_L, \tag{3.30}$$

where $\overline{GM}_T$ and $\overline{GM}_L$ is the transverse, and lateral metacentric height, respectively. With these restoring coefficients, the restoring forces for a surface vessel can be written as

$$\boldsymbol{\tau}_{hs} = -\boldsymbol{g}_{sv}(\boldsymbol{\eta}) = -\begin{bmatrix} 0 \\ 0 \\ \rho g A_{wp} z \\ \rho g V \overline{GM}_T \phi \\ \rho g V \overline{GM}_L \theta \end{bmatrix}. \tag{3.31}$$

Thruster forces

The propulsion system exercises the main driving force, that make the vehicle move to a desired location or along a predefined path. It is essential to transform the forces that each thruster exercises, to the forces executed in the different DOFs, and vice versa. This is often referred to as the *thruster configuration* (Sørensen, 2013). The control forces can be defined as

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{u}, \tag{3.32}$$

where $\boldsymbol{u} = [u_1, u_2, ..u_n]^T$ is the control inputs, and $\boldsymbol{K} = diag[K_{11}, K_{22}, ..K_{nn}]^T$ is a thrust coefficient matrix. In order to transform the thruster forces to forces and moments in the correct DOF, the *thrust configuration* matrix $\boldsymbol{T}(\boldsymbol{\alpha})$ is introduced, where $\boldsymbol{\alpha}$ is the optional angle of the thrusters location, compared to the BODY frame x-axis. The actuator forces can be described as

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u}. \tag{3.33}$$

Furthermore, the commanded thrust to one of the propellers can according to Carlton (2012, Eq. (6.1)) be described as

$$u = K_T \rho D^4 n|n|, \tag{3.34}$$

where K_T is the thrust coefficient, D is the propeller diameter and n is the revolution per second (rps). K_T is related to the efficiency of the propeller, and accounts for the thrust losses. Moreover K_T is found by *open-water tests*, and is usually performed in a cavitation tunnel or towing tank (Sørensen, 2013, Sec. 9.3).

However, Oosterveld and Van Oossanen (1975) have investigated parameters influencing the thrust efficiency, and found that K_T can be written as a polynomial of the following parameters

$$K_T = f\left(J_a, \frac{P}{D}, \frac{A_E}{A_0}, Z, R_N, \frac{t}{c}\right). \tag{3.35}$$

Here, J_a is the advance ratio, $\frac{P}{D}$ is the pitch ratio, $\frac{A_E}{A_0}$ is the blade area ratio, Z is the number of propeller blades. R_N is the Reynolds number, while $\frac{t}{c}$ expresses the ratio between the thickness and chord length of the blade section. Finally, the advance ratio J_a and the open water efficiency η_0 are defined in Oosterveld and Van Oossanen (1975, Eq. (6) and (7)) as

$$J_a = \frac{V_a}{nD}, \text{ and } \eta_0 = \frac{J_a K_T}{2\pi K_Q}. \tag{3.36}$$

Open water tests can be utilized to obtain values for K_T. In (3.36) V_a is the inflow velocity to the propeller, and K_Q is the torque coefficient, also given as a polynomial function of the same parameters as in (3.35). In Figure 3.2, a diagram showing how the commanded thrust $\boldsymbol{\tau}$ are transformed to a corresponding speed n.

Environmental forces

Based on the environment, the vehicle of consideration is working within, different forces can affect the behavior. Commonly, the environmental forces are divided into: ocean current, described in Section 3.1.4, wave forces, and lastly the effect from wind.

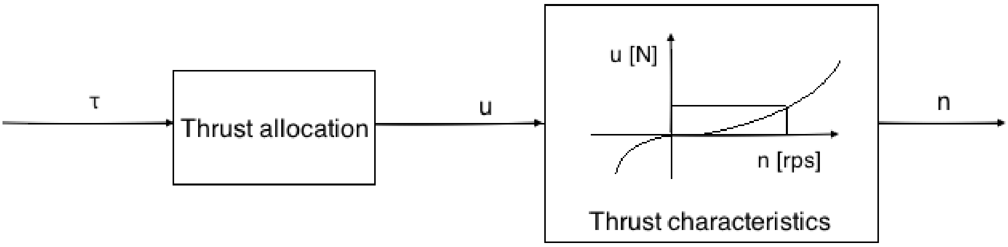


Figure 3.2: The corresponding speed n as a function of the commanded thrust τ.

In certain areas and weather conditions, forces due to wind and waves may be of significant magnitude, and must therefore be taken into consideration. However, in this master thesis, the effect of wave and wind is not considered. The main reason is that it is assumed that Neptunus is working beneath the sea surface, and thus the wave force will be of low contribution. For the same reason, no wind force will contribute. For the surface vessel CSEI, it is assumed that the working area is in the test basin, where no waves or wind are present. Thus the environmental forces from waves and wind are

$$\boldsymbol{\tau}_{\text{env}} \approx 0. \tag{3.37}$$

Umbilical forces

For Neptunus to communicate with the operator, information is transmitted through an umbilical, which send information regarding the thrusters, camera and depth sensors. The umbilical contributes to the dynamic system by generating drag force.

In order to model the drag forces on the umbilical, *Morison's equation* can be taken into consideration Faltinsen (1990, Eq. 3.34). The drag-dominated term of Morison's equation is used to evaluate the drag forces from the environment on a cylinder strip with infinitesimal length ds

$$dF_{ii} = \frac{1}{2}\rho C_{Di} D v_i |v_i| ds. \tag{3.38}$$

The drag moments are determined as

$$dM_{ii} = \frac{1}{2}\rho C_{Di} D v_i |v_i| s ds. \tag{3.39}$$

v_i $(i = 1, 2, .. 6)$ is the velocity in the desired direction, C_D is the *drag coefficient*, and is commonly determined based on experiments. s is the length of the umbilical segment.

Current assumption

Using the definition of the rigid body Coriolis matrix, together with the assumption of a *constant* and *irrotational* current, the following property is valid (Hegrenæs and Hallingstad, 2011, Prop. 2).

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r, \tag{3.40}$$

where the relative velocity $\boldsymbol{\nu}_r$ is defined as in Section 3.1.4.

Resulting mathematical model

With the obtained expressions for the kinematics and kinetics, a resulting mathematical model can be set up.

Mathematical model for an underwater vehicle:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu}_r \tag{3.41}$$

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}_{uv}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{um}. \tag{3.42}$$

Mathematical model for a surface vessel:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu}_r \tag{3.43}$$

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}_{sv}(\boldsymbol{\eta}) = \boldsymbol{\tau}. \tag{3.44}$$

3.2 System identification for Neptunus

In this section, the system identification for Neptunus and Cybership Enterprise I will be established. For Neptunus, software analysis and towing tests have been conducted in order to obtain the added mass, damping and restoring forces. In addition, some relations from the literature have been utilized.

The system identification is more extensive than the one performed for Neptunus by Follestad et al. (2014, Ch. 5), where parameters were found using estimation methods. Before stating the results related to the system identification, a brief overview of the software analysis and towing tests are presented.

3.2.1 Software analysis in HydroD

To obtain more satisfying and accurate parameters related to the added mass, the DNV program *HydroD* is used. HydroD is a computer application for computation of hydrostatics, wave loads and motion response for marine vessels (Det Norske Veritas, 2014). It should be mentioned that even though HydroD is designed for analyzing the response of surface floating vessel, it is usable for underwater vehicles as well.

Another software application is GeniE (2015), provided by DNV, which makes a panel model that can be analyzed with HydroD. A drawback with GeniE is that it needs a finite-element model as input, which does not exist for Neptunus. Consequently, GeniE is not further used in the analysis.

To perform simulations in HydroD, a meshed model is needed. Consequently, some adjustments must be applied to the Inventor model from Follestad et al. (2014, Ch. 3). The steps from the Inventor model to a suitable meshed model originate from the work done by Hval (2012) and Berg (2012).

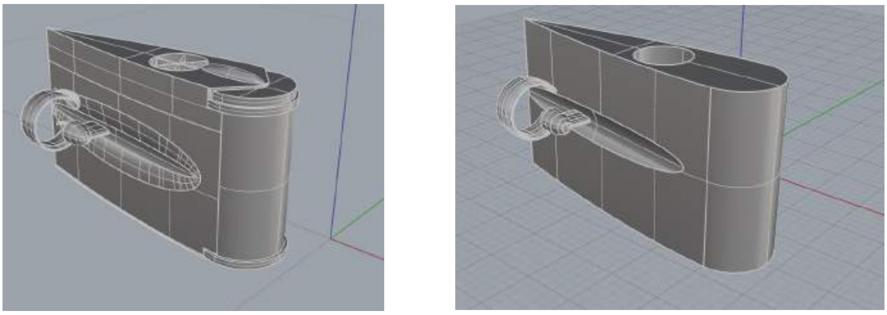
Meshing in Rhinoceros

To mesh the Neptunus model, the design and analyzing program *Rhinoceros*[1] is used. Due to the amount of circular and cylindrical parts, Rhinoceros' mesh tools are not able to create a closed mesh on the original design model from Follestad et al. As a result, the meshed model of Neptunus could not be closed, and calculations on the ROV's performance and behavior were meaningless. To obtain one meshed solid, simplifications in the original, imported Inventor model are performed. This involves replacing cylindrical parts with cubic parts, as well as reducing the number of parts. To not affect the

[1] See rhino3d.com/features for product information

appearance such that the behavior changes in significant manner, the simplification must be done with carefully.

The original and simplified model can be seen in Figure 3.5a and 3.5b, respectively. Some features on the original model have been removed, such as the parts around the controller board container, the GoPro attachment point at the top, and the gratings for the vertical thruster.



(a) The original model of Neptunus.

(b) The reduced model of Neptunus.

Figure 3.3: The original and reduced model of Neptunus. Note the reduction of plates and features on the simplified model.

For simplicity in further analysis in HydroD, the coordinate system for Neptunus is changed, such that the center of origin is placed centered in the model. This will reduce the vector from CO to CG, and thus reduce the cross coupling terms in the inertia mass matrix.

The obtained model can further be meshed, by using Rhinoceros' mesh tools. The simplified model of Neptunus, imported from Inventor, is a *Non-Uniform Rational B-spline* (NURBS) object, which is the common model output from a graphical design program. Since this model is quite simple, a coarse mesh is applied. This will also reduce the simulation time in HydroD. The meshed model used in HydroD for simulations is presented in Figure 3.4, and consists of triangular elements. Notice also the mesh density area around the horizontal thrusters. On the original model, this was the area which Rhinoceros was not able to mesh properly, due to the amount of cylindrical elements. Using triangle elements has been proved to properly cover the area of consideration, and is the most efficient and optimal shape to use (D'Azevedo, 1991).

To use the meshed model in HydroD, it is exported as a WAMIT-file (*.GDF*). Neptunus is imported as a panel model with 1573 nodes and 2277 plates.

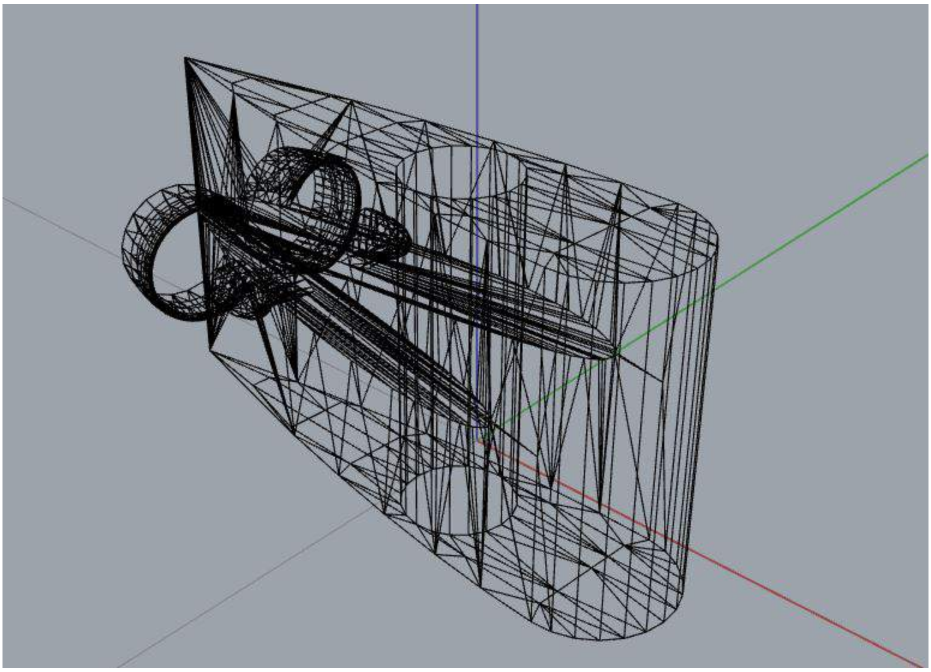
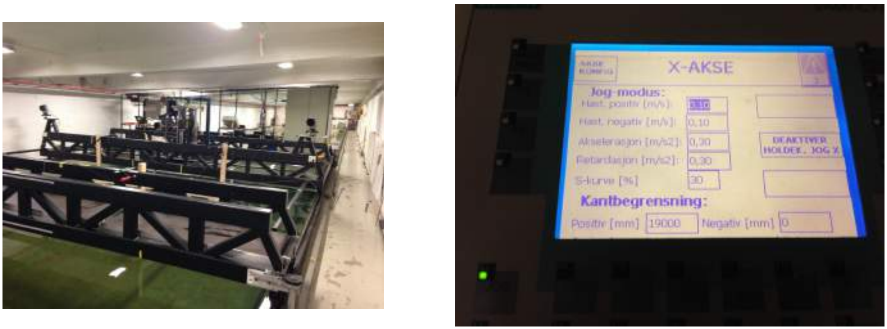


Figure 3.4: The meshed model of Neptunus.

Set up in HydroD

With a panel model, HydroD can calculate the hydrostatic and hydrodynamic forces, using potential theory. Consequently, the viscous damping forces can not be established using this approach. With a panel model, potential theory will only consider the potential damping, and since the ROV is located beneath the sea surface, this damping is of low contribution. To execute the analysis in HydroD, a wizard of 17 steps is followed, see Appendix A.2.

Two runs in HydroD has been conducted. Since it is mainly a software program for surface vessels, a too high radius of gyration in roll and pitch is set when the vehicle is under water. As a solution, one run where the ROV is placed straight below the surface, and one run where the ROV is placed on 300 m depth. The roll and pitch parameters is found using the first run, while the rest is found with the second run. Upon completing the simulation, a result file is generated. In this file, the vector from CO to CB ($\boldsymbol{r}_g$) and



X-AKSE

Jog-modus:

Hast. positiv [m/s]: 0,10

Hast. negativ [m/s]: 0,10

Akselerasjon [m/s2]: 0,30

Retardasjon [m/s2]: 0,30

S-kurve [%] 30

Kantbegrensning:

Positiv [mm] 19000 Negativ [mm] 0

(a) Towing wagon.

(b) Control board.

Figure 3.5: Towing tests in the MC-lab.

added mass matrix $\boldsymbol{M}_{A,\text{HydroD}}$ can be found.

$$\boldsymbol{M}_{A,\text{HydroD}} = \begin{bmatrix} 0.4187 & 0.1093 & 0.0057 & 0.0381 & -0.1400 & -0.0310 \\ 0.1166 & 2.9746 & -0.0326 & 1.0327 & -0.0474 & -0.4732 \\ -0.0016 & -0.0340 & 0.4745 & -0.0012 & 0.0160 & 0.0059 \\ 0.0420 & 1.0271 & 0.0002 & 0.3271 & -0.0126 & -0.1632 \\ -0.1432 & -0.0436 & 0.0273 & -0.0114 & 0.0942 & 0.0131 \\ -0.0226 & -0.4358 & 0.0035 & -0.1504 & 0.0091 & 0.1907 \end{bmatrix}, \quad (3.45)$$

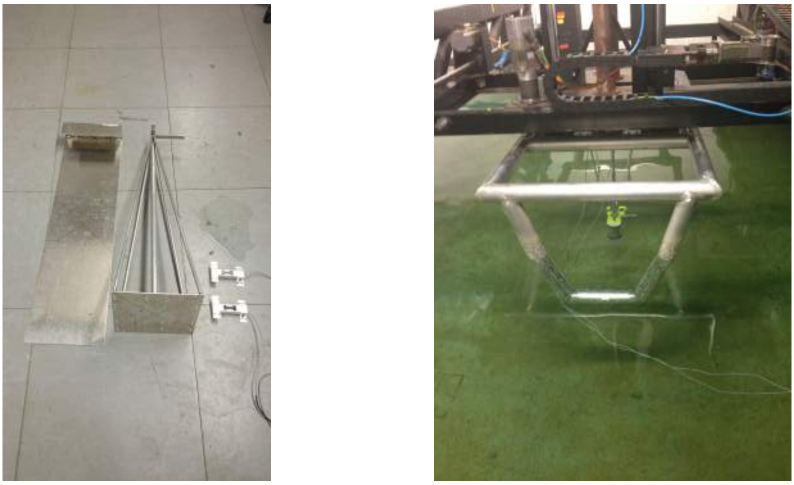
and

$$\boldsymbol{r}_b = \begin{bmatrix} 0 & 0 & 0.0755 \end{bmatrix}^T. \quad (3.46)$$

The resulting added mass matrix is non-dimensional, and must be multiplied with certain parameters related to density, characteristic length and displaced volume in order to obtain correct results. This is calculated in Section 3.2.3.

3.2.2 Towing tests

The towing tests were performed in the MC-lab at Tyholt during week 10, 2015. Some of this work can be found in Eidsvik (2015), with a different point of view. Due to the small size of Neptunus, the full scale model can be used in the towing tests. Thus, no scaling similarities (in terms of *geometric*, *kinematic*, or *dynamic*) have to be considered (Steen, 2011, Sec. 1.7). The towing tests focuses on the diagonal damping terms. According to Fossen (2011, Ex. 6.2), the diagonal damping terms are dominating.



(a) Bracket and force meters. Courtesy of Eidsvik (2015).

(b) Neptunus fastened to the wagon for towing in sway.

Figure 3.6: Towing test set up of Neptunus.

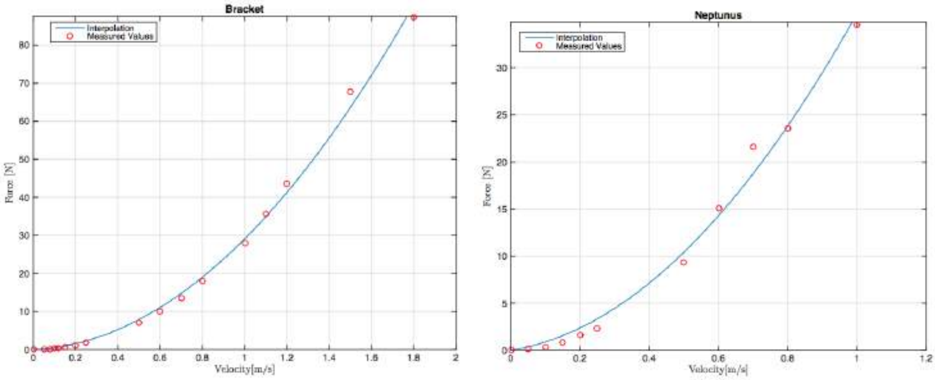
Set up in MC lab

The towing wagon (see Figure 3.5) is used to induce forces on the ROV in the different DOF. The ROV was connected to a bracket, which consists of a metal plate at the top, which is fastened to the wagon, some vertical rods, and a fastening point for the ROV at the bottom. Two measurement devices for measuring the force are fastened to the metal plate at the top.

In Figure 3.6a, the bracket used for towing is showed. Based on the DOF of interest, the ROV is fastened accordingly. In Figure 3.6b, Neptunus is fastened for towing in sway. The measured forces and moments, in addition to other parameters, such as the time, position and velocities, are saved to *.asc*-files. For measuring the forces in the translational DOFs (surge, sway, and heave), the wagon is moved forwards and backwards in the basin. For the angular DOF (roll, pitch, and yaw), the frame where the ROV is attached (see Figure 3.6) is rotating around it's axis.

Execution

The damping forces are typically divided into linear and quadratic terms. The linear term is dominating while the velocities are low, while the quadratic term is becomes more dominating with higher induced velocities. Thus, to characterize these effects from towing, both low and higher velocities have been considered. During these tests,



(a) Bracket.

(b) Neptunus and bracket.

Figure 3.7: Drag curve in surge.

constant velocity in the governing DOF has been achieved, so that the damping forces can be found.

A constant acceleration has been used for the tests related to linear and quadratic damping. For translational directions, the acceleration is set to 0.30 $\mathrm{m/s^2}$. For angular directions, the acceleration is set to 100 $\mathrm{deg/s^2}$. A series of run were performed with increasing velocity on the wagon. The translational and angular velocities of the ROV used in the tests are provided in Table A.1 and A.2.

The raw measurements contains a significant amount of noise, so Buttherworth filter with cut off frequency $\omega_c = 10$ Hz was used.

Results and data processing

The data results from the towing tests are used to generate drag curves in six DOF. A MATLAB-script by Eidsvik (2015) is used to select satisfying force samples for each velocity. Subsequently, a mean force for these samples is found. Linear and quadratic regression techniques are applied to create an interpolated drag curve in the different DOFs. To obtain the drag forces for Neptunus, the contributions from the bracket must be subtracted.

As an example, the drag curves in surge for Neptunus attached to the bracket, and the bracket itself are seen in Figure 3.7. The interpolation function for Neptunus and bracket is

$$F_{\text{Nep+bracket}}(u) = 6.294u + 29.076u^2, \tag{3.47}$$

Linear damping	Quadratic damping
X_u = - 2.291	$X_{u\|u\|}$ = - 4.008
Y_v = - 4.980	$Y_{v\|v\|}$ = - 35.216
Z_w = - 15.190	$Z_{w\|w\|}$ = - 10.304
K_p = - 0.009	$K_{p\|p\|}$ = - 0.191
M_q = 0.050	$M_{q\|q\|}$ = - 0.481
N_r = - 0.261	$N_{r\|r\|}$ = - 0.320

Table 3.2: The linear and quadratic damping terms for Neptunus.

while the function for the bracket is

$$F_{\text{bracket}}(u) = 4.003u + 25.068u^2. \tag{3.48}$$

Thus, by simple subtraction, the drag curve in surge is found by

$$F_{\text{Nep}}(u) = 2.291u + 4.008u^2. \tag{3.49}$$

This procedure is conducted in all six DOF. The results for the drag forces are stated in Table 3.2. The drag curves in all 6 DOF are presented in Appendix A.5.

3.2.3 Resulting system identification

Center of gravity and center of buoyancy

As a design property, CG is set to coincide with the CO for Neptunus. CB is taken directly from HydroD. Thus, $\boldsymbol{r}_g$ and $\boldsymbol{r}_b$ are

$$r_g = \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad r_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0755 \end{bmatrix}. \tag{3.50}$$

Rigid body mass

The design model of Neptunus can be used to investigate the properties related the rigid body mass matrix. The model is exported as an *.stp* file, and imported into *Solidworks*[2],

[2] See solidworks.com/ for product information.

another design program. Evaluation of the rigid body inertia matrix in Solidworks gives

$$\boldsymbol{I}_g = \begin{bmatrix} 0.01540253700 & 0.00000017800 & -0.00020733700 \\ 0.00000017800 & 0.05070300000 & 0.00000006967 \\ -0.00020733700 & 0.00000006967 & 0.03989500000 \end{bmatrix}. \tag{3.51}$$

It is observed that the cross-coupled moment of inertia terms give very low contribution. Moreover, using (3.18), the rigid body mass matrix is evaluated to be

$$\boldsymbol{M}_{RB} = \begin{bmatrix} 3.4600 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 3.4600 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 3.4600 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0154 & 0.0000 & -0.0002 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0507 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.0002 & 0.0000 & 0.0399 \end{bmatrix}. \tag{3.52}$$

By setting $\boldsymbol{r}_g$ to zero, the off-diagonal terms become zero. Some information regarding the rigid body dynamics will be lost using this assumption. However, these radius of gyration generated effects are assumed to be small, and will not affect the vehicle's dynamics.

Rigid body Coriolis

By using the rigid body mass matrix in (3.52) together with the rigid body Coriolis matrix parameterization in (3.19), $\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)$ can be calculated.

Added mass

The added mass matrix from HydroD is given in (3.45). It is non-dimensional, and must be multiplied with the density ρ and the volume V to reveal the correct value, with respect to dimension. In addition,

- the roll, pitch, yaw component in surge, sway, and heave direction are multiplied with the characteristic length, i.e $L = 0.4$ m.
- the surge, sway, and heave component in roll, pitch, and yaw direction are to be multiplied with the characteristic length.
- the roll, pitch, and yaw component in roll, pitch, and yaw direction are to be multiplied with the characteristic length squared, i.e L^2.

The density of salt water is $\rho = 1025\ kg/m^3$, and the submerged volume of Neptunus is $V = 0.004789\ m^3$. Thus, the added mass matrix from HydroD is

$$\boldsymbol{M}_{A,\text{nonsym}} = \begin{bmatrix} 2.0555 & 0.5364 & 0.0282 & 0.0748 & -0.2749 & -0.0609 \\ 0.5722 & 14.6033 & -0.1599 & 2.0280 & -0.0931 & -0.9293 \\ -0.0076 & -0.1670 & 2.3295 & -0.0024 & 0.0314 & 0.0117 \\ 0.0824 & 2.0170 & 0.0003 & 0.2569 & -0.0099 & -0.1282 \\ -0.2811 & -0.0855 & 0.0535 & -0.0090 & 0.0740 & 0.0103 \\ -0.0444 & -0.8559 & 0.0069 & -0.1181 & 0.0071 & 0.1498 \end{bmatrix}. \tag{3.53}$$

However, as stated in Section 3.1.5, a property of the added mass matrix is $\boldsymbol{M}_A = \boldsymbol{M}_A^T \geq 0$. It can be noticed that (3.53) does not fulfill this property. Consequently, some symmetry considerations must be applied. The approach used in this thesis is to create symmetry by taking the mean of the matrix values of concern, as expressed in Figure A.2.. The resulting added mass matrix is

$$\boldsymbol{M}_A = \begin{bmatrix} 2.0555 & 0.5543 & 0.0103 & 0.0786 & -0.2780 & -0.0527 \\ 0.5543 & 14.6033 & -0.1634 & 2.0225 & -0.0893 & -0.8926 \\ 0.0103 & -0.1634 & 2.3295 & -0.0011 & 0.0425 & 0.0093 \\ 0.0786 & 2.0225 & -0.0011 & 0.2569 & -0.0094 & -0.1232 \\ -0.2780 & -0.0893 & 0.0425 & -0.0094 & 0.0740 & 0.0087 \\ -0.0527 & -0.8926 & 0.0093 & -0.1232 & 0.0087 & 0.1498 \end{bmatrix}. \tag{3.54}$$

Added mass Coriolis

The added mass Coriolis matrix is found (as for the rigid body forces) by using the added mass matrix from (3.54) with the parameterization given in (3.25). The resulting matrix follows directly.

Linear and quadratic damping

Using the processed results from Table 3.2, matrices consisting of linear and quadratic damping can be established. The linear damping matrix is defined to be

$$\boldsymbol{D}_L = \begin{bmatrix} 2.2907 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.9804 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15.1897 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0091 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0501 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2605 \end{bmatrix}, \tag{3.55}$$

while the quadratic damping matrix is

$$\boldsymbol{D}_q(\boldsymbol{\nu}_r) = \begin{bmatrix} 4.008|u_r| & 0 & 0 & 0 & 0 & 0 \\ 0 & 35.216|v_r| & 0 & 0 & 0 & 0 \\ 0 & 0 & 10.304|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.191|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.481|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.320|r| \end{bmatrix}. \quad (3.56)$$

Once again it is noticed that the conducted towing tests only reveals the diagonal damping terms. This is obviously not entirely correct, since Neptunus may most certain have cross-coupled damping terms. However, the diagonal terms are most often dominating compared to the cross-coupled terms, and may be sufficient to use in many engineering approaches.

Restoring vector

Follestad et al. (2014, Ch. 5 and 6) argues that Neptunus is slightly positively buoyant. Due to the ventilation holes, the buoyancy can be adjusted. The displaced volume from the HydroD calculations is $V_d = 0.004789\ m^3$, which is not correct since it assumes that the Neptunus body is fully enclosed. Accounting for the ventilation holes, the displaced volume is

$$\nabla = 0.003378\ \ m^3. \quad (3.57)$$

Hence, the buoyancy force is calculated to be

$$B = \rho g \nabla = 33.966\ \ N. \quad (3.58)$$

With $W = \rho g = 33.943$ N, the expression in (3.29) is utilized to find the restoring forces. Consequently, the restoring force is determined to be

$$\boldsymbol{g}_{uv}(\boldsymbol{\eta}) = \begin{bmatrix} -0.0234\sin(\theta) \\ 0.0234\cos(\theta)\sin(\phi) \\ 0.0234\cos(\theta)\cos(\phi) \\ -2.5644\cos(\theta)\sin(\phi) \\ 2.5644\sin(\theta) \\ 0.0000 \end{bmatrix}. \quad (3.59)$$

Thrust characteristics

In order to determine the thruster forces empirically, the thrust coefficient K_T must be found, thus, (3.34) can be utilized. The determination of this thrust coefficient can be done using the Wageningen B-series propeller diagrams (Van Lammeren et al., 1969).

For Neptunus, the pitch ratio is calculated to be $\frac{P}{D} = 0.934 \approx 0.950$, the number of propeller blades is 3, and the expanded area ratio is approximated to be 0.950 (see Table 2.2). Consequently, Figure A.1 can be used to find K_T.

By defining a set of advance ratios, the thrust coefficient K_T can be read out from the diagram, and thus all the parameters in (3.34) are known. As examples:

$$J_a = 0.00, \;\Rightarrow\; K_T = 0.10, \;\Rightarrow\; u = 0.001785|n|n, \tag{3.60}$$

$$J_a = 0.20, \;\Rightarrow\; K_T = 0.09, \;\Rightarrow\; u = 0.001607|n|n. \tag{3.61}$$

Plotting of these relations within the range of the propeller speed n, gives us the thruster characteristics (see Figure 3.8). The developed thruster characteristics can be compared to open water tests, performed by Stackpole (2015). The propeller efficiency test revealed a produced force for the Graupner propellers at approximately 12 N (see Figure 3.9), and by setting K_T to 0.05, the thruster characteristics reveals comparable results. Thus, the propellers for Neptunus is capable of producing 12 N in forwards and backwards direction.

Thruster allocation

There are two horizontal thrusters, placed on each side of the body. In addition, there is a vertical thruster, located straight above the center of gravity. Notice that on Neptunus, the horizontal thrusters are placed an angle α with respect to the x-axis, see Figure 3.10. Based on this information, the thruster configuration matrix can be established. Consequently, the thrust in the DOFs will be

$$\boldsymbol{\tau} = \begin{bmatrix} \cos(\alpha) & \cos(\alpha) & 0 \\ \sin(\alpha) & -\sin(\alpha) & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ L_1\sin(\alpha) + L_2\cos(\alpha) & -L_1\sin(\alpha) - L_2\cos(\alpha) & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \tag{3.62}$$

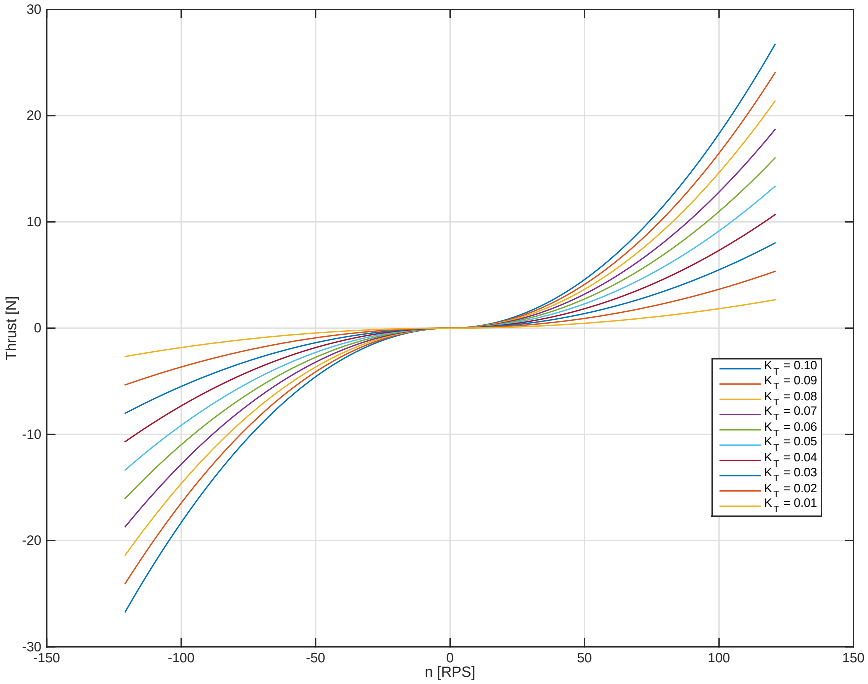


Figure 3.8: Thrust characteristics for the propellers on Neptunus.

which will transform the given force to the appropriate DOF. The angle α is evaluated to 8.914°, while the distances L_1 and L_2 are evaluated to 0.180 and 0.044 m, respectively. In the thruster allocation, u_1 represents right horizontal thruster, u_2 represents left horizontal thruster, while u_3 is the vertical.

Umbilical forces

To model the drag forces generated due to the presence of the umbilical, (3.38) and (3.39) are used. The diameter is measured to be $D = 0.01$ m. Further, the drag coefficient is found using the recommended practice regarding environmental conditions and environmental loads from Det Norske Veritas (2007, App. E). Using an ellipse, with D/L ratio of 1, the drag coefficient C_D is set to 1. For simplicity, $C_D = 1$ is set in all DOFs. Hence, the (differentiated) umbilical forces are

$$dF_{ii} = 5.125|v_i|v_i ds, \qquad i = 1,2,3 \tag{3.63}$$

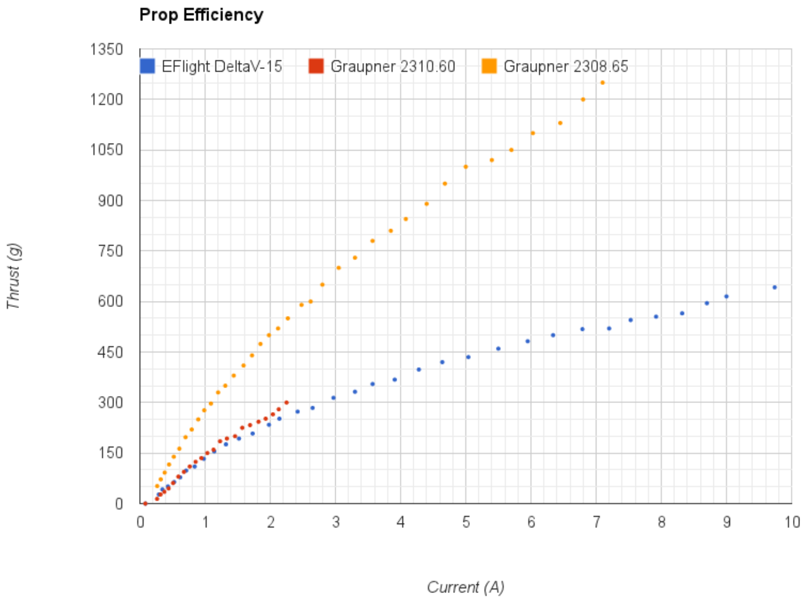


Figure 3.9: Preliminary propeller testing of Graupner 2308.65 - the propellers on Neptunus. Courtesy of OpenROV (2013).

for translational DOF, and

$$dM_{ii} = 5.125|v_i|v_i sds, \qquad i = 4,5,6 \tag{3.64}$$

for rotational DOF. Note that these forces are given for a infinitesimal length, and must be integrated over the umbilical length. This can be expressed as

$$F_{ii} = 5.125|v_i|v_i(\eta_i - \eta_0), \qquad i = 1,2,3 \tag{3.65}$$

for the translational DOF, and

$$M_{ii} = 2.563|v_i|v_i(\eta_i - \eta_0)^2, \qquad i = 4,5,6 \tag{3.66}$$

for the rotational DOF. Here η_0 is the initial position for the umbilical. Investigation of these expressions reveals that for the entire length of the umbilical in the sea, significant drag forces are generated.

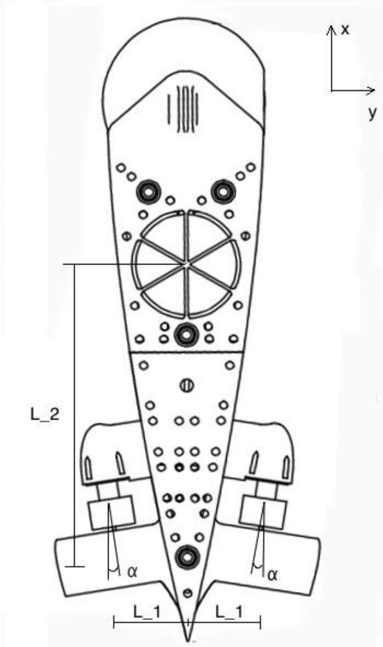


Figure 3.10: Thruster locations on Neptunus, seen from above

3.3 System identification for Cybership Enterprise I

The system identification for CSEI have been conducted earlier by NTNU (2015) and Skåtun (2011), and is only briefly presented. This is done in order to have a complete model for the further development of a motion control system.

For CSEI, towing tests have been performed in surge, sway, and yaw direction, and added mass and damping forces have been calculated. For this reason, only a 3 DOF model will be considered. These forces, as for Neptunus, have been curve fitted to obtain accurate parameters. Even though some of these drag forces are of third order, only linear and quadratic damping will be considered. This is mainly due to the low operating velocities.

3.3.1 Resulting system identification

The set up will follow from the established mathematical model from (3.43)- (3.44).

Center of gravity

The vector from CO to CG is found to be

$$\boldsymbol{r}_g = \begin{bmatrix} 0.0375 & 0.0000 & 0.0000 \end{bmatrix}^T. \tag{3.67}$$

Rigid body mass

The mass of CSEI is measured to be $m = 14.79$ kg, while the inertia moment in yaw is $I_z = 1.76$ kgm^2. With $\boldsymbol{r}_g$ as in (3.67), the rigid body mass matrix is

$$\boldsymbol{M}_{RB} = \begin{bmatrix} 14.790 & 0 & 0 \\ 0 & 14.790 & 0.554 \\ 0 & 0.554 & 1.760 \end{bmatrix}. \tag{3.68}$$

Rigid body Coriolis

From the rigid body mass matrix, a parameterization that fulfills the Coriolis property, $\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) = -\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)^T$ is

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & -14.790r & -0.554r \\ 14.790r & 0 & 0 \\ 0.554r & 0 & 0 \end{bmatrix}. \tag{3.69}$$

Added mass

The results from the towing tests reveal the added mass matrix to be

$$\boldsymbol{M}_A = \begin{bmatrix} 2.000 & 0 & 0 \\ 0 & 10.000 & 0 \\ 0 & 0 & 1.000 \end{bmatrix}. \tag{3.70}$$

Added mass Coriolis

As for the rigid body Coriolis matrix, a parameterization that fulfills the property $\boldsymbol{C}_A(\boldsymbol{\nu}_r) = -\boldsymbol{C}_A(\boldsymbol{\nu}_r)^T$ is

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & 10.000v \\ 0 & 0 & 2.000u \\ -10.000v & -2.000u & 0 \end{bmatrix}. \tag{3.71}$$

Linear and non-linear damping

The damping matrices are expressed as

$$\boldsymbol{D}_L = \begin{bmatrix} 0.655 & 0 & 0 \\ 0 & 1.330 & 0 \\ 0 & 0 & 1.900 \end{bmatrix}, \tag{3.72}$$

and

$$\boldsymbol{D}_q(\boldsymbol{\nu}_r) = \begin{bmatrix} -0.355|u_r| & 0 & 0 \\ 0 & 2.776|v_r| & 0.845|r| \\ 0 & 0.805|v_r| & 0.750|r| \end{bmatrix}. \tag{3.73}$$

Restoring forces

Considering only the horizontal plane of motion for CSEI, i.e. the surge, sway and yaw motion, no restoring forces will be present. Hence

$$\boldsymbol{g}_{sv}(\boldsymbol{\eta}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.74}$$

Thruster forces

The thruster mapping from the governing DOF to the different thrusters can be obtained using (3.33). Decomposition of the generated thrust forces from Figure 2.9 can be used to find the thrust configuration matrix. The work is done by NTNU (2015).

$$\boldsymbol{T}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos(\alpha_{\text{VSP1}}) & \cos(\alpha_{\text{VSP1}}) & 0 \\ \sin(\alpha_{\text{VSP1}}) & \sin(\alpha_{\text{VSP1}}) & 1 \\ L_{x,\text{VSP}}\cos(\alpha_{\text{VSP1}}) - L_{y,\text{VSP}}\sin(\alpha_{\text{VSP1}}) & L_{x,\text{VSP}}\cos(\alpha_{\text{VSP2}}) - L_{y,\text{VSP}}\sin(\alpha_{\text{VSP2}}) & L_{\text{bt}} \end{bmatrix}, \tag{3.75}$$

where α_{VSP1} and α_{VSP2} are the foil pitching angles. Moreover, the thrust coefficient matrix is given as

$$\boldsymbol{K} = \begin{bmatrix} K_{\text{VSP1}} & 0 & 0 \\ 0 & K_{\text{VSP2}} & 0 \\ 0 & 0 & K_{\text{bt}} \end{bmatrix}. \tag{3.76}$$

For closed loop control systems development, solving the thrust allocation equation with respect to u and α is complicated. Thus, some simplifications are made. In the following, it is assumed that

$$\begin{aligned} \alpha_{\text{VSP1}} &= \alpha_{\text{VSP2}}, \\ K_{\text{VSP1}} &= K_{\text{VSP2}}, \\ u_{\text{VSP1}} &= u_{\text{VSP2}}. \end{aligned} \tag{3.77}$$

With these assumptions, the simplified control allocation is given as

$$\underbrace{\begin{bmatrix} X \\ Y \\ N \end{bmatrix}}_{\boldsymbol{\tau}_e} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & L_{x,\text{VSP}} & L_{\text{bt}} \end{bmatrix}}_{\boldsymbol{T}_e} \underbrace{\begin{bmatrix} K_{\text{VSP1}} & 0 & 0 \\ 0 & K_{\text{VSP1}} & 0 \\ 0 & 0 & K_{\text{bt}} \end{bmatrix}}_{\boldsymbol{K}_e} \underbrace{\begin{bmatrix} u_{\text{VSP,x}} \\ u_{\text{VSP,y}} \\ u_{\text{bt}} \end{bmatrix}}_{\boldsymbol{u}_e}, \tag{3.78}$$

where $K_{\text{VSP1}} = 1.165$ and $K_{\text{bt}} = 2.629$. $L_{x,\text{VSP}}$ and L_{bt} are according to Table 2.4 -0.4575 and 0.3875, respectively. A visual representation of the simplification of the thruster allocation is seen in Figure 3.11. The two VSPs are replaced with a single virtual VSP, giving the approximately same behavior. Moreover, the actual control forces are

$$u_{\text{VSP1}} = u_{\text{VSP2}} = \sqrt{(u_{\text{VSP,x}})^2 + (u_{\text{VSP,y}})^2}, \tag{3.79}$$

and

$$\alpha_{\text{VSP1}} = \alpha_{\text{VSP2}} = \text{atan2}(u_{\text{VSP,x}}, u_{\text{VSP,y}}). \tag{3.80}$$

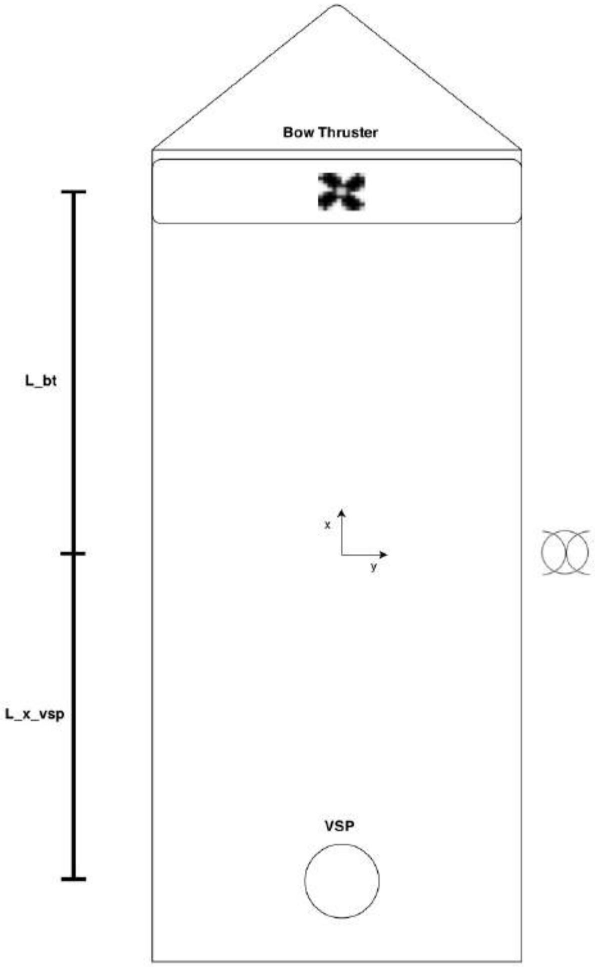


Figure 3.11: Simplified thruster allocation for CSEI Courtesy of Valle (2015).

3.4 Simulations Neptunus

In this section the mathematical model of Neptunus with the related parameters is evaluated. The simulation model form the basis for the later development of the motion control system, for path following purposes.

To validate the parameters obtained in this master thesis, a comparison with the parameters found by Follestad et al. (2014, Ch. 5) is performed.

The simulations include current forces unless otherwise indicated. The parameters are based on typical values found in the Trondheim's fjord (Ellingsen, 2004). Following the Gauss-Markov process formulation from Section 3.1.4, $V_{\max} = 0.02$ m/s, $V_{\min} = 0.001$ m/s, and $\mu_c = 0.01$. The current direction is $\beta_c = \pi$. The step size in all the scenarios is $h = 0.01$.

3.4.1 Assumptions

For comparison reasons related to the results given by Follestad et al. (2014, Ch. 6) and for convenience in the simulation studies, some assumptions are introduced.

Umbilical forces

In the forthcoming simulations, the drag forces induced by the umbilical are neglected

$$\boldsymbol{\tau}_{um} := 0 \tag{3.81}$$

This assumption is based on two reasons. The first one is that the umbilical was neglected during the simulations in Follestad et al. (2014, Ch. 6), due to low diameter, and thus low contribution. To obtain compliance in the tests in this master thesis, the umbilical is not considered.

The second reason is that continuous models of umbilicals are hard to solve when the properties change along the length of the cable Driscoll et al. (2000).

Added mass

By investigating the shape of Neptunus, some reductions can be made regarding the cross-coupled added mass terms. According to Faltinsen (1990, Ch. 3), symmetry around the axes can reduce the complexity of the hydrodynamic matrices. For Neptunus, top-bottom and port-starboard symmetry are registered. Thus, the added mass matrix is reduced to

$$\boldsymbol{M}_A = \begin{bmatrix} 2.0555 & 0 & 0 & 0 & 0 & 0 \\ 0 & 14.6033 & 0 & 0 & 0 & -0.8926 \\ 0 & 0 & 2.3295 & 0 & 0.0425 & 0 \\ 0 & 0 & 0 & 0.2569 & 0 & 0 \\ 0 & 0 & 0.0425 & 0 & 0.0740 & 0 \\ 0 & -0.8926 & 0 & 0 & 0 & 0.1498 \end{bmatrix} \tag{3.82}$$

in the simulation study.

3.4.2 Scenarios

In order to test the performance of Neptunus, three scenarios are considered.

- Scenario 1: straight line;
- Scenario 2: circle;
- Scenario 3: rise towards the surface.

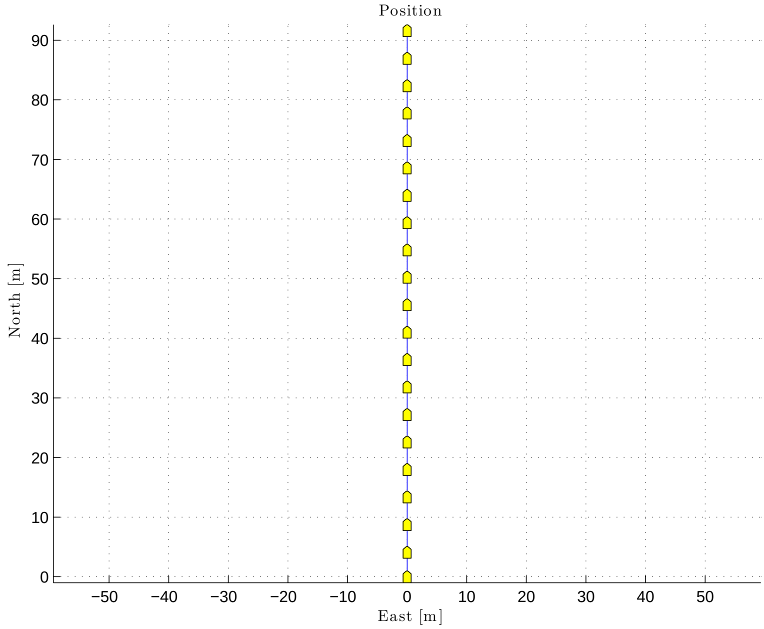


Figure 3.12: The position of Neptunus for Scenario 1.

Scenario 1: straight line

In the first test case, the behavior when moving straight forward is investigated. For this case the thruster input is

$$\boldsymbol{u} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T, \tag{3.83}$$

and the initial state vector

$$\boldsymbol{\eta}_0 = \begin{bmatrix} 0 & 0 & 40 & 0 & 0 & 0 \end{bmatrix}^T. \tag{3.84}$$

From Figure 3.12 and 3.13, it is observed that the ROV is moving straight forward, with no sideslip angle. This implies that the thrusters are placed convenient, and do not contribute to sideways or upwards movement. Since the surge dynamics in the mathematical model are decoupled, this behavior is expected. It is also noticed that the buoyancy is approximately zero, which implies a neutrally buoyant vehicle. From the velocity plot in Figure 3.14, it is possible to see that the current is working against the direction of the ROV. The angular rates are evaluated to be approximately zero, leading to a satisfying passively stable behavior.

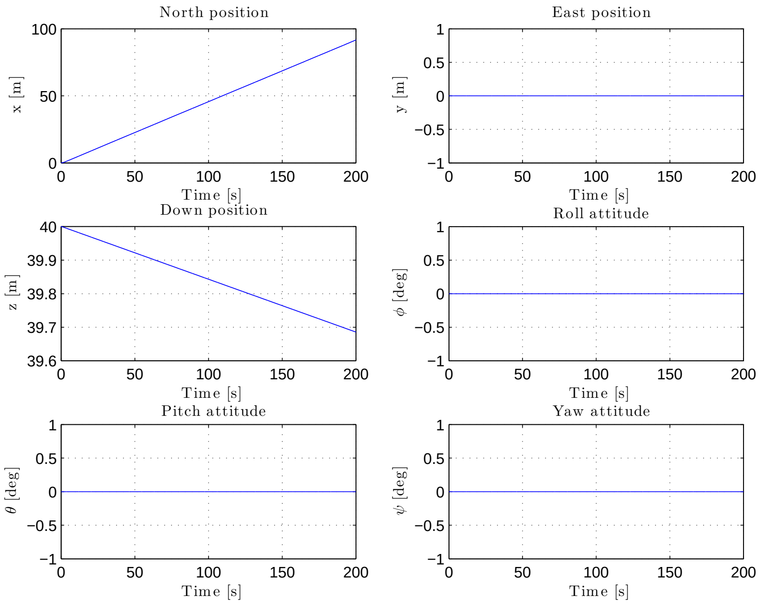


Figure 3.13: The position and attitude of Neptunus for Scenario 1.

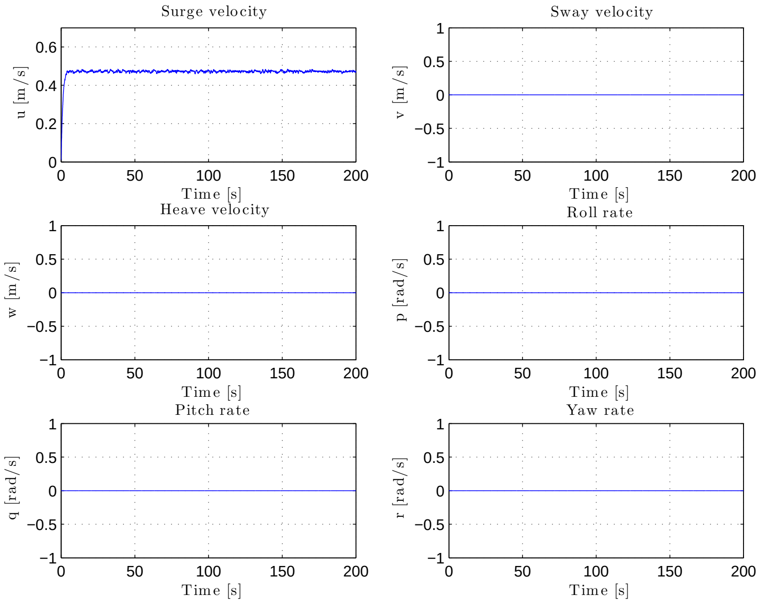


Figure 3.14: The velocities and rates of Neptunus for Scenario 1.

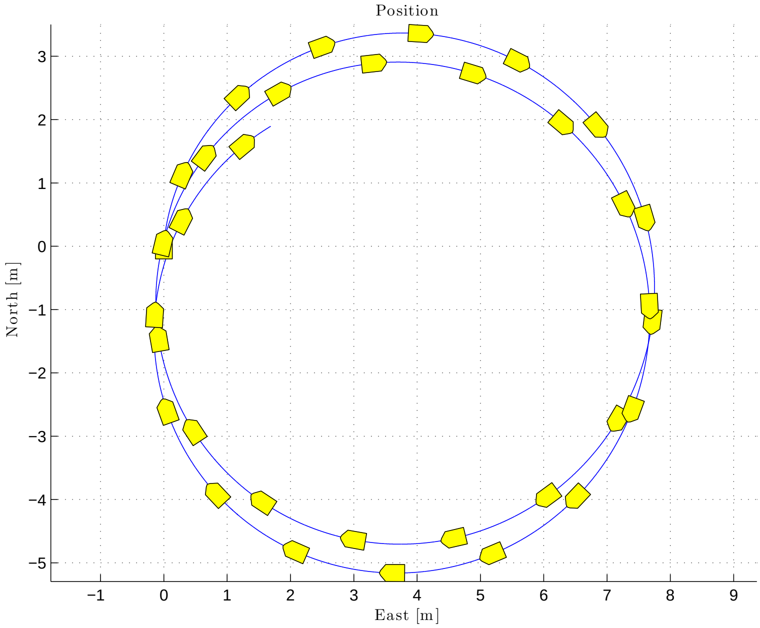


Figure 3.15: The position of Neptunus for Scenario 2.

Scenario 2: circle

In the next test case the maneuverability of the ROV will be tested performing a circular motion. The simulation time, step size and initial state vector are the same as for Scenario 1. The thrust input is

$$\boldsymbol{u} = \begin{bmatrix} 0.6 & 0.3 & 0 \end{bmatrix}^T \tag{3.85}$$

For this case, the current is turned off. This is done in order to test the turning capabilities of the ROV without environmental disturbances influencing the performance. In addition, no current is in compliance with the scenario considered in Follestad et al. (2014, Sec. 6.3.2).

From the North-East plot in Figure 3.15, it is observed that Neptunus has satisfying turning capabilities. The response in Scenario 2 is further presented in Figure 3.16, which presents the state vector behavior in all 6 DOFs. In Figure 3.17 an approximately constant yaw rate is reported, leading to a stable behavior. The rest of the velocities and rates were stable and as expected. This concludes that no Munk moment is present, which is a destabilizing moment in yaw. The Munk moment can arise when there is significant difference in the added mass in surge and sway, which is the case for Neptunus (Faltinsen, 1990, Ch. 6).

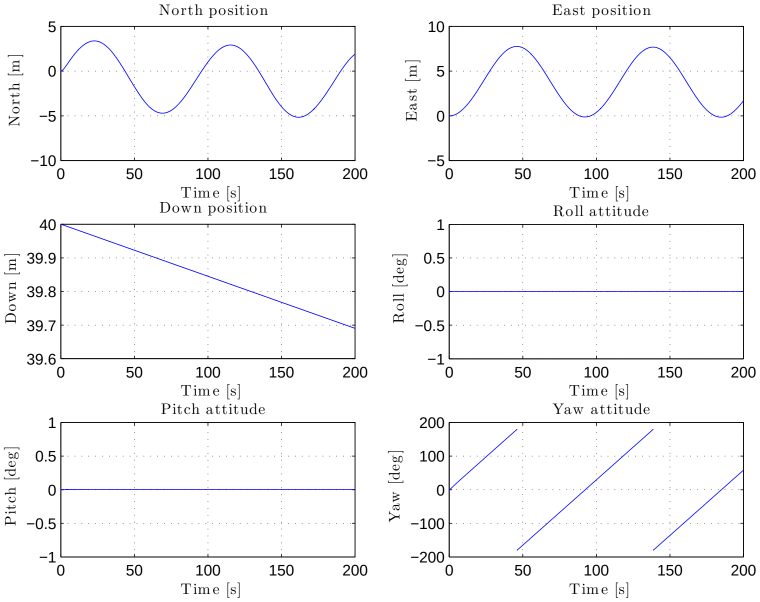


Figure 3.16: The position and attitudes of Neptunus for Scenario 2.

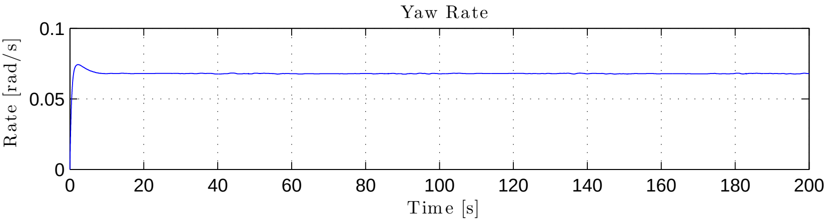


Figure 3.17: The yaw rate for Neptunus for Scenario 2.

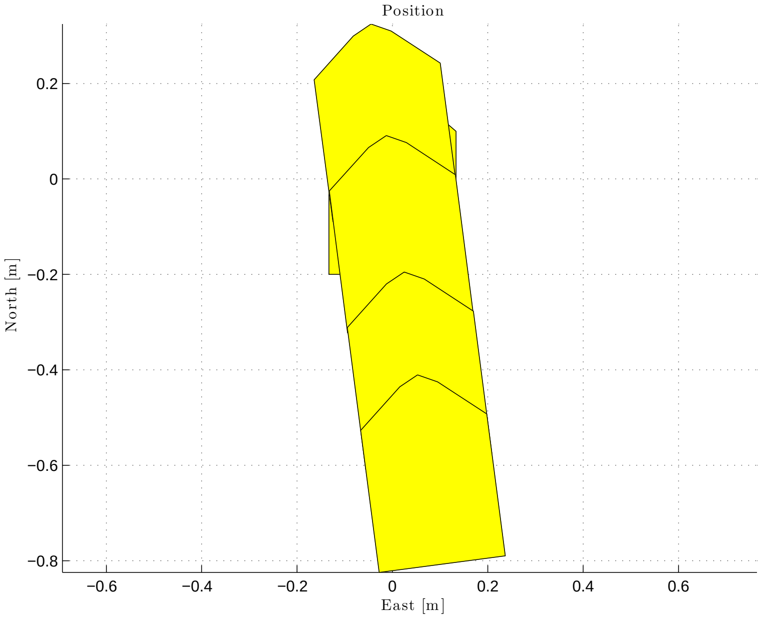


Figure 3.18: The position of Neptunus for Scenario 3.

Scenario 3: rise towards the surface

For the third scenario, the focus is on the vertical thruster, and Neptunus' passive stability properties. By applying thrust in the (negative) vertical direction, the rising properties can be investigated. The initial state vector is

$$\boldsymbol{\eta}_0 = \begin{bmatrix} 0 & 0 & 40 & 10 & 25 & 0 \end{bmatrix}^T \tag{3.86}$$

where the roll and pitch angles are given in degrees. Finally, the thrust input is set to

$$\boldsymbol{u} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T \tag{3.87}$$

In Figures 3.18 and 3.19, the motion response can be seen. With the vertical thruster turned on, the ROV is rising. Notice also the stabilization of the roll and pitch angles. The rapid roll and pitch oscillations around equilibrium are due to the low damping. Due to a initial roll and pitch angle, and the fact that the vertical thruster is on, the roll and pitch motion do not converge entirely to zero. This may be due to an error in the numerical simulation. Motion like this behavior is not experienced in full scale tests. However, these oscillations are small. Combined with the restoring forces, presented in Figure 3.20, it is concluded that Neptunus is showing satisfying stability characteristics and can be treated as a passively stable ROV.

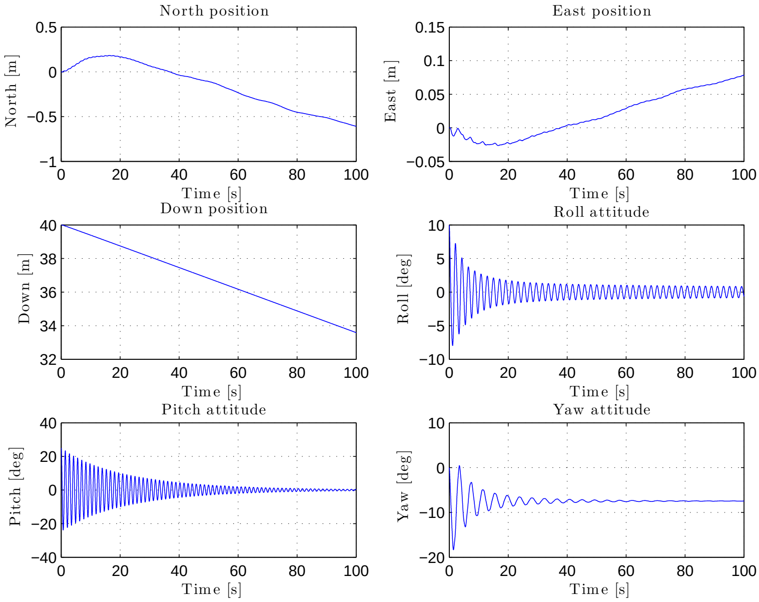


Figure 3.19: The position and attitude of Neptunus for Scenario 3.

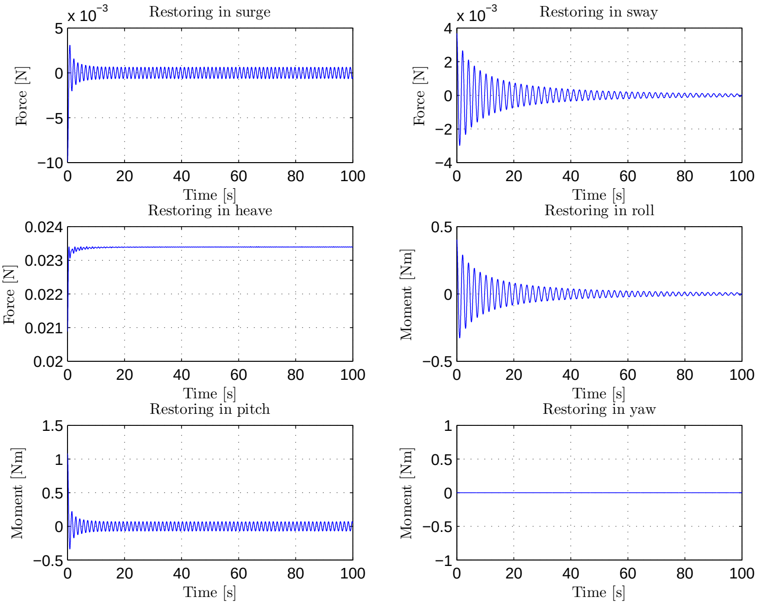


Figure 3.20: The restoring forces for Neptunus for Scenario 3.

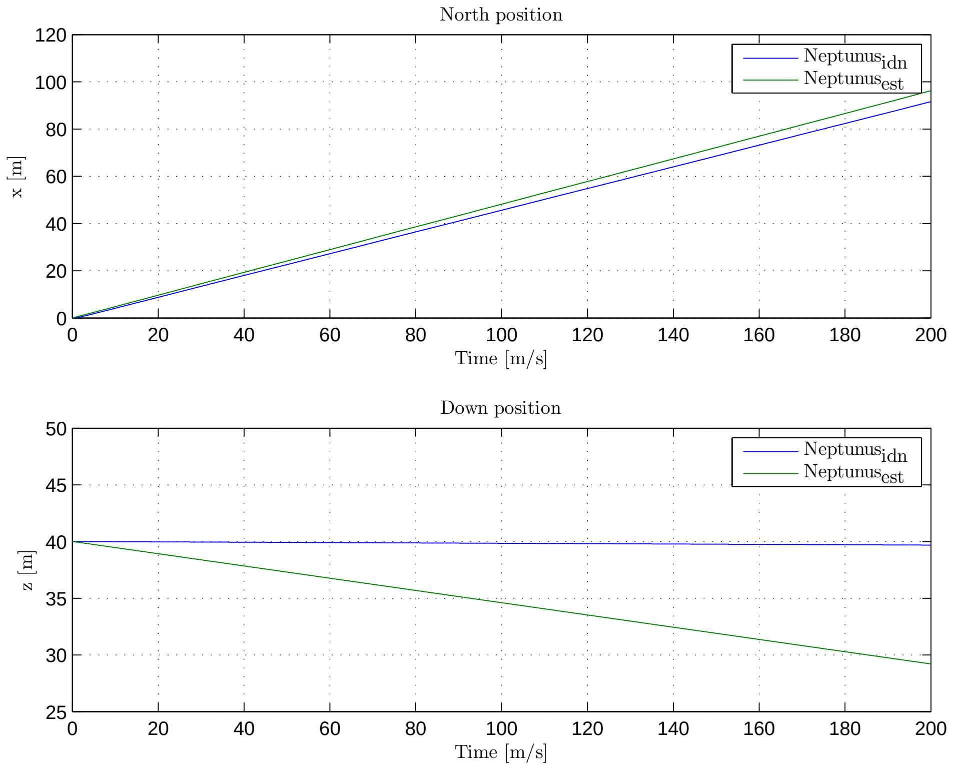


Figure 3.21: The identified and estimated response of Neptunus in North and Down direction for scenario 1.

3.4.3 Comparison

The three scenarios revealed expected behavior, and the performance is satisfying in the different tests. The behavior of Neptunus in Scenario 1 was compared against the behavior from the same scenario by Follestad et al. Results revealed that the responses were in compliance. In the comparison, Neptunus$_{\text{idn}}$ represents the model identified by towing tests and software analysis. Neptunus$_{\text{est}}$ represents the estimated model.

By looking at the North and Down movement in Figure 3.21, it is observed that the distance traveled North is approximately the same. This concludes that the estimated parameters in surge are in compliance with the parameters found by experiments.

The big difference in depth is present due to different buoyancy considerations. In the project thesis, a more positively buoyant vehicle was considered, compared to a neutrally buoyant vehicle in this master thesis. The chambers for adjusting the weight and buoyancy for Neptunus are simply altered to obtain neutrally buoyant performance.

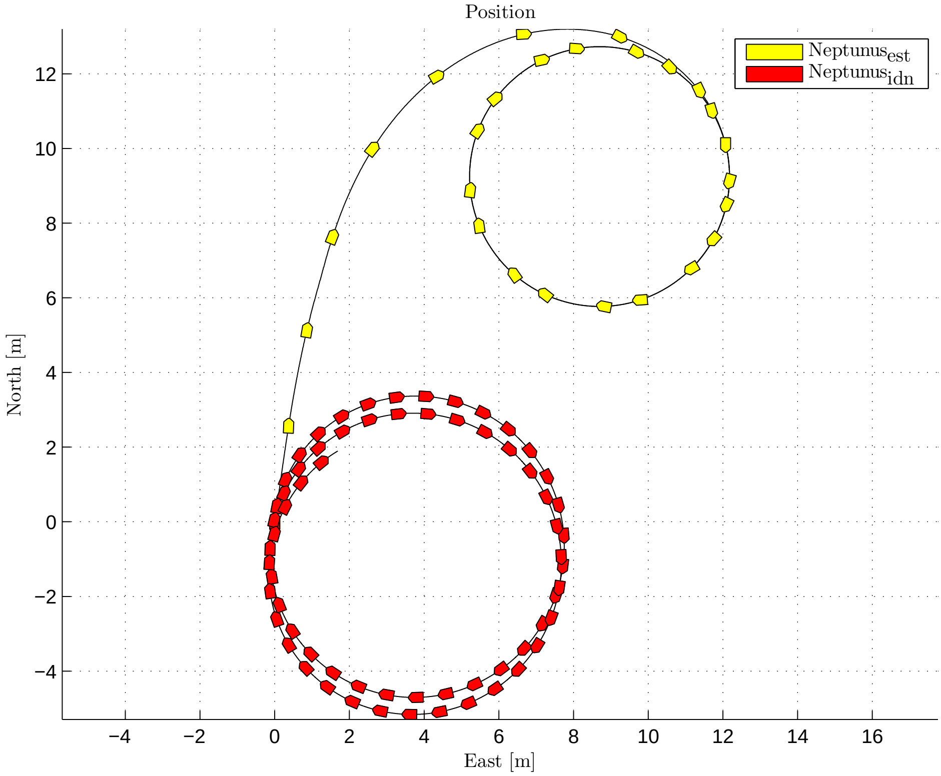


Figure 3.22: Neptunus in North East position for Scenario 2.

By comparing Scenario 2 with the new parameters against the parameters found by Follestad et al. (2014), a different behavior was registered. As seen in Figure 3.22, the new model goes faster into a circle motion. The main reason is in the assumption of a box in the project thesis, while a more correct geometric consideration is present in this master thesis, in addition to higher damping in yaw.

A comparison for the estimated and identified Neptunus model during Scenario 3 can be seen in Figure 3.23. It is observed that the identified model has a more rapid roll and pitch response, due to the lower damping terms. As a consequence, the estimated model has less rapid oscillations. This could be due to the lower damping, but also to numerical simulation errors, since the behavior is not present in the real life. However, both models renders stable roll and pitch dynamics. The old model is moving more backwards, which is due to lesser damping in surge and bigger assumed frontal area.

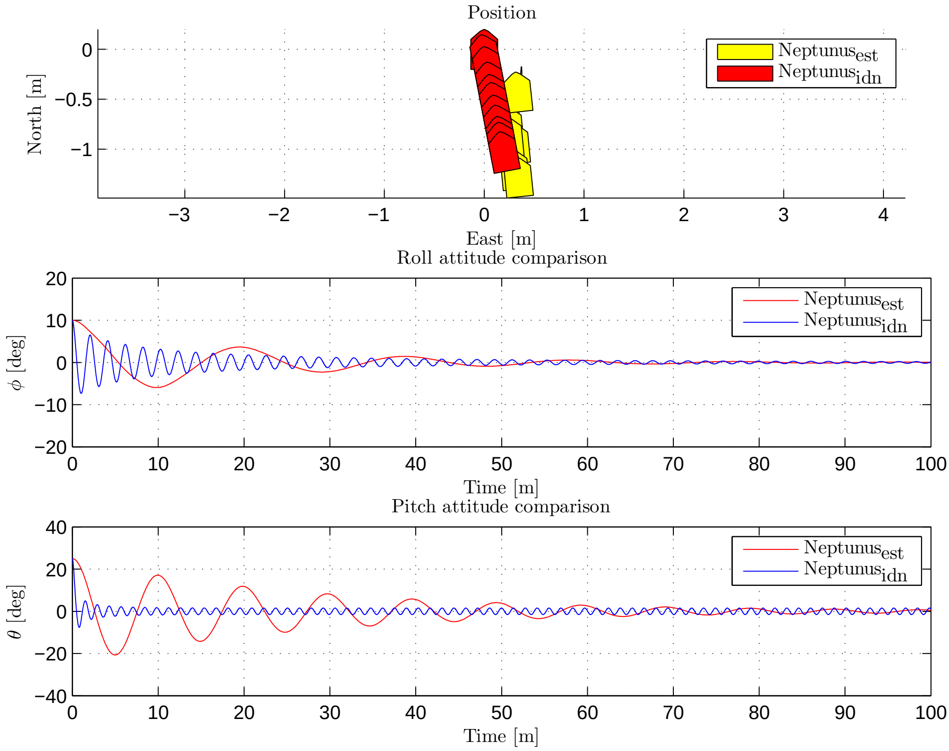


Figure 3.23: Comparison of Scenario 3 for Neptunus.

3.4.4 Concluding remarks

Three scenarios for Neptunus have been considered. The response of Neptunus reveals a stable vehicle, with good maneuvering capabilities. Even though it is underactuated, it is capable of moving along arbitrary paths. However, in presence of strong sideways current, it cannot counteract these forces, since no lateral thruster is included in the design.

From the results, it is concluded that the estimated parameters are in compliance with the parameters found by experimental tests and software analyses. However, due to the simplifying box-shaped assumption by (Follestad et al., 2014, p. 91), the tests showed some differences in certain tests.

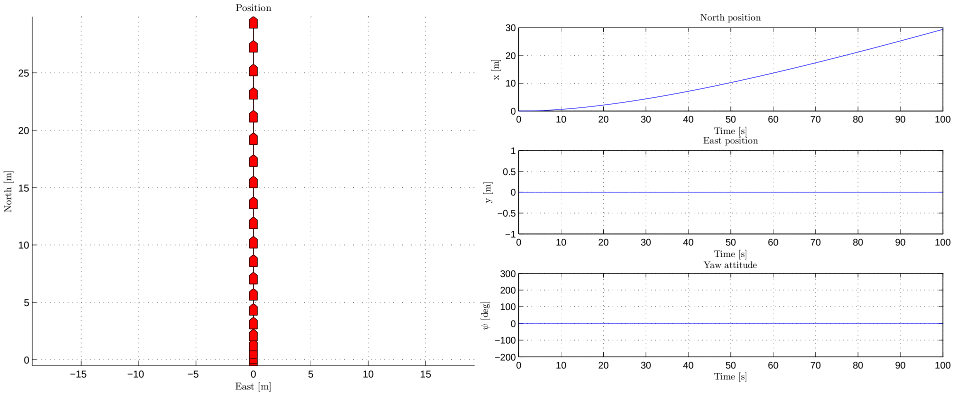
The turning capabilities are better than first assumed. This is proven in Scenario 2. The main reason is that the identified model is modeled with the proper foil shape, in stead of the box-shape considered at first. In addition, the damping is smaller in yaw than estimated.

Neptunus is proven to be passively stable in roll and pitch. This was also the conclusion with the old model. From Scenario 3, and the related comparison, it is revealed that the roll and pitch motion converge fast to equilibrium, however some small oscillations due to low roll and pitch damping. Based on the restoring forces and the buoyancy characteristics of Neptunus, these oscillations can be neglected. In the following, these degrees of freedom are not considered, since no actuators or control laws are needed to control these states.

Due to the ventilation holes, and chambers for placing buoyancy and weight elements, Neptunus can be designed to have an arbitrary buoyancy. Follestad et al. (2014) set the buoyancy higher than it currently is, and thus some differences are identified and discussed. It is further assumed that the vehicle is neutrally buoyant, and thus the heave state is not considered in the motion control system.

Consequently, in the path following control systems, a 3 DOF model consisting of x, y and ψ as states is used. The 3 DOF model, with numerical values are given in Appendix A.6.

Thruster losses are not accounted for in the simulation model. Consequently, it takes only 1 N to drive forwards with a speed of 0.5 m/s. Considering that the maximum speed is around 1.3 - 1.5 m/s, and is capable of producing 12 N, additional thruster dynamics should be implemented.



(a) The North East position.

(b) The position in 3 DOF.

Figure 3.24: CSEI Scenario 1.

3.5 Simulations Cybership Enterprise I

3.5.1 Scenarios

To test the response of the mathematical model of CSEI, a series of benchmark tests are conducted and evaluated. As for Neptunus, 3 tests are performed. All the tests are conducted with a stepsize $h = 0.01$, and the time duration is set to $t = 100$ s. For all tests, current is present. The current parameters are the same as for Neptunus. The initial position is the same in all scenarios, and is $\boldsymbol{\eta}_0 = [0\ 0\ 0]^T$.

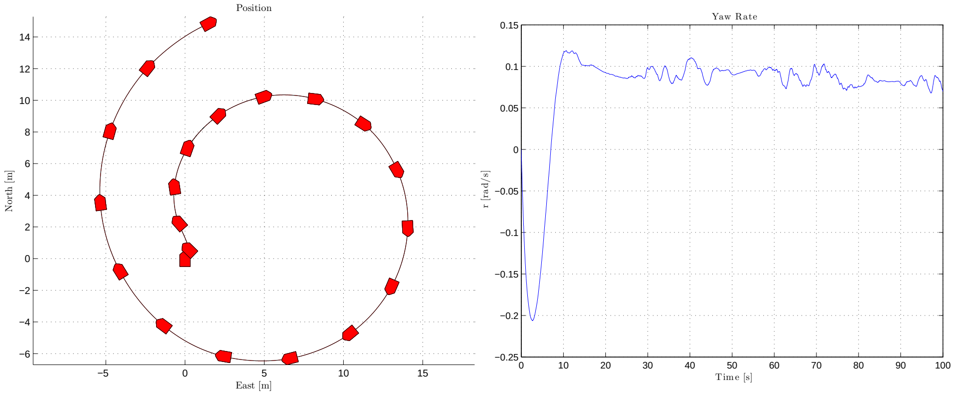
Scenario 1: straight line

In this scenario, the CSEI's capabilities will be investigated when driving forward, with a constant thrust input

$$\boldsymbol{u} = \begin{bmatrix} 0.2 & 0.0 & 0.0 \end{bmatrix}^T \tag{3.88}$$

The North-East and 3 DOF position response is presented in Figure 3.24a and 3.24b, respectively. The results reveal pure forward motion, without induced sway and yaw motion. This is in compliance with the expected results, since the surge dynamics in the simulation model are decoupled.

It is noticed that CSEI is slower in the start of the simulation. Because of the mass and added mass in surge, some time is needed to make CSEI move with a constant surge velocity.



(a) Position.

(b) Yaw rate.

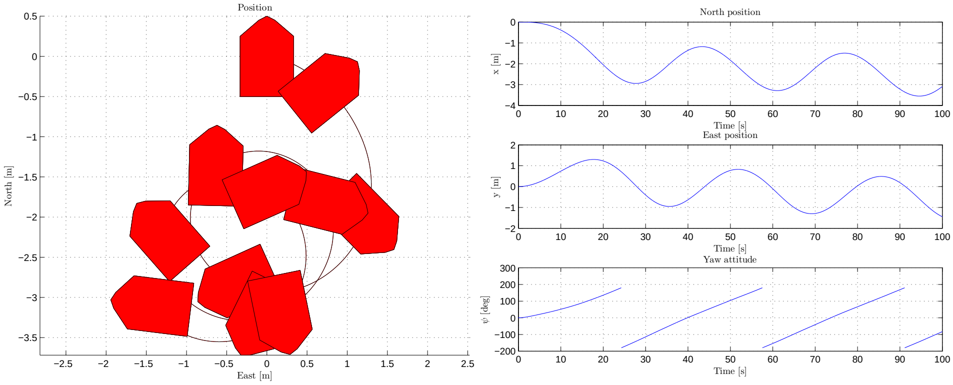
Figure 3.25: CSEI Scenario 2.

Scenario 2: circle

The turning capabilities for CSEI is tested. In this scenario, only the VSPs are turned on. This opens for investigation of the response under circular motion when CSEI is a underactuated vehicle. This reveals a preliminary emulation of a rudder-propeller transformation for CSEI. The thrust vector is

$$\boldsymbol{u} = \begin{bmatrix} 0.3 & 1.0 & 0.0 \end{bmatrix}^T \tag{3.89}$$

For Scenario 2, the turning capabilities for CSEI are satisfying. However, CSEI is not as stable as Neptunus, which showed approximate constant yaw rate under the same conditions. The resulting motion is seen in Figure 3.25a. The non-constant turning rate is shown in Figure 3.25b. The oscillations in the yaw rate are due to the presence of the slowly varying current.



(a) North East position.

(b) 3 DOF position.

Figure 3.26: Scenario 3.

Scenario 3: bow-thruster

The bow thruster response is tested with

$$\boldsymbol{u} = \begin{bmatrix} 0.0 & 0.0 & 0.2 \end{bmatrix}^T \tag{3.90}$$

The behavior is as expected: CSEI turns around it's own axis, and moves backwards. This is in compliance with how the response is expected to be in real life. The behavior is presented in Figure 3.26a and 3.26b.

3.5.2 Concluding remarks

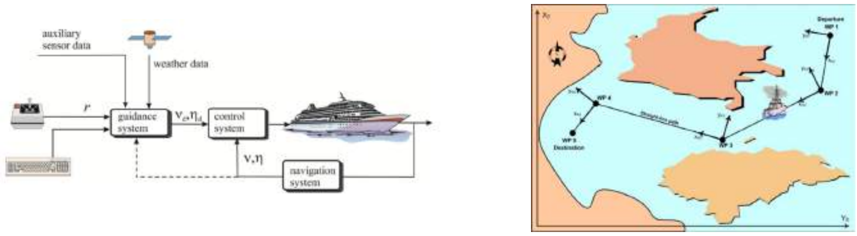
As for Neptunus, three scenarios for CSEI have been conducted. The scenarios have revealed the response of the mathematical model of CSEI under different thrust inputs. CSEI is a stable surface vessel, and the response is satisfying when giving a commanded thrust. This model of CSEI will be considered as the plant model for the path following control systems.

Chapter 4

Guidance and path following systems

During the last century, a focus area has been to automatically control marine vessels between predefined waypoints or along a path. Today, the control algorithms are able to make vessels maneuver with high accuracy and precision without human interference. Features as collision avoidance, current accounting, target tracking, and fault tolerant path following have been investigated thoroughly. In Figure 4.1 a general closed loop guidance can be seen. Commanded signals are given, and the controller makes the vessel follow the reference signal. In Figure 4.1b, a path defined by waypoints is presented.

A pioneer within marine control systems in Norwegian context is Jens Glad Balchen. He founded the department of engineering cybernetics at *Norges Tekniske Høgskole* (NTH, former NTNU), and was one of the men behind the first commercial dynamic positioning system for a marine vessel (Bjørnstad, 2009, Sec. 2.5). Much of the research within this area is related to the findings in his work.



(a) Guidance system. Courtesy of Fossen (2011).

(b) Waypoints. Courtesy of Skjetne (2005).

Figure 4.1: Closed loop guidance control system, and a generated path.

A set of guidance control systems are given in Fossen (2011, Ch. 10). Here *target tracking*, *pure pursuit*, and *constant bearing* are typical guidance systems for tracking a point, either moving or standing still. In addition, trajectory tracking is covered. Breivik and Fossen (2004) propose a lookahead based *line of sight* (LOS) controller designed for straight line and circle following for a marine vessel. Skjetne (2005) presents various maneuvering control designs, where the aim is to solve both a *geometric* and a *dynamic* task. This control design has been implemented and tested on the model ship Cybership II in the MC lab (Skjetne et al., 2004). Different planning and paths for marine vessels have been investigated by Dahl (2013), which covers simple straight line paths, to more complex *Dubin's path* introduces *Fermat's spiral*.

Guidance and path following control systems have also been implemented and tested at model ships in the MC lab, as a part of the master thesis. Orsten (2014) implements a LOS guidance and control method on CSEI, to tow icebergs. *Non-linear PID* and *LgV-backstepping* controllers were implemented to perform LOS steering for CSEI by Tran (2014).

In this chapter waypoint tracking and path following control algorithms for Neptunus and CSEI are proposed. The controllers vary in complexity, from simple heading controllers to more advanced maneuvering based controllers.

Since both Neptunus and CSEI in some control systems are treated as underactuated vessels, some modifications related to work space and thruster allocation must be taken into consideration. In the simulations, ocean current is present, and must be accounted for in the control designs. This is often done by implementing integral action. Another method is to use adaptive control and estimation theory to find the current forces, and further counteract them in the control system. In Do et al. (2004) a robust adaptive controller is implemented on a surface vessel to estimate and account for the current forces. Adaptive control in full scale tests on the ODIN ROV is performed in Antonelli et al. (2001).

For the operator to give signals to the vehicle, without risking high transients in the control system, a reference model is often implemented. Information regarding reference models are provided in Section 4.3.

The different guidance and path following systems are presented in this chapter, with accompanying simulations, both for Neptunus and CSEI. In this chapter, the mathematical models established in Chapter 3 will be used as basis for the control design.

4.1 Control of underactuated vehicles

When dealing with motion control systems, it is important to distinguish between underactuated and fully actuated vessels. A fully actuated vessel is capable in producing thrust in all DOF simultaneously, while a underactuated vessel can only generate thrust in certain degrees of freedom. Some definitions regarding motion control of desired degrees of freedom need to be familiar in this context (Fossen, 2011, Sec. 9.4):

- *Configuration space*: A n-dimensional space, where all possible positions and orientations that a craft may attain, possibly subject to external constraints.
- *Work space*: A reduced space of dimension $m < n$, where the control objective is defined.

In addition, r denotes the number of thrusters. ROV Neptunus is an underactuated vehicle, since it can only produce thrust in surge, yaw and heave. It is not equipped with thruster to produce pure sway, roll, or pitch motion. Thus, in the configuration space, $n > r$, and Neptunus is underactuated. Moreover, with the reduced model of Neptunus, where surge, sway, and yaw motion are considered, Neptunus is still underactuated. With two thruster, i.e. $r = 2$, and $m = 3$.

CSEI is underactuated in the configuration space, since it can not produce thrust in heave, roll, or pitch. However, in the 3 DOFs work space, CSEI is fully actuated. With a bow thruster, and two VSPs, it is capable of producing thrust in surge, sway and yaw.

By disabling the bow thruster, an assumption of mapping the VSPs into a rudder-propeller system, it will be treated as an underactuated vessel.

4.2 Thruster allocation

The motion control system calculates the desired actuation in the various DOF. In order to map these actuation forces to the thruster on the vehicle, a *thruster allocation* must be established. The thruster allocation serves the correct amount of forces to the different thrusters. Recalling the thruster forces as

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u}, \tag{4.1}$$

the interest is now on the inverse of this equation, i.e

$$\boldsymbol{u} = (\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K})^{-1}\boldsymbol{\tau}. \tag{4.2}$$

The inverse of $\boldsymbol{T}(\boldsymbol{\alpha})$ may not be invertible. A solution is to use the *Moore-Penrose pseudo inverse* to find the inverse of $\boldsymbol{T}(\boldsymbol{\alpha})$ (Fossen and Johansen, 2006, Eq. (15)):

$$\boldsymbol{T}^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{T}(\boldsymbol{\alpha})^{T}(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}(\boldsymbol{\alpha})^{T})^{-1}, \tag{4.3}$$

where $\boldsymbol{W}$ is a matrix for weighting the thrusters. Thus, thruster input is written as

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{T}^{\dagger}\boldsymbol{\tau}. \tag{4.4}$$

4.2.1 Neptunus

Recalling the thruster configuration from Figure 3.10 and in (3.62). By inserting into (4.4), thruster allocation is

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0.5061 & 0.0000 & 8.0000 \\ 0.5061 & 0.0000 & -8.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} X \\ Y \\ N \end{bmatrix}. \tag{4.5}$$

Here, $\boldsymbol{W} = \boldsymbol{I}$ has been used to weight the thrusters.

4.2.2 CSEI

Two different approaches on the thruster allocation have been considered for CSEI.

Underactuated case

Treating CSEI as an underactuated vessel, the bow thruster is turned off, and the two VSPs are mapped into a rudder-propeller system.

$$(u_{\text{VSP}}, \alpha_{\text{VSP}}) \rightarrow (u^*_{\text{VSP}}, \delta), \tag{4.6}$$

where u^*_{VSP} is the forward velocity, and δ is the angle of the VSP foils, treated as a rudder. To develop a representation of this, the yaw dynamics are considered linear, and given as the first order Nomoto (1957) model

$$T\dot{r} + r = K\delta, \tag{4.7}$$

where T and K are time constants. To obtain proper values for these constants, *Kempf's zigzag maneuver* is considered, which is a standardized maneuver test from the *International Towing Tank Conference* (ITTC) in 1963.

The test is performed by changing the rudder back and forth between -20° and 20°. A saturation on α_{VSP} is set to (40, -40) degrees. The rudder angle is kept constant until the vessel reaches the desired heading, before the rudder is reversed. This process in ongoing for five rudder changes. In Figure 4.2, a zigzag maneuver for CSEI is performed. Notice that the stern doesn't *slide* out, which is a normal behavior for conventional rudder-propeller ships. The change in α (or δ) occurs instantly.

Under the assumption that the yaw response to CSEI is approximately linear, the zigzag maneuver can be used to find T and K. According to Skogestad (2003), the time constant is defined as *the time it needs for the step response to reach* $(1-\frac{1}{e}) \approx$ *63 % of the final value.* Using this definition on the first and second step response in the zigzag maneuver, the mean time constant is evaluated to

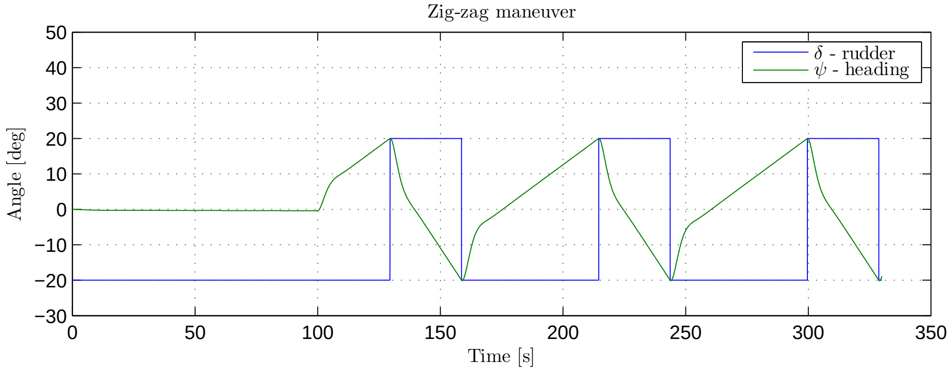
$$T = 0.63\left(\frac{(214.56-158.65)+(158.65-129.50)}{2}\right) = 27.79. \tag{4.8}$$

Furthermore, K can be obtained when the acceleration is zero, such that

$$K = \frac{r}{\delta}. \tag{4.9}$$

In Figure 4.3 r and $\dot{r}$ are presented. Investigating r when $\dot{r}$ is zero, and finding the corresponding δ yields

$$K = \frac{0.1495}{\frac{\pi}{9}} = 0.43. \tag{4.10}$$



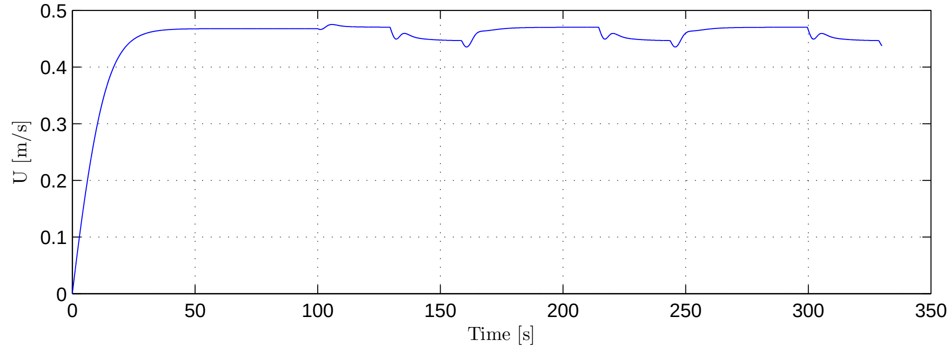


Figure 4.2: Zigzag maneuver for CSEI.

Fully actuated case

When the bow thruster is activated, CSEI is fully actuated. The thruster allocation is found by simple inversion of (4.1). Hence, the allocation is

$$\begin{bmatrix} u_{\mathrm{VSP,x}} \\ u_{\mathrm{VSP,y}} \\ u_{\mathrm{bt}} \end{bmatrix} = \begin{bmatrix} 0.8584 & 0.0000 & 0.0000 \\ 0.0000 & 0.3936 & -1.0158 \\ 0.0000 & 0.2059 & 0.4501 \end{bmatrix} \begin{bmatrix} X \\ Y \\ N \end{bmatrix}. \tag{4.11}$$

In Figure 4.4 the different thruster setups for CSEI is presented.

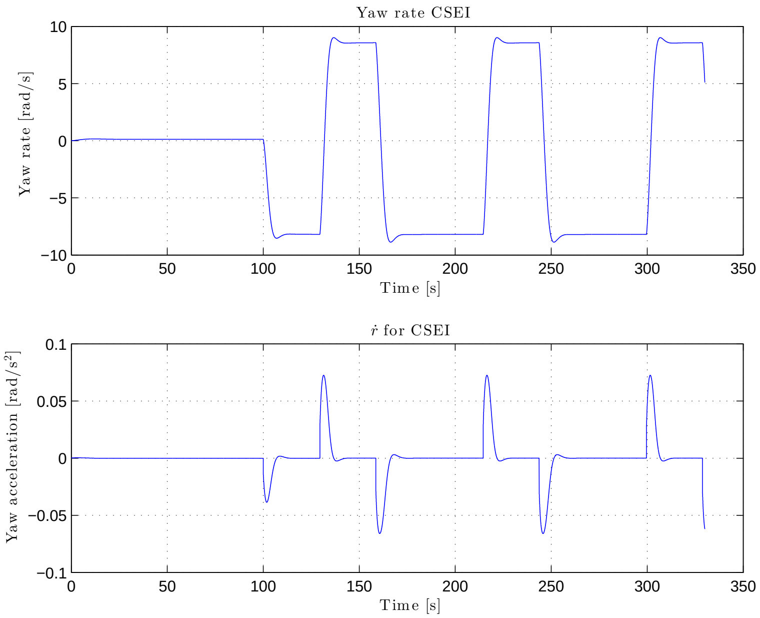
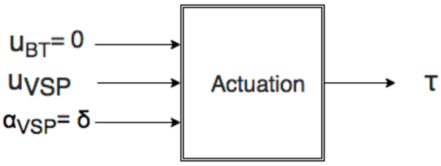
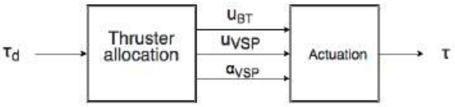


Figure 4.3: The yaw rate and its derivative for the zigzag maneuver.



(a) Underactuated.



(b) Fully actuated.

Figure 4.4: Thruster configurations for CSEI.

4.3 Reference models

In order to avoid instant changes in the desired references, which may cause high transients and instability, a reference model is often implemented. The reference model creates a smooth change in the desired reference signal, which the vehicle is able to follow. In its simplest form this can be a use of a low-pass filter. It is important to note that no generic representation of a reference model exist, and the performance of the reference model depends on the tuning. It is important that the bandwidth of the reference model is set lower than the bandwidth of the motion control system. This implies that the motion control system will follow the reference, and not conversely, with stable behavior.

A common way to design the reference model is to create a simple replica of the system of consideration. For a marine vessel, a *mass-spring-damper* structure is often used, i.e (Fossen, 2011, Eq. (10.19))

$$h_{lp}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.12}$$

where ζ is the damping ratio, and ω_n is the natural frequency. Based on the system, several reference models are purposed in the literature. In this master thesis, the *velocity-reference model* is taken into consideration. It provides smooth signals to the velocity and acceleration. Fossen (2011, Eq. (10.23)) writes the model in matrix form as

$$\ddot{\boldsymbol{v}}_d + 2\boldsymbol{\zeta}\boldsymbol{\Omega}\dot{\boldsymbol{v}}_d + \boldsymbol{\Omega}^2\boldsymbol{v}_d = \boldsymbol{\Omega}^2\boldsymbol{r}^b, \tag{4.13}$$

where $\boldsymbol{v}_d$ is the desired velocity vector, $\boldsymbol{\zeta}$ is the damping ratio vector, and $\boldsymbol{\Omega}$ is the natural frequency vector. $\boldsymbol{r}_b$ is the commanded input signal from the operator. A block diagram showing the reference model is seen in Figure 4.5.

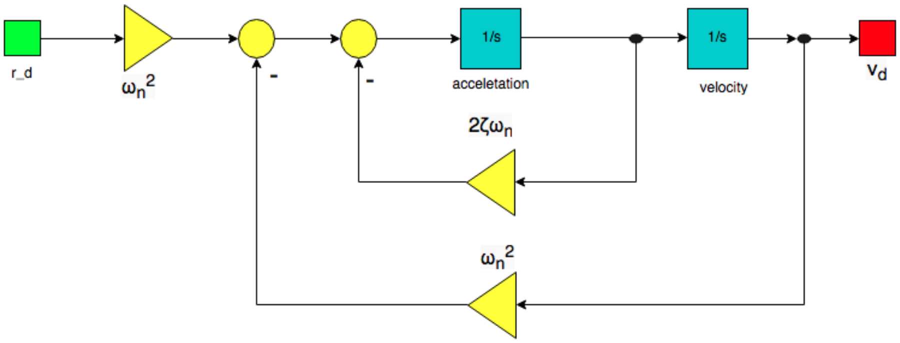


Figure 4.5: Velocity reference model.

An example of a generated reference signal, compared to a signal that has not been sent through a reference model, is shown in Figure 4.6. Here, a commanded velocity in surge steps from 0 to $u_r = 0.6$ m/s after 10 seconds. The damping ratio is set to $\zeta = 1$, while the natural frequency is chosen as $\omega_n = 0.8$ rad/s. It is observed that u_d provides a smoother reference for the vehicle to follow, and thus instant changes in the control system can be avoided.

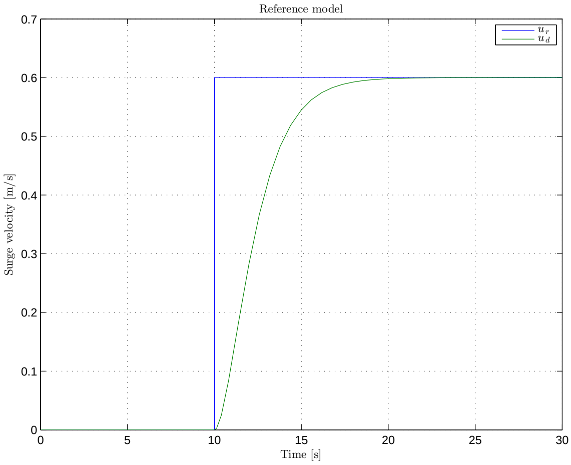


Figure 4.6: The desired velocity u_d versus the commanded signal u_r.

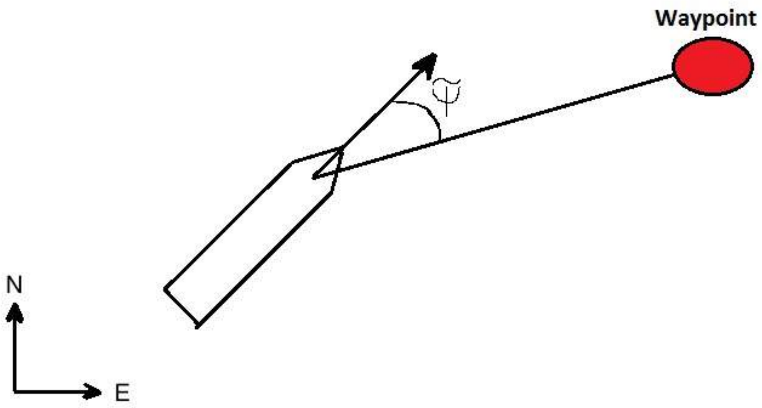


Figure 4.7: Heading on waypoint.

4.4 Heading on waypoints

The first control algorithm of concern is basic, and only concerns minimizing the error between the current heading ψ and the desired heading ψ_d. Dahl (2013) refers to this as *heading on waypoints*, see Figure 4.7. As long as the error $\tilde{\psi} = \psi - \psi_d$ is minimized, the object will hit the waypoint at some point.

In its simplest form, heading on waypoint can be solved with a P controller, which regulates the error to zero, i.e.

$$\tau_\psi = -K_p(\psi - \psi_d) = -K_p\tilde{\psi}, \tag{4.14}$$

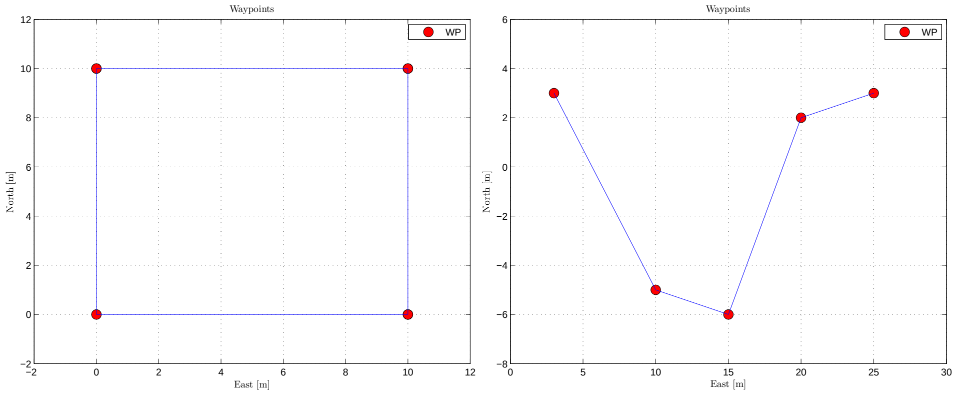
where $K_p > 0$ represents the proportional gain, and is a tunable parameter. However, in the presence of current or other disturbances, integral action is needed to remove the steady state errors:

$$\tau_\psi = -K_p\tilde{\psi} - K_i\int_0^t \tilde{\psi}(\tau)d\tau, \tag{4.15}$$

where K_i is the integral gain. The desired heading ψ_d is found by

$$\psi_d = \text{atan2}\left(\frac{y_i - y(t)}{x_i - x(t)}\right), \tag{4.16}$$

where x_i, y_i is the x and y position of the current waypoint. $x(t)$ and $y(t)$ is the vehicle position in North East. Moreover, it is desirable for the vehicle to follow a commanded velocity towards the waypoint. In presence of disturbances, the surge velocity is con-



(a) Squared path.

(b) Curved path.

Figure 4.8: Heading on WP scenarios.

trolled by

$$\tau_u = -K_p \tilde{u} - K_i \int_0^t \tilde{u}(\tau) d\tau. \tag{4.17}$$

Here the velocity error is $\tilde{u} = u - u_d$. With these control laws simulations can be conducted on Neptunus and CSEI.

4.4.1 Simulations

In this thesis, two scenarios using heading on waypoints will be considered. A set of waypoints forming a square is in the first scenario, while the second scenario is a curved path. In Figure 4.8 the two set of waypoints are presented.

The change of direction towards the next waypoint occurs when the vehicle enters the *circle of acceptance*. The circle of acceptance has radius R_{acc}, and when

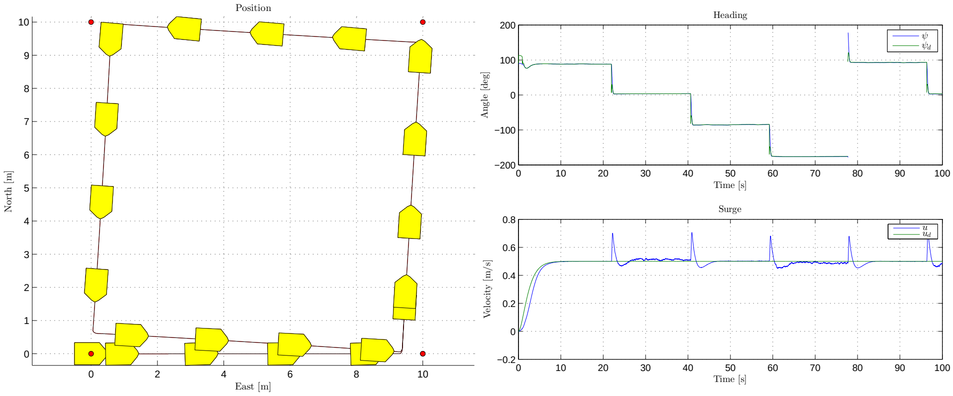
$$\sqrt{(x_i - x)^2 + (y_i - y)^2} \leq R_{\text{acc}}, \tag{4.18}$$

is fulfilled, the vehicle is to move against the next waypoint (x_{i+1}, y_{i+1}). For the simulations, the current is turned on, and is defined as in Section 3.4. The initial position vector for both vehicles in Scenario 1 and 2 is

$$\boldsymbol{\eta_0} = \begin{bmatrix} 0 & 0 & \frac{\pi}{2} \end{bmatrix}^T. \tag{4.19}$$

Parameter	**Value**
u_d	0.5 m/s
ω_n	0.8 rad/s
ζ	1.0
$K_{p,u}$	2.0
$K_{i,u}$	5.0
$K_{p,\psi}$	10.0
$K_{i,\psi}$	10.0
R_{acc}	0.5

Table 4.1: Heading on WP parameters for Neptunus.



(a) Position.

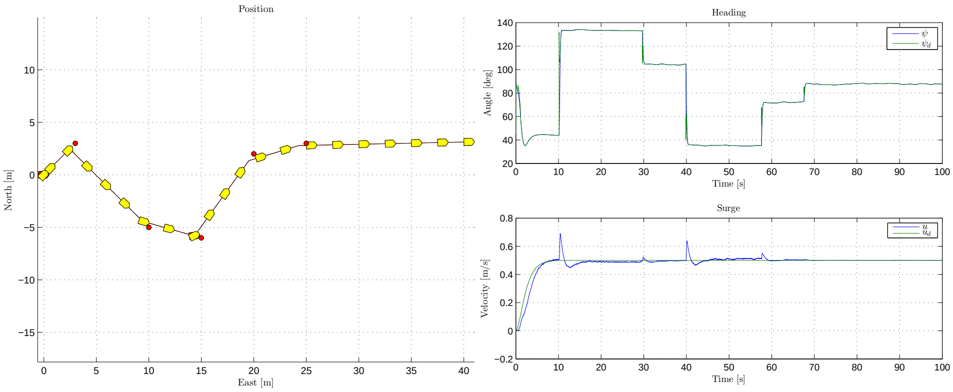
(b) Heading and velocity.

Figure 4.9: Neptunus, Scenario 1 heading on waypoints.

Neptunus

The position for Scenario 1 is seen in Figure 4.9a. The desired versus the measured velocity and heading are shown in Figure 4.9b. The same plot results are provided in Figure 4.10a and 4.10b for Scenario 2. Parameters related to the control system are provided in Table 4.1.

From the results of Scenario 1 and 2, it is seen that heading on waypoints provides satisfying performance. Neptunus is able to move between the waypoints, following both the desired heading, and also the desired velocity. It successfully changes the waypoint when it's within the circle of acceptance. For the squared path, Neptunus experiences some over rise in heading. This is probably due to numerical errors in the simulation, as wrap around algorithms are present to map the heading into (π, -π) Notice the small oscillations in the surge velocity, which occurs due to the ocean current. However, the integral effect removes this error.



(a) Position.

(b) Heading and velocity.

Figure 4.10: Neptunus, Scenario 2 heading on waypoints.

Parameter	**Value**
u_d	0.5 m/s
ω_n	0.8 rad/s
ζ	1.0
ω_b	0.4
$K_{p,u}$	10.0
$K_{i,u}$	1.0
$K_{p,\psi}$	26.0
$K_{i,\psi}$	1.6
R_{acc}	0.5

Table 4.2: Heading on WP parameters.

CSEI

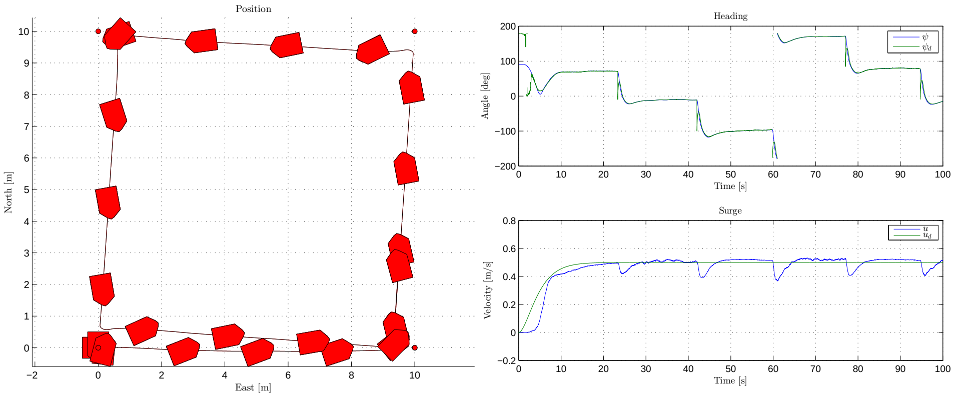
In this control algorithm, only the VSPs are enabled. The surge controller will directly control u_{VSP}, while the heading controller will control δ. Hence, the VSPs will work as a propeller with a rudder. This feature is quite common for conventional surface vessels. The parameters related to the control system are given in Table 4.2. The gains for the heading controller are tuned based on the results given in Appendix B.1, where K_p is tuned according to

$$K_p = \frac{T\omega_b^2}{0.41K}, \tag{4.20}$$

and K_i according to

$$K_i = \frac{\omega_b K_p}{6.4}. \tag{4.21}$$

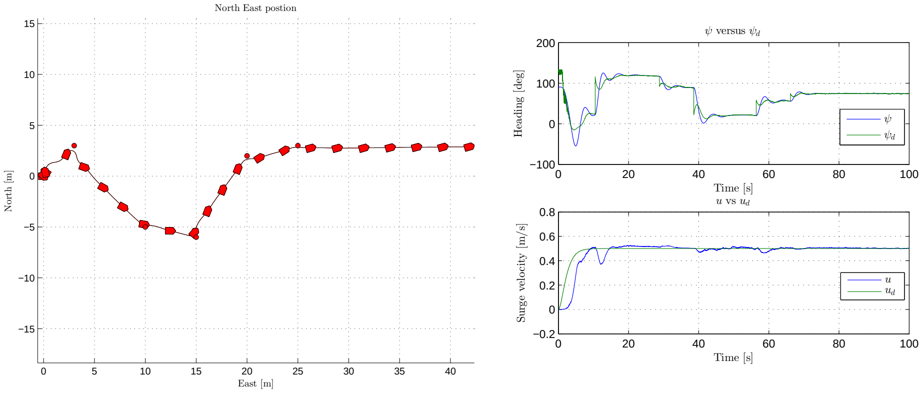
The results follow the same structure as for Neptunus. Scenario 1 is seen in Figure 4.11, while Scenario 2 can be investigated in Figure 4.12. As the results reveal, CSEI is capable of navigate through the waypoints. The transient in the heading response in the beginning of the scenarios are due to the inertia of CSEI.



(a) Position.

(b) Heading and velocity.

Figure 4.11: CSEI, Scenario 1 heading on waypoints.



(a) Pposition.

(b) Heading and velocity.

Figure 4.12: CSEI, Scenario 2 heading on waypoints.

Discussion

Two scenarios have been considered, and simulations performed for Neptunus and CSEI. From the results, it is concluded that the heading on waypoints algorithm gives satisfying performance. Both the vehicles are capable navigating through the waypoints. However, it is clear that Neptunus follows the desired heading with higher accuracy than CSEI. The main reason is that the VSPs are *transformed* into a rudder with propeller, in addition to geometrical considerations. A higher yaw moment is generated from the CSEI thruster than for Neptunus.

An advantage with this algorithm, is the rather simple structure. It has low computational requirements, and it is intuitive. On the other side, it is not a robust control design. The control algorithm is only considering one waypoint a time, and the only concern is to minimize the heading error, while holding a certain speed. It does not take into account the next waypoint before the vehicle is within the circle of acceptance. Consequently, a stuttering behavior around the change of waypoints is registered.

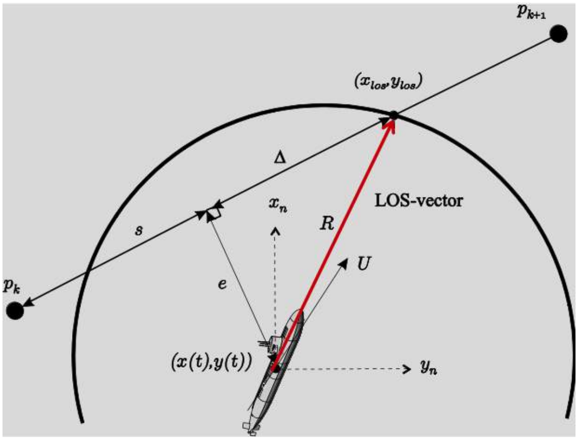


Figure 4.13: Lookahead based LOS. Courtesy of Fossen (2011).

4.5 Lookahead based line-of-sight steering

An other common path following controller is the *lookahead-based* LOS steering algorithm. Unlike the heading on waypoint controller, this control structure takes the next two waypoints into consideration. The objective is to make the vessel converge to the line between these two waypoints, see Figure 4.13.

Considering the linked path between the points p_{k+1} and p_k. The lookahead based algorithm forces the vehicle to converge onto the path. This is done by minimizing the *cross-track* error e. Furthermore, the aggressiveness of the converging can be tuned with a parameter called *lookahead distance,* Δ. Low Δ represents an aggressive behavior, while high Δ gives a slower convergence to the path. Consequently, if the distance between two points is long, and there is no need to converge fast, a high Δ is preferable. On the other side, for a short path, a lower Δ, and a more aggressive convergence is preferred. Breivik and Fossen (2009) divide the lookahead-based LOS steering into two assignments

$$\chi_d(e) = \chi_p + \chi_r(e), \tag{4.22}$$

where χ_p is the path-tangential angle, and it is given as

$$\chi_p = \alpha_i = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i), \tag{4.23}$$

and $\chi_r(e)$ is the velocity-path relative angle, and written as

$$\chi_r(e) = \text{atan}\left(\frac{-e}{\Delta}\right). \tag{4.24}$$

Using trigonometry considerations in Figure 4.13, the cross track error becomes

$$e = -(x - x_i)\sin(\alpha_i) + (y - y_i)\cos(\alpha_i), \tag{4.25}$$

where x, and y represent the current position of the vehicle. Furthermore, the desired heading ψ_d is defined as

$$\psi_d = \chi_d(e) - \beta_r, \tag{4.26}$$

where the sideslip angle is $\beta_r = \sin^{-1}\left(\frac{v_r}{U_r}\right)$. The error $\tilde{\psi}$ can now be minimized by the PI controller

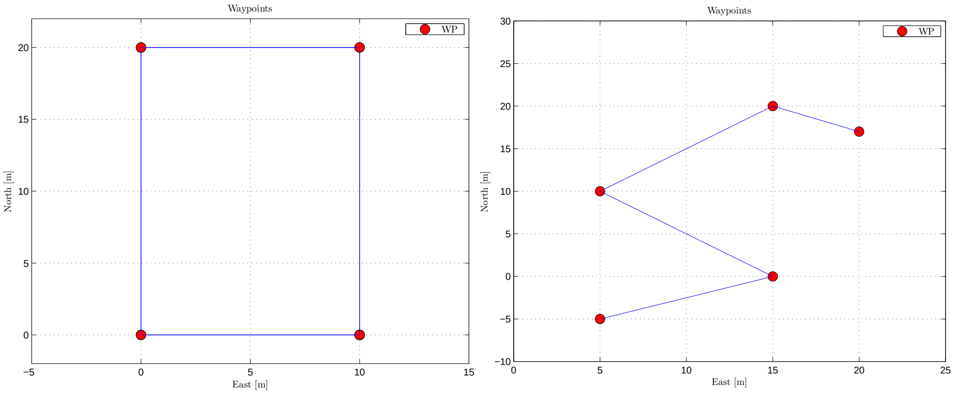
$$\tau_\psi = -K_p\tilde{\psi} - K_i\int_0^t \tilde{\psi}(\tau)d\tau. \tag{4.27}$$

To control the vehicle's speed to a desired speed, a PI controller is used

$$\tau_u = -K_p\tilde{u} - K_i\int_0^t \tilde{u}(\tau)d\tau. \tag{4.28}$$

4.5.1 Simulations

Two scenarios are considered with lookahead based LOS. These two paths can be viewed in Figure 4.14.



(a) Rectangular path.

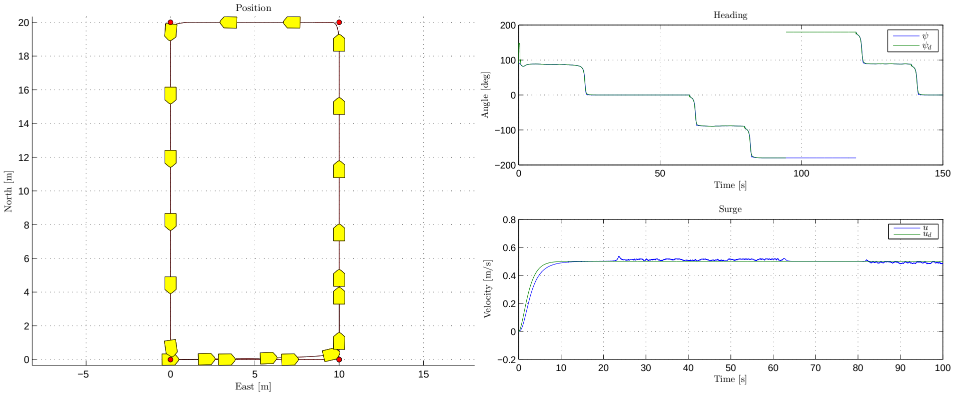
(b) Curved path.

Figure 4.14: Scenarios for lookahead-based LOS control algorithm.

To test the aggressiveness of the controller, two different values of Δ will be assessed. In all scenarios the initial vector for Neptunus and CSEI is set to $\boldsymbol{\eta}_0 = [0\ 0\ \frac{\pi}{2}]^T$. Moreover, the current is present with the same parameters as in Section 4.4.1. The duration of the simulations is $t = 150$ s, the and the circle of acceptance is set to $R_{\text{acc}} = 1$ m.

Parameter	Value
u_d	0.5 m/s
ω_n	0.8 rad/s
ζ	1.0
$K_{p,u}$	10.0
$K_{i,u}$	5.0
$K_{p,\psi}$	10.0
$K_{i,\psi}$	10.0
R_{acc}	1.0 m
Δ_1	0.1
Δ_2	5.0

Table 4.3: Lookahead-based LOS control algorithm parameters for Neptunus.



(a) Position.

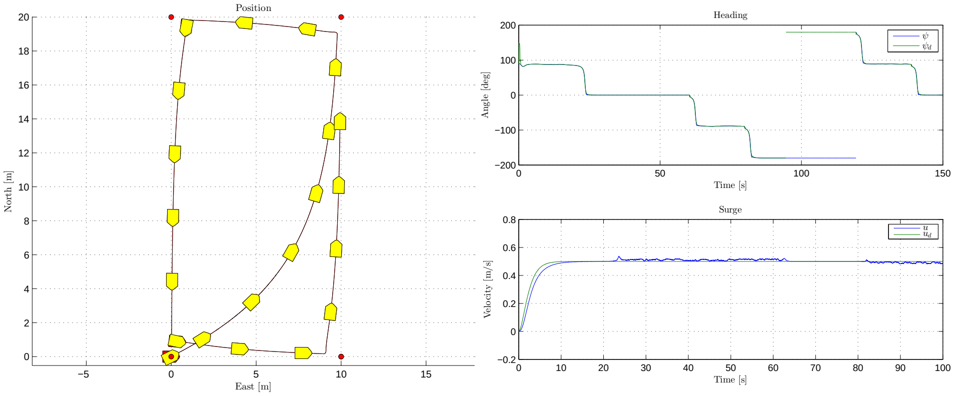
(b) Heading and velocity.

Figure 4.15: Neptunus, Scenario 1 for lookahead based LOS ($\Delta = 0.1$).

Neptunus

A summary of the parameters used in the control system are presented in Table 4.3. The response for Scenario 1, with the different lookahead distances can be seen in Figure 4.15 and 4.16. It is observed from the plots that Neptunus follows the path with a satisfying performance. It is especially worth mentioning the differences using the lookahead distances Δ_1 and Δ_2. For Δ_1, there is clearly aggressive convergence to the straight lines, as opposed to Δ_2, that brings a slower convergence. This difference can especially be noticed on the line between (0,10) and (20,10).

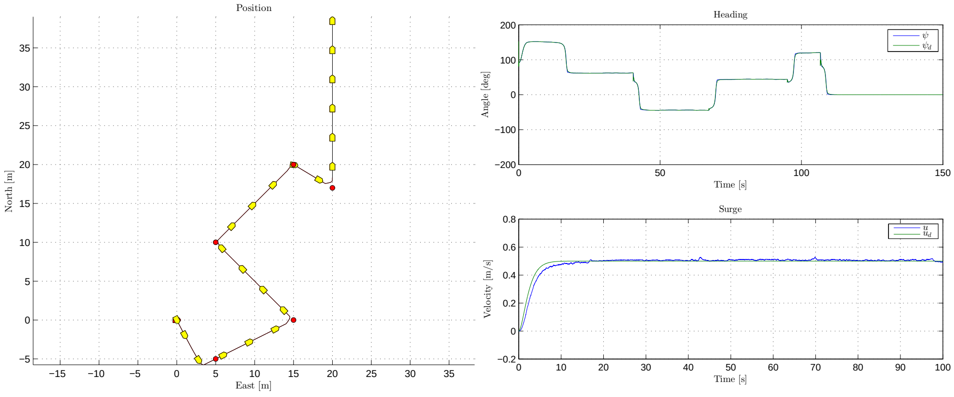
Moreover, the response for Scenario 2 is presented in Figure 4.17 and 4.18. Also here, Δ_1 and Δ_2 give different responses. However, after the first line, the difference is small, since the difference between the waypoints are relatively small, and the waypoint loca-



(a) Position.

(b) Heading and velocity.

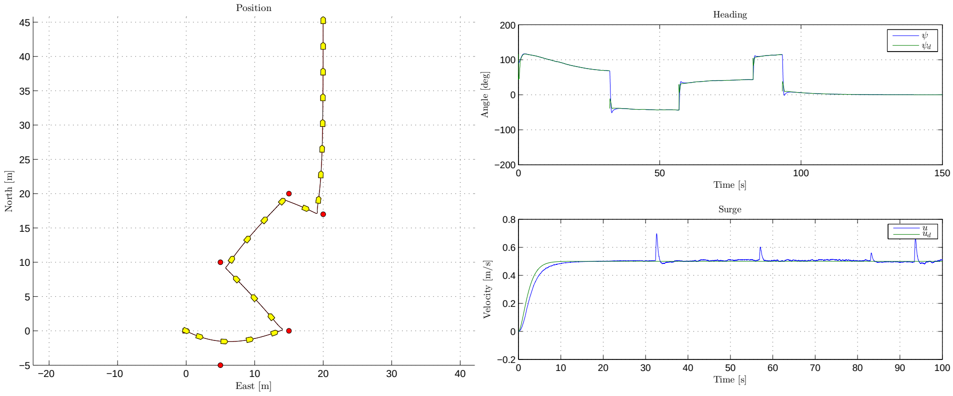
Figure 4.16: Neptunus, Scenario 1 for lookahead based LOS ($\Delta = 5.0$).



(a) Position.

(b) Heading and velocity.

Figure 4.17: Neptunus, Scenario 2 for lookahead based LOS ($\Delta = 0.1$).



(a) Position.

(b) Heading and velocity.

Figure 4.18: Neptunus, Scenario 2 for lookahead based LOS ($\Delta = 5.0$).

tion makes Neptunus converge fast, with low dependence of the aggressiveness. However, for the first line from (-5,6) to (0,15) a different aggressiveness is observed.

Parameter	**Value**
u_d	0.5 m/s
ω_n	0.2 rad/s
ζ	1.0
ω_b	0.4
$K_{p,u}$	25.0
$K_{i,u}$	4.0
$K_{p,\psi}$	26.0
$K_{i,\psi}$	1.6
R_{acc}	1.0 m
Δ_1	0.1
Δ_2	5.0

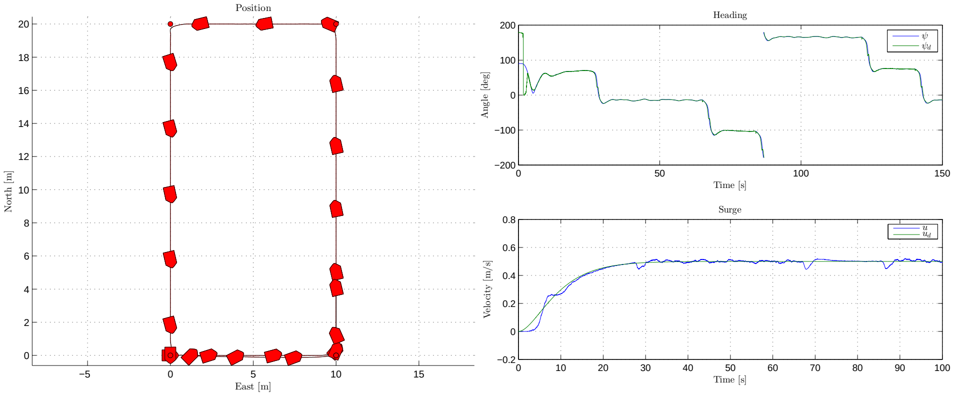
Table 4.4: Lookahead-based LOS control algorithm parameters for Neptunus.

CSEI

The same scenarios are conducted for CSEI. Parameters related to the control system are provided in Table 4.4. The controller gains in yaw have been tuned as in the heading on waypoints control algorithm. Also, in this control system, only the VSPs are active.

Scenario 1 with Δ_1 and Δ_2 as lookahead distance are presented in Figure 4.19 and 4.20, respectively. The rectangular path is followed with satisfying performance, where the heading and speed assignment is followed, even with the presence of current. Also, Δ_1 gives more aggressive convergence than for Δ_2. In the section between (0,10) and (20,10) it is possible to see that the behavior is more aggressive.

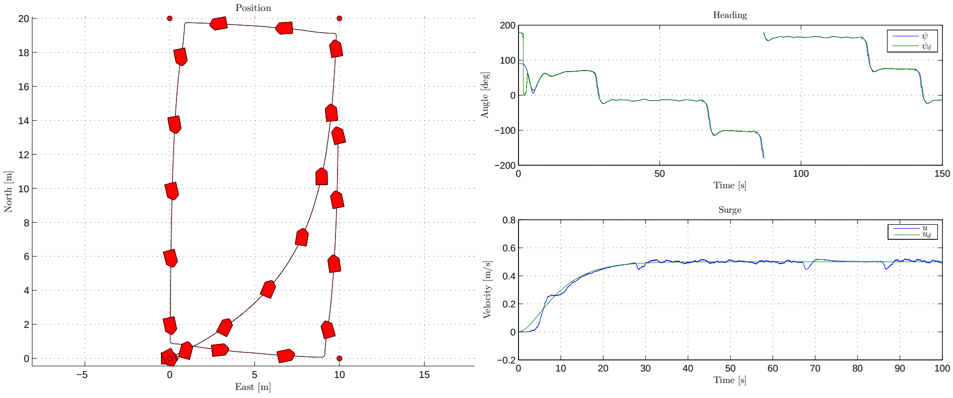
Scenario 2 is further presented in Figure 4.21 and 4.22, also here two values of Δ are presented. The same response as for Neptunus is observed.



(a) Position.

(b) Heading and velocity.

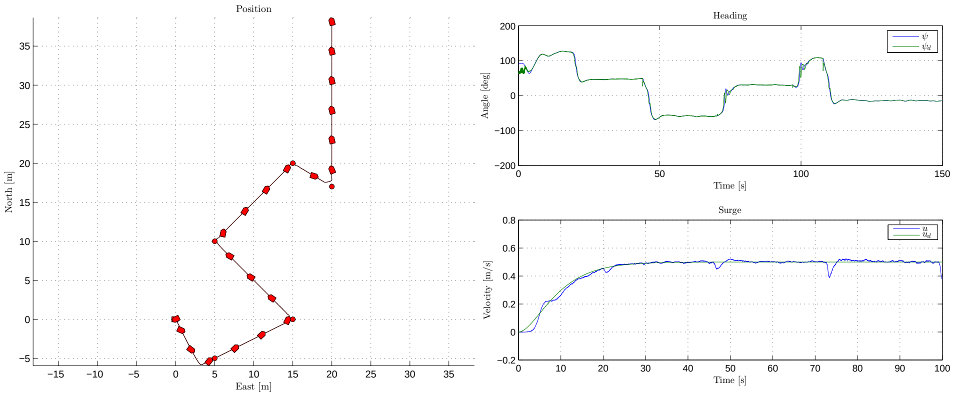
Figure 4.19: CSEI, Scenario 1 for lookahead based LOS ($\Delta = 0.1$).



(a) Position.

(b) Heading and velocity.

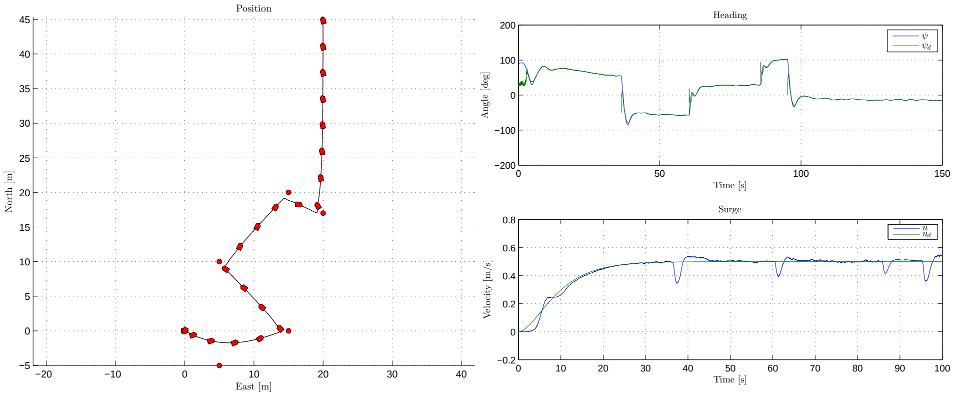
Figure 4.20: CSEI, Scenario 1 for lookahead based LOS ($\Delta = 5.0$).



(a) Position.

(b) Heading and velocity.

Figure 4.21: CSEI, Scenario 2 for lookahead based LOS (Δ = 0.1).



(a) Position.

(b) Heading and velocity.

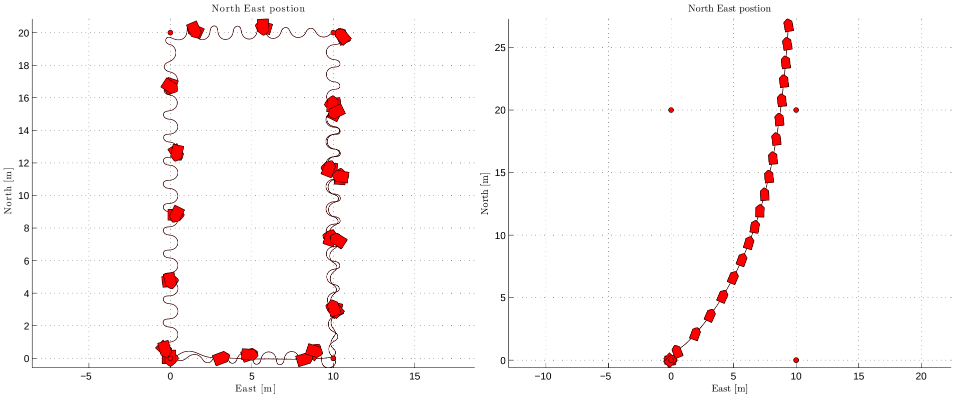
Figure 4.22: CSEI, Scenario 2 for lookahead based LOS (Δ = 5.0).

Discussion

Lookahead-based LOS provides a well working steering capability for Neptunus and CSEI. It is a more intelligent guidance system, and more predictive than the heading on waypoints steering. Since the lookahead-based LOS takes two waypoints into consideration, the performance will be smoother. An additional tuning parameter (Δ) is present, that will also tune how aggressive the steering algorithm will be. With proper tuning, the control algorithm reveals good performance, without oscillations.

Wrong tuning, or narrow set of waypoints can be crucial for the performance of the lookahead-based LOS control algorithm. A high lookahead distance can lead to a too slow convergence to the path. Consequently, the vehicle can miss the waypoint, and cause a drift off from the path.

On the other hand, a too aggressive behavior (small Δ) can lead to a unstable convergence, and instability. As a result, the vehicle's heading will oscillate on the path. These two cases are visualized in Figure 4.23.



(a) Low Δ.

(b) High Δ.

Figure 4.23: A too low and too high Δ, causing unstable motion.

4.6 Maneuvering control design

In the last path following control design, a desired path is parametrized, and the objective for the vehicle is to follow this path with a desired dynamic behavior. According to Skjetne (2005, Ch. 2), the introduction of this concept can be formulated into a *maneuvering problem*. Moreover, the maneuvering problem is consisting of two tasks, given in the prioritized order;

- **Geometric task**: Force the position $\boldsymbol{\eta}$ to converge to a desired parametrized path $\boldsymbol{\eta}_d(s)$, where s is the path parameter. Mathematically, this is written as

$$\lim_{t\to\infty} |\boldsymbol{\eta} - \boldsymbol{\eta}_d(s)| = 0.$$

- **Dynamic task**: Satisfy one or more of the assignments, given as

a time assignment

$$\lim_{t\to\infty} |s(t) - \tau(t)| = 0,$$

a speed assignment

$$\lim_{t\to\infty} |\dot{s}(t) - v_s(s(t), t)| = 0,$$

or an acceleration assignment

$$\lim_{t\to\infty} |\ddot{s}(t) - \alpha(\dot{s}(t), s(t), t)| = 0.$$

In this master thesis, the geometric task is to converge to a parametrized ellipsoid. This is the most prioritized task. The dynamic task is to converge to a desired speed along the path.

4.6.1 Ellipsoid path parameterization

The path parameter s is used to develop a path. Using the formula for an ellipsoid with center in (x_c, y_c) and radii (r_x, r_y), the parametrized path is given as

$$x_d(s) = x_c + r_x \cos(2\pi s), \quad s \in (0,1), \tag{4.29}$$

$$y_d(s) = y_c + r_y \sin(2\pi s), \quad s \in (0,1). \tag{4.30}$$

The operator can specify the center of the ellipsoid, and also the radii in x and y. In Figure 4.24, a parametrized ellipsoid is presented. It is noticed that when $s = 1$, the ellipsoid is complete, and the path parameter is reset. The desired heading along the

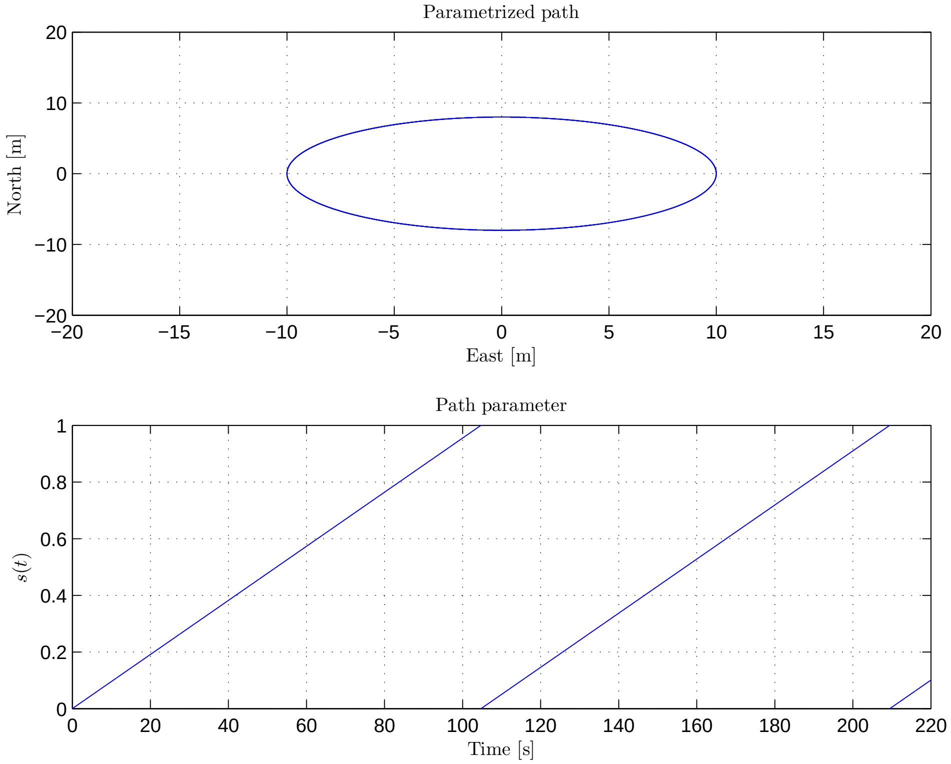


Figure 4.24: Ellipsoid path generated by a path parameter s, with radii (10,8). $s(t) = \frac{t}{105}$.

path is calculated to be

$$\psi_d(s) = \text{atan2}(y_d^s, x_d^s), \tag{4.31}$$

where (x_d^s, y_d^s) is the tangential vector along the path.

4.6.2 Speed assignment

The dynamic task is to move along the parametrized path with a constant speed u_{ref}. This means that $|\dot{\boldsymbol{p}}_d| = |u_{\text{ref}}|$ must be true, where $\boldsymbol{p}_d$ is the desired position in the North-East plane. Moreover, this can be written as

$$|\dot{\boldsymbol{p}}_d| = |\boldsymbol{p}_d^s \dot{s}| = \sqrt{x_d^s(s)^2 \dot{s}^2 + y_d^s(s)^2 \dot{s}^2} = \sqrt{x_d^s(s)^2 + y_d^s(s)^2}\,|\dot{s}| = |u_{\text{ref}}|. \tag{4.32}$$

With this relation, the speed assignment is stated as

$$\dot{s} \rightarrow v_s(s) := \frac{u_{ref}}{\sqrt{x_d^s(s)^2 + y_d^s(s)^2}}. \tag{4.33}$$

4.6.3 Maneuvering control design by backstepping

In order to obtain a control law for steering Neptunus and CSEI along a parametrized path with a desired speed, a *backstepping* control design is designed. The backstepping approach will follow the same procedure as Skjetne et al. (2004). The CPM of consideration is

$$\begin{aligned}
&\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \\
&\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{R}^T(\psi)\boldsymbol{b} = \boldsymbol{\tau},
\end{aligned} \tag{4.34}$$

where $\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A$ and $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu})$. The bias term $\boldsymbol{b}$ furthermore represents slowly-varying unmodelled terms, where current is the dominating effect. Thus, an integral action is needed to avoid steady state error.

The error states are defined as

$$\boldsymbol{z}_1 = \boldsymbol{R}^T(\psi)(\boldsymbol{\eta} - \boldsymbol{\eta}_d(s)) \tag{4.35}$$

$$\boldsymbol{z}_2 = \boldsymbol{\nu} - \boldsymbol{\alpha}, \tag{4.36}$$

where $\boldsymbol{\alpha}$ is a virtual controller, and is to be defined. To account the bias term, an integral action is implement as

$$\boldsymbol{\xi} = \int_0^t \boldsymbol{z}_2(\tau)d\tau. \tag{4.37}$$

Introducing integral action in the $\boldsymbol{z}_2$-state reveals satisfying stability properties, even for varying bias' (Skjetne and Fossen, 2004). This will be shown in the following sections.

Step 1 Time differentiation of $\boldsymbol{z}_1$ gives

$$\dot{\boldsymbol{z}}_1 = \dot{\boldsymbol{R}}^T(\psi)(\boldsymbol{\eta} - \boldsymbol{\eta}_d(s)) + \boldsymbol{R}^T(\psi)(\dot{\boldsymbol{\eta}} - \boldsymbol{\eta}_d^s(s)\dot{s}). \tag{4.38}$$

Using that $\dot{\boldsymbol{R}}(\psi) = r\boldsymbol{S}\boldsymbol{R}(\psi)$

$$\boldsymbol{S} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = -\boldsymbol{S}^T, \tag{4.39}$$

the dynamics become

$$\dot{\boldsymbol{z}}_1 = -r\boldsymbol{S}\boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{\alpha} - \boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s(s)\dot{s}. \tag{4.40}$$

Further, a *control Lyapunov function* (CLF) is chosen to be

$$V_1 = \frac{1}{2}\boldsymbol{z}_1^T\boldsymbol{z}_1. \tag{4.41}$$

By differentiating the CLF with respect to time

$$\dot{V}_1 = \boldsymbol{z}_1^T(-r\boldsymbol{S}\boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{\alpha} - \boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s(s)\dot{s}). \tag{4.42}$$

Noticing that $\boldsymbol{z}_1^T r\boldsymbol{S}\boldsymbol{z}_1 = 0$, a virtual control law $\boldsymbol{\alpha}$ can be selected:

$$\boldsymbol{\alpha} = -\boldsymbol{K}_p\boldsymbol{z}_1 + \boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s(s)\upsilon_s(s). \tag{4.43}$$

with the controller gain $\boldsymbol{K}_p = \boldsymbol{K}_p^T > 0$ to give *input-to-state-stability* (ISS) dynamics. Hence, the result of step 1 is

$$\dot{V}_1 = -\boldsymbol{z}_1^T\boldsymbol{K}_p\boldsymbol{z}_1 + \boldsymbol{z}_1^T\boldsymbol{z}_2 + \boldsymbol{z}_1^T\boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s(s)[\upsilon_s(s) - \dot{s}], \tag{4.44}$$

$$\dot{\boldsymbol{z}}_1 = -\boldsymbol{K}_p\boldsymbol{z}_1 - r\boldsymbol{S}\boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s(s)[\upsilon_s(s) - \dot{s}]. \tag{4.45}$$

To aim into the next step, $\boldsymbol{\alpha}$ is differentiated with respect to time

$$\dot{\boldsymbol{\alpha}} = \boldsymbol{\sigma} + \boldsymbol{\chi}\dot{s}, \tag{4.46}$$

where

$$\boldsymbol{\sigma} = -\boldsymbol{K}_p(\boldsymbol{\nu} - r\boldsymbol{S}\boldsymbol{z}_1) - r\boldsymbol{S}\boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s\upsilon_s(s) - \boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d\upsilon_s^t(s), \tag{4.47}$$

and

$$\boldsymbol{\chi} = \boldsymbol{K}_p\boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s + \boldsymbol{R}^T(\psi)\left(\boldsymbol{\eta}_d^{s^2}\upsilon_s(s) + \boldsymbol{\eta}_d^s\upsilon_s^s(s)\right). \tag{4.48}$$

Step 2 Next, the $\boldsymbol{z}_2$ state with the integral action is considered. By time differentiation

$$\begin{aligned}\dot{\boldsymbol{\xi}} &= \boldsymbol{z}_2,\\ \dot{\boldsymbol{z}}_2 &= \dot{\boldsymbol{\nu}} - \dot{\boldsymbol{\alpha}}.\end{aligned} \tag{4.49}$$

Applying the CLF

$$V_2 = V_1 + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{K}_b\boldsymbol{\xi} + \frac{1}{2}\boldsymbol{z}_2^T\boldsymbol{M}\boldsymbol{z}_2, \tag{4.50}$$

and by time differentiation

$$\begin{aligned}\dot{V}_2 = &-\boldsymbol{z}_1^T\boldsymbol{K}_p\boldsymbol{z}_1 + \boldsymbol{z}_1^T\boldsymbol{z}_2 + \boldsymbol{z}_1^T\boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s(s)[\upsilon_s(s) - \dot{s}] + \boldsymbol{\xi}^T\boldsymbol{K}_b\boldsymbol{z}_2\\ &+\boldsymbol{z}_2^T\left(\boldsymbol{\tau} - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{R}^T(\psi)\boldsymbol{b} - \boldsymbol{M}\boldsymbol{\sigma} - \boldsymbol{M}\boldsymbol{\chi}\dot{s}\right).\end{aligned} \tag{4.51}$$

Now, the control law $\boldsymbol{\tau}$ can be established in order to obtain satisfying stability properties. Assigning the control law as

$$\boldsymbol{\tau} = -\boldsymbol{z}_1 + \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{M}\boldsymbol{\sigma} + \boldsymbol{M}\boldsymbol{\chi}\upsilon_s(s) - \boldsymbol{K}_b\boldsymbol{\xi}, \tag{4.52}$$

gives

$$\dot{V}_2 = -\boldsymbol{z}_1^T\boldsymbol{K}_p\boldsymbol{z}_1 - \boldsymbol{z}_2^T\boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{\rho}[\upsilon_s(s) - \dot{s}] + \boldsymbol{R}^T(\psi)\boldsymbol{b}, \tag{4.53}$$

and

$$\boldsymbol{M}\dot{\boldsymbol{z}}_2 = -\boldsymbol{z}_1 - \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{R}^T(\psi)\boldsymbol{b} - \boldsymbol{K}_b\boldsymbol{\xi} + \boldsymbol{M}\boldsymbol{\chi}[\upsilon_s(s) - \dot{s}]. \tag{4.54}$$

Here, $\boldsymbol{\rho}$ is

$$\boldsymbol{\rho} = \boldsymbol{z}_1^T\boldsymbol{R}^T(\psi)\boldsymbol{\eta}_d^s + \boldsymbol{z}_2^T\boldsymbol{M}\boldsymbol{\chi}. \tag{4.55}$$

By solving the speed assignment, the equilibrium $(\boldsymbol{z}_1,\boldsymbol{\xi},\boldsymbol{M}\boldsymbol{z}_2) = 0$ is *uniformly globally asymptotically stable* (UGAS) for $b = 0$ (Skjetne and Fossen, 2004). However, a transformation of $\boldsymbol{\xi}$ to $\tilde{\boldsymbol{\xi}}$ such that $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi} - \boldsymbol{K}_b^{-1}\boldsymbol{b}$, gives the closed loop system

$$\begin{aligned}
\dot{\boldsymbol{z}}_1 &= -\boldsymbol{K}_p\boldsymbol{z}_1 - r\boldsymbol{S}\boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{R}(\psi)^T\boldsymbol{\eta}_d^s(s)[\upsilon_s(s) - \dot{s}] \\
\dot{\tilde{\boldsymbol{\xi}}} &= \boldsymbol{z}_2 \\
\boldsymbol{M}\dot{\boldsymbol{z}}_2 &= -\boldsymbol{z}_1 - \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{R}^T(\psi)\boldsymbol{b} - \boldsymbol{K}_b\tilde{\boldsymbol{\xi}} + \boldsymbol{M}\boldsymbol{\chi}[\upsilon_s(s) - \dot{s}].
\end{aligned} \tag{4.56}$$

Again, following the argumentation from Skjetne and Fossen, the equilibrium $(\boldsymbol{z}_1,\tilde{\boldsymbol{\xi}},\boldsymbol{M}\boldsymbol{z}_2) = 0$ is UGAS, even for $\boldsymbol{b} \neq 0$. This conclusion follows from *Matrosov's theorem.* The theorem is stated and the above conclusion is proved in Appendix B.2.

Solving the speed assignment $\dot{s} \rightarrow \upsilon_s(s)$, several methods are proposed:

- tracking update law,
- gradient update law,
- modified gradient update law,
- and filtered gradient update law.

In this master thesis, only tracking and modified gradient update laws are considered Skjetne (2005, Ch. 3).

Tracking update law is defined by setting

$$\dot{s} = \upsilon_s(s), \tag{4.57}$$

which directly solves the dynamic task. It is called a tracking update law due to the tracking parameterization that arises. A drawback with this update law is that path generation occurs independently of the position of the vehicle. Thus, the reference leaves the vehicle, and the dynamic task is not fulfilled.

Modified gradient update law is defined by assigning

$$\dot{s} = \upsilon_s(s) - \frac{\mu V^s(x,s)}{|\boldsymbol{\eta}_d^s|}, \quad \mu \geq 0, \tag{4.58}$$

which solves the dynamic task as $\boldsymbol{\eta}(t) \to \boldsymbol{\eta}_d(s(t))$. This update law takes feedback from the Lyapunov gradient, and takes the error between the path and the vehicle into consideration. This update law makes the path generation depending on the position of the vehicle. The term modified is due to the inclusion of the desired position gradient norm $|\boldsymbol{\eta}_d^s|$.

In correspondence with the maneuvering control design, the speed assignment derivatives with regards to s and t are

$$\begin{aligned} \upsilon_s^s(s) &= \frac{-\left(x_d^s(s)x_d^{s^2}(s) + y_d^s(s)y_d^{s^2}(s)\right)}{\left(x_d^s(s)^2 + y_d^s(s)^2\right)^{\frac{3}{2}}} u_{\text{ref}} \\ \upsilon_s^t(s) &= 0. \end{aligned} \tag{4.59}$$

4.6.4 Simulations

To test the performance of the maneuvering controller, a scenario with tracking update law and modified gradient update law will be conducted, both for Neptunus and for CSEI. The path parameters are provided in Table 4.5. The initial position for Neptunus and CSEI in the simulations is $\boldsymbol{\eta}_0 = [0\ 0\ 0]^T$. Again current is given as a slowly varying Gauss-Markov process. The current parameters are given in Section 3.4.

Parameter	**Value**
r_x	10.0 m
r_y	10.0 m
x_c	0.0 m
y_c	0.0 m

Table 4.5: Path parameters for the ellipsoid.

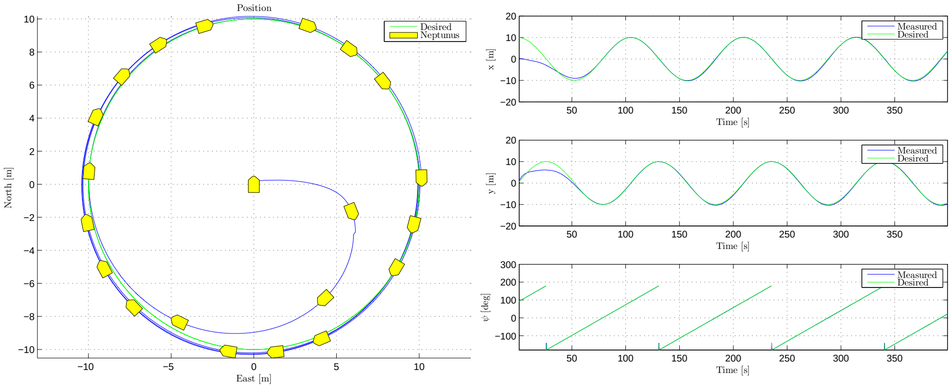
	Tracking update law	**Gradient update law**
$\boldsymbol{K}_p$	diag(10,1,10)	diag(10,1,5)
$\boldsymbol{K}_d$	diag(1,0.2,10)	diag(1,0.2,30)
$\boldsymbol{K}_b$	diag(0.001,0.1,0.001)	diag(0.01,0.1,0.01)
u_d	0.6 m/s	0.6 m/s
μ	0	0.0001

Table 4.6: Controller gains in maneuvering control system for Neptunus.

Neptunus

Parameters related to the control system are presented in Table 4.6. The scenarios reveal passable results. With the chosen gains, the ROV is capable of following the ellipsoid path. The geometric task from the tracking update law can be seen in Figure 4.25. It concludes that Neptunus converges to the path with the desired heading within reasonable time. A small cross-track error occurs since Neptunus is underactuated, and can not produce thrust in lateral direction. The speed assignment, which is the secondary task, is represented in Figure 4.26. It is verified that Neptunus maintains the desired speed along the path, with small deviations.

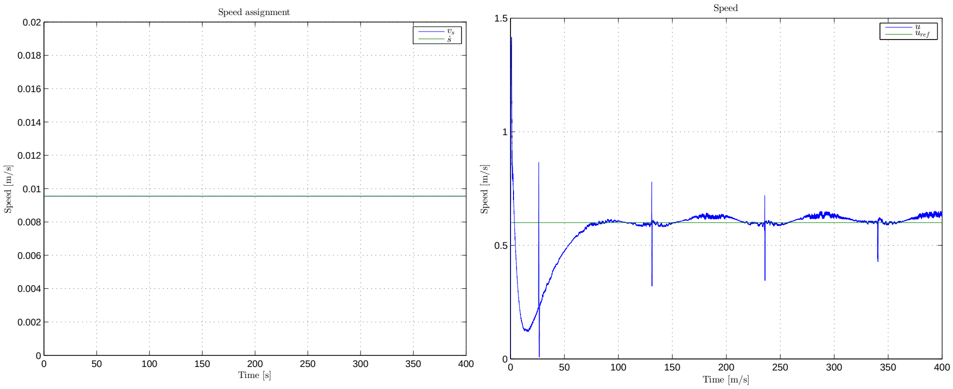
Moreover, the results from the modified gradient update scenario is represented in Figures 4.27 and 4.28. Even though it completes the circular motion, the gradient update law is proven to be less accurate than tracking update. This desire to create a lateral force influences the yaw moment, and consequently a smaller and smaller circular motion is induced. This also influences the dynamic task, resulting in an error in the speed assignment.



(a) Position.

(b) Geometric task.

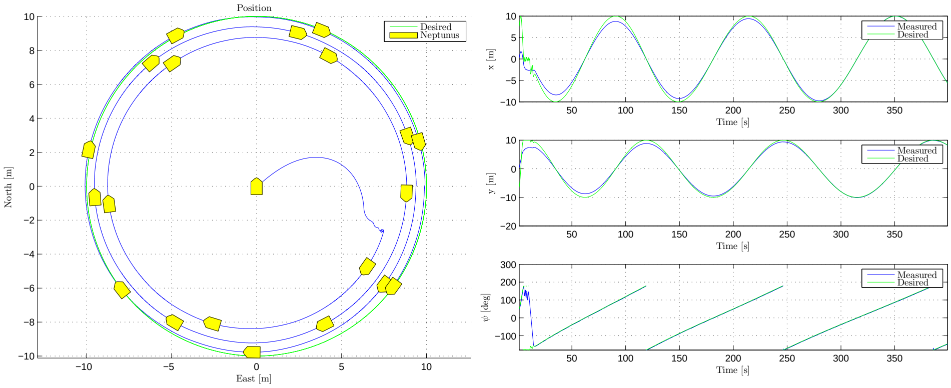
Figure 4.25: Neptunus, tracking update law path following.



(a) Speed assignment.

(b) Speed

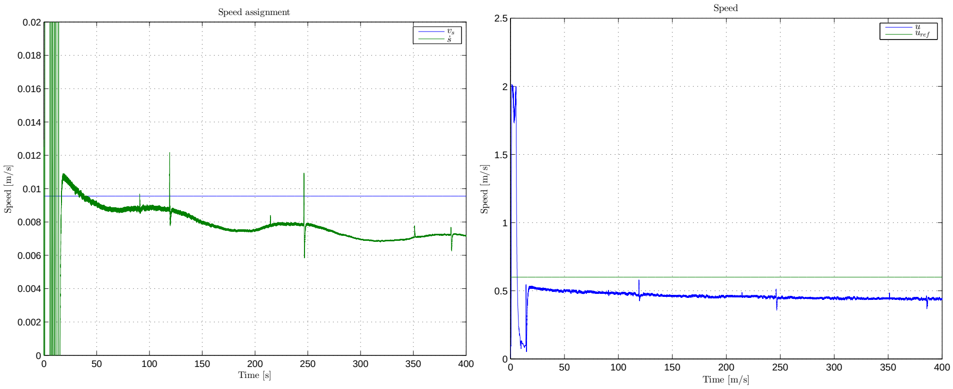
Figure 4.26: Neptunus, tracking update law. Dynamic performance.



(a) Position

(b) Geometric task.

Figure 4.27: Neptunus, modified gradient update law path following.



(a) Speed assignment.

(b) Speed

Figure 4.28: Neptunus, modified gradient update law. Dynamic performance.

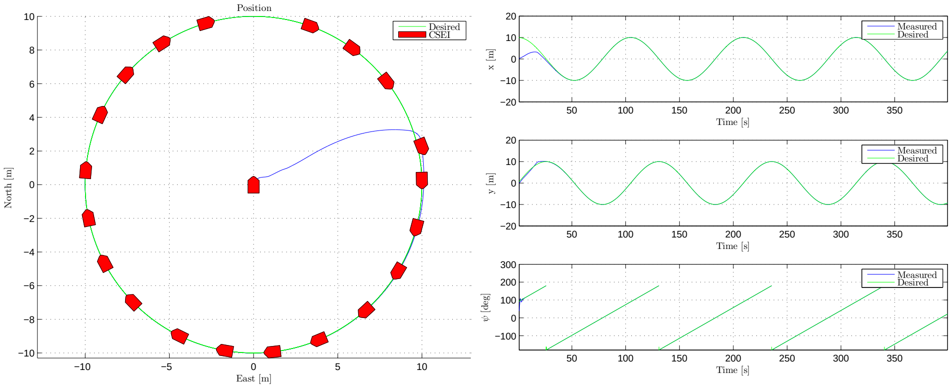
	Tracking update law	**Gradient update law**
$\boldsymbol{K}_p$	diag(0.1,1,50)	diag(0.1,1,50)
$\boldsymbol{K}_d$	diag(2000,3000,1000)	diag(2000,3000,2000)
$\boldsymbol{K}_b$	diag(0.001,0.1,0.001)	diag(0.001,0.1,0.001)
u_d	0.6 m/s	0.6 m/s
μ	0	0.001

Table 4.7: Controller gains in maneuvering control system for CSEI.

CSEI - Full actuation

The control system parameters are listed in Table 4.7. As for Neptunus, a tracking update as well as a modified gradient update law are considered. Since the backstepping controller is based on the non-linear *dynamic positioning* (DP) model, full actuation for CSEI is assumed. The geometric and dynamic task for the tracking update law is presented in Figures 4.29 and 4.30. Accurate tracking performance is concluded, and the desired speed along the path is followed satisfactorily. The geometric convergence is the primary task, which causes the high speed in the beginning. The oscillations in the speed are mainly caused by the current. The implemented integral action successfully removes the steady state error.

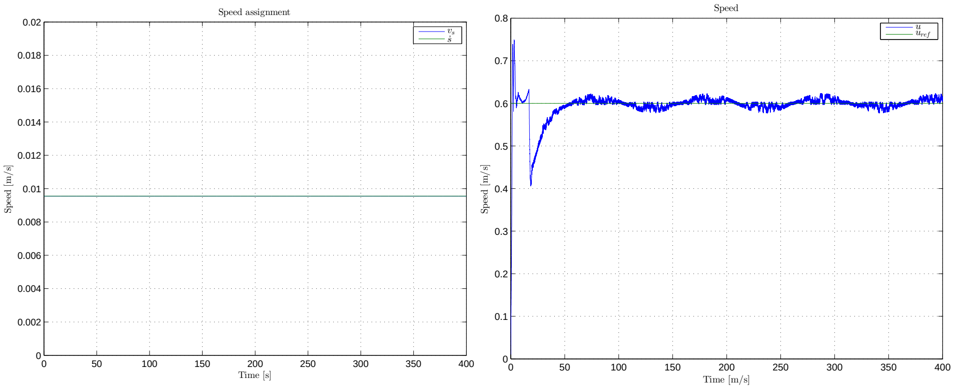
The results for the modified gradient update law are represented in Figure 4.31 and 4.32. Satisfying tracking is conducted along the path, with the commanded speed. In contrast to the tracking update law, the path parameterization takes the current position of CSEI into consideration. It makes the convergence to the path occur at a desirable location, and waits for CSEI to converge before the parameterization continues. Hence, CSEI enters the path at an other location than in the previous case. This can be validated by the $\dot{s}$-transients in Figure 4.32a.



(a) Position.

(b) Geometric task.

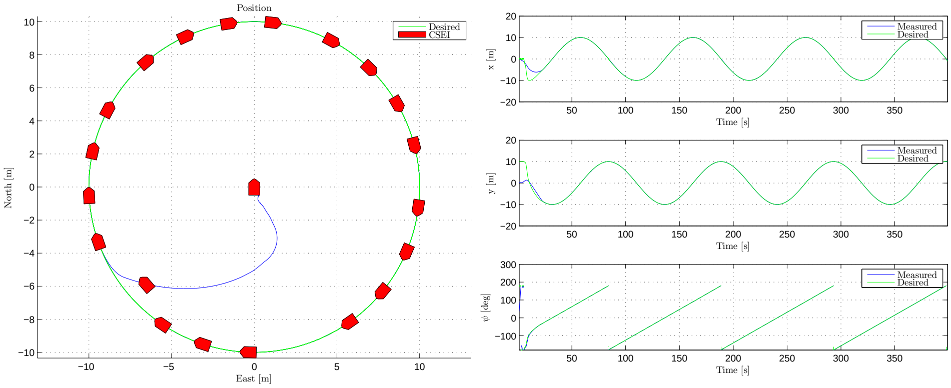
Figure 4.29: CSEI, tracking update law path following



(a) Speed assignment.

(b) Speed.

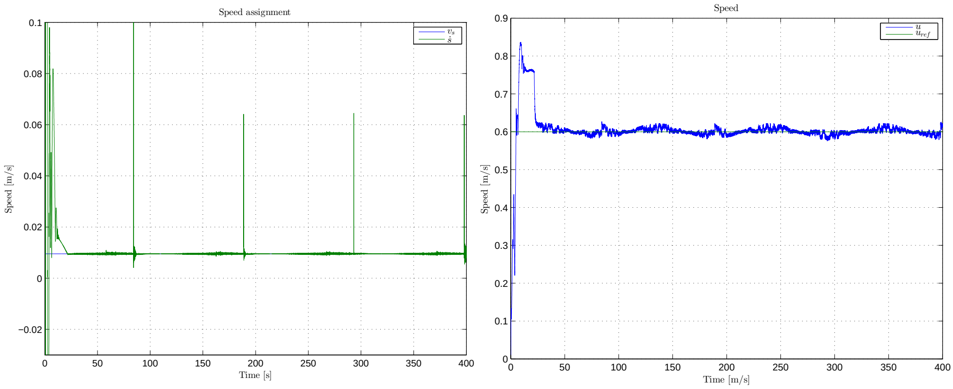
Figure 4.30: CSEI, tracking update law. Dynamic performance.



(a) Position.

(b) Geometric task.

Figure 4.31: CSEI, modified gradient update law path following.



(a) Speed assignment.

(b) Speed.

Figure 4.32: CSEI, modified gradient update law. Dynamic performance.

	Tracking update law	**Gradient update law**
$\boldsymbol{K}_p$	diag(1,1,15)	diag(1,1,15)
$\boldsymbol{K}_d$	diag(2000,2,200)	diag(2000,2,200)
$\boldsymbol{K}_b$	diag(0.01,0,0.001)	diag(0.01,0,0.001)
u_d	0.6 m/s	0.6 m/s
μ	0	0.0001

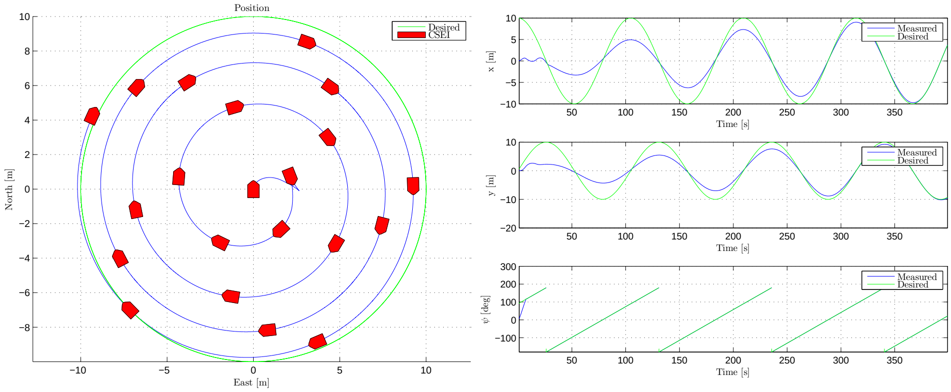
Table 4.8: Controller gains in maneuvering control system for CSEI.

CSEI - underactuated

For the case where the VSPs are mapped into a rudder and a propeller and the bow thruster disabled, the controller gains are as in Table 4.8.

The control forces in X and N are now assigned to u^*_{VSP} and δ. For the tracking update law, the geometric task can be seen in Figure 4.33. It is clear from the result that CSEI is not following the path accurately. From the North-East visualization, CSEI travels in circles, but does not converge to the path. However, the desired heading is followed. By looking at the dynamic task in Figure 4.34, the speed of the vessel is increasing, even after the desired speed is reached. Consequently, CSEI will not converge to the path using this control algorithm consideration.

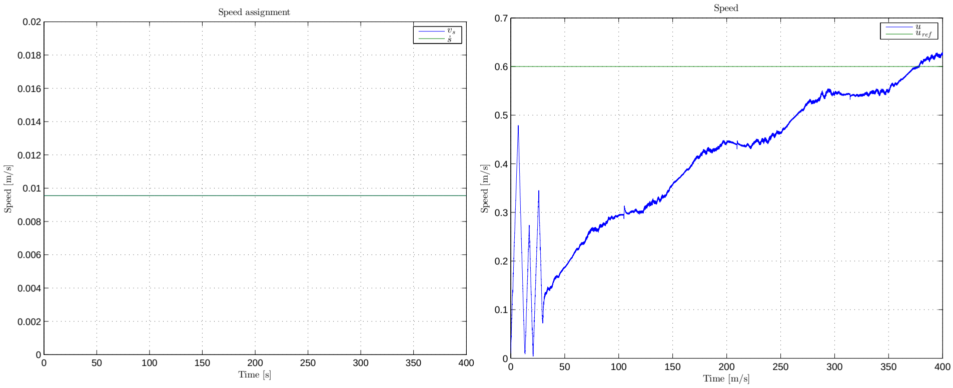
Considering the modified gradient update law scenario, the same tendencies are registered. In the underactuated case, CSEI is not capable of follow the desired path (see Figure 4.35). The modified gradient update law does not contribute to make the vessel converge. It has been tried to increase μ, in order to weight the gradient update law higher. However, this increased in a unstable update law. The dynamic response is presented in Figure 4.36.



(a) Position.

(b) Geometric task.

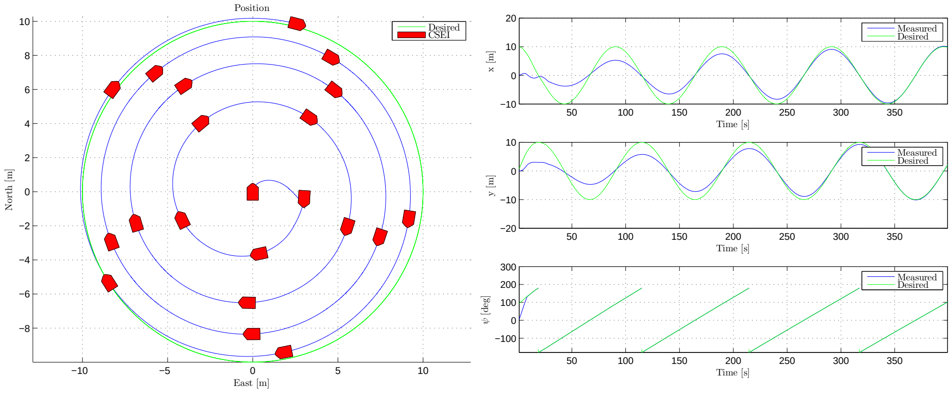
Figure 4.33: Virtually ruddered CSEI tracking update law path following.



(a) Speed assignment.

(b) Speed.

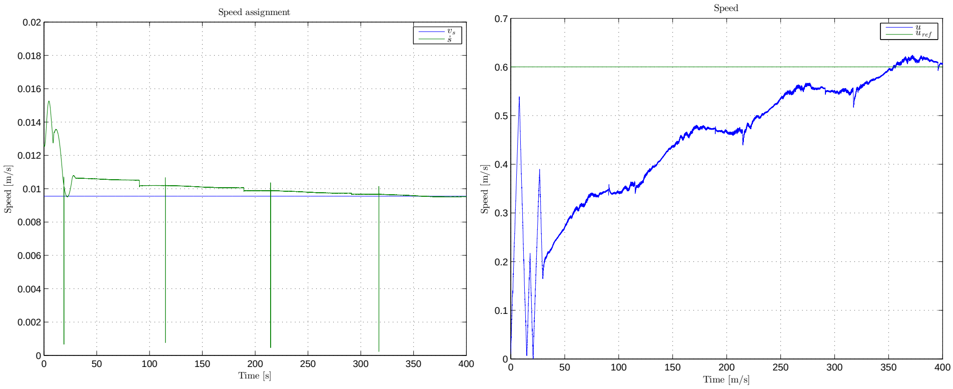
Figure 4.34: Virtually ruddered CSEI tracking update law. Dynamic performance.



(a) Position.

(b) Geometric task.

Figure 4.35: Virtually ruddered CSEI, modified gradient update law path following.



(a) Speed assignment.

(b) Speed.

Figure 4.36: Virtually ruddered CSEI, modified gradient update law. Dynamic performance.

	Tracking update law with LOS
$\boldsymbol{K}_p$	diag(10,1,10)
$\boldsymbol{K}_d$	diag(1,0.2,10)
$\boldsymbol{K}_b$	diag(0.001,0.1,0.001)
u_d	0.6 m/s
μ	0
Δ	5

Table 4.9: Controller gains in maneuvering control system for Neptunus.

4.7 Maneuvering-based line of sight

Recall the results from Section 4.6.3. Both for Neptunus and the underactuated case, perfect tracking is not obtained. To add an extra functionality to the maneuvering based control design, a LOS-algorithm is added. This is done by expanding the desired heading ψ_d to

$$\psi_d = \text{atan2}\left(y_d^s(s), x_d^s(s)\right) + \text{atan}\left(\frac{-e}{\Delta}\right), \tag{4.60}$$

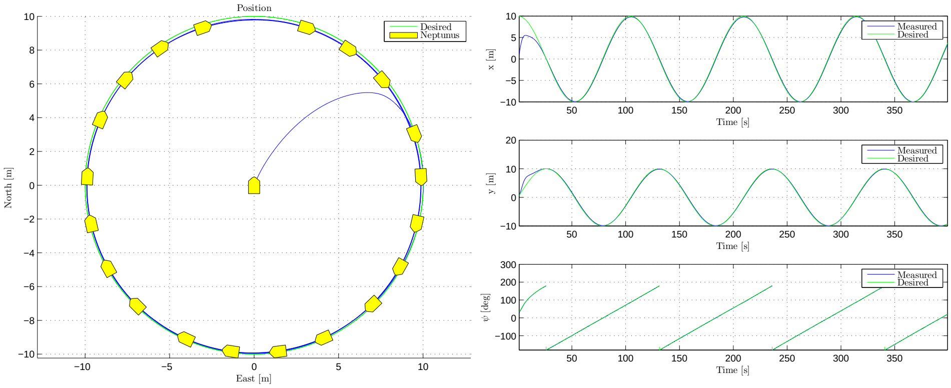
where e is the cross track error, and Δ is the lookahead distance (see Section 4.5). The control law renders the same result as for the maneuvering by backstepping design.

4.7.1 Simulations

During the simulations, only the tracking update law will be considered.

Neptunus

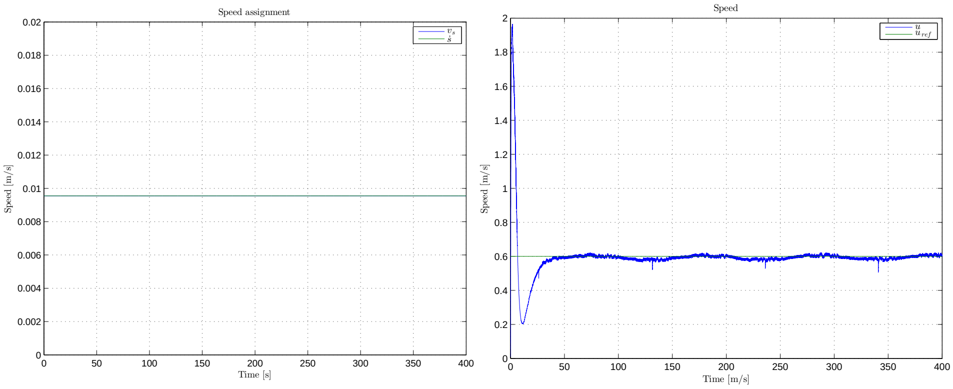
The parameters for Neptunus are provided in Table 4.9. The geometric task in Figure 4.37 reveals an accurate path following, with close to zero error. With the extra controllable parameter Δ, the error in x and y, which is present in the normal backstepping control algorithm, is removed. Further, the dynamic task in Figure 4.38 is followed identically, due to the tracking update law. However, the speed of Neptunus is maintained around the desired speed. The small oscillations are due to the current.



(a) Position.

(b) Geometric task.

Figure 4.37: Neptunus tracking update law with LOS.



(a) Speed assignment.

(b) Speed.

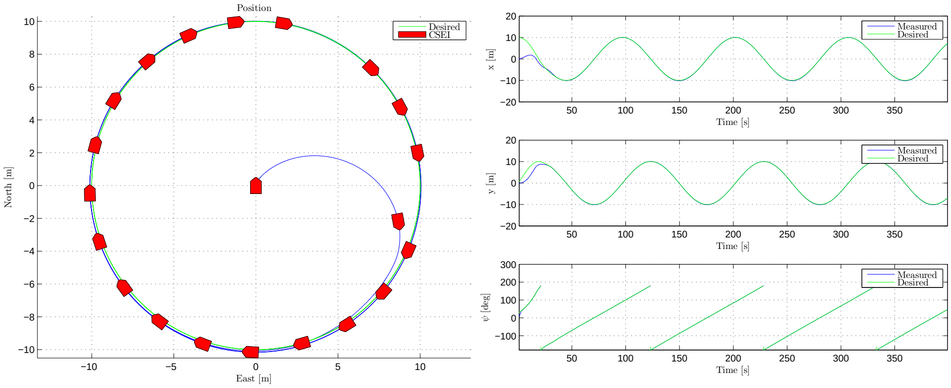
Figure 4.38: Neptunus, tracking update law with LOS. Dynamic performance.

	Tracking update law with LOS
$\boldsymbol{K}_p$	diag(1,1,15)
$\boldsymbol{K}_d$	diag(2000,2,200)
$\boldsymbol{K}_b$	diag(0.01,0.1,0.001)
u_d	0.6 m/s
μ	0
Δ	5

Table 4.10: Controller gains in maneuvering control system for CSEI.

CSEI - underactuated

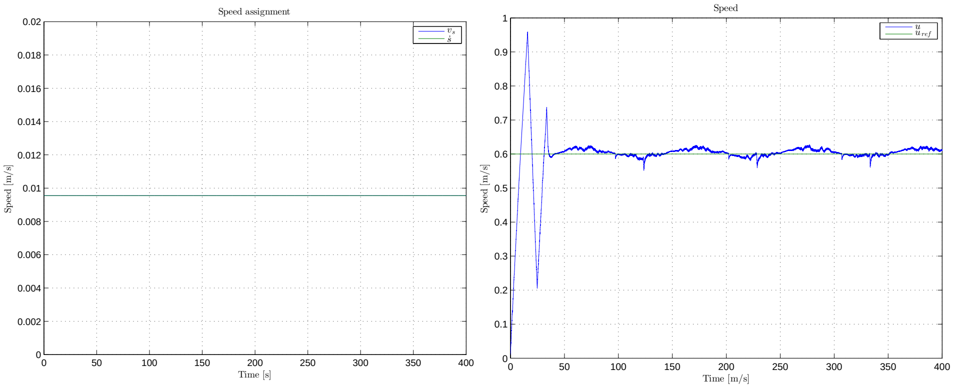
The control system parameters are given in Table 4.10. A better path following performance is obtained with the use of a LOS implemented to the maneuvering control system when the bow thruster is disabled. This can be verified by the geometric task in Figure 4.39. Small deviations are registered in the beginning, but when CSEI has converged to the path, it stays on the path. Moreover, the dynamic task is presented in Figure 4.40. Even though some transients are present in the start, the speed of CSEI follows the desired after CSEI has converged to the path.



(a) Position.

(b) Geometric task.

Figure 4.39: Virtually ruddered CSEI, tracking update law with LOS.



(a) Speed assignment.

(b) Speed.

Figure 4.40: Virtually ruddered CSEI, tracking update law with LOS. Dynamic performance.

4.7.2 Discussion

Two maneuvering based control algorithms have been established for Neptunus and CSEI. Normal maneuvering by backstepping revealed adequate results for Neptunus. However, some cross-track error is present, due to the bad ability to actuate in sway. For CSEI, satisfying results are registered in the case of full actuation. This comes as no surprise, since the backstepping controller is based on the DP model with full state actuation. On the other side, no x-y tracking is achieved in the case when the bow thruster is turned off. It performs circle motion, but with no convergence to the path.

Neptunus is more capable to follow the path in this control algorithm than CSEI. The reason is that a small sway component is generated due to the thruster angles, in addition to its geometrical shape.

A maneuvering based LOS is considered to make a underactuated vehicle converge to the path. The results revealed that increased tracking properties are achieved when cross-track error regulation is implemented. The obtained result is especially improved for CSEI.

Chapter 5

Remote control

Remote control enables the operator to be at a different location than the system of consideration. For instance, an underwater vehicle operating in the arctic can be monitored and controlled from a base station in Trondheim.

Bindingsbø (2013) states that unmanned installations can lead to efficient and reliable operations. With remote control and monitoring, people are removed from hazardous areas, thus increasing the personnel safety. Furthermore, unmanned designs lead to lower *capital expenditure* (CAPEX) and *operational expenditure* (OPEX) for the operators.

Remote control systems vary in complexity, availability, robustness, and price. However, the principle is in many cases the same: I/O signals wirelessly transmitted between the vehicle and the operator. For a underwater vehicle, it may involve problems due to the reflection of radio signals under water (Kunz et al., 2008).

This chapter covers communication implementation between a tablet and CSEI. With an adequate *user interface* (UI), the vessels performance is both monitored and controlled from a remote location. The application will utilize the software and hardware existing at the MC lab.

5.1 National Instruments applications

5.1.1 LabView, VeriStand, and cRIO

In order to operate the model vehicles in the MC-lab, the hardware and software provided from National Instruments are used. The input and output from the vessel is processed by a cRIO, located on board.

In addition, the simulation model of the vessel with a motion control system is also loaded onto the cRIO. The cRIO is a real time computer, able to process data at a rate up to 1.33 GHz, depending on the cRIO type. Consequently, it is a proper alternative for motion control and data processing. The data from the cRIO is transmitted via Ethernet or WiFi to a host computer (NTNU, 2015).

Via the software programs LabView and VeriStand, the host computer can monitor and control the behavior of the vessel. LabView is a modeling design environment, where mathematical models of the vessels, controllers, guidance, observers and other desired features are developed. The routines and graphical interfaces are created by using *Virtual Instruments* (VIs). The VIs consist of a block-diagram, a front panel, and a connector panel. The front panel represents the graphical interface, created by blocks and signal routing in the block diagram. The connector panel enables communication with other VIs[1].

To configure the real time system, VeriStand is used. VeriStand creates a real time engine based on the modules designed in LabView, which is used to execute commands related to monitoring and control of the physical system.

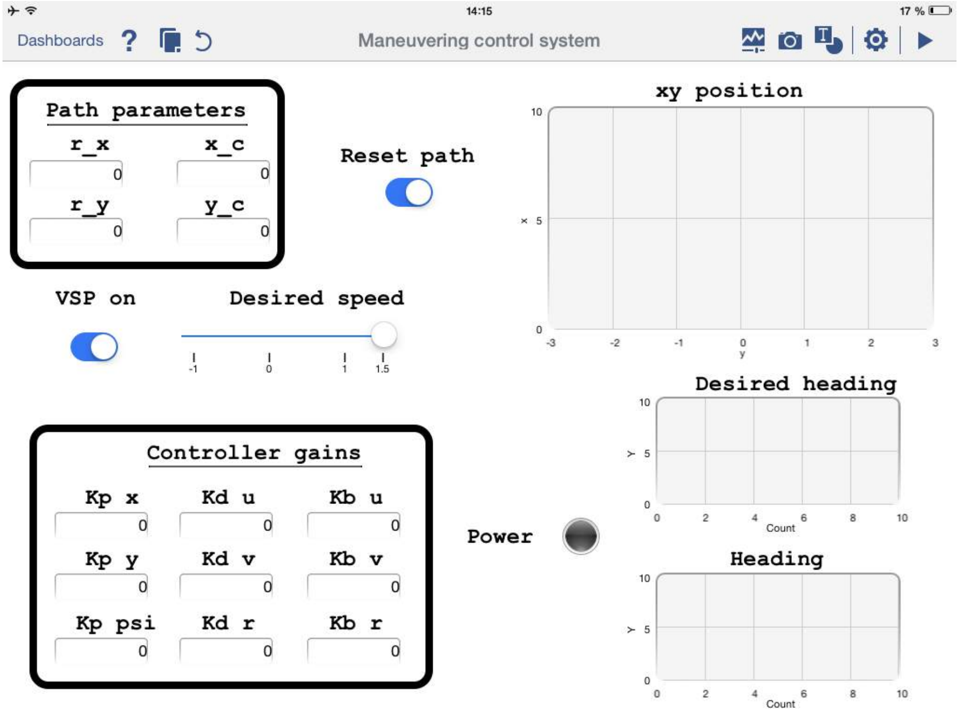
National Instruments has provided an expansion, where it is possible to build MATLAB/Simulink models for use in VeriStand.

5.1.2 Data Dashboard

National Instrument's *Data Dashboard*, is an extension of LabView to handheld devices, such as Android phones and tablets, as well as Apple's iPad [2]. Unfortunately, Data Dashboard is currently not supported on iPhone. The application lets the user create custom user interfaces, consisting of typical LabView displays, such as indicators, charts and gauges. Monitoring and control of a physical system, or a model located on the host computer, can be done via the phone or the tablet. An example of a dashboard can be seen in Figure 5.1.

[1] See http://www.ni.com/white-paper/4752/en/ for more information

[2] See information at ni.com/tutorial/13757/en



Path parameters

r_x 0

x_c 0

r_y 0

y_c 0

Reset path

VSP on

Desired speed -1 0 1 1.5

Controller gains

Kp x 0

Kd u 0

Kb u 0

Kp y 0

Kd v 0

Kb v 0

Kp psi 0

Kd r 0

Kb r 0

Power

Figure 5.1: Data Dashboard UI.

Data Dashboard utilizes the *shared variable* feature of LabView. This enables network write and read access to variables. The shared variables feature can be assigned to a set of data types, such as arrays, doubles, and boolean.

Furthermore, the defined shared variables can be found by the Data Dashboard application by specifying the IP-address where the variables are located. This can be on a host computer, or on a real time target, such as a cRIO.

A drawback with Data Dashboard is that the application is not directly compatible with NI VeriStand. Originally, it is only compatible with the LabView environment. However, by creating a custom device that runs variables from VeriStand in LabView simultaneously, Data Dashboard can be used together with VeriStand. This custom device is established in Section 5.2.

5.1.3 Custom device

NI VeriStand includes several software applications, which meet the requirements of the users. Either it is displays, signal routing, or joystick interaction. In addition, National Instruments has provided a *custom device* application, that enables the users to

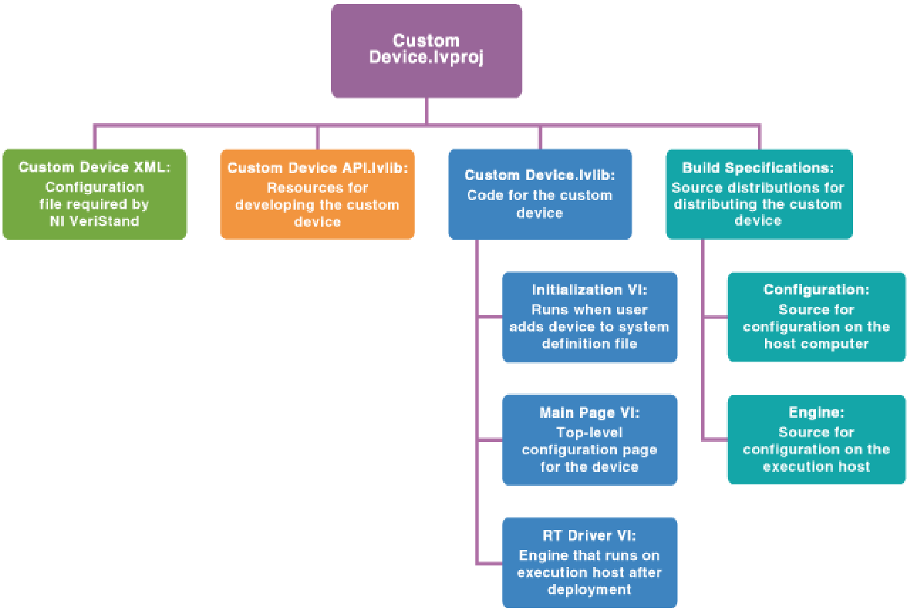
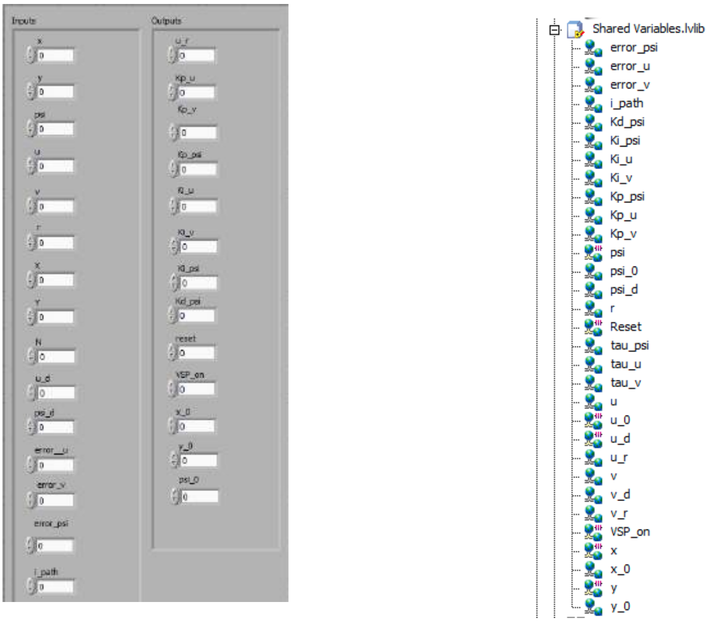

Custom Device.lvproj
Custom Device XML: Configuration file required by NI VeriStand
Custom Device API.lvlib: Resources for developing the custom device
Custom Device.lvlib: Code for the custom device
Build Specifications: Source distributions for distributing the custom device
Initialization VI: Runs when user adds device to system definition file
Main Page VI: Top-level configuration page for the device
RT Driver VI: Engine that runs on execution host after deployment
Configuration: Source for configuration on the host computer
Engine: Source for configuration on the execution host

Figure 5.2: The build up of a custom device. Courtesy of National Instruments (2015).

create their own devices. This introduces more functionality to the NI programs.

Setting up a custom device from scratch requires in-depth knowledge of LabView and VeriStand. However, NI has provided a *custom device template*, that creates a set of initialization files and VIs. These files can be modified to obtain the performance and behavior of interest. The structure for a custom device is given in Figure 5.2.

For additional functionality and to ease the work of setting up a custom device, Stephen (2014b) has provided an extension to this template, which adds additional files that can be modified. See Appendix C.1 for more information.



Inputs

x 0
y 0
psi 0
u 0
v 0
r 0
X 0
Y 0
N 0
u_d 0
psi_d 0
error__u 0
error_v 0
error_psi 0
i_path 0

Outputs

u_r 0
Kp_u 0
Kp_v 0
Kp_psi 0
Ki_u 0
Ki_v 0
Ki_psi 0
Kd_psi 0
reset 0
VSP_on 0
x_0 0
y_0 0
psi_0 0

(a) Some of the input and outputs to the iPad, defined as channel data.

Shared Variables.lvlib
- error_psi
- error_u
- error_v
- i_path
- Kd_psi
- Ki_psi
- Ki_u
- Ki_v
- Kp_psi
- Kp_u
- Kp_v
- psi
- psi_0
- psi_d
- r
- Reset
- tau_psi
- tau_u
- tau_v
- u
- u_0
- u_d
- u_r
- v
- v_d
- v_r
- VSP_on
- x
- x_0
- y
- y_0

(b) Shared variable library.

Figure 5.3: Channel data and shared variables library.

5.2 Application setup

Using the custom device template, and the easy custom device tool expansion, the custom device application can be set up. The application will use shared variables as inputs and outputs to the tablet, and thus the performance of the system of consideration can be remotely controlled and monitored. The application is based on the work by Stephen (2014a), which has set up a connection between NI VeriStand and iPad or Android devices. It was originally designed for control and monitoring the performance of an engine. The original custom device is attached to this thesis electronically, and in the following the alterations done for control and monitoring CSEI are presented.

First, the variables of consideration must be set up. The set up of the variables is done in the Channel Data control file. These variables depend on the application and the system of consideration. In this master thesis, CSEI with a path following control system will be used. Consequently, position, velocity, controller gains, and guidance parameters are typical to include. In Figure 5.3a, some channel data in the Data Dashboard custom device can be seen.

Notice that the channel data marked as input are inputs to the iPad, and not to the host. Consequently, the outputs of the custom device is input to the host. In addition, a library containing shared variables, which are used for communication over WiFi, must be established. It is beneficial to call these shared variables the same as the channel data inputs and outputs (see Figure 5.3b).

Furthermore, in the init.VI file, the channel data and shared variables library are loaded. In this routine, the IP connection to the host is established. Then start.VI runs, and loads the channel data and libraries into the loop. The execute.VI runs as long as the custom device is running, and it is located within in the Main Page VI routine. In execute.VI, the channel data inputs and outputs are linked to the shared variables library. It is also in this routine where the values of these variables are updated during each loop. The execute.VI can be seen in Figure 5.4. When the loop ends, stop.VI is launched. This routine stops the WiFi connection, and gives the channel data default values.

Recall the build up of a custom device in Figure 5.2. Located to the left, an XML-file for NI VeriStand configuration. In order to run the custom device on a real time target, such as a cRIO, modifications must be implemented in this file. The cRIO is running VxWorks as operating system, and the custom device must take this OS into consideration. By adding VxWorks as a supported target in the XML-target, the custom device is ready to run on the cRIO. The code for VxWorks support is given in Appendix C.2.

By building the developed custom device, with the accompanying XML-file and libraries, real time communication between NI VeriStand and mobile devices can be performed.

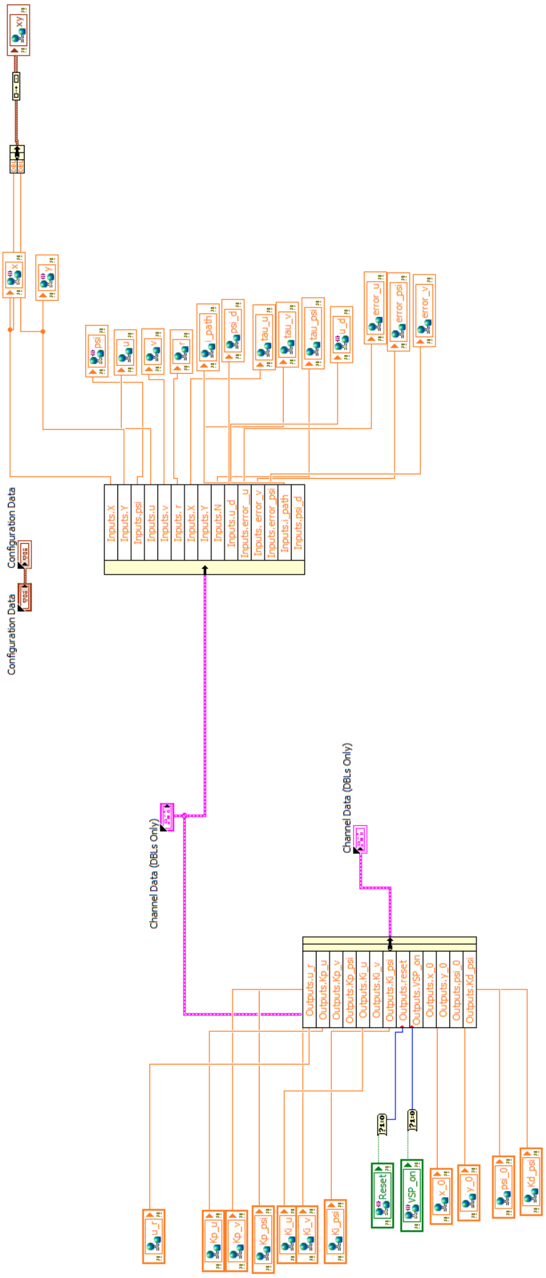


Figure 5.4: The execute.VI routine.

5.2.1 xy-plot

An extra modification in the custom device allows for xy-position representation in the application. Since the Data Dashboard application is sensitive to the input in the displays, a modification on the custom device is done in order to obtain xy-position of the vessel in the MC lab.

In input to the xy-graph on Data Dashboard must be an array with two inputs, x and y. Thus, a custom control that sorts the x any y into a shard variable is implemented. The control is called *xyClusterArray.ctl*, provided by Idland (2015), and is attached to this master thesis electronically.

Further, the input to the array are the shared variables x and y (see Figure 5.4). Hence, by compilation, this array can be the input to the xy-graph on Data Dashboard. In Figure 5.5, the xy-plot presenting the xy position of CSEI in the MC lab during a test run.

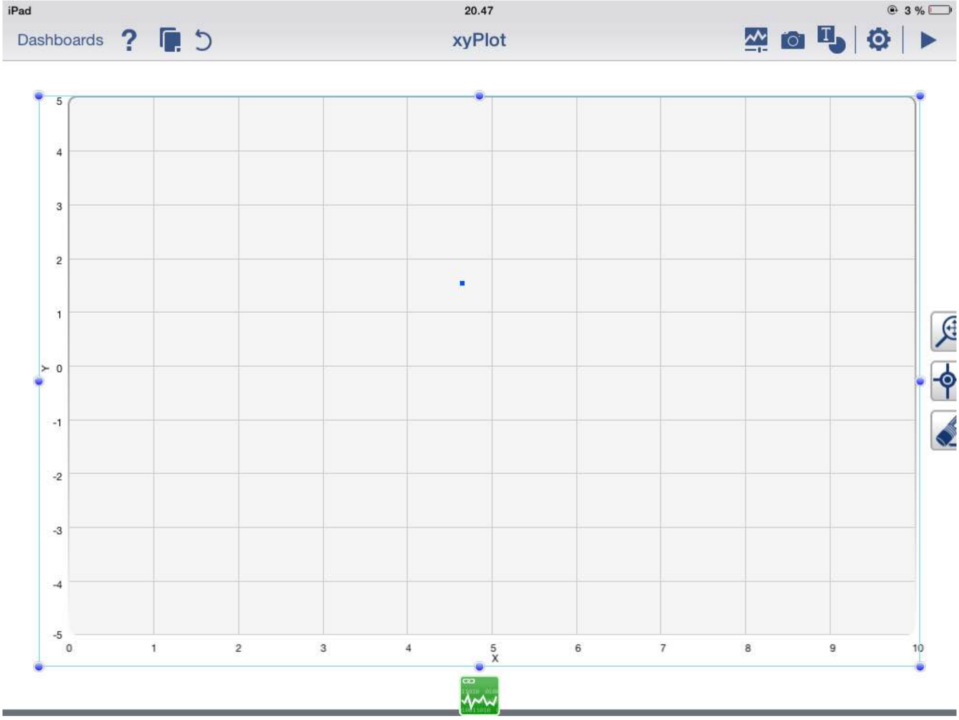


Figure 5.5: CSEI (blue point) position on Data Dashboard.

5.3 Application test on CSEI

In this section, the application will be connected to the cRIO on CSEI. The test will use an iPad to manually control CSEI. With an adequate user interface created by the data dashboard displays, the behavior can be controlled and monitored.

5.3.1 Fail-to-safe

To account for a situation where the communication between the vessel and iPad is lost, a fail-to-safe routine is implemented. CSEI can manually be controlled using a PlayStation controller. This controller is the highest level of control, and can take over if something goes wrong with the control systems. Based on the control mode, the controller can manually control the thruster, or the direction. The hierarchy is presented in Figure 5.6.

The figure illustrates 4 control modes:

- force control,
- thruster control,
- VeriStand UI,
- and iPad UI.

The force and thruster control mode are by using the PlayStation controller. The change of control mode is done by using *square*, *triangle*, and *circle*, *cross* buttons on the game pad controller. The calculated thruster forces are mapped into PWM signals. This feature provides a safe way of controlling the vessel.

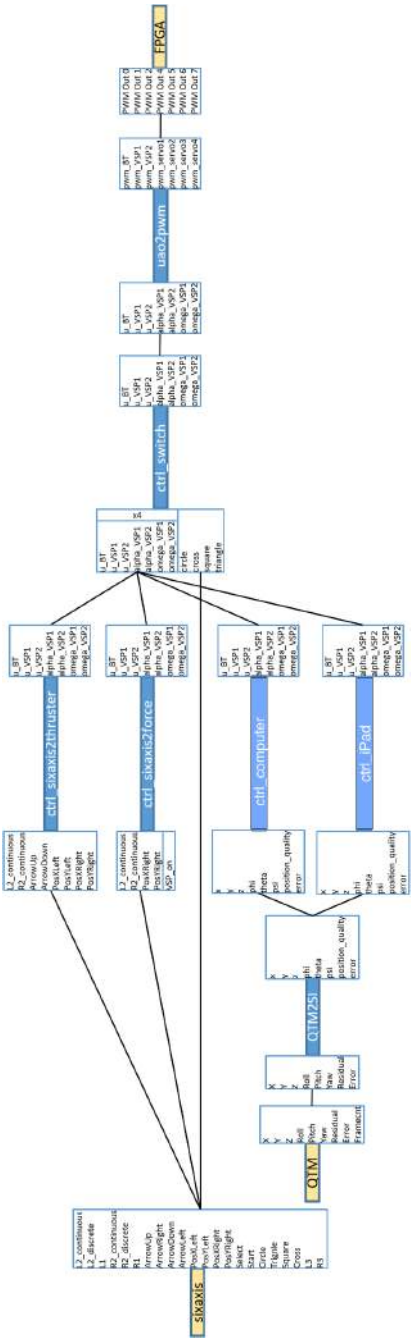
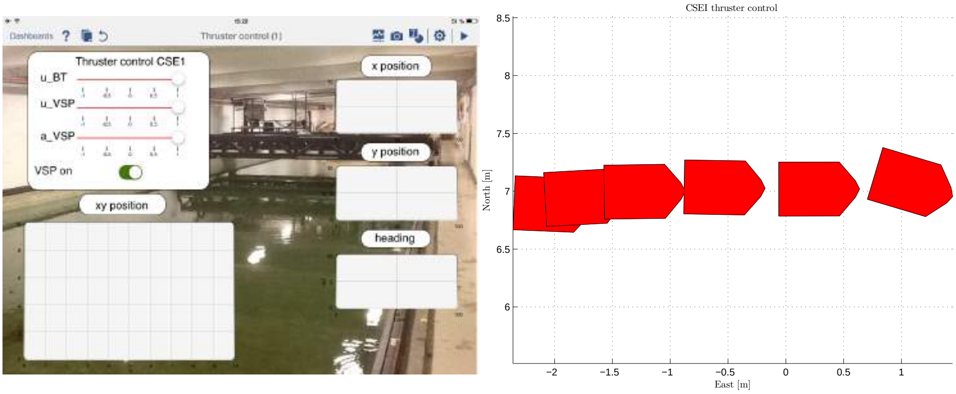


Figure 5.6: The different control modes for CSEI.

5.3.2 Thruster control from iPad

In this section, the thrusters are manually controlled by the iPad. The user interface used for control is presented in Figure 5.7a. A test run was conducted in the MC lab, and the thruster was controlled with the use of the Data Dashboard application. The objective was to test the robustness of the application, and that commands were given and received in real time. The tests revealed a satisfying performance for the application with no sign of unstable communication. A sample of the position during the test is seen in Figure 5.7b.



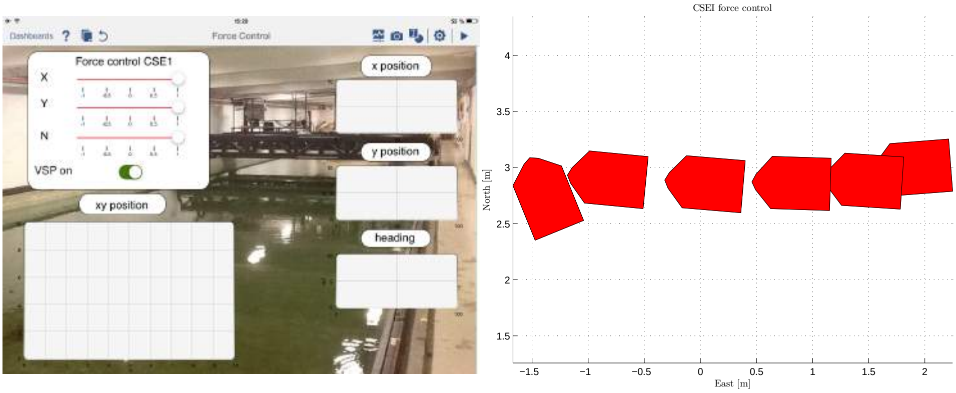
(a) iPad UI.

(b) Position.

Figure 5.7: Remote individual CSEI thruster control.

5.3.3 Force control from iPad

Here the iPad is used to control the forces in surge, sway, and the yaw moment for CSEI. The user interface is viewed in Figure 5.8a. As for individual thruster control, a test run was conducted. Results for driving along a straight line is given in Figure 5.8b.



(a) iPad UI.

(b) Position.

Figure 5.8: Remote CSEI forces and moments control.

5.4 Concluding remarks

The custom device tools are successfully used in order to make Data Dashboard communicate with NI VeriStand. The final solution with the switching between control modes renders a robust way of controlling the vehicle. In the events of drift offs, or the communication loss between CSEI and the iPad, the control can be taken with the PlayStation controller.

However, in the case of communication loss of the PlayStation controller, no redundancy is present in the system. A solution could be to implement a *watchdog* timer function. This function keeps track if communication is present to the PlayStation controller, and can safely turn off the system in the case of communication loss. However, a stable and well working communication with the PlayStation controller has proven itself stable and reliable.

The iPad works well to receive and send data. Changes are applied to the thruster of CSEI in real time. The application offers a set of controls and indicator that can be used for control and monitoring purposes.

A drawback with Data Dashboard is that custom displays are not possible to create or import. This allows the user only to use the displays provided by the application, and may lead to limitations both operationally and graphically.

The displays used for monitoring and control are sensitive to which data type the shared variable is. Some graphs will only allow shared variables of type array, where both x and y values are specified. This must be specified in the custom device during the design stage.

The LabView/VeriStand framework is expensive, leading to an unsuitable solution for a low-cost vehicle, such as Neptunus. A solution can be to set up a communication protocol within the existing Neptunus OS. Further, the internet browser in the Android device or iPad can be used to load the dashboard. Another solution is to make an application for remote control and monitoring the ROV from the iPad.

Chapter 6

Hardware-in-the-loop testing

According to Sørensen (2013, Sec. 2.10.4), marine vessels and their control systems are becoming more demanding and complex, and thus there is an increased need to test and verify the response. In order to reduce the risks related to the control software, several methodologies have been developed. These methodologies test the response of the systems in different cases, verify the behavior, and address failures and improvements before sea trials and commissioning.

A common method for testing the performance of a computer-based system is the *hardware-in-the-loop* (HIL) simulation. According to Skjetne and Egeland (2006), a HIL simulator can be defined as

> *A real-time simulator, constructed by hardware and software, that is configured for the control system under consideration, embedded in external hardware, and interfaced to the target system or component through appropriate I/O. During execution the target system or component will not experience any qualitative difference from being integrated to the real system.*

During a HIL test, the control system is connected to a extensive simulation model which represents the vessel, and operates real time. The HIL-test can give an indication on how the control system is working full scale. By applying tests within a simulation module, the risk of ruining e.g. actuators on the vehicle during commissioning is reduced. Moreover, by conducting HIL tests, no hazards are present for the personnel. In addition, it causes no down time for the rest of the system. In Figure 6.1, a representation of a HIL test is given.

Johansen et al. (2005) point out that some of the key benefits with the HIL testing are:

- realistic testing at an early stage,

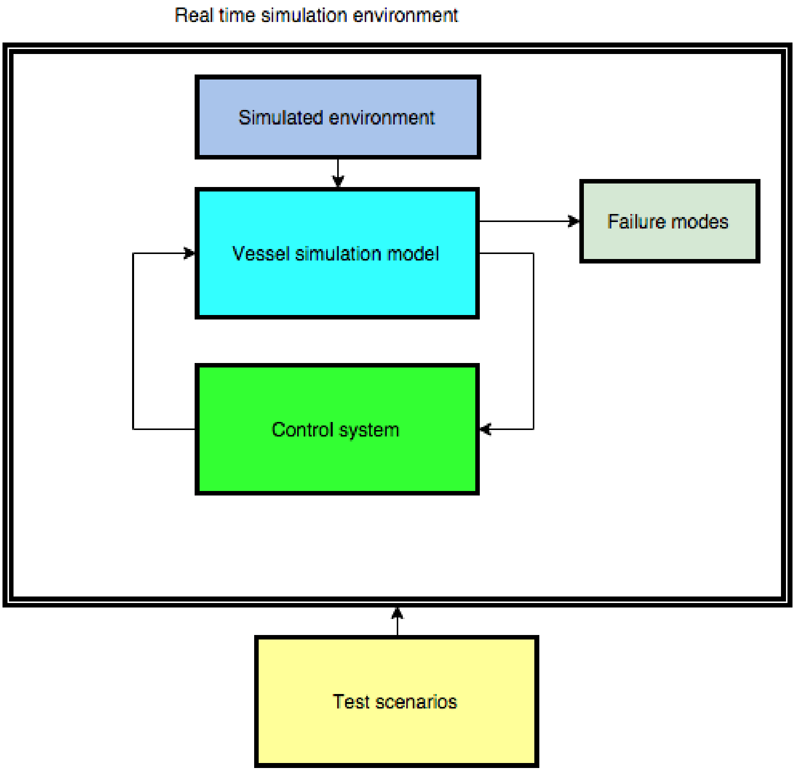


Figure 6.1: HIL testing environment.

- arbitrarily environmental conditions can be tested,
- complex failure methods can be applied,
- and that HIL-testing provides a non-destructive testing environment for the vehicle.

In this chapter, the designed and implemented control systems from Chapter 4 are further tested. In this chapter, HIL testing is used to test the performance of the control algorithms. Due to technical limitations in the MC lab, only CSEI will be HIL tested.

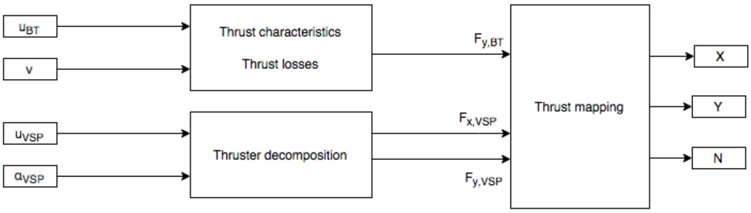


Figure 6.2: The thruster mapping for CSEI in the HIL set up.

6.1 CSEI HIL-model

6.1.1 Thruster dynamics

The main difference between the simulation model of CSEI from Section 3.3 is in the thruster dynamics. While the control inputs in the simulation model is $\boldsymbol{u} = [u_{\text{VSP,x}}, u_{\text{VSP,y}}, u_{\text{bt}}]^T$, the control inputs in the HIL model is

$$\boldsymbol{u}_{\text{HIL}} = \begin{bmatrix} u_{\text{bt}} \\ u_{\text{VSP}} \\ \alpha_{\text{VSP}} \end{bmatrix} \tag{6.1}$$

Hence, the same assumption as in Section 3.3.1 with one VSP, where $\alpha_{\text{VSP1}} = \alpha_{\text{VSP2}}$ is applied. Moreover, the bow thruster characteristics and thruster losses have been calculated. These expressions are given in Appendix A.7.1. In addition, thruster force identifications for the VSPs have been performed through model-scale experiments. The thruster forces have been decomposed into X and Y forces. These calculations are provided in Appendix A.7.2. Finally, the calculated forces are mapped correctly to the geometric location of the thrusters. This mapping is presented in Figure 6.2. Note the dependency of the velocity when dealing with the bow thruster. High velocity leads to a lower effect of the bow thruster. The rest of the kinetics, as well as the kinematics is the same as in the system identification in Section 3.3.

6.1.2 Observer

Since no position measurements are available in the MC lab, an observer is added in the HIL model for comparison basis. Observer design and theory is not a part of this thesis, such that the observer used in HIL simulations will only be presented briefly. The position measurements are lowpass filtered, before the velocity vector is found by derivation. Furthermore, the differentiated velocity vector is low pass filtered. At last, the position derivative is multiplied with the transpose of the rotation matrix.

$$\bar{\boldsymbol{\eta}}_{\boldsymbol{i}} = \alpha_1 \boldsymbol{\eta}_{\boldsymbol{i}} + (1-\alpha_1)\bar{\boldsymbol{\eta}}_{\boldsymbol{i-1}} \tag{6.2}$$

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{i}} = \frac{\bar{\boldsymbol{\eta}}_{\boldsymbol{i}} - \bar{\boldsymbol{\eta}}_{\boldsymbol{i-1}}}{h} \tag{6.3}$$

$$\dot{\bar{\boldsymbol{\eta}}}_{\boldsymbol{i}} = \alpha_2 \dot{\boldsymbol{\eta}}_{\boldsymbol{i}} + (1-\alpha_2)\dot{\bar{\boldsymbol{\eta}}}_{\boldsymbol{i-1}} \tag{6.4}$$

$$\hat{\boldsymbol{v}} = \boldsymbol{R}^T(\psi)\dot{\bar{\boldsymbol{\eta}}}_{\boldsymbol{i}} \tag{6.5}$$

Here, $\bar{\boldsymbol{\eta}}_{\boldsymbol{i}}$ is the filtered position, and α_1 and α_2 is the filter time constants. In all the simulations, α_1 and α_2 is set to 0.1. This is also the case for the model-scale tests.

	Squared path	**Curved path**
$K_{p,u}$	60	100
$K_{i,u}$	0.5	5
$K_{p,\psi}$	20	1
$K_{i,\psi}$	0.01	1
u_d	0.6 m/s	0.6 m/s
R_{lim}	0.5 m	2 m

Table 6.1: Controller gains for CSEI during heading on WP HIL-test.

6.2 HIL testing the control systems

With the defined HIL model of CSEI, the established control algorithms from Chapter 4 can be validated. The thruster dynamics introduce more non linearity in the system, and thus the responses are expected to be different.

To guarantee that the control systems and the vessel simulation model is running real time, simulations on the cRIO are considered. The control algorithms, together with the HIL simulation model of CSEI are compiled in MATLAB as a *.out*-file, compatible with the VxWorks OS on the cRIO. Consequently, the VeriStand project can be set up, and the models loaded onto the cRIO for real-time simulations. Control and monitoring are performed both from a host-computer, and from the iPad.

Within the HIL simulation environment, current is present, acting as an unknown force to the control system. The parameters are the same as in Section 3.4, i.e $V_{\max} = 0.02$ m/s, $V_{\min} = 0.001$ m/s, $\mu_c = 0.01$, and current direction $\beta_c = \pi$.

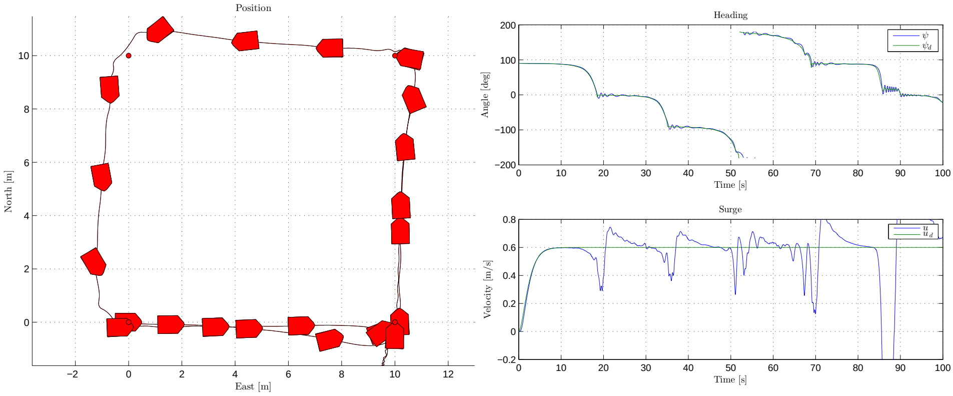
6.2.1 Heading on waypoints

CSEI's performance will be tested in the HIL environment by considering the same two set of waypoints as in Section 4.4.1: a 10 meter square, and a curved path. Gains related to the control system are found in Table 6.1.

The resulting HIL response for the two paths are presented in Figures 6.3 and 6.4. It is clear that CSEI is capable of following the paths with satisfying performance. However, stuttering behavior around the waypoints are observed. This is due to the assumption of the VSPs acting as a propeller with a rudder. When the circle of acceptance is reached, and the new waypoint is selected in the controller, the angle on the VSPs are changed. As opposed to a conventional propeller-rudder system, the speed on the vessel is depended on the VSP angles. Consequently, a change in α_{VSP} will affect the the forward velocity.

Due to the added thruster dynamics in the HIL model, the forward speed must be quite high to make the VSPs guide the vessel through the path. With a too low forward speed, the vessel will not have enough power to move forwards and turn at the same time. Thus, it will nearly stand still and turn around. With the model-scale behavior of CSEI in mind, a forward speed of 0.6 m/s is high.

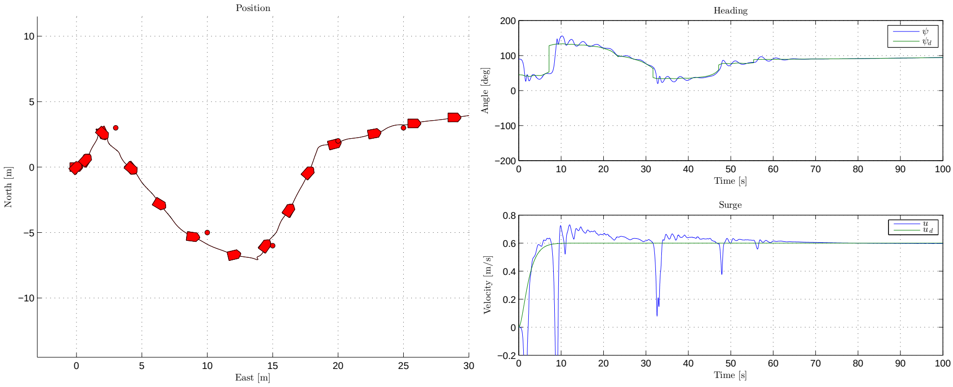
A implementation of a derivative gain could remove some of the oscillations in heading and surge. However, since no velocity measurements are available in the MC lab, just an uncertain observer, no controller with derivative action is considered.



(a) Position.

(b) Heading and velocity.

Figure 6.3: HIL test CSEI, heading on waypoints. Squared path.



(a) Position.

(b) Heading and velocity.

Figure 6.4: HIL test CSEI, heading on waypoints. Curved path.

	Squared path	**Curved path**
$K_{p,u}$	100	100
$K_{i,u}$	5	5
$K_{p,\psi}$	20	20
$K_{i,\psi}$	1	1
u_d	0.6 m/s	0.6 m/s
R_{lim}	1 m	2 m
Δ	3	5

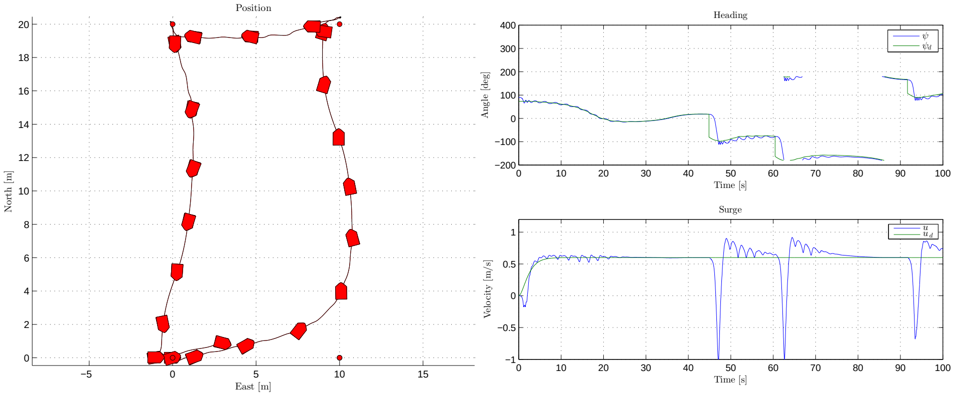
Table 6.2: Controller gains for CSEI during lookahead based LOS HIL-test.

6.2.2 Lookahead-based line of sight

Next control algorithm of consideration is the lookahead based line-of-sight. Also here the paths are the same as in the simulations (see Section 4.5.1). Parameters related to the control system are presented in Table 6.2.

The North-East response, in addition to the controller objectives can be seen in Figure 6.5 for the rectangular path, and in Figure 6.6 for the curved path. The main difference between lookahead-based LOS and heading on waypoints can be seen in the corner (10,10) for the rectangular case. CSEI takes waypoint i and $i+1$ into consideration, and slides into the path as a function of Δ. Also for this case, the vessel is capable of following the paths with a acceptable speed assignment. Due to the sudden change in waypoints, the forward speed is reduced significantly. Again this is because the change in α_{VSP} affects the forward velocity.

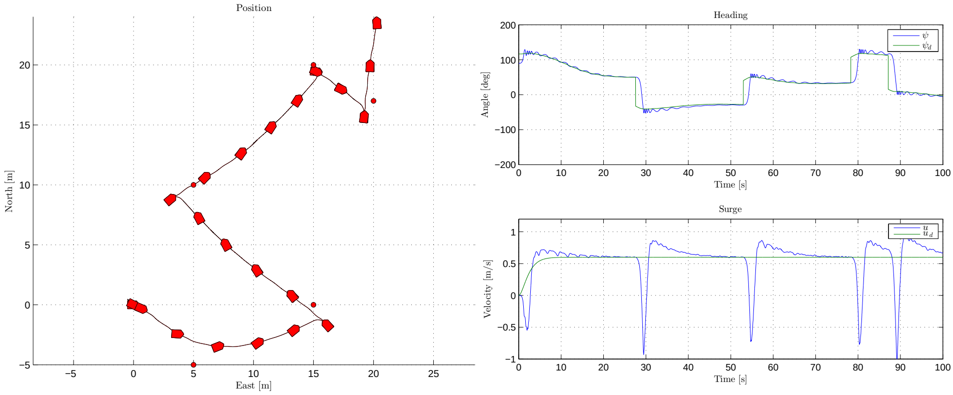
Since only the VSPs are active in this control system, the same requirement for forward speed as in heading on waypoints is registered. In order to move forwards and turn, a speed at above 0.5 m/s must be set. As for heading on waypoints, oscillations in the heading and velocity are observed. These oscillations could be avoided, by the implementation of a derivative action.



(a) Position.

(b) Heading and velocity.

Figure 6.5: HIL test CSEI, lookahead based LOS. Squared path.



(a) Position.

(b) Heading and velocity.

Figure 6.6: HIL test CSEI, lookahead based LOS. Curved path.

	Tracking update law	**Gradient update law**
$\boldsymbol{K}_p$	diag(0.1,1,1)	diag(1,1,2)
$\boldsymbol{K}_d$	diag(100,100,1000)	diag(2000,3000,2000)
$\boldsymbol{K}_b$	diag(0.1,0.1,0.01)	diag(0.001,0.1,0.001)
u_d	0.6 m/s	0.6 m/s
μ	0	0.001

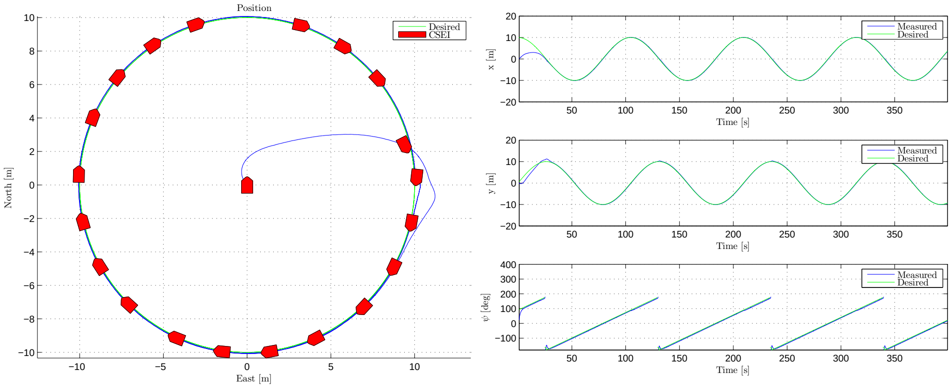
Table 6.3: Controller gains for HIL-testing the maneuvering backstepping control system for CSEI.

6.2.3 Maneuvering by backstepping - full actuation

For validation and comparison purposes, the same circular path $(r_x,r_y) = (10,10)$ as in the simulation studies is considered. HIL simulations both with tracking update law and modified gradient update law are performed. Also, under this control system the current is present. In Table 6.3, controller gains and other relevant parameters are stated.

The results for the backstepping controller reveal strong path following capabilities, both for the tracking and modified gradient update law. Referring the the tracking update law scenario in Figure 6.7 and 6.8, both the geometric and dynamic task is fulfilled. Very low geometric errors are observed, and CSEI holds the desired speed along the path. The small oscillations are due to the ocean current. Considering the modified gradient update law scenario in Figure 6.9 and 6.10, the convergence to the path is observed. Some transients in the dynamic task are registered in the beginning. This is caused by the path generation is waiting for CSEI to converge.

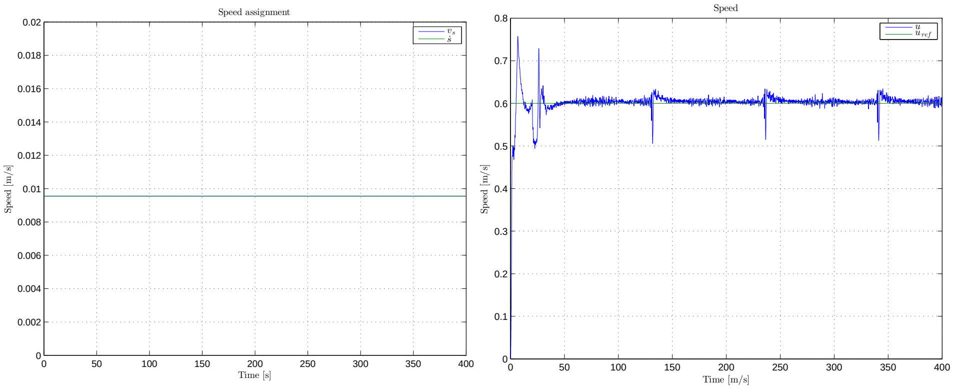
HIL testing has only been considered for the fully actuated case of CSEI. This is because the underactuated case was proven in Section 4.6.4 to not give satisfying results that converges $\boldsymbol{\eta}$ to $\boldsymbol{\eta}_d$. In addition, the error in x and y was to large to consider for tests in the MC lab.



(a) Position.

(b) Geometric task.

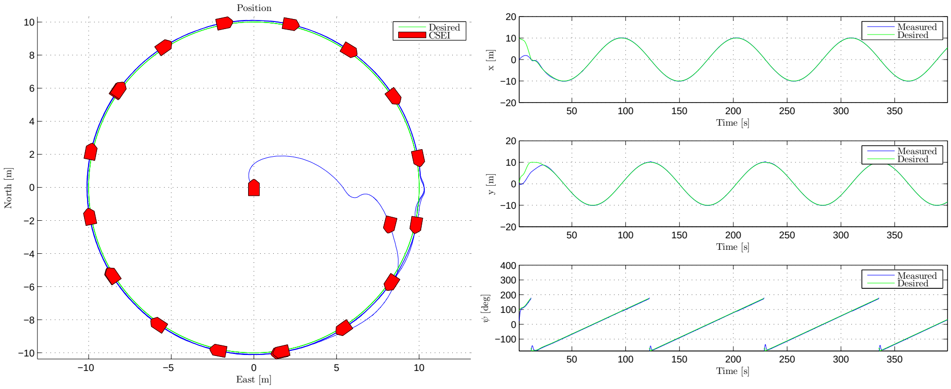
Figure 6.7: HIL test CSEI, tracking update law path following.



(a) Speed assignment.

(b) Speed.

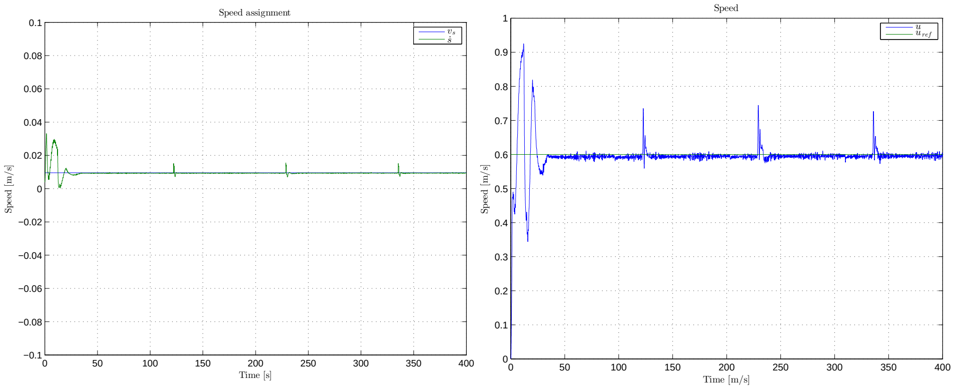
Figure 6.8: HIL test CSEI, tracking update law. Dynamic performance.



(a) Position.

(b) Geometric task.

Figure 6.9: HIL test CSEI, modified gradient update law path following.



(a) Speed assignment.

(b) Speed.

Figure 6.10: HIL test CSEI, modified gradient update law. Dynamic performance.

	Tracking update law with LOS
$\boldsymbol{K}_p$	diag(1,1,15)
$\boldsymbol{K}_d$	diag(2000,2,20)
$\boldsymbol{K}_b$	diag(10,10,1)
u_d	0.6 m/s
μ	0
Δ	5

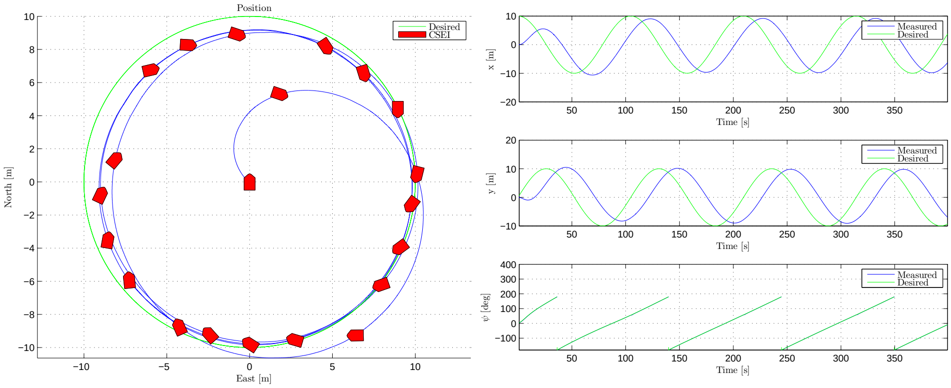
Table 6.4: Controller gains in maneuvering control system for CSEI.

6.2.4 Maneuvering line of sight

The last HIL-test of consideration is regarding the maneuvering LOS control algorithm. Since full actuation renders accurate path following characteristics, maneuvering LOS is only tested on the underactuated case of CSEI. As for the simulation study, only tracking update law are considered. The controller gains are given in Table 6.4.

The geometric task in Figure 6.11 reveals capable path following results. The convergence is not as strong as for the fully actuated case. Notice also that the integral action is gained up considerably due to the fully actuated case. The underactuated case is more vulnerable to disturbances.

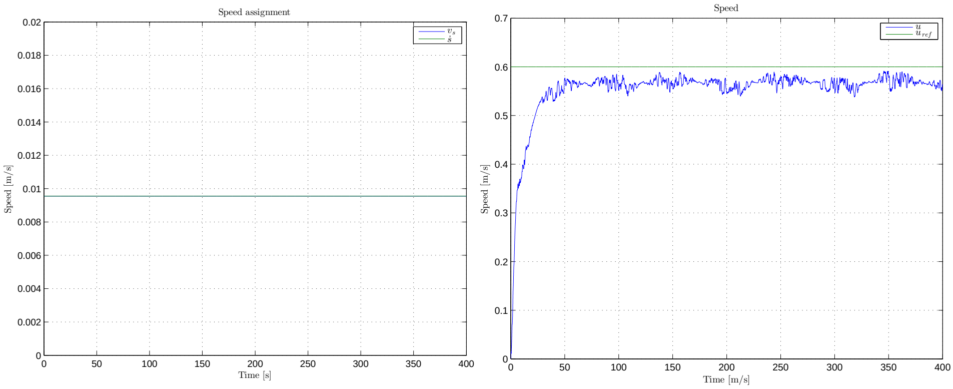
Moreover, in the dynamic task presented in Figure 6.12 it is observed that an approximate constant velocity is held along the path. However, some errors occur here. A reason may be unmodelled dynamics in the system, which causes uncertainties, and thus errors. In addition, the observer is strongly depended on the time constants α_1 and α_2. Wrong constants may impact the dynamic task, and consequently cause error in the geometric task as well.



(a) Position.

(b) Geometric task.

Figure 6.11: HIL test virtually ruddered CSEI, maneuvering LOS path following.



(a) Speed assignment.

(b) Speed.

Figure 6.12: HIL test virtually ruddered CSEI, maneuvering LOS. Dynamic performance.

6.3 Concluding remarks

Four motion control algorithms have been HIL tested. All the control algorithms gave responses in compliance to the expectations. For the Nomoto assumption in heading on waypoints and lookahead-based LOS, the performance was not optimal around the waypoints or in the speed assignment. Due to the thruster dynamics in the HIL-model, a high speed must be applied to the thruster for moving forwards. This is not in compliance with the model-scale vessel. As a result, some impact is seen in the speed vs. desired speed task.

Under maneuvering by backstepping, full actuation is applied, and a strong path following capability is observed. As a conclusion, the maneuvering backstepping control algorithm renders the best performing path following algorithm for CSEI. A drawback however, is that full actuation is needed to run the maneuvering system.

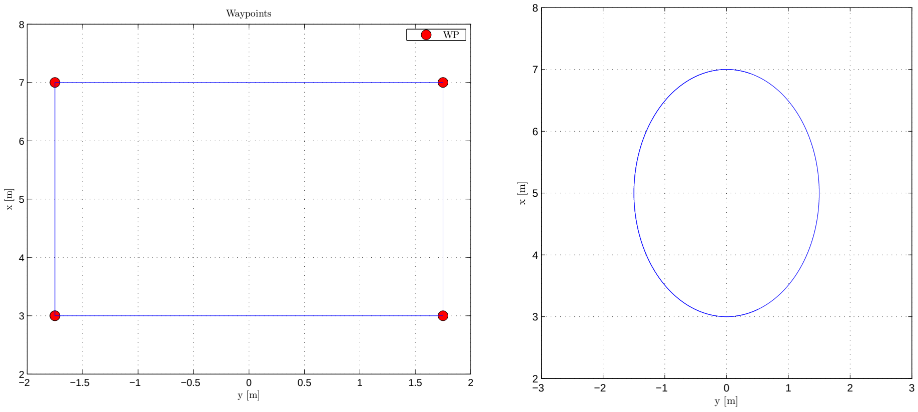
Moreover, a maneuvering control design with implemented LOS is tested on the Nomoto approximation model of CSEI. The results give satisfying path following capabilities. However, some errors are present, and may be caused by unmodelled dynamics and choice of observer.

In addition, backstepping techniques is a more advanced control algorithm, and need velocity inputs to work optimal. Velocity measurements are not directly applicable in the MC lab, only through the observer.

Chapter 7

Model scale experiments

In this chapter, the control algorithms will be tested on CSEI. Remote control and monitoring of the performance is done from the iPad. Unlike the simulation and HIL test studies, the area in the MC lab is considerably smaller. Thus, a change in the path is necessary. In Figure 7.1 below, the paths for the heading on WP, lookahead-based LOS, and maneuvering by backstepping control algorithms are presented. The rectangular path is used to test the first two control systems, while the ellipsoid path is for the latter control systems. The iPad UIs are provided in Appendix C.3.



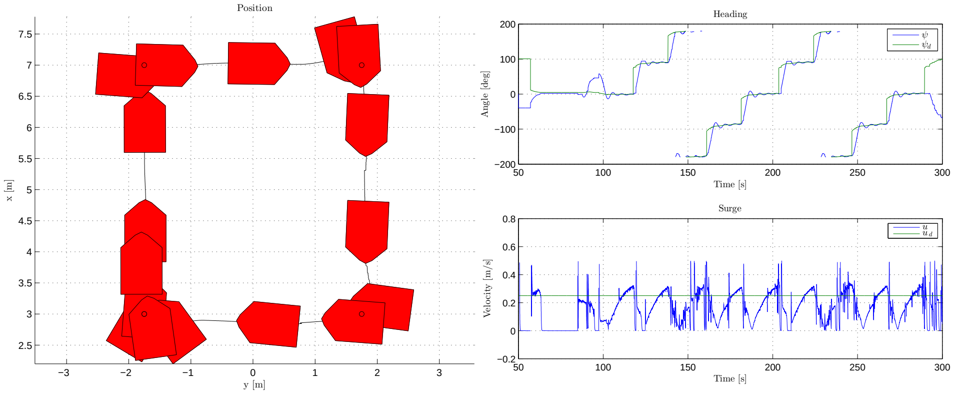
(a) Rectangular.

(b) Ellipsoid.

Figure 7.1: The paths for CSEI to follow in the MC lab.

Parameter	Value
u_d	0.2 m/s
$K_{p,u}$	1.0
$K_{i,u}$	0.1
$K_{p,\psi}$	10.0
$K_{i,\psi}$	1

Table 7.1: Controller gains, model scale test on CSEI, with the heading on waypoints control system.



(a) Position

(b) Heading and surge velocity.

Figure 7.2: Virtually ruddered CSEI model scale test, heading on waypoints.

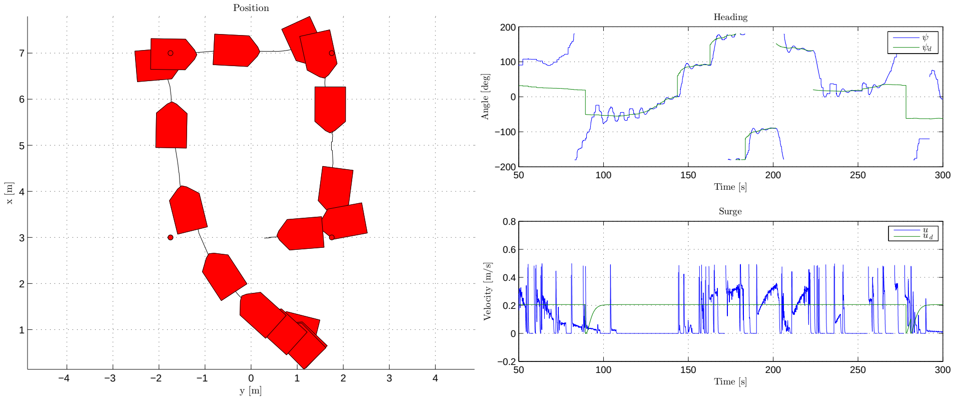
7.1 Heading on waypoints

The heading on waypoints control algorithm is tested on CSEI with the gains given in Table 7.1. These values are tuned online using the iPad, until satisfying performance in heading is obtained. A section of the run is presented in Figure 7.2a and 7.2b below.

It can be concluded from the position and the heading objective that CSEI is following the path in a satisfying manner. Low error is registered. The speed assignment is not followed. The main reason is based on the discussion from the HIL testing, where the propeller-rudder assumption makes it hard for the control algorithm to follow the path at the desired speed. On CSEI, additional unmodelled dynamics and uncertainties are present, which results in low accuracy on the speed controller. Consequently, a weighting on the tuning must be made, and choose whether that the heading or speed assignment should be dominating.

Parameter	Value
u_d	0.2 m/s
$K_{p,u}$	1.0
$K_{i,u}$	0.2
$K_{p,\psi}$	2.0
$K_{i,\psi}$	0.1
Δ	2

Table 7.2: Controller gains, model scale test on CSEI, with the lookahead-based LOS control system.



(a) Position.

(b) Heading and surge velocity.

Figure 7.3: Virtually ruddered CSEI model scale results, lookahead based LOS.

7.2 Lookahead-based line of sight

CSEI is also commanded to follow the rectangular path with the lookahead-based control algorithm. A satisfying heading performance is obtained by choosing the gains as in Table 7.2. A section of the response for CSEI along the path is given in Figure 7.3a and 7.3b.

Also the convergence to the straight line segments of the path can be seen in full scale. From the start position, it slides into the line as a function of the lookahead distance Δ. The Nomoto model assumption works well for the geometric task, and the heading is following the desired heading without large deviations. As for the heading on WP control algorithm, the speed assignment is not working satisfying.

	Tracking update law
$\boldsymbol{K}_p$	diag(1,1,0.1)
$\boldsymbol{K}_d$	diag(2000,2000,0)
$\boldsymbol{K}_b$	diag(1,1,1)
u_d	0.1 m/s
μ	0

Table 7.3: Controller gains in maneuvering by backstepping control system for CSEI.

7.3 Maneuvering by backstepping

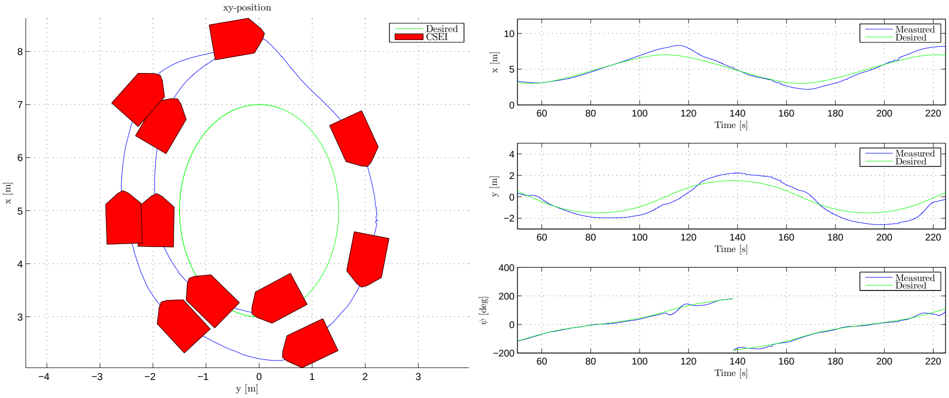
To introduce a more advanced control algorithm in model-scale, maneuvering by backstepping is considered. As for the previous control algorithms, the controller gains were tuned online by the iPad. These values are provided in Table 7.3. For this model scale experiment, full actuation is considered, leading to that the bow thruster is enabled.

A part of the resulting run is given in Figures 7.4 and 7.6. From the geometric task it is observed that CSEI performs ellipsoid motion, but some errors are present. For perfect tracking, a full state feedback is needed. In the MC lab, no velocity feedback is present. Consequently, the velocities are provided by an observer. The accuracy and reliability of this observer is questionable, whereas peaks in the yaw rate are registered around the -π,π wrap around. From the dynamic task, noise on CSEIs speed along the path can be observed. This may cause peaks and instability in the control system. However, the dynamic task is better followed than the previous control algorithms. Moreover, unmodelled dynamics are probably a source that causes some errors.

It can be seen that also for maneuvering by backstepping algorithm, the speed along the path contains much noise. An extra improvement in the observer may reduce these errors, and provide more accurate following. Additional tuning could be performed for better performance. However, this is an advanced motion control algorithm, which are depended on reliable feedback to work optimally. Recalling the control law as

$$\boldsymbol{\tau} = -\boldsymbol{z}_1 + \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{M}\boldsymbol{\sigma} + \boldsymbol{M}\boldsymbol{\chi}v_s(s) - \boldsymbol{K}_b\boldsymbol{\xi}, \tag{7.1}$$

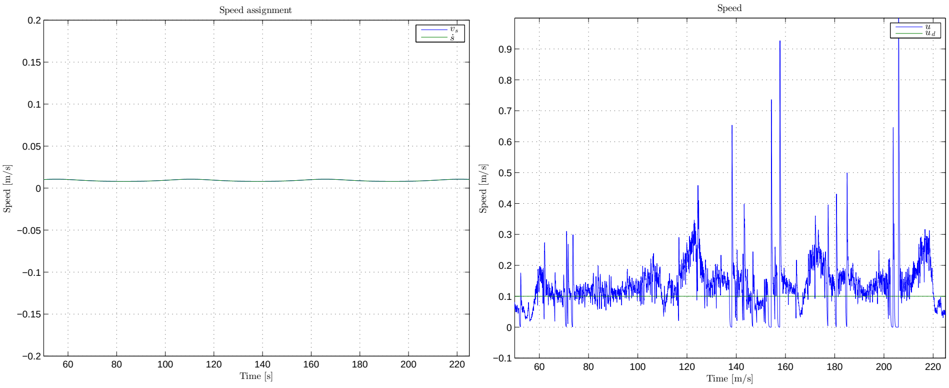
It can be verified that velocity measurements are needed both is the error state $\boldsymbol{z}_2$, and also in the damping and Coriolis matrices. Since only $\hat{\boldsymbol{\nu}}$ is provided, and may not be correct, errors are introduced.



(a) Position.

(b) Geometric task.

Figure 7.4: CSEI model scale test, maneuvering by backstepping path following.



(a) Speed assignment.

(b) Speed.

Figure 7.5: CSEI model scale test, maneuvering by backstepping. Dynamic performance.

	Tracking update law
$\boldsymbol{K}_p$	diag(1,1,10)
$\boldsymbol{K}_d$	diag(20,20,0)
$\boldsymbol{K}_b$	diag(0.1,0.1,0)
u_d	0.1 m/s
μ	0
Δ	1

Table 7.4: Controller gains in maneuvering LOS control system for CSEI.

7.4 Maneuvering line of sight

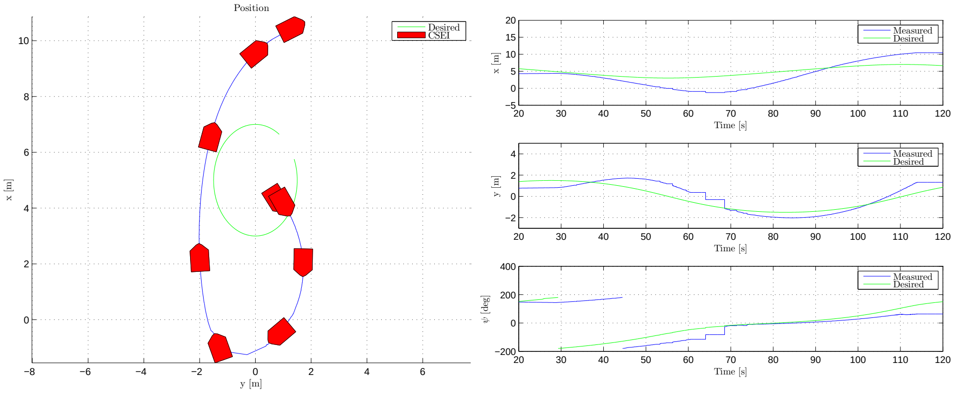
A model scale test of CSEI treated as an underactuated vessel is considered as the final scenario. The gains are given in Table 7.4.

In Figure 7.6, the geometric task in presented. It is seen that unsuccessful following is conducted. CSEI tries to converge to the path, but is unable to make the turn, and drifts off. Further, the dynamic task in Figure 7.7 reveals measurement noise on the speed measurements. Due to the lack of proper measurements, and a accurate observer, errors in the control system are present. It can be seen that since a tracking update law is running, CSEI is trying to catch up with the path parameterization speed. Consequently, a higher speed is registered.

CSEIs bad capability of following the desired path with the maneuvering LOS control algorithm may be caused by several factors. Wrong tuning may cause instability. Especially the lookahead distance Δ seems to be to large, such that the path is unsuccessfully followed. However, higher and lower lookahead distances have been implemented, without improved performance.

Another reason can be caused by the saturation in α_{VSP}. Under the assumption that this angle works as the rudder for CSEI, a saturation of +- 30° is implemented. For higher saturation, a better turning capability, and thus better path following may be achieved. However, a too high working area for α_{VSP} causes too rapid changes. The result will be a dominating stuttering behavior, which stops the forward motion of CSEI.

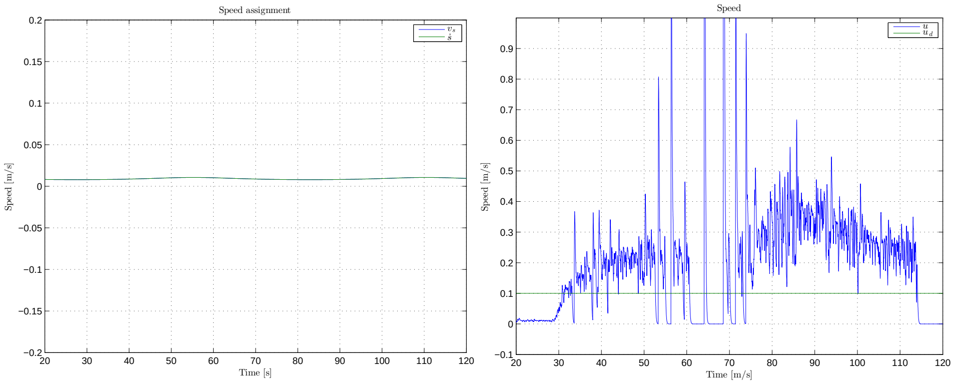
Maneuvering LOS control system is an advanced control system, depended on accurate feedback and a well known mathematical model. Too much uncertainties and unmodelled dynamics are reasons for an unstable motion control system. Also in this case, there is no guarantee for a correct representation of the damping and Coriolis matrices, which is needed in the control law.



(a) Position.

(b) Geometric task.

Figure 7.6: Virtually ruddered CSEI model scale test, maneuvering LOS path following.



(a) Speed assignment.

(b) Speed.

Figure 7.7: Virtually ruddered CSEI model scale test, maneuvering LOS. Dynamic performance.

7.5 Concluding remarks

The four path following control algorithms are tested on CSEI in the MC lab with passable performance. The geometric tasks of the control algorithms, namely to follow a desired heading along the path are successfully followed. The positioning system in the MC lab provides stable and accurate measurements.

Heading on WP and lookahead-based LOS reveal a proper path following performance, where the heading was followed throughout the waypoints. However, the speed assignment are not followed, mainly due to measurement noise, the sudden change of waypoints, and the rudder-propeller assumption.

Due to the lack of accurate velocity measurements, some errors were registered related to the speed assignments. Especially in yaw, where the observed calculated peak values of the yaw rate around the $-\pi$,π area. As a result, this gain were not active in the maneuvering control system. In addition, unmodelled dynamics and model uncertainties caused an x-y offset along the path.

The maneuvering LOS control algorithm did not reveal satisfying results (drift off). Due to several reasons, too much uncertainties and tuning dependence were introduced, resulting in a bad path following capability.

The maneuvering control systems are the most advanced control systems dealt with in this thesis. As a consequence, such systems are hard to debug in the case of mistakes, or saturation implementations.

Using the iPad to monitor the performance of CSEI, in addition to tune the gains online reveals a success. The commands are sent and given safely, with real time response.

Chapter 8

Conclusion and further work

8.1 Summary and Conclusions

In Chapter 3, mathematical model for Neptunus and CSEI are established using the robot-like vectorial notation from Fossen (1991). A full system identification for Neptunus is conducted, where theoretical results on modeling, software analysis and full scale tests are treated and commented.. The obtained model gives satisfying performance, in compliance with real experiments. The identified model is compared against the model from Follestad et al. (2014), which is based on estimated parameters. The result show that the estimated parameters are in compliance with the identified parameters. It is concluded that the estimated values are suitable in a preliminary design phase. Furthermore, Neptunus was proven to be passively stable, i.e. self-stabilizing in roll and pitch DOFs.

In Chapter 4, four motion guidance systems are designed and simulated using a set of pre-defined paths (heading on WP, lookahead-based LOS, maneuvering by backstepping, and maneuvering LOS). Neptunus is underactuated, and can only be controlled in surge and yaw. Heading on WP and lookahead-based LOS were able to successfully guide Neptunus along the desired path, despite the presence of current. The maneuvering guidance system relies on full actuation to remove steady state errors and guarantee convergence to the desired state. Due to Neptunus' underactuation, steady state errors were registered during the experiments. However, due to the good working turning capabilities for Neptunus, errors were negligible in the simulation study. Implementing an extra LOS algorithm to the maneuvering control system provided a better convergence to the path.

The Voith Schneider propellers of CSEI are mapped into a rudder-propulsion system. A Nomoto model is used to model the yaw dynamics: this model is further used in the heading on WP and lookahead-based LOS systems. The performance is satisfying, providing accurate tracking. However, the same problem as for Neptunus occurred for CSEI in the backstepping control system. Treating CSEI as an underactuated vessel gives bad tracking properties. By enabling the bow thruster, and treating CSEI as a fully actuated vehicle, accurate and satisfying performance are obtained. For the underactuated case of CSE low tracking capability are obtained. However, increased performance are obtained by using the maneuvering LOS control algorithm. Introducing an extra control law that deal with the cross track error, convergence to the path is seen.

For remote control and monitoring purposes, a custom device is developed using the LabView/VeriStand framework. These programs are used in the MC lab. The device enables communication between the vehicles controlled by the cRIO and mobile devices running iOS or Android. The design enables possibility of using the x-y plot in Data Dashboard, which shows the position of CSEI in the basin. The result shows a stable communication, where data was sent and received real-time. The device is interfaced in a switching system between the tablet, host computer and a Play Station controller. Thus, redundancy and safety are present.

For further evaluation of the control algorithms, HIL and model-scale testing were performed for CSEI. The HIL tests revealed satisfying results for all guidance systems, this was not the case for the model-scale. Due to lack of velocity measurements in the MC lab in addition to model uncertainties, not all control algorithms gave satisfying results. For heading on waypoints and lookahead-based LOS the path was successfully followed, but not with the desired speed. This is due to the assumption of a rudder-propeller system. Change of heading will also affect the produced forward thrust.

The maneuvering by backstepping algorithm it is considered too complex for optimal performance in the MC lab, since it needs full state feedback (position and velocities). Consequently, either velocity measurements or a proper working observer should be present in the lab. The backstepping algorithm is able to perform circular tracking with good performance, but with x-y offset. The implemented maneuvering LOS algorithm seems to have low tracking capability when CSEI is treated as an underactuated vehicle. CSEI tried to converge to the path, but it drifts off instead. The iPad was used to tune the controller gains online, and to monitor the performance.

From the model-scale tests in the MC lab, it is possible to conclude that a simple control design works best, given the available sensors and their accuracy. Even though the speed assignment is not fulfilled, no x-y offset occurs, which is the most prioritized task.

8.2 Recommendations for Further Work

The first recommendation is to establish full lab-functionalities for ROV Neptunus. This includes proper implementation of sensors on the ROV, in addition to an underwater positioning system. With this installed, the path following control algorithms developed in this thesis can be re-evaluated in full-scale.

Moreover, it could be useful to obtain full scale thruster characteristics for Neptunus to improve the model. In the simulation model in this thesis, no thrust losses are considered. The inclusion of thrust losses would make the model more accurate.

For increased performance and accuracy, a better observer should be implemented for CSEI. This will contribute to increase the accuracy and performance in the control algorithms, especially regarding the velocities. In addition, sensors for measuring the velocities should be evaluated for implementation. This could for instance be a *Doppler Velocity Log* (DVL).

To further account the effect of the current, adaptive control can be considered. By the implementation of adaptive update laws, the current can be estimated, and further accounted for in the control system. Current can be simulated by driving the towing wagon (where the cameras are located) with altering speed in the basin. Adaptive theory can also be used to estimate parameters for CSEI in a more accurate way.

Regarding the communication between the operator and CSEI in the MC lab, additional actions with respect to the safety could be considered. Today, the game pad controller can override the computer and tablet in case of errors or communication loss. However, no actions in case of game pad communication loss are implemented. A solution could be to implement watchdog timer functions, that safely shut down the system in case of communication loss or drift-off.

Bibliography

Antonelli, G. (2014). *Underwater Robots.* Springer. London.

Antonelli, G., Chiaverini, S., Sarkar, N., and West, M. (2001). Adaptive control of an autonomous underwater vehicle: experimental results on odin. *Control Systems Technology, IEEE Transactions on,* 9(5):756–765.

Balchen, J., Andresen, T., and Foss, B. (2003). Reguleringsteknikk. *Tapir, Trondheim, Norge.*

Barnitsas, M., Ray, D., and Kinley, P. (1981). KT, KQ and efficiency curves for the Wageningen B-series propellers.

Bennett, S. (1996). A brief history of automatic control. *IEEE Control Systems Magazine,* 16(3):17–25.

Berg, V. (2012). Development and commissioning of a DP system for ROV SF 30K. *Master thesis, Department of marine technology, NTNU.*

Bertram, V. (2008). Unmanned surface vehicles - a survey. *Skibsteknisk Selskab, Copenhagen, Denmark.*

Bindingsbø, A. U. (2013). Operation and maintenance of normally not manned (NNM) installations. In *Lecture notes: TMR4555 Driftsteknikk.*

Bjørnstad, S. (2009). *Shipshaped: Kongsberg industry and innovations in deepwater technology, 1975-2007.*

Breivik, M. (2010). Topics in guided motion control of marine vehicles. *PhD thesis, Department of Engineering Cybernetics, NTNU. Trondheim, Norway.*

Breivik, M. and Fossen, T. I. (2004). Path following of straight lines and circles for marine surface vessels. In *Proc. IFAC Conf. Contr. Appl. Marine Systems, IFAC, Ancona, Italy.*

Breivik, M. and Fossen, T. I. (2009). Guidance laws for autonomous underwater vehicles. *IN-TECH.*

Carlton, J. (2012). *Marine propellers and propulsion.* Butterworth-Heinemann, London.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.

Christ, R. D. and Wernli Sr., R. L. (2014). *The ROV Manual, A User Guide for Remotely Operated Vehicles.* Elsevier. Amsterdam.

Dahl, A. R. (2013). Path planning and guidance for marine surface vessels. *Master thesis, Department of Engineering Cybernetics, NTNU.*

D'Azevedo, E. F. (1991). Optimal triangular mesh generation by coordinate transformation. *SIAM Journal on scientific and statistical computing*, 12(4):755–786.

Det Norske Veritas (2007). Recommended practice DNV-RP-C205—environmental conditions and environmental loads. *Høvik, Norway.*

Det Norske Veritas (2014). SESAM User Manual HydroD. Accessed on March 23, 2015 `projects.dnvgl.com/Sesam/manuals/HydroD_UM.pdfl`.

Do, K., Jiang, Z.-P., and Pan, J. (2004). Robust adaptive path following of underactuated ships. *Automatica*, 40(6):929–944.

Dragland, A. (2014). *Tryggere med førerløse skip.* Accessed on March 31, 2015 `forskning.no/teknologi-marin-teknologi/2014/08/tryggere-med-forerlose-skip`.

Driscoll, F., Lueck, R., and Nahon, M. (2000). Development and validation of a lumped-mass dynamics model of a deep-sea ROV system. *Applied Ocean Research*, 22(3):169–182.

Eidsvik, O. A. N. (2015). Determination of hydrodynamic coefficients for a ROV. *Master thesis, Department of marine technology, NTNU.*

Ellingsen, I. H. (2004). Internal tides and the spread of river plumes in the Trondheim fjord. *PhD Thesis, NT Faculty, NTNU. Trondheim, Norway.*

Faltinsen, O. (1990). *Sea loads on ships and offshore structures.* Cambridge university press, London.

Follestad, J., Sandved, F., and Valle, E. (2014). Low cost ROV design, based on testing, simulations and analysis of OpenROV. *Project Thesis, Department of Marine Technology, NTNU.*

Fossen, T. I. (1991). *Nonlinear modelling and control of underwater vehicles.* PhD thesis, Department of Engineering Cybernetics, NTNU. Trondheim, Norway.

Fossen, T. I. (2011). *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons Ltd. Chichester.

Fossen, T. I. and Johansen, T. A. (2006). A survey of control allocation methods for ships and underwater vehicles. In *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*, pages 1–6. IEEE.

GeniE (2015). *Product information.* Accessed on February 2, 2015 `dnvgl.com/services/software-for-structural-analysis-of-ship-and\-offshore-structures-sesam-genie-1096`.

Graupner (2015a). Graupner 2308.65 propeller. Accessed on March 23, 2015 `graupner.de/en/products/b5d08e81-ce25-4ebd-9370-6193af870648/2308.65/product.aspx`.

Graupner (2015b). *Graupner 2358.BL product information.* Accessed on June 02, 2015 `cmc-versand.de/Graupner/Voith-Schneiderantrieb-m-Bru-arnr-6-2358.BL.html`.

Hegrenæs, Ø. and Hallingstad, O. (2011). Model-aided INS with sea current estimation for robust underwater navigation. *Oceanic Engineering, IEEE Journal of*, 36(2):316–337.

Hobbyking (2015). 2213N 800Kv brushless motor. Accessed on March 23, 2015 `hobbyking.com/hobbyking/store/__8622__2213N_800Kv_Brushless_Motor.html`.

Hval, M. N. (2012). Modelling and control of underwater inspection vehicle for aquaculture sites. *Master thesis, Department of marine technology, NTNU.*

Ickes, B. (1970). A new method for performing digital control system attitude computations using quaternions. *AIAA journal*, 8(1):13–17.

Idland, T. K. (2015). Marine cybernetics vessel CS Saucer. design, construction, and control. *Master thesis, Department of marine technology, NTNU.*

Jacobs, E. N., Ward, K. E., and Pinkerton, R. M. (1933). The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. Technical report, DTIC Document.

Johansen, T. A., Fossen, T. I., and Vik, B. (2005). Hardware-in-the-loop testing of DP systems. In *Dynamic positioning conference.*

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45.

Khalil, H. K. (2002). Nonlinear systems. *Prentice Hall, Inc. New Jersey.*

Kunz, C., Murphy, C., Camilli, R., Singh, H., Bailey, J., Eustice, R., Jakuba, M., Nakamura, K., Roman, C., Sato, T., et al. (2008). Deep sea underwater robotic exploration in the ice-covered arctic ocean with AUVs. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3654–3660. IEEE.

Loria, A., Panteley, E., Popovic, D., and Teel, A. R. (2002). An extension of Matrosov's theorem with application to stabilization of nonholonomic control systems. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 2, pages 1528–1533. IEEE.

National Instruments (2015). *Custom device framework.* Accessed on June 6, 2015
`zone.ni.com/reference/en-XX/help/372846C-01/veristandmerge/cust_device_framework/`.

National Oceanic and Atmospheric Administation (2015). Ocean. Accessed on March 30, 2015
`noaa.gov/ocean.html`.

Nomoto, K. (1957). On the steering qualities of ships. *International Shipbuilding Progress*, 4(35).

NTNU (2015). Handbook of Marine HIL simulation and Marine cybernetics laboratories. Viewed 2015-06-02
`ntnu.no/imt/lab/cybernetics`.

Oceaneering (2015). *Spectrum and Millenium product information.* Accessed on June 05, 2015
`oceaneering.com/rovs/rov-systems/`.

Oosterveld, M. W. C. and Van Oossanen, P. (1975). Further computer-analyzed data of the Wageningen B-screw series.

OpenROV (2013). *Preliminary propeller efficiency testing.* Accessed on June 6, 2015
`forum.openrov.com/t/preliminary-propeller-efficiency-testing/1686`.

OpenROV (2015). *About the OpenROV project.* Accessed on June 05, 2015
`openrov.com/about.html`.

Orsten, A. (2014). Automatic reliability-based control of iceberg towing in open waters. *Master thesis, Department of marine technology, NTNU.*

Oxford Dictionary (2015a). *Kinematics definition.* Accessed on June 04, 2015
`oxforddictionaries.com/definition/english/kinematics`.

Oxford Dictionary (2015b). *Kinetics definition.* Accessed on June 04, 2015
`oxforddictionaries.com/definition/english/dynamics#dynamics__2`.

Rensvik, E., Sørensen, A., and Rasmussen, M. (2003). Maritime industrial IT. In *Proc. 9th Int. conf. Marine Engineering Systems (ICMES), Helsinki, Finland.*

Rolls Royce (2015). *Rolls-Royce is Exploring Unmanned Ships.* Accessed on June 05, 2015
`news.filehippo.com/2014/02/rolls-royce-exploring-unmanned-ships/`.

Royce, R. (2015). *Swarm of USVs.* Accessed on June 05, 2015
`theverge.com/2014/2/25/5447278/unmanned-cargo-ships\`
`-development-rolls-royce`.

Seabotics (2015). *LBV500-5 product information.* Accessed on June 05, 2015
`seabotix.com/products/lbv300-5.htm`.

Skåtun, H. N. (2011). Development of a DP system for CS Enterprice I with Voith Schneider thrusters. *Master thesis, Department of marine technology, NTNU.*

Skjetne, R. (2005). The maneuvering problem. *PhD thesis, Department of Engineering Cybernetics, NTNU. Trondheim, Norway.*

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

Skjetne, R. and Fossen, T. I. (2004). On integral control in backstepping: Analysis of different techniques. In *American Control Conference, 2004. Proceedings of the 2004*, volume 2, pages 1899–1904. IEEE.

Skjetne, R., Smogeli, Ø., and Fossen, T. I. (2004). Modeling, identification, and adaptive maneuvering of cybership II: A complete design with experiments. In *Proc. IFAC Conf. on Control Applications in Marine Systems*, pages 203–208.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of process control*, 13(4):291–309.

Slipway (2014). *Anchor Handling Tug Aziz / Arif.* Accessed on April 27, 2015
`modelslipway.com/aziz.htm/`.

SNAME (1950). The Society of Naval Architects and Marine Engineers: Nomenclature for treating the motion of a submerged body through a fluid. In *Technical and Research Bulletin No 1-5.*

Sørensen, A. J. (2013). Marine cybernetics: Modelling and control. *Lecture Notes, UK-13-76, Department of Marine Technology, the Norwegian University of Secience and Technology, Trondheim, Norway.*

Stackpole, E. (2015). Preliminary propeller effiency testing. Accessed on January 17, 2015
`forum.openrov.com/t/preliminary-propeller-efficiency-testing/1686`.

Steen, S. (2011). Motstand og propulsjon: propell- og foilteori. *Lecture Notes, UK-2011-99, Department of Marine Technology, the Norwegian University of Secience and Technology, Trondheim, Norway.*

Stephen, B. (2014a). *Data Dashboard custom device add on.* Accessed on February 10, 2015
`decibel.ni.com/content/docs/DOC-25942/version/1`.

Stephen, B. (2014b). *Easy Custom Device Tool.* Accessed on February 27, 2015
`decibel.ni.com/content/docs/DOC-14654`.

Tran, D. N. (2014). Line-of-sight-based maneuvering control design, implementation, and experimental testing for the model ship CS Enterprise I. *Master thesis, Department of marine technology, NTNU.*

Valle, E. (2015). Marine telepresence system. *Master thesis, Department of marine technology, NTNU.*

Van Lammeren, W., Van Manen, J., and Oosterveld, M. (1969). The Wageningen B-screw series.

Appendix A

Mathematical modeling

A.1 Wageningen B-series propellers

To determine the produced thruster forces, Wageningen B-series can be used as a indication. Using the B-series, the thrust coefficient K_T can be found. K_T is a according to Carlton (2012, Eq. (6.1)) given as

$$K_T = \frac{u}{\rho n^2 D^4} \tag{A.1}$$

Further, K_T written as a function of parameters on the form (Barnitsas et al., 1981, Eq. (5))

$$K_T = f\left(J_a, \frac{P}{D}, \frac{A_E}{A_0}, Z, R_N, \frac{t}{c}\right) \tag{A.2}$$

Here, J_a is the advance ratio, $\frac{P}{D}$ is the pitch ratio, $\frac{A_E}{A_0}$ is the blade area ratio, Z is the number of propeller blades. R_N is the Reynolds number, while $\frac{t}{c}$ express the ration between the thickness and chord length of the blade section.

Using these parameters as input, a suitable Wageningen B-series figure is used to find K_T. An example of a B-series figure is seen in Figure A.1.

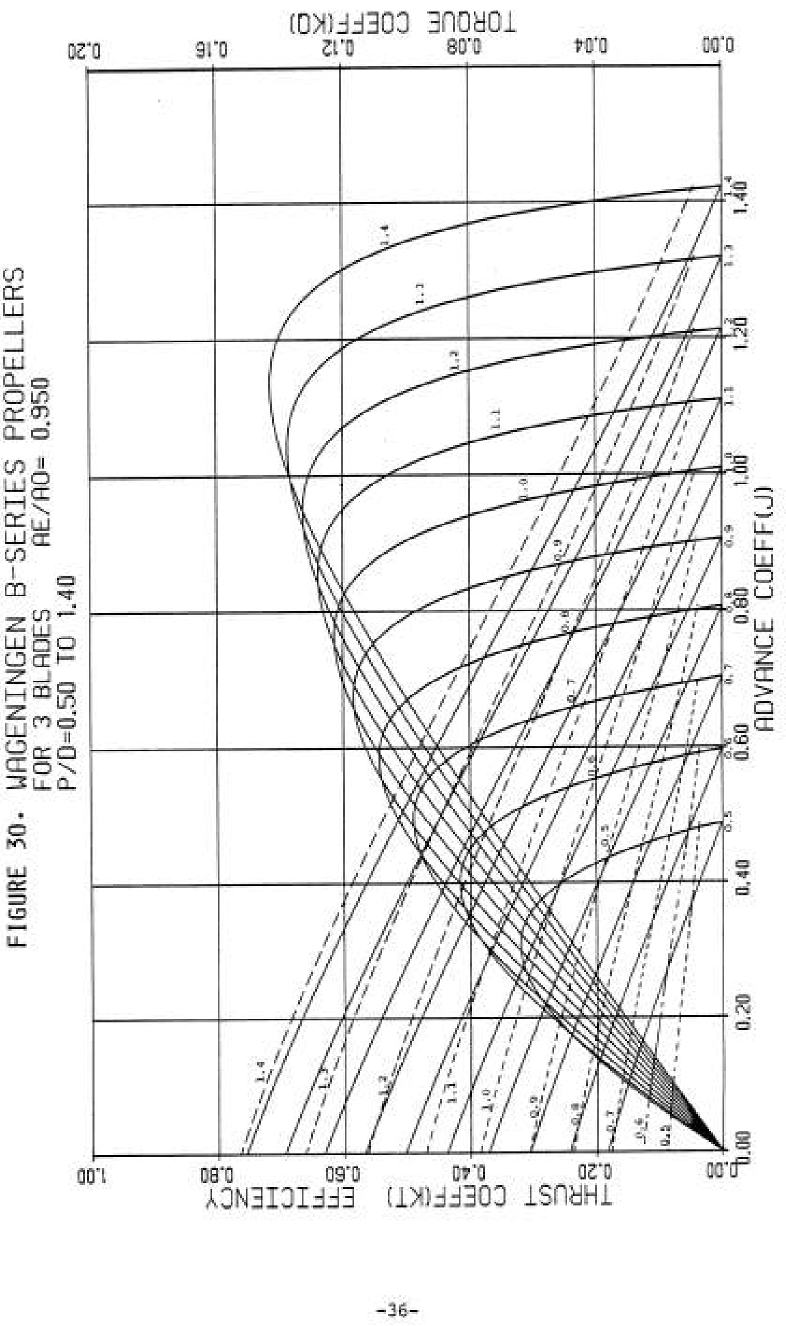


Figure A.1: Wageningen B-series propeller diagram for a propeller with 3 blades, an expanded area ratio of 0.950, and pitch / diameter ratios varying from 0.500 to 1.40. Courtesy of Barnitsas et al. (1981).

A.2 Simulation in HydroD

To run an analysis in HydroD, the Wadam Wizard tool is used. This tool leads the user through a series of 17 steps in order to set up the model and loading conditions. For non-advanced users of HydroD, this wizard is a good alternative. In the following a brief go through of the simulation set up is presented.

- **Step 1: Create/Edit Direction:** The direction of the applied loads is specified in this step. In this case, the direction is set from 0 to 360 degrees, stepping 15 degrees each iteration.

- **Step 2: Frequency Set:** The frequency can be set in terms of the period, wavelength, or frequency (all are related, and related to potential theory). Since we are dealing with a simulation of a ROV, this property is not of importance.

- **Step 3: Edit Location:** Specifies the air and water properties. Salt water is considered ($\rho = 1025\ kg/m^3$) and the water depth is set to 300 m. Air properties and kinematic viscosity of the fluid is set to default.

- **Step 4: Edit Frequency Domain Condition:** Here the specified Direction and Frequency from step 1 and 2 are chosen to governing in the simulation.

- **Step 5: Edit Hydro Model:** Here, it is chosen whether the model is fixed or free floating, in addition to the baseline z-position, *aft perpendicular* (AP), and *forward perpendicular* (FP) x-direction. For Neptunus, it is free floating, and the baseline z-direction is set to -0.5 m. AP is -0.242 m, and FP is 0.158 m.

- **Step 6: Edit Panel Model:** In this step, the meshed .GDF-model of Neptunus from Neptunus is inserted. Here, symmetry properties can be described. In addition, translation of the model to a defined point can be done in this step. For Neptunus, the two latter options is not of consideration.

- **Step 7: Create Strip Model:** Not of interest for the simulation of Neptunus.

- **Step 8: Create Bilge Keel:** Not appliciable for an underwater vehicle.

- **Step 9: Create Pressure Panels:** Select certain panels in the model, where pressure is applied. For a ROV, the pressure is approximately constant around the body. Thus, this step is neglected in this analysis.

- **Step 10: Create Load Cross Section:** Not of interest for this analysis.

- **Step 11: Create Structure Model:** Not used in this analysis. A structure model may not alter much compared to a panel model. Some differences may be present in terms of mesh size. To use a structure model in a HydroD analysis, the input must be a .FEM file.

- **Step 12: Edit Loading Condition:** In this step the distance from the ROV to the free surface in defined, in addition to a potential trim and heel angle. For the simulation of Neptunus, the distance from CO to the free surface is set to be 1 m.
- **Step 13: Edit Mass Model:** Parameters related to mass and buoyancy is calculated in this step. HydroD is not able to give the correct mass of Neptunus, due to the removal of ventilation holes in the simplified model. The correct weight of Neptunus is 3.43 kg, while HydroD calculates it to be 4.91 kg. However, the calculated mass from HydroD can be overwritten to the correct value. Since HydroD assume that the mass is homogeneous distributed, the change of weight will not affect the analysis. In step 13, the vectors from CO to CB, CO to CG, and the radii of gyrations can be found.
- **Step 14: Create GZ-curve:** Since the analysis is focusing on the behavior under the free surface, GZ-curves are disregarded.
- **Step 15: Create Wadam Offbody Points:** Not of consideration in this thesis.
- **Step 16: Edit Run:** Here changes can be done related to the previous steps, in terms of input models and loading conditions. In addition, output preferences and tolerances for the simulation can be adjusted at this step. For Neptunus, default tolerances are used.
- **Step 17: Create Analysis:** In the final step of this wizard, the analysis, based on the steps 1-16, is executed.

With the analysis of Neptunus carried out, HydroD generates a text-file with the results. From this text file, non-dimensional mass, added mass and restoring matrices can be found. This text-file is enclosed electronically to this master thesis.

```
M_A_sym          =                     []; % Symmetric added mass matrix

for i            = 1:6
    for j    = 1:6

    if i     == j

     M_A_sym(i,j) = M_A(i,j) % Avoid changes in diagonal elements

    else

     M_A_sym(i,j) = (M_A(i,j) + M_A(j,i))/2 % Take the mean of cross-coupled elements

        end
    end
end
```

Figure A.2: Matlab code for creating symmetry in the added mass matrix for HydroD. M_A is the non-symmetric matrix from HydroD.

A.3 Symmetric added mass matrix

To fulfill the property of a diagonal, and positive semidefinite added mass matrix, i.e $\boldsymbol{M}_A = \boldsymbol{M}_A^T \geq 0$, symmetry considerations are needed. Using the Matlab script in Figure A.2, the added mass matrix will fulfill the needed property.

The resulting added mass matrix, when using the Matlab code, is

$$\boldsymbol{M}_{A,\text{sym}} = \begin{bmatrix} 2.0555 & 0.5543 & 0.0103 & 0.0786 & -0.2780 & -0.0527 \\ 0.5543 & 14.6033 & -0.1634 & 2.0225 & -0.0893 & -0.8926 \\ 0.0103 & -0.1634 & 2.3295 & -0.0011 & 0.0425 & 0.0093 \\ 0.0786 & 2.0225 & -0.0011 & 0.2569 & -0.0094 & -0.1232 \\ -0.2780 & -0.0893 & 0.0425 & -0.0094 & 0.0740 & 0.0087 \\ -0.0527 & -0.8926 & 0.0093 & -0.1232 & 0.0087 & 0.1498 \end{bmatrix} \tag{A.3}$$

A.4 Towing test in MC lab

The translational and rotational velocities that the towing wagon was run with, is given in Table A.1 and A.2, respectively. The acceleration was set to constant for the runs. In translational tests, the acceleration was 0.30 m/s^2. Regarding the rotational DOF, the acceleration was set to 100 deg/s^2.

DOF	Linear drag [m/s]	Quadratic drag [m/s]
Surge	0.05	0.5
	0.10	0.6
	0.15	0.7
	0.20	0.8
	0.25	1.0
Sway	0.05	0.3
	0.08	0.4
	0.10	0.5
	0.12	0.6
	0.15	0.8
		1.0
		1.1
Heave	0.05	0.5
	0.10	0.6
	0.15	0.7
	0.20	0.8
	0.25	1.0
		1.2
		1.5
		1.8

Table A.1: The towing tests regarding translational DOF. The numbers indicate the velocities for the ROV (towing wagon).

DOF	Linear drag [deg/s]	Quadratic drag [deg/s]
Roll	1.00	10.00
	2.00	15.00
	3.00	20.00
	4.00	25.00
	5.00	
Pitch	1.00	10.00
	2.00	15.00
	3.00	20.00
	4.00	25.00
	5.00	
Yaw	1.00	10.00
	2.00	12.50
	3.00	15.00
	4.00	17.50
	5.00	20.00

Table A.2: The towing tests regarding angular DOF. The numbers indicate the velocities for the ROV (towing wagon).

A.5 Drag curves Neptunus

In this section, the drag curves for Neptunus in all 6 DOF are presented. They can in further work be used as comparison basis for adaptive estimation and control purposes, in addition to full scale test considerations.

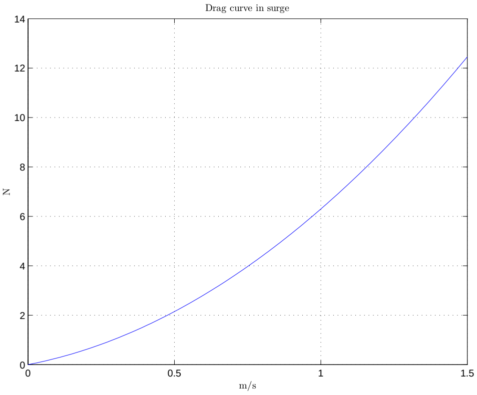


Figure A.3: Drag curve in surge.

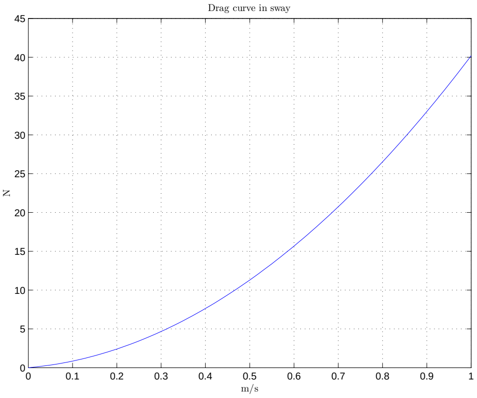


Figure A.4: Drag curve in sway.

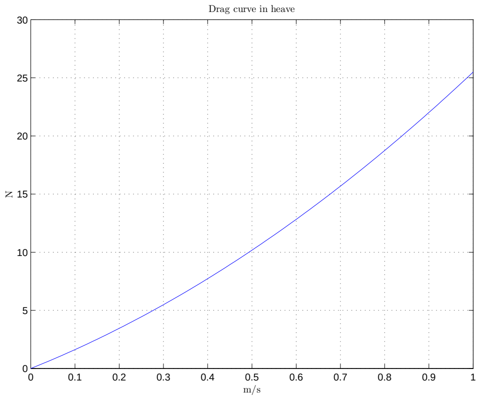


Figure A.5: Drag curve in heave.

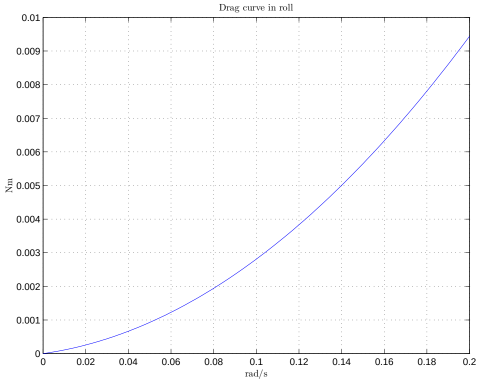


Figure A.6: Drag curve in roll.

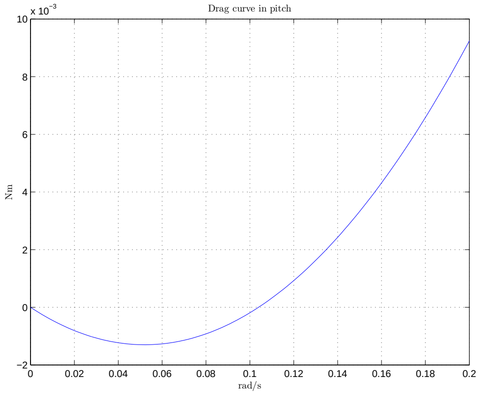


Figure A.7: Drag curve in pitch.

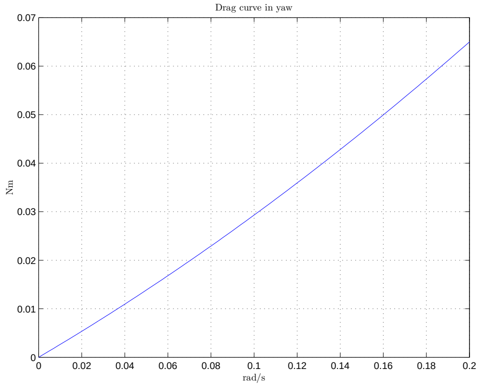


Figure A.8: Drag curve in yaw.

A.6 Neptunus 3 DOF model

Regarding the control system related to path following, a reduced DOF model will be considered. As concluded in Section 3.4, Neptunus shows passive stability properties in roll and pitch.

Kinematics

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{A.4}$$

where $\boldsymbol{\eta} = [x\ y\ \psi]^T$, $\boldsymbol{\nu} = [u\ v\ r]^T$, and

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

Kinetics

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}_{uv}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{A.6}$$

where

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A = \begin{bmatrix} 5.5155 & 0 & 0 \\ 0 & 18.0633 & -0.8926 \\ 0 & -0.8926 & 0.1897 \end{bmatrix}, \tag{A.7}$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} 2.2907 + 4.0077|u| & 0 & 0 \\ 0 & 4.9804 + 35.2159|v| & 0 \\ 0 & 0 & 0.261 + 0.320|r| \end{bmatrix}, \tag{A.8}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -14.6044v \\ 0 & 0 & 2.0555u \\ 14.6044v & -2.0555u & 0 \end{bmatrix}, \tag{A.9}$$

$$\boldsymbol{g}_{uv}(\eta) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, \tag{A.10}$$

and

$$\boldsymbol{\tau} = \begin{bmatrix} \cos(\alpha) & \cos(\alpha) & 0 \\ \sin(\alpha) & \sin(\alpha) & 0 \\ L_1 \sin(\alpha) + L_2 \cos(\alpha) & -L_1 \sin(\alpha) - L_2 \cos(\alpha) & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{A.11}$$

where L_1 is measured to be 0.17996 m, L_2 = -0.04433 m, and α = 8.9144 deg.

A.7 Thruster forces for Cybership Enterprise I

The thruster forces are calculated based on towing tests of CSEI (NTNU, 2015).

A.7.1 Characteristics and losses for bow thruster

The thrust characteristics and thrust losses are calculated to be

```
function F_y = fcn(u_BT,u)

% Thruster characteristics

if u_BT > 0.0879
   a =       2.515;
   b =      0.1174;
   c =      -2.807;
   d =      -1.131;
   y0 = a*exp(b*u_BT) + c*exp(d*u_BT);
elseif u_BT < -0.059
   a =        -548;
   b =       0.193;
   c =       548.2;
   d =      0.1992;
   y0 = a*exp(b*u_BT) + c*exp(d*u_BT);
else
   y0 = 0;
end

% Thruster losses due to forward speed

a0 =      0.7561;
a1 =      0.3075;
b1 =      -10.22;
c1 =       2.947;
F_y = y0*(a0 + a1*atan(b1*u +c1))/1.1385;
```

A.7.2 Decomposition of Voith Schneider propeller forces

The decomposition of the forces from the VSPs is calculated using

```
function [ F_x , F_y ] = fcn(alpha,u_VSP)

% Calculating the force in x-direction
```

```
a1 =       1.107  ;
b1 =       1.001  ;
c1 =       1.564  ;
a2 =      0.3471 ;
b2 =      0.1397 ;
c2 =      -1.599  ;
a3 =      0.1558 ;
b3 =       1.993  ;
c3 =       1.886  ;
a4 =      0.1599 ;
b4 =       0.218  ;
c4 =       1.513  ;
a5 =     0.08237 ;
b5 =       2.986  ;
c5 =      -1.601  ;
a6 =     0.04641 ;
b6 =       3.964  ;
c6 =       1.738  ;
F_x =   u_VSP*( a1*sin(b1*alpha+c1) + ...
               a2*sin(b2*alpha+c2) + ...
               a3*sin(b3*alpha+c3) + ...
               a4*sin(b4*alpha+c4) + ...
               a5*sin(b5*alpha+c5) + ...
               a6*sin(b6*alpha+c6) );

% Calculating the force in y-direction

a1 =      0.7936 ;
b1 =       1.001  ;
c1 =     0.01605;
a2 =      0.2112 ;
b2 =       1.997  ;
c2 =     0.03229;
a3 =     0.07608;
b3 =       2.991  ;
c3 =       2.986  ;
a4 =     0.04817;
b4 =       3.999  ;
c4 =     -0.8668 ;
a5 =     0.01757;
b5 =       4.904  ;
c5 =      -2.622 ;
a6 =    -0.02089;
b6 =       5.068  ;
c6 =      0.2548 ;
F_y =   u_VSP*( a1*sin(b1*alpha+c1) + ...
               a2*sin(b2*alpha+c2) + ...
               a3*sin(b3*alpha+c3) + ...
               a4*sin(b4*alpha+c4) + ...
```

```
            a5*sin(b5*alpha+c5) + ...
            a6*sin(b6*alpha+c6));
```

Appendix B

Control theory

B.1 Controller gains Nomoto model of CSEI

In Section 4.2.2, the yaw dynamics of CSEI defined to be treated as a first order Nomoto model, i.e

$$T\dot{r} + r = K\delta, \tag{B.1}$$

where T and K are Nomoto time constants. r is the yaw rate, while δ is the rudder angle. With $\dot{\psi} = r$, the model can be written as

$$T\ddot{\psi} + \dot{\psi} = K\delta \tag{B.2}$$

To control the heading of CSEI such that $\psi \rightarrow \psi_d$, with the presence of current, the following controller is proposed

$$\delta = -K_p(\psi - \psi_d) - K_i \int_0^t \psi(\tau) - \psi_d(\tau) d\tau \tag{B.3}$$

By Laplace transform of (B.2)

$$s^2 T\psi(s) + s\psi(s) = -KK_p(\psi(s) - \psi_d(s)) - K\frac{K_i}{s}(\psi(s) - \psi_d(s)) \tag{B.4}$$

and solving with respect to $\psi(s)$ gives

$$\psi(s) = \frac{K\left(K_p + \frac{K_i}{s}\right)}{s^2 T + s + KK_p + \frac{K_i}{s}} \psi_d(s) \tag{B.5}$$

By applying the *Final Value Theorem* for a step input $\psi_d(s) = \frac{1}{s}\psi_d$

$$\lim_{t\to\infty} \psi(t) = \lim_{s\to 0} s\psi(s) = \lim_{s\to 0} s\frac{\psi_d}{s}\left(\frac{K\left(K_p + \frac{K_i}{s}\right)}{s^2T + s + KK_p + \frac{K_i}{s}}\right) = \psi_d\frac{KK_i}{KK_i} = \psi_d \tag{B.6}$$

Even in the presence of disturbances, it is guaranteed that $\psi \to \psi_d$. Furthermore, the Nomoto model is compared to the natural mass-spring damper equation

$$\ddot{\psi} + 2\zeta\omega_n\dot{\psi} + \omega_n^2 = 0. \tag{B.7}$$

Since r is not controlled, it can be stated by comparing (B.7) and (B.2) that

$$\omega_n^2 = \frac{KK_p}{T}. \tag{B.8}$$

From Fossen (2011, Eq. (12.106)) the relation between the bandwidth ω_b and ω_n is

$$\omega_b = \omega_n\sqrt{\sqrt{2}-1} \approx 0.64\omega_n \tag{B.9}$$

Hence, the proportional controller can be defined as

$$K_p = \frac{T\omega_b^2}{0.41K}. \tag{B.10}$$

According to Fossen (2011, Pg. 374), a rule of thumb on the integral gain is

$$K_i = \frac{\omega_b K_p}{6.4}. \tag{B.11}$$

B.2 Stability of non-linear DP vehicle model with integral action

In Chapter 3, the mathematical models of Neptunus and CSEI were established. These models were further used to obtain a common 3DOF CPM, used to develop path following control systems. In Section 4.6, a maneuvering control design was established based on the CPM, which is often called the non-linear DP vehicle model.

In this appendix, the conclusion that make the origin of the dynamics UGAS is elaborated, with the use of Skjetne and Fossen (2004).

B.2.1 Stability tools

Stating (B.23) on general form gives a non autonomous system, i.e.

$$\dot{x} = f(t,x) \tag{B.12}$$

According to Khalil (2002, Lemma 4.5, Pg. 150), uniformly global stability (UGS) can be defined as

Definition 1: The equilibrium point x = 0 of (B.12) is *uniformly stable* (US) if and only if there exist a class $\mathscr{K}$ function α and a positive constant c, independent of t_0, such that

$$|x(t)| \leq \alpha(|x(t_0|), \quad \forall t \geq t_0 \geq 0, \ \forall |x(t_0)| < c \tag{B.13}$$

If (B.13) holds for any initial condition x_0, the origin is *uniformly globally stable* (UGS). Moreover, the same stability can be established using Lyapunov theory, using Khalil (2002, Th. 4.8, p. 151).

Theorem 1: Let $V : [0,\infty) \times D \rightarrow R$ be a continuously differentiable function such that

$$W_1(x) \leq V(t,x) \leq W_2(x) \tag{B.14}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t,x) \leq 0 \tag{B.15}$$

$\forall \ t \leq 0$ and $\forall x \in D$, where $W_1(x)$ and $W_2(x)$ are continuous positive definite functions on D. Then, $x = 0$ of (B.12) is uniformly stable. See Khalil (2002, p. 151) for proof. A convenient theorem for proving UGAS based on UGS systems is the *extension of Matrosov's theorem*, by Loria et al. (2002).

Theorem 2: The origin $x = 0$ of (B.12) is UGAS if

Assumption 1: The origin of (B.12) is UGS.

Assumption 2: There exist integers $j, m > 0$ and for each $\Delta > 0$, there exist

- a number $\mu > 0$.
- locally Lipschitz continuous functions V_i: : $\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^m$, i ∈ (1,...,j)
- a (continuous) function ϕ: $\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^m$, i ∈ (1,...,j)
- continuous functions Y_i: $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, i ∈ (1,...,j)

such that, for almost all $(t,x) \in \mathbb{R} \times \mathsf{B}(\Delta)$,

$$max[|V_i(t,x)|, |\phi(t,x)|] \leq \mu \tag{B.16}$$

$$\dot{V}_i(t,x) \leq Y_i(x, \phi(t,x)) \tag{B.17}$$

Assumption 3: For each integer $k \in (1,...,j)$

$$(z,\psi) \in \mathsf{B}(\Delta) \times \mathsf{B}(\mu), \ \ Y_i(z,\psi) = 0, \ \ \forall i \in (1,...,k-1) \tag{B.18}$$

implies

$$Y_k(z,\psi) \leq 0 \tag{B.19}$$

Assumption 4: For

$$(z,\psi) \in \mathsf{B}(\Delta) \times \mathsf{B}(\mu), \ \ Y_i(z,\psi) = 0, \ \ \forall i \in (1,...,j) \tag{B.20}$$

implies

$$z = 0 \tag{B.21}$$

If assumptions 1-4 hold, the origin $x = 0$ of B.12) is UGAS. With theorem 1 and 2, UGAS of (B.23) can be proven.

B.2.2 Stability of maneuvering control system

The maneuvering control design rendered the following closed loop system

$$\begin{aligned}
\dot{\boldsymbol{z}}_1 &= -\boldsymbol{K}_p\boldsymbol{z}_1 - r\boldsymbol{S}\boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{R}(\psi)^T\boldsymbol{\eta}_d^s(s)[v_s(s) - \dot{s}] \\
\dot{\tilde{\boldsymbol{\xi}}} &= \boldsymbol{z}_2 \\
\boldsymbol{M}\dot{\boldsymbol{z}}_2 &= -\boldsymbol{z}_1 - \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{R}(\psi)^T\boldsymbol{b} - \boldsymbol{K}_b\tilde{\boldsymbol{\xi}} + \boldsymbol{M}\boldsymbol{\chi}[v_s(s) - \dot{s}]
\end{aligned} \tag{B.22}$$

Assuming that the speed assignment is solved, the following dynamics are obtained

$$\begin{aligned}\dot{\boldsymbol{z}}_1 &= -\boldsymbol{K}_p\boldsymbol{z}_1 - r\boldsymbol{S}\boldsymbol{z}_1 + \boldsymbol{z}_2\\ \dot{\tilde{\boldsymbol{\xi}}} &= \boldsymbol{z}_2\\ \boldsymbol{M}\dot{\boldsymbol{z}}_2 &= -\boldsymbol{z}_1 - \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{R}(\psi)^T\boldsymbol{b} - \boldsymbol{K}_b\tilde{\boldsymbol{\xi}}\end{aligned} \tag{B.23}$$

From Lyapunov's direct method, the origin of the system is UGS. Moreover, it is assigned

$$V_1 = \frac{1}{2}\boldsymbol{z}_1^T\boldsymbol{z}_1 + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{K}_b\boldsymbol{\xi} + \frac{1}{2}\boldsymbol{z}_2^T\boldsymbol{M}\boldsymbol{z}_2, \quad V_2 = \boldsymbol{\xi}^T\boldsymbol{M}\boldsymbol{z}_2 \tag{B.24}$$

Considering that $\boldsymbol{\eta}_d(s)$ is bounded, V_1 and V_2 are bounded for bounded $(\boldsymbol{z}_1,\boldsymbol{\xi},\boldsymbol{z}_2)$. By differentiation of V_1 and V_2

$$\dot{V}_1 = -\boldsymbol{z}_1^T\boldsymbol{K}_p\boldsymbol{z}_1 - -\boldsymbol{z}_2^T\boldsymbol{K}_d\boldsymbol{z}_2 \tag{B.25}$$

$$\dot{V}_2 = \boldsymbol{z}_2^T\boldsymbol{M}\boldsymbol{z}_2 + \boldsymbol{\xi}^T\left(-\boldsymbol{z}_1 - \boldsymbol{K}_d\boldsymbol{z}_2 + \boldsymbol{R}(\psi)^T\boldsymbol{b} - \boldsymbol{K}_b\tilde{\boldsymbol{\xi}}\right) \tag{B.26}$$

Furthermore, $\dot{V}_1 := Y_1$ and $\dot{V}_2 := Y_2$. Consequently, assumption 1 and 2 are fulfilled. Next, when $\boldsymbol{\xi} \neq 0$, then if

$$Y_1 = 0 \Rightarrow Y_2 < 0, \quad Y_1 = Y_2 = 0 \Rightarrow (\boldsymbol{z}_1,\boldsymbol{\xi},\boldsymbol{z}_2) = 0 \tag{B.27}$$

Consequently, assumptions 1-4 are fulfilled, and the origin is UGAS.

Appendix C

Remote control and monitoring

C.1 Easy custom device tool

The two main groups are *strict type defined controls* and *custom device VIs*. These can be further divided into;

Strict type defined controls

- **Configuration data**: The establishment of data that are to be configured in NI VeriStand. This can be inputs or outputs to the system.
- **Channel data**: Same as for configuration data, but here data defined as channels are established. A channel can for example be data from another device, such as an iPad or an other tablet device.
- **Real-Time system data**: Includes data or references that are to be included in the VeriStand run-engine.

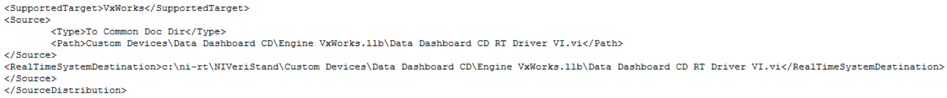
Custom device VIs

- **Init VI**: initializes the data, and set up the tasks.
- **Start VI**: starts up the tasks before the loop running the custom device begins
- **Sub VIs**: if any sub-VIs are implemented to the custom device, these must be activated before the loop starts. In this routine, the sub VIs are started from the library.

- **Execute VI**: is the main VI of the custom device. This is the routine where the main work is conducted. For every iteration, the data and channels are being updated with the latest value.
- **Stop VI**: is activated when the custom device is stopped running. This routine closes down the data references and channels, and shuts down tasks that are running.

C.2 Vx Works compatibility

In order to make the custom devices developed in LabView compatible for Vx Works, which is the operating system on cRIO, the code in Figure C.1 must be included in the XML-file.



```
<SupportedTarget>VxWorks</SupportedTarget>
<Source>
        <Type>To Common Doc Dir</Type>
        <Path>Custom Devices\Data Dashboard CD\Engine VxWorks.llb\Data Dashboard CD RT Driver VI.vi</Path>
</Source>
<RealTimeSystemDestination>c:\ni-rt\NIVeriStand\Custom Devices\Data Dashboard CD\Engine VxWorks.llb\Data Dashboard CD RT Driver VI.vi</RealTimeSystemDestination>
</Source>
</SourceDistribution>
```

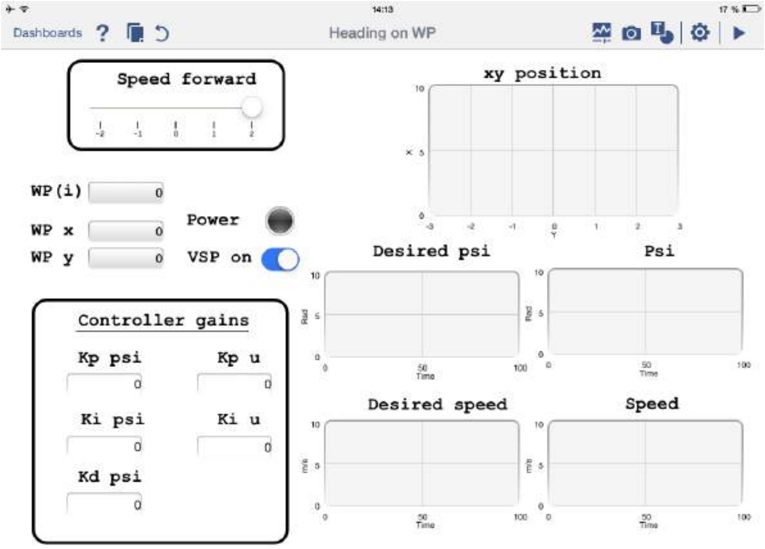
Figure C.1: Enabling for Vx Works support in the custom device.

The inclusion of this code in the XML file makes the custom device compatible for Vx Works.

C.3 iPad user interfaces

The iPad user interfaces used in the full scale tests of CSEI are provided in the sections below.

Heading on waypoints



WP(i) 0

WP x 0

WP y 0

VSP on

Kp psi 0

Kp u 0

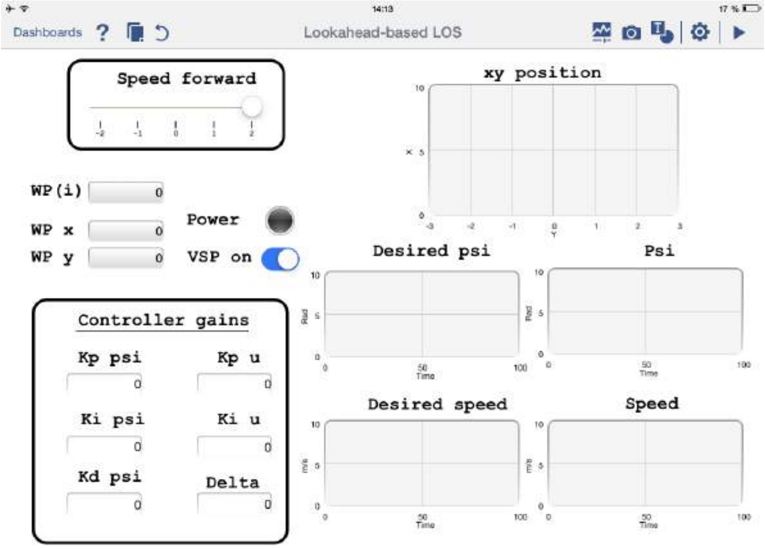
Ki psi 0

Ki u 0

Kd psi 0

Figure C.2: Heading on waypoints UI.

Lookahead-based line of sight



WP (i) 0

WP x 0

WP y 0

Power

VSP on

Controller gains

Kp psi 0

Kp u 0

Ki psi 0

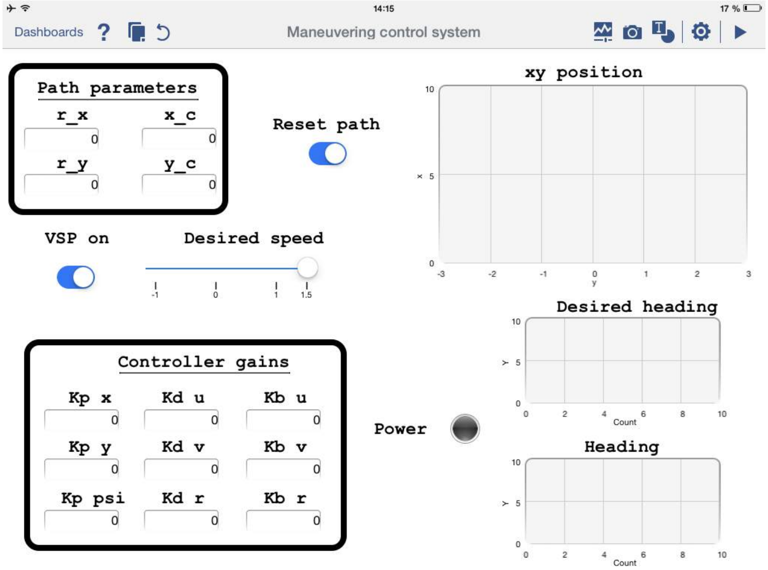
Ki u 0

Kd psi 0

Delta 0

Figure C.3: Lookahead-based LOS UI.

Maneuvering control system



Path parameters

r_x 0

x_c 0

r_y 0

y_c 0

Reset path: on

VSP on: on

Desired speed: 1.5

Controller gains

Kp x 0

Kd u 0

Kb u 0

Kp y 0

Kd v 0

Kb v 0

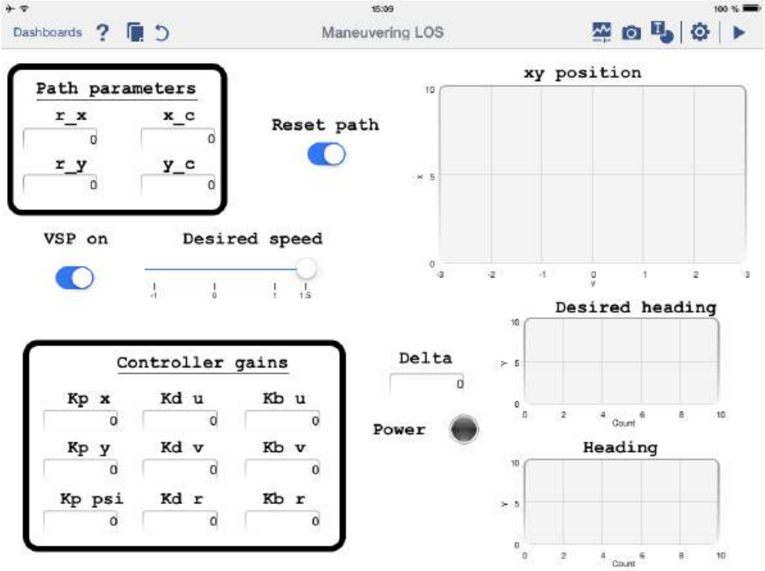
Kp psi 0

Kd r 0

Kb r 0

Figure C.4: Maneuvering by backstepping UI.

Maneuvering LOS



15:09 100 %

Dashboards

Maneuvering LOS

Path parameters

r_x 0

x_c 0

r_y 0

y_c 0

Reset path

VSP on

Desired speed

Controller gains

Kp x 0

Kd u 0

Kb u 0

Kp y 0

Kd v 0

Kb v 0

Kp psi 0

Kd r 0

Kb r 0

Delta 0

Power

Figure C.5: Maneuvering LOS UI.