Master's thesis

2019

Edvard Meyer Flaatten

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Edvard Meyer Flaatten

Autonomous Docking of Ship using Proximity Sensors

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Autonomous Docking of Ship using Proximity Sensors

Edvard Meyer Flaatten

Marine Technology
Submission date: June 2019
Supervisor: Roger Skjetne
Co-supervisor: Jon Bjørnø

Norwegian University of Science and Technology
Department of Marine Technology

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

MSC THESIS DESCRIPTION SHEET

Name of the candidate:	Flaatten, Edvard Meyer
Field of study:	Marine control engineering
Thesis title (Norwegian):	Autonom kailegging av skip ved bruk av nærhetssensorer
Thesis title (English):	Autonomous docking of ship using proximity sensors

Background

For autonomous ships, the docking and takeoff maneuvers in a harbor area are especially critical and involve complex maneuvering by a skilled ship pilot. This involves understanding of:

- the ship dynamics (inertial delays, responses to strong currents, wind gusts, and propulsion, etc.),
- the hydrodynamic effects of using propulsion and rudders near harbor structures,
- the optimal entrance paths and speed regulation for good docking maneuvers, and
- the sensors, displays, and monitoring variables to use for necessary information feedback.

The objective of this thesis is to study the problem of generating good entrance path and speed references for a ship using a conventional sensor suite combined with proximity sensors. The project shall as a start consider fully-actuated ships that have DP functionality as a basis for the maneuvering controller. Underactuated maneuvering may thereafter be considered first when the fully actuated case is well understood. The thesis shall specifically study and present the integration of proximity sensors with the path-generating and path-following algorithms.

Work description

1) Perform a background and literature review to provide information and relevant references on:
   - Practical considerations and experience related to docking of ships.
   - Proximity sensor applicable for ship docking.
   - Noise filtering of sensor signals.
   - Path planning algorithms suitable for ships.
   - MC-Lab and C/S Inocean Cat I Drillship (CSAD) model ship.

   Write a list with abbreviations and definitions of terms, explaining relevant concepts related to the literature study and project assignment.

2) Investigate and select a suitable path planning algorithm for autonomous docking. Based on this, design and implement a path-generating algorithm for docking a ship, including reference speed along the path, taking into account the practical issues to be respected, that is, docking feasibility of the path and speed. Give an overview and discuss what sensor measurements that are needed to generate the path and to perform the path-following at different stages.

3) Design and implement a control function for autodocking:
   - Create a control law able to follow the desired path to the harbor side.
   - Create a control law for maneuvering the ship close to the harbor side using proximity sensors.
   - Make an observer capable of necessary noise filtering, dead reckoning, and wildpoint rejection.
   - Consider and present specifically how the proximity sensors are integrated into the autodocking control function.

4) Simulate some autodocking scenarios using proximity sensors, on a relevant simulation model.

5) Install proximity sensors on CSAD. Implement the path-generating algorithm and autodocking control algorithms on CSAD. Document with a topology drawing of the autodocking control system.

6) Test your proposed autodocking function on CSAD in MC-Lab as a proof of concept. Present the experimental setup. Run the experiments, both in calm conditions and at least one wave condition. Analyze, present, and discuss the results.

Specifications

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

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

The report shall be organized in a logical structure to give a clear exposition of background, results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is not expected to be longer than 70 A4-pages, 100 B5-pages, from introduction to conclusion, unless otherwise agreed upon. It shall be written in English (preferably US) and contain the following elements: Title page, abstract, acknowledgements, thesis specification, list of symbols and acronyms, table of contents, introduction with objective, background, and scope and delimitations, main body with problem formulations, derivations/developments and results, conclusions with recommendations for further work, 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 an electronic copy to the main supervisor and department according to NTNU administrative procedures. The final revised version of this thesis description shall be included after the title page. Computer code, pictures, videos, dataseries, etc., shall be included electronically with the report.

Start date: January, 2019 **Due date:** As specified by the administration.

Supervisor: Roger Skjetne
Co-advisor(s): Jon Bjørnø (on CSAD issues).

Trondheim, 24.04.2019

Roger Skjetne
Supervisor

Preface

This master thesis concludes my Master's Degree and five years studying Marine Technology at the Norwegian University of Science and Technology. It was written in the spring of 2019, and is to some extent a continuation on the Project thesis carried out the fall of 2018. The topic was altered at the start of this semester, based on what was possible to do in terms of practical experiments. The thesis contains a literature review on relevant topics for the thesis and control system design with the purpose of performing autonomous docking of ships. Results have been obtained both from simulations and experiments.

Experiments were conducted in the The Marine-Cybernetics Laboratory(MC-lab) at NTNU. This proved to be challenging, time-consuming and very frustrating at times, but in the end very rewarding. The learning outcome has been great, especially concerning use and installation of hardware and control system design. Furthermore, it has become clear how difficult it is to make things work in experiments compared to simulations.

Preferably the reader should have knowledge about marine cybernetics, control theory and hydrodynamics before reading this thesis.

Edvard Meyer Flaatten

Edvard Meyer Flaatten

Trondheim, June 25, 2019

Acknowledgements

I have received much help throughout the work with my thesis, and there are several persons I would like to thank.

First I would like to thank my supervisor Prof. Roger Skjetne for guidance during the thesis and for being there to answer my questions when it was needed.

My co-advisor, Jon Bjørnø, has also been of great help. Regarding the experiments and issues concerning the C/S Inocean Cat I Drillship(CSAD) model ship, his competence was invaluable.

Furthermore, I would like to thank Torgeir Wahl, for helping me with instaling the proximity sensors on CSAD and challenges with Qualisys Tracking Manager and Oqus. I would also like to thank Trond Innset who helped me construct a wooden dock for the experiments.

Fellow master students Aksel Knudsen Nordstoga, Håvard Snefjellå Løvås and Sondre Haug, whom I have shared many hours in the MC-lab with, also deserve a thank you. All of us used CSAD for our experiments, and the collaboration with them has been great.

Last but not least, i would like to thank all my fellow students in the class of 2019 and especially office C1.062. Thank you for all the memories.

Abstract

This thesis presents the design of a motion control system for autonomous docking of ships using proximity sensors. Simulations and experiments have been carried out using the C/S Inocean Cat I Drillship model vessel. The primary focus of the thesis is on how to integrate the proximity sensors into the control system, but also how to generate suitable entrance paths and velocity references for a docking operation. The proximity sensors are used to measure the distance between the ship and the side of the quay, in addition to the relative orientation of the ship compared to the quay.

The docking problem is separated into two phases. The first phase is the path following phase, where the ship is to follow a pre-generated path to a position just outside the quay. The second phase is the docking phase where the proximity senors are used to maneuver the ship to a position and orientation where it is parallel to the quayside with a reference distance between the ship and the quay. For the path following phase, a path following algorithm has been designed to make sure that the vessel moves faster in the beginning and slows down as it approaches the last waypoint of the path outside the quay. When it reaches the last waypoint the velocity is approximately zero. For the docking phase the guidance system is designed to make sure the vessel is taken slow and steady close to the quay.

For the simulations, two methods are proposed and tested. These two methods are identical for the path following phase, but they differ for the docking phase. For the docking phase the first method, called method 1, uses the proximity sensor measurements directly in the controller to control the heading and sway motion of the vessel. For the second method, called method 2, the proximity sensor measurements are integrated into the guidance system for the docking phase. The results from the simulations showed that method 2 was the superior method, as method 1 struggles when the current becomes too strong.

For the experiments, only method 2 was tested due to its better performance. The experiments were conducted in the Marine Cybernetics Laboratory at Norwegian University of Science and Technology. The results are promising. The vessel follows the desired position both for the path following phase and the docking phase, and the the ship is maneuvered slow and steady to the desired position and heading right next to the harbor. However, the control of the heading of the vessel could have been better.

Two proximity sensors that use sound waves, where installed on C/S Inocean Cat I Drillship. Furthermore, a dock was constructed. This was done to make it possible to perform autonomous docking experiments in the Marine Cybernetics Laboratory, but also to facilitate for further research regarding docking of ships in the laboratory.

Sammendrag

Denne avhandlingen presenterer utformingen av et kontrollsystem for autonom kailegging av skip ved hjelp av nærhetssensorer. Simuleringer og eksperimenter har blitt utført med moddelfartøyet C/S Inocean Cat I Drillship. Hovedfokuset for avhandlingen er på hvordan man integrerer nærhetssensorene inn i kontrollsystemet, men også hvordan man genererer egnede baner og hastighetsreferanser for en kaileggingsopreasjon. Nærhetssensorene brukes til å måle avstanden mellom skipet og kaien, i tillegg til skipets relative orientering i forhold til kaien.

Selve kaileggingensoperasjonen er delt i to faser. Den første fasen er banefølging hvor skipet skal følge en forhåndsgenerert bane til en posisjon like utenfor kaien. Den andre fasen er angår selve kaileggingen hvor nærhetssensorene brukes til å manøvrere skipet til en posisjon og retning(heading) hvor den er parallell med kaien og med en referanseverdi mellom skipet og kaien. For banefølgingsfasen, er en banefølgings-algoritme designet for å sørge for at fartøyet beveger seg raskere i begynnelsen og bremser ned når det nærmer seg slutten av banen. Når fatøyet ankommer det siste veipunktet på banen, skal hastigheten være omtrent null. For kaileggingsfasen er kontrollsystemet utformet for å sikre at fartøyet tas sakte og kontrolert inntil kaien.

For simuleringene er to metoder foreslått og testet. Disse to metodene er identiske for banefølgingsfasen, men de er forskjellige for kaileggingsfasen. For kaileggingsfasen bruker den første metoden, kalt metode 1, nærhetssensormålingene direkte i kontroleren for å styre fartøyets posisjon og orientering. For den andre metoden, kalt metode 2, er nærhetssensormålingene integrert i styringssystemet for dockingsfasen. Resultatene fra simulasjonene viste at metode 2 var den beste metoden, ettersom metode 1 sliter når strømmen blir for sterk.

For de eksperimentelle forsøkene ble bare metode 2 testet fordi den ga best resultater i simulerignene. Forsøkene ble utført i Marin Kybernetikk Laboratoriet ved Norges teknisk-naturvitenskapelige universitet. Resultatene er lovende. Fartøyet følger den ønskede posisjonen både for banefølgingsfasen og kaileggingsfasen, og skipet blir manøvrert sakte og rolig til ønsket posisjon og heading intil kaisiden. Kontrollen av fartøyets retning(heading) kunne imidlertid vært bedre.

To nærhetssensorer som bruker lydbølger, ble installert på C/S Inocean Cat I Drillship. I tillegg ble det laget en kai. Dette ble gjort for å gjøre det mulig å utføre de eksperimentelle forsøkene i Marin Kybernetikk Laboratoriet, men også for å legge til rette for videre forskning på kailegging av skip i laboratoriet.

Contents

List of Figures

List of Tables

Abbreviations

AWACS 2 Active Wave Absorption Control System. 3

cRIO compact Reconfigurable Input/Output. 38

CSAD C/S Inocean Cat I Drillship. 3, 11

DHI Institute for Water and Environment. 3

DOF Degrees of freedom. 3, 16

ESC Electronic Speed Controller. 38

GNSS Global Navigation Satellite System. 3

IMU Inertial Measurement Unit. 4

MC-Lab Marine Cybernetics Laboratory. 2, 3

MIMO Multiple-Input and Multiple-Output. 22

NED North-East-Down. 16

NI National Instruments. 38

PID Proportional-Integral-Derivative. 22

PM Pierson-Moskowitz. 19

QTM Qualisys Track Manager. 3

ROS Robot Operating System. 3

RPi Raspberry Pi. 38

TAPM Thruster-Assisted Position Mooring. 3

UGAS Uniform Global Asymptotically Stable. 26

UGES Uniform Global Exponentially Stable. 26

UUB Uniformly ultimated bounded. 26

WF Wave-frequency. 19

Nomenclature

B Beam of ship. 13, 20

H Wave height. 3

H_s Significant wave height. 3

K_w Gain constant for waves. 19

LOA Length overall of ship. 13

N Moment acting on a vessel in body-fixed reference frame in yaw. 17

X Force acting on a vessel in body-fixed reference frame in the surge direction. 17

Y Force acting on a vessel in body-fixed reference frame in the sway direction. 17

Δ Mass displacement. 37

Δ_p Slope gain for docking path. 34

Δ_u Slope gain for path following. 32

$\boldsymbol{C}$ Coriolis matrix. 17

$\boldsymbol{C}_A$ Hydrodynamic Coriolis and centripetal matrix. 18

$\boldsymbol{C}_{RB}$ Rigid-body Coriolis and centripetal matrix. 18

$\boldsymbol{D}$ Damping matrix. 18

$\boldsymbol{K}$ Tunable wave forces gain matrix

. 19

$\boldsymbol{K}_1$ Observer gain matrix. 26

$\boldsymbol{K}_3$ Observer gain matrix. 27

$\boldsymbol{K}_4$ Observer gain matrix. 27

$\boldsymbol{K}_d$ Derivative gain matrix. 22

$\boldsymbol{K}_i$ Integral gain matrix. 22

$\boldsymbol{K}_p$ Proportional gain matrix. 22

$\boldsymbol{M}$ Mass matrix. 17

$\boldsymbol{R}(\psi)$ Rotation matrix from NED to Body. 16

$\boldsymbol{\eta}$ Position and orientation vector. 16, 17

$\boldsymbol{\eta}_d$ Desired position vector. 21

$\boldsymbol{\eta}_p$ Vessel parallel coordinates. 22

$\boldsymbol{\nu}$ Linear and angular velocity vector decomposed in the body-fixed reference frame. 16, 17

$\boldsymbol{\nu}_c^b$ Velocity of the ocean current expressed in the body-fixed reference frame. 19

$\boldsymbol{\nu}_c^n$ Velocity of the ocean current expressed in the basin-fixed reference frame. 19

$\boldsymbol{\nu}_d$ Desired velocity vector. 21

$\boldsymbol{\omega}_{b/n}^b$ Angular velocity in body-fixed coordinate frame. 16

$\boldsymbol{\tau}_{wave1}$ First-order wave-induced forces. 19

$\boldsymbol{\xi}$ Wave frequency motion vector. 26

$\boldsymbol{v}_{b/n}^b$ Linear velocity in body-fixed coordinate frame. 16

λ Damping coefficient. 19

$\boldsymbol{\Theta}_{nb}$ attitude in Euler angles. 16

$\mathbf{p}_{b/n}^n$ Position in the NED coordinate frame. 16

ω_0 Dominating wave frequency. 19

ω_b Control bandwidth. 23

ω_c Cutoff frequency. 35

ω_n Natural frequency. 23

σ Wave intensity. 19

σ_ψ Activation function for heading correction. 33

σ_p Activation function for initiating the docking phase. 34

ζ Relative damping ratio. 23

a^b Body fixed commanded acceleration. 24

c Speed of sound. 2

g Gravitational acceleration. 20

l_1 Distance measured from proximity sensor 1. 14

l_2 Distance measured from proximity sensor 2. 14

l_d Desired distance between the ship and harbor side

. 23

l_{ref} Reference distance between the ship and the dock. 35

s_1 Path parameter for the path following. 31

s_2 Path parameter for the docking. 33

t_r Return time of sound wave. 2

u_c Velocity of the ocean current expressed in the body-fixed reference frame in the x-direction. 19

u_{dock} Desired speed along docking path. 34

u_d Desired speed along a path. 32

v_c Velocity of the ocean current expressed in the body-fixed reference frame in the y-direction. 19

v_s Speed assignment for the path parameter s_1. 32

w Gaussian white noise. 14

D Draught. 37

T Draft. 37

Chapter 1

Introduction

1.1 Motivation

The president of Rolls Royce Marine Mikael Mäkinen says in the report from Rolls Royce Marine (2016) that "Autonomous shipping is the future of the maritime industry. As disruptive as the smartphone, the smart ship will revolutionise the landscape of ship design and operations". In Stensvold (2018a) it was recently made clear that Wärtsilä has taken big steps towards an autonomous ferry, with the successful testing of their system on the ferry Folgefonn. The system made it possible for the ferry to perform docking and take-off autonomously in addition to following and planning of an optimal route between ports. Another project which is set to be done in 2020 is the autonomous container feeder Yara Birkeland, where Kongsberg is responsible for development and delivery off all the key enabling technologies (Stensvold, 2018b).

An important piece of the puzzle that is an autonomous ship, is the docking and take off phase. For ships, the docking and takeoff maneuvers in a harbor environment can be risky and critical and involves complex ship maneuvering by a skilled ship pilot. Making this operation autonomous has the potential of reducing the risk and increase the efficiency.

1.2 Background

The purpose of this section is to perform a literature review and provide information and references on relevant topics for the thesis. This thesis is a continuation of the project thesis Flaatten (2018) with the same topic. Although, the topic has been altered from the project thesis, based on what was found out and what is possible to do in the MC-lab.

1.2.1 Proximity sensor applicable for ship docking

A proximity sensor detects objects in its vicinity without any physical contact. There exists many types of proximity sensors and they utilize different methods of sensing. Examples of proximity sensors are:

- Sonar
- Inductive
- Doppler
- Infrared
- Capacitive
- Inductive
- Radar

An acoustic proximity sensor was used by Spange (2016) for autonomous docking in the Marine Cybernetics Laboratory (MC-Lab). It was used to compliment a lidar sensor. The proximity sensor was used in the autonomous mapping process, and detected obstacles outside of the lidars scan-plane.

An acoustic proximity sensor, sends out a wave of sound and measures the time it takes for it to return. Based on the return time, the distance between the sensor and the object the wave has been reflected by is calculated using the following equation:

$$d = \frac{ct_r}{2} \tag{1.1}$$

where c is the speed of sound and t_r is the return time of the sound wave.

1.2.2 Noise filtering of sensor signals

This section will present some possible methods for filtering a signal from sensors such as proximity sensors. Noise is undesired changes a signal is exposed to throughout capture, storage, transmission, processing, or conversion. In other words, noise can be said to be anything that is not the true signal, which contains only the desired sensor information. Possible methods for filtering noise are:

- **Low pass filter:** Removes high frequencies and leaves the low frequencies of a signal.
- **High pass filter:** Removes low frequencies and leaves the high frequencies of a signal
- **Band pass filter:** Leaves only a limited range of frequencies.
- **Gaussian smoothing:** Cuts of the high frequency components of the frequency spectrum

When introducing a noise filter, the signal will undergo a phase drop. Dessen (2018) presents a toolkit for designing filters in feedback loops with the goal of minimizing the phase drop of the signal.

1.2.3 MC-Lab

The MC-Lab is operated by the Department of Marine Technology at NTNU (NTNU, 2019). It is mainly used by master and PhD-candidates. It can also be used by external users. The lab is equipped with an advanced instrument package, making it suitable for testing of motion control systems. The dimensions of the tank is 40m x 6.45m x 1.5m (L x B x D). The relatively small size of the lab is another advantage when testing motion control systems as, among other things, the ship models are easy to deploy and retrieve from the water. Its small size also limits what maneuvers that are possible to perform.

The lab is also equipped with a towing carriage making it possible to carry out hydrodynamic tests as well. The towing carriage has the capability of performing precise 6 Degrees of freedom (DOF) movements of the models. The towing carriage can be controlled manually from the console atop the carriage itself, or it can be controlled using a computer.

Matlab/Simulink and Opal are the primary software programs used in the lab, but also Robot Operating System (ROS) is used for some of the models. Onboard the model vessels a QNX real-time operating system runs on the target computer, and a host PC presents the desired data and results real-time with LabView.

The lab emulates a Global Navigation Satellite System (GNSS) with a software called Qualisys Track Manager (QTM) and Oqus cameras. Qualisys is a supplier of systems capable of capturing motion and analysis of the motion data. The system uses optical capture technology to obtain 3D and 6DOF motion measurements. The vessel is equipped with a number of silver spheres, which the Oqus cameras pick up, and by method of triangulation the system is able to capture 6 DOF position data.

The tank also has a wave maker, which according to NTNU (2019), has the following capabilities:

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

The wave maker consists of a servo actuator, which drives a six meter wide paddle. It uses Active Wave Absorption Control System (AWACS 2) and has a Institute for Water and Environment (DHI) wave synthesizer which can produce both regular and irregular waves.

1.2.4 C/S Inocean Cat I Drillship model ship

C/S Inocean Cat I Drillship (CSAD) is a model of the Statoil Cat I Arctic Drillship, in the scale 1:90 (NTNU, 2017). It was built and instrumented in 2016. The main purpose of the model is to serve as a platform for Thruster-Assisted Position Mooring (TAPM) research. The model has three azimuth thrusters in the aft as well as three in the fore. It also has a moon-pool in the middle of the model for turret and mooring lines.

Bjørnø (2016) describes how the model was constructed, equipped and assembled. In addition Bjørnø (2016) proposes and developed a TAPM system for the vessel. Other relevant literature includes Frederich (2016), where a constrained optimal thrust allocation algorithm is developed and tested on the ship model. Udjus (2017) concerned force field identification and position control of CSAD using Inertial Measurement Unit (IMU).

1.2.5 Practical considerations and experience related to docking of ships

This section will concern regular docking i.e. docking which happens non-autonomously with a human pilot. It will address how the rudders and propellers are used when a ship is to dock and what the most common causes for accidents when docking. Much of this section will be based on Murdoch et al. (2012), where the five golden rules of safe docking are presented as follows:

1. Slow speed
2. Controlled Approach
3. Planning
4. Team work
5. Checking Equipment

These rules are based on the most common accident causes for docking.

Accident causes

The majority of docking accidents occur because of simple mistakes made by an individual, and very often it is due to high speed. Lack of communication on the bridge is also a cause that often comes up. Sometimes it is not clear who is in charge of the operation and who should execute the maneuvering commands and very often accidents happen because of operator error. Murdoch et al. (2012) mentions a case where the master of the ship sailed directly towards the dock with the plan of dropping the anchor to reduce the speed. The approach speed was too high and the ship hit the quayside as the ship could not be controlled. In another case the helmsman was given orders by the pilot to keep starboard helm but executed port helm instead, leading to the ship hitting the dock. Equipment failure is also critical. Examples of equipment failure can include interruption of the power supply and freezing of the motor settings. Insufficient knowledge of the ships characteristics such as turning radius and how the ship reacts to different rudder and thrust commands, also often leads to accidents. Failure to allow for environmental forces like wind and current is also often the cause of accident scenarios. Docking requirements are individual for each dock, and failure to understand the docking requirements can have devastating consequences. For example if the angle of approach is too shallow, the risk of hitting the dock increases. It is also important to know what is best when deciding if you want the stern, bow, port side or starboard side to the dock.

Many of these things can be avoided by introducing autonomous docking. Most noticeably the human operator failure and errors based on insufficient communication. A proper planning phase will still be needed and of course checking if the equipment is up and running and working properly is also necessary. An autonomous docking system should be able to perform a controlled approach towards the dock with sufficiently low speed and accounting for possible environmental forces without any human operator.

The pivot point and lateral motion

The pivot point is the point which the ship turns about in yaw. In Fossen (2011) it is defined as a point on the centerline measured from the CG about which the ship turns. Consequently it has no sideways movement. The ship rotates about this point when a moment is applied to the ship like a rudder or thrust action. For a turning ship the pivot point will change position depending on the magnitude and position of the applied forces creating the yaw moment.

The pivot point is rarely located at the ships longitudinal center, thus the ship will move laterally when turning. Knowledge about the ships lateral movement when turning is essential when maneuvering the ship close to hazards. A lateral movement to starboard will occur, when the ship is moving forward and turning to port. And vice versa when moving forward and turning to starboard, a lateral movement to port will be introduced.

Rudder

A rudder is a foil that pivots about a vertical axis. Rudders are dependent on water passing over it to create lift. The placement of the rudder is usually at the stern of the ship to take advantage of the forward pivot point, and make the distance between them longer to enhance the moment it creates. The main propeller(s) and the ship moving through water helps create a water flow over the rudder. If there is no water flow over the rudder, the ship will be unable to make use of the rudder for maneuvering and control of the ship can be lost. A slow or disturbed flow over the rudder will reduce its effectiveness and yield a poor response from the ship to rudder actions.

Thrusters

The two most common thrusters are bow thrusters and azimuth thrusters. They can work as an alternative to the rudder and can be used to initiate and maintain a turn. Bow thrusters can be placed in the bow or the stern and work laterally. Lateral thrusters are most effective when the ship has no forward(or backward) speed and will loose much of its effectiveness between 2-5 knots. When the ship reaches this speed the slipstream from the thruster will merge with the general flow around the hull of a forward moving ship When used they will create a lateral force at their position, creating a moment about the pivot point. The magnitude of the moment and effectiveness depends on the distance between the thruster and the pivot point. According to Murdoch et al. (2012) it is important, when the ship only has one bow thruster, to not be too focused on the position of the bow when docking. The main focus should be to get the stern in place. The bow can later be controlled with the bow thruster. Bow thrusters are best used when the ship is to dock or take-off with ship side to the dock. Modern ships fitted with one or several lateral thruster, can usually dock without the help of tugs.

Transverse thrust is a phenomenon which is very important to know about for a master of a ship. Transverse thrust is the tendency a forward or backwards running propeller to move the stern or bow laterally. The reason for transverse thrust is the interaction between the hull, propeller and rudder. For example for a ship with a propeller spinning to the right for forward speed, will have a small inclination towards the bow moving to the port side. The transverse thrust will vary with water depth and speed, and it is important for the master of the ship to be aware of this effect. It can also be used to the masters advantage when maneuvering the ship, as it can help bring the ship up alongside the quayside when docking.

Forward speed, approach and kick ahead

As mentioned earlier, the most important thing when docking a ship is to have sufficiently low approach speed. The ship speed should be kept at the minimum level required to maintain control of the ship and not loosing the effect of the rudder. A critical moment is when the ship is to stop. This is done by setting the engine/propeller to work astern. This can be done by changing the direction of revolution for the propeller or the pitch on the blades. But when a propeller starts to rotate astern the flow over the rudder is disrupted and the effect of the rudder is unpredictable and unresponsive.

A possible way of dealing with the ship not responding to rudder actions is called kick ahead. This involves putting the engines ahead for a short burst of power to help create a flow over the rudders to initiate and/or maintain a turn. The goal is to increase the turning effect of the rudder without increasing the speed of the ship. If kicks are performed too frequently the speed will eventually increase. Kicks can also be performed by setting the engine astern. When doing the kick ahead, the angle of the rudder should be at its max limit if maximum steering force is wanted. It is also worth noting that if the angle of the rudder is not at the maximum limit, a bigger portion of the power will go to increasing the forward speed instead of help the turning. In addition it is important to reduce the engine power before the rudder angle is reduced again for the same reason.

Docking procedures

Different tools like tugs and anchors can be used to aid in the docking process. In this thesis, these tools are not considered. The docking procedures presented in this section regards ships that has actuators in the form of main propeller, rudder and bow thruster(s). In this case it is possible to distinguish between two situations in terms of actuator use. The first situation is when the engine and rudder are used alone. It is important to be aware of the centrifugal force causing lateral movement when turning. This force can be used to the pilots advantage in docking scenarios. The other actuator case is the use of thrusters and only thrusters. When turning using only thrusters, it will simply move the bow to port or starboard without any lateral motion.

Figure 1.1 illustrates port-side docking. The straight black arrows represent the lateral motion of the ship when turning to starboard. the curved arrows represent the direction of the turning yaw moment. The point ***P** is the pivot point.

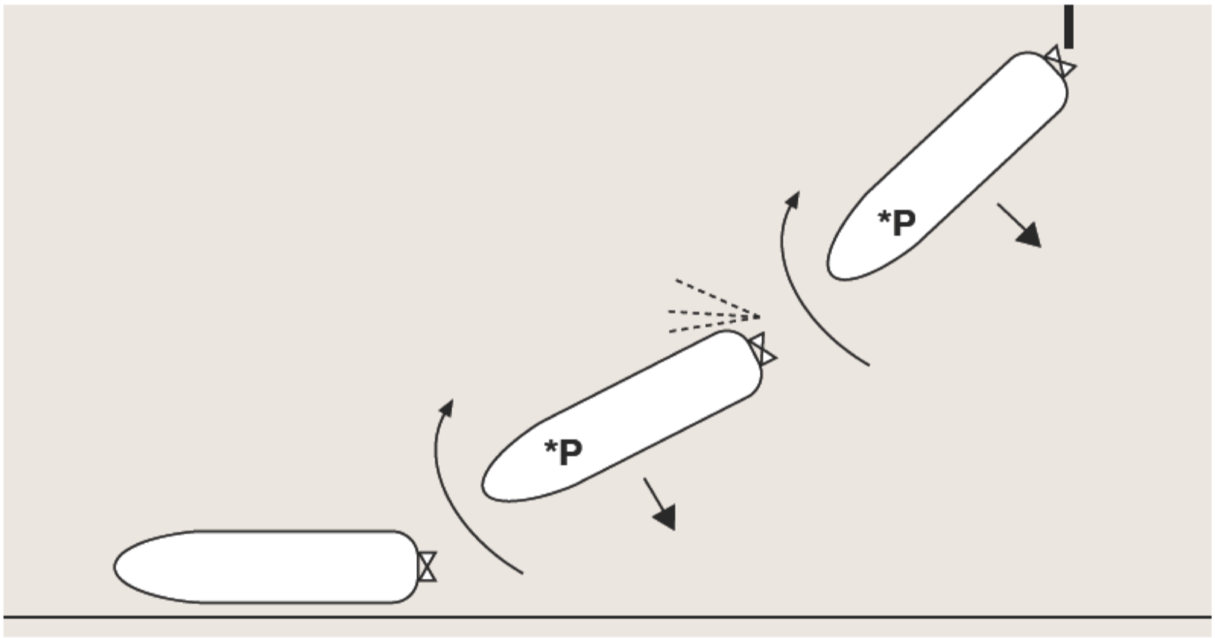


Figure 1.1: Port-side docking. [Courtesty: Murdoch et al. (2012)]

The ship should approach the quayside at an angle. Astern thrust will be used to stop the ship in addition to rotate the ship by bringing the stern to port. The transverse thrust will be the reason for the stern motion to port. When parallel to the dock, the ship can use power astern to create more transverse thrust and kicks ahead with suitable rudder angles to maneuver into the desired position. If the ship starts going backwards, the lateral motion will change to starboard which can be useful in certain situations.

The approach speed is probably the most crucial thing to get correct when docking. The speed should be selected so that it is balanced by the astern power needed to stop the ship. The greater the forward speed, the greater the astern power needed to stop the ship and the greater the transverse thrust is created. This can lead to the stern hitting the dock and/or the bow moving away from the dock side.

Often it is required to dock between two ships. Figure 1.2 illustrates this scenario. Many ship require the use of tugs when docking between two other ships.

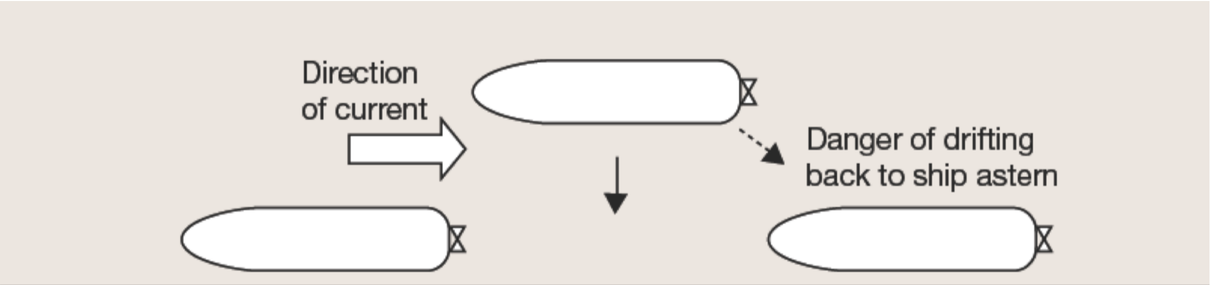


Figure 1.2: Docking between two ships. [Courtesty: Murdoch et al. (2012)]

In this scenario it is especially important to consider the current. The slower the speed of the ship is, the more it is affected by the current. A lack of respect for the current can make the ship collide with one of the ships or the quayside. When docking between two ships, the usual procedure starts with positioning the ship outside of the two other ships at the desired stern and bow position. The next step is to slowly move the ship in between the other ships keeping the ship parallel to the quayside, using thrusters. The other approach is to bring the stern or bow in first and then, bring the remaining end in afterwards.

1.3 Objective of the Thesis

The objective of the thesis is to study the docking problem of generating a suitable entrance path and speed references for a ship using a conventional sensor suite combined with proximity sensors. It is of special interest how the proximity sensors are integrated into the systems responsible for the path-following and path-generation. Firstly the problem should be studied for a fully actuated ship, and the case of an underactuated ship can be considered when the fully actuated case is properly understood.

Another objective is to facilitate for further research regarding autonomous docking of ships in the MC-lab. This mainly involves the installation of proximity sensors on CSAD and the construction of a dock, but also implementation of control system software on CSAD.

1.4 Problem Description

The problem to be solved in this thesis, is to make CSAD safely dock autonomously. The problem concerns entering a quayside and perform a docking operation, and it can be separated into two phases:

- **Phase 1:** Maneuver along a pre-generated path where the final waypoint of the path will be just outside the quayside. The ship approaches the last waypoint of the path approximately parallel to the quayside.
- **Phase 2:** Is the docking operation itself, which involves maneuvering the ship to a position and orientation where it is parallel to the quayside at a reference distance between the side of the ship and harborside.

A human operator will specify the desired docking location, while the autonomous docking system will handle the rest. This means that the location of the dock must be known by the operator. The following assumptions have been made for the problem:

1. Static environment, meaning the environmental forces will remain the same(constant variance and mean of the process) throughout the docking operation. In addition there will be no stationary obstacles or dynamic obstacles, e.g. other vessels, in the way of the ship.
2. The problem concerns a fully actuated ship.
3. When the autodocking process is initiated, starting with the path following, it is assumed that the ship is in a reasonable position relative to the desired docking location. Reasonable position means a position where a regular/manual docking operation would start from.
4. The measurements available are of the position and orientation of the ship and measurements from the proximity sensors. There will be two proximity sensors installed on the ship, one in the front and one in the aft, and located on the side of the ship which will be the closest to the quay.

The system will be tested and simulated using Matlab/Simulink on a computer, followed by practical experiments in the MC-Lab using CSAD.

1.5 Contribution of the Thesis

The contributions of the thesis comes both in the form of a software control system and hardware, and it can be summed up by the following points:

1. Two methods for autonomous docking using proximity sensors.
2. Installing proximity sensors on CSAD.
3. Constructing a dock which can be used in the MC-lab.

The installation of the proximity sensors on CSAD and the construction of the dock, allows for further experiments and testing on the topic of autonomous docking in the MC-lab. This will probably be most relevant for master theses to come, but possibly also PhD research.

1.6 Outline of the Thesis

1.6.1 Structure

Chapter 1 is the introduction. The reader is introduced to the problem and relevant background information. The problem formulation, objective and contribution of the thesis also finds place in this chapter.

Chapter 2 is the mathematical modeling. This chapter describes a 3 degree of freedom maneuvering model of CSAD, modeling of environmental forces and proximity sensors.

Chapter 3 concerns the controller and observer design. Different controllers are presented as well as the observer used.

Chapter 4 addresses the guidance and path generation for the docking operation.

Chapter 5 presents the experimental setup. Information about the MC-lab and CSAD is in this chapter. In addition the installation of the proximity sensors on the model vessel and the construction of a wooden dock is covered in this chapter.

Chapter 6 is the results. Here the results from the simulations as well as the experiments are presented.

Chapter 7 is the discussion. In this chapter mainly the results are discussed but also other relevant issues that have emerged throughout the thesis.

Chapter 8 is the concluding remarks and summarizes the main contributions of the thesis.

Chapter 9 proposes some possible continuations on the work done in this thesis.

Appendix A contains the matlab code for the modeling of the proximity sensors.

Appendix B contains a QR-code and URL leading to a video of the experiment carried out in the thesis.

1.6.2 Notation

For vectors and matrices the bold style font has been used. One dimensional scalars are in regular style font.

Chapter 2

Mathematical Modeling

This chapter will describe the mathematical modeling of the vessel CSAD, including the kinetics and kinematics, and the modeling of the proximity sensors and environmental loads.

2.1 Modeling of Proximity Sensors

This section will explain, how the modeling of the proximity sensors have been implemented for the simulations in Simulink. The two sensors are placed on the port side of the ship. The foremost sensor is denoted sensor 1 and the sensor in the aft of the ship is denoted sensor 2. The coordinates of their location in the body fixed reference frame is set to be

$$(x^b_{s,1}, y^b_{s,1}) = (\frac{LOA}{4}, -B/2) \tag{2.1}$$

$$(x^b_{s,2}, y^b_{s,2}) = (-\frac{LOA}{2}, -B/2) \tag{2.2}$$

where $LOA = 2.578m$ and $B = 0.440m$. This implies that the distance d between the two sensors will be 1.9335 meters. Figure 2.1 illustrates their approximate locations. The two locations are marked with red stars.

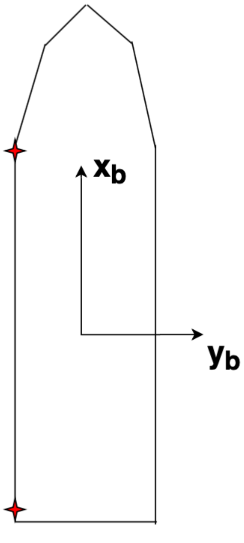


Figure 2.1: Location of the proximity sensors on CSAD for the simulations.

The idea of the location of the proximity sensor is that they should be placed where the ship side stops being parallel with the x_b axis, and one in each end. Thus they will be able to give information of the relative angle between the ship and the harbor side, in addition to the distance between the sensors/ship and the harbor side. The sensors will work perpendicular to the side of the ship. This is illustrated in figure 2.2.

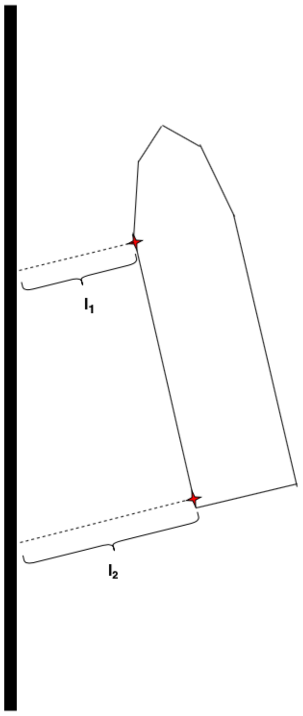


Figure 2.2: Ship with proximity sensors measuring the distance to the harbor.

The measurements of the sensors will be denoted and defined as

$$l_{m,1}(t) = l_1(t) + w(t) \tag{2.3}$$

$$l_{m,2}(t) = l_2(t) + w(t) \tag{2.4}$$

where $l_1(t)$ and $l_2(t)$ are the true distances between the harbor side and sensors and w is zero mean Gaussian white noise. We also have that $l_1(t) > 0$ and $l_2(t) > 0$. The variance of the noise is set to be 0.04 cm^2 according to Cuspsensors (2019).

To model the distances l_1 and l_2 mathematically, firstly the location of the proximity sensors in the basin-fixed reference frame must be calculated. The location of the proximity sensors and the harbor must be in the same reference system to calculate the distance between them, and the harbor is defined in the basin-fixed reference frame. The location of the sensors in the basin-fixed reference frame is

$$(x^n_{s,1}, y^n_{s,1}) = (N + cos(\psi)x^b_{s,1} - sin(\psi)y^b_{s,1}, E + cos(\psi)x^b_{s,1} + sin(\psi)y^b_{s,1}) \tag{2.5}$$

$$(x^n_{s,2}, y^n_{s,2}) = (N + cos(\psi)x^b_{s,2} - sin(\psi)y^b_{s,2}, E + cos(\psi)x^b_{s,2} + sin(\psi)y^b_{s,2}) \tag{2.6}$$

Here N is the x-position of the vessel in the basin-fixed reference frame, E is the y-position in the basin-fixed reference frame and ψ is the heading. When the location of the proximity sensors in $\{n\}$ is known, the distances l_1 and l_2 can be found. Firstly, an equation for each of the dashed lines in figure 2.2 must be found. The lines are perpendicular to the ship side and passes through the point where the proximity sensors are located. These lines can be used to find the intersections of the lines and the line representing the harbor side. When the two intersections have been calculated, in the form of coordinates in $\{n\}$, the Pythagoras sentence can be used to calculate the lengths l_1 and l_2. A straight line can be described with the equation

$$y = ax + b \tag{2.7}$$

where a is the slope of the line and b is where the line intersects the y-axis. To calculate where two straight lines intersects, the method described below has been used. Two lines, denoted y_1 and y_2, of the same form as equation 2.7 is the basis of the method. The method is simply as follows:

1. Set $y_1 = y_2$, and solve for x.
2. When the x coordinate is calculated, insert the numerical value into either the equation for y_1 or y_2 and solve for y.

If done correctly and intersection point is calculated. Of course if the two lines have the same slope they will not intersect, and the method will not work. In addition it must be taken into account that the lines can be either vertical or horizontal, meaning the lines can not be of the exact form from equation 2.7. The code in Matlab for the proximity sensor calculations, can be found in appendix A.1.

2.2 Kinematics

Three different reference frames are used for the autonomous docking control system. The coordinate frames typically used to describe the position and orientation of a ship are the North-East-Down (NED) coordinate frame and the Body(-fixed) coordinate frame(Perez and Fossen, 2007).

- The NED coordinate frame $\{n\}$ is the local geographical coordinate system fixed to the earth surface. The x-axis points towards the true north, the y-axis points east and the z-axis points down. The plane made up of the x-axis and the y-axis are tangential to the earth surface. The origin of the the coordinate frame is placed at a point of interest. The NED coordinate system is used to describe the vessels position in addition to the direction of the environmental forces wind, current and waves. The frame is also assumed inertial for ships operating at a constant latitude and longitude, so called flat earth navigation. In this thesis, $\{n\}$ is represented by the basin-fixed reference frame.
- The body-fixed coordinate frame $\{b\}$ is fixed to the vessel. The x-axis is pointing towards the bow, the y-axis is pointing starboard and the z-axis is pointing downwards. The coordinates of the origin of the frame is usually made up of half of the ship length($lpp/2$), the design waterline and the longitudinal centre line representing the port/starboard symmetry. The body fixed reference frame is mainly used for expressing measurements of the acceleration and velocity of the vessel.
- The third reference frame, is the harbor-fixed reference frame $\{h\}$. The x-axis of the reference frame is set to be parallel to the harbor. The y-axis is perpendicular to the harbor and the z-axis points downwards, making it parallel with the z-axis in $\{n\}$. The origin of $\{h\}$ is calculated based on the relative orientation of the ship compared to the harbor, and the measurements from the two proximity sensors. The origin of the harbor fixed reference frame, is placed with a negative translation equal to the mean value of l_1 and l_2 along the y-axis of $\{b\}$ and from the center of $\{b\}$. This is of course assuming the position of the ship is similar to the one in figure 2.2.

In accordance with $\{n\}$ and $\{b\}$ coordinate frames the orientation, position and velocities of a ship in 6 DOF can be expressed as

$$\boldsymbol{\eta} = [\mathbf{p}^n_{b/n}\ \boldsymbol{\Theta}_{nb}]^T = [x, y, z, \phi, \theta, \psi]^T \tag{2.8}$$

$$\boldsymbol{\nu} = [\boldsymbol{v}^b_{b/n}\ \boldsymbol{\omega}^b_{b/n}]^T = [u, v, w, p, q, r]^T \tag{2.9}$$

where $\mathbf{p}^n_{b/n}$ is the vessels position in $\{n\}$, $\boldsymbol{\Theta}_{nb}$ is the attitude of the vessel in Euler angles and $\boldsymbol{v}^b_{b/n}$ and $\boldsymbol{\omega}^b_{b/n}$ is respectively the linear and angular velocities of the vessel expressed in {b}.

For vessels operating at the surface the number of DOFs is often reduced from six to three. The reduction assumes that the pitch and roll angles are small, which is a good approximation for most conventional ships(Fossen, 2011). In this case equations (2.8) and (2.9) are reduced to

$$\boldsymbol{\eta} = [x, y, \psi]^T \tag{2.10}$$

$$\boldsymbol{\nu} = [u, v, r]^T \tag{2.11}$$

The rotation matrix $\boldsymbol{R}(\psi)$ used in this instance is defined as

$$\boldsymbol{R}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

and is used in the transformation

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.13}$$

For rotation transformation when considering six DOFs, two additional rotation matrices must be included. The rotation matrix from (2.12) uses the yaw angle ψ, but for six DOFs the pitch and roll angles also have to be used. When assuming small pitch and roll angles from the three DOF simplification, these two matrices will be approximately equal to the identity matrix.

2.3 Kinetics - 3 DOF Maneuvering model

A 3 DOF maneuvering model of the vessel can be found in NTNU (2017), and will be presented in this section. This model is valid for low speed, and does not include cross-coupled damping terms. The parameter identification is based on towing tests in the MC-Lab.

The model is given by

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.14}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\nu)\nu + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{env} + \boldsymbol{\tau}_{thrusters} \tag{2.15}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are defined as in 2.10 and 2.11 respectively, and

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{env} + \boldsymbol{\tau}_{thrusters} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} \tag{2.16}$$

where X and Y are the forces acting on the vessel in body-fixed reference frame in the surge and sway direction respectively. Furhtermore, N is the moment acting on the vessel in body-fixed reference frame in yaw. The mass matrix $\boldsymbol{M}$ is given by

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.17}$$

The Coriolis matrix $\boldsymbol{C}$ is given by

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v - (mx_g - Y_{\dot{r}})r \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v + (mx_g - Y_{\dot{r}})r & -(m - X_{\dot{u}})u & 0 \end{bmatrix} \tag{2.18}$$

where

$$\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}) \tag{2.19}$$

where $\boldsymbol{C}_{RB}$ is the Coriolis force matrix and $\boldsymbol{C}_A$ is the centripetal force matrix.

The damping matrix $\boldsymbol{D}$ is given by

$$\boldsymbol{D} = -\begin{bmatrix} X_u{+}X_{|u|u}|u|{+}X_{uuu}u^2 & 0 & 0 \\ 0 & Y_v{+}Y_{|v|v}|v|{+}Y_{vvv}v^2 & Y_r{+}Y_{|r|r}|r|{+}Y_{rrr}r^2 \\ 0 & N_v{+}N_{|v|v}v{+}N_{vvv}v^2 & {+}N_r{+}N_{|r|r}|r|{+}N_{rrr}r^2 \end{bmatrix} \tag{2.20}$$

where $X_u, X_{|u|u}, X_{uuu}, Y_v, Y_{|v|v}, Y_{vvv}, Y_r, Y_{|r|r}, Y_{rrr}, N_v, N_{|v|v}, N_{vvv}, N_r, N_{|r|r}, N_{rrr}$ are drag coefficients, and

$$\boldsymbol{D}(\boldsymbol{\nu}) = \boldsymbol{D} + \boldsymbol{D}(\boldsymbol{\nu}) \tag{2.21}$$

The numerical values of the parameters used in the matrices above, can be found in table 2.1, 2.2, 2.3.

Table 2.1: Rigid body parameters for CSAD

Parameter	Value
m	127.92
I_z	61.967
x_g	0

Table 2.2: Added mass parameters for CSAD

Parameter	Value
$X_{\dot{u}}$	3.262
$Y_{\dot{v}}$	28.89
$Y_{\dot{r}}$	0.525
$N_{\dot{v}}$	0.157
$N_{\dot{r}}$	13.98

Table 2.3: Drag coefficients in surge, sway and yaw for CSAD

Parameter	Numerical value
X_u	-2.332
$X_{\lvert u\rvert u}$	0
X_{uuu}	-8.557
Y_v	-4.673
$Y_{\lvert v\rvert v}$	0.3976
Y_{vvv}	313.3
N_r	-0.01675
$N_{\lvert r\rvert r}$	-0.01148
N_{rrr}	0.0003578

2.4 Environmental forces

The environmental forces considered in the simulations of the autodocking control system, is waves and current. Wind has not been included.

2.4.1 Modeling of Current

The current in $\{n\}$ has been modeled as irrotational and constant, where $\boldsymbol{\nu}_c^n= [u_c, v_c, 0]^T$. The current speed in in the x-direction is denoted u_c and the current speed in the y-direction is denoted v_c. The current vector is expressed in $\{b\}$ by using the 3DOF Euler angle rotation matrix:

$$\boldsymbol{\nu}_c^b = \boldsymbol{R}^T(\psi)\boldsymbol{\nu}_c^n \tag{2.22}$$

meaning $\boldsymbol{\nu}_c^b$ is the current velocities expressed in the body-fixed reference frame.

2.4.2 Modeling of waves

It is common to distinguish between two type of effects from the wave-induced forces:

- **First-order wave-induced forces** is zero mean oscillatory Wave-frequency (WF) motion.
- **Second-order wave-induced forces** is the nonzero and slowly varying component of the wave forces.

The wave forces has been modeled using a linear approximation for first-order wave-induced forces $\boldsymbol{\tau}_{wave1}$, such that

$$\boldsymbol{\tau}_{wave1} \approx \boldsymbol{K}\boldsymbol{H}_s(s)\boldsymbol{w}(s) \tag{2.23}$$

where $\boldsymbol{K}$ is a constant tunable gain which approximates the amplitude-to-force transfer function. The numerical values of the gain matrix is selected such that the impact the wave forces has on $\boldsymbol{\eta}$ is of a appropriate size. The matrix $\boldsymbol{H}_s = diag\{h^{\{1\}}, h^{\{2\}}, h^{\{3\}}\}$ is an approximation of the linear wave response. There will be one transfer function for each degree of freedom given by

$$h(s) = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^s} \tag{2.24}$$

where the gain constant K_w is defined as

$$K_w = 2\lambda\omega_0 s\sigma \tag{2.25}$$

where λ is a damping coefficient and ω_0 is the dominating wave frequency. The wave intensity is denoted σ. What values to use for these parameters depends on which wave spectrum is being used, which in this case is the Pierson-Moskowitz (PM) spectrum. For the PM spectrum, according to Fossen (2011), we have the following λ and ω_0 values:

Table 2.4: λ values for the PM spectrum.

ω_0	λ
0.5	0.2565
0.8	0.2573
1.1	0.2588
1.4	0.2606

The recommended value is $\lambda = 0.26$. For the PM spectrum the wave intensity is given by

$$\sigma = \sqrt{\frac{A}{\omega_0^5} exp(-\frac{B}{\omega_0^4})} \tag{2.26}$$

where it is retrieved from Myrhaug and Lian (2009) that

$$A = 0.0081 g^2 \tag{2.27}$$

$$B = 0.74(\frac{g}{V})^4 \tag{2.28}$$

where B is the wind speed at 19.5 meters altitude and g is the acceleration of gravity. For equation 2.23 we also have that $w^{\{dof\}}(s)$ is a zero-mean Gaussian white noise process with unity power across the spectrum, i.e. $P_{ww}^{\{dof\}}(\omega) = 1.0$.

2.5 Froude Scaling

Froude scaling has been utilized to check if the magnitude of the parameters and variables in the simulations and experiments, are of a reasonable magnitude in full scale. To go from a distance in meters in the model scale to the same distance in meters in full scale, one must multiply with 90. Scaling the time to fullscale is achieved by multiplying with $\sqrt{90}$.

Chapter 3

Design of Controller and Observer

3.1 Controller design

The task of designing an autodocking system has been separated in to two phases:

- **Phase 1:** Is the phase where the ship is to follow the generated path by the path planning algorithm. The phase can also be referred to as the path following phase.
- **Phase 2:** This phase is where the ship is to maneuver close and parallel to the harbor side. This phase can also be referred to as the docking phase.

Two different control systems for autonomous docking is proposed in this thesis. They will be referred to as method 1 and method 2, and an overall description of the two systems and their differences follows:

- **Method 1:** This method utilizes two different controllers. The first controller is responsible for the path following, and the second controller takes over when the docking phase is initiated. The controller responsible for the path following is described in section 3.1.1, while the controller responsible for performing the docking operation is described in 3.1.2.
- **Method 2:** This method utilizes only one controller, namely the controller from section 3.1.1. When the docking phase is initiated, the guidance system uses the proximity sensor to generate an appropriate path towards the side of the harbor. The guidance system is described in chapter 4.

3.1.1 Controller for Phase 1

As mentioned earlier, the phase 1 controller will be responsible for making the ship follow the path to the docking location. The output from the guidance system for phase 1, described in chapter 4, is the desired position and heading ($\boldsymbol{\eta}_d$) and the desired velocity ($\boldsymbol{\nu}_d$) given by

$$\boldsymbol{\eta}_d = \begin{bmatrix} x_d \\ y_d \\ \psi_d \end{bmatrix} \tag{3.1}$$

$$\boldsymbol{\nu}_d = \begin{bmatrix} u_d \\ v_d \\ r_d \end{bmatrix} \tag{3.2}$$

The controller is created as a Multiple-Input and Multiple-Output (MIMO) Proportional-Integral-Derivative (PID) controller. The control law is based on Fossen (2011), and is given by

$$\boldsymbol{\tau} = -\boldsymbol{K}_p \boldsymbol{R}^T(\boldsymbol{\eta})\tilde{\boldsymbol{\eta}} - \boldsymbol{K}_d \tilde{\boldsymbol{\nu}} - \boldsymbol{K}_i \int_0^t \boldsymbol{R}^T(\boldsymbol{\eta})\tilde{\boldsymbol{\eta}}(\tau)d\tau + \boldsymbol{D}\boldsymbol{\nu}_d \tag{3.3}$$

where $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_d$, $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \boldsymbol{\nu}_d$, $\boldsymbol{\tau} = [\tau_X \; \tau_Y \; \tau_N]^T$. Acceleration feedforward was not included as the desired acceleration $\dot{\boldsymbol{\nu}}_d$ would have been very low and is thus assumed zero. In addition we have that the proportional gain matrix $\boldsymbol{K}_p$ is positive definite, diagonal and constant. The same goes for the integral and derivative gain matrices $\boldsymbol{K}_i$ and $\boldsymbol{K}_d$.

Calculating the controller gains

According to Fossen (2011), if low speed is assumed, the model in equation 2.15 can be simplified to

$$\dot{\boldsymbol{\eta}}_p = \boldsymbol{\nu} \tag{3.4}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{\tau} \tag{3.5}$$

here $\boldsymbol{\eta}_p$ is vessel parallel coordinates, defined as

$$\boldsymbol{\eta}_p = \boldsymbol{R}^T(\psi)\boldsymbol{\eta} \tag{3.6}$$

The terms $\boldsymbol{D}(\boldsymbol{\nu})$ and $\boldsymbol{C}(\boldsymbol{\nu})$ from 2.15 can be neglected if integral action is included to compensate for the drift and ocean currents. Equation 3.5 written out, yields

$$\begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \dot{\boldsymbol{\nu}} - \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \boldsymbol{\nu} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} \tag{3.7}$$

Thus, the subsystems in surge, sway and yaw individually looks like

$$(m - X_{\dot{u}})\dot{u} - X_u u = \tau_X \tag{3.8}$$

$$(m - Y_{\dot{v}})\dot{v} - Y_{\dot{r}}\dot{r} - Y_v v = \tau_Y \tag{3.9}$$

$$(I_z - N_{\dot{r}})\dot{r} - Y_{\dot{r}}\dot{v} - N_r r = \tau_N \tag{3.10}$$

The simplification can be taken one step further if the three subsystems are decoupled by assuming $\dot{r} \approx 0$ for equation 3.9 and $\dot{v} \approx 0$ for equation 3.10. Thus, we end up with the three decoupled subsystems

$$(m - X_{\dot{u}})\dot{u} - X_u u = \tau_X \tag{3.11}$$

$$(m - Y_{\dot{v}})\dot{v} - Y_v v = \tau_Y \tag{3.12}$$

$$(I_z - N_{\dot{r}})\dot{r} - N_r r = \tau_N \tag{3.13}$$

By pole placement, the gain matrices are given by

$$\boldsymbol{K}_p = \begin{bmatrix} (m - X_{\dot{u}})\omega_n^2 & 0 & 0 \\ 0 & (m - Y_{\dot{v}})\omega_n^2 & 0 \\ 0 & 0 & (I_z - N_{\dot{r}})\omega_n^2 \end{bmatrix} \tag{3.14}$$

$$\boldsymbol{K}_d = \begin{bmatrix} (m - X_{\dot{u}})(2\zeta\omega_n - \frac{-X_u}{m - X_{\dot{u}}}) & 0 & 0 \\ 0 & (m - Y_{\dot{v}})(2\zeta\omega_n - \frac{-Y_v}{m - Y_{\dot{v}}}) & 0 \\ 0 & 0 & (I_z - N_{\dot{r}})(2\zeta\omega_n - \frac{-N_r}{I_z - N_{\dot{r}}}) \end{bmatrix} \tag{3.15}$$

$$\boldsymbol{K}_i = \frac{\omega_n}{10}\boldsymbol{K}_p \tag{3.16}$$

where ζ is usually in the range of 0.8-1.0, and ω_n is determined by

$$\omega_n = \frac{1}{\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}}\omega_b \tag{3.17}$$

where ω_b is the control bandwidth. Regarding its numerical value, it is usually set to be 0.01 rad/s for large tankers and 0.1 rad/s for smaller vessels and underwater vehicles. The expression for the integral gain is based on a rule of thumb, stating the integrator should be ten times slower than the natural frequency ω_n. The equations 3.14 - 3.17 are used as a basis for the controller gains.

3.1.2 Controller for Phase 2

For phase 2, three controllers have been designed. There is one controller for each degree of freedom(surge, sway and yaw). This controller will make sure the ship is safely maneuvered into its desired end state, which is parallel to the harbor side and sufficiently close to the harbor so that the human crew can moor the ship. The desired velocity from the guidance system for phase 2, is

$$\boldsymbol{\nu}_d = \begin{bmatrix} u_d \\ v_d \end{bmatrix} \tag{3.18}$$

A desired speed for the yaw controller is not needed, so only desired surge and sway velocity is used. In addition, the desired distance between the ship and the harbor, denoted l_d, is also an output from the guidance system. The measurements from the two proximity sensors will be used to measure the distance between the ship and the harbor side, and the relative angle of the ship compared to the harbor side. The measurements from the two proximity sensors are, as mentioned in section 2.1, denoted $\boldsymbol{l}_m = [l_{m,1},\ l_{m,2}]$.

Surge speed controller

The surge controller, is responsible for minimizing the ships movement in the surge direction. A surge speed controller is selected for this task, and the desired surge speed will be $u_d = 0$. The method selected to design the surge speed controller is state feedback linearization as described in chapter 13.2 in Fossen (2011). The basis of this controller design is the surge subsystem model, which can be written as

$$(m - X_{\dot{u}})\dot{u} - X_u u - X_{|u|u}|u|u = \tau_X \tag{3.19}$$

This model assumes a constant or slowly varying forward speed, and that the surge speed is much greater than the sway speed. This means

$$U = \sqrt{u^2 + v^2} \approx u \tag{3.20}$$

But this will not be the case for phase 2 where the ship will spend much time moving primarily sideways, meaning $U \approx v$. Still, the model in equation 3.19 has been used as a basis for the control design.

State feedback linearization is used to design the controller. The method is designed to transform the nonlinear system dynamics into a linear system. Thus, the control law is selected to be

$$\tau_X = ma^b - X_{|u|u}|u|u - X_u u \tag{3.21}$$

Equation 3.21 inserted into 3.19, will transform the nonlinear system dynamics into a linear system. For the control law, a^b is the commanded acceleration in surge. The controller is chosen to be a PI controller given by

$$a^b = \dot{u}_d - K_p\tilde{u} - K_i \int_0^t \tilde{u} d\tau \tag{3.22}$$

where $\tilde{u} = u - u_d$ and $\dot{u}_d$ is the acceleration feedforward. The controller gains is selected according to

$$K_p = 2\lambda \tag{3.23}$$

$$K_i = \lambda^2 \tag{3.24}$$

These controller gains leads to the second order error dynamics

$$m(\dot{\tilde{u}} - a^b) = m(\dot{\tilde{u}} + 2\lambda\tilde{u} + \lambda^2 \int_0^t \tilde{u} d\tau) = 0 \tag{3.25}$$

meaning the pole will be placed in $s = -\lambda$. Because the desired surge speed is zero, the acceleration feedforward component will also be zero.

Surge controller accounting for current

The surge speed controller described above, does not perform well when there is current present as the ship will drift. Thus, the need of a controller which can perform well when there is current present, presented itself. This controller will use the harbor fixed reference system $\{h\}$. The PI controller designed is of the following form:

$$\tau_X = -K_p\tilde{x} - K_i \int_0^t \tilde{x} d\tau \tag{3.26}$$

where $\tilde{x} = x^h - x_d^h$ is the error in the harbor fixed reference frame between the x-position of the ship and the desired x-position. The controller gains K_p and K_i are selected to be the same as for the phase 1 controller in the sway direction, meaning

$$K_p = (m - X_{\dot{u}})\omega_n^2 \tag{3.27}$$

$$K_i = \frac{\omega_n}{10}K_p \tag{3.28}$$

Sway Controller

The sway controller uses the measurements from the proximity sensors, to position the ship at the desired distance from the quayside. The controller is selected to be a PID controller:

$$\tau_Y = -K_p\tilde{l} - K_d\dot{\tilde{l}} - K_i\int_0^t \tilde{l}d\tau \tag{3.29}$$

where $\tilde{l} = min(l_{m,1}, l_{m,1}) - l_d$. The proximity sensor which is closest to the harbor side, i.e. has the lowest numerical value measurement, is selected because it is the most critical. It is the most critical in the sense that it will tell which end of the ship which is closest to the harbor side and collision between the ship and harbor have to be prevented. An alternative is to use the mean value of the two sensor measurements, but in some cases this will not be suitable as one side of the ship can be very close to the harbor while the other side is not.

The controller gains K_p, K_d and K_i are selected to be the same as for the phase 1 controller in the sway direction, meaning

$$K_p = (m - Y_{\dot{v}})\omega_n^2 \tag{3.30}$$

$$K_d = (m - Y_{\dot{v}})(2\zeta\omega_n - \frac{-Y_v}{m - Y_{\dot{v}}}) \tag{3.31}$$

$$K_i = \frac{\omega_n}{10}K_p \tag{3.32}$$

These gains are only used as a basis, so trial and error has lead to adjustment of these gains to get the actually used gains.

A reference model, described in 4.5, will create suitable velocity and position trajectories for the controller to follow.

Heading Controller

The ship will be parallel with the harbor side if $l_1 = l_2$. Thus, this is the control objective of the heading controller. The controller is selected to be a PI controller:

$$\tau_N = -K_p\tilde{l_m} - K_i\int_0^t \tilde{l_m}d\tau \tag{3.33}$$

where $\tilde{l_m} = l_{m,1} - l_{m,2}$. Thus if the difference is negative, meaning the aft of the ship is closer to the harbor than the fore of the ship, a clockwise moment(in body frame) will be induced by the controller, and vice versa.

The gains of the controller are selected to be the same as for the phase 1 controller for yaw, which leads to

$$K_p = (I_z - N_{\dot{r}})\omega_n^2 \tag{3.34}$$

$$K_i = \frac{\omega_n}{10} K_p \tag{3.35}$$

3.2 Observer Design

The task of an observer is to estimate the states of a real system, both measured and unmeasured. Instrument measurements can be noisy, disturbed or/and incomplete, which makes it necessary to include an observer. An observer is also used for wave filtering to avoid feeding first-order wave induced forces into the controller, resulting in unnecessary wear and tear of the actuators. This is achieved by using the estimated pose and heading of the vessel, instead of the measured pose and heading as input for the controller. For a large vessel the actuators does not have the capacity to counteract the high frequency disturbances, thus it is necessary to use the estimated states.

3.2.1 Nonlinear Passive Observer

The observer of choice in this thesis, is the nonlinear passive observer. Compared to a Kalman filter, it is easier to tune. When tuning a Kalman filter it can be hard to relate the tuning parameters to real physical quantities, and the process can be time consuming.

The design of the observer is based on Fossen (2011). For the observer, the following assumptions is necessary to prove passivity:

- **Assumption 1:** $\boldsymbol{w} = \boldsymbol{0}$ and $\boldsymbol{v} = \boldsymbol{0}$. In the analysis of the observer the zero mean white noise terms are excluded. Including these terms will result in the error dynamics becoming Uniformly ultimated bounded (UUB) for the Lyapunov function analysis instead of Uniform Global Asymptotically Stable (UGAS) or Uniform Global Exponentially Stable (UGES).
- **Assumption 2:** $\mathbf{R}(y_3) = \mathbf{R}(\psi)$, i.e. $y_3 = \psi + \psi_w \approx \psi$. In extreme weather conditions the contribution of the wave induced yaw disturbance ψ_w is usually less than five degrees, making this a good approximation. For calmer conditions the disturbance is normally less than one degree.

The equations for the nonlinear passive observer are as follows:

$$\dot{\hat{\boldsymbol{\xi}}} = \boldsymbol{A}_w \hat{\boldsymbol{\xi}} + \boldsymbol{K}_1(\boldsymbol{\omega}_0)\tilde{\boldsymbol{y}} \tag{3.36}$$

$$\dot{\hat{\boldsymbol{\eta}}} = \boldsymbol{R}(y_3)\hat{\boldsymbol{\nu}} + \boldsymbol{K}_2\tilde{\boldsymbol{y}} \tag{3.37}$$

$$\dot{\hat{\boldsymbol{b}}} = -\boldsymbol{T}^{-1}\hat{\boldsymbol{b}} + \boldsymbol{K}_3\tilde{\boldsymbol{y}} \tag{3.38}$$

$$\boldsymbol{M}\dot{\hat{\boldsymbol{\nu}}} = -\boldsymbol{D}\hat{\boldsymbol{\nu}} + \boldsymbol{R}^T(y_3)\hat{\boldsymbol{b}} + \boldsymbol{\tau} + \boldsymbol{R}^T(y_3)\boldsymbol{K}_4\tilde{\boldsymbol{y}} \tag{3.39}$$

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\eta}} + \boldsymbol{C}_w\hat{\boldsymbol{\xi}} \tag{3.40}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ is the estimation error and $\boldsymbol{K}_1(\boldsymbol{\omega}_0)$ together with $\boldsymbol{K}_{2,3,4}$ are the observer gain matrices. The dimensions of the matrices are respectively 6x3 and 3x3. In addition, $\boldsymbol{A}_w$ and $\boldsymbol{C}_w$ are constant matrices describing the sea state and $\boldsymbol{\xi}$ is the wave frequency motion vector. Low-pass action is introduced with $\boldsymbol{T} > 0$, instead of only having pure integral action of the white noise. The matrix $\boldsymbol{A}_w$ is assumed Huruwitz and describes the first-order WF-induced motion in a mass-spring-damper system, which yields

$$\boldsymbol{A}_w = \begin{bmatrix} \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \mathbf{\Omega}^2 & -2\mathbf{\Gamma}\mathbf{\Omega} \end{bmatrix} \tag{3.41}$$

where $\mathbf{\Gamma} = diag(\lambda_1, \lambda_2, \lambda_3)$ and $\mathbf{\Omega} = diag(\omega_1, \omega_2, \omega_3)$. The realtive damping ratios of the wave spectrum λ_i are usually set between 0.05 and 0.2. The frequencies ω_i are the dominating wave response frequencies.

The gain matrices are calculated using

$$K_{1i}(\omega_{oi}) = -2(\zeta_{ni} - \lambda_i)\frac{\omega_{ci}}{\omega_{oi}} \tag{3.42}$$

$$K_{1(1+3)}(\omega_{oi}) = 2\omega_{oi}(\zeta_{ni} - \lambda_i) \tag{3.43}$$

$$K_{2i} = \omega_{ci} \tag{3.44}$$

knowing that

$$\boldsymbol{K}_i(\boldsymbol{\omega}_0) = \begin{bmatrix} diag\{K_{11}(\omega_{o1}), K_{12}(\omega_{o2}), K_{13}(\omega_{o3})\} \\ diag\{K_{14}(\omega_{o1}), K_{15}(\omega_{o2}), K_{16}(\omega_{o3})\} \end{bmatrix} \tag{3.45}$$

$$\boldsymbol{K}_2 = diag\{K_{21}, K_{22}, K_{23}\} \tag{3.46}$$

$$\boldsymbol{K}_3 = diag\{K_{31}, K_{32}, K_{33}\} \tag{3.47}$$

$$\boldsymbol{K}_4 = diag\{K_{41}, K_{42}, K_{43}\} \tag{3.48}$$

where ω_{oi} is the wave spectrum peak frequency and ω_{ci} is the filter cutoff frequency. Typical value for the notch filter parameter ζ_{ni} is 1.0. For the remaining gain matrices $\boldsymbol{K}_3$ and $\boldsymbol{K}_4$, the following tuning rule is used:

$$1/T_i << K_{3i}/K_{4i} < \omega_{oi} < \omega_{ci} \tag{3.49}$$

3.3 Thrust Allocation

The thrust allocation used in the simulations are from Frederich (2016). It is treated as a black box, but it is an constrained optimal thrust allocation which makes use of a quadratic regulator algorithm to optimize the allocation.

Chapter 4

Guidance and Path Planning

The guidance system is responsible for generating appropriate setpoints for the position and heading of the vessel. The desired heading is denoted ψ_d and the desired position is

$$\boldsymbol{p}_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix} \tag{4.1}$$

This chapter concerns how the guidance subsystem generate the desired position and heading signal. The desired position and heading will be input to the controller, which has the task of getting the vessel to the desired position and heading, i.e. $\boldsymbol{\eta} \to [\boldsymbol{p}_d^T,\ \psi_d]^T$. The guidance block utilizes the measurements from the proximity sensors and a parameterized path to generate the desired position and heading signals.

As mentioned earlier in chapter 3, two methods is proposed. The guidance system utilized in method two, is described in sections 4.2, 4.3 and 4.4. Method 1 also uses the path following method from section 4.2 for phase 1, but when the docking controller takes over for the path following controller it does not use the docking path generation described in section 4.3. Instead it uses the guidance system explained in section 4.5.

The methods presented in section 4.2, 4.3.2 and 4.4 is proposed by supervisor Roger Skjetne and based on Skjetne (2005).

4.1 Parameterized Path

A set of waypoints is needed to create a parameterized path. Typically the first waypoint will be the current position of the ship, and the last waypoint will be a user specified position close to the quay. The waypoints can be organized, such that

$$\boldsymbol{WP} = [\boldsymbol{p}_0,\ \boldsymbol{p}_1,\ \dots\ \boldsymbol{p}_n] \tag{4.2}$$

where n is the number of subpaths running between the waypoints and $\boldsymbol{p}_i = [x_i\ y_i]^T$ is the coordinates of waypoint number $i = 0, 1, 2 \dots, n$. The method used to generate the path is hybrid parametrization and it is from Skjetne (2019). A set of waypoints is specified and a curve passing through these waypoints is constructed. Each of the subpaths is expressed as a polynomial in s of a specified order. At the waypoints these polynomials are concatenated to construct a continuous and sufficiently differentiable path. The order of the polynomial is what determines the curves differentiability. The complete desired path is as follows:

$$\boldsymbol{p}_d(s_1) = [x_d(s_1), y_d(s_1)]^T \tag{4.3}$$

The path parameter s_1 will have a numerical value in the range $[0, n]$. This number will be one less than the number of waypoints. When the value of the path parameter is zero the desired position will be the first waypoint, and when the path parameter is equal to one the desired position is the second waypoint and so forth. Now examine the polynomials of order k, given as

$$x_{d,i}(s_1) = a_{k,i}s_1^k + \ldots + a_{1,i}s + a_{0,i} \tag{4.4}$$

$$y_{d,i}(s_1) = b_{k,i}s_1^k + \ldots + b_{1,i}s + b_{0,i} \tag{4.5}$$

where $a_{j,i}$ and $a_{j,i}$ are coefficients to be calculated. The number of methods which can be used to determine these coefficients are many. The most apparent method is to make a large set of $2n(k+1)$ linear equations,

$$\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{b} \tag{4.6}$$

where $\boldsymbol{\phi}^T = [\boldsymbol{a}^T, \boldsymbol{b}^T]$ and solve for $\boldsymbol{\phi} = \boldsymbol{A}^{-1}\boldsymbol{b}$. This method does encounter some numerical difficulties as the number of subpaths increase. An alternative to this method is to calculate the coefficients for each subpath independently. The continuity at the connection points between the subpaths, is ensured by assigning numerical values common for the adjacent subpaths. A design constant $\lambda > 0$ specifies the slope at the waypoints.

Along the path, we have that the unit tangent vector is defined as

$$\boldsymbol{T}_d(s_1) = \frac{\boldsymbol{p}_d^s(s_1)}{|\boldsymbol{p}_d^s(s_1)|} \tag{4.7}$$

where $\boldsymbol{p}_d^s(s_1) = [x_d^s(s_1), y_d^s(s_1)]$ is the first derivative at (x_d, y_d) corresponding to s_1. With the tangent vector known, the normal unit vector becomes

$$\boldsymbol{N}_d(s_1) = \boldsymbol{J}\boldsymbol{T}_d(s_1) \tag{4.8}$$

where

$$\boldsymbol{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{4.9}$$

Thus we have that

$$\boldsymbol{T}_d(s_1)^T\boldsymbol{N}_d(s_1) = 0 \tag{4.10}$$

4.1.1 Selection of waypoints

The first waypoint for the path $\boldsymbol{p_0}$, will be the current position of the ship. The last waypoint of the path, $\boldsymbol{p_n}$ will be operator specified and placed just outside the harbor. The number of waypoints in between the first and the last, is selected to be one. This means the path is generated from only three waypoints. The middle waypoint is found by the following equations

$$x_1 = x_0 + c_x(x_2 - x_0) \tag{4.11}$$
$$y_1 = y_0 + c_y(y_2 - y_0) \tag{4.12}$$

where $c_x \in [0, 1]$ and $c_x \in [0, 1]$ are constants. They specify where in the middle of the first and last waypoint the second waypoint is placed. If $c_x = 0.5$ this means the x-coordinate of the second waypoint is exactly in the middle of the x-coordinates of the first and the last waypoint. A higher value than this, will place the x-coordinate of the waypoint closer to the x-coordinate of the last waypoint, and a lower value will place it closer to the first waypoint. The same goes for c_y. These constants are selected knowing that we want the path to run approximately parallel to the dock towards the last waypoint.

Figure 4.1 below shows an example of how the waypoints can be selected, and the resulting path. The red lines in the figure represents the side of the harbor. Here, the current position of the ship and the first waypoint is $\boldsymbol{p}_0 = [6, 1]$ and the last waypoint is selected to be $\boldsymbol{p}_2 = [12.5, -1.7]$. The middle waypoint $\boldsymbol{p}_1$ is placed by selecting the constants to be $c_x = 0.3$ and $c_y = 0.8$.

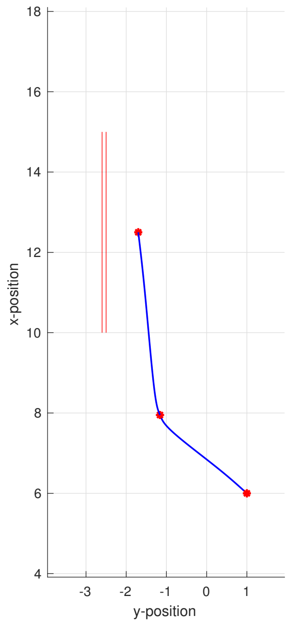


Figure 4.1: Waypoint selection

From the figure it can be seen that the end of the path runs approximately perpendicular to the side of the quay.

4.2 Phase 1 - Path following

The path is assumed given by the signal $\boldsymbol{p}_d(s_1)$, where s_1 is the path parameter which continuously parameterizes the path. How fast the path parameter increases, is adjusted with the speed assignment

$$\dot{s_1} = v_s(t, s_1) \tag{4.13}$$

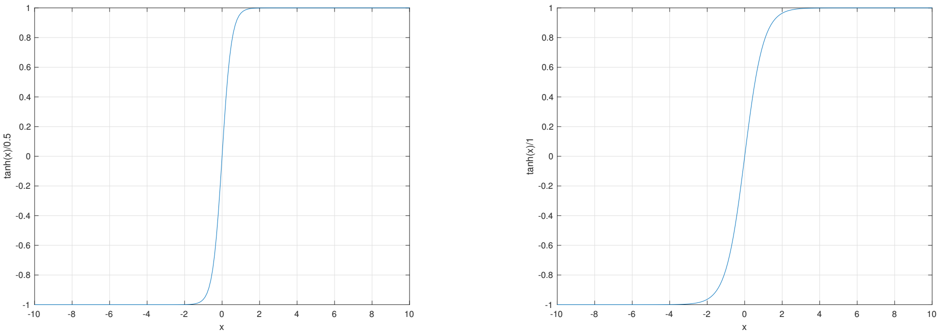
The speed assignment v_s is chosen such that the ship comes to rest, i.e. zero speed, when s_1 reaches the final waypoint. For the speed assignment, we have that when $s_1 = n$ then $\boldsymbol{p}_d(s_1) = \boldsymbol{p}_n$. The design of the speed assignment should be such that when $s_1 < n$ the speed of the ship is positive and negative for $s_1 > n$. When $s_1 = n$ the speed should be zero. This leads to the following proposition:

$$\dot{s}_1 = v_s(t, s_1) := \frac{u_d(t)}{|\boldsymbol{p}_d(s_1)|}\omega(s_1) \tag{4.14}$$

where $u_d(t)$ is set by the operator and is the desired speed along the path with unit $[m/s]$. The function ω, is selected as

$$\omega(s_1) := \tanh(\frac{n - s_1}{\Delta_u}) \tag{4.15}$$

where Δ_u is a gain determining the slope of $\omega(s_1)$ close to $n - s_1 = 0$. This function ensures that the speed will reduce to zero as the ship approaches the last waypoint. Figure 4.2 shows how the function $\omega(s_1)$ varies with the selection of Δ_u. The higher the Δ_u the steeper the slope, and vice versa.



(a) $\tanh(\frac{x}{1})$

(b) $\tanh(\frac{x}{0.5})$

Figure 4.2: Two $\tanh(\frac{x}{\Delta_u})$ functions with different slopes

The desired heading along the path is equal to the angle of the tangent, meaning

$$\psi_d(s_1) = \angle \boldsymbol{T}_d(s_1) = \text{atan2}(y_d^s(s_1), x_d^s(s_1)) \tag{4.16}$$

4.3 Phase 2 - Docking

For this phase it is assumed that the proximity sensors are activated. In this section, for simplicity, the measurements from the proximity sensors $l_{m,1}$ and $l_{m,2}$ will be denoted respectively l_1 and l_2 instead. From section 2.1 we also have that the distance between the sensors is denoted d.

4.3.1 Heading correction

The ship is parallel to the quay if $\Delta l = l_1 - l_2 = 0$, which implies if $\Delta l \neq 0$ the ship is not parallel to the quay. The latter will usually be the case when the ship arrives at the last waypoint, thus a heading correction is needed. The heading correction $\Delta\psi$ is calculated by

$$\tan(\Delta\psi) = \frac{\Delta l}{d} \implies \Delta\psi = \text{atan2}(\Delta l, d) \tag{4.17}$$

Thus, the reference angle for ψ should be

$$\psi \to \psi_d(s_1) + \sigma_\psi(t)\Delta\psi \tag{4.18}$$

where $\sigma_\psi\in \{0, 1\}$ is an activation function. The activation function is equal to zero for phase 1, and becomes 1 when the heading correction is initiated. This is typically at the instant when phase 2(docking) starts. An alternative is that the activation is user specified, meaning the operator determines when $\sigma_\psi = 1$ and the heading correction initiates.

The heading correction can also be low-pass filtered to generate a smoother heading signal for the controller instead of a step input.

4.3.2 Docking path

This method assumes that the vessel has arrived at the last waypoint just outside the quay. It includes a parameterized path starting from $p_d(s_1)$ and ending at a desired distance from the quay. It is also assumed that the unit normal vector $\boldsymbol{N}_d(s_1 = n)$ is approximately perpendicular to the harbor and pointing towards it. The normal unit vector from equation 4.8 is appropriate for port side docking, but for starboard docking the normal unit vector created by using the matrix $-\boldsymbol{J}$ must be used.

A path is created along the unit normal vector $N_d(s_1)$. Another path parameter s_2 is introduced to parameterize the path. The docking path is given as

$$\boldsymbol{p}_{dock}(s_1, s_2) = s_2\boldsymbol{N}_d(s_1) \tag{4.19}$$

It is worth noting that $s_2 = 0$ yields $p_{dock}(s_1, s_2) = 0$. In addition when $s_2 > 0$, the distance $|p_{dock}(n, s_2) - p_{dock}(n, 0)| = |s_2|$ because of the unit length of the normal vector $\boldsymbol{N}_d(s_1)$. This means the unit of s_2 is $[m]$.

For the docking, l_{ref} is the reference distance to the side of the harbor. Furthermore, there is two possible ways of using the distance measurements from the proximity sensors:

$$l(t) = \min\{l_1(t), l_2(t)\} \tag{4.20}$$

or

$$l(t) = \frac{l_1(t) + l_2(t)}{2} \tag{4.21}$$

Primarily method 4.20 is used. The speed assignment for the path variable $\dot{s}_2$ is proposed as

$$\dot{s_2} = \sigma_p(t)u_{dock}\tanh(\frac{\tilde{l}}{\Delta_p}) \tag{4.22}$$

where $\tilde{l} = l - l_{ref}$, u_{dock} is the desired docking speed and Δ_p is a gain which decides the slope when $\tilde{d} = 0$. In addition, another activation function $\sigma_p \in \{0, 1\}$ is introduced. The activation function will be zero for the path following, and 1 when the docking operation starts. It can be activated by a user input, an autonomous layer or both. The purpose of the speed assignment is to ensure that the ship moves slow and carefully along the docking path $p_{dock}(s_1, s_2)$. With the proposed speed assignment, it is made sure that when $\tilde{l} > 0$ a positive speed towards the quay is commanded. A negative speed away from the harbor is dictated when $\tilde{l} < 0$ and zero speed when $\tilde{l} = 0$. Thus, a asymptotic stability property of s_2 is implied at the position where $\tilde{l} = 0$.

Continuing on the topic of stability if t_0 is the starting time of the docking procedure, the initial measurement $l_0 = l(t_0)$ can be registered as the ship is positioned at the last waypoint $p_d(n)$. For the docking path, it is evident that

$$l = l_0 - s_2 \tag{4.23}$$

$$\tilde{l} = l - l_{ref} = l_0 - s_2 - l_{ref} = -(s_2 - s_{2,ref}) := -\tilde{s_2} \tag{4.24}$$

where $s_{2,ref} = l_0 - l_{ref}$. Consequently, setting the activation function to be $\sigma_p = 1$ results in

$$\dot{\tilde{s_2}} = \dot{s_2} = u_{dock} \tanh(\frac{\tilde{l}}{\Delta_p}) = u_{dock} \tanh(\frac{-\tilde{s_2}}{\Delta_p}) = -u_{dock} \tanh(\frac{\tilde{s_2}}{\Delta_p}) \tag{4.25}$$

which confirms that $\tilde{s_2} = 0$ is UGAS.

4.4 Combined reference

The reference signal from the path following phase and the docking phase is combined in to one reference signal. The result is the following guidance system:

$$\dot{s_1} = \frac{u_d(t)}{|\boldsymbol{p}_d(s_1)|}\omega(s_1) \tag{4.26}$$

$$\dot{s_2} = \sigma_p(t) u_{dock} \tanh(\frac{\tilde{l}}{\Delta_p}) \tag{4.27}$$

$$\psi \to \psi_d(s_1) + \sigma_{psi}(t)\,\text{atan2}(\Delta l, d) \tag{4.28}$$

$$\boldsymbol{p} \to \boldsymbol{p}_d(s_1) + \boldsymbol{p}_{dock}(s_1, s_2) = \boldsymbol{p}_d(s_1) + s_2 \boldsymbol{N}_d(s_1) = \boldsymbol{p}_d(s_1) + s_2 \boldsymbol{J} \frac{p_d^s(s_1)}{|\boldsymbol{p}_d^s(s_1|} \tag{4.29}$$

The velocity reference can be obtained by differentiation of equation 4.29. This yields

$$\dot{\boldsymbol{p}}_d = \boldsymbol{p}_d^{s_1} \dot{s}_1 + \boldsymbol{p}_{dock}^{s_2} \dot{s}_2 \tag{4.30}$$

Regarding the activation functions σ_ψ and σ_p, they are activated at the same instant when $s_1 < n - \mu$. In this context μ is a low threshold value, as when $s_1 \approx n$ the ship is approximately at the last waypoint. An other possibility is to activate the heading correction first by activating σ_ψ, and then only when the desired heading is achieved the docking path can be activated. In this case the ship will already be parallel with the harbor, when the docking motion is initiated.

An alternative method, which is probably just as good or even better, is to use a circle of acceptance. Meaning when the position of the ship has stabilized within a certain radius of the last waypoint for a desired amount of time, the activation function can be turned on.

4.5 Guidance system for the docking phase

This section concerns the docking controller used in method 1. As mentioned in section 3.1.2, the desired heading is achieved when $l_1 = l_2$, meaning the ship is parallel to the side of the harbor. This is used in the heading controller. For the surge control, the setpoint for the surge speed controller will be $u_d = 0$. If the other controller for surge is used, i.e. the surge position controller, the desired position will be given in the harbor fixed reference system as x^h_d.

Regarding the sway controller, the reference distance from the harbor is as mention earlier l_{ref}. A reference model is used for the sway controller, to generate a smooth reference trajectory from outside the harbor to the reference distance. The reference model used is a low-pass filter, given by

$$\frac{l_d}{l_{ref}}(s) = h_{lp} = \frac{\omega_c}{s + \omega_c} \tag{4.31}$$

where $\omega_c = 1/T_c$ is the cutoff frequency. The desired speed is set to be the time derivative $\dot{l}_d$. The cutoff frequency is selected so that the ship is slowly taken close to the quay. For the proximity sensor measurement, equation 4.20 is used.

Chapter 5

Experimental Setup

This chapter concerns the experimental setup used. This mainly involves the C/S Inocean Cat I Drillship (CSAD) model with its hardware and software, the Qualisys Tracking Manager(QTM), the basin and the wooden dock.

5.1 CSAD

A significant part of the information in this section is based on Bjørnø (2016) and NTNU (2017), so for additional information regarding software, hardware and the construction of CSAD, the reader is referred to those two sources.

The model is a 1:90 scale model of the Statoil Cat I Arctic Drillship, and the main dimensions of CSAD are:

Table 5.1: Main dimensions of CSAD

Parameter	Value
LOA	2.578 [m]
B	0.440 [m]
D	0.211 [m]
T	0.133 [m]
Δ	127.92 [kg]

5.1.1 Actuators

The actuators on CSAD are six azimuth thrusters, three in the fore of the ship and three in the aft of the ship. The position of the thrusters can be found in table 5.2 and is shown in figure 5.1. The thrusters are positioned according to the full-scale design of the ship.

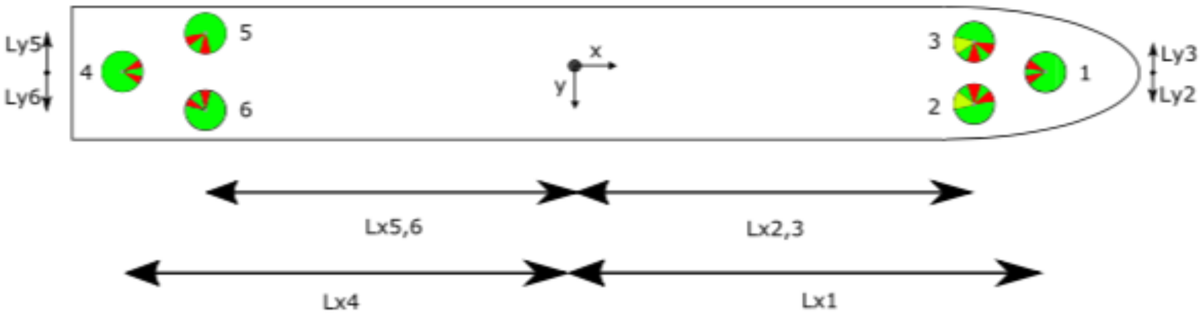


Figure 5.1: Position of actuators. [Courtesy: Frederich (2016)]

Table 5.2: Position of the azimuth thrusters

Thruster	Position X [m]	Position Y [m]
Thruster 1	1.0678	0.0
Thruster 2	0.9344	0.11
Thruster 3	0.9344	-0.11
Thruster 4	-1.1644	0.0
Thruster 5	-0.9911	-0.1644
Thruster 6	-0.9911	0.1644

The thrusters are of the type Aero-naut Precision Schottel, and the diameter of the thrusters are 30 millimeters.

5.1.2 Hardware control system and Power supply

The power source for the ship are 6 parallel connected 12V 12Ah batteries. They are placed close to the middle of the ship, three on each side of the moon-pool. The control system on the ship consists of

- One National Instruments (NI) compact Reconfigurable Input/Output (cRIO) embedded controller.
- One Raspberry Pi (RPi) single-board computer.
- Six Electronic Speed Controller (ESC), one for each thruster. Each of them are connected to six motors and controls the thruster speed.
- Six servos, one for each thruster, controlling the angle of the thruster.

The cRIO on the ship is the model called cRIO-9024, and it is a real-time embedded industrial controller. Using the NI software called VeriStand it runs real-time control systems written in either LabView or Simulink. It is connected to four FPGA modules for both analog and digital Input/Output.

The RPi is responsible for the communication between the cRIO and the proximity sensors described in section 5.1.3. Usually there is another RPi which provides comunication between the cRIO and a Sixaxis controller, in the form of a Playstation controller, which can be used for manual control of the vessel. But as the cRIO only has room for one of them to be connected, the Sixaxis controller was removed and replaced by the RPi for the proximity sensors. This removes the ability to manually control the vessel. When the RPi is connected it will continuously send the signals from the two proximity sensors to the cRIO over an ethernet cable. The RPi works as an embedded system and is connected to the two proximity sensors through two wires, i.e. the connection between the proximity sensors and the RPi is not wireless.

The thrusters are driven by brushless UMA-2820-950 DC motors. The motors are connected to ESC's of the type O.S. OCA-150 50 A BL. The ESC's are controlled with PWM signals. Originally the motors have a lot more power than necessary, so the power is constrained.

The servos controlling the angle of the thrusters are geared one to one with the thruster. This means that one degree rotation of the servos results in one degree rotation of the thrusters. The servos are of the type Dynamixel MX-106R. Initially they were connected to the thrusters with zero offset, but regular use of the boat has gradually shifted the offsets between the servos and the corresponding thrusters. Table 5.3 contains information about the latest servo offsets.

Table 5.3: Angle offset between servo and thrusters

Thruster	Offset [Deg]
α_1	-94
α_2	-33
α_3	7
α_4	11
α_5	82
α_6	58

5.1.3 Proximity Sensors

The two installed proximity sensors are of the type HC-SR04, and uses sonar to measure distance to an object (Cytron-Technologies, 2013). Table 5.4 gives the main features of the sensors.

Table 5.4: Main features of the proximity sensors

Feature	Value
Power Supply	+5V DC
Quiescent Current	<2mA
Working Current	15mA
Effectual Angle	< 15°
Ranging Distance	2cm – 400 cm/1" - 13ft
Resolution	0.3 cm
Measuring Angle	30°
Trigger Input Pulse width	10uS
Dimension	45mm x 20mm x 15mm

The sensors both have a ultrasonic transmitter and receiver module. The timing diagram is shown in figure 5.2.

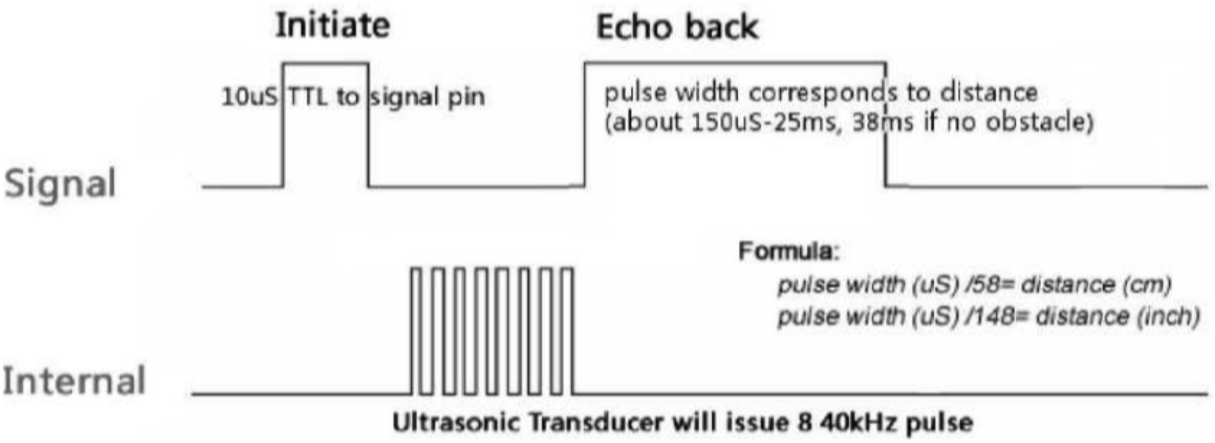

Initiate
Echo back
10uS TTL to signal pin
pulse width corresponds to distance
(about 150uS-25ms, 38ms if no obstacle)
Signal
Formula:
pulse width (uS) /58= distance (cm)
pulse width (uS) /148= distance (inch)
Internal
Ultrasonic Transducer will issue 8 40kHz pulse

Figure 5.2: Timing diagram of HC-SR04 proximity sensor. [Courtesy: Cytron-Technologies (2013)]

To start the measurement the sensor must receive a pulse of 5V for at least 10μs. This will trigger the sensor which will transmit a cycle of eight ultrasonic bursts at 40 kHz and wait for it to be reflected. When the receiver detects the reflected signal, it will set the Echo pin to high (5V) and delay for a period (width) which is proportional to the distance. To calculate the distance, it uses the width of the Echo pin. The distance in centimeters can be calculated by

$$d = \frac{t}{58} \tag{5.1}$$

where t is the width of the Echo pulse in microseconds. The distance can also be calculated by using the speed of sound according to equation 1.1.

Figure 5.3 shows one of the proximity sensors installed on the ship. They are fastened using duct tape, as it was necessary to be able to remove and attach them easily. The sensors are very light, meaning the duct tape has no problem with keeping the sensors in place. The sensors are connected 1.6 cm from the edge of the ship, which is important to know when specifying the desired distance between the side of the ship and the dock in the control system.

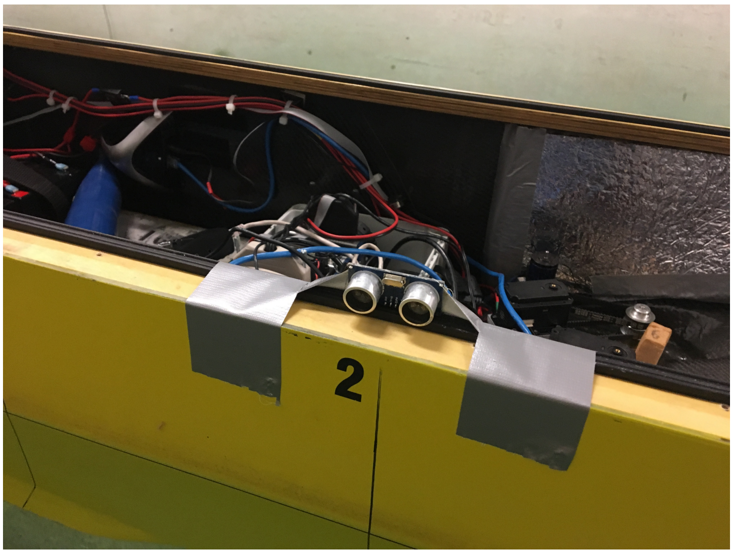


Figure 5.3: Proximity sensor mounted on CSAD

Figure 5.4 shows where the two proximity sensors are mounted to the model, one in the front and one in the aft. The distance between the sensors where measured to be $d = 1.29$ meters.

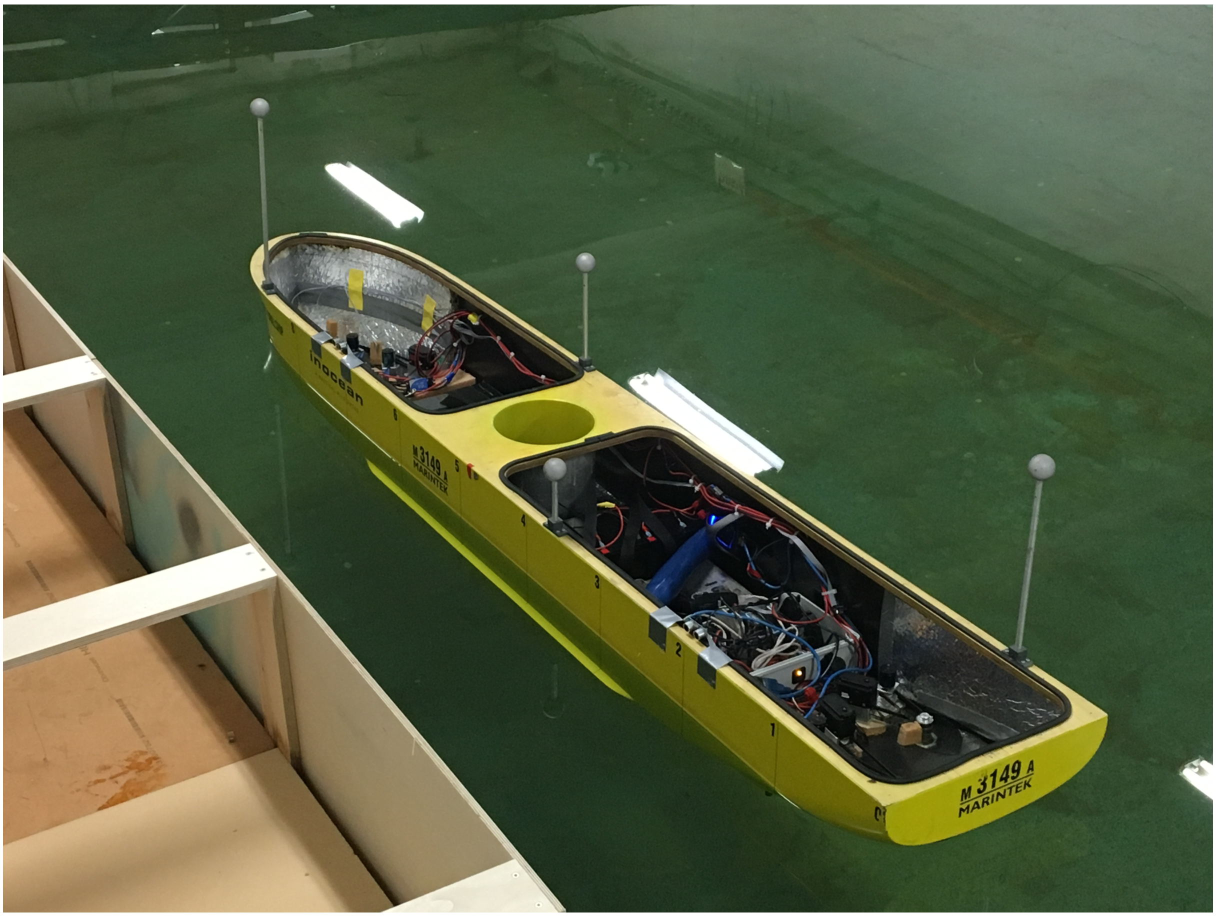


Figure 5.4: The two proximity sensor mounted to the side of CSAD

The proximity sensor signals where filtered using a low-pass filter, to reduce noise and remove wildpoints. The measurements where filtered by using

$$l_{pro} = \frac{\omega_c}{s + \omega_c} l \tag{5.2}$$

where l_{pro} is processed/filtered signal, ω_c is the cut-off frequency and l is the proximity sensor measurement.

5.2 Qualisys Tracking Manager and Oqus

The QTM software transmits the position and heading signal of the vessel to the cRIO over the MC-lab Wifi. The position and heading of the vessel is obtained by triangulation. Four spheres are attached to the vessel and three Oqus cameras are used to find the position of the the spheres. The position of the spheres are then used to calculate the heading and position of the vessel. The spheres can be seen in figure 5.4. On-board the vessel the cRIO reads the QTM position data through a Custom Device module named Oqus. The Oqus software listens to the network data from QTM, and forwards the position and heading signal to the software control system.

5.3 System overview and communication

How the information flows between the hardware components on CSAD is of importance. Figure 5.5 below shows the information flow. The dashed lines means the information is transported through Wifi, and the solid lines means the information is transported using wires. For example the information from the RPi to the cRIO goes over Ethernet.

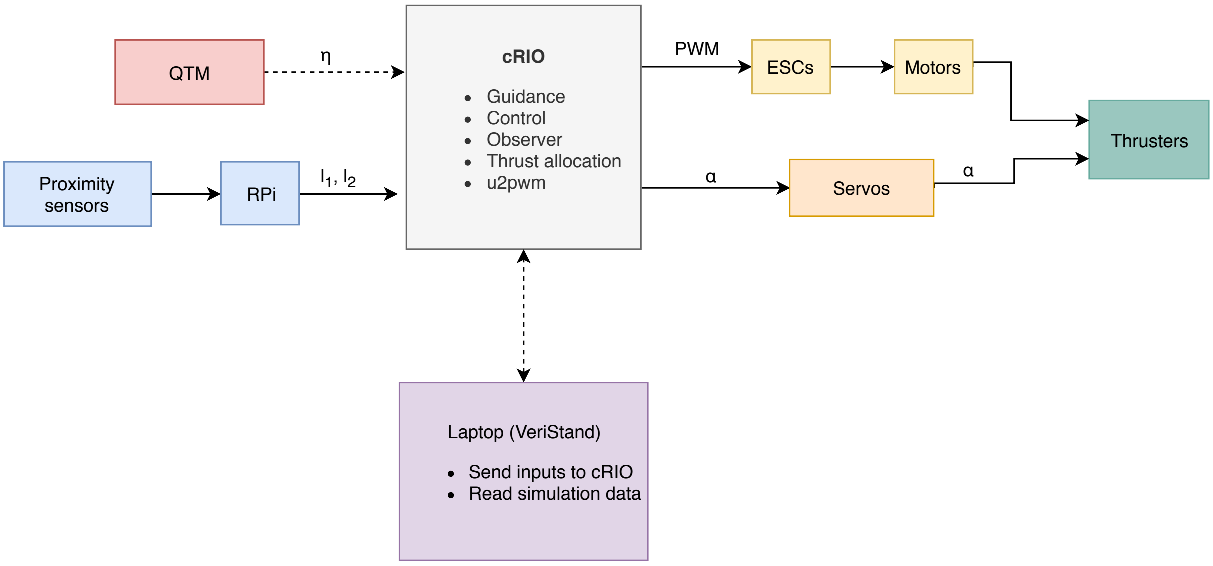


Figure 5.5: Information flow between the hardware components on CSAD

As mentioned before there is one ESC, motor and servo for each of the six thrusters.

A topology drawing of the software can be found in figure 5.6.

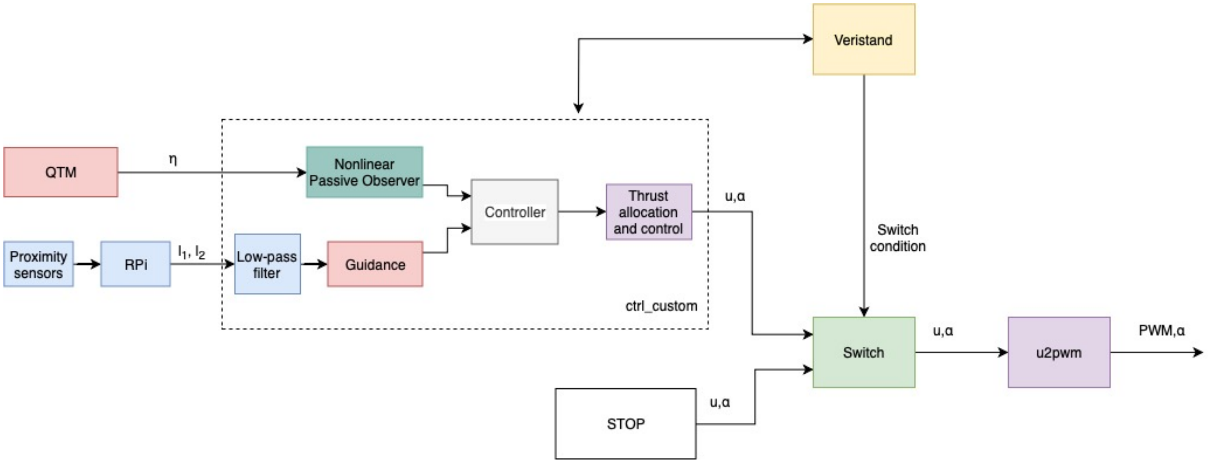


Figure 5.6: Software topology drawing

The box defined by the dotted line is called *ctrl_custom* and contains the observer, controller, guidance system and thrust allocation. In other words, these are the systems described in chapters 3 and 4, in addition to the thrust allocation. This corresponds to the simulink diagram called *ctrl_custom.slx*. The thrust allocation for the experiments is treated as a black box, but it is a thrust allocation from Lyngstadaas (2018) where the angles of all the thrusters are fixed. The thrust allocation from the simulations was replaced in the experiments because it did not work properly. The most apparent problem was that all of the thrusters seemed to spin continuously, resulting in uncontrolled motion of the vessel. The fixed thrust allocation worked much better due to its simplicity and predictability.

The other essential simulink diagram is called *u2pwm.slx*. This converts the desired force for each thruster, **u**, to PWM signals which are input to the ECS's. From figure 5.6 it can be seen that there is a mode called STOP. This system simply stops the vessel, by setting the thrust of all thrusters to zero. There is a switch mechanism that changes between the STOP mode and the *ctrl_custom* mode. The switch condition is a boolean value, specified by an operator through the VeriStand user interface. The switch mechanism and STOP mode are actually a part of the Simulink diagram u2pwm.slx, but for illustration purposes they were separated in the topology diagram.

The VeriStand project run on a separate laptop, which communicates with the cRIO through the MC-lab Wifi. The VeriStand project contains a user interface where signals can be plotted real-time, and specify inputs that can be sent to the cRIO.

5.4 Construction of dock

A dock was constructed using plywood plates, wooden planks and a buoyancy element. The buoyancy element consists of two plates of a plastic material called divinycell H60. The plywood plates are used as the side of the dock. Wooden planks are used to stiffen up the construction and to create a framework which the plywood plates are connected to. The buoyancy element works as a stable platform and is used to make the dock float. Figure 5.7 shows the dock in the basin in the MC-lab.

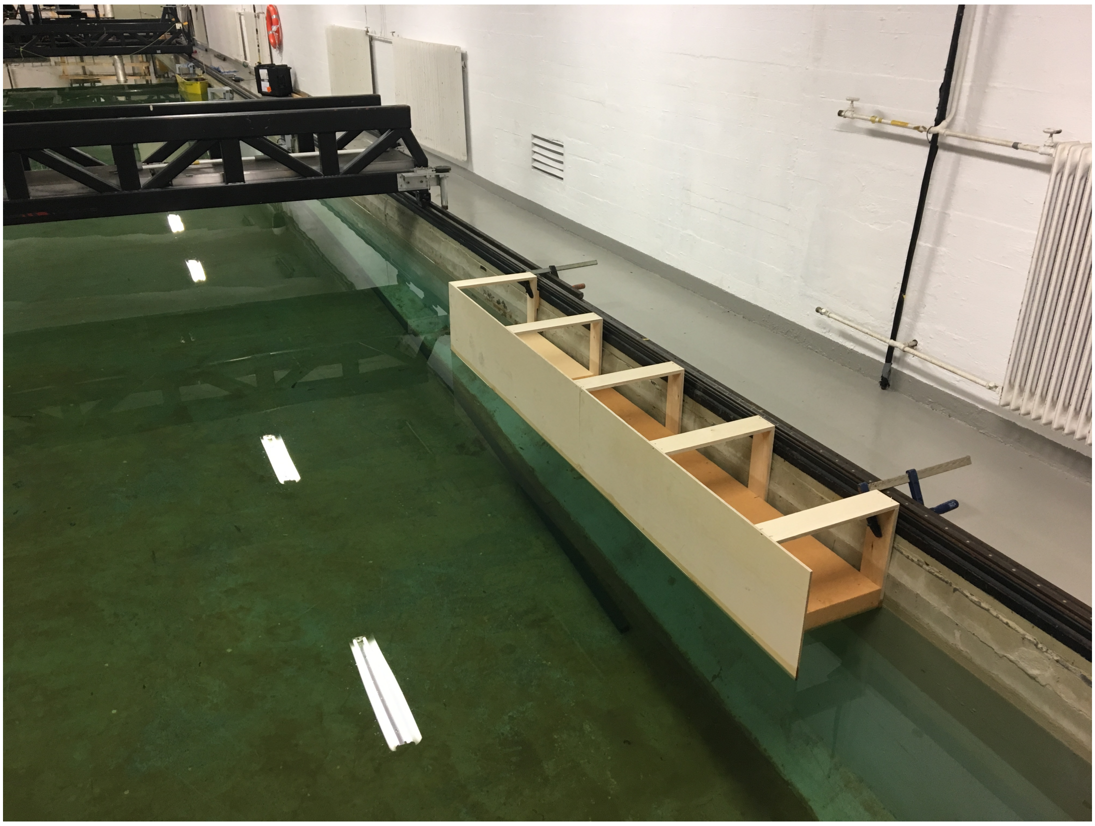


Figure 5.7: Dock for the experiments

Two forcing tools, one at each end, are used to fasten the dock to the edge of the basin. Figure 5.8 displays how this is done. The forcing tools can also be seen in figure 5.7.

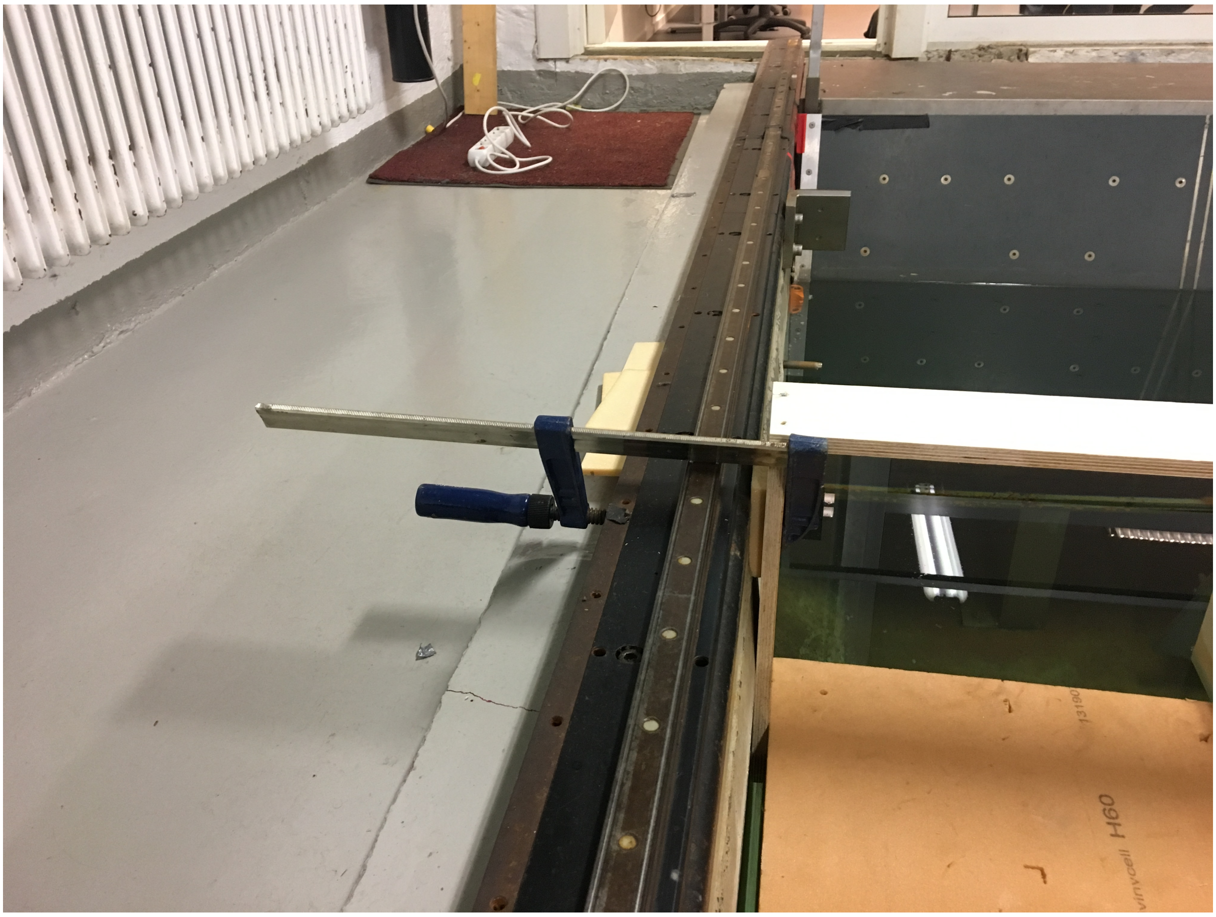


Figure 5.8: Forcing tool fastening the dock to the side of the basin

The purpose of the dock is to create a clear docking location for the CSAD model ship. In addition the plywood plates provides a straight and even surface for the proximity sensors to work with. The alternative was to make the ship dock against the side of the basin itself. This would probably work fine, but the Qualisys tracking system struggles with dropouts close to the sides of the basin. Thus the floating dock helps move the docking location approximately half a meter further into the basin, where the Qualisys tracking system works better due to less signal dropouts.

5.5 Reference filter for experiments

As mentioned earlier the ship cannot be controlled with the sixaxis controller, due to the limited number of input ports in the cRIO on-board CSAD. The controller together with a reference model consisting of a low-pass filter cascaded with a mass-spring-damper system is used for maneuvering the vessel around in the basin. The reference filter can be expressed in according to Fossen (2011) in the vectorial form

$$\boldsymbol{\eta}_d^{(3)} + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\eta}}_d + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d + \boldsymbol{\Omega}^3\boldsymbol{\eta}_d = \boldsymbol{\Omega}^3\boldsymbol{r}^n \tag{5.3}$$

where the damping is set to be $\boldsymbol{\Delta} = \boldsymbol{I}$, the natural frequencies $\boldsymbol{\Omega} = diag(0.11, 0.11, 0.15)$ and $\boldsymbol{r}^n$ is the reference input.

Chapter 6

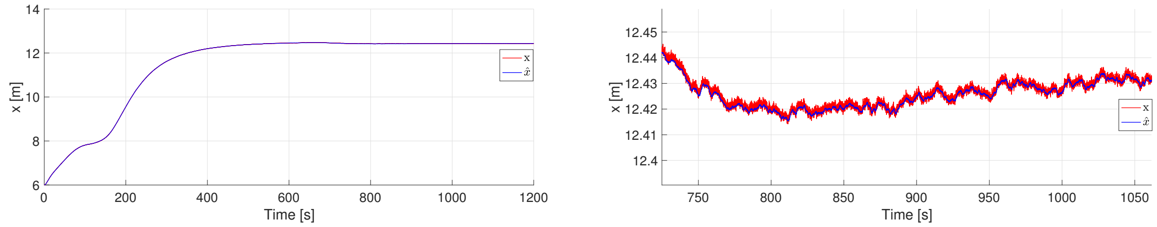
Results

This chapter contains the results of the two docking methods used for the simulations, and the results from the experiments in the MC-Lab with the CSAD model. This chapter will mainly present the results while the discussion of the results can be found in chapter 7. Although some key observations and performance assessments regarding the results will be included in this chapter.

6.1 Observer verification

6.1.1 Simulations

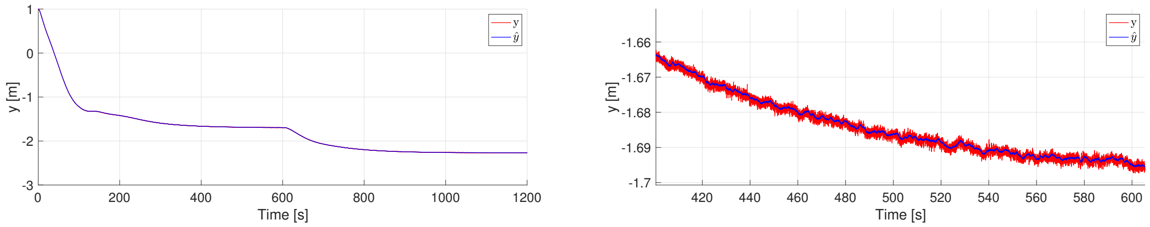
The following results of the observer is from method 2, but the same observer has been used for both methods. The position and heading signal is disturbed by both waves and measurement noise. Figures 6.1, 6.2 and 6.3 shows the estimated position and heading against the measured corresponding values. The estimated value is represented by the blue line and the measured value is the red line.



(a) x-position vs. estimated x-position

(b) Zoomed in view of the x-position vs. estimated x-position

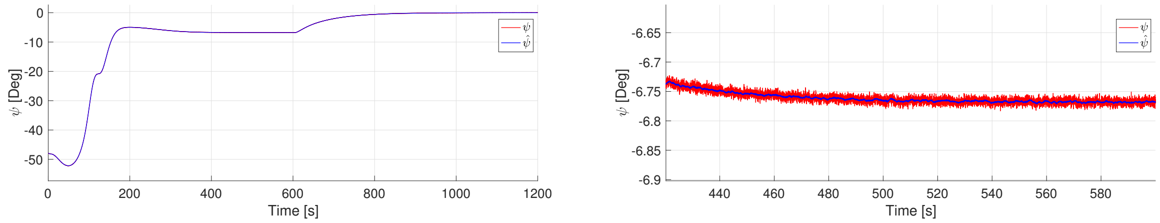
Figure 6.1: Observer verification for the x-position



(a) Measured y-position vs. estimated y-position

(b) Zoomed in view of the measured y-position vs. estimated y-position

Figure 6.2: Observer verification for the y-position



(a) Measured heading vs. estimated heading

(b) Zoomed in view of the measured heading vs. estimated heading

Figure 6.3: Observer verification for the heading.

Figures 6.4, 6.5 and 6.6 shows the estimated velocities against the true velocities. The red line is the actual velocity and the blue line is the estimated velocity.

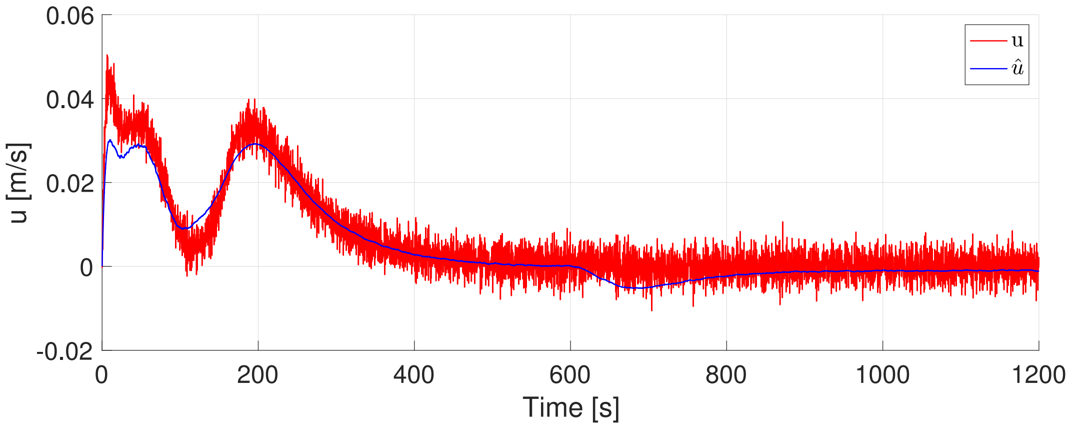


Figure 6.4: Simulation surge velocity estimation

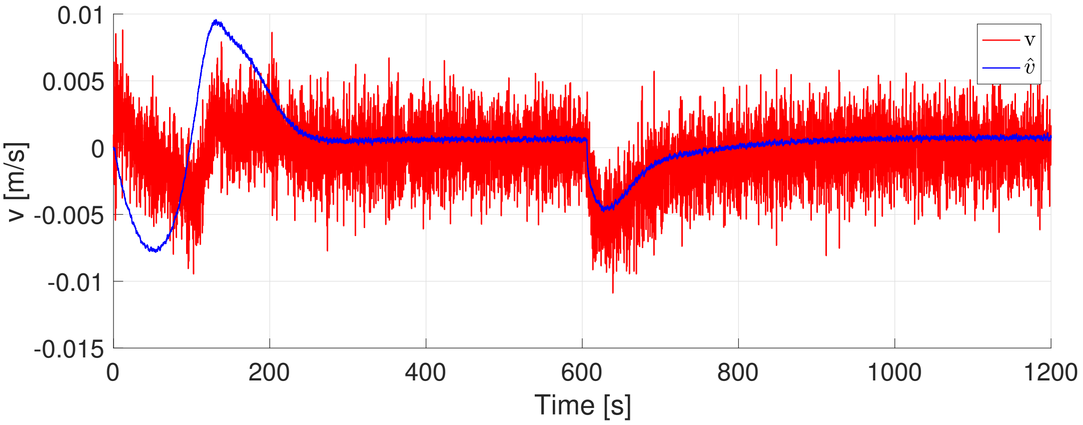


Figure 6.5: Simulation sway velocity estimation

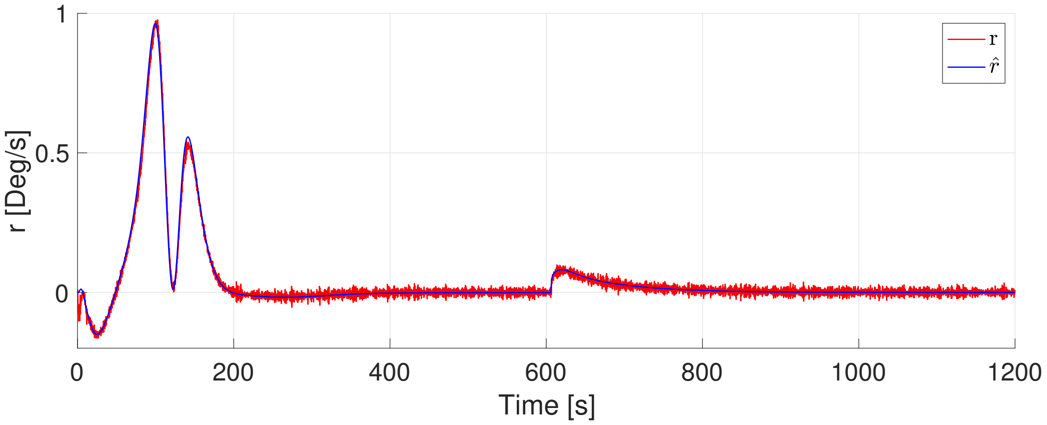
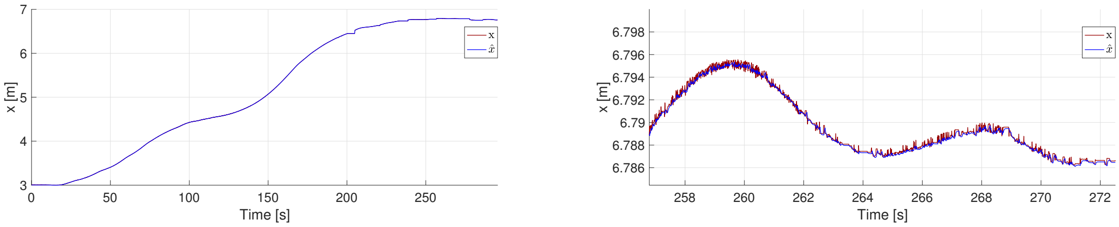


Figure 6.6: Simulation yaw velocity estimation

6.1.2 Experiments

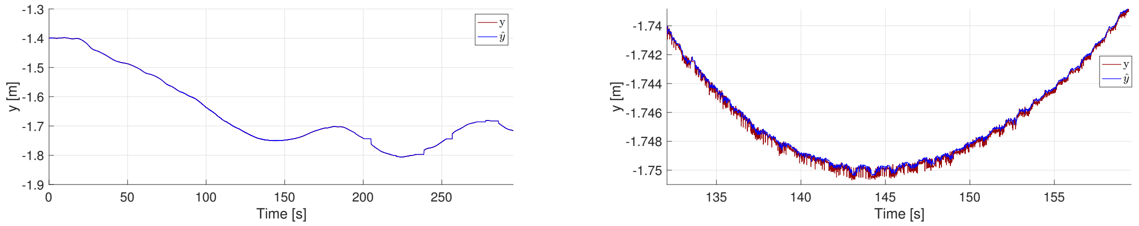
The observer verification for the experiments was executed as a separate and individual run. Figure 6.7, 6.8 and 6.9 shows the estimated position and heading against the corresponding measurement signals.



(a) Measured x-position vs. estimated x-position

(b) Zoomed in view of the x-position vs. estimated x-position

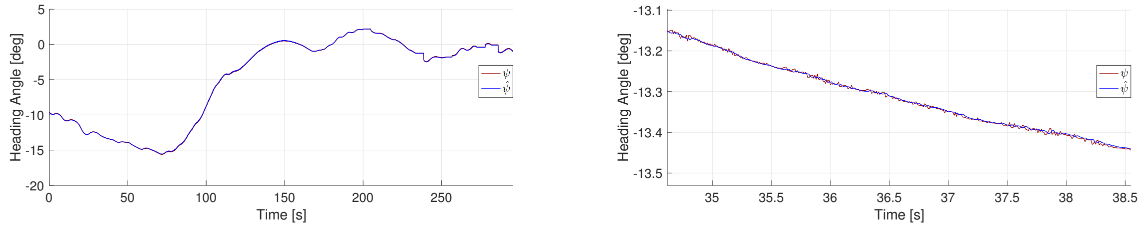
Figure 6.7: x-position observer verification for the experiments



(a) Measured y-position vs. estimated y-position

(b) Zoomed in view of the measured y-position vs. estimated y-position

Figure 6.8: y-position observer verification for the experiments



(a) Measured heading vs. estimated heading

(b) Zoomed in view of the measured heading vs. estimated heading

Figure 6.9: Heading observer verification for the experiments.

For the experiments there is no true velocity to compare the velocity estimates against. Figure 6.10, 6.11 and 6.12 shows the estimated surge velocity, sway velocity and yaw velocity respectively.

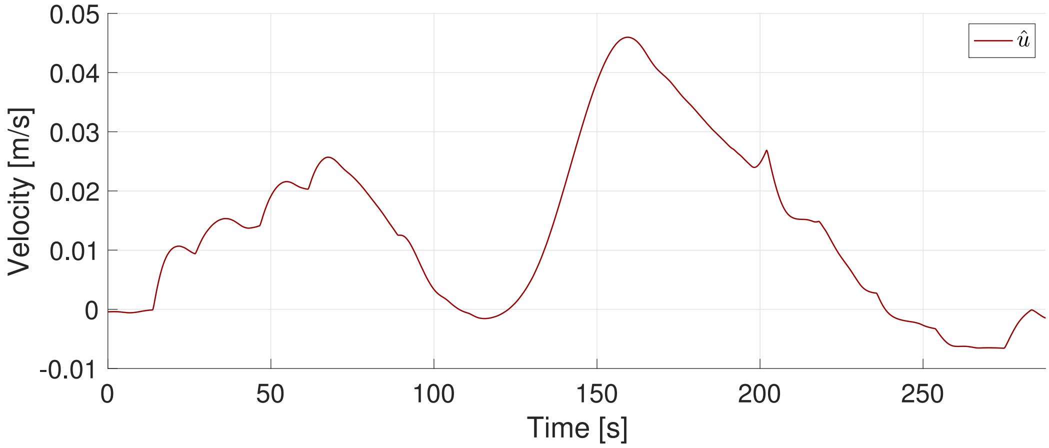


Figure 6.10: Surge velocity estimate for the experiments.

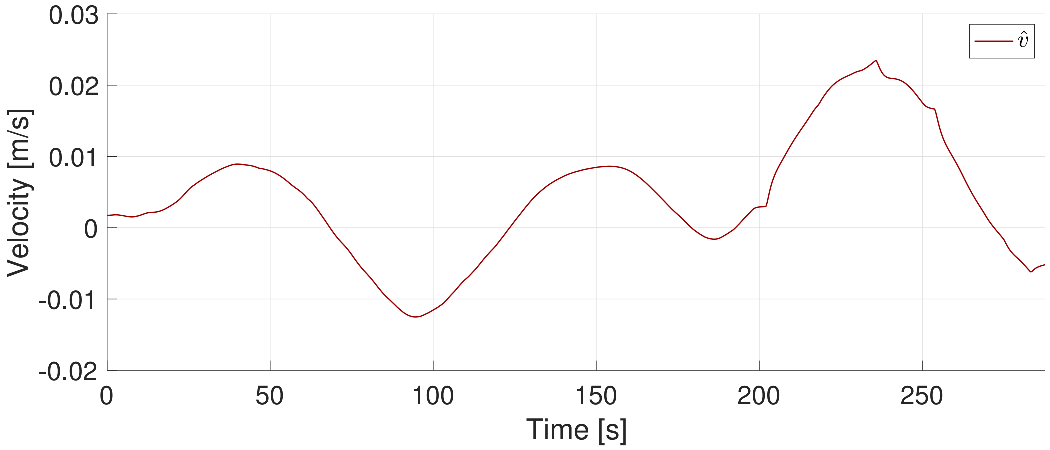


Figure 6.11: Sway velocity estimate for the experiments.

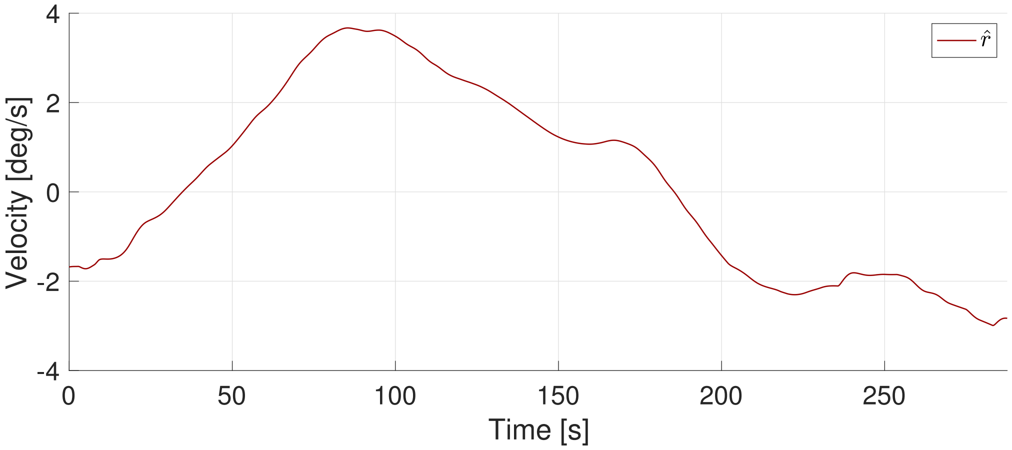


Figure 6.12: Yaw velocity estimate for the experiments.

6.2 Simulations

6.2.1 Method 1

For the simulation showcased in this section, the last waypoint is $p_n = (x_n, y_n) = (12.5, -1.7)$ and the path following speed is set to $u_d = 0.06$. If the speed is transformed to full scale using Froude scaling, the speed will be $u_d \approx 1.1$ knots. The desired distance between the ship and the harbor side is set to be $l_{ref} = 0.011$ meters, which in full scale is approximately 1 meter. The constant current is set to be $\nu_c = [0.1, -0.1]$. Figure 6.13 below, shows the xy-plot of the ships movement. The green lines represent the edge of the basin, and the red line is the side of the harbor.

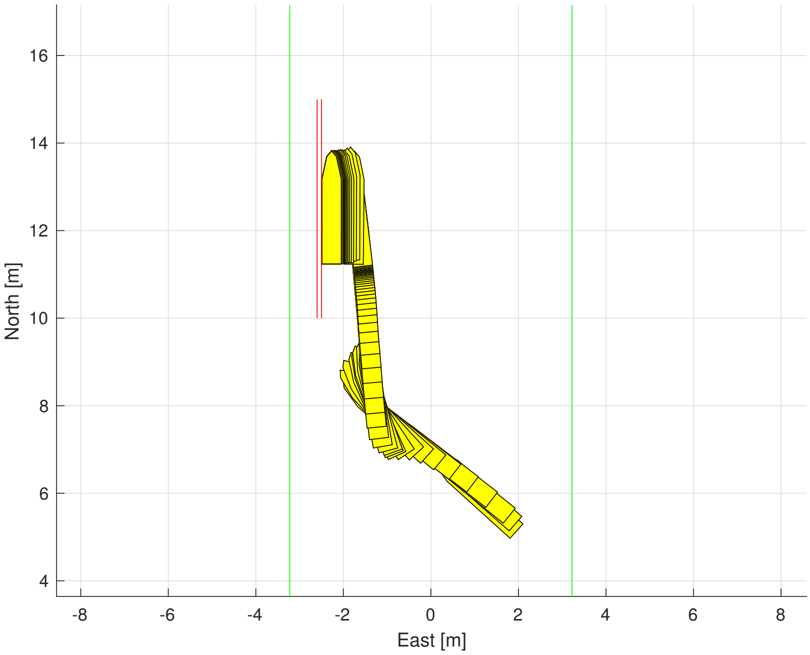


Figure 6.13: Resulting xy-plot of the docking operation for simulations of method 1.

Figure 6.14 shows the desired and estimated ship position in the x-direction, respectively represented with a red and blue line. Phase 2, i.e. the docking phase, is initiated just after 600 seconds has passed.

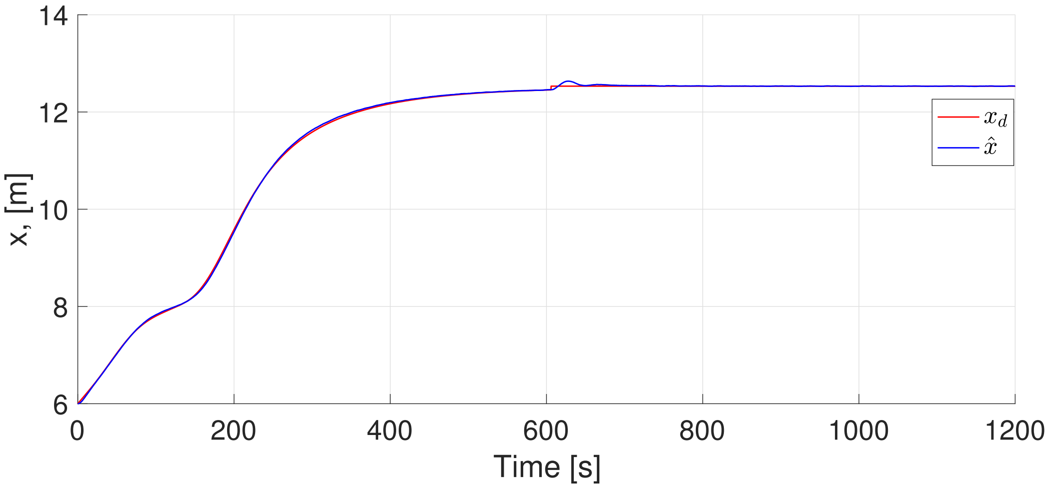


Figure 6.14: Simulation results from method 1 showing the desired x-position and estimated x-position.

Figure 6.15 shows the desired and estimated ship position in the y-direction.

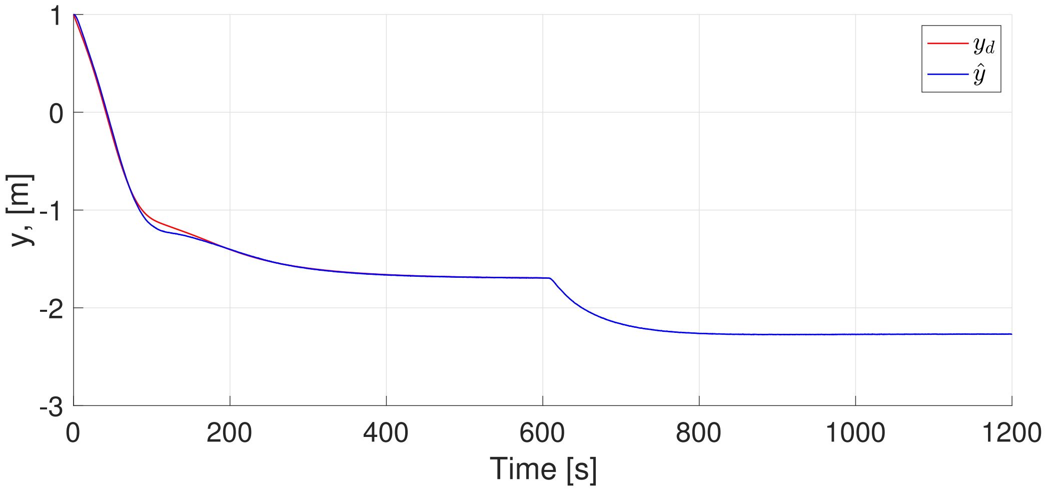


Figure 6.15: Simulation results from method 1 showing the desired y-position and estimated y-position.

Figure 6.16 shows the desired and estimated heading of the ship. For the docking phase, the control of the heading and the sway movement is not based on the position and heading measurements, but the proximity sensors. Thus the desired heading ψ_d and the desired y-position y_d does not exist during the docking phase.

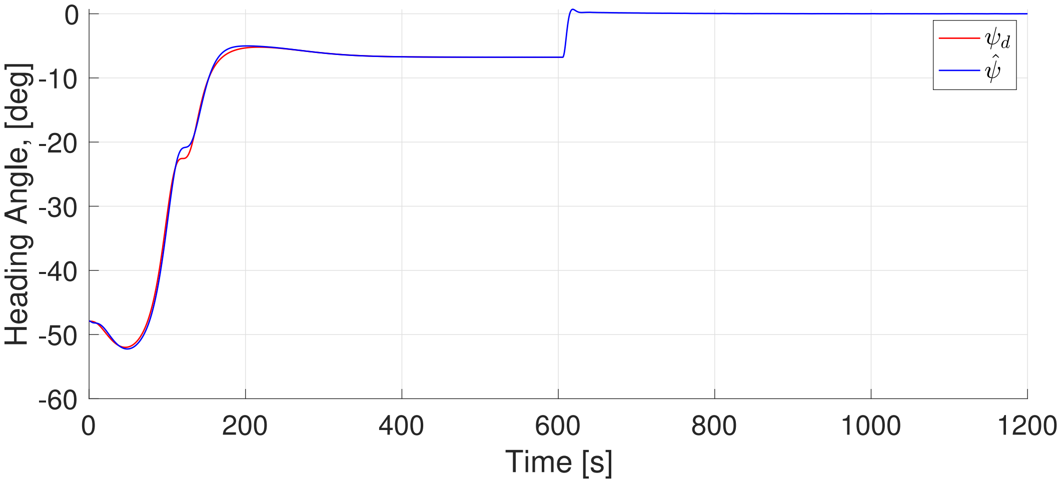


Figure 6.16: Simulation results from method 1 showing the desired heading and estimated heading.

Figure 6.14, 6.15 and 6.16 shows that the vessel follows the desired position and heading trajectories well. The docking phase is also carried out reasonably well. Although when the switch to the docking controller happens, a small jump in the response can be noticed. This is particularly visible in figure 6.14, but also to some extent for the heading. This is further discussed in section 7.2.

The measurements from the proximity sensors is shown in figure 6.17, and are represented by the red and blue lines. The green line is the reference distance, i.e. the desired distance between the side of the ship and the quay.

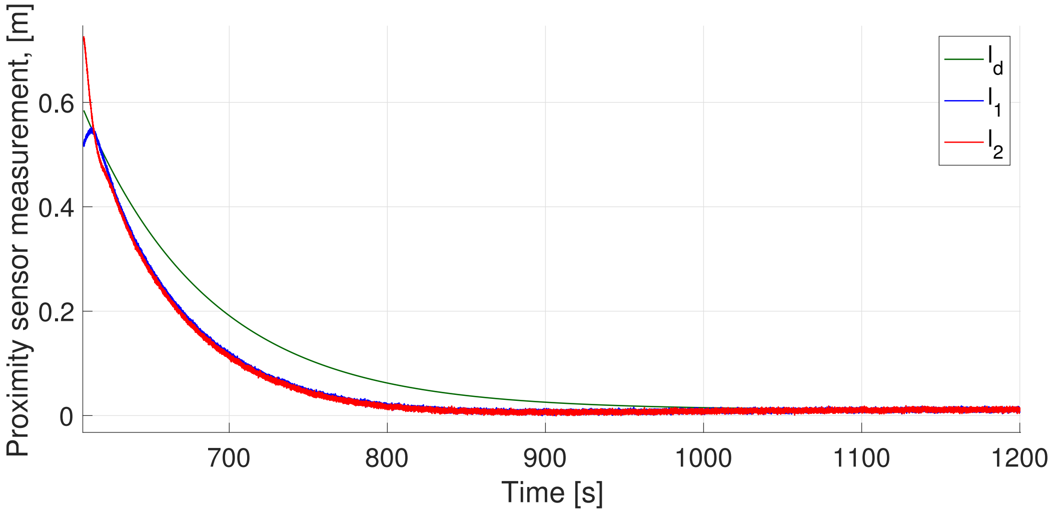
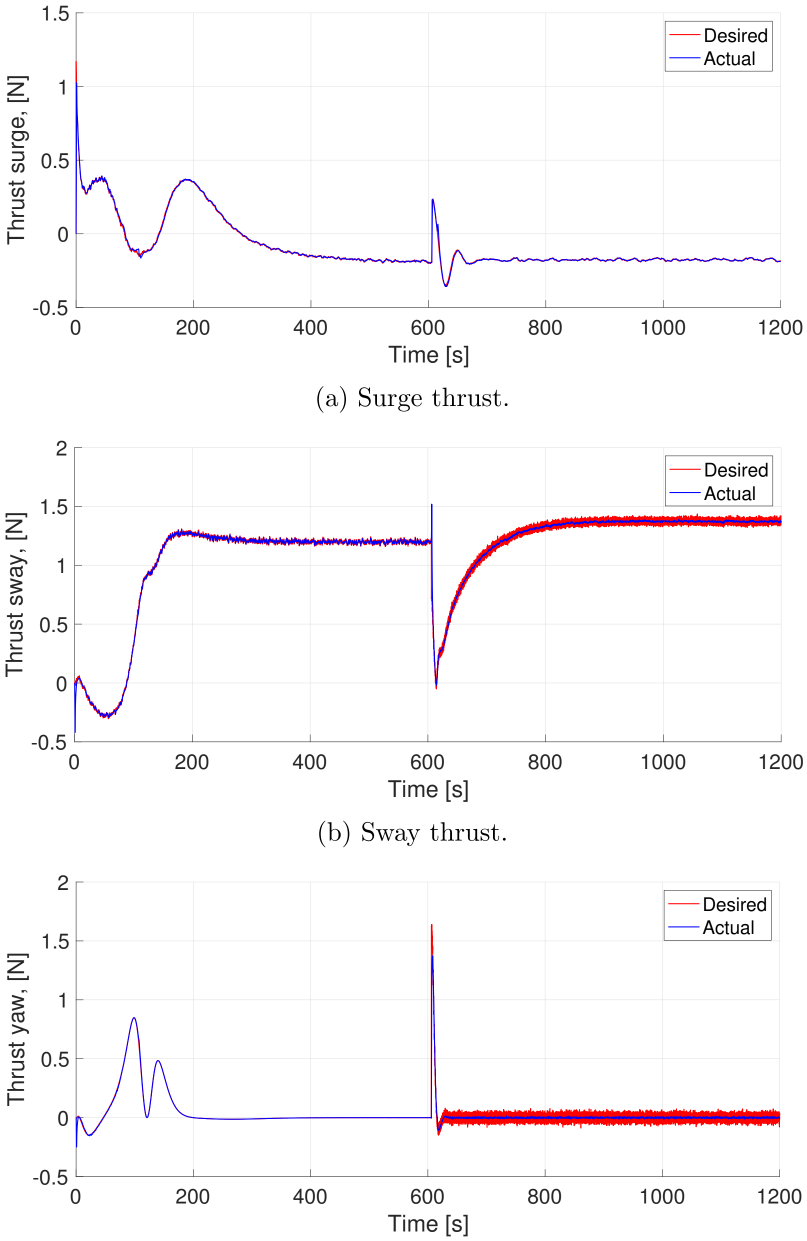


Figure 6.17: Simulation results from method 1 showing the measurements from the two proximity sensors and the reference distance.

The figure shows the heading controller manages to achieve the control objective of $l_1 = l_2$ very fast, meaning the ship is parallel to the harbor. The sway controller also manages to reach the desired reference distance from the harbor, although the sway motion is ahead of the desired sway motion as the proximity sensor measurements are lower than the desired distance between the harbor and the ship, l_d.

The commanded control forces from the controller and the actual applied forces after the thrust allocation algorithm is also of interest. Figure 6.18 displays the control forces and applied forces in the surge, sway and yaw direction.



(a) Surge thrust.

(b) Sway thrust.

(c) Yaw thrust.

Figure 6.18: Results from method 1 showing commanded thrust from the controller and applied thrust after the thrust allocation.

Figure 6.18b and 6.18c shows a change in the thrust between phase 1 and 2. The control forces become greater in amplitude and has a high frequency. The different behaviour is because of the different controllers used in the two phases.

6.2.2 Method 1 with stronger current

These results are included to show the performance of method 1 when the current is stronger. The full scale current is set to be $\nu_c = [0.3, -0.3]$, which is the same as for method 2 which is presented in section 6.2.3. Figure 6.19, 6.20 and 6.21 shows the desired and estimated ship position and heading for the stronger current.

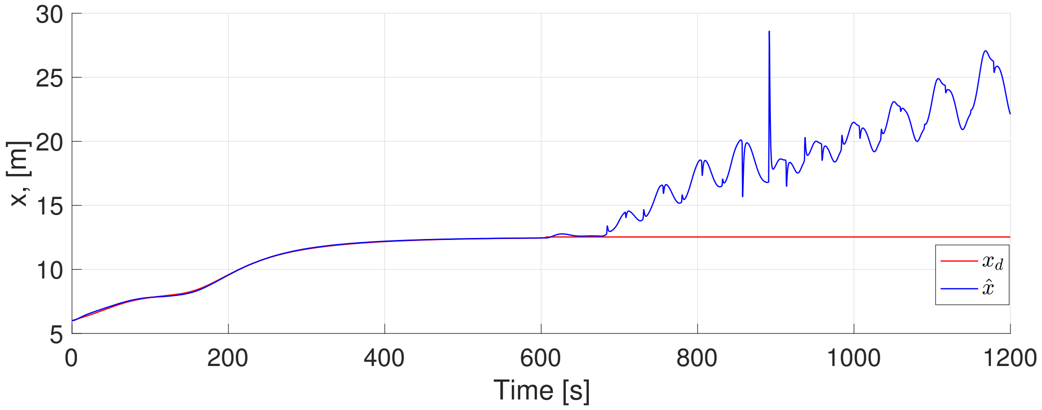


Figure 6.19: Position over time in the x-direction with stronger current.

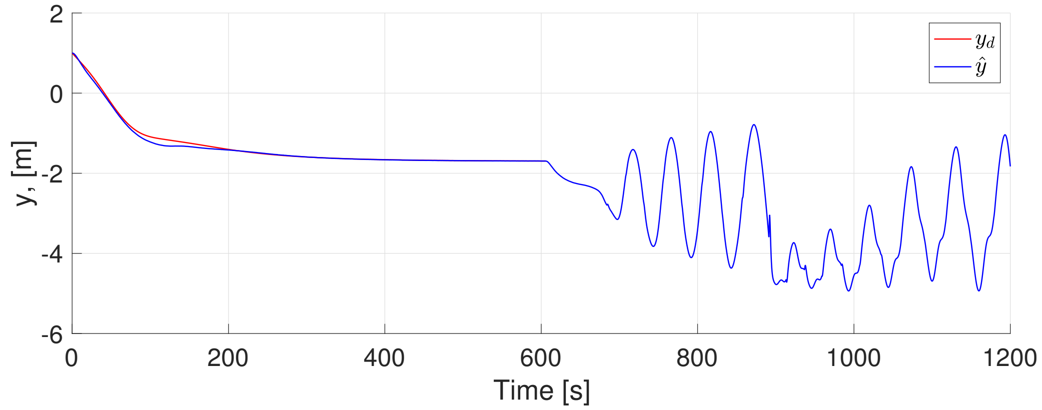


Figure 6.20: Position over time in the y-direction with stronger current.

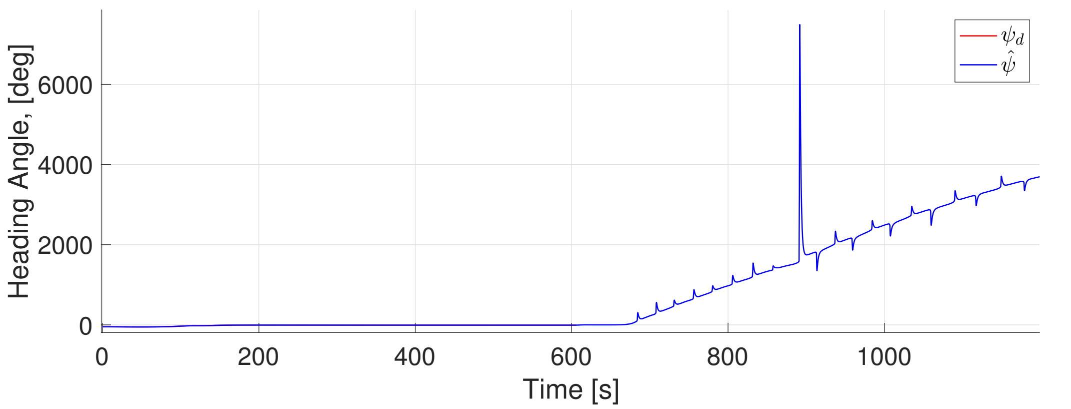


Figure 6.21: Heading over time with stronger current.

The plots clearly show that the method perform badly for a stronger current. It should be noted that much of the bad performance is due to how the proximity sensors are modeled. This is discussed in more detail in section 7.2. Nonetheless the case is included to show that the ship will hit the harbor. This can be seen from figure 6.20 knowing that the harbor is the straight line defined by y = -2.5. The ship collides with the harbor at $t \approx 676$. After this the modeling of the proximity sensors will be insufficient, and the results will not make much sense.

6.2.3 Method 2

For the simulation showcased in this section, the last waypoint is $p_n = [12.5, -1.7]$ and the path following speed is set to $u_d = 0.06$. If the speed is transformed to full scale using Froude scaling, the speed will be $u_d \approx 1.1 knots$. The desired distance between the ship and the harbor side is set to be $l_{ref} = 0.011$ meters, which is approximately one meter in full scale. The constant current is set to be $\nu_c = [0.3, -0.3]$ for large scale. Figure 6.22 shows the ship at several time instants from the start position until the final desired docking state. The time step between each plotted position of the ship is constant. Again the green lines represent the edge of the tank, while the red line is the edge of the harbor.

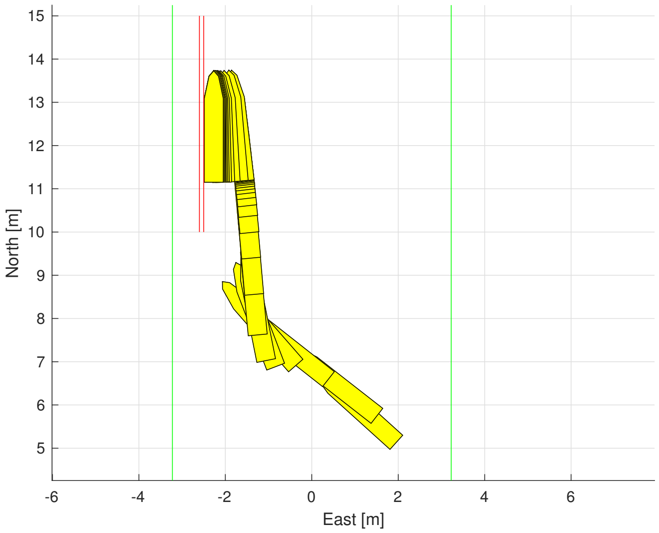


Figure 6.22: Resulting xy-plot of the docking operation for simulations of method 2.

Figure 6.23 shows the desired and estimated ship position in the x-direction. Phase 2, i.e. the docking phase, is initiated just after 600 seconds has passed.

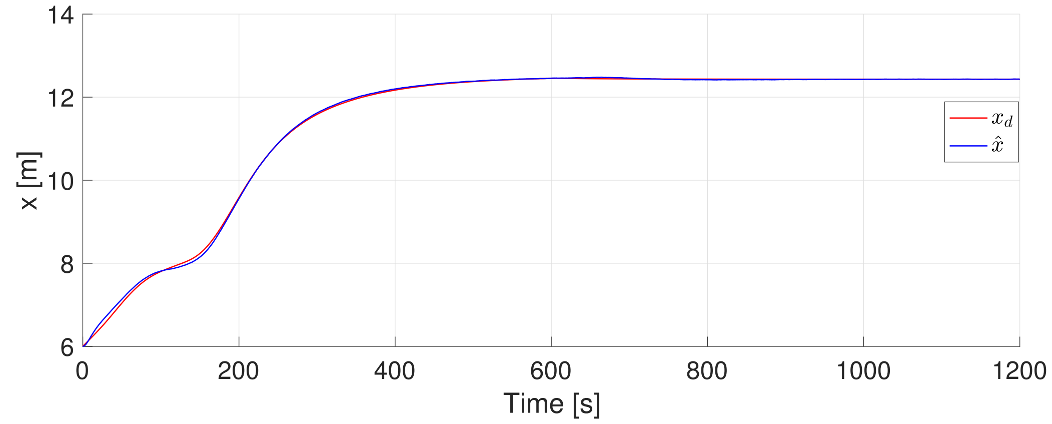


Figure 6.23: Simulation results from method 2 showing the desired x-position and estimated x-position.

From figure 6.24 shows the desired and estimated ship position in the y-direction.

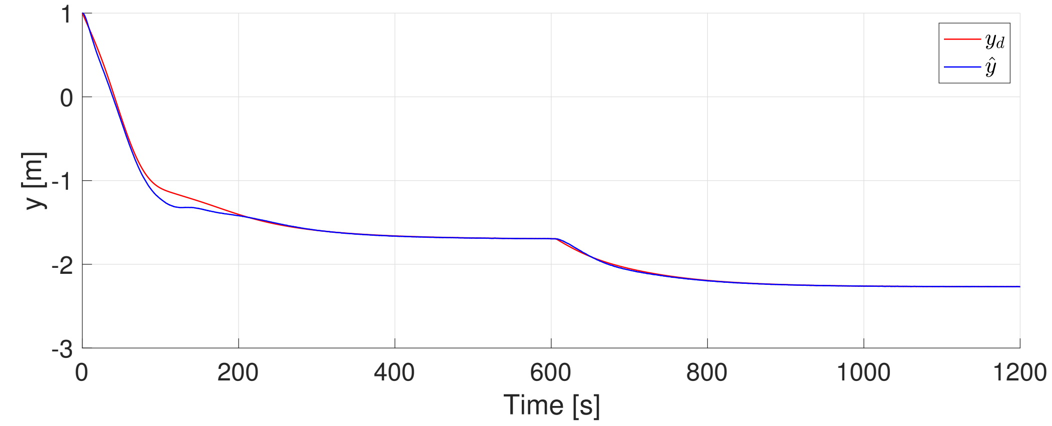


Figure 6.24: Simulation results from method 2 showing the desired y-position and estimated y-position.

Figure 6.25 shows the desired and estimated heading of the ship.

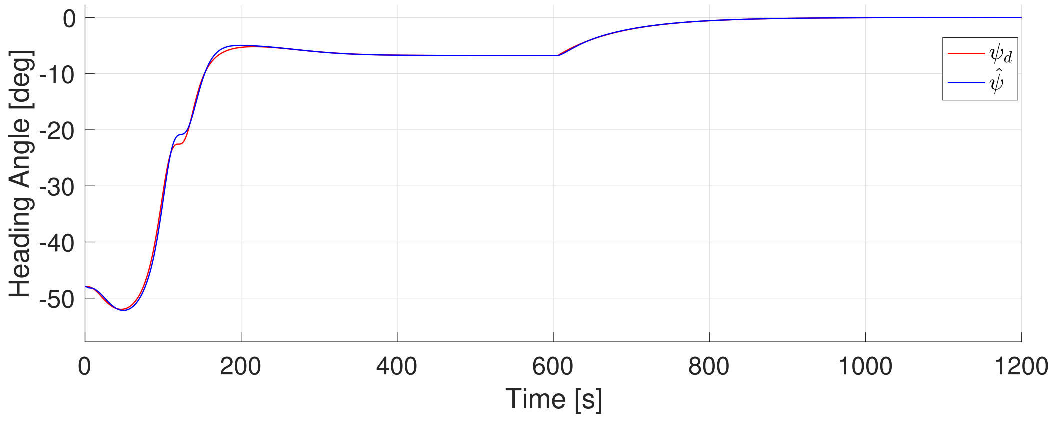


Figure 6.25: Simulation results from method 2 showing the desired heading and estimated heading.

It can be seen from the three figures above that the ship tracks the desired position and heading well for both phases.

The measurements from the proximity sensors is shown in figure 6.26, and are represented by the red and blue lines. The green line is the reference distance, i.e. the desired distance between the side of the ship and the quay. The figure displays that the ship moves slowly and controlled towards the harbor, while never touching it.

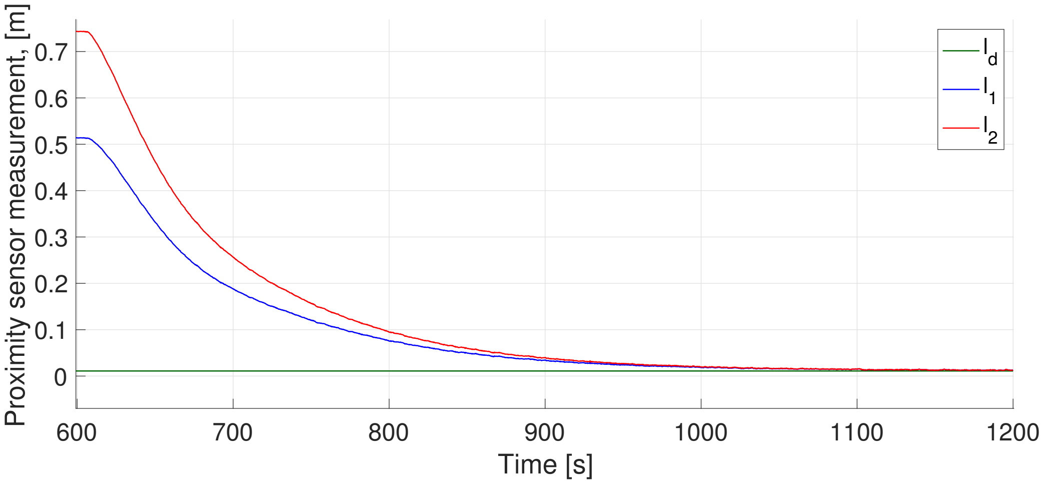
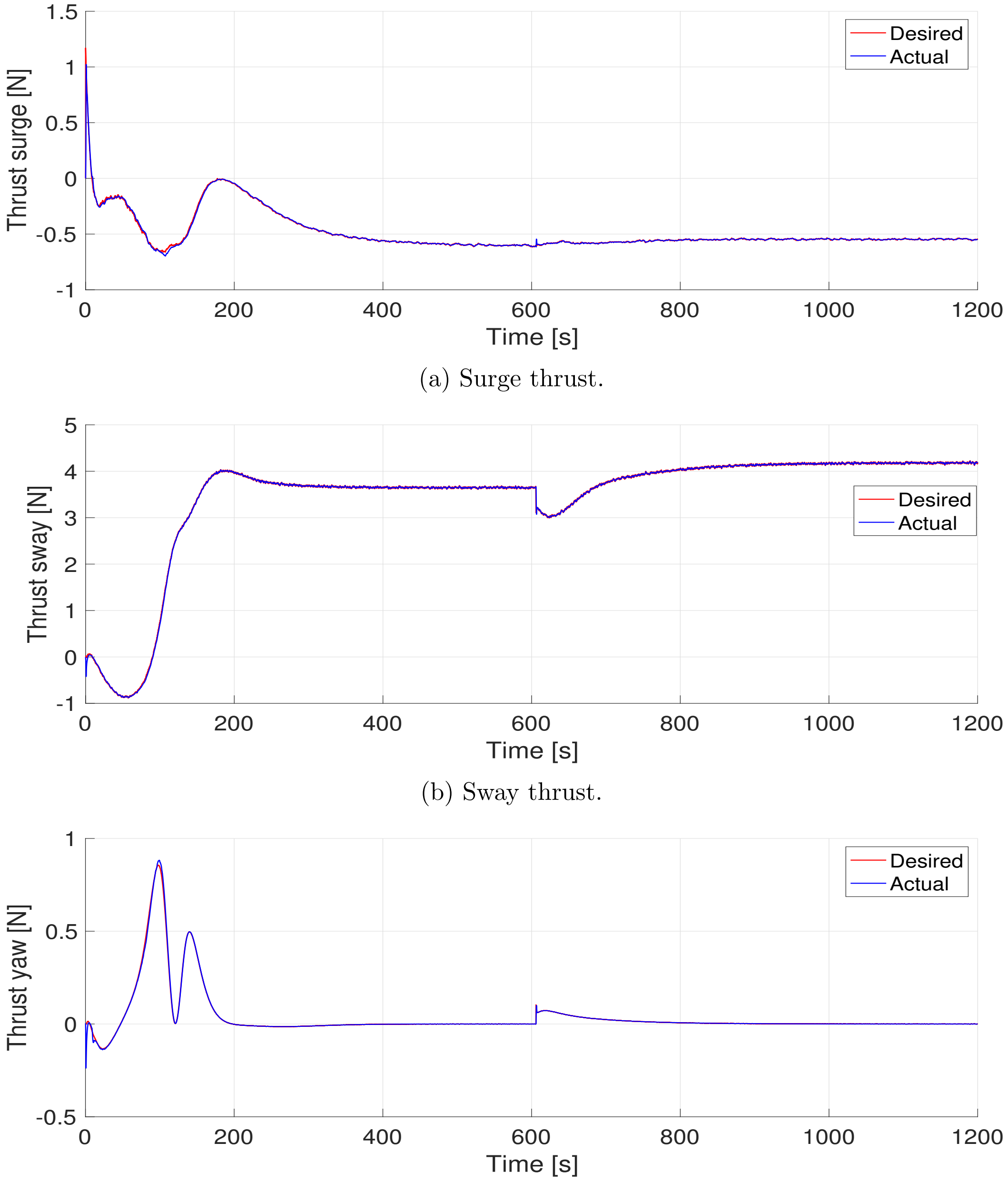


Figure 6.26: Simlation Results from method 2 showing the measurements from the two proximity sensors and the reference distance.

The commanded control forces from the controller and the actual applied forces after the thrust allocation algorithm, is also of interest. Figure 6.27 displays the desired forces from the controller against the forces applied after the thrust allocation for the surge, sway and yaw direction.



(a) Surge thrust.

(b) Sway thrust.

(c) Yaw thrust.

Figure 6.27: Results from method 2 showing commanded thrust from the controller and applied thrust from the thrust allocation.

The control forces behave smoothly, but a jump can be observed at the transition to phase 2. This is most prominent for the sway and yaw direction. This is expected as it occurs when the heading correction is initiated and the ship is to move towards the harbor.

6.3 Experiments

For the experiments, method 2 was used due to its superior results from the simulations. Why it is considered the superior method, is discussed in chapter 7. No current or waves, except from the very small waves the ship generate itself, are present. The reference distance between the ship and the dock was selected to be $l_d = 0.03$ meters as the proximity sensors are mounted approximately 1.6 centimeters centimeter from the side of the ship. This implies that the actual desired distance between the quay and the side of the ship will be 1.6 centimeters less than l_d, which in this case is 1.4 centimeters. The last waypoint has the coordinates $\boldsymbol{p}_n = [6.78, -1.75]$. Figure 6.28, 6.29 and 6.30 shows the estimated position and heading against the desired position and heading. The path following, phase 1, is initiated at approximately t = 80s. Before phase 1 is initiated the ship is told to go to $\boldsymbol{\eta}_d = [3, -1.4, -10°]$, where the docking operation is initiated from.

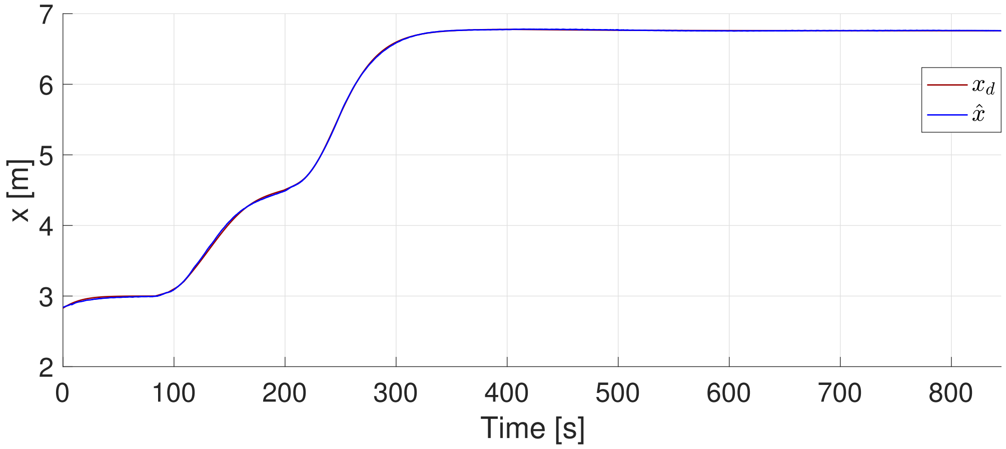


Figure 6.28: Estimated x-position and desired x-position for the experimental results.

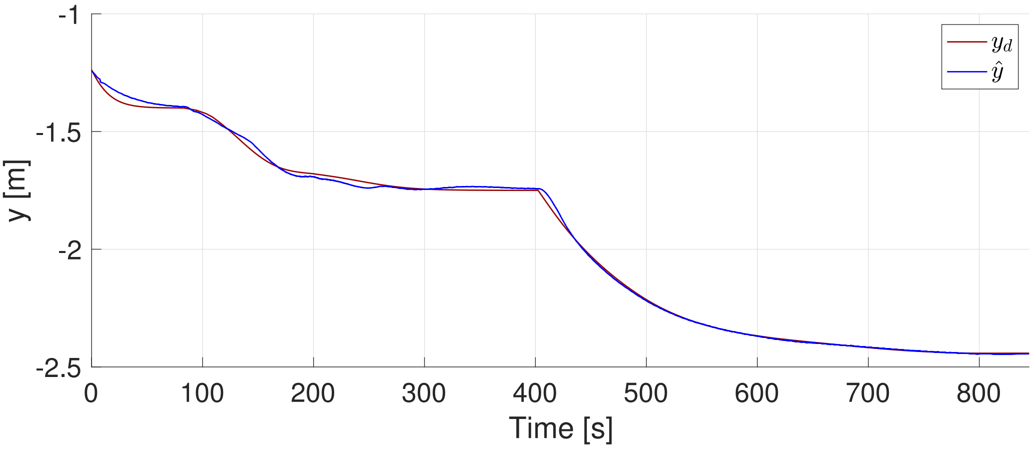


Figure 6.29: Estimated y-position and desired y-position for the experimental results.

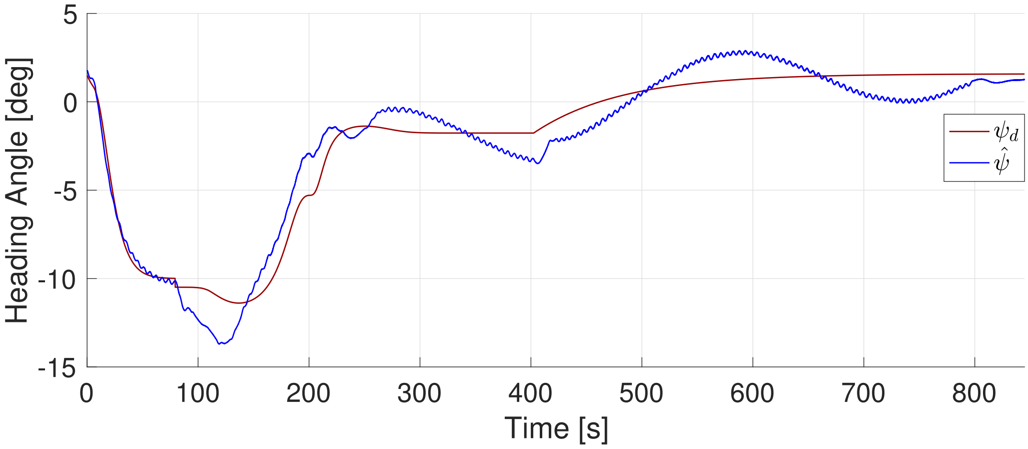


Figure 6.30: Estimated heading and desired heading for the experimental results.

The tracking of the position works well, but the control of the heading is not very good. Possible reasons for this is discussed in section 7.3.

The resulting xy-plot can be found in figure 6.31. Where the red lines are the harbor side, and the green lines are the edges of the tank. The exact location of the dock in the lab experiments is not known, so the red lines in the figure is only an estimation of its location. Nonetheless, it should be very close to the true location.

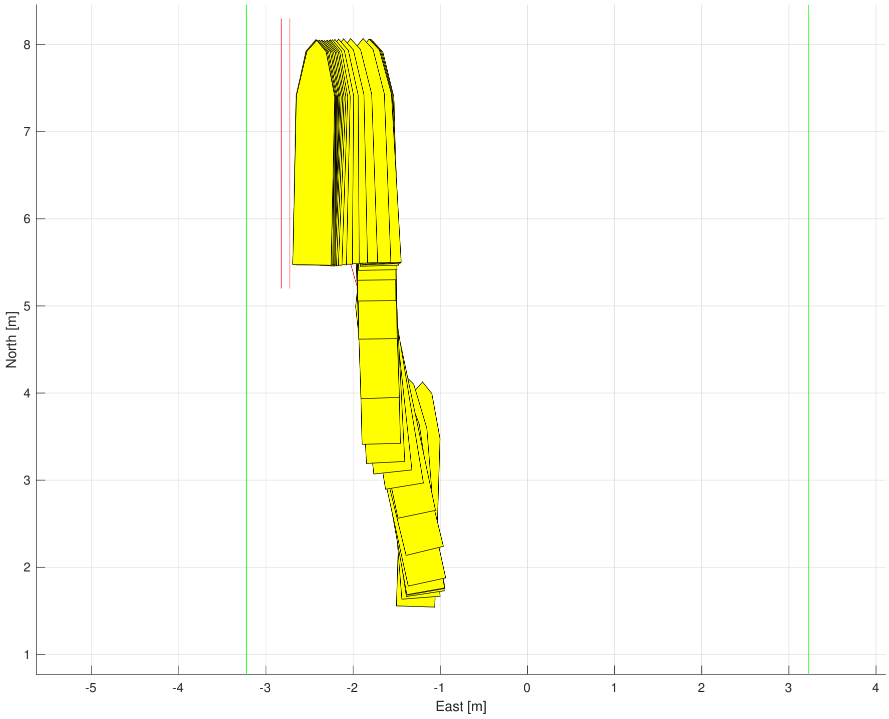


Figure 6.31: Resulting xy-plot from the experiments.

The signals from the proximity sensors are also of great interest. The low-pass filtered proximity sensor measurements together with the desired reference distance, is plotted in figure 6.32 below.

The green line is the desired distance between the ship and quay, and the red and blue lines are the low-pass filtered proximity sensor measurements.

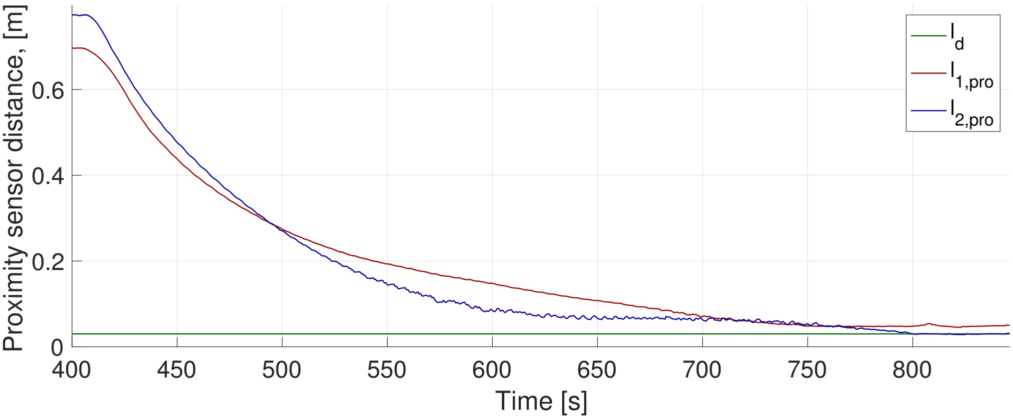


Figure 6.32: Proximity sensor measurements for the experimental results.

Proximity sensor 2 reads 0.03 meters, while proximity sensor 1 read 0.05 meters at the end of the run. Thus the heading correction has not worked properly. Nonetheless, the distance between the ship and the harbor is controlled towards the reference distance as sensor 2 reaches the desired distance of 0.03 meters.

The effect of the low-pass filtering of the proximity sensor measurements, is illustrated in figure 6.33. This is for sensor 2, which is the sensor closest to the aft of the model. The green line is the measured value, while the red line is the filtered signal.

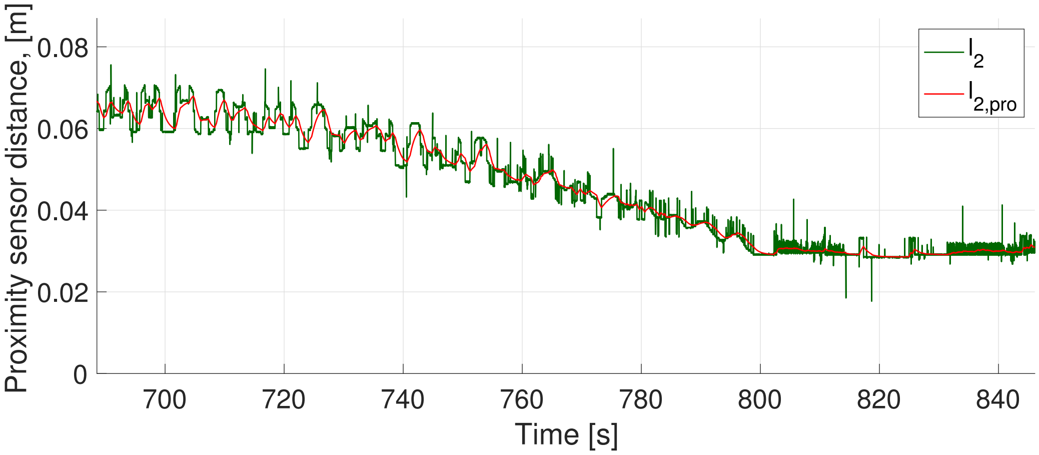


Figure 6.33: Low-pass filtering the proximity sensor measurements in the experiment.

A lot of the noise is filtered out. In addition, it may seem like the noise is reduced in magnitude as the distance measured gradually becomes smaller.

The control forces and the estimated applied forces after the thrust allocation is shown in figure 6.34, 6.35 and 6.36.

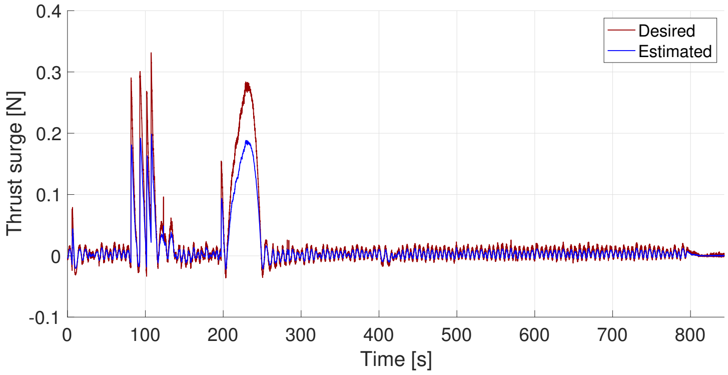


Figure 6.34: Control force and estimated applied force in surge for the experiment.

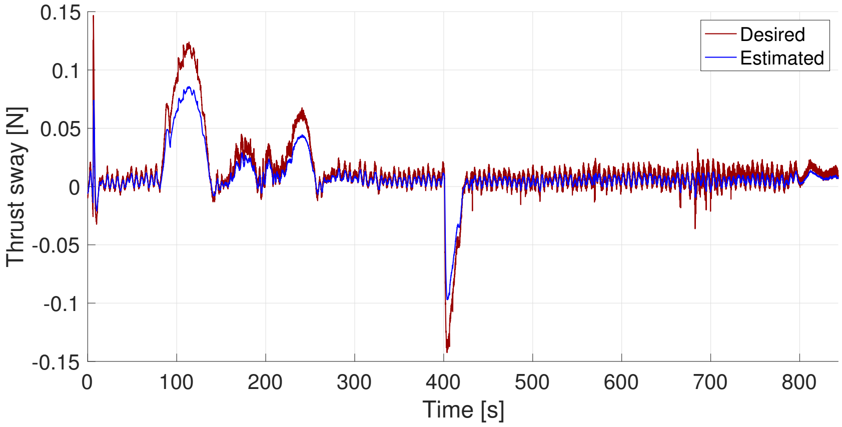


Figure 6.35: Control force and estimated applied force in sway for the experiment.

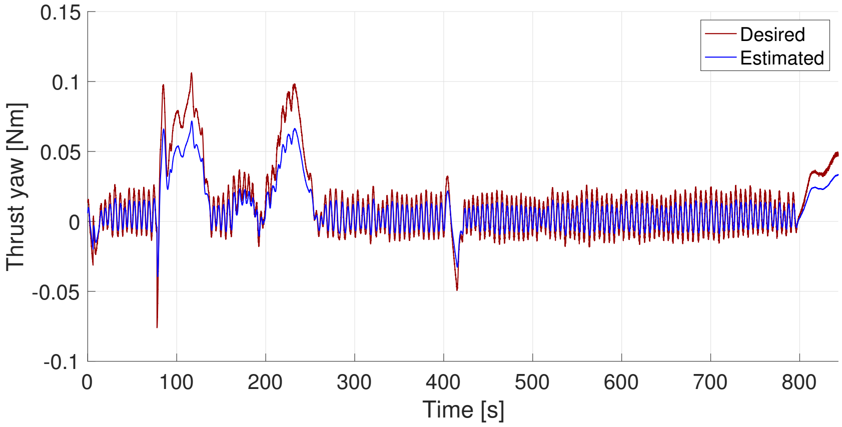


Figure 6.36: Control force and estimated applied force in yaw for the experiment.

6.3.1 Video of experiment

A QR-code and an URL leading to a video of the experiment can be found in appendix B. It should be noted that the video is from another run than presented in this chapter.

Chapter 7

Discussion

This chapter contains a discussion of the results. It is organized in the same way as the results chapter. Firstly the observer performance is discussed, followed by the simulations and experiments.

7.1 Observer

7.1.1 Simulations

The position and heading estimations performed by the nonlinear passive observer are working quite well. Most of the noise is filtered out, leaving mostly a smooth position and heading signal. The observer could potentially be tuned to filter out even more of the first order wave frequency motion. In full scale CSAD is a quite large vessel, and the actuators cannot move it fast enough to suppress the high frequency disturbances. On the other hand in model scale, the motors are electric and able to move the ship relatively fast and respond faster to changes in the control forces. Thus if less of the first order motion is filtered out and if the ship is very close to the side of the harbor and waves are coming in from the side, this can positively affect the control systems ability to counteract the first order wave frequency motion and prevent the ship from hitting the harbor.

The velocity estimation works well and filters out much of the high frequency disturbances, and is close to the mean of the true value of the velocities. The sway velocity seems to be estimated a little bit to high and the surge velocity a little bit to low, but this is very minimal.

7.1.2 Experiments

The observer used in the experiments, is tuned to follow the measurements too much. More high frequency disturbances should be filtered out. Nonetheless the observer was used, and yielded satisfactory results. The measurements from Qualisys usually contains a minimum amount of noise, so the under-filtering of the signals worked out fine.

The velocity estimations are smooth, but the yaw rate is very off. For example it is estimated to be close to 4 degrees per second after approximately 80 seconds. This is way to high. If we consider figure 6.9 for the same time, a more reasonable estimation would be close to 0.2 degrees per second as the heading changes from -15 degrees to -10 degrees in approximately 25 seconds. This happens from t = 75s to t = 100s, which is where the estimated velocity is estimated to be the highest.

The surge and sway velocity estimations seems to be better. The surge velocity is positive at almost all times and has its peak after approximately 160 seconds. This makes sense because at t $\approx$ 160 seconds the heading of the vessel is approximately zero meaning the surge velocity will be the same as the derivative of the position in the x-direction. From figure 6.7a it can be seen that the curve is the steepest at t = 160 seconds, which is also when the surge velocity has its peak. In addition, it is almost always positive which makes sense as the ship is always moving forward in the run. For the sway velocity the estimations seem to be a bit off. For example at the beginning, the sway velocity should be negative as the ship is moving in the negative y-direction with a heading not to far from zero. Still the velocity is estimated to be positive. Nonetheless the velocity estimations are smooth and without any high frequency disturbances which is desirable.

7.2 Simulations

The path following phase lasts from start until right after 600 seconds has passed. This phase is completely identical for method 1 and method 2, and the vessel tracks the desired position and heading well. This can be seen from figure 6.14, 6.15 and 6.16. The ship moves faster in the beginning and slows down as it approaches the last waypoint outside the harbor. When it reaches the last waypoint the velocity is approximately zero. Thus the control of the ship velocity works as intended.

For the docking phase, phase 2, the two methods differ. The results from section 6.2.1 shows that method 1 works reasonably well, and performs a successfully docking. Though it is apparent that the transition between the two phases, where there is a switch between controllers, is not optimal. The variation in the control forces between the two phases, but more importantly the small jump in the response confirms this. It is most apparent in figure 6.14, where it can be seen that the ship overshoots its desired x-position even though the change in x_d for the transition is very minimal. The reason for this is most likely the integral terms for the phase 2 controller being zero at the start of phase 2. Thus there is no current compensation. The problem becomes very clear when looking at the results for the stronger current presented in section 6.2.2, where the ship collides with the harbor. The main reason the results in this section look so bad, is the modeling of the proximity sensors. For the modeling of the proximity sensors the harbor side is defined as a line of infinite length, and the same goes for the proximity sensors which are modeled as straight lines of infinite length perpendicular to the ship side. The measurements will be equal to the length between the location of the proximity sensors on the ship, and where the lines of the proximity sensors cross the line defining the harbor. Thus the proximity sensor modeling always provides a measurement even though the proximity sensors does not actually hit the harbor defined by the red line in figure 6.13. This will give very wrong proximity sensor readings when things go bad, such as in the case displayed in section 6.2.2. Nonetheless the case is included to show the bad performance of method 1 for stronger currents. After the ship hits the harbor the modeling of the proximity sensors will be insufficient, and the results will not make much sense. Furthermore, there is nothing stopping the ship from passing straight through the harbor in the simulations as the harbor is only taken into account by the proximity sensors and there is no physically present harbor. This is also another cause for the bad proximity sensor measurements. In addition it should be mentioned that the modeling of the proximity sensors works as intended when the ship does not hit the harbor.

The results from method 2 is considerably better. The transition between the two phases is much more seamless and it has no problems dealing with the current method 1 struggles with. This can be observed in figure 6.23, 6.24 and 6.25. Furthermore the control forces and moments from figure 6.27 behaves the same for the entirety of the simulation. For phase 2 the ship approaches the dock slow and steady and stabilizes at the desired reference distance. This can be seen from figure 6.26. The heading correction works properly which can be seen from figure 6.25. The desired heading for the docking phase becomes zero degrees which is correct, as the harbor is parallel with the x-axis of the basin.

7.3 Experiments

The experiment was terminated a bit to early. The ship has arrived at its desired position, but the desired heading has not been achieved. Although achieving the desired heading involves the ship being even further away from being parallel to the dock as the heading correction did not work properly. Nonetheless the run was terminated to early, and the ship should have been given the time to stabilize at the desired position and heading.

It can be seen from figures 6.28 and 6.24 that the velocity of the ship gets lower and lower as it approaches the last waypoint just outside the harbor. Of course the figures show position but the velocity will be the derivative of the position, which makes it clear that the ship comes to rest at the last waypoint with approximately zero speed. When the docking phase is initiated after the path following, the ship is taken slow and steady towards the side of the dock.

Based on the proximity sensor readings at the end of figure 6.32 the error of the heading correction can be calculated to be $atan2(0.03 - 0.05, 1.29) = -0.88°$. This is the error compared to the actual heading of the ship, not the desired heading. If this is taken into account, the error of the heading correction is of approximately $0.88° + 0.3° = 1.18°$. This is not very much, but for a large ship such as CSAD it will have a greater effect. This is visible from figure 6.32 where the front of the ship is approximately 2 meters further away from the harbor side compared to the aft of the ship, at the locations of the proximity sensors. This is of course converted to full scale distances. The heading correction error is not huge, but there is room for improvement. For example a second heading correction can be applied when the ship is closer to the dock. For the experiment the heading correction happens when the ship is about 0.7 meters from the dock. From figure 6.32 it can be seen that the noise seems to reduce in amplitude as the measured distance between the sensor and object becomes smaller, thus the precision of the heading correction should increase if it happens closer to the dock. A possibility is to carry out heading corrections as long as the difference between the two proximity sensor measurements is greater than a specific value. Thus, when the difference drops below this specific value, the ship should be sufficiently parallel to the dock and no more heading corrections will be conducted. Another possible solution is to use the heading controller from method 1, as it works directly on the the proximity sensor measurements and has the control objective of making $l_1 = l_2$.

There are several possible reasons for the heading correction not to work as good as it can. Experimental uncertainties such as the proximity sensors not pointing exactly horizontal or perpendicular to the ship side, will negatively affect the precision of the heading correction. Furthermore, this effect will be greater and greater the further away from the dock the sensors are. Thus, making sure the proximity sensors are attached properly and pointing in the right direction is essential. Switching the duct tape out with a more solid and stable attachment method is a possibility. In addition, noise can also cause a bad heading correction. Better filtering of the proximity sensor measurements can help prevent this. Although, too much filtering action is not desired due to phase delay. This involves filtering away too much of the original proximity sensor measurements.

The filtering of the proximity sensor measurements seems to work reasonably well, which can be seen from figure 6.32. The high-frequent oscillations and fluctuations in the signal will be both due to noise and the motion of the vessel. Most of the noise and wild-points are filtered out, while not filtering out the signal component caused by the ship motion. Though the heading correction indicates that it can be done better.

The control of the position works well, both for the path following and the docking phase. On the other hand, the control of the heading does not work as desired. Firstly the heading has high frequent oscillations with amplitude of approximately a quarter of a degree. Watching the experiment live, this is not visible to the naked eye. What is more critical is the larger lower frequency oscillations. The most probable reason for this is the faulty yaw rate estimates, as discussed in 7.1.2. The observer estimates the yaw rate to be substantially larger than it should be in reality. This will cause a too large derivative action of the controller, which will work against achieving the desired heading. The integral action of the controller has to build up to try and compensate for the derivative action. Thus, when the heading eventually enters the other side of the desired heading, the integral action has been built up and it takes time to for it to work in the other direction again. This can be the cause of the lower frequency oscillations visible in figure 6.30. Another possible reason for the problem can be the controller being badly tuned.

Regarding the control forces, they are considerably lower than the control forces in the simulations. The most significant reason is the absence of current and waves in the experiments. In addition the actual vessel dynamics will differ from the vessel dynamics in the simulations as it is based on a mathematical model which will not be able to precisely capture the real vessel dynamics. The control forces do oscillate quite a bit, but the amplitude is low. The oscillations are natural when the position of the ship is close to the desired position, and the sign of the error will alternate between being positive and negative. The oscillations could possibly be reduced by better tuning of the observer.

Chapter 8

Conclusion

The focus of this thesis has been on finding a method for autonomous docking for ships using proximity sensors. Two different methods for autonomous docking is proposed. Both of them are tested in simulations, which made it clear that method 2 was the superior method. The transition between the two phases for a stronger current present, did not work well for method 1. For the same current velocity, method 2 performs very well and has no problems bringing the ship slow and steady to the desired position and heading right next to the harbor. Furthermore the path following phase, phase 1, works as desired. The vessel moves faster in the beginning and slows down as it approaches the last waypoint outside the harbor. When it reaches the last waypoint the velocity is approximately zero. Thus, a solution to the objective of finding suitable velocity references for a ship which is to dock have been found.

The integration of proximity sensors into the control system has also been of great importance in this thesis. The proximity sensor measurements have been integrated into the guidance system for method 2, which yielded good results. In addition, for method 1 it was attempted to use the proximity sensor measurements directly in the controller for heading and sway motion control. It turned out that this method did not work properly. Method 1 has some merit to it, but further development of the method and some other solutions must be found to make it viable.

Two acoustic proximity sensors has been installed on C/S Inocean Arctic Drillship. In addition a dock has been constructed using wooden planks, plywood plates and a buyancy element made of a plastic material. This was mostly to make it possible to perform docking experiments using proximity sensors, but it also facilitates for further research regarding docking of ships in the MC-lab.

Method 2 was also tested in the MC-lab, with promising results. The vessel followed the desired position both for the path following phase and the docking phase. The control of the heading of the vessel could have been better. The most probable causes for this is a bad yaw rate estimate and a controller which can be better tuned. Furthermore, there is room for improvement concerning the heading correction for the docking phase and some potential solutions are proposed and discussed.

Chapter 9

Further Work

This chapter provides some potential continuations to the work done in this thesis.

The observer could be updated to include proximity sensor readings for the docking phase. When the last waypoint of the path has been reached and the docking phase is initiated, the proximity sensor measurements alone should be enough to estimate the ships position and heading. Thus, if loss of position and heading measurements occur, the proximity sensors can be used for the estimations.

For big ships with a lot of surface area above the waterline, the forces from the wind can be substantial. Thus, including wind in the simulations is of interest. For the controller the possibility of adding wind feedforward action should be looked at.

For real scale vessels, the proximity sensors used in the experiments would not work. The maximum range of the sensors are 4 meters and during the experiments it was discovered that when the sound waves had to travel over one meter before it hit an object over the water, the measurements became very noisy and impossible to read. The reason for this could be some of the sound waves being reflected of the water surface and back towards the sensor, resulting in incorrect measurements. Thus replacing the proximity sensors with a lidar or radar can be looked at. These sensor use light waves instead of sound waves, they have a higher range and are not that vulnerable to the ambient conditions at sea.

The path planning part of the problem has not been studied a lot in this thesis. Further study of the generation of good entrance paths should be conducted.

For the experiments the most natural continuation is to implement the proposed solutions from section 7.3 for improving the heading correction. The most natural thing to do is to try the method of performing more heading corrections as the ship gets close to the dock. If this does not work, it is possible to try and implement the heading controller from method 1.

Hydrodynamic effects of using thrusters and rudders near harbor structures should also be studied further. It is not clear how hydrodynamic effects such as backwash can be implemented in the simulations, but for the experiments the plywood plates can be extended to go deeper down into the water. This will increase the surface area of the dock, and the backwash effect appearing from the thrusters pushing water into the side of the dock will possibly increase.

Bibliography

Eric Murdoch, Ian W. Dand, and Chris Clarke. A masters guide to: Berthing. February 2012.

Preben Frederich. Constrained optimal thrust allocation for c/s inocean cat i drillship. Master's thesis, July 2016.

Cytron-Technologies. Product user's manual – hc-sr04 ultrasonic sensor, 2013.

Tore Stensvold. Nå kan folgefonn seile automatisk fra havn til havn og legge til selv. November 2018a.

Tore Stensvold. Yara birkeland skal bygges i norge. August 2018b.

Edvard Meyer Flaatten. Instrumentation and methods for auto-docking of autonomous ships. Master's thesis, December 2018. Project thesis.

Joachim Spange. Autonomous docking for marine vessels using a lidar and proximity sensors. Master's thesis, 2016.

Fredrik Dessen. Optimizing order to minimize low-pass filter lag. March 2018.

NTNU. Marine cybernetics teaching laboratory, 2019. Retrieved the 30th of April 2019 from `https://www.ntnu.edu/imt/lab/cybernetics`.

NTNU. Cybership arctic drillship user manual, 2017. Retrieved from `https://github.com/NTNU-MCS/MC_Lab_Handbook/blob/master/Version%202.0/CSAD%20Handbook/CSAD.pdf`.

Jon Bjørnø. Thruster-assisted position mooring of c/s inocean cat i drillship. Master's thesis, July 2016.

Guttorm Udjus. Force field identification and positioning control of an autonomous vessel using inertial measurement units. Master's thesis, June 2017.

Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons, Ltd, 2011.

Cuspsensors. Ultrasonic sensors parallax ping vs hc-sr04, 2019. Retrieved from `https://cuspsensors.files.wordpress.com/2015/05/ultrasonic_sensor_comparison.pdf` 29. april 2019.

Tristan Perez and Thor I. Fossen. Kinematic models for manoeuvring and seakeeping of marine vessels. *Modeling, Identification and Control*, 28, 2007.

Dag Myrhaug and W. Lian. *Marine Dynamics Lecture Notes*. 2009.

Roger Skjetne. *The Maneuvering Problem*. PhD thesis, Trondheim, Norway, 2005.

Roger Skjetne. Marine control systems ii lecture 7: Maneuvering control design, 2019. Department of Marine Technology Norwegian University of Science and Technology.

Ole Nikolai Lyngstadaas. Ship motion control concepts considering actuator constraints. Master's thesis, 2018.

Appendices

Appendix A

Matlab code

A.1 Proximity Sensors

```
function [l1,l2] = proximity_sensors(harbor, eta)
%% Calculates the distance from each of the two proximity sensors to the harbor side
H_vertical = false;
H_horizontal = false;

ps1_vertical = false;
ps1_horizontal = false;

ps2_vertical = false;
ps2_horizontal = false;

%% Pose of the ship
x = eta(1);
y = eta(2);
psi = eta(3);

%% Ship characteristics
L = 2.578;
W = 0.440;

%% Calculation of the position of the proximity sensors in the tank fixed reference frame
tmpR=[cos(psi) -sin(psi); sin(psi) cos(psi)];
boat = tmpR*[L/2 .9*L/2 .5*L/2 -L/2 -L/2 .5*L/2 .9*L/2 L/2;
               0    W/4    W/2  W/2 -W/2   -W/2   -W/4   0];

ps1 = [y + boat(2,6), x + boat(1,6)];
ps2 = [y + boat(2,5), x + boat(1,5)];

%% Harbor side line equation
if harbor(1,1) == harbor(1,2) % Harbor is vertical in NE coordinates
    % line given by E = constant
    a = 0;
    b = harbor(1,1);
    H_vertical = true;
elseif harbor(2,1) == harbor(2,2) % Harbor horisontal
    % line given by N = constant
    a = harbor(2,1);
    b = 0;
    H_horizontal = true;
else
    % Line given by N = aE + b
    a = (harbor(2,2) - harbor(2,1))/(harbor(1,2) - harbor(1,1));
    b = -a*harbor(1,1) + harbor(2,1);
end

H = [a b];

%% Proximity sensor line equations
% Proximity sensor 1
point1 = [y + boat(2,3), x+boat(1,3)];

if point1(2) == ps1(2) % Proximity sensor points horizontal
    ps1_a = 0;
    ps1_b = ps1(2);
    ps1_horizontal = true;
elseif point1(1) == ps1(1) % Proximity sensor points vertical
    ps1_a = (ps1(1));
    ps1_b = 0;
    ps1_vertical = true;
else
    ps1_a = (point1(2) - ps1(2))/(point1(1) - ps1(1));
    ps1_b = -ps1_a*ps1(1) + ps1(2);
end

ps1_eq = [ps1_a, ps1_b];

% Proximity sensor 2
point2 = [y + boat(2,4), x + boat(1,4)];

if point2(2) == ps2(2) % Proximity sensor points horizontal
    ps2_a = 0;
    ps2_b = ps2(2);
    ps2_horizontal = true;
elseif point2(1) == ps2(1) % Proximity sensor points vertical
    ps2_a = ps2(1);
    ps2_b = 0;
    ps2_vertical = true;
else
    ps2_a = (point2(2) - ps2(2))/(point2(1) - ps2(1));
    ps2_b = -ps2_a*ps2(1) + ps2(2);
end

ps2_eq = [ps2_a, ps2_b];
```

```

%% Intersection of harbor line and sensor lines
% Calculation of l1 and l2, proximity sensor distances
% Sensor 1
if ps1_vertical && H_vertical
    l1 = -1;
elseif ps1_horizontal && H_horizontal
    l1 = -1;
elseif ps1_horizontal && H_vertical
    l1 = abs(H(2) - ps1(1));
elseif ps1_vertical && H_horizontal
    l1 = abs(H(1) - ps1(2));
elseif H_vertical
    intersection_E1 = H(2);
    intersection_N1 = ps1_eq(1)*intersection_E1 + ps1_eq(2);
    l1 = sqrt((ps1(1) - intersection_E1)^2 + (ps1(2) -
        intersection_N1)^2);
else
    intersection_E1 = (ps1_b - b)/(a - ps1_a);
    intersection_N1 = a*intersection_E1 + b;
    l1 = sqrt((ps1(1) - intersection_E1)^2 + (ps1(2) -
        intersection_N1)^2);
end

% Sensor 2
if ps2_vertical && H_vertical
    l2 = -1;
elseif ps2_horizontal && H_horizontal
    l2 = -1;
elseif ps2_horizontal && H_vertical
    l2 = abs(H(2) - ps2(1));
elseif ps2_vertical && H_horizontal
    l2 = abs(H(1) - ps2(2));
elseif H_vertical
    intersection_E2 = H(2);
    intersection_N2 = ps2_eq(1)*intersection_E2 + ps2_eq(2);
    l2 = sqrt((ps2(1) - intersection_E2)^2 + (ps2(2) -
        intersection_N2)^2);
else
    intersection_E2 = (ps2_b - b)/(a - ps2_a);
    intersection_N2 = a*intersection_E2 + b;
    l2 = sqrt((ps2(1) - intersection_E2)^2 + (ps2(2) -
        intersection_N2)^2);
end
```

Appendix B

Video of experiment

Below follows a QR code leading to a video of the experiment. The video shows the experiment in 4x speed, and from two different angles.

Figure B.1: QR-code for video of experiment

The url-code for the same video is:

`https://vimeo.com/344195500`