Emilie Thunes

Online risk modeling for autonomous ships with experimental results

June 2019

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

Master's thesis

2019

Emilie Thunes

Master's thesis

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Online risk modeling for autonomous ships with experimental results

Emilie Thunes

Marine Technology
Submission date: June 2019
Supervisor: Asgeir J. Sørensen
Co-supervisor: Ingrid Bouwer Utne,
Børge Rokseth

Norwegian University of Science and Technology
Department of Marine Technology

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

MASTER THESIS IN MARINE CYBERNETICS

Spring 2019

FOR

STUD. TECHN. Emilie Thunes

Online risk modeling for autonomous ships with experimental results

Work description (short description)

In the last decade the number of crew in oceangoing cargo vessels have decreased, as well as the development of optical sensors have seen a great progress. The development of optical sensor technology and computer power have allowed the level of automation on vessel navigation and control to improve drastically. Where the main motivation for increased autonomy level in deep-sea cargo are associated with reduced operational cost and increased safety. The work in this thesis is divided into two parts. The objective of the first part is to demonstrate how hardware-in-the-loop simulations can assist the control design development process for the ship model CS Enterprise I. Backstepping and linear PID control design will be investigated. The objective of the second part is to develop an online risk model with focus of technical risk influencing factors during DP operation. Furthermore, a case study will be performed on the ship model CS Enterprise I where the ORA will be tested. The second part is developed in collaboration with Ina Bjørkum Arneson.

Scope of work

- Develop a control system for CS Enterprise I with DP manoeuvring capabilities.
- Demonstrate relevant cases of the proposed control system for the model ship with hardware-in-the-loop simulation and model scale testing in the Marine Cybernetics laboratory.
- Identify relevant risk factors associated with an autonomous surface ship operation and develop a Bayesian belief network for an DP operation.
- Propose an online risk model applicable for DP operation including collision, loss of navigation sensors and thrusters and propulsion capabilities.
- Propose a strategy on how to integrate ORA with the DP and tracking control system for autonomous operations with planning and replanning.
- Simulate different scenarios in Simulink and hardware-in-the-loop.
- Demonstrate relevant cases of the online risk analysis system on the ship model CS Enterprise I, consider a DP operation, and evaluate strengths and limitations in the ORA.

The report shall be written in English and edited as a research report including literature survey, description of mathematical models, description of control algorithms, simulation results, model test results, discussion and a conclusion including a proposal for further work. Source code should be provided. It is supposed that Department of Marine Technology, NTNU, can use the results freely in its research work, unless otherwise agreed upon, by referring to the student's work.

The master thesis should be submitted within June 11, 2019.

Co-supervisor: Prof. Ingrid B. Utne, Phd Børge Rokseth

Professor Asgeir J. Sørensen
Supervisor

Abstract

The maritime industry is today experiencing considerably progress in the fields of autonomy, which have allowed the level of automation on vessel navigation and situation awareness to improve drastically. Autonomous vessels are a relatively new concept, and thus challenging the way vessels are designed, tested as well as introducing new challenges related to verification and safety. The development of safe and robust control systems is a significant step towards highly autonomous ships. This thesis consists of two parts. Part I is covering hardware-in-the-loop (HIL) simulations of a linear and nonlinear control system, with the purpose to increase the safety and fault tolerance in the control design. The control systems developed are dynamical positioning, DP, maneuvering with backstepping and linear conventional DP with PID. The nonlinear control design is considerably more advanced compared to linear PID. Thus, a general discussion regarding linear and nonlinear control systems was presented. The proposed control systems are validated with simulation in Simulink, hardware-in-the-loop, and model scale testing in the Marine Cybernetics laboratory at NTNU. The results revealed successful DP maneuvering and linear DP control design.

Part II of the thesis is covering the development of an online risk model. For highly autonomous surface vessels to be operative, a risk model making intelligent decisions must be implemented into the control system of the ship. The proposed online risk model is based on the risk of collision for a continuously autonomous ship during a DP operation. A Bayesian belief network was constructed based on risk influencing factors regarding situation awareness, environmental conditions, and power loss. Three different scenarios concerning the loss of situation awareness, extreme weather, inadequate power and maintenance have been defined and tested. For each event, the Bayesian belief network calculated the increased or decreased risk of collision. Following, a decision model was designed to intervene with the mitigating measures if the belief in high risk of the collision, had exceeded the predefined threshold value. The proposed model was tested with simulation in Simulink, HIL, and model scale experiments. Based on the objective of the online risk model, it was concluded that the model successfully makes a decision based on the current event. Furthermore, the applicability and limitations of the online risk model are discussed and a proposal on further work are suggested.

Sammendrag

Den maritime industrien opplever i dag betydelige fremskritt i sammenheng med autonomi, noe som gjør det mulig å forbedre automatiseringsnivået rundt navigasjon og situasjonsforståelse. Autonome fartøy er et relativt nytt konsept som utfordrer måten skip er designet og testet, samt innfører nye utfordringer knyttet til verifisering og sikkerhet. Utvikling av skire og robuste kontrollsystem er et betydelig viktig skritt i prosessen mot høyautonome skip. Denne masteroppgaven består av to deler. Del I dekker hardware-in-the-loop (HIL) simulering av et lineært og ulineært kontrollsystem, der formålet er øke sikkerheten og feiltoleranse i kontrolldesignet. Kontrollsystem som er utviklet er dynamisk posisjonering, DP, manøvrering med backstepping og en konvensjonell lineær PID regulator. Ulineære kontrolldesign er betydelig mer avansert sammenlignet med lineær PID. Det er dermed gitt en generell diskusjon angående ulineær og lineære kontrollsystem. De sistnevnte kontrollsystemene er testet ved simulering i Simulink, HIL, og modellskala i Marin Kybernetikk laboratoriet ved NTNU. Resultatene indikerte vellykket DP manøvrering og lineær DP kontrolldesign.

Del II av oppgaven dekker utviklingen av en online risikomodell. For at høyautonome skip skal være operative, må det implementeres en risikomodell i kontrollsystemet som kan ta intelligente beslutninger. Den foreslåtte risikomodellen er basert på risiko for kollisjon for et kontinuerlig autonomt skip, under en DP-operasjon. Et Bayesiansk nettverk ble konstruert basert på risiko påvirkende faktorer med hensyn til situasjonsforståelse, miljøforhold og strømforsyning. Tre forskjellige scenarier omhandlende tap av situasjonsbevissthet, ekstremvær, utilstrekkelig strøm og vedlikehold ble definert og testet. For hver hendelse kalkulerte det Bayesianske nettverket den økte eller reduserte risikoen for kollisjon. Deretter ble en beslutningsmodell utformet for å utføre risiko reduserende tiltak dersom troen på høy risiko for kollisjon hadde overskredet den forhåndsdefinerte terskelverdien. Modellen var deretter testet ved simulering i Simulink, HIL og ved modellskala eksperimenter. Basert på resultatene, ble det konkludert at modellen klarte å ta riktige avgjørelser avhengig sitasjonen. Avslutningsvis ble en diskusjon basert på anvendeligheten og begrensninger ved modellen utført.

Preface

This master's thesis's is a part of the Master of Science in Marine Technology at the Norwegian University of Science and Technology (NTNU) with a specialization in marine cybernetics. Case study II in the thesis is performed in collaboration with Ina Bjørkum Arneson, and the thesis is written during the spring of 2019.

Writing this thesis has been extremely interesting and challenging. My desire to understand more about the future of autonomous ships, where much explored. I have learned a lot about nonlinear control systems, hardware-in-the-loop and especially how the industry thrives towards safer and more cost-efficient ships.

I would like to extend my biggest gratitude to my supervisor, Professor Asgeir J. Sørensen, for the inspirational consultations that helped me shape the theme and outline of the thesis. A big thanks to Ina Bjørkum Arneson for good discussions and excellent teamwork. Thanks to co-supervisors Professor Ingrid Bouwer Utne and PhD Børge Rokseth. I would also like to thank PhD candidate Einar Skiftestad Ueland for guidance at the Marine Cybernetics laboratory.

Thanks to Grieg Star for letting me join a vessel passage from Vlissingen to Santander, which gave me a better understanding of navigation on conventional cargo vessels. Lastly, I would like to thank my fellow students, friends and especially my family, for inspiration and support.

Nomenclature

Acronyms

AIS Automatic Identification System

AMOS Centre for Autonomous Marine Operations and Systems

ASV Autonomous Surface Vehicles

AUV Autonomous Underwater Vehicle

BBN Bayesian belief network

BT Bow thruster

CLF Control Lyapunov function

CPT Conditional probability table

CUS Continuously unmanned ship

DNT Dynamic Navigation Task

DOF Degrees of freedom

DP Dynamic positioning

GAS Globally automatically stable

GES Globally exponentially stable

GNSS Global Navigation Satellite System

HIL Hardware-in-the-loop

HMI human-machine interface

IMO International Maritime Organisation

IUMI International Union of Marine Insurance

JONSWAP Joint North Sea Wave Project

MASS Maritime Autonomous Surface Ship

MIMO Multiple-input and multiple-output

MUNIN Maritime Unmanned Navigation through Intelligence in Networks

NFAS Norwegian forum for autonomous ships

ODD Operation Design Domain

OOW Officer on watch

ORA Online risk analysis

PID Proportional–integral–derivative controller

PUB Periodically unmanned bridge

PUS Periodically unmanned ship

$RADAR$ RAdio Detection And Ranging

ROV Remotely operated vehicle

RPM revolutions per minute

SA Situation awarness

SCC Shore Control Centre

$SINTEF$

$SISO$ Single input single output

$UGAS$ Uniformly globally automatically stable

USV Under water surface vehicle

VSP Voith schneider propellers

B0 Conditionally and periodically unmanned bridge

SOLAS International Convention for the Safety of Live at Sea

List of Symbols

$\boldsymbol{\eta}$ NED positions $[x, y, \psi]$

$\boldsymbol{\nu}$ BODY linear velocities $[u, v, r]$

$\boldsymbol{\nu}_c$ BODY current velocities

$\boldsymbol{\nu}_r$ BODY relative velocities

$\boldsymbol{\tau}$ Control input

$\boldsymbol{C_A}$ Coriolis-centripetal matrix added mass

$\boldsymbol{C_{RB}}$ Coriolis-centripetal matrix ridgid body

$\boldsymbol{C}$ Coriolis-centripetal matrix

$\boldsymbol{D}$ Damping matrix

$\boldsymbol{G}$ Restoring matrix

$\boldsymbol{K}$ Gain matrix

$\boldsymbol{L}$ Injection gains observer

$\boldsymbol{M}$	System inertia matrix
$\boldsymbol{u}$	Control force to the actuators
$\boldsymbol{w}$	Vector of environmental disturbances
δ	Thruster angle
μ	Fluid memory
ϕ	Euler angle, roll
ψ	Euler angle, yaw
θ	Euler angle, pitch
b	bias
$\mathbf{g}(\boldsymbol{\eta})$	Restoring forces
$\mathbf{R}_b^n(\boldsymbol{\Theta})$	Rotation matrix about xyz from frame **b** to frame **n**, rotated by $\boldsymbol{\Theta}$

Contents

List of figures

Tables

Chapter 1

Introduction

1.1 Motivation

Automation is the process, often computerized, that implement a specific and predefined method to execute certain operations without a human controlling it [5]. In recent years automation has been a well-used word in context with a wide range of industries. Several industries have started to implement autonomy on different levels, where mainly the maritime industry is experiencing substantial changes and opportunities. The potential of autonomy is tremendous today, due to considerable progress in the field of sensors and computer power. Companies like Tesla have done a significant development of autonomous cars, and Kongsberg Gruppen, which is developing the world's first fully autonomous, electrical and zero emission container ship, Yara Birkeland. For ship owners, increasing fuel costs is one of the most considerable operational costs. With highly autonomous ships it is possible to increase voyage days, without the extra crew costs. From the MUNIN project, it was stated that for a typical medium-sized bulk carrier it is possible to save approximately 50% fuel when reducing the speed of the vessel with 30%, even when counting the extra voyage days [13].

In the last decade, the number of crew in oceangoing cargo vessels has decreased, as well as the development of optical sensors, have seen significant progress. The development of optical sensor technology and computer power has allowed the level of automation on vessel navigation and situation awareness to improve, which contributes towards autonomous surface vehicles (ASVs). ASVs are becoming more and more interesting for the shipping industry, due to potentially higher cost-efficient trade routes as well as potentially safer operations. The crew working in the maritime industry are frequently exposed to tasks in demanding environments, which might induce dangerous situations. It is estimated that 75 % - 96 % of all marine accidents are due to human error[5]. With increased autonomy level for the vessels, errors introduced by fatigue, lack of knowledge and organizational failures can be reduced.

Autonomous vessels are a relatively new concept, and thus challenging the way vessels are designed, tested as well as introducing new challenges related to verification and safety. For highly autonomous surface vessels to be operative, a risk model making intelligent decisions should be implemented into the control system of the ship. For this to be possible, an detailed understanding of the vessel's surroundings should be provided with accurate sensor data. It is inconceivable that new risk factors related to software erroneous, sensor failures, and cybersecurity will emerge in line with the increasingly intelligent technology onboard the ship. This suggests that advanced testing with hardware-in-the-loop should be performed in order to increase the safety and fault tolerance in the embedded systems. The objective of this thesis is to gain more knowledge of hardware-in-the-loop simulation of nonlinear control systems, before the development of an

online risk model and verification with experimental results will be presented. The following Section describes the scientific question.

1.2 Problem formulation and research questions

The aim of the thesis are reduced to the two following research question:s

1. *How can hardware-in-the-loop simulations assist the control design process, in specific the backstepping control design?*
2. *Is it possible to develop an online risk model based on risk of collision, in order to increase the situation awareness of a continuously unmanned ship during a DP operation?*

The work in this thesis is divided into two parts. The objective of the first part is to demonstrate how hardware-in-the-loop simulations can assist the control design development process. Backstepping and linear PID control design will be investigated. The objective of the second part is to develop and demonstrate an online risk model with focus of technical risk influencing factors during DP operation. The second part is developed in collaboration with Ina Bjørkum Arneson.

The following bullet points will be considered:

- Develop a control system for CyberShip Enterprise I with DP maneuvering capabilities.
- Demonstrate relevant cases of the proposed control system for the model ship with hardware-in-the-loop simulation and model scale testing in the Marine Cybernetics laboratory.
- Identify relevant risk factors associated with a autonomous surface ship operation and develop a Bayesian belief network.
- Propose an online risk model applicable of dynamic positioning, including collision, loss of navigation sensors and thrusters and propulsion capabilities.
- Demonstrate relevant cases of the online risk model with hardware-in-the-loop simulation and on the ship model CyberShip Enterprise I.
- Evaluate strengths and limitations in the online risk analysis.

1.3 Contributions

The main contributions of this thesis are divided into the two parts. A general discussion of positive outcomes and challenges regarding case study I and II are included. Hence, the main contributions are:

- Background review regarding situation awareness, autonomy taxonomy and risk aspects associated with autonomous surface ships.
- A brief review of Bayesian belief networks and control theory.
- Case study II: Development of backstepping and linear PID control design, including hardware-in-the-loop simulation and experimental results.
- Case study II: Online risk model for DP operation with a continuously autonomous ship, including hardware-in-the-loop simulation and experimental results.

1.4 Outline of the thesis

The thesis is divided into six chapters. This section gives a brief introduction to each Chapter. All necessary code are attached.

Chapter 1 - Introduction

Chapter 1 consist of motivation, problem formulation and research question, contributions and outline of the thesis.

Chapter 2 - Background and preliminaries

Chapter 2 consist of six main parts: Situation awareness, levels of autonomy, risk aspects associated with autonomous surface ships, online risk analysis, hardware-in-the-loop simulations and control theory. In the first Section, situation awareness, an introduction of situation awareness and relevant sensors are given. The second Section gives an introduction to different levels of autonomy defined by NTNU AMOS and NFAS. Following, an brief discussion regarding risk aspects associated with autonomous surface ships is given. In the fourth section an overview of the motivation for online risk modelling is given, followed by an introduction to Bayesian belief networks and fuzzy logic. Lastly the concept of hardware-in-the-loop simulation and introduction to relevant control theory is reviewed. From chapter 2, situation awareness and levels of autonomous surface ships are retrieved from the author's project thesis.

Chapter 3 - Case study I: Control system, CyberShip Enterprise I

Chapter 3 introduce Case study I: Control system, CyberShip Enterprise I. The case study contains a description of the process plant model and control plant model. Secondly, thrust allocation and observer design for CyberShip Enterprise I is developed. Following, backstepping and linear PID controller design are developed, tested with simulation in Simulink, hardware-in-the-loop and model scale testing in the Marine Cybernetics laboratory. Lastly, a discussion regarding the experimental results and a general discussion regarding backstepping and linear PID will be give.

Chapter 4 - Case study: Online risk analysis

Chapter 4 introduce case study II: Online risk model. The case study contains a description of of the Bayesian belief network developed, the implementation of the online risk model and the three different scenarios tested. Following, simulation and experimental results are illustrated. Lastly, an discussion regarding the results and strengths and limitations of the model will be given. This case study is performed in collaboration with Ina Bjørkum Arneson.

Chapter 5 - Experimental setup

Chapter 5 describes the experimental setup used in case study I and II. Firstly, characteristics of CyberShip Enterprise I is given. Following a description of the hardware and software architecture of the hardware-in-the-loop and Marine Cybernetics laboratory setup. Lastly, the software used for the online risk model will be given. This chapter is written in collaboration with Ina Bjørkum Arneson.

Chapter 6 - Conclusions and further work

Chapter 6 covers the conclusion of the research questions and main challenges which might be improved in further work.

Chapter 2

Background and preliminaries

The following Chapter is divided into six main parts, covering the necessary background material and preliminaries. The first section concerns situation awareness related to conventional and autonomous surface ships. Secondly, a review on the definitions of levels of autonomy is given. Following, an introduction to risk aspects associated with autonomous surface ships and motivation for online risk modeling. Lastly, necessary background material on control theory is given.

2.1 Situation awareness

For a vessel to operate with some level of autonomy, an accurate understanding of the vessel's surroundings is imperative. Situation awareness can be expressed as having a good perception of your vessel's surroundings at all times, understanding and predicting how the surroundings can affect your vessel. A ship's surroundings are a constantly evolving picture. Hence, essential parameters in order to have a good situation awareness are knowledge of other vessels in the current area, communications between ship traffic services and other smaller boats, and a understanding about the weather, sea state, depth of water, tide and current, are also crucial. Achieving an understanding of all the surrounding parameters can be challenging. Thus the need for sensors that provide correct and accurate information of the vessel's surroundings is necessary. Figure 2.1 illustrates the process of situation awareness, SA.

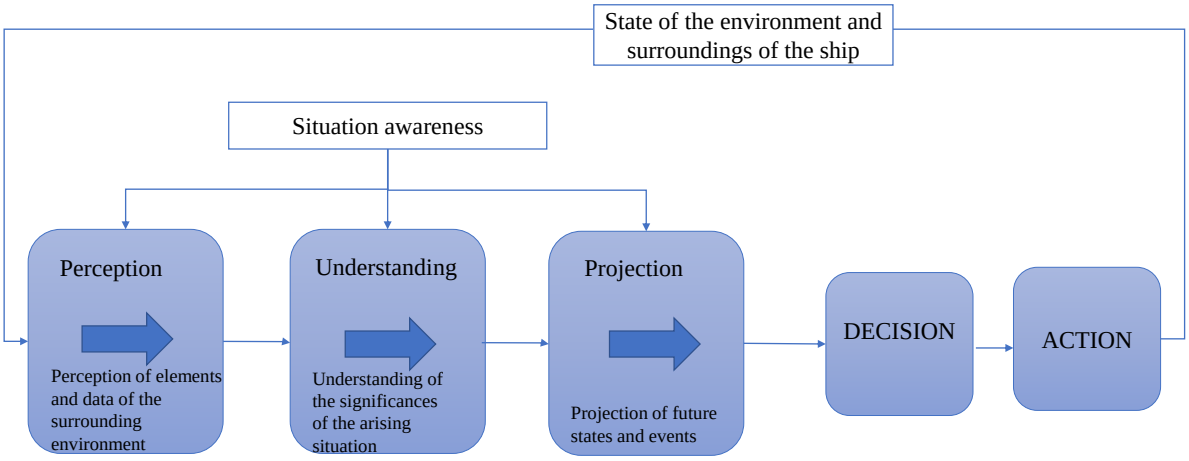


Figure 2.1: Illustration of SA

Today the majority of cargo vessels anti-collision characteristics are heavily based on data from AIS, RADAR, and bridge manning. Following an overview of the primary sensors used on non-autonomous surface vessels today, as well as sensors which can increase the situation awareness in case of partly or fully autonomous vessels.

2.1.1 GNSS

GNSS, Global Navigation Satellite System, refers to satellites from space providing signals that transmit position and time data which are used to determine location. GNSS provides global coverage. GNSS is the most commonly used navigation system for marine vessels. At least four satellites must be visible in order to compute a reliable position estimate when using a satellite navigation system. However, if the vessel is entering a shadowed region, the position will be lost if redundant signals are not available.

2.1.2 RADAR

RADAR, RAdio Detection And Ranging, is an object detection system that uses radio waves to determine the distance, angle, and velocity of an object. The RADAR system is developed by the military and consists of a transmitter that produces electromagnetic waves, a transmitter/receiving antenna and a receiver and processor used in order to determine the properties of the object. Marine RADARS is used to measure the bearing and distance of a ship and is one of the most important equipment in ship navigation.

2.1.3 AIS

AIS provides information about heading and vessel structure and creates a virtual picture in real time. AIS is only required on larger vessels, hence a great deal of data will get lost in congested areas where fishing boats and smaller passenger vessels are present.

2.1.4 LIDAR

LIDAR is a technology that provides accurate and precise real-time position information of moving objects and infrastructure, thus often used for mapping the environment around a vessel. The LIDAR has a minimal return off the water, which makes it suitable for the detection of objects in the ocean. Thus, they can recognize individual ships when clustered together, separate size, and detect very small boats. This is done by illuminating the vessel's surrounding environment with a laser and analyzing the reflected pulses with a sensor. LIDAR is ideal for sensor fusion with RADAR, camera systems, AIS and electronic charts [8].

2.1.5 Drones

Drones are an essential part in the Industrial Internet of Things, a network where machines, advanced analytic, people and communication technologies are combined in order to exchange and analyze data, which can help towards smarter and faster solutions. The drone technology is continuously evolving and can be used to increase work efficiency, productivity, accuracy, and reducing operational costs. According to the Amazone LLC affiliate advertising program website, drone technology are divided into seven generations, where the majority of toady's drones are in generation 6. [6] have defined generation 6 and 7 as:

Generation 6: Commercial Suitability, Safety and Regulatory Standards-Based Design, Platform Payload Adaptability, Automated Safety Modes, Intelligent Piloting Models and Full Autonomy, Airspace Aware [6].

Generation 7: Commercial Suitability, Fully Compliant Safety Regulatory Standards-Based Design, Platform Payload Interchangeability, Automated Safety Modes, Enhanced Intelligent Piloting Models and Full Autonomy, Full Airspace Awareness, Auto Action (takeoff, land, mission execution) [6].

The 7th generation drone is underway, called 4DRobotics. Which is the first Smart Drone. That is a fully autonomous drone with accurate sensors, self-monitoring and built in safeguards [6]. Thus, drone technology will provide great opportunities in the commercial sectors. Already during the last couple of years, drones have become a common tool in pioneering firms in the construction, agriculture, mining, surveying, and real estate industry. Airbus is an example of a commercial drone company, which this year launched a drone for advanced inspections of aircraft. Further, Airbus and Wilhelmsen Ship Service have collaborated in developing a drone for an unmanned delivery system [11]. Whether a drone is used in the media industry, real estate or in the maritime industry, it possesses the ability to reach the most challenging areas with reduced human resources.

A drone has a excellent precision and can collect a big amount of data and information. Accordingly, the potential for drones in the shipping industry is significant. By utilizing a drone separates or in combination with cameras, the complete situation awareness will increase. Thus, by making a drone an everyday tool on a vessel passage, the officer on a vessel can identify obstacles several miles away from the ship's current position. A drone with object detection, can detect both small and large objects in a range of different environments at sea. In addition to increasing the situation awareness, drones can be used for inspections of the vessel and deliveries between vessels. Respectively, this implies that a drone has great potential and will possibly be an excellent resource for a ship owner, and a beneficial device at a control shore center.

2.2 Levels of autonomy

Automation is a process where a specific and predefined method is implemented to perform certain operations without a human controlling it, this process is often computerized. An automatic system is a system that can perform a given set of operations without instructions from a human, while an autonomous system is a system that implements a form of self-governance. The interest in unmanned autonomous vehicles has increased drastically in the last couple of years. Consequently, several different research activities concerning unmanned cargo ships are in progress today [32]. The concept of autonomous systems may lead to smarter solutions, higher cost efficiency, and reduced transport volumes. However, autonomous systems may also lead to emerging risks and challenges related to safety. In order to discuss the safety in context with autonomy, a study of the levels of autonomy will be reviewed. The following Chapter is divided into three Sections. Firstly the taxonomy regarding autonomous systems from NTNU AMOS will be reviewed, followed by taxonomy regarding autonomous ships from the Norwegian Forum for autonomous ships. Lastly, a brief introduction of a shore control center will be given

2.2.1 Taxonomy - NTNU AMOS

In the article *Risk management of autonomous marine systems and operations* from NTNU AMOS, Center for Autonomous Marine Operations and Systems, autonomy is defined as: *a sub-system's own ability of integrated sensing, perceiving, analyzing, communicating, planning, decision-making, and acting to achieve its goals as assigned by its human operator(s) through designed human-machine interface (HMI)* [28]. Thus, an unmanned system can be defined as a subcategory of autonomous systems. Autonomy may be divided into levels based on the independence of a human operator or external systems. NTNU AMOS have defined the definitions for unmanned and manned autonomous systems and operations into the following four levels [28]:

Level 1: Automatic operation
At this level of autonomy the system operates automatically. The system is often programmed to do specific tasks. If any deviations from the boundaries of the operation arises the system is controlled by a human operator. The human operator does also directs and control all high-level mission planning functions through a human-machine-interface (HMI).

Level 2: Management by consent
The system automatically makes recommendations for mission or process actions related to specific functions. That is, the system can prompt the human operator at important points in time for information or decisions. Thus, the system can perform many functions independently of human control when delegated so. At this level the system might have limited communications capabilities due to distances.

Level 3: Semi-autonomous operation/management by exception
At this level the system automatically executes mission-related functions when response time is too short for human intervention. The human operator may intervene in the operation, and has the power to cancel or redirect actions within defined time lines. This level can be called human - supervisory control, as the human operators attention is only brought to exceptions for certain decisions.

Level 4: Highly autonomous operation
The system automatically executes missions or process related functions in an unstructured environment with ability to plan and re-plan the mission or process. This system can be called

human-out-of-the loop, as the system is independent and intelligent.

2.2.2 Taxonomy - The Norwegian Forum for autonomous ships

The Norwegian Forum for autonomous ships (NFAS) is an initiative established by the Norwegian Maritime Administration, The Norwegian Coastal Administration, The Federation of Norwegian Industries and SINTEF Ocean in 2016. NFAS is an interest group with the aim to provide knowledge, discussion, and development of autonomous ships. NFAS has developed a set of high-level definitions on autonomous ship types, as well as stated the relationship between the different levels of autonomy. The definitions presented in this Section are from the report, *Definitions for Autonomous Merchant Ships*, and will be used further in the report [5].

NFAS has defined autonomous as a ship that can perform a set of defined operations with no or reduced attention from a bridge crew. Accordingly, the term autonomous vessel can be used to describe a vessel both with and without bridge crew, as long as the vessel performs a set of defined operations with reduced attention from the bridge crew. Further, autonomy is defined as the result of applying advanced automation to a vessel that implements a form of self-governance. According to the definitions of NFAS autopilot is not a form of autonomy, as the vessel is following a given heading, and is not selecting any alternative strategies. Figure 2.2 illustrates the classification of autonomous maritime vehicles.

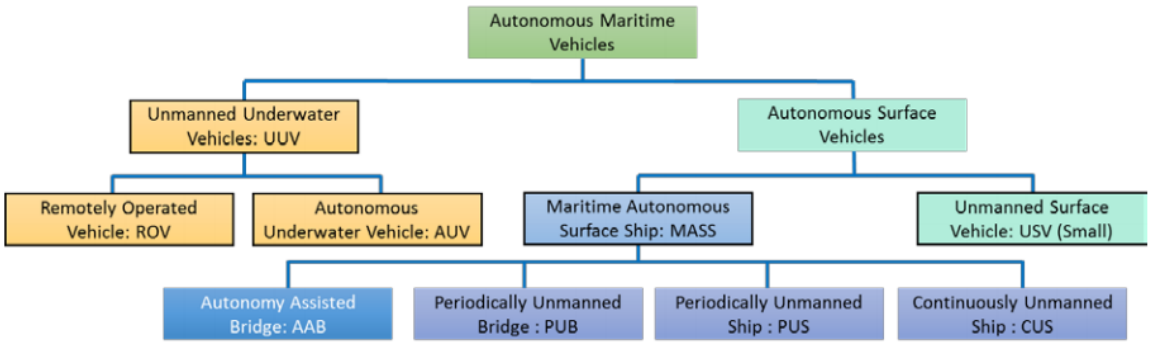


Figure 2.2: Taxonomy NFAS: Definitions of different types of autonomous maritime vehicles [5]

In this master thesis the research is based on a merchant vessel, hence Maritime Autonomous Surface Ship, MASS, will be investigated further. MASS can be divided into the four definitions listed below.

- **Autonomy Assisted Bridge (AAB)**: The bridge is always manned, and the crew can intervene in ongoing functions.
- **Periodically Unmanned bridge (PUB)**: The vessel can operate without crew in limited periods, typical on open oceans, but crew is present on the vessel.
- **Periodically Unmanned Ship (PUS)**: The vessel operates without crew during deep-sea passage. A boarding team enters to control the vessel when approaching port.
- **Continuously Unmanned Ship (CUS)**: The vessel is designed for unmanned operation of the bridge at all times, except during special emergencies.

Vessel type	Crew Always on Bridge	Crew Available on Ship	Crew Never on Bridge
AAB	X		
PUB		X	
PUS			X
CUS			X

Table 2.1: Four definitions of MASS

2.2.3 Shore control center

A shore control center (SCC) is the vessel owners center for monitoring and control. Hence it will be used as a backup or remote-control center, depending on the level of autonomy [4]. The overall goal is that the SCC will ensure a remote control or supervision over the vessel, this can be the vessels navigation, machinery, maintenance of the mechanical and electrical installations on the ship, propulsion or overall security of the vessel. A shore control center can simplify the technology and increase the safety and security of autonomous vessels. The MUNIN project [4] stated that one human operator will manage several similar ships, where the human operator can take full control of the ship and handle it if an unwanted situation arises. The setup of the SCC is as a full-scale vessel bridge, consisting of experienced masters and the officer on watch, OOW. In figure 2.3, a proposed organization structure from SINTEF is illustrated. The figure illustrates one operator per sixth vessel and one supervisor per the thirtieth vessel. In addition, a relieve operator, situation room, engineer, and captain per thirtieth vessel.

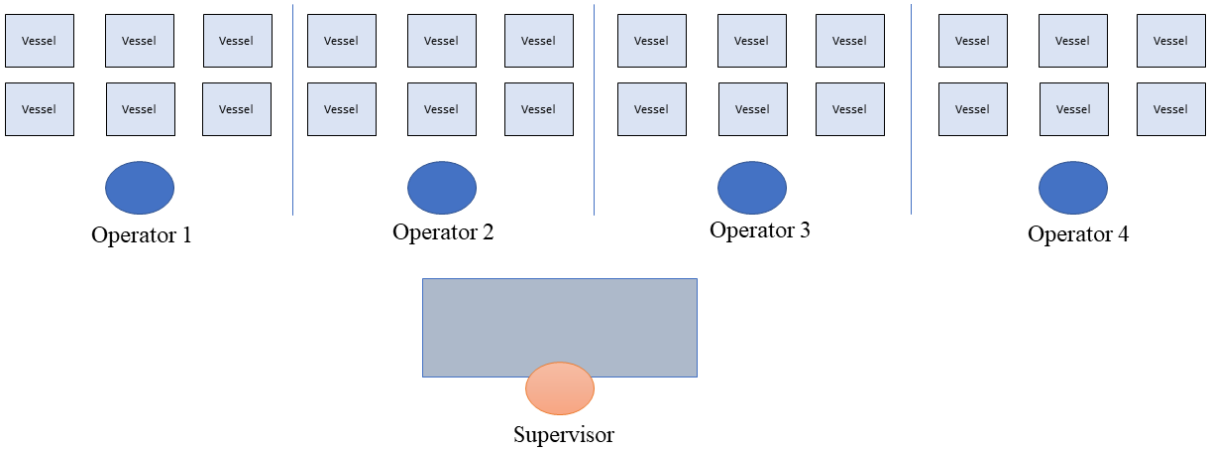


Figure 2.3: Proposed organisation of shore control center [20]

2.3 Risk aspects associated with autonomous surface ships

Increased safety, reduced operational cost, and decreased emissions are some of the benefits associated with autonomous surface vehicles. In the last decades the maritime industry has developed drastically, and in particular in terms of navigation equipment. Today, the majority of surface vessels are minimum equipped with AIS, radar, and bridge manning. Despite the extensive development of sensors and navigation equipment, the numbers from the International Union of Marine Insurance (IUMI) are indicating an increase in the number of accidents related to the shipping industry [18]. Depending on LoA, an autonomous system may introduce smarter, more efficient operations or support to the bridge crew during decision-making. However, the increasing level of autonomy introduces new risk aspects and failure modes [28]. In this section, different risk aspects associated with different levels of autonomous surface vehicles will be discussed

An autonomous surface vehicle can be divided into two parts: Software and hardware. The software part consists of decision-making for the system and communication between the different hardware parts. The hardware on the vessel is the physical systems such as engines, thrusters, sensors, etc. For an autonomous vehicle to operate in the near future, the autonomous vessel needs to provide equivalent safety to conventional vessels. It follows that it is critical that the vessel have robust and redundant hardware and software. In other words, minimum risk conditions should be established in order to prevent a hazardous event. Cybersecurity is an increasingly prominent concern in context with autonomous systems. The number of cyber-attacks is increasing, with growing digitization worldwide. In June 2017 a significant cyber-attack called notPetya attacked several companies in Europe and America. Where the business interruption of the Danish shipping company A.P. Moller-Maersk was estimated among $200 - $300 million in lost revenues. More recently (2019) one of the world's biggest aluminum producer, Norske Hydro, was hit by an unknown cyber-attack demanding a ransom. In other words, identifying potential security vulnerabilities associated with the embedded control systems, network, and sensor systems are considered crucial for autonomous surface vehicles to be operating.

Autonomous systems are computerized and mostly dependent on long-range wireless satellite communication. *A hazard is a source of danger that may cause harm to an asset* [19]. Related to autonomous systems power loss, grounding, and collision with undetected obstacles in the oceans, are possible hazards. The latter hazards may reduce situation awareness and control of the vessel. In such situations, it is a reason to believe that a shore control center will be contacted and given command of the vessel. However, current experiences indicate that in a wide range of circumstances communication links can be disturbed or broken [28]. Another risk influencing factor is GPS spoofing, which describes attempts on attacking a GPS receiver by broadcasting false GPS signals. For a fully autonomous surface vessel, GPS spoofing could impact the navigation of the vehicle. By attacking the sensors of the ship, a malicious user can cause denial-of-service (DoS) attacks, as well as gain full control of the vessel. These RIFs are present at conventional surface vehicles. However, an autonomous system is to a more significant extent, dependent on sensors, algorithms, and decision-making form an embedded system. Consequently, when designing autonomous systems, it is critical that the systems' security vulnerabilities are exploited and discovered early on [31].

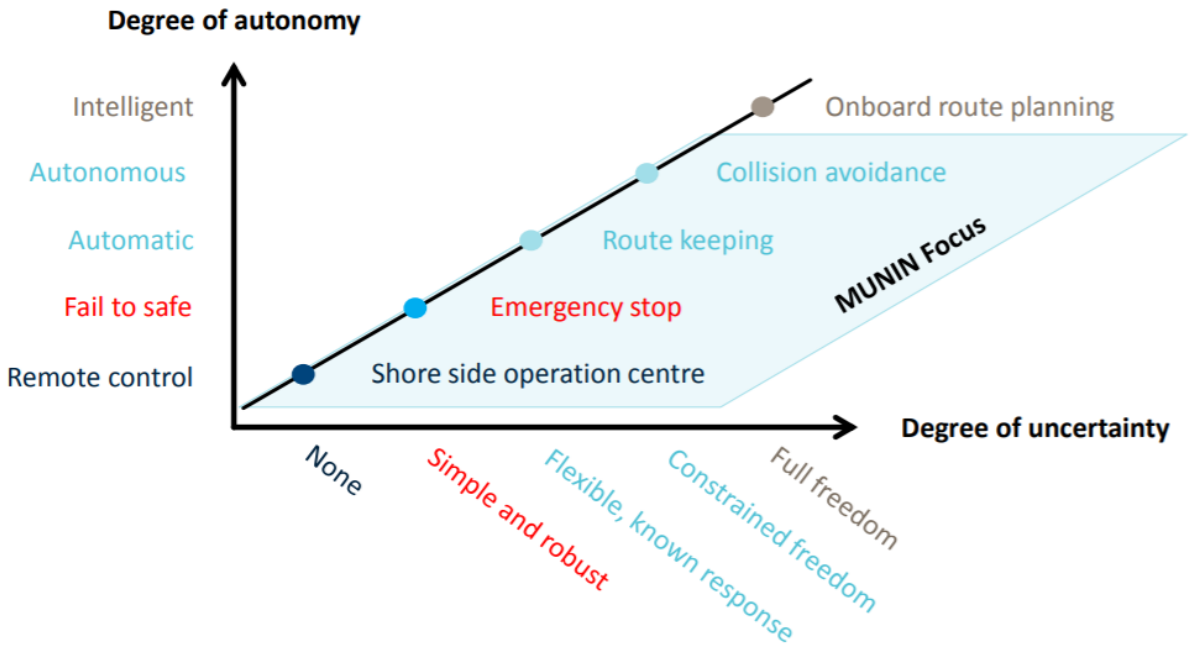


Figure 2.4: An illustration of the *right level* described by SINTEF [20]

2.4 Risk analysis

For a conventional cargo vessel passage, the bridge is manned with a minimum of two men at all times. In some passages the officer on watch (OOW) might work an entire shift with only one task, looking at the radar screen. Low activity level for hours might result in mental fatigue and reduced responsiveness. By introducing a periodically unmanned bridge, the OOW and additional crew on the bridge can engage in other tasks during the work shift. Today the deck officer's duty and responsibilities is to navigate safely and guarantee that the vessel compiles with regulations. From ABB's paper on *B0 - a conditionally and periodically unmanned bridge* one can generalise the responsibilities of the OOW into the following [16]:

1. Check navigational equipment in use at regular interval of time
2. Prepare, execute and monitor a safe passage plan
3. Ask for support whenever requires
4. Contact the master when need arise
5. Not leave the bridge unattended during the assigned watch

The concept of B0 presents the general conditions where it is possible to transfer the monitoring responsibility from the OOW at the bridge to a machine, fully or over an ascertained period. During a vessel voyage of a conventional vessel, the bridge crew is consistently performing a real-time risk assessment. The bridge crew is looking out of the windows performing object detection, using the navigation system and taking decisions based on the weather forecast and surroundings of the vessel. The development of unmanned vessels represents a solution for the reduction of operational and environmental expenses. When considering autonomous vessels, online risk assessment is of particular importance. In this Section, an introduction to Bayesian belief networks and fuzzy logic will be given.

2.4.1 Bayesian belief networks

The Bayesian belief network model is based on Bayes's rule and conditional probability. Bayesian networks are a risk analysis method commonly used for reasoning under uncertainty [1]. In other words, BN is a probabilistic graphical model, directed acyclic graph, based on data and expert opinions. Each edge in the graph represents a conditional dependency, and each node corresponds to a unique random variable which might be discrete or continuous [12]. An edge is connecting random variables, A and B, which can be expressed as (A, B). The direction of the edges represents causalities between the variable.

The purpose of using Bayesian networks is to obtain a compact factorized representation of a joint probability distribution. Each node is conditionally independent of its non-descendants given its parents, i.e., satisfying the local Markov property. This is a significant simplification, as it results in fewer parents relative to the overall size of the network. Bayesian networks are causal, i.e., cycles are not allowed, contrary to Markov networks. The joint Bayesian network distribution can be simplified and expressed as:

$$P(X_1, .., X_m) = \prod_{i=a}^{n} P(X_i|X_1, ..., X_{i-1}) = \prod_{i=a}^{n} P(X_i|Parents(X_i)) \tag{2.1}$$

The links between the nodes indicate which nodes are directly influenced by one another. However, if a node is not directly connected to another node by a link, this does not directly indicate that they are entirely independent of each other. It might be a dependency through another node. If there is a link from variable B to variable A, this indicates that variable B is a parent of A and A is a child of B. In figure 2.5 an illustration of a serial connection is represented. In this serial connection case, B can directly influence C, and C can influence B indirectly. However, if A is known, C cannot affect our belief in B. In other words, B and C are d-separated given A. If a d-separated variable is passing evidence to one of the variables, that will have no impact in our belief in that variable as d-separated variables are independent.

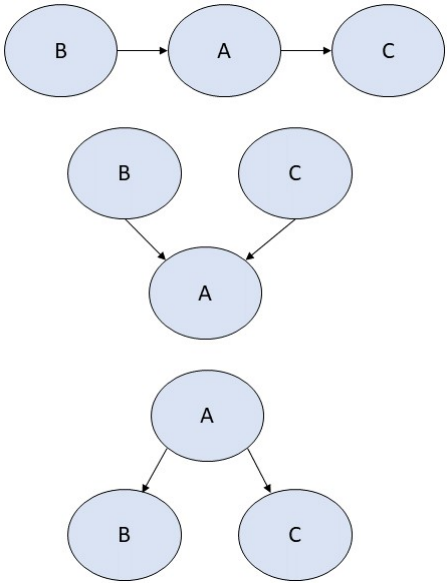


Figure 2.5: Serial connection (upper), diverging connection (middle) and converging connection (lower

BBN modelling methodology

Following, a simple description of the development of BBN will be given based on the methodology presented in the article *A Bayesian approach to risk modeling of autonomous subsea intervention operation* [9]. When constructing BBN several approaches can be used. In this thesis, a generic 8 step BBM modeling method presented in [9] is utilized.

Step 1: Identify target node. When developing a BBN model, the first step is to define the target node. A target node could, for instance, be the probability for collision during vessel passage or probability for grounding. In other words, the target node is the motivation for calculating the joint probability distribution, hence the problem that needs to be solved.

Step 2: Identify nodes. When the target node is defined, a risk identification of the case should be performed. By this, identification of all the possible hazards should be identified.

Step 3: Structure causal relations between nodes. When developing a BNN it is of importance that the node and causal relationships are represented as close as possible to the real-world problem. As mentioned the arcs represents causal relationships, as well as define parent-child relationship.

Step 4: Identify different states of the nodes. When the different nodes and the causal relations between the nodes are determined, the different states of the node should be specified. When defining the different states, it is vital that they can be represented by data or measurements [1]. It is normal to divide the states into best and worst, or low, medium, high.

Step 5: Analyse causal relations. Next, it is convenient to determine which nodes that are independent of each other, and which nodes that are d-separated. This can help to decrease the CPTs, which is a high advantage. In this step, one does also want to review if the d-separation there occurs when supposed.

Step 6: Quantify the model. When the nodes, states, and causal relationship are determined the conditional probability tables, CPT, can be allocated. In some cases the BBN can be extensive, thus it can be challenging to allotting CPTs. However, different methods can be used to decrease the network; an example is fuzzy logic.

Step 7: Update evidence according to scenarios/data. Evidence can be updated by the access of existing data or scenario generation. When new evidence is accessed, the model is updated, and a new joint probability distribution is achieved.

Step 8: Interpret results. In this step on want to observe the different results of the different states with different evidence. Inference can be made in this step.

2.4.2 Fuzzy logic

Risk models are traditionally developed based on probability. The classical approach to probability is based on a set of situations within a finite number, where all possible outcomes have the same likelihood [19]. In the classical approach, an element is included or not included in the set based on precise information regarding the situation. However, in some cases, lack of experience, data, or entangled cause-and-effect relationship make it challenging to determine the risk with traditionally risk models. In this case, fuzzy logic is commonly used. Fuzzy logic is convenient in large scale risk management, where a quantitative probability model is not available [21]. In fuzzy logic, an element is included in the set based on a degree of truth, commonly between 0 and 1. An important detail of fuzzy sets is that there are no strict rules regarding how the relationships are defined, as long as it is consistency. Fuzzy logic can be used in combination with Bayesian belief networks.

2.5 Hardware-in-the-loop simulation

Hardware-in-the-loop simulation (HIL) is a technique used for advanced testing of embedded systems. In the automotive and aerospace industries, HIL testing is shown to be the best method to meet the requirements for performance and safety. HIL testing is also utilized in NASA to verify mission-critical software components for spacecrafts. In the process towards partly or fully autonomous surface vessels, a significant amount of vessels will be equipped with software-based control systems. Thus, software related problems in combination with hardware-errors or human errors may lead to downtime, reduced safety, and increased cost. It follows that full-scale vehicle testing will be expensive and time-consuming to perform for an autonomous vessel where the need to test every possible scenario is crucial. As a result, HIL is a testing method increasingly utilized. The method provides scalability and repeatability of each experiment. Thus the consistency of each experiment or test improves, and the variation decreases [14]. HIL testing is applied for extensive software testing and verification of the embedded system and is often used under development of power system and marine systems, to verify the stability, operation or fault tolerance of the system. Benefits of HIL testing are safer and more robust automation system [14].

Figure 2.6: Hardware-in-the-loop simulation [10]

In HIL testing a real-time computer is used as a virtual representation of the plant model, where the controller design can be tested. The objective of HIL testing is verification and testing of the computer software/simulation model using a simulator that is capable of simulating the vessel dynamics, thrusters, propulsion system, sensors, position reference system, power generators, etc. of the particular ship.

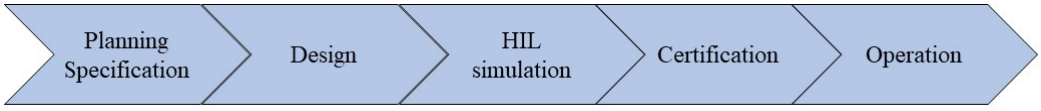


Figure 2.7: Hardware-in-the-loop simulation: Development process of control systems

The simulator is usually viewed as a black box, meaning no knowledge of the inner workings of the control system is necessary [10]. The HIL box is connected to a network or a bus interface, which is the part that makes it possible to simulate feedback and command signals between the simulator and control system. Independent hardware-in-the-loop testing is a method normally used to assist in verification and testing of an embedded system. However, DNV GL is today offering a 3rd party certification of the HIL testing to the customer. In general, a control system interacts with several components in the vessel system through input and output signals (I/O). Alternatively, a HIL simulator is isolating the control system from the vessel's surroundings, and replace the I/O - signals with simulations from HIL [10]. Hence, realistic imitation of the vessel environment and systems is achieved as the control system does not recognize any different from the real-world and simulation. Figure 2.8 gives an illustration of the concept.

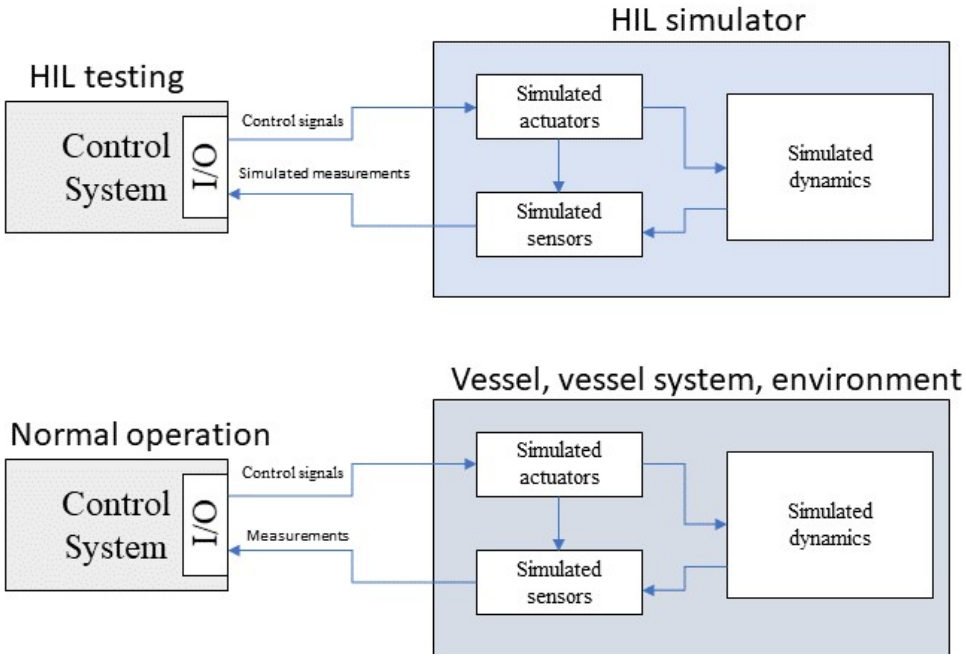


Figure 2.8: HIL simulation vs normal operation [25]

2.6 Control theory

In [17], figure 2.9, a control architecture for unmanned underwater vehicles are proposed. The architecture is divided into three levels: the Mission layer, Guidance, and optimization layer and control execution layer. The mission layer is the part where the mission is planned and re-planned based on inputs from sensor measurements. The guidance and optimization layer is the part of the architecture where way-points are managed and supplied to the controller. In addition, data from the payloads are processed and delivered to the mission layer. The control and execution layer consist of the controller and navigation sensors, i.e., data from the navigation sensors are analyzed. In this subsection, the controller box is to be analyzed.

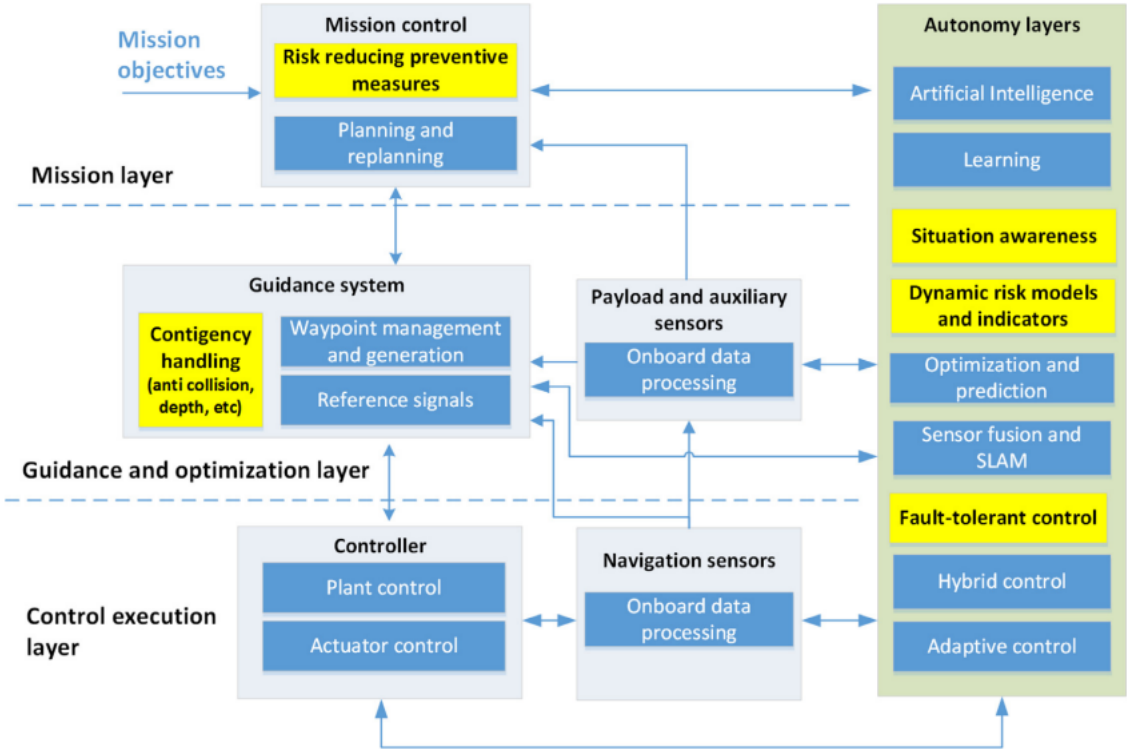


Figure 2.9: Control architecture for unmanned underwater vehicles [17]

The controller box does often consists of an autopilot system or a dynamic positioning system. An autopilot control system is 2 degrees of freedom voyage control, where the objective is to determine the yaw and surge force in order to keep a desired heading. In this thesis, dynamic positioning will be considered. The purpose of dynamic positioning, DP, system is to keep the position or heading of the vessel relative to the seabed, while the vessel is subjected to external forces: wind, current, and waves. This is achieved by the use of two or more propulsive devises where these devices are controlled by inputs from instruments, gyro-compass, satellite navigation, etc. In other words, it is extremely important that the necessary redundancy requirements are met in order to meet the designed positioning capability [26]. Firstly the different reference frames will be considered, followed by an introduction of the linear PID algorithm and lastly an introduction to integration backstepping.

2.6.1 Reference frame

When analyzing the motion control of a marine vessel or marine craft, it is necessary to consider the reference frame the vessel or object, is considered in. In navigation, a reference frame is a predefined coordinate system used to define the motion of an object relative to another coordinate system - the coordinate system to the environment the object is navigating in. The table below represents the notation of SNAME (1950) for marine vessels.

DOF		Forces and moments	Linear and angular velocities	Positions and Euler angles
1	Motion in the x-direction (surge)	X	u	x
2	Motion in the y-direction (sway)	Y	v	y
3	Motion in the z-direction (heave)	Z	w	z
4	Rotation about the x-axis (roll, heel)	K	p	ϕ
5	Rotation about the y axis (pitch, trim)	M	q	θ
6	Rotation about the z axis (yaw)	N	r	ψ

Table 2.2: The notation of SNAME (1950) for marine vessels [7]

The North-East-Down (NED) coordinate system $n = (x_n, y_n, z_n)$ is defined relative to the earth's reference ellipsoid. In this reference system the x-axis points towards true North, the y-axis point towards true East and the z-axis points downwards normal to the Earth's surface. In other words, this is the coordinate system we refer to in our everyday life. The body-fixed reference frame $b = (x_b, y_b, z_b)$ with the origin in the center of the vessel, o_b, is a moving reference frame. For a vessel the body axes $b = (x_b, y_b, z_b)$ are usually chosen to coincide with the principal axes of inertia. Thus, x_b - longitudinal axis (directed from aft to fore), y_b - transversal axis (directed to starboard) and z_b - normal axis (directed from top to bottom). In other words, when the vessel is moving towards true North, Body - and NED - reference frames are parallel. The NED- and Body - reference frames are illustrated in figure 3.9.

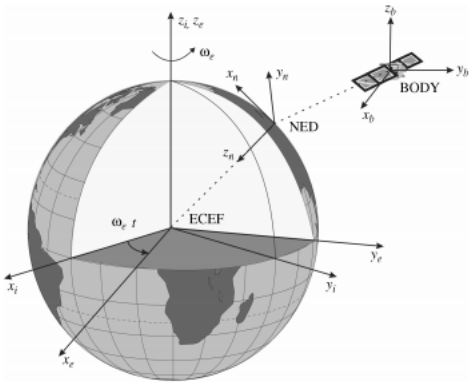


Figure 2.10: Reference frames: NED and Body [7]

2.6.2 Maneuvering problem

In the path following control problem, an entire path from a to b is considered. From Skjetne Ph.D. thesis *The Maneuvering Problem* the path following control problem is stated as the following: "*In vehicle applications the practice is often to manually set the forward propulsion to a desired value, and then actively use the steering capacity of the vehicle to reach and stay on the path*" [22]. In the maneuvering control problem, one would instead trace each point one by one out of the path. This is achieved by comprising the maneuvering problem into two tasks, the geometric task, and the dynamic task. The task of the geometric part is to follow the given path, and thus have the most crucial job in the maneuvering control problem. In other words, for a continuous function, the geometric task is to force the output η to converge to η_d which is the desired parameterized path. Mathematically this can be expressed as:

Geometric task:

$$\lim_{x\to\infty} |y(t) - y_d(s(t))| = 0 \tag{2.2}$$

The second task, the dynamic task, has the job of satisfying the desired dynamical behavior along the path. This could, for instance, be to achieve the optimal speed. Accordingly, for a surface vessel, the aim would be that the vessel speed is sacrificed in order the keep the desired path. In other words, the vehicle should satisfy the speed assignment in order to trace the path. That is, to control $\dot{s}$ in order to converge to a desired speed assignment $U_s(s,t)$. However, the desired dynamical behavior along the path can also be expressed in terms of time assignment and acceleration assignment along the path. The time assignment is defined as to be at specific points along the path at a specific time. Accordingly, the acceleration assignment can be expressed as to obtain a desired acceleration along the path. Mathematically this is expressed the following:

Dynamic task:

1. Speed assignment: Where s is forced to converge to a desired time assignment τ(t),

$$\lim_{t\to\infty} |\dot{s}(t) - \tau(t)| = 0 \tag{2.3}$$

2. Time assignment: Where $\dot{s}$ is forced to converge to a desired speed assignment $u_s(s,t)$,

$$\lim_{t\to\infty} |s(t) - u_s(s(t),t)| = 0 \tag{2.4}$$

3. Acceleration assignment: Where $\ddot{s}(t)$ is forced to converge to a desired acceleration assignment $\alpha(\dot{s}(t), s(t), t)$,

$$\lim_{t\to\infty} |\ddot{s}(t) - \alpha(\dot{s}, s(t), t)| = 0 \tag{2.5}$$

2.6.3 Path parameterization

In order to achieve the maneuvering control problem, the goal is to follow a desired and parameterized path. The path parameterization can be obtained in several different ways. The path parameterization can be continuous, discrete, or hybrid, which is a combination of the two latter. The focus of this thesis will be continuous.

The desired position with first and second derivatives is given below, where p_x is the desired position in the x-direction and p_y is the desired position in y-direction.

$$\eta_d = \begin{bmatrix} p_x(s) & p_y(s) & arctan\left(\frac{p_y^s(s)}{p_x^s(s)}\right) \end{bmatrix}^T \tag{2.6}$$

$$\eta_d^s = \begin{bmatrix} p_x^s(s) & p_y^s(s) & \frac{p_x^s(s)p_y^{2s}(s) - p_y^s(s)p_x^{2s}(s)}{p_x^s(s)^2 + p_y^s(s)^2} \end{bmatrix}^T \tag{2.7}$$

$$\eta_d^{2s} = \begin{bmatrix} p_x^{2s}(s) & p_y^{2s}(s) & \frac{p_x^s(s)p_y^{3s}(s) - p_y^s(s)p_x^{3s}(s)}{p_x^s(s)^2 + p_y^s(s)^2} - \frac{(p_x^s p_y^{2s} - p_y^s p_x^{2s})(2p_x^s(s)p_x^{2s}(s) + 2p_y^s(s)p_y^{2s}(s))}{(p_x^s(s)^2 + p_y^s(s)^2)^2} \end{bmatrix}^T \tag{2.8}$$

A speed assignment $U(s,t)$ for $\dot{s}$ that makes the vessel follow a path at a constant speed U_{ref} is given in (2.9).

$$\lim_{t\to\infty} |\dot{s}(t) - U_s(s,t)| = 0 \tag{2.9}$$

Using this speed assignment, $U_s(s,t)$ can be expressed by (2.10), and $U_s(s,t)^s$ can be expressed by (2.11).

$$U_s(s,t) = \frac{U_{ref}}{\sqrt{p_x^s(s)^2 + p_y^s(s)^2}} \tag{2.10}$$

$$U_s(s,t)^s = -\frac{1}{2}\frac{U_{ref}}{(p_x^s(s)^2 + p_y^s(s)^2)^{\frac{3}{2}}} \cdot \left(2p_x^s(s)p_x^{2s}(s) + 2p_y^s(s)p_y^{2s}(s)\right) \tag{2.11}$$

Using initial position $p_d(0) = (x_0, y_0)$ and final position $p_d(1) = (x_1, y_1)$ a straight-line parametrization is proposed in (2.12)

$$\eta_d = \begin{bmatrix} (x_1 - x_0)s + x_0 \\ (y_1 - y_0)s + y_0 \\ arctan(\frac{y_1 - y_0}{x_1 - x_0}) \end{bmatrix} \tag{2.12}$$

This results in variables $p_x, p_y, p_x^s, p_y^s, p_x^{2s}, p_y^{2s}, p_x^{3s}, p_y^{3s}$ as below, and the second and third derivatives are zero.

$$p_x = (x_1 - x_0)s + x_0 \qquad p_x^s = (x_1 - x_0) \tag{2.13}$$

$$p_y = (y_1 - y_0)s + y_0 \qquad p_y^s = (y_1 - y_0) \tag{2.14}$$

2.6.4 Proportional-integral-derivative control

A conventional linear proportional-integral-derivative, PID, controller is a well-used controller in the industry. This is mainly due to the PID controllers simplicity, applicability, and functionality. However, a downside with the linear conventional PID controller is that it can be quite time consuming to tune the controller gains. The PID - algorithm can be expressed as:
In the frequency domain:

$$\begin{aligned} G(s) &= K_P + K_I\frac{1}{s} + K_D s \\ &= K_P(1 + \frac{1}{T_1}s + T_D s) \end{aligned} \tag{2.15}$$

In the time domain:

$$\begin{aligned} u(t) &= K_P e(t) + K_I \int_0^t e(t')dt' + +K_D\frac{de(t)}{dt} \\ &= K_P e(t) + \frac{1}{T_i}\int_0^t e(t')dt' + +K_D\frac{de(t)}{dt} \end{aligned} \tag{2.16}$$

Where $K_p > 0$ is the proportional gain, $K_I > 0$ is the integral gain, and $K_D > 0$ is the derivative gain. The proportional term of the equation is providing a control gain proportional to the error signal e(t) — I.E., between the setpoint and a measured value. A high K_p can cause an unstable system with oscillations, contrary if the proportional gain is too low, the control action may give a too low response or not respond at all. The integral gain is expressed by K_I. The integral term sums the error term over time. Thus, for every small error, the integral term will increase until zero error is achieved. In other words, the integral term eliminates the steady-state error produced in the P-term. However, the downside with the I-term is that it can contribute to overshoot the setpoint. The shorter the period it integrates over, the more aggressively it acts. The derivative term predicts the system's behaviour and will try to decrease the error. In other words, the derivative term has a damping effect on the system, where the objective is to cancel out the error produced by the P, and I terms.

2.6.5 Integrator backstepping

This section is fully based on *Handbook of Marine Craft Hydrodynamics and Motion control* in [7].
Backstepping is a method used for the construction of a feedback control law through a recursive construction of a control Lyapunov function. The main difference between feedback linearization methods and backstepping is that feedback linearization methods cancel all nonlinearities in the system, in contrast to backstepping, where more flexibility is obtained. In other words, when using backstepping, the designer is given the possibility to utilize *good* nonlinearities and eliminate *bad* nonlinearities by, for instance, adding nonlinear damping. In practice, it is demanding to obtain precise models in order to achieve the cancellation of all nonlinearities. Thus the robustness obtained through backstepping is convenient. Vectorial backstepping utilizes the structural properties of a nonlinear MIMO, multivariate input multivariate output, systems. This simplifies design and analysis. The vectorial backstepping method is developed by Fossen and Berge (1997) [7]. Following, the main idea of integrator backstepping is to be derived by considering a simple nonlinear scalar system:

$$\dot{x}_1 = f(x_1) + x_2 \tag{2.17}$$

$$\dot{x}_2 = u \tag{2.18}$$

$$y = x_1 \tag{2.19}$$

Where $x_1 \epsilon \mathbb{R}$, $x_2 \epsilon \mathbb{R}$, y $\epsilon \mathbb{R}$ and u $\epsilon \mathbb{R}$. The design object of integrator backstepping is to render the equilibrium point GES or GAS. The only equilibrium point with y = 0 is (x_1, x_2) = (0, -f(0)) which is equivalent to $\dot{x}_1$ = f(0) + x_2. Further, the system is considered as two SISO cascaded systems. Hence, the recursive design starts with the system x_1 followed by x_2. In addition, it is known that the relative degree of the system is two. Accordingly, it is known that the control input appears in the second step. I.e., only two step's is needed to choose the control law.

$$z = \phi(x) \tag{2.20}$$

$$x = \phi^{-1}(z) \tag{2.21}$$

A change of coordinates is introduced during the recursive design. The existence of an inverse transformation is guaranteed as the backstepping transformation is a global diffeomorphism.
Step 1: The first backstepping variable is chosen as:

$$z_1 = x_1 \tag{2.22}$$

The objective is to regulate the output y = x_1, i.e. the state x_2 is chosen as an *virtual control* input:

$$x_2 := \alpha_1 + z_2 \tag{2.23}$$

Where α_1 is the stabilizing function and z_2 is the new state variable. The system is written as:

$$\dot{z}_1 = f(z_1) + \alpha_1 + z_2 \tag{2.24}$$

z_2 is introduced in order to couple z_1 to the system in the nest step. Hence, z_2 is not considered any further in this step. A CLF for the z_1 system is chosen:

$$\begin{aligned} V_1 &= \frac{1}{2} z_1^2 \\ \dot{V}_1 &= z_1 z_2 \\ &= z_1(f(z_1 + \alpha_1) + z_1 z_2) \end{aligned} \tag{2.25}$$

Following, a stabilizing function, α_1, is chosen as a feedback linearizing controller in order to provide the necessary feedback for the z_1.

$$\alpha_1 = -f(z_1) - k_1 z_1 \tag{2.26}$$

Where $K_1 > 0$ is the feedback gain:

$$\dot{V}_1 = -k_1 z_1^2 + z_2 \tag{2.27}$$

$$\dot{z}_1 = -K_1 z_1 + z_2 \tag{2.28}$$

In figure 2.11 an illustration of the stabilizing function and the new state variable is shown.

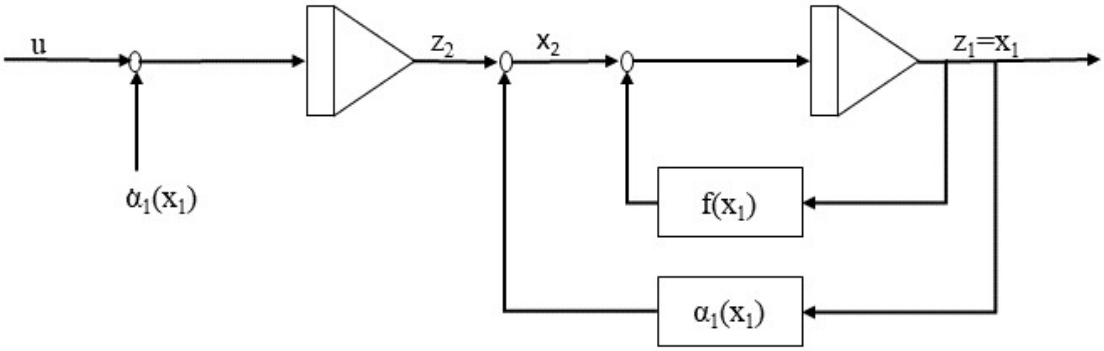


Figure 2.11: Stabilizing of the x_1 indicate stabilization of the stabilizing function α_1

Step 2:
In order to compute the z_2 dynamics equation 2.23 is differentiated:

$$\begin{aligned} \dot{z}_2 &= \dot{x}_2 - \dot{\alpha}_1 \\ &= u - \dot{\alpha}_1 \end{aligned} \tag{2.29}$$

Further, a CLF is chosen for the system:

$$\begin{aligned} V_2 &= V_1 + \frac{1}{2} z_2^2 \\ \dot{V}_2 &= \dot{V}_1 + \dot{z}_2 z_2 \\ &= -k_1 z_1^2 + z_2(u - \dot{\alpha}_1 + z_1) \end{aligned} \tag{2.30}$$

Accordingly, the control law is chosen as

$$u = \dot{\alpha}_1 - z_1 - k_2 z_2 \tag{2.31}$$

with $k_2 > 0$ yields:

$$\dot{V}_2 = -K_1 z_1^2 - k_2 z_2^2 < 0, \qquad \forall z_1 \neq 0, z_2 \neq 0 \tag{2.32}$$

Chapter 3

Case study I: Control system, CyberShip Enterprise I

In the last decade, the development of optical sensors and computer power have seen significant progress, resulting in motivation and opportunities concerning developments of autonomous surface vehicles. For autonomous surface vehicles, an important task for the control system is to steer the vehicle at the desired path. The maneuvering problem is a control problem combining path tracking and path following [22]. By implementing a maneuvering control algorithm, the desired path for the vehicle, and the desired motion along the path is separated into two different tasks. The development of safe and robust dynamic positioning, DP, is a significant stepping stone towards autonomous shipping. The objective for DP is to to keep the position or heading of the vessel relative to the seabed. This indicates that the DP system should be able to resist external forces and move from one place to another at low speed. Towards autonomous shipping, DP capability is of significant importance concerning auto-docking. In this case study, a maneuvering control design is developed for CyberShip Enterprise I. Following, a dynamic positioning system with a PID controller will be implemented. Simulation of the control system will be performed within Simulink, hardware-in-the-loop, and the Marine Cybernetics laboratory. Lastly, a discussion regarding the experimental results and comparison of a backstepping and PID controller design will be given.

3.1 Case description

The objective of this case study is to develop a maneuvering control algorithm with backstepping. In addition, the control system will be verified and tested with simulation in Simulink, hardware-in-the-loop, and model scale testing. The testing of the model will be performed in the Marine Cybernetics laboratory with the model ship CS Enterprise I. The experimental setup, vessel, and lab characteristic is described in Chapter 5. All necessary code are attached. The following tasks will be solved for:

1. Propose a control plant model for CS Enterprise I.
2. Develop thrust allocation.
3. Develop observer design.
4. Implementation of DP maneuvering control design.
5. Implementation of linear PID controller design.

3.2 Process plant model

A process plant model, PPM, is a model of the actual physical process and its primary purpose is to reflect the real dynamics of the ship [26]. The PPM is often called the simulation model and used for controller testing and verification [2]. The process plant model describes the real vessel dynamics accurately in 6 DOFs, and includes control inputs, sensor output, and environmental disturbances. The nonlinear low-frequency model for the vessel is described as following:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{C}_{A}(\boldsymbol{\nu_r})\boldsymbol{\nu_r} + \mathbf{D}(\kappa, \boldsymbol{\nu_r}) + \mu + \mathbf{G}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{3.1}$$

In the model $\boldsymbol{M\dot{\nu}}$ represents the generalized forces, where $\boldsymbol{M} \in \mathbb{R}^{6x6}$ is the system inertia matrix including added mass. The second term, $\boldsymbol{C(\nu)}_{RB}\nu \in \mathbb{R}^{6x6}$ and $\boldsymbol{C(\nu_r)}_{RB}\nu_r \in \mathbb{R}^{6x6}$, expresses the generalized Coriolis and centripetal forces. It follows that the first Coriolis and centripetal forces term are forces for the rigid body and the second term accounts for added mass. In the process plant model the relative velocity, $\boldsymbol{\nu}_r$, are used. It follows that $\boldsymbol{\dot{\nu}} \in \mathbb{R}^{6x6}$ is the vessel acceleration in body frame, and $\boldsymbol{\nu_r = \nu - \nu_c} \in \mathbb{R}^{6x6}$ the relative velocity. Where $\boldsymbol{\nu_c}$ is the current velocity in the body frame. The generalized damping and current forces are indicated by $\boldsymbol{D(\kappa, \nu_r)} \in \mathbb{R}^{6x6}$. The damping term consists of a linear second-order wave induces damping component and a non-linear component, where the non-linear damping term consists of drag coefficients found by model tests. μ represents the fluid memory effect, and the restoring matrix is expressed by $\boldsymbol{G(\eta)}$. $\boldsymbol{\tau}$ includes all external forces such as mooring, ice, thrusters, and the environment. For controller and observer design only the main physical properties of the system are necessary, which is a control plant model.

3.3 Control plant model

The difference between a PPM and the control plant model (CPM) is that the CPM is a simplified model of the real plant described in the PPM. However, it is essential that the control plant model describes the exact process as sufficiently as possible to reflect the real process. The mathematical model of CS Enterprise I is based on system identification done in previous Master's Thesis [24]. The model is valid for low-speed, and an approximation of the 6 DOF model is made. Only 3 DOF will be accounted for in the control design as the motions in roll (ϕ), pitch (θ) and heave (ω) have a minimal effect when designing a control system for low-speed. A 3 DOF model is valid for both stationkeeping and low-speed manoeuvring [7, p. 152]. The proposed control plant model is expressed in (5.1):

$$\begin{aligned} \boldsymbol{\dot{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{\nu} \\ \boldsymbol{M\dot{\nu}} &= \boldsymbol{-C(\nu)\nu - D(\nu)\nu + \tau} \end{aligned} \tag{3.2}$$

In the CPM for CS Enterprise I, the pose and velocity vectors are expressed as $\boldsymbol{\eta} \in \mathbb{R}^{3x3}$ and $\boldsymbol{\nu} \in \mathbb{R}^{3x3}$ in (3.3). For the pose vector, the position is expressed in (x, y) and the yaw angle or heading by ψ in the basin frame. For the velocity vector, surge and sway velocities are expressed by u and v, and r expresses the yaw rate in the CS Enterprise 1 vessel frame. $\boldsymbol{M\dot{\nu}}$ represents the generalized forces, where $\boldsymbol{M} \in \mathbb{R}^{3x3}$ in (3.4) is the system inertia matrix. The damping matrix is indicated by $\boldsymbol{D(\nu)} \in \mathbb{R}^{3x3}$. $\boldsymbol{M}$ and $\boldsymbol{D(\nu)}$ are dependent on the rigid body and hydrodynamic added mass parameters, as well as hydrodynamic damping parameters. $\boldsymbol{R(\psi)} \in \mathbb{R}^{3x3}$ is the rotation matrix, and $\boldsymbol{\tau} \in \mathbb{R}^{3x3}$ describes the thrust force and moments, where (X, Y) is the surge and sway force vector, and N is the yaw moment. Necessary coefficients, thrust force and moments ranges are listed in Chapter 5.

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \qquad \boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ r \end{bmatrix} \qquad \boldsymbol{\tau} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} \qquad \boldsymbol{R}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} = \boldsymbol{M}^T > 0 \tag{3.4}$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} d_{11}(u) & 0 & 0 \\ 0 & d_{22}(v,r) & d_{23}(v,r) \\ 0 & d_{32}(v,r) & d_{33}(v,r) \end{bmatrix} \tag{3.5}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & (-mx_g + Y_{\dot{r}})r + (-m + Y_{\dot{v}})v \\ 0 & 0 & (m - X_{\dot{u}})u \\ (mx_g - Y_{\dot{r}})r + (m - Y_{\dot{v}})v & (-m + X_{\dot{u}})u & 0 \end{bmatrix} \tag{3.6}$$

3.4 Thruster allocation

$\boldsymbol{\tau}$ is the control vector of forces and moment in surge, sway and yaw which will be provided by the thrust allocation. Thrust allocation is allocating generalized control forces to the actuators by control inputs $\boldsymbol{u}$. In this thesis, a 3 DOF model is considered. Thus the thrust load vector can be reduced from (3.7) to (3.8).

$$\boldsymbol{\tau} = \begin{bmatrix} f \\ l \times f \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \\ l_y f_z - l_z f_y \\ l_z f_x - l_x f_z \\ l_x f_y - l_y f_x \end{bmatrix} \tag{3.7}$$

$$\boldsymbol{\tau} = \begin{bmatrix} f_x \\ f_y \\ l_x f_y - l_y f_x \end{bmatrix} \tag{3.8}$$

In figure 3.1, an illustration of the model ship is presented. The vessel is equipped with two Voith Schneider Propellers aft and one bow thrusters in the front. The Voith Schneider propellers can take azimuth angle values in the range of $[-\pi, \pi]$, while the bow thruster is not able to take any angle. The range of the input values for $\boldsymbol{u}$ is thus [-1, 1] for the bow thruster and [0, 1] for the VSP thruster. Hence, the VSP thrusters are not able to produce negative force without flipping the thrust angle α to 180 degrees. Assuming linear mapping $F_i = k_i u_i$ for $i = 1, 2, 3$, equation (3.9), combined with knowing the value of $F_{max,VSP}$, $F_{max,BT}$ and the dimension of the vessel it can be concluded that the gains must be $k_1 = 1.03$, $k_2 = 1.03$ and $k_3 = 2.629$.

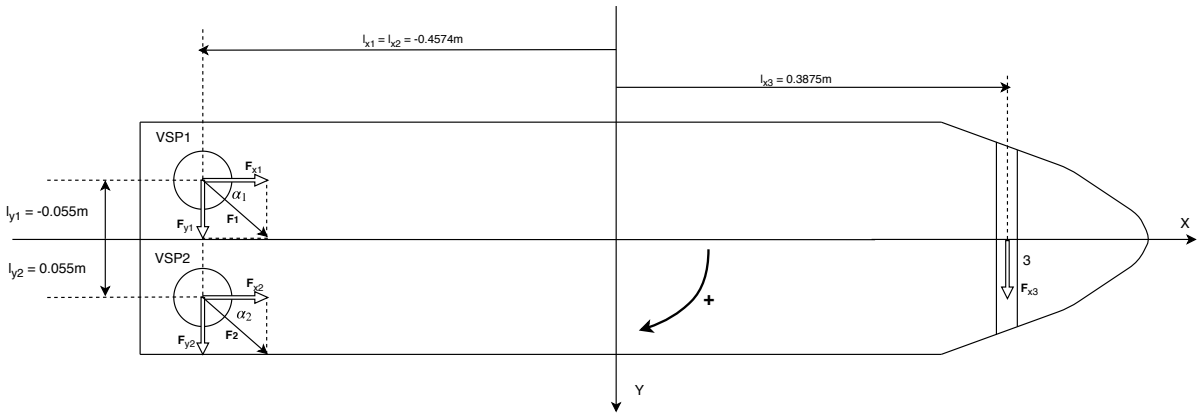


Figure 3.1: Thruster placement on CS Enterprise I

	l_x	l_y	F_{max}
BT	0.3877	0	2.629[N]
VSP1	-0.4574	-0.055	1.03[N]
VSP2	-0.4574	0.055	1.03[N]

Table 3.1: Length parameters of the model ship in meters

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u} \tag{3.9}$$

In $\boldsymbol{B}(\boldsymbol{\alpha})$ the rows represent the degree of freedom, while the columns express the thrusters and decoupling. In this case only surge, sway and yaw are accounted for due to the reduced control plant model. Further, it is assumed that $\boldsymbol{BK}$ is non-singular and that the azimuth angles α_i are fixed in constant directions. Thus, $\boldsymbol{B}(\boldsymbol{\alpha}) = \boldsymbol{B}$ and the relation can be expressed as:

$$\boldsymbol{u} = (\boldsymbol{BK})^{-1}\boldsymbol{\tau_{cmd}} \tag{3.10}$$

Following, the thrust vectors are expressed in rectangular coordinates. Given the force vector and the gains $\boldsymbol{K}$ the input angles can be calculated. The thrust configuration matrix for rotable actuators can be expressed as in (3.11). By using $\boldsymbol{f} = \begin{bmatrix} f_{1,x} & f_{1,y} & f_{2,x} & f_{2,y} & f_3 \end{bmatrix}^T$ and the relation $\boldsymbol{\tau} = \boldsymbol{Bf}$ - $\boldsymbol{B}$ can be written as:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ -l_{y1} & l_{x1} & -l_{y2} & l_{x2} & l_{x3} \end{bmatrix} \tag{3.11}$$

$$\alpha_i = arctan2\left(\frac{f_{i,y}}{f_{i,x}}\right) \qquad\qquad u_i = \frac{f_{i,x}}{k_i cos(\alpha_i)} \tag{3.12}$$

In order to avoid an infinite amount of solutions, due to an undetermined linear set of equations, the thrust allocation for f_{cmd} is solved:

$$\begin{aligned} &\min_{\boldsymbol{f_{cmd}}} && \boldsymbol{f_{cmd}}^T \boldsymbol{f_{cmd}} \\ &\text{subject to} && \boldsymbol{B f_{cmd}} = \boldsymbol{\tau_{cmd}} \end{aligned}$$

Which results in (3.13):

$$\boldsymbol{f_{cmd}} = \boldsymbol{B}^{\dagger}\boldsymbol{\tau_{cmd}} \tag{3.13}$$

The solution of the least-squares optimization problem is expressed in (3.13), where $\boldsymbol{B}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{B}$. The explicit solution of the least-squares optimization problem is found in Appendix A.1. Further, the generalized control forces $\boldsymbol{\tau_{cmd,2}} = \begin{bmatrix} 2 & 0 & 0.5 \end{bmatrix}^T$ are used. The updated values for τ_{cmd} are accordingly obtained from `MATLAB`. Hence, the angle and input for each thruster can be calucated with equation (3.12).

$$\boldsymbol{f_{cmd,1}} = \begin{bmatrix} 0.5771 \\ -0.0616 \\ 0.4229 \\ -0.0616 \\ 1.1231 \end{bmatrix} \qquad \boldsymbol{f_{cmd,2}} = \begin{bmatrix} 1.0571 \\ -0.2922 \\ 0.9429 \\ -0.2922 \\ 0.5844 \end{bmatrix} \tag{3.14}$$

	$\alpha_{cmd,1}$	$u_{cmd,1}$
VSP1	-0.1063	0.5635
VSP2	-0.1445	0.4149
BT	0	0.4272

	$\alpha_{cmd,2}$	$u_{cmd,2}$
VSP1	-0.2697	1.0648
VSP2	-0.3005	0.9584
BT	0	0.2223

Table 3.2: The set (α_i, u_i) for each thruster, corresponding to $\boldsymbol{f_{cmd,1}}$ and $\boldsymbol{f_{cmd,1}}$

3.5 Observer design

The low speed dynamics of the model ship CyberShip Enterprise I is given by (3.15) [24]:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} &= -\boldsymbol{D}\boldsymbol{\nu} + \boldsymbol{R}(\psi)^T\boldsymbol{b} + \boldsymbol{\tau} \end{aligned} \tag{3.15}$$

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{3.16}$$

A drawback with Kalman filter and the backstepping observer is that a relatively large number of parameters must be determined through experimental testing. The DP observer used in this case study is based on the *Observer for simplified DP model: Design and proof* by Værnø and Skjetne [29]. The observer is designed by copying the plant dynamics and adding injection terms. The goal is to find the diagonal matrices $\boldsymbol{L_1}, \boldsymbol{L_2}$ and $\boldsymbol{L_3}$, the injection gains, in order to achieve global stability.

The position and heading error are defined as $\bar{\boldsymbol{\eta}} = \boldsymbol{\eta} - \hat{\boldsymbol{\eta}}$, the velocity error $\bar{\boldsymbol{\nu}} = \boldsymbol{\nu} - \hat{\boldsymbol{\nu}}$ and the bias error is defined as $\bar{\boldsymbol{b}} = \boldsymbol{b} - \hat{\boldsymbol{b}}$. Assuming that only the measurement from $\boldsymbol{\eta}$ is available. In addition the diagonal matrices $\boldsymbol{L_1}, \boldsymbol{L_2}$ and $\boldsymbol{L_3}$ expresses the injection gains. The injection gains are found by tuning. Following the observer (3.17) and closed loop error dynamics (3.18) are obtained. Full deduction is given in Appendix 3.18.

$$\begin{aligned} \dot{\hat{\boldsymbol{\eta}}} &= \boldsymbol{R}(\psi)\hat{\boldsymbol{\nu}} + \boldsymbol{L_1}\bar{\boldsymbol{\eta}} \\ \boldsymbol{M}\dot{\hat{\boldsymbol{\nu}}} &= -\boldsymbol{D}\hat{\boldsymbol{\nu}} + \boldsymbol{R}^T(\psi)\hat{\boldsymbol{b}} + \boldsymbol{\tau} + \boldsymbol{R}^T(\psi)\boldsymbol{L_2}\bar{\boldsymbol{\eta}} \\ \dot{\hat{\boldsymbol{b}}} &= \boldsymbol{L_3}\bar{\boldsymbol{\eta}} \end{aligned} \tag{3.17}$$

$$
\begin{aligned}
\dot{\bar{\boldsymbol{\eta}}} &= \boldsymbol{R}(\psi)\bar{\boldsymbol{\nu}} - \boldsymbol{L_1}\bar{\boldsymbol{\eta}} \\
\boldsymbol{M}\dot{\bar{\boldsymbol{\nu}}} &= -\boldsymbol{D}\bar{\boldsymbol{\nu}} + \boldsymbol{R}(\psi)^T\bar{\boldsymbol{b}} - \boldsymbol{R}(\psi)^T\boldsymbol{L_2}\bar{\boldsymbol{\eta}} \\
\dot{\bar{\boldsymbol{b}}} &= -\boldsymbol{L_3}\bar{\boldsymbol{\eta}}
\end{aligned}
\tag{3.18}
$$

In addition Værnø and Skjente state that the equilibrium point $(\bar{\boldsymbol{\eta}}, \bar{\boldsymbol{\nu}}, \bar{\boldsymbol{b}}) = (0, 0, 0)$ of equation (3.18) is UGAS if the system renders the following conditions [29]:

- $\boldsymbol{D} + \boldsymbol{D}^T \geq 0$.
- $\boldsymbol{L_1}$, $\boldsymbol{L_2}$ and $\boldsymbol{L_3}$ are symmetric and positive definite.
- $\boldsymbol{L_1 L_3} = \boldsymbol{L_3 L_1}$.
- $\boldsymbol{L_1 L_2} + \boldsymbol{L_2 L_1} - 2\boldsymbol{L_3}$ is symmetric and positive definite.
- ${\boldsymbol{L_3}}^{-1}\boldsymbol{L_1} - {\boldsymbol{L_2}}^{-1}$ is symmetric and positive definite.

Accordingly, the injection gains are found by HIL - testing and verification of the mentioned criteria. Hence, the observer will ensure stable if the matrix gains are chosen as (3.19).

$$
\boldsymbol{L_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 50 \end{bmatrix} \quad \boldsymbol{L_2} = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 50 \end{bmatrix} \quad \boldsymbol{L_3} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{3.19}
$$

3.6 Controller design

In this Section two different controller designs will be implemented to the control plant model for CS Enterprise I. Firstly, the implementation of a backstepping controller with DP maneuvering will be presented, followed with associated tuning method and simulations in Simulink, HIL and model scale testing. Secondly, an implementation of a linear PID controller will be reviewed followed by tuning method for the controller gains and simulation in Simulink, HIL, and model scale testing.

3.6.1 Backstepping

Implementation of DP maneuvering control design

In this Section, the maneuvering control law will be implemented to solve the maneuvering object for the vessel model.

Step 1: In (3.20) the position and heading error vector is expressed in body frame. The control Lyapunov function, CLF, chosen to solve the manoeuvring problem expressed in (3.22). V_1 is differentiated in appendix A.3.1 and results in (3.23).

$$\boldsymbol{z}_1 := \boldsymbol{R}(\psi)^T(\boldsymbol{\eta} - \boldsymbol{\eta}_d(s)) \tag{3.20}$$

$$\boldsymbol{z}_2 := \boldsymbol{\nu} - \boldsymbol{\alpha}_1(\boldsymbol{\eta}, s, t) \tag{3.21}$$

$$V_1(\boldsymbol{\eta}, s) = \frac{1}{2}\boldsymbol{z}_1^T\boldsymbol{z}_1 = \frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\eta}_d(s))^T(\boldsymbol{\eta} - \boldsymbol{\eta}_d(s)) \tag{3.22}$$

$$\dot{V}_1 = \boldsymbol{z}_1^T(\boldsymbol{\nu} - \boldsymbol{R}(\psi)^T\boldsymbol{\eta}_d^s(s)\dot{s}) \tag{3.23}$$

Assuming $\nu = \alpha_1$, α is chosen in the following manner:

$$\alpha_1(\eta, s, t) = -\boldsymbol{K_{p1}}z_1 + \boldsymbol{R(\psi)}^T\boldsymbol{\eta}_d^s(s)U_s(s,t), \qquad \boldsymbol{K}_p = \boldsymbol{K}_p^T > 0 \tag{3.24}$$

By inserting $\alpha_1(\eta, s, t)$ (3.24) into equation (3.23), the following equation is obtained. This is derived in A.3.2:

$$\dot{V}_1 = -\boldsymbol{z}_1^T\boldsymbol{K}_p\boldsymbol{z}_1 - V_1^s(U_s(s,t) - \dot{s}) \tag{3.25}$$

$$\dot{V}_1 \leq -\lambda_{min}\boldsymbol{K}_p|\boldsymbol{z}_1|^2 - V_1^s(U_s(s,t) - \dot{s}) \tag{3.26}$$

It follows that corresponding terms for the straight line parameterization are inserted for $\boldsymbol{\eta}_d$ and $\boldsymbol{\eta}_d^s$, equation (2.6) and (2.7). Hence, $V_1^s(\boldsymbol{\eta}, s)$ is derived by differentiating (3.22) with respect to s.

$$V_1^s(\boldsymbol{\eta}, s) = -\boldsymbol{z}^T\boldsymbol{R}(\psi)^T\boldsymbol{\eta}_d^s = -(\boldsymbol{\eta} - \boldsymbol{\eta}_d)^T\boldsymbol{\eta}_d^s \tag{3.27}$$

In order to to account for varying magnitude of $V(\boldsymbol{\eta}, s)_1^s$ along the parameterization path, a normalizing modification is applied. This is due to a possible nonlinear parameterization. Consequently, the modified gradient update law (3.28) is implemented into equation (3.24). As a result $\boldsymbol{z}_2$ and $\boldsymbol{\alpha}_1(\boldsymbol{\eta}, s, t)$ are added into $\dot{V}_1$. $\dot{V}_1$ is expressed in (3.29), full derivation is given in A.3.4.

$$\dot{s} = U_s(s,t) - \frac{\mu}{|\boldsymbol{\eta}_d^s(s)|}V_1^s(\boldsymbol{\eta}, s), \mu \leq 0 \tag{3.28}$$

$$\dot{V}_1 = \boldsymbol{z}_1^T \boldsymbol{z}_2 - \boldsymbol{z}_1^T \boldsymbol{K}_{p1} \boldsymbol{z}_1 - \frac{\mu}{|\boldsymbol{\eta}_d^s(s)|} V_1^s(\boldsymbol{\eta}, s)^2 \tag{3.29}$$

Differentiating α_1:

$$\dot{\alpha}_1(\boldsymbol{\eta}, s, t) = -\boldsymbol{K}_{p1} \dot{\boldsymbol{z}}_1 + (\boldsymbol{R}(\psi)^T \boldsymbol{\eta}_d^s(s) U_s(s,t)) \tag{3.30}$$

$$= -\boldsymbol{K}_{p1} \dot{z}_1 + \dot{\boldsymbol{R}}(\psi)^T \boldsymbol{\eta}_d^s(s) U_s(s,t) + \boldsymbol{R}(\psi)^T \dot{\boldsymbol{\eta}}_d^s U_s(s,t) + \boldsymbol{R}(\psi)^T \boldsymbol{\eta}_d^s \dot{U}_s(s,t)$$

$$= -\boldsymbol{K}_{p1}(\boldsymbol{R}(\psi)^T \tilde{\boldsymbol{\eta}}(s) - \boldsymbol{R}(\psi)^T S(r) \tilde{\boldsymbol{\eta}}(s)) - \boldsymbol{R}(\psi)^T S(r) \tilde{\boldsymbol{\eta}}_d^s(s) U(s,t)_s + \boldsymbol{R}(\psi)^T \boldsymbol{\eta}_d^{2s}(s) \dot{s} U_s(s,t) + \boldsymbol{R}(\psi)^T \boldsymbol{\eta}_d^s(s) U_s^s(s,t) \dot{s}$$

Step 2: Differentiating z_2:

$$\dot{\boldsymbol{z}}_2 = \dot{\boldsymbol{\nu}} - \dot{\alpha}(\boldsymbol{\eta}, s, t) \tag{3.31}$$

$$\boldsymbol{M} \dot{\boldsymbol{z}}_2 = \boldsymbol{M} \dot{\boldsymbol{\nu}} - \boldsymbol{M} \dot{\alpha}_1 \tag{3.32}$$

$$\boldsymbol{M} \dot{\boldsymbol{z}}_2 = -\boldsymbol{D} \boldsymbol{\nu} + \boldsymbol{R}(\psi)^T \boldsymbol{b} + \boldsymbol{\tau} - \boldsymbol{M} \dot{\alpha}_1$$

$$\dot{\boldsymbol{z}}_2 = \boldsymbol{M}^{-1}(-\boldsymbol{D} \boldsymbol{\nu} + \boldsymbol{R}(\psi)^T \boldsymbol{b} + \boldsymbol{\tau}) - \dot{\alpha}_1$$

The second step of the CLF is expressed as:

$$V_2(\boldsymbol{\nu}, \boldsymbol{\eta}, s, t) = V_1(\boldsymbol{\eta}, s, t) + \frac{1}{2} \boldsymbol{z}_2 \boldsymbol{M} \boldsymbol{z}_2 \tag{3.33}$$

$$\dot{V}_2(\boldsymbol{\nu}, \boldsymbol{\eta}, s, t) = \dot{V}_1(\boldsymbol{\eta}, s, t) + \frac{1}{2} \dot{\boldsymbol{z}}_2^T \boldsymbol{M} \boldsymbol{z}_2 + \frac{1}{2} \boldsymbol{z}_2^T \boldsymbol{M} \dot{\boldsymbol{z}}_2 \tag{3.34}$$

$$\dot{V}_2 = \boldsymbol{z}_1^T \boldsymbol{z}_2 - \boldsymbol{z}_1^T \boldsymbol{K}_p \boldsymbol{z}_1 - \boldsymbol{z}_1^T \frac{\mu}{|\boldsymbol{\eta}_d^s(s)|} V_1^s(\boldsymbol{\eta}, s) + \boldsymbol{z}_2^T(-\boldsymbol{D} \boldsymbol{\nu} + \boldsymbol{R}(\psi)^T \boldsymbol{b} + \boldsymbol{\tau} - \boldsymbol{M} \dot{\alpha}_1)$$

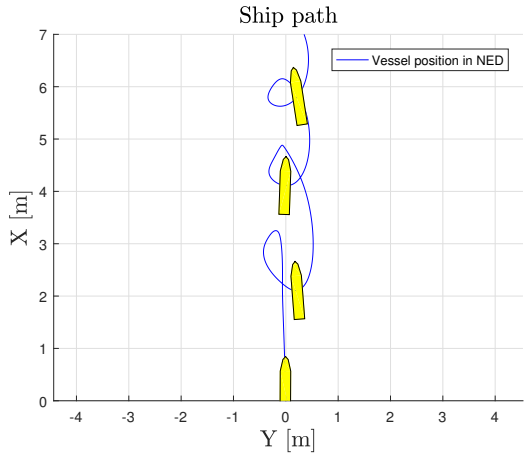
Following, the the control law for $\boldsymbol{\tau}$ which renders $\dot{V}_2$ negative definite is chosen and implemented into the control system:

$$\boldsymbol{\tau} = -\boldsymbol{z}_1 + \boldsymbol{D} \boldsymbol{\nu} - \boldsymbol{R}(\psi)^T \boldsymbol{b} + \boldsymbol{M} \dot{\alpha} - \boldsymbol{K}_{p2} \boldsymbol{z}_1 \tag{3.35}$$

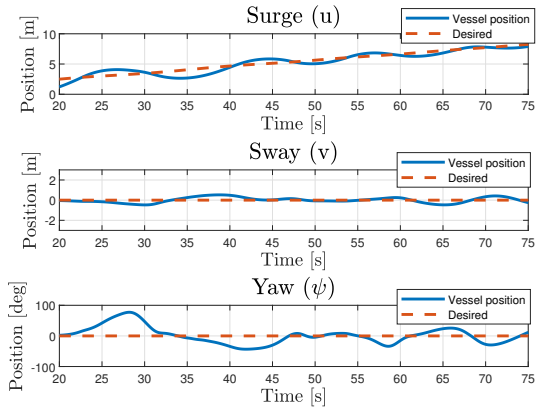
$$\dot{V}_2 = -\boldsymbol{z}_1^T \boldsymbol{K}_p \boldsymbol{z}_1 - \boldsymbol{z}_2^T \boldsymbol{K}_{p2} \boldsymbol{z}_2 < 0$$

Tuning of the manoeuvring control design

To obtain a stable and robust maneuvering control algorithm, the optimal controller gains are established from a simulation in Simulink and hardware-in-the-loop. Firstly, simulation in Simulink is performed, followed by simulation in HIL. The results in this Section are retrieved from the simulation in HIL. K_{p1} and K_{p2} are the controller gains, and μ is connected to the modified gradient update law, which is implemented into the control law. μ is linearly influencing the contribution of the $V_1^s(\eta, s)$ in (3.28). In addition, it is stated that $\mu \leq 0$. Consequently, the first stable result obtained is with $\mu = 0.01$ and the controller gains: $K_{p1} = diag\{\begin{bmatrix}1 & 1 & 1\end{bmatrix}\}$, $K_{p2} = diag\{\begin{bmatrix}1 & 1 & 1\end{bmatrix}\}$. It is noticed that bigger μ drives to faster convergence to the path. Even so, it is noticed that the ship is not able to handle bigger μ than 0.01. This suggests that the vessel is not able to handle the sharp turns developed from fast convergence. In figure 3.3a the results form the latter controller gains are illustrated. The plots display that the controller overshoots. In figure 3.3b it is shown that the vessel oscillates around the desired path with approximately 1 meter in surge and 0.5 meters in sway. After 27 seconds it is shown in the yaw position plot, that the vessel is given a yaw angle of approximately 70 °. Accordingly, the vessel is compensating for the overshoot in surge with a heading angle of 70 °. This is illustrated in the ship plot at [3 0 0]. All plots obtained are started from the interval where the simulation is started. That is if the start value at the time axis is 15 seconds, the simulation started after 15 seconds.



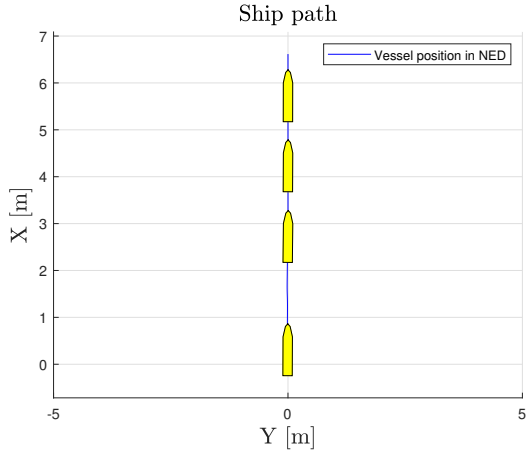
(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



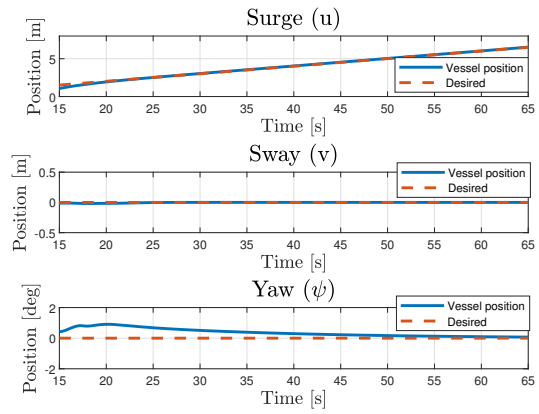
(b) The vessel position in surge, sway and yaw.

Figure 3.2: Backstepping: Tuning of μ, K_{p1} and K_{p2}

The previous controller gains is not optimal, however $\mu = 0.01$ is tested further. The following controller gains are tested: $K_{p1} = diag\{\begin{bmatrix}0.1 & 0.1 & 0.01\end{bmatrix}\}$ and $K_{p2} = diag\{\begin{bmatrix}20 & 20 & 20\end{bmatrix}\}$. In figure 3.3b the movement of the vessel is illustrated. It is shown that the vessel movement is converging to the desire path perfectly. The yaw angle have a small overshoot of 1°, which is insignificant. In other words, the vessel follows the desired path perfectly and the optimal gains are obtained.



(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



(b) The vessel position in surge, sway and yaw.

Figure 3.3: Backstepping: Tuning of μ, K_{p1} and K_{p2}

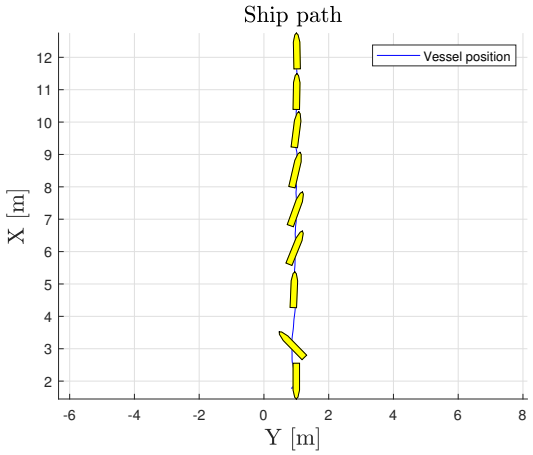
Simulation setup

The maneuvering control algorithm for the kinematic model is tested in Simulink, hardware-in-the-loop, and the Marine Cybernetics laboratory. A description of the equipment used during HIL and laboratory testing is found in Chapter 5. The gains established during tuning are used in all experiments and simulations. First, a presentation of the results from the simulation in Simulink will be given followed by the results from the simulation in HIL. Lastly, the results from model-scale testing in the Marine Cybernetics laboratory will be presented. All plots obtained

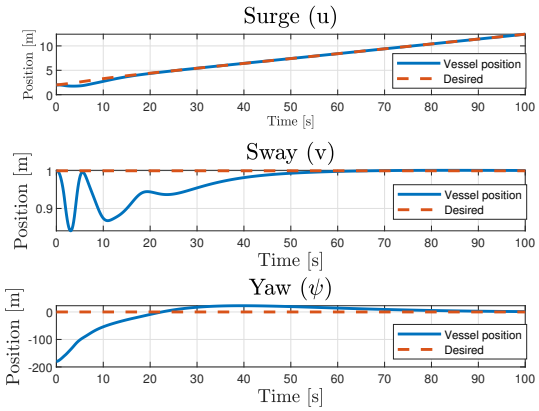
are started from the interval where the simulation is started. That is, if the start value at the time axis is 15 seconds, the simulation started after 15 seconds.

Simulation in Simulink

During simulation in Simulink the initial position of the straight line parameterization is x = 2 [m], y = 1 [m] and $\psi = -\pi$ [rad]. The initial position is chosen in order to show that the maneuvering can be started for an initial position, in addition, a delay of 10 seconds is implemented. In figure 3.4a the ship position of the vessel is illustrated in the NED reference frame and figure 3.4b display the vessel position in surge, sway and yaw. In the time domain 0 - 20 seconds in figure 3.4b, it is displayed that the heading is decreasing from $-180°$ to $0°$. From the ship path plot, it is illustrated that the ship is turning $180°$ in the first seconds of simulation, as expected. Hence, the vessel is succeeding in maneuvering the straight line.



(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction

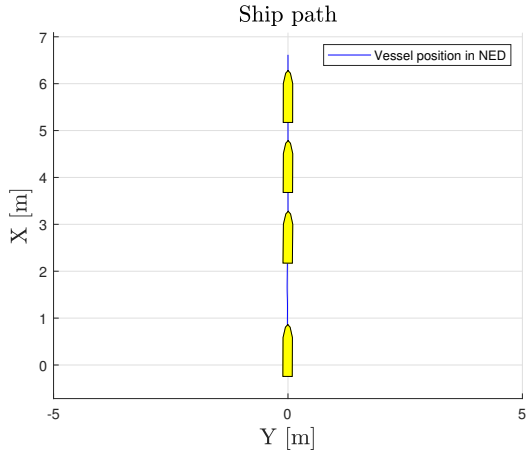


(b) The vessel position in surge, sway and yaw

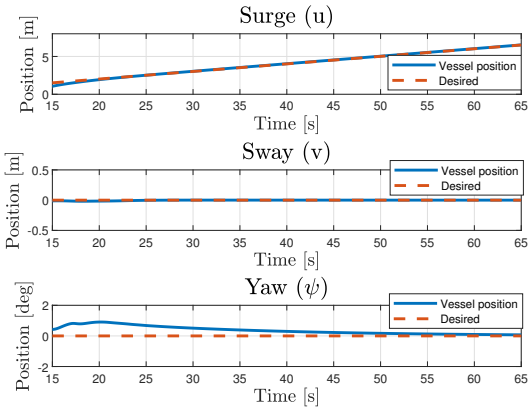
Figure 3.4: Manoeuvring control: Simulation in Simulink

Hardware-in-the-loop simulation

The results from hardware-in-the-loop simulation are displayed in figure 3.5. The ship path in NED is illustrated in figure 3.5a and the ship movement in surge, sway and yaw are displayed in figure 3.5b. From the plots, it is indicated that the maneuvering control algorithm is succeeding in following the straight line parameterized in surge, sway and yaw.



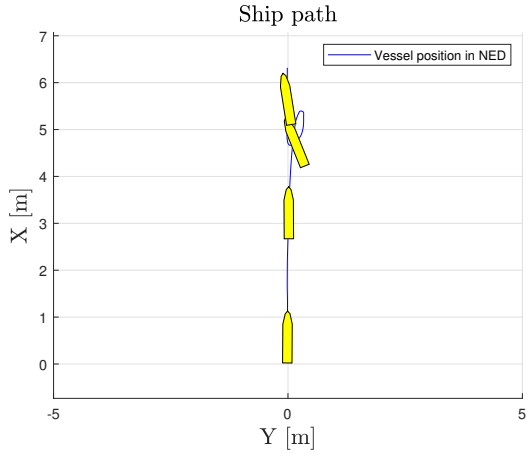
(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



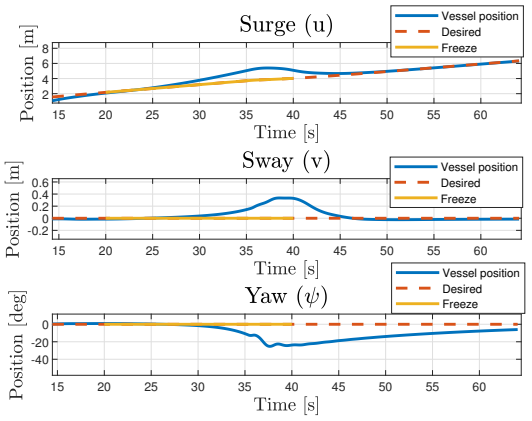
(b) The vessel position in surge, sway and yaw

Figure 3.5: Manoeuvring control: Hardware-in-the-loop testing

In addition, the control system should manage to handle dead reckoning. Dead reckoning is tested by simulating a loss of signal. In figure 3.6 the ship path and movement is displayed. In figure 3.6b loss of signal is indicated with the yellow line. Accordingly, loss of signal is simulated in the time domain 20 - 40 seconds. The results demonstrate that the estimator manages to follow the true position for roughly 5 seconds in surge, sway and yaw. The signal is retrieved after 40 seconds, and the desired position in surge and sway are recovered 5 seconds later. However, the biggest effect of the signal freeze is on the heading. Consequently, the yaw angle is nearly recovered after 60 seconds.



(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



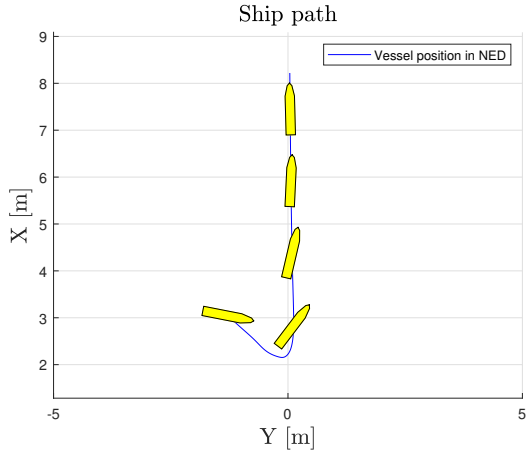
(b) The vessel position in surge, sway and yaw

Figure 3.6: Manoeuvring control: Hardware-in-the-loop testing with dead reckoning

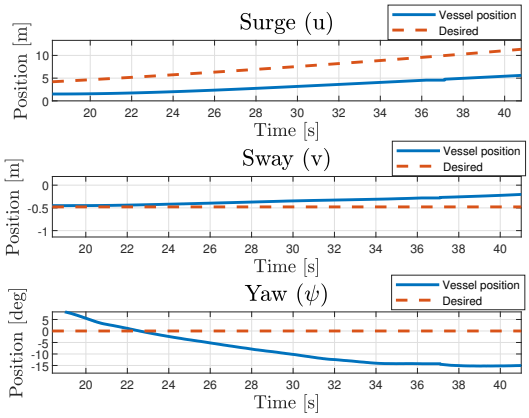
Model scale testing in laboratory

The model scale testing is performed in the Marine Cybernetics lab. The path of the model ship is illustrated in figure 3.7a and the movement in surge, sway and yaw are shown in figure 3.7b. In the latter plot, it is seen that the vessel drifts 0.5 meters in sway during the time domain 15

- 40 seconds. In addition to the drift in sway, there is a 3 meter offset between the desired path in surge and the actual position.



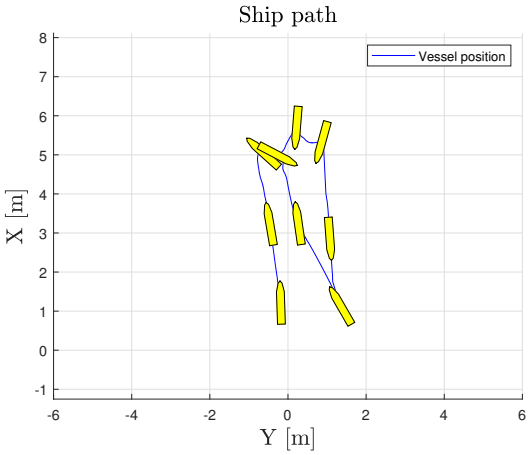
(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



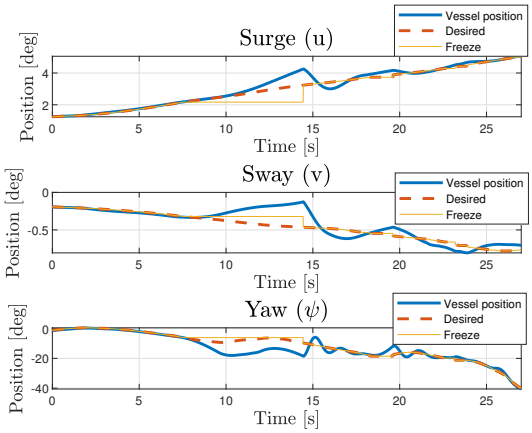
(b) The vessel position in surge, sway and yaw

Figure 3.7: Manoeuvring control: model scale testing in laboratory

Similarly to the HIL testing, dead reckoning is tested in the laboratory. In figure 3.8 the vessel position in NED and the movement in surge, sway and yaw are illustrated. The loss of the signal is simulated in the time domain 7 - 14 seconds, indicated by the yellow line. From figure 3.8b it is illustrated that the vessel position is maintained for approximately 2 seconds in surge and almost no time in sway and yaw. Following, the ship overcompensates when the signal is recovered. The ship path illustrates the movement of the model ship. After 7 seconds, the ship model are at position [2 -0.5 0].



(a) Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



(b) The vessel position in surge, sway and yaw

Figure 3.8: Manoeuvring control: model scale testing in laboratory

3.6.2 PID

Implementation of linear PID

The proportional-integral-derivative control is a linear controller well used in the industry. This is mainly due to the PID controllers simplicity, applicability, and functionality. In Section 2.6.4 a brief explanation of the controller is given. The PID controller continuously calculates an error $\boldsymbol{e}(t)$ between the desired setpoint and the current position of the vessel. The desired setpoint, $\boldsymbol{\eta} = [\boldsymbol{N}, \boldsymbol{E}, \psi]$, and measured value is described in NED reference frame. However, the control plant model is expecting its input in the BODY reference frame. For this reason, the setpoint and measured input value are rotated from NED to the BODY reference frame with a rotation matrix. $\boldsymbol{G_P}$, $\boldsymbol{G_I}$ and $\boldsymbol{G_D}$ describes the controller gains for the proportional, integral and derivative term. The control inputs implemented into the control system are based on Section 2.6.4. The following control inputs are implemented into the control system:

$$\boldsymbol{\tau_{PD}} = -\boldsymbol{G_P e_2}(t) + -\boldsymbol{G_D e_1}(t) \tag{3.36}$$

$$\dot{\boldsymbol{\tau}}_{\boldsymbol{i}} = -\boldsymbol{G_I e_2}(t) \tag{3.37}$$

$$\boldsymbol{\tau}_{tot} = \boldsymbol{\tau_{PD}} + \boldsymbol{\tau_I} \tag{3.38}$$

$$\boldsymbol{e_2} = \boldsymbol{R}^T(\psi_d)[\hat{\boldsymbol{\eta}} - \boldsymbol{\eta_d}] \tag{3.39}$$

$$\boldsymbol{e_1} = \dot{\boldsymbol{e_2}} \tag{3.40}$$

Reference model

A reference model is implemented in order to smoothen the step signal, i.e. desired setpoint, sent into the controller. The reference model generates a trajectory that suits the dynamics of the vessel. The reference model creates this trajectory by low-pass filtering the input step. Following, the filtered input is considered in the mass, damper, and spring system. To achieve stability and proper tracking of the signal, the reference model has to be slower than the dynamics of the control system. Consequently, the bandwidth of the reference model should be lower than the bandwidth of the control system. The reference model implemented is illustrated in figure 3.9 and based on the reference model presented in [7].

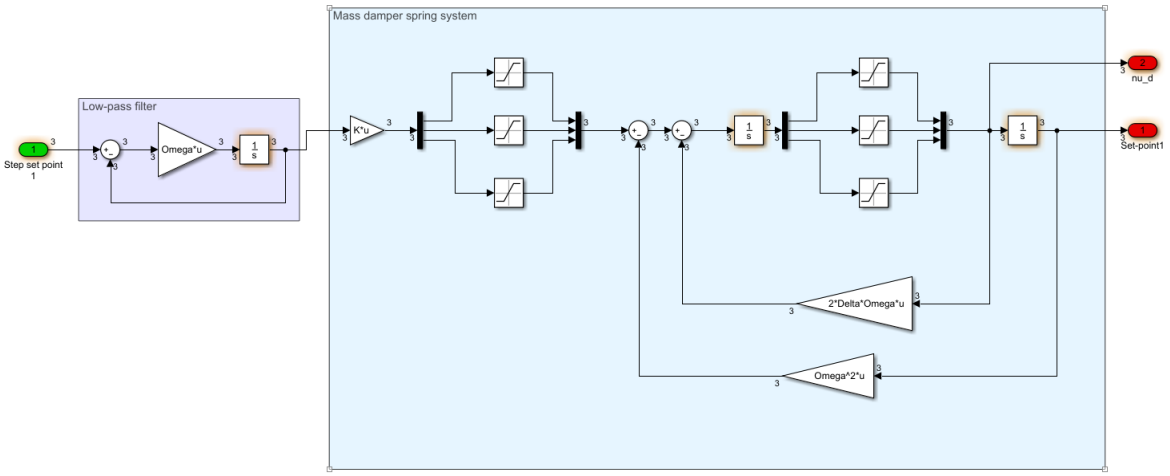


Figure 3.9: Reference model implemented [7]

For this model the relative damping ratios and natural frequencies are given by equation (3.41) and (3.42). Where $\Delta > 0$ and $\Omega > 0$.

$$\boldsymbol{\Delta} = diag\{\zeta_1, \zeta_2, ..., \zeta_n\} \tag{3.41}$$

$$\boldsymbol{\Omega} = diag\{\omega n1, \omega n2, ..., \omega n_n\} \tag{3.42}$$

In the implemented model $\boldsymbol{\Delta} = diag\{\begin{bmatrix}1 & 1 & 1\end{bmatrix}\}$ and $\boldsymbol{\Omega} = diag\{\begin{bmatrix}0.5 & 0.5 & 0.5\end{bmatrix}\}$

Tuning of controller gains for PID

The process of determining the optimal magnitude for the P, I and D gains to obtain the desired response of the controller, can be challenging and time-consuming. Several different methods can be utilized in order to acquire the ideal response for a PID controller. Common methods are:

1. Guess and check.
2. Ziegler Nichols method
3. PID control with acceleration feedback [7]

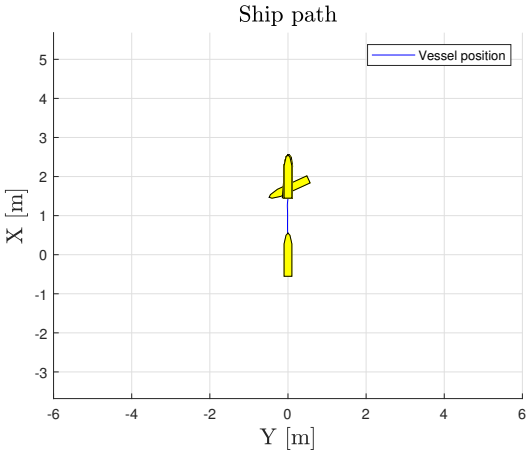
Method 3, *PID control with acceleration feedback*, from [7] is performed for the PID controller. Since acceleration feedback is not used in this controller, $\boldsymbol{K_m} = 0$. Following, there is no mooring system and therefore, no stiffness. Hence, the linear restoring matrix $\boldsymbol{G_{3x3}} = 0$ for surge, sway, and yaw in DP. The damping coefficient $\boldsymbol{d}$ is the damping in surge, sway and yaw. $\boldsymbol{M}$ is the total mass, which includes rigid body mass and added mass of the vessel. The equation reduces to the following:

$$\mathbf{G_p} = \mathbf{M}\boldsymbol{\omega}_n^2 \tag{3.43}$$

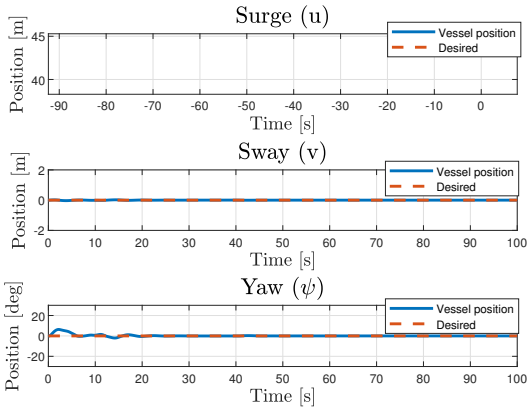
$$\mathbf{G_d} = 2\zeta\boldsymbol{\omega}_n\mathbf{M} - \mathbf{d} \tag{3.44}$$

$$\mathbf{G_i} = \frac{\boldsymbol{\omega}_n}{10}\mathbf{G_p} \tag{3.45}$$

To obtain a stable and robust PID algorithm, the optimal controller gains are established from the simulations in Simulink and hardware-in-the-loop. Firstly, simulation in Simulink is performed, followed by simulation in HIL. The objective with the tuning method is to determine the optimal ω_n and ζ, consequently optimal $\boldsymbol{G_p}$, $\boldsymbol{G_d}$ and $\boldsymbol{G_i}$ shall be achieved from the equations in (3.48). In the simulation in Simulink, the critically damped system is obtained with $\zeta = diag\{[1, 1, 1]\}$. Following, the optimal response of the system was obtained with $\omega_n = diag\{[5, 15, 3]\}$. The results of the latter gains are illustrated in figure 3.10. All plots obtained are started from the interval where the simulation is started. That is if the start value at the time axis is 15 seconds, the simulation started after 15 seconds.



(a) [PID: Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



(b) PID: The vessel position in surge, sway and yaw.

Figure 3.10: PID: Simulation in Simulink

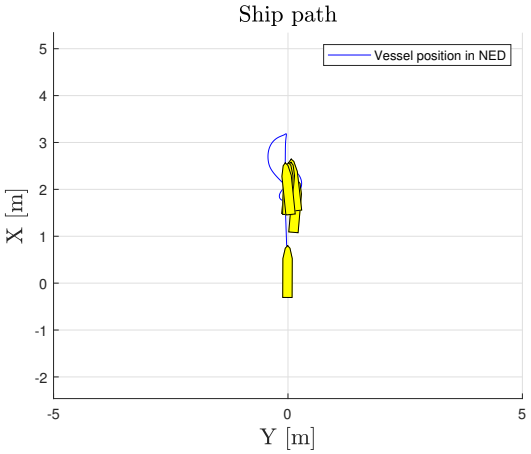
In figure 3.11 $\zeta = diag\{[1,1,1])\}$ and $\omega_n = diag\{[5,15,3]\}$ are used. However, in order to improve the response of the controller gains are adjusted to the following:

$$\mathbf{G_p} = \mathbf{M}\boldsymbol{\omega}_n^2 \cdot 1.5 \tag{3.46}$$

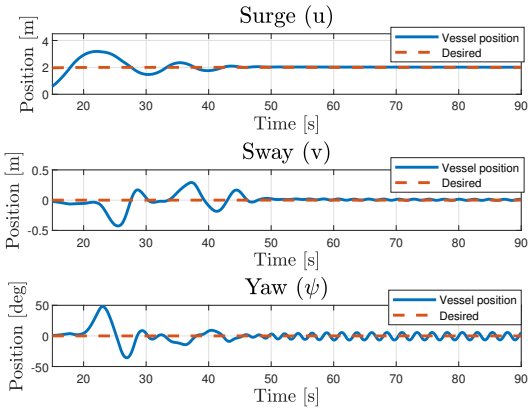
$$\mathbf{G_d} = 2\zeta\boldsymbol{\omega}_n\mathbf{M} - \mathbf{d} \tag{3.47}$$

$$\mathbf{G_i} = \frac{\boldsymbol{\omega}_n}{10}\mathbf{G_p} \cdot 1.5 \tag{3.48}$$

In figure 3.11a the ship path in NED are illustrated and in figure 3.11b the ship movement in surge, sway and yaw are displayed. After nearly 23 seconds it is illustrated that the vessel overshoots with approximately 1 meter in surge. Consequently, the controller compensates with a yaw angle of 50° and movement in negative sway direction in order to reattain the desired position. This is also indicated in the plot of the ship path where the vessel overshoots to 3 meters, rotate and attain the desired position at [2 0 0]. After 50 seconds, the desired position is obtained. However, small oscillations around the setpoint in yaw are present. This could indicate that the $\boldsymbol{G_p}$ gain is tuned a bit too aggressively. However, the oscillations are small in size.



(a) [PID: Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



(b) PID: The vessel position in surge, sway and yaw.

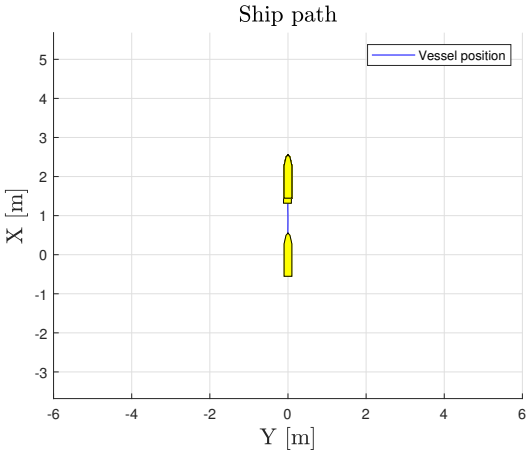
Figure 3.11: PID: Simulation in HIL

Simulation setup

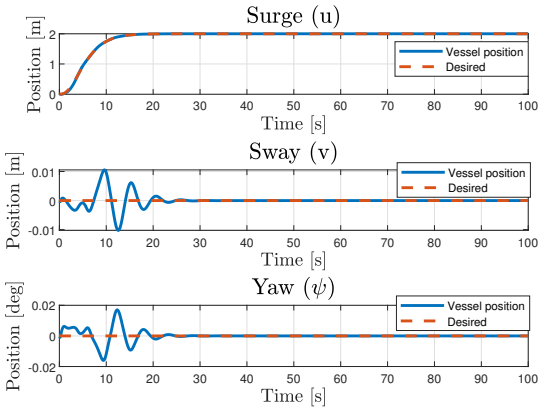
The PID algorithm for the model ship is tested in Simulink, hardware-in-the-loop, and the Marine Cybernetics laboratory. A description of the equipment used during HIL and laboratory testing is found in Chapter 3. Firstly, a presentation of the results from the simulations in Simulink will be given followed by the results from the simulation in HIL. Lastly, the results from model-scale testing in the Marine Cybernetics laboratory will be presented. The tuning gains obtain in the previous Section, 3.6.2, are used in all further simulations. All plots obtained are started from the interval where the simulation is started. That is, if the start value at the time axis is 15 seconds, the simulation started after 15 seconds.

Simulation in Simulink

The objective of the PID algorithm is to obtain the desired position relative to the seabed. In figure 3.12a the ship path of the vessel during simulation in Simulink is displayed. In figure 3.12b the vessel movement in surge, sway and yaw are illustrated. The start point of the vessel is [0 0 0] and the desired setpoint is [2 0 0]. The results in figure 3.12 illustrates that the vessel is able to obtain the desired setpoint. In the time interval 0 - 20 seconds, it is displayed that the vessel follows the reference signal perfectly in surge, sway and yaw small oscillations are displayed. These oscillations are of magnitude 0.01 meter in sway and 0.02° in yaw, thus acceptable.



(a) [PID: Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction

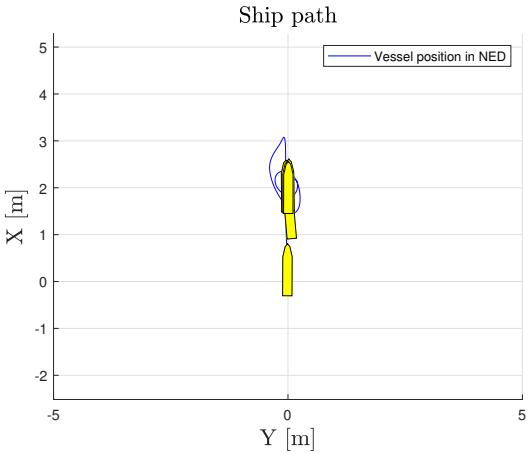


(b) PID: The vessel position in surge, sway and yaw.

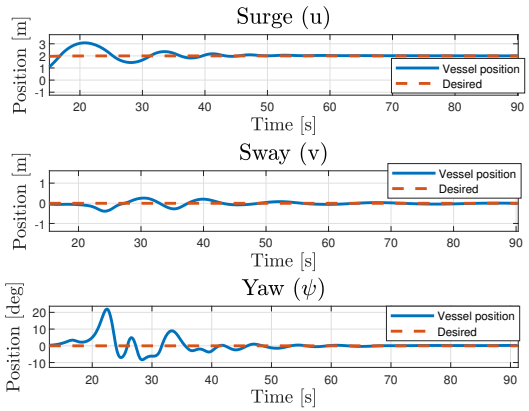
Figure 3.12: PID: Simulation in Simulink

Simulation in HIL

In figure 3.13 the results from hardware-in-the-loop simulation is illustrated. In figure 3.13a the ship plot illustrates that the start point of the vessel is [0 0 0] and desired position is [2 0 0]. In figure 3.13b the movement of the vessel is illustrated. It is shown that in the time interval 0 - 23 seconds, the vessel overshoots 1 meter in surge direction. Consequently, small oscillations in surge, sway and a heading of 20° is obtained in order to compensate for the overshoot. After 50 seconds, the desired setpoint is achieved without any oscillations.



(a) [PID: Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction

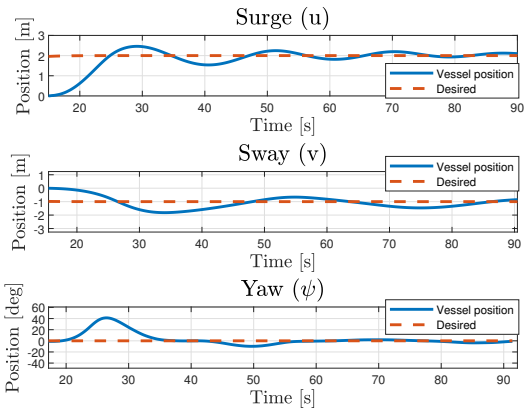


(b) PID: The vessel position in surge, sway and yaw.

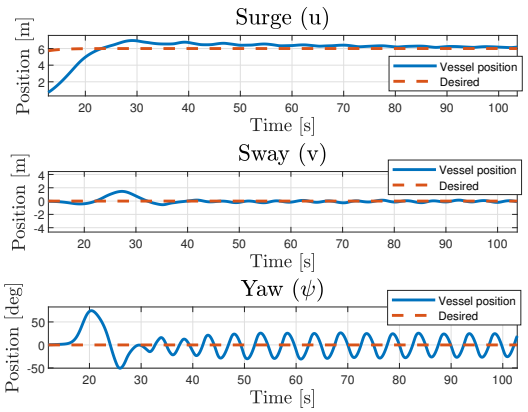
Figure 3.13: PID: Simulation in HIL

In order to verify the robustness of the PID controller a simulation in HIL with setpoint [2 -1 0], figure 3.14, is performed. In figure 3.14a the movement of the vessel is illustrated. In surge, the rise time is longer compared to the previous example. The overshoot in surge is decreased to 1 meter, however an overshoot of 1 meter occurs in sway. In other words, the settling time to the setpoint [2 -1 0] is approximately 90 seconds. In figure 3.14 a HIL simulation with setpoint [6 0

0] is performed. Similarly to the previous simulations in HIL, the vessel overshoots with 1 meter after 30 seconds. The desired position is obtained in sway after 40 seconds and in surge after 70 seconds. However, in the time domain 30 - 100 seconds, the vessel is experiencing constant oscillations of ±25°.



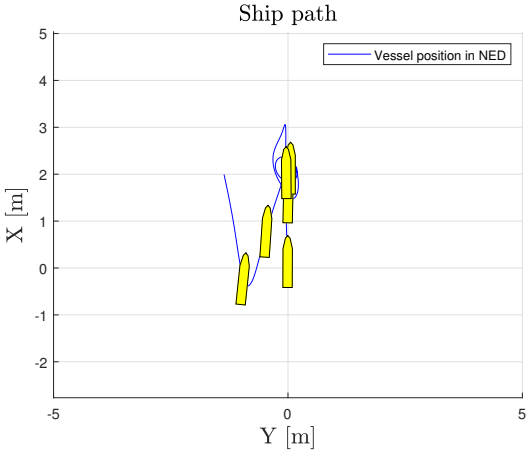
(a) [PID: The vessel position in surge, sway and yaw



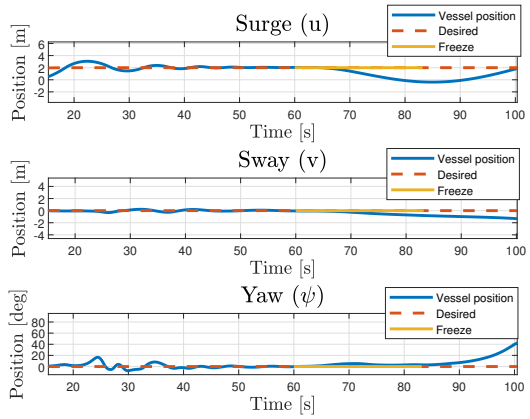
(b) PID: The vessel position in surge, sway and yaw.

Figure 3.14: PID: Simulation in HIL

Lastly, dead reckoning is tested by simulating a loss of signal. In other words, the objective is to verify how well the control system manages to handle dead reckoning. In figure 3.15a the ship path and movement are displayed and in figure 3.15b the vessel position is illustrated where the yellow line indicates the time interval of the signal freeze. The results demonstrate the estimator and vessel manage to follow the true position for 10 seconds. In the time domain 80 - 90 seconds an offset of -4 meters is present in surge. The vessel is able to pass the desired position in surge after 100 seconds. However, the vessel is not able to obtain the desired path. In the ship path plot, the blue line indicates the vessel position after 100 seconds. Hence, the vessel has an offset of -2 meters in sway direction. The vessel was not able to reach the desired position after dead reckoning during this simulation.



(a) [PID: Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction

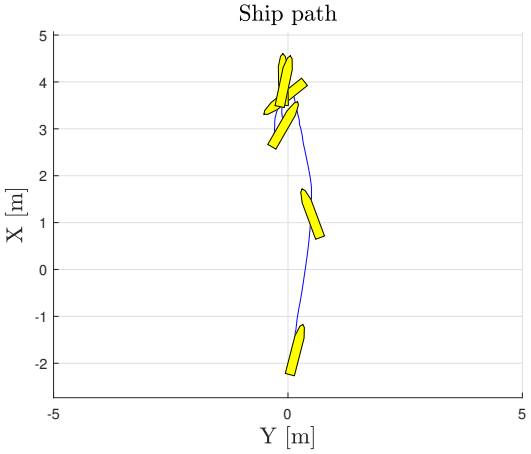


(b) PID: The vessel position in surge, sway and yaw when dead reckoning

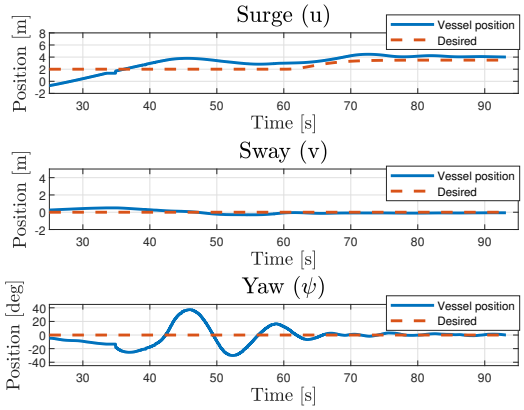
Figure 3.15: PID: Simulation in HIL with dead reckoning

Model scale testing in laboratory

The model scale testing is performed in the Marine Cybernetics laboratory. The path model is illustrated in figure 3.16a, and the movement in surge, sway and yaw are displayed in figure 3.16b. In the latter plot, it is illustrated that the start position during model-scale testing has an offset of -1.5 meters in X - direction. In surge, the vessel has an offset of 1 meter in X - direction after nearly 55 seconds. Before the vessel is able to stabilize the setpoint is changed to 3.5 meters in X - direction. The new setpoint is reached after 78 seconds.



(a) PID: Ship path in NED. X indicate meters in North direction and Y indicate meters in East direction



(b) PID: The vessel position in surge, sway and yaw.

Figure 3.16: PID: model scale testing in laboratory

3.7 Discussion

This Section provides a discussion regarding the experimental results of the two controller design proposed. Following, a general discussion regarding stability, robustness and design methodology of backstepping controller and PID controller.

3.7.1 Experimental results: Manoeuvring control design

The optimal gains for the maneuvering control design were found by trial and error. It follows that μ is linearly influencing the contribution of $V_1^s(\eta, s)$ in (3.28). Consequently, it was noticed that bigger μ drives to faster convergence to the path. However, the model ship did not handle the sharp turns developed from fast convergence. Comparing the different results from the HIL simulation in figure 3.5, one can clearly see that the backstepping controller gives satisfactory results in X and Y position. A small offset of $\pm 1^\circ$ in the heading was present in the time domain 15 to 25 seconds. The heading deviation was small, thus acceptable. It follows that the vessel succeeded in following the straight line parameterization during HIL testing. The results from the laboratory testing is illustrated in figure 3.7. It is displayed that the model ship drifts in negative Y direction when the vessel applies a positive moment. In other words, the model ship did not manage to rotate instantly when commanded to. The vessel did also have trouble to execute surge movement. A possible reason is that the thrust allocated to the actuators is not executed perfectly. Hence, the two VSPs on the model ship is inducing RPM irregularities which the controller struggles to compensate for. The experimental results did also display that the measured heading was quite far from desired. A possible reason for the difficulties in the heading control could be coupling effects in sway and yaw. When calculating the desired forces in sway, a force in yaw will also occur which is not accounted for in the model.

To illustrate how well the model ship managed to handle temporarily loss of position and heading measurements, dead reckoning was performed. The results from dead reckoning in HIL simulation indicated that the vessel managed to keep its desired position for approximately 5 seconds in surge, sway and yaw. In addition, the vessel managed to reattain its desired position in the latter directions after 5 seconds. The results from the model scale vessel deviated from the true position after approximately 2 seconds for surge, and almost immediately for sway and yaw. Consequently, the vessel overcompensated when the position signal is recovered. This suggests that a smoother transition to the actual position and heading should be accounted for with a filter or reference model. Another improvement would be to add the Coriolis effect into the model. This might reduce the quick deviation during dead reckoning.

3.7.2 Experimental results: PID controller design

Finding the values for $\boldsymbol{G_P}$, $\boldsymbol{G_I}$ and $\boldsymbol{G_D}$ to achieve a satisfying response from the controller, was accomplished with the method *PID control with acceleration feedback* form [7]. The results of the adequate gains retrieved are illustrated in figure 3.11. Tuning a PID controller can be challenging and time-consuming. The results illustrated in figure 3.11 is obtained from the HIL simulation. The results suggest that the optimal controller gains were not fully achieved. However, the PID algorithm yields stability and thus, acceptable results. When considering the results from HIL simulation in figure 3.13 it is clear that the vessel is able to reach its desired position [2 0 0]. However, the vessel overshoots 1 meter after 20 seconds, as expected from the tuning results. It follows that the setpoint is reached in surge, sway and yaw with a settling time of 50 seconds.

The robustness of the controller was verified by simulating different setpoints in HIL. The results indicated that the rise time in surge was considerably longer when controlling coupled motion. In addition, it was noticed that the vessel constantly oscillated around the setpoint in surge and sway when adjusting for coupled motion in the latter degrees of freedom. In some cases, the controller struggled when the desired setpoint was with considerable distance from the current position. This could be caused by the reference model not being able to meet the desired response for all amplitudes of the input signals. A reference model can often yield a satisfactory response for one operation point. However, the response for another input could result in a completely different behavior. This could be solved by saturation of the velocity and acceleration elements. The results from the laboratory testing indicated that the vessel managed to reach the desired setpoint and change setpoint. However, for the illustrated scenario, the vessel did not have time to stabilize at first setpoint, before having a new setpoint. Compared to the results from HIL testing, the vessel struggled more in laboratory testing. A possible reason for the small oscillations after 40 seconds may be a bad signal from the cameras in the laboratory. Similar to the backstepping method, dead reckoning was performed in order to increase the operability by fault tolerant control. The dead reckoning for the PID controller was unfortunately only tested in HIL. The results indicated that the vessel managed to keep its true position for about 10 seconds after the signal freeze in surge and sway. In surge a deviation of 2 meters occurs after 20 seconds of dead reckoning. In sway and surge the the deviation is minor. Compared to the HIL results from dead reckoning with backstepping, the PID controller managed the keep its true position 5 seconds longer. This could be explained by the reference model.

3.7.3 General discussion of backstepping and PID

Marine systems and ships are highly nonlinear. The dynamic behavior of marine ships can change significantly during different vessel operational conditions and fault effects [26]. Conventional ship control systems are often designed under the assumption that the kinematic and dynamics of the motion can be linearized. During this case study, only low-speed applications were considered. From Section 2.6.5 it is stated that Backstepping is a technique for designing stabilizing control systems for nonlinear dynamical systems. By introducing state variables, controlling parameters, and a stabilizing function into the mathematical model of the system, the stabilizing function can compensate for unwanted nonlinearities. Hence, global stability of the dynamical system is obtained. In other words, the backstepping controller does not require the resulting input-output dynamics to be linear. For a linear PID controller, feedback linearization is usually used for eliminating nonlinearities of the system. However, nonlinear systems contain *nonlinear phenomena*, which is not possible to predict by linearization [15]. In other words, linearization can cause limitations, as it is an approximation about an operating point.

For industrial use, integral control is one of the preeminent components in feedback control [23]. It follows that integral control has the characteristics of removing steady-state errors and increasing the robustness of the controller. In Section 3.7.1 and 3.7.2 the experimental results from integral backstepping and PID controller indicate this. The backstepping method can be challenging to apply, and do commonly reveal numerous problems. When comparing the tuning process of PID and integration backstepping controller, the tuning process for backstepping was surprisingly less demanding than anticipated. One of the main challenges related to backstepping is the selection of stabilization functions and identification of the associated parameters [30]. As stated in Section 3.6.1, the tuning process of the backstepping controller design was performed throughout trial and error with simulation in Simulink and HIL before model-scale testing was performed. The model ship is used in several student projects and master thesis, hence good knowledge about the vessel dynamics is given. In addition, tuning in Simulink and HIL made the transition over to model scale testing substantially easier as desired. This suggests that good insight into the model simplified the tuning process of the backstepping method.

For the PID controller several tuning methods exist, however the method *PID control with acceleration feedback* was performed and acceptable gains were retrieved. Compared to the recursive backstepping method, the PID methodology is significantly easier. The PID controller is largely applied in the industry due to the simplicity, applicability, and functionality of the controller. From the experimental results in Section 3.6.2 it is illustrated that the PID controller has excellent stability, however poorer when controlling coupled motion. In several control problems, the low-speed assumption limits the accuracy of the analysis. This motivates towards nonlinear control design.

Chapter 4

Case study: Online risk model

An increasing level of autonomy in the shipping industry may lead to safer solutions and more cost-efficient operations. The development of safe and robust dynamic positioning, DP, is a significant stepping stone towards autonomous shipping. The objective for DP is to to keep the position or heading of the vessel relative to the seabed. That indicates that the DP system should be able to resist external forces and move from one place to another at a low speed. Towards autonomous shipping, DP capability is of significant importance concerning auto-docking. In this case study, an online risk model is to be developed for a continuously unmanned ship. The objective with an online risk model is to replace the decision-making and risk assessment of the officer on watch. In addition, an online risk model can support the decision-making of the officer on watch, and thus reduce the workload of the human operator [28]. That is partly unmanned bridge. Hence, an online risk model and a DP are of great importance for a vessel to operate fully or partly autonomous. The main scientific contribution of this thesis is to demonstrate a decision model for a continuously unmanned ship, CUS. The decision model developed will be integrated into the control system of the ship model CS Enterprise I. Following, the model will be tested in simulation and model scale in the Marine Cybernetics laboratory.

The DP control system developed in case study I, will be utilized with the linear PID controller. Following, quantitative modeling of the technical risk influencing factors, RIFs, will be performed. In this thesis modeling of organizational and human risk influencing factors are not considered. However, a discussion regarding the sensitivity of the analysis will be performed. Bayesian approaches are widely used in the offshore and gas industry to support the decision-making of parameters containing uncertainties [9]. Thus, a Bayesian belief network will be developed in order to perform a quantitative modeling of technical risk influencing factors, RIF. Following, the BBN will be integrated into the control system of the ship model. Lastly, three different scenarios will be simulated with Simulink, hardware-in-the-loop and in model scale in the Marine Cybernetics laboratory. All necessary code are attached.

4.1 Case description

The objective with this case study is to develop, verify, and test an online risk analysis in simulation and model scale. The testing of the model will be performed in the Marine Cybernetics laboratory with CS Enterprise I. The experimental setup, vessel and lab characteristics, are described in Chapter 5. Hence, the following tasks will be solved:

1. A Bayesian belief network will be developed in GeNIe in order to provide a qualitative modelling of technical risk influencing factors.
2. Description of the integration strategy of the ORA into the control system of CS Enterprise I.
3. A brief discussion of strengths and limitations with the ORA.
4. Lastly a detailed description of each scenario tested, will be given. Following, each scenario will be simulated and tested in Simulink, Hardware-in-the-loop (HIL) and in model scale.

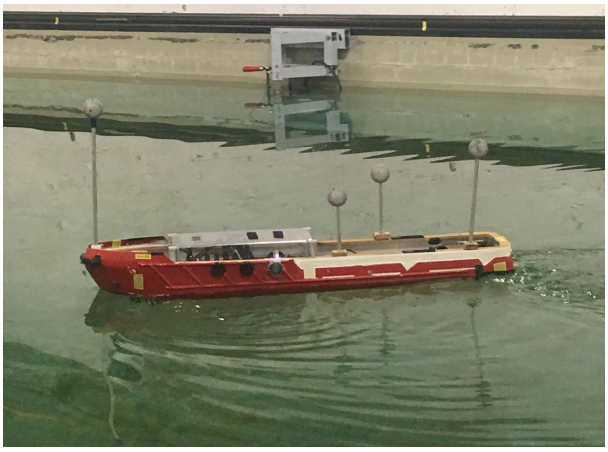


Figure 4.1: Cybership Enterprise I

4.2 BBN risk analysis

In order to develop an online risk model for a surface vessel, a Bayesian belief network is developed. The primary purpose of this case study is to demonstrate the concept of an online decision model for an autonomous system, and thus the BBN developed is simplified. The ship considered in this Chapter is a CUS, continuously unmanned ship. Hence, the vessel is designed for unmanned operation of the bridge at all times, except emergencies. In other words, under special emergencies, a shore control center will be contacted. The objective of a BBN is to identify relevant factors, which can influence the critical event. That is, determine all risk influencing factors or hazards that might increase the probability for the critical event. The network created consists of nodes and directed arcs, indicating state, relationship, and condition of the relevant factors. Each node in the BBN contains a variable of two or more possible states.

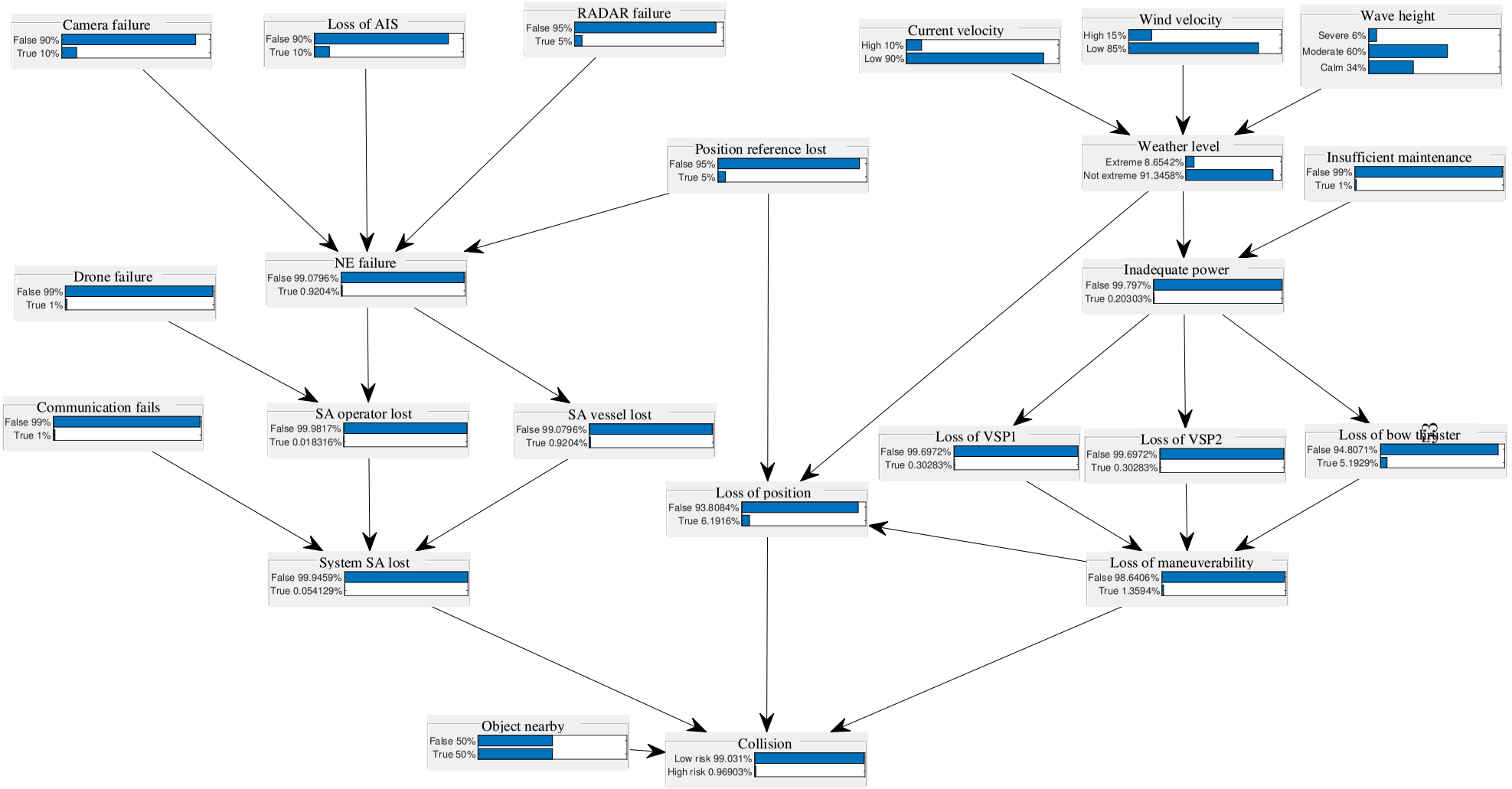
In this BBN, the critical event is *Collision*, where the states of the event is determined as 'High risk' or 'Low risk'. To determine how each specific node contributes to the exact probability of collision is challenging. Hence, *fuzzy logic* is utilized to define which factors in the BBN that contribute towards the higher and lower probability of the end event, collision. Where 'High risk' refers to a state in which the risk of collision is unacceptably high. The probability of collision for a surface vessel should be extremely low. The risk influencing factors and hazards that are included in this network are technical. This decision is made based on the fact that during autonomous operations, human involvement during operation is low. The network consists of 22 nodes that influence the probability of collision. These 22 nodes are divided into two groups:

1. Sensor failure and situation awareness.
2. Environmental and power.

The Bayesian belief network is illustrated in figure 4.2. The left side of the BBN consist of group 1, which consists of four top nodes that are related to sensor failure of navigational equipment: Camera, AIS, RADAR, and position reference lost. These four top nodes are collected into a *NE failure* node, i.e., navigation equipment failure, which affects the total situation awareness (SA). However, if the *NE failure* node is assigned to the state 'True', all situation awareness on the vessel is lost. Correspondingly, *SA vessel lost* node is 'True'. For a shore control center to take control over the surface vessel, it is suggested to install a drone on deck of the ship. Hence, the drone should be automatically started in such a particular scenario, and the situation awareness for an SCC operator shall be recovered. In Chapter 2 under Section 2.1.5 existing and future drone technology is discussed.

The right side of the BBN consist of group 2, which consists of three top nodes related to the environment: Current velocity, wind velocity, and wave height. The *Current velocity* node and *Wind velocity* node is divided into two states, 'True' and 'False', while the *Wave height* node is divided into 'Sever', 'Moderate' and 'Calm'. Following, these three top nodes are parents of the *Weather level* node. The Weather level node is divided into two states 'Extreme' and 'Not extreme'. This node is parents of *Inadequate power* and *Loss of position*, and thus contributes to position loss and power loss. However, inadequate power can also be caused by insufficient maintenance. The CS Enterprise I vessel is equipped with three thrusters - two Voith Schneider propellers and one bow thruster. These thrusters are obviously influenced and fully dependent on the power supply. In addition, the maneuverability of the vessel is dependent on the three thruster nodes and contributes to both the Loss of position node and Collision node. The last input to the end node is an *object nearby* node. The *object nearby* node is developed to compensate for missing sensors on the model ship. Thus, by giving the node evidence of 'True', equals a detected object nearby. All nodes are obtained from simulation by a binary variable representing

the evidence. In figure 4.2, an illustration of the BBN is given.


Camera failure
False 90%
True 10%
Loss of AIS
False 90%
True 10%
RADAR failure
False 95%
True 5%
Current velocity
High 10%
Low 90%
Wind velocity
High 15%
Low 85%
Wave height
Severe 6%
Moderate 60%
Calm 34%
Position reference lost
False 95%
True 5%
Weather level
Extreme 8.6542%
Not extreme 91.3458%
Insufficient maintenance
False 99%
True 1%
NE failure
False 99.0796%
True 0.9204%
Drone failure
False 99%
True 1%
Inadequate power
False 99.797%
True 0.20303%
Communication fails
False 99%
True 1%
SA operator lost
False 99.9817%
True 0.018316%
SA vessel lost
False 99.0796%
True 0.9204%
Loss of VSP1
False 99.6972%
True 0.30283%
Loss of VSP2
False 99.6972%
True 0.30283%
Loss of bow thruster
False 94.8071%
True 5.1929%
Loss of position
False 93.8084%
True 6.1916%
System SA lost
False 99.9459%
True 0.054129%
Loss of maneuverability
False 98.6406%
True 1.3594%
Object nearby
False 50%
True 50%
Collision
Low risk 99.031%
High risk 0.96903%

4.3 Implementation of online risk model

This Section describes the integration strategy of the online risk model, ORM, with the control system of CS Enterprise I. The ORM for the DP vessel is developed based on a Bayesian belief network, BBN, described in Section 4.2. The BBN is used to calculate whether the risk of collision is sufficiently high to justify risk mitigation measures. Figure 4.3 illustrates the proposed ORM, which is developed to increase the safety and level of autonomy during vessel operation. However, the proposed model is a simplified model, developed in order to illustrate the concept of an online risk model. Hence, the framework presented is concentrated around sensor failure, situation awareness, environmental conditions, and power. The flowchart in figure 4.3 describes the process of the model. The gray blocks illustrate the sensors which are sending data of possible risk influencing factors, i.e., object nearby with collision course, severe wave height, maintenance level, etc., to the scenario block. However, due to limited sensors on CS Enterprise I, a synthetic analysis is preformed. Hence, it is assumed the vessel is aware of its surroundings given that relevant sensors are functioning. It is also assumed that the vessel is aware of the sea state, i.e. accurate onboard sea state estimation algorithm is available. When a scenario is defined, the online risk algorithm verifies whether any changes in the scenario have occurred during the last time step. If any new evidence is found the probability is re-calculated in the BBN. If not, no need for re-calculation. Hence, unforeseen interdependencies in the navigation equipment, power system, DP system, or other environmental challenges might be discovered early on by the ORM. The probability of collision is then inserted into the *decision* block.

The *decision* block proposed in this case study is developed in order to distinguish four different decisions, where the purpose is to avoid a collision. Depending on the probability of collision and detected scenarios, an action is chosen. The objective of the decision block is to determine if the vessel can continue with the current velocity or operation. In other words, the decision block needs to analyze failures and fault, and determine if the current probability of collision is tolerable. The probability of collision for a surface vessel should be extremely low. However, it would be challenging to illustrate the concept in the laboratory. For the purpose of verification and testing of the online risk model, the accepted probability limit of collision during this case study is $P_{collision} \leq 6\%$. Hence, in order for an action to be taken, the probability limit needs to be exceeded. The following actions are implemented:

- Start an extra generator when inadequate power is supplied to the thrusters.
- Change the DP setpoint.
- Activate drone and give the command to associated shore control center.
- Activate Alarm when evidence regarding maintenance, operator contacted, change of setpoint or extra generator started.

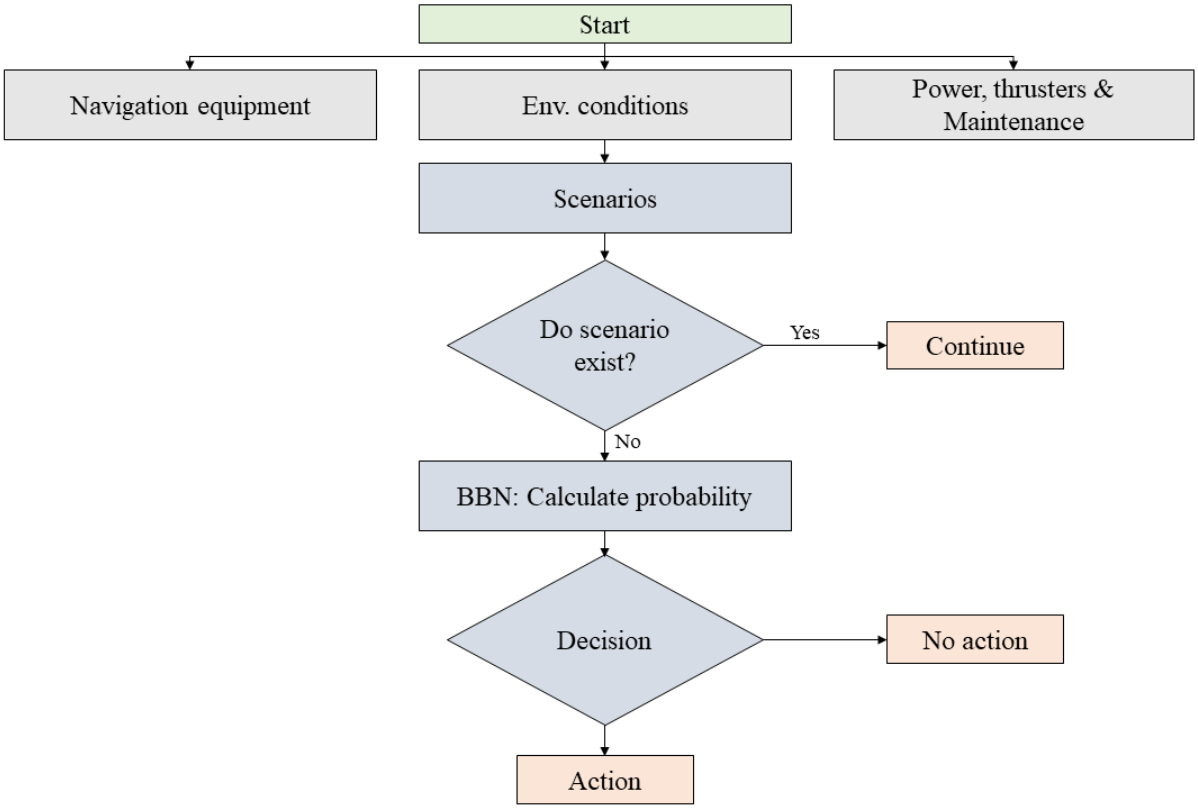


Figure 4.3: Flowchart of online risk model: Gray blocks indicate collective blocks of sensors, which is not available on CS Enterprise I. The blue blocks, indicate the ORM which was implemented into the control system. The orange blocks indicate output from the decision model and input to the controller.

In figure 4.3 an illustration of the control system for CS Enterprise I is given.

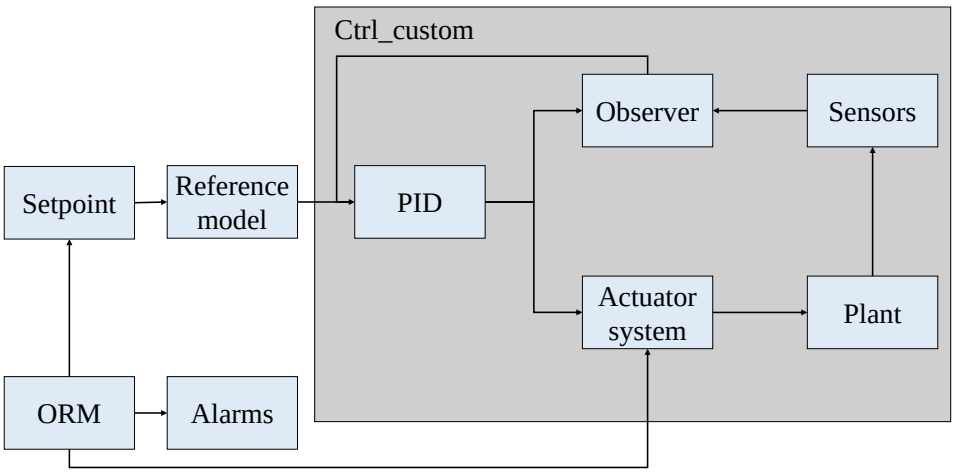


Figure 4.4: Representation of the ctrl_custom Simulink block diagram, along with elements added for the purpose of the thesis. ORM is short for Online Risk Model.

4.4 Scenario I: Shore control center

4.4.1 Simulation setup

Scenario I is developed in order to illustrate the concept of how an online risk model can be used in context with an continuously autonomous vessel, and in particular how the SCC can be alerted and commanded to take over the steering of an vessel. Scenario I consist of a DP operation, where the vessel is changing setpoint from the vessel's current position to the waypoint [2 0 0]. The start position of the vessel is approximately [0 0 0], thus the vessel is moving in North direction. An obstacle is detected by the vessel's navigation equipment after 5 seconds. 15 seconds later, the vessel is experiencing technical issues and a camera failure occurs, followed by loss of AIS signal 10 seconds later. Lastly, 20 seconds after previous event, the vessel is experiencing RADAR failure. In other words, the vessel have lost all situation awareness after 50 seconds of operation. The timeline is listed below:

- 5 seconds: Object nearby detected.
- 20 seconds: Camera failure occurs.
- 30 seconds: Loss of AIS signal.
- 50 seconds: RADAR failure

It is suggested that a drone shall be placed at the unmanned vessel at all times. Hence, in order to recover the situation awareness of the continuously unmanned ship, the ORM is developed to automatically start the drone. Thus, the SCC is given situation awareness and can control the vessel remotely. Hence, the objective during this scenario is to illustrate how an online risk analysis can be used in decision making when the situation awareness on the vessel is lost.

During simulation a conversion block is implemented, in order to compensate for the SCC. The conversion block have the inputs Lstick U/D, Lstick L/R, L2 and R2, similar to the Sixaxis controller used in the Marine Cybernetics laboratory. During simulation manual control is illustrated with the LstickL/R equal 1. In other words, positive sway movement. During lab testing, the sixaxis controller is used. The connection between GeNIe and the hardware used in HIL and laboratory testing did connect successfully. Hence, during HIL and laboratory testing the probabilities calculated from simulation in Simulink are implemented directly to the decision algorithm. All plots obtained, are started from the interval where the simulation is started. That is, if the start value at the time axis is 15 seconds the simulation started after 15 seconds.

4.4.2 Results: Simulink

The Bayesian belief network simulated for this particular scenario is illustrated in figure 4.5. The BBN is simulated in GeNIe with `MATLAB` and Simulink, an explanation of the software are found in Section 5.1.4. The different beliefs for each node are updated simultaneously throughout the simulation. The green text indicate that the beliefs from the previous listed events are updated. It can be seen that *NE failure* and *SA vessel lost* nodes are children of the camera, AIS and RADAR nodes, thus largely dependent on the latter nodes. 30 seconds into the simulation, the beliefs in the following nodes are updated: Object nearby, Camera failure and Loss of AIS. Hence, the probability of 'High risk' of collision is 2 %. After 50 seconds, a RADAR failure occurs and the probability of 'High risk' of collision is increasing to 6.5%. Consequently, the decision part of the online risk model will activate the drone, alert the SCC and give the SCC fully command of the vessel.

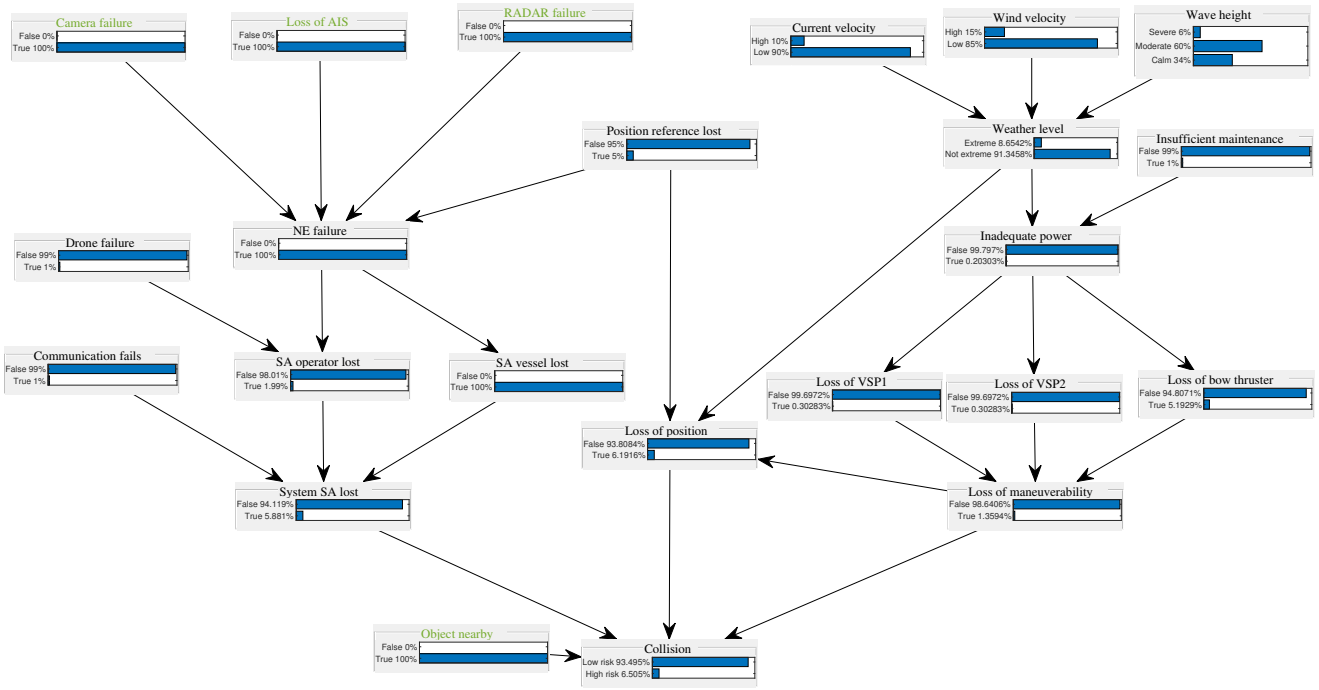
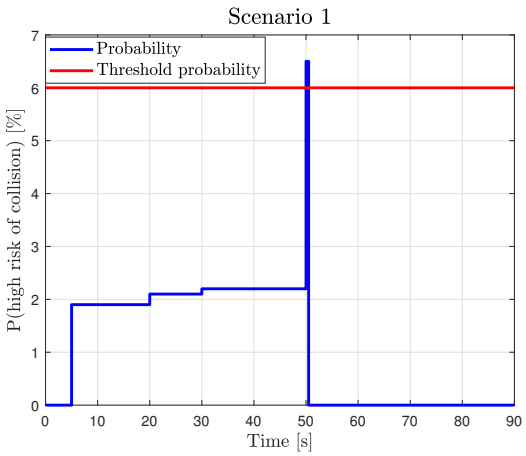


Figure 4.5: Scenario I: BBN

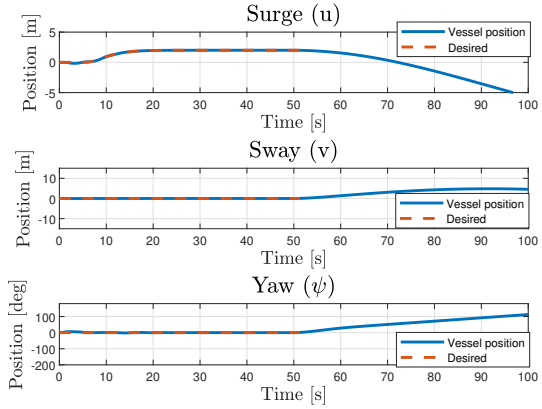
Time [s]	Node	Belief	'High Risk' of Collision	Decision
5	Object nearby	True	2%	-
20	Camera failure	True	2%	-
30	Loss of AIS signal	True	2%	-
50	RADAR failure	True	6.5 %	Control SCC

Table 4.1: Scenario I: Timeline of BBN for each failure

From figure 4.6b it is shown that the setpoint in surge, sway and yaw are reached quickly. However, after 50 seconds the vessel is maneuvering in positive surge direction and negative sway direction. This is caused by the online risk model, as the probability for 'High risk' have exceeded the value of 6%, and the manoeuvring of the vessel is switched from the DP controller to manual control. In other words, the steering command is given to the SCC. During manual control, the thrust allocation is given movement in positive sway. That is, movement in East direction. In figure 4.6a the development of the risk levels for scenario 1 is plotted.



(a) Scenario I: Evolving probability for Scenario 1



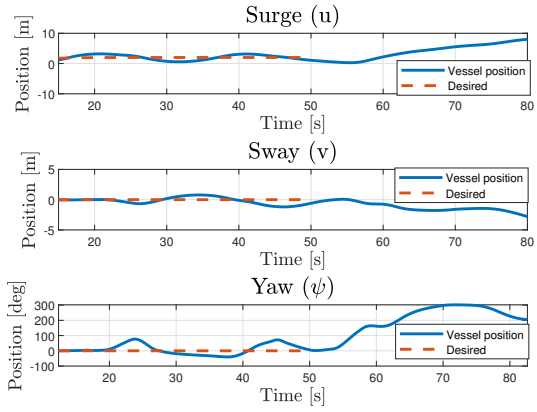
(b) Scenario I: The vessel movement during hardware-in-the-loop testing

Figure 4.6: Scenario I: The vessel movement and risk level during simulation

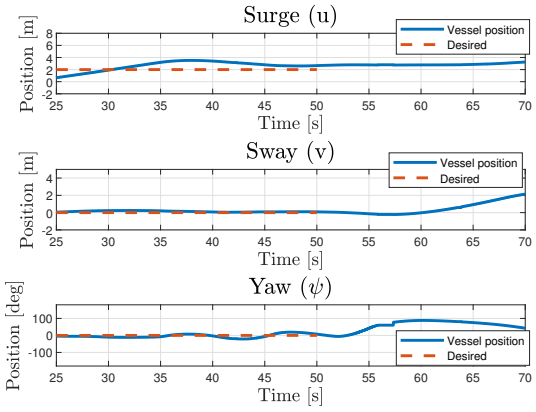
4.4.3 Results: HIL and LAB

In this Section the results from hardware-in-the-loop testing will be presented, followed by model scale testing in the Marine Cybernetics laboratory. Figure 4.7a displays the different forces calculated by the controller during HIL testing of scenario I. The start position of the vessel is approximately [0 0 0], and the setpoint is [2 0 0]. The dashed line indicate that the ship is controlled by the PID controller. The first subplot illustrates the surge movement of the vessel. It is shown that the setpoint in surge is reached after approximately 18 seconds. However, the vessel have small oscillations around the setpoint. From the plot of the surge movement, it is shown that the vessel is reaching its desired position after 15 seconds, until around 35 seconds where small oscillations occurs. After 50 seconds the manoeuvring of the vessel is changed to manual control, thus no dashed line. The vessel is given positive yaw movement followed by force in sway direction.

In figure 4.7b the different forces calculated by the control system during model scale testing in laboratory are illustrated. The vessel is given the identical setpoint as in simulation and HIL testing. However, in model scale testing the start point was approximately [-2 0 0]. From the first plot, the surge movement is displayed. The vessel reaches its setpoint after 45 seconds. Compared to the results from simulation and HIL, the settling time is longer during model scale testing. After 50 seconds the ORM is noticing loss of situation awareness on the vessel. Hence, the operator is given command of the vessel. I.e. manual control of the vessel with the Sixaxsis controller.



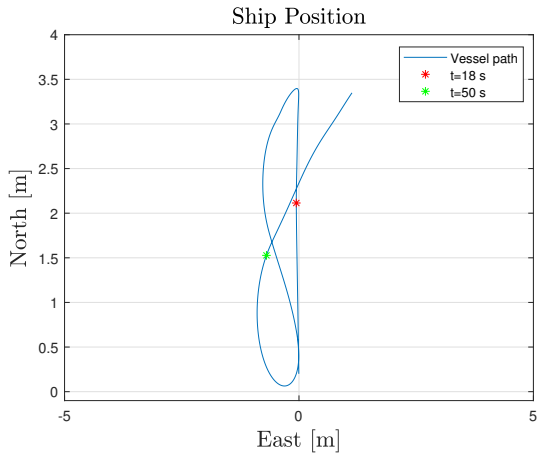
(a) Scenario I: The vessel movement during hardware-in-the-loop testing



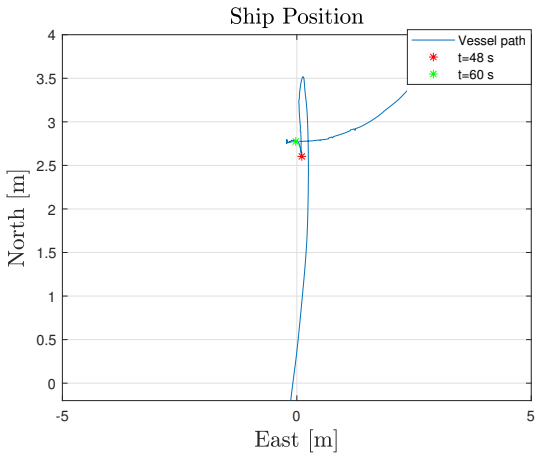
(b) Scenario I: The vessel movement during model scale testing

Figure 4.7: Scenario I: Vessel movement HIL and model scale

From figure 4.8 the vessels movement in NED is displayed. Figure 4.8a contains the position of the ship during HIL testing. As previously stated, the vessel is reaching its desired position after 18 seconds, this is indicated by the red dot. However, the vessel is struggling to keep the setpoint and after 50 seconds the vessel is at position [1.5 0 0], indicated by the green dot. When the control of the vessel is shifted to manual, the vessel is manoeuvred 60° in East direction. In figure 4.8b the path of the ship model during testing is illustrated. Accordingly, it can be seen that the operator is manoeuvring the vessel away from the previous path. I.e. away from the detected object. The vessel is given positive sway force and positive yaw moment. The positive yaw moment is resulting in a 90° turn, thus movement in East direction according to NED.



(a) Scenario I: Vessel path during HIL testing



(b) Scenario I: Vessel path during model scale testing

Figure 4.8: Scenario I: Vessel path HIL and model scale

Figure 4.9, displays the Veristand interface during HIL testing and model scale testing in the laboratory. All led lights in the user interface are positioned left to the text. Hence, the green led light left to 'Operator' indicate that the vessel is manually controlled by the shore control center. This is also indicated by the DP/manualout variable. The red led left to 'DP indicate that dynamic positioning is off, as expected. In addition the green led light left to 'Power' indicate that the vessel is supplied with sufficient power. The black alarms left to 'Setpoint changed' and Maintenance' indicate that associated alarms are not altered.

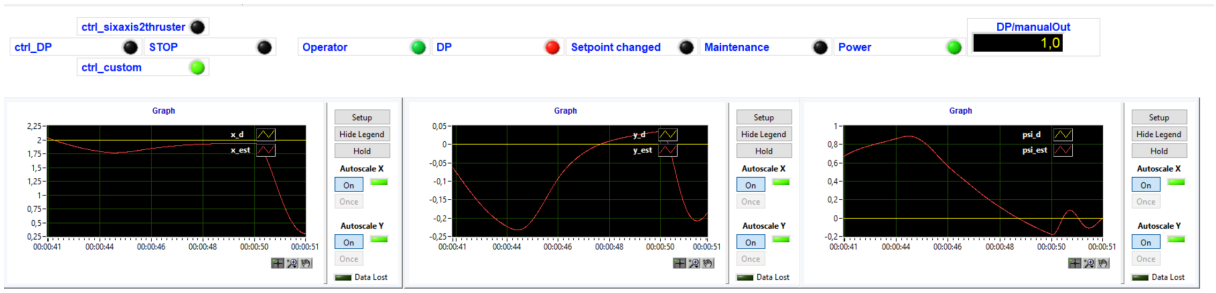


Figure 4.9: Scenario I: Veristand user interface

4.5 Scenario II: Weather and maintenance

4.5.1 Simulation setup

Scenario II is developed in order to clarify the case where a continuously autonomous vessel is operating in harsh weather with unsatisfactory maintenance. The online risk model is explained in Section 4.3, and is developed to change setpoint if the probability of 'High risk' for collision exceeds the probability of 6 % when an object nearby is detected. To decrease the risk of collision, the mitigating measure is to change setpoint of the vessel. Similar to scenario I, this scenario consist of an vessel which is controlled by a PID controller in order to reach the setpoint [2 0 0].

The sequence of of event's during the DP operation are as following. 5 seconds into the operation an object nearby the vessel is detected. 20 seconds later, the wave height have evolved to 'Severe'. Consequently, 5 seconds later the online risk model is updating the weather level to 'Extreme weather'. Lastly, 35 seconds after the updated weather report, an alarm concerning unsatisfactory maintenance is on. The sequence of the different events listed:

- 5 seconds: Object nearby is detected.
- 20 seconds: Severe wave height is detected.
- 25 seconds: Extreme weather level is detected.
- 60 seconds: Alarm "Operations and maintenance unsatisfactory".

All plots obtained, are started from the interval where the simulation is started. That is, if the start value at the time axis is 15 seconds the simulation started after 15 seconds.

4.5.2 Results: Simulink

In figure 4.10 the Bayesian belief network for scenario II is illustrated. The BBN is simulated in MATLAB, Simulink and GeNIe. The different beliefs listed above are updated simultaneously through out the simulation. In addition, the different probabilities are calculated and updated in GeNIe. In table 4.2 the results from the BBN for each failure event are established. Hence, it follows that a detected object nearby results in 2% 'High risk' of probability for collision. Following, 10 seconds later, the probability of 'High risk' for collision is increasing to 5%, due to updated belief in wave height. After 25 seconds the probability of 'High risk' of collision have exceeded the limit in the decision model, thus the vessel is commanded to change setpoint in order to avoid collision. 35 seconds later, an alarm concerning insufficient maintenance is on. For each of these steps, a BBN is calculated. Figure 4.10 contains the final result during simulation of scenario II.

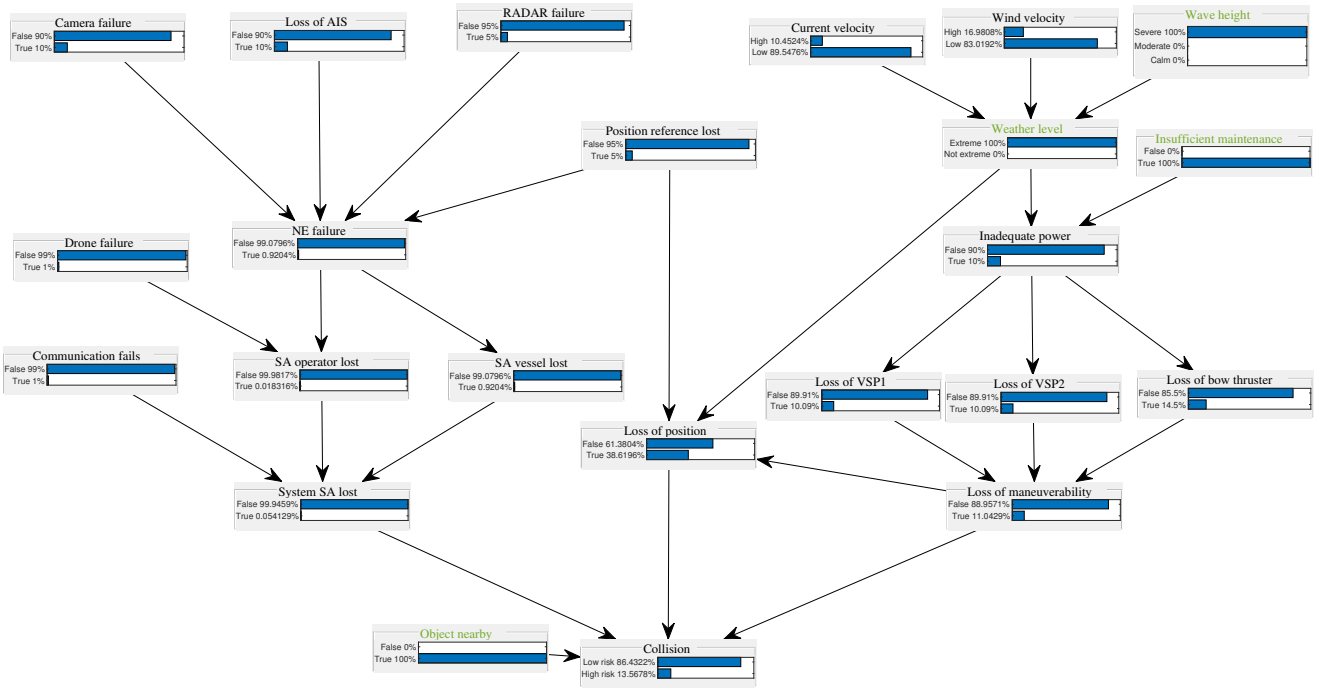
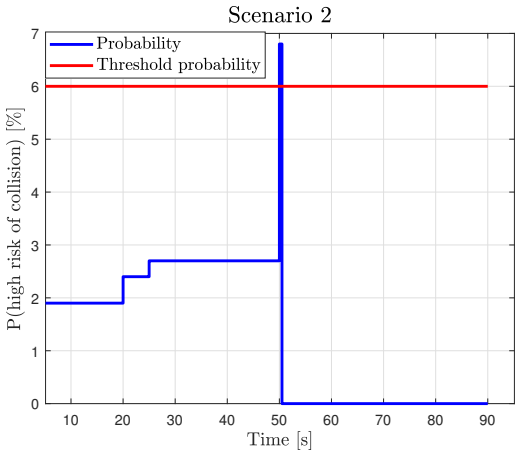


Figure 4.10: Scenario II: BBN

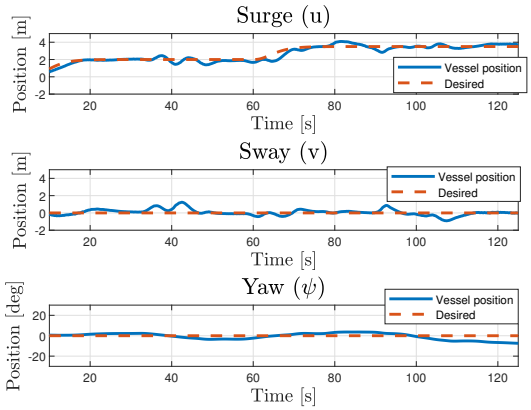
Time [s]	Node	Belief	'High Risk' of Collision	Decision
5	Object nearby	True	2%	-
20	Wave height	Severe	5%	-
25	Weather level	Extreme	6%	Change setpoint
60	Insufficient maintenance	True	13.5 %	ALARM

Table 4.2: Scenario II: Timeline of BBN for each failure event

The evolving probability during simulation of scenario 2 is illustrated in figure 4.11a. In figure 4.11b the vessels movement during simulation is calculated. The first subplot displays the movement in surge direction. Hence, the vessel is reaching its desired position, 2 meters North, after approximately 18 seconds. However, after 40 seconds the vessel is experiencing small oscillations around the setpoint. The small oscillations are mainly caused by two small jumps in sway direction. After 60 seconds, the setpoint is changed to 3.5 meters as expected from the Bayesian belief network.



(a) Scenario II: Evolving probability during simulation



(b) Scenario II: Vessel movement during simulation

Figure 4.11: Scenario II: Evolving probability and vessel movement during simulation

Figure 4.12 displays the ships position in NED during simulation. The yellow dot illustrates that the vessel is reaching its setpoint after 20 seconds. After 60 seconds the decision model is changing the setpoint, due to 'High risk' of collision. As expected the new setpoint [3.5 0 0] is reached by the vessel after 82 seconds.

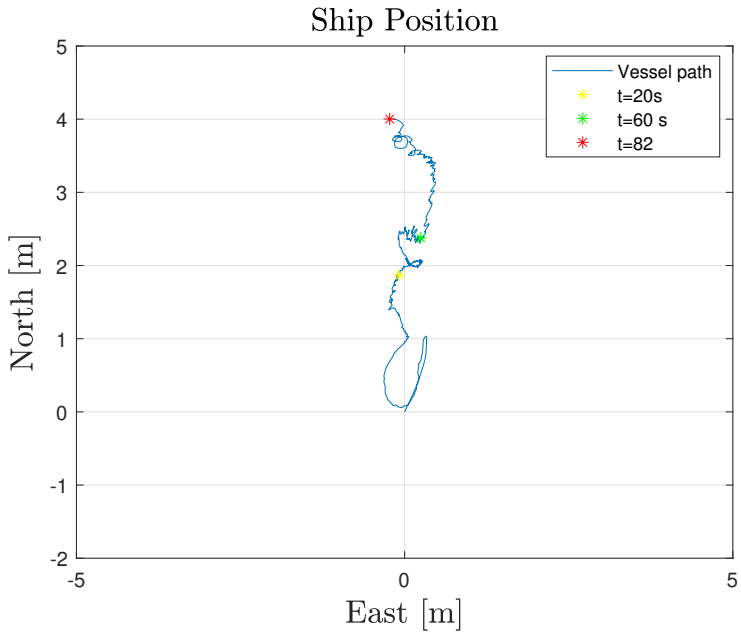
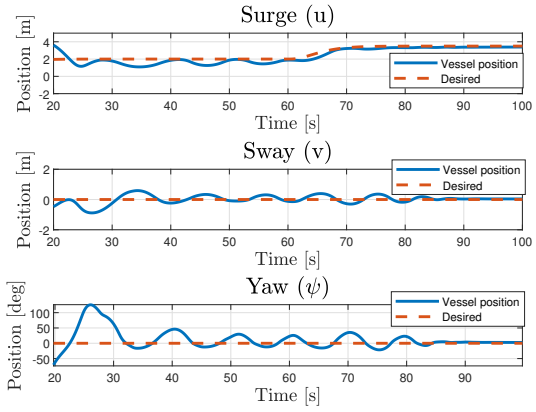


Figure 4.12: Scenario II: Vessel path during HIL and model scale testing

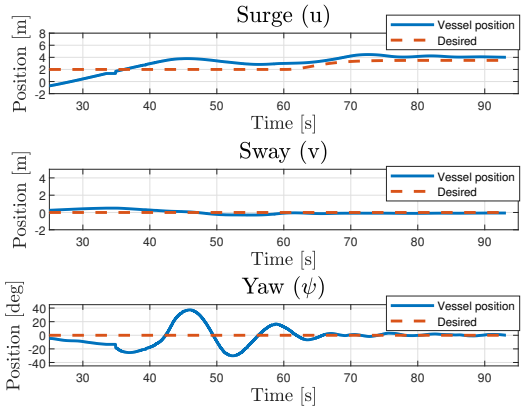
4.5.3 Results: HIL and LAB

In this Section the results from hardware-in-the-loop testing will be presented, followed by model scale testing in the Marine Cybernetics laboratory. Figure 4.13a displays the movement in surge, sway and yaw during HIL testing of scenario II. The start position during HIL testing is [4 -1 -π], and the setpoint [2 0 0]. In the ship movement plot, figure 4.13a, the blue line indicate the vessels actual position, the red dashed line indicate the desired position. Hence, it is illustrated that the vessel oscillates around the desired setpoint in all motions. However, after 60 seconds the decision model change setpoint. This is due to the limit of the probability 'High risk' of collision, which is exceeded. The new setpoint [3.5 0 0] is reached after 7 seconds in surge, and after 85 seconds in sway and yaw.

The results from the model scale testing are illustrated in figure 4.13b. The first subplot displays the change of position in surge direction. Hence, the start position during model scale testing have an offset of -1.5 meters in X - direction. In surge, the vessel have an offset of 1 meter in X - direction after nearly 55 seconds. After 60 seconds the decision model changes the setpoint to 3.5 meters in North direction. This is expected, as an increase in the probability for collision appears. However, the the vessel did not get enough time to stabilize at first setpoint. The new setpoint is reached after nearly 78 seconds, similar to the results from HIL testing. Due to the size of the basin and the range of the cameras, only a small change in the setpoint was given.



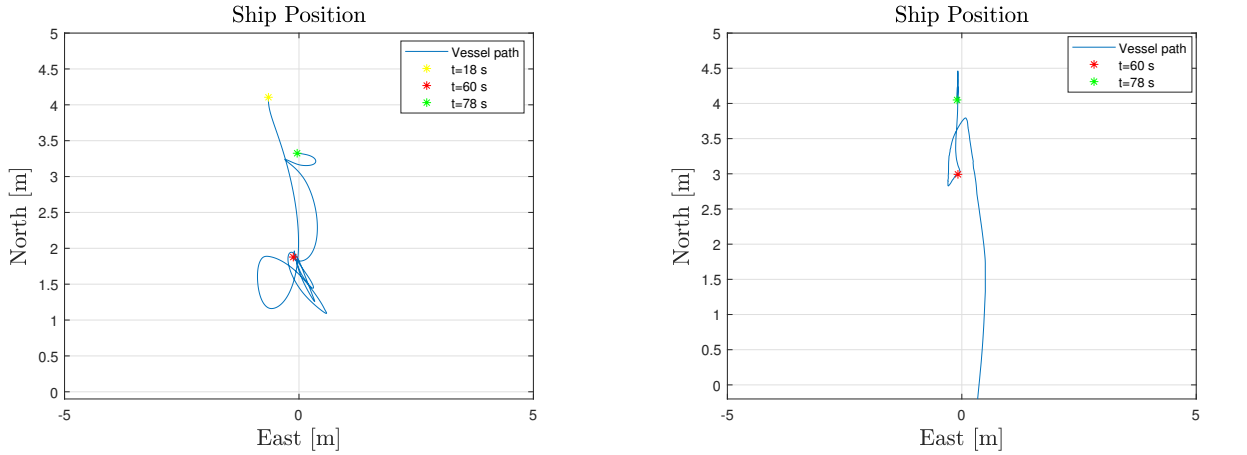
(a) Scenario II: Vessel movement during hardware-in-the-loop testing



(b) Scenario II: Vessel movement during model scale testing

Figure 4.13: Scenario II: Vessel movement during simulation

In figure 4.14 the ship position during HIL and model scale testing are illustrated. The left plot, figure 4.21a illustrates the ship position during HIL testing in NED. The start position of the vessel is indicated by 18 seconds. Thus it is illustrated that the start position is [4 -1 $-\pi$]. After 60 seconds the set point is changed, this is illustrated by the red dot. Following, the green dot indicate the position of the vessel after 78 seconds. Hence, the vessel have an small offset from the new setpoint. In figure 4.21b the ship model's position in NED is illustrated. From the figure it is shown that the vessel have a small offset in sway direction. After 60 seconds the ORM have changed the setpoint, and the model ship reaches the setpoint after 80 seconds.



(a) Scenario II: Vessel path during HIL testing

(b) Scenario II: Vessel path during model scale testing

Figure 4.14: Scenario II: Vessel path

Figure 4.22, displays the Veristand interface during HIL testing and model scale testing in the lab. All led lights in the user interface are positioned left to the text. Hence, the green led light left to 'DP' indicate that the vessel is controlled by the DP controller. In addition The green led light left to 'Power' indicate that the vessel is supplied with sufficient power. In addition, it is alerted that the setpoint is changed and that the maintenance level is insufficient. That is, the model have received an updated belief in the *Weather level* and *Insufficient maintenance* nodes.

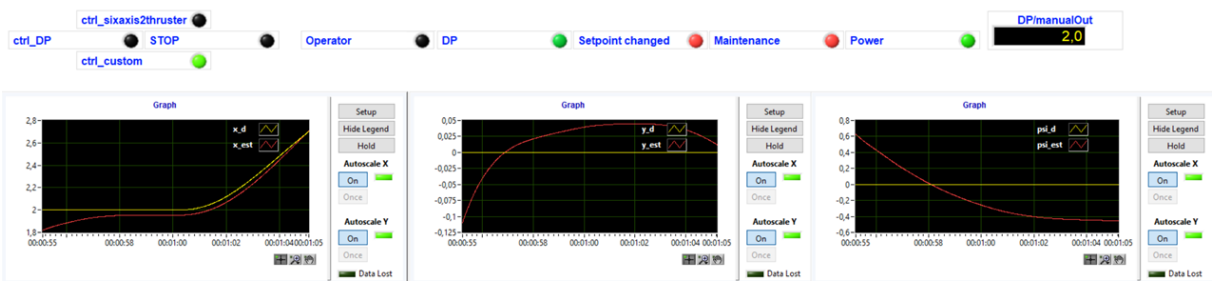


Figure 4.15: Scenario II: Veristand

4.6 Scenario III: Power loss

4.6.1 Simulation setup

Loss of position can occur in the event of extreme weather, for example, if large waves of wind gust temporally forces the vessel out of position, or if the manoeuvrability of the vessel is lost or reduced to an inadequate level. This may be caused by the loss of vessel thrusters, which in turn may occur due to inadequate power supply. The objective of scenario III is to illustrate how the online risk model can compensate for loss of power by starting an extra generator. It is assumed that the extra generator provide adequate power. Similarly to scenario I and II, the vessel is given the setpoint [2 0 0], and controlled by a PID controller. During the DP operation an alarm on inadequate power appears and the mission of the online risk model is to start an extra generator in order to compensate for the power loss. The power loss occurs 30 seconds into the operation.

4.6.2 Results: Simulink

After 5 seconds the navigation equipment of the vessel is detecting an object nearby. Consequently, the probability for collision is increasing to 2 % of 'High risk' for collision. Following, after 30 seconds the BBN is updating its belief in inadequate power, to true. 10 seconds later, an extra generator is started and the thrusters is given a satisfactory amount of power. In figure 4.16 the BBN for the scenario is illustrated when the power is recovered. Table 4.3 contains a summary of the scenario, probability and decisions.

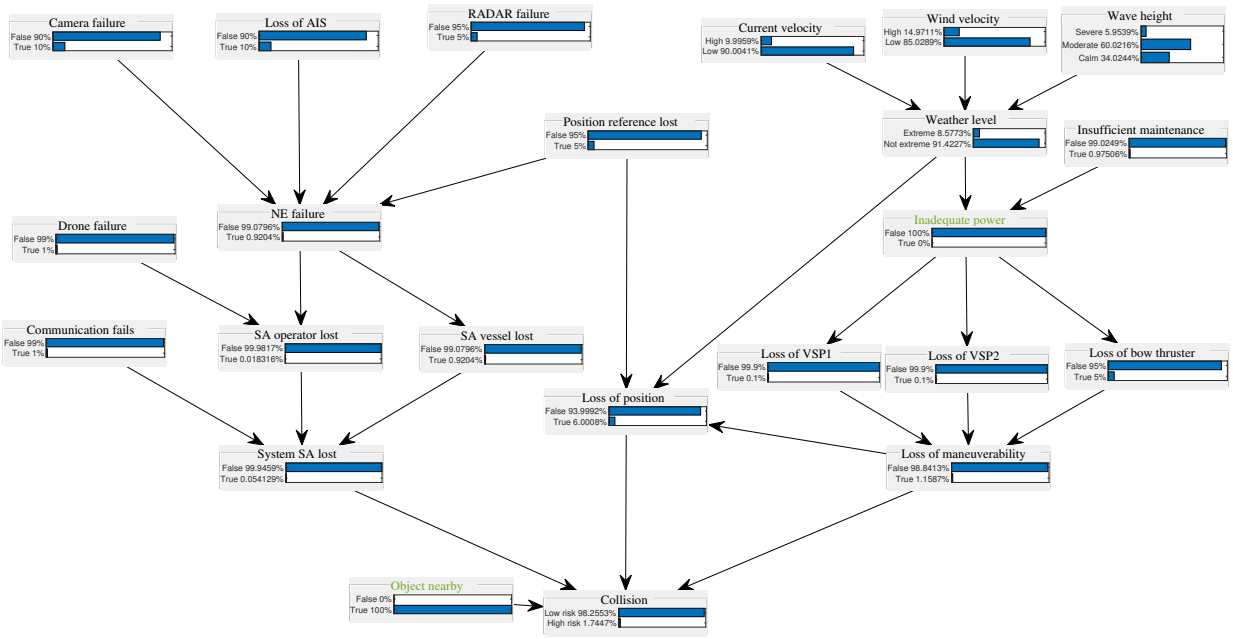
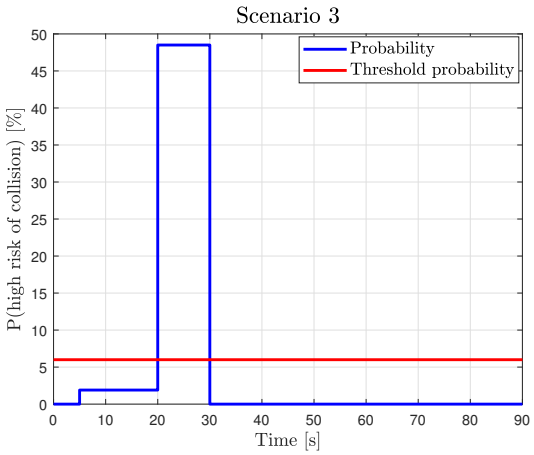


Figure 4.16: Scenario III: BBN

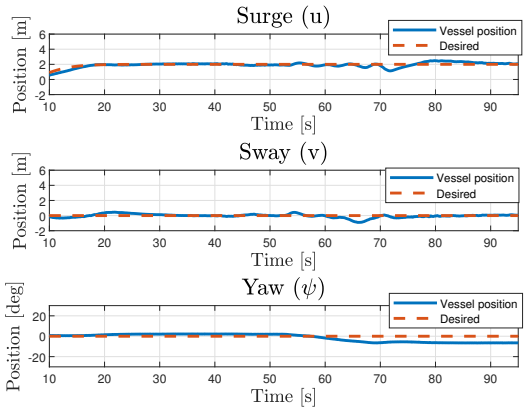
Time [s]	Node	Belief	'High Risk' of Collision	Decision
5	Object nearby	True	2%	-
30	Inadequate power	True	48%	Start extra generator
40	Extra generator	True	2%	-

Table 4.3: Scenario III: Timeline of BBN for each failure

The evolving risk level during simulation of scenario 2 is illustrated in figure 4.17a. It is displayed that inadequate power increase the risk level for collision considerably, thus expected. The vessels movement in surge, sway and yaw during simulation is displayed in figure 4.17b. Thus, the desired position in surge is reached after nearly 15 seconds.



(a) Scenario III: Evolving probability during simulation



(b) Scenario III: Vessel movement during simulation

Figure 4.17: Scenario III: Evolving probability and vessel movement during simulation

In figure 4.18 the first subplot displays the alarms. This plot show that after 30 seconds the power on the vessel is lost. However, the power is recovered after 40 seconds. In other words, the online risk model is starting the extra generator based on the updated belief on inadequate power. From the third subplot the thrust input to the thrust allocation is plotted. In the interval 30 - 40 seconds the plot indicate power loss.

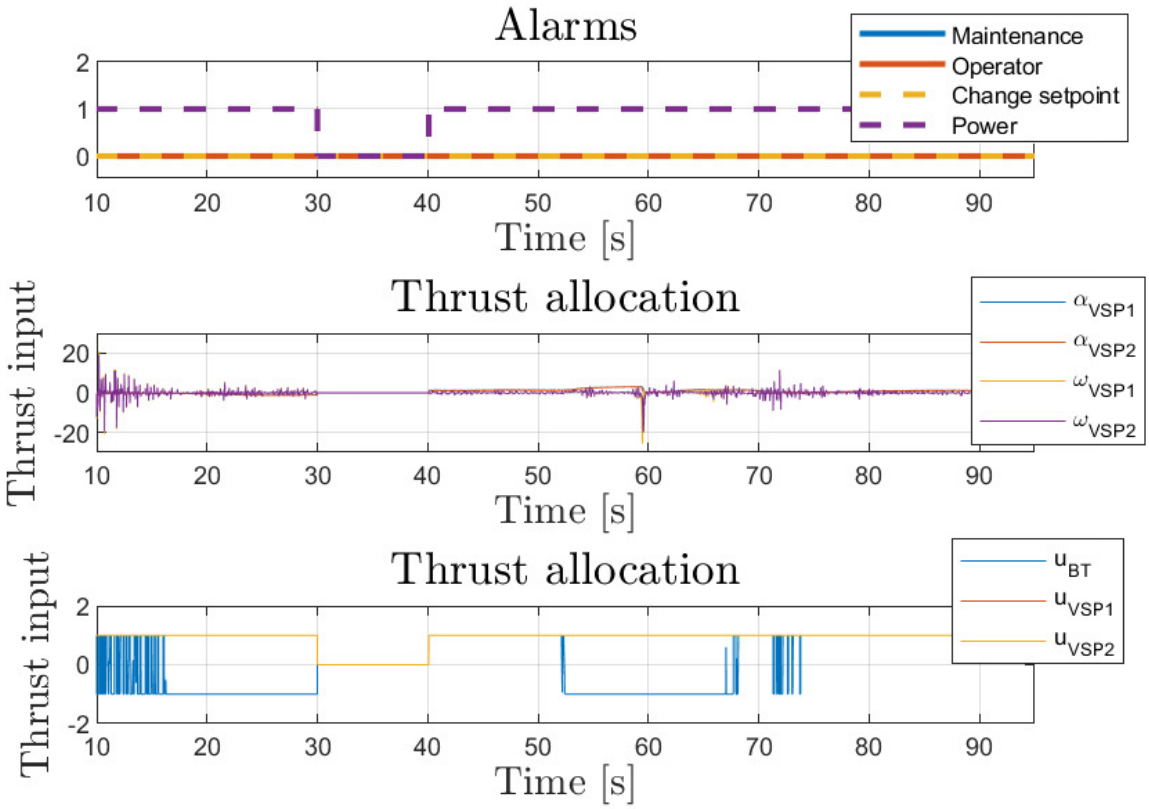
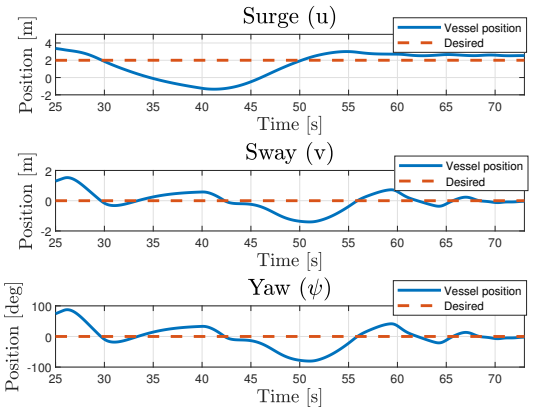


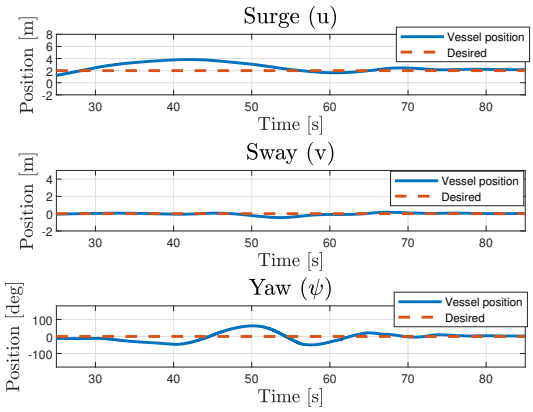
Figure 4.18: Scenario III: Thrust allocation input and alarms

4.6.3 Results: HIL and LAB

In this Section the results from HIL and LAB will be shown. Firstly, the HIL testing results will be presented, followed by the model scale results from LAB testing. The start position during HIL testing is [4, 1, π] and the setpoint is [2 0 0] similar to the previous scenarios. In figure 4.19a the first subplot displays the movement in surge. As a result of the start position the vessel is heading for the setpoint. However, in the time domain 30 - 40 seconds a power loss occur. The power loss is clearly shown in the surge plot. After 27 seconds, the latter plot indicate that the vessel position is decreasing, and in the interval 30 - 40 seconds the vessel continue to drift in the same direction. At 40 seconds the power is on, thus the power to the thrusters are recovered and the vessel is headed towards the desired position.



(a) Scenario III: Vessel movement during HIL testing

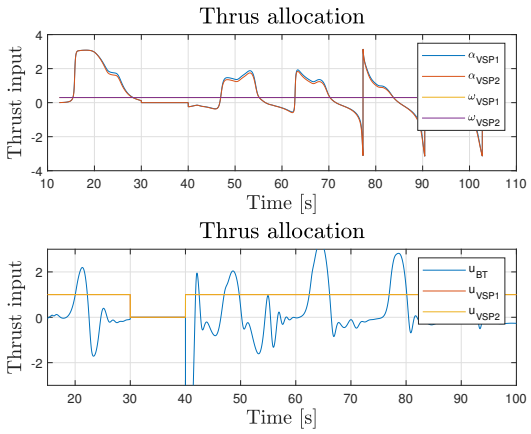


(b) Scenario III: Vessel movement during model scale testing

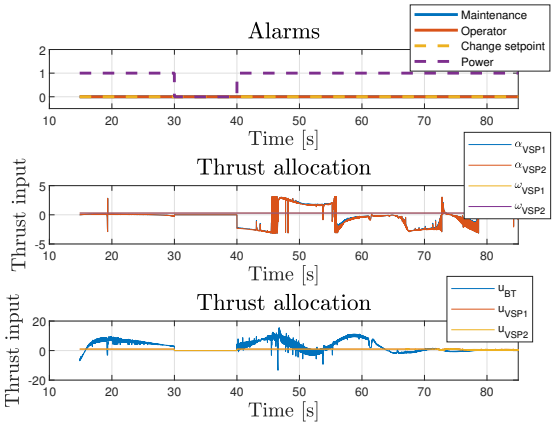
Figure 4.19: Scenario III: Vessel movement during HIL and model scale testing

In figure 4.19b the motion in surge, sway and yaw during model scale testing in lab is displayed. If looking at the sway and yaw plots, it is shown that vessel reached its desired position in sway and yaw after 20 seconds. However, after 30 seconds the vessel have an decreasing angle in yaw. The decreasing angle is likely due to the movement in surge. If looking at the first subplot it is shown that the vessel reaches its desired position after 25 seconds. However, the vessel drifts past the setpoint. This is caused by the power loss. As seen during HIL testing, the ship keep its current direction when the power loss occurs, thus the vessel keep moving with the same direction and steepness. At 40 seconds an extra generator is started, hence the model ship reaches the setpoint 35 seconds later. If looking at the yaw angle, the setpoint is reached after 75 seconds. In other words, the results is as excepted from the HIL testing.

In figure 4.20a the thrust input for the vessel during HIL testing is displayed. As excepted the thrust input equals zeros during the interval 30 - 40 seconds. After 40 seconds an extra generator is started and the power is retrieved.



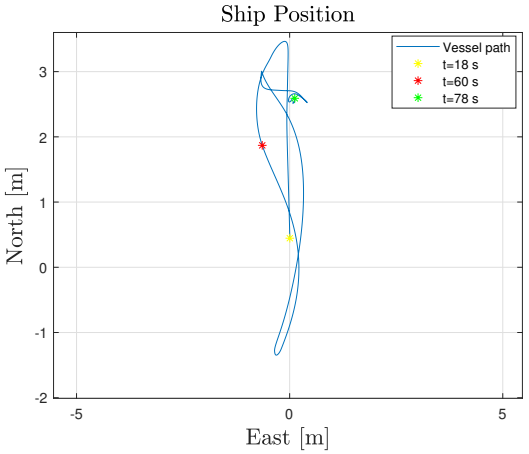
(a) Scenario III: HIL testing



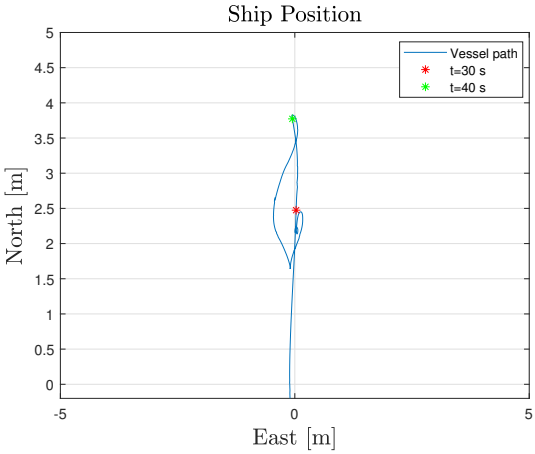
(b) Scenario III: Thrust allocation input and alarms during model scale testing

Figure 4.20: Scenario III: Thrust allocation during HIL and model scale testing

In figure 4.20b the thrust inputs and alarms during model scale testing in LAB is displayed. The first subplot indicate that the alarm for 'Power inadequate' is activated. The two remaining subplots illustrates the thrust input to the thrust allocation. As expected the thrust input equals zeros in the interval 30 - 40 seconds, and is retrieved at 40 seconds. During LAB and HIL testing the saturation block on the thrust allocation was removed. In figure 4.21 the ship path during HIL testing and model scale lab testing is illustrated.



(a) Scenario III: Vessel path during HIL testing



(b) Scenario III: Vessel path during model scale testing

Figure 4.21: Scenario III: Vessel path

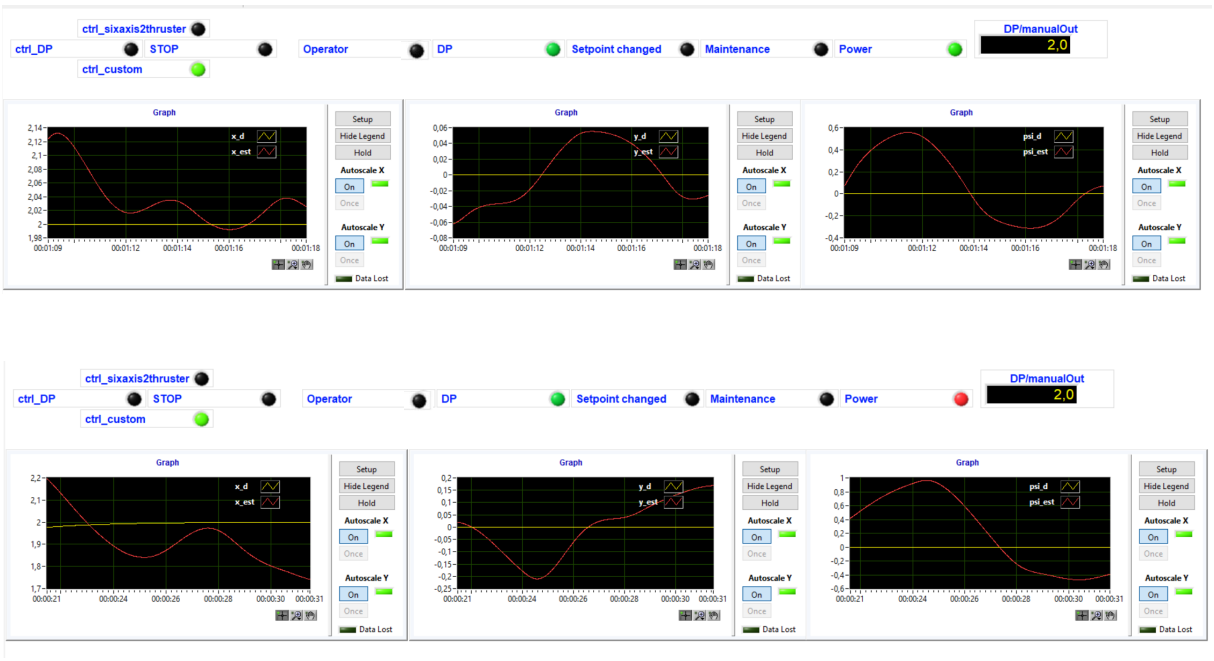


Figure 4.22: Scenario III: Veristand

4.7 Discussion

For a surface vehicle, situation awareness can be described as having a good perception and understanding of the surroundings of a vehicle at all times. The surroundings of a ship is a continually evolving picture, which today is the responsibility of the officer on watch. The objective of an online risk model is to replace the decision-making and risk assessment of the officer on watch, and thus reduce the workload of the operator fully or partly. The primary outcome of this thesis is the framework developed for online risk analysis. The online risk model was tested and verified with simulations of three different scenarios in Simulink, hardware-in-the-loop, and model scale testing in the Marine Cybernetics laboratory. This Section aims to discuss the strengths and limitations in the online risk model.

The framework presented is concentrated around situation awareness, environmental conditions, and maintenance. The first scenario tested was developed to demonstrate how an online risk model can be used in combination with a shore control center. From the results in Section 4.4 it was illustrated that the decision model granted the associated shore control center command of the vessel when evidence on loss of situation awareness occurred. In other words, the online risk model behaved as desired. However, a possible extension to this scenario would be to introduce a safe mode earlier in the time series. Consequently, the SCC could be altered in case of a triggering event such as loss of AIS or RADAR failure. This suggests that the SCC would obtain the possibility to handle deviations from normal operations before the situation awareness on the vessel is lost.

The objective of the second scenario was to show how the decision model handles harsh weather and unsatisfactory maintenance. The simulation results for scenario II is illustrated in Section 4.5. It is immediately apparent that the decision model is able to calculate whether the risk of collision is sufficiently high to justify mitigating risk measures. However, lack of precise information and environmental conditions are risk influencing factors that may require operator control [28]. Similarly to scenario I, a possible extension is to introduce an alert to the SCC in case of harsh weather. The third scenario demonstrated the case where inadequate power was supplied to the thrusters. In this scenario a threshold value should be implemented for power. That is, if it is high probability of inadequate power an extra generator should be triggered to prevent power loss. In this thesis the risk influencing factors are represented with discrete values in the Bayesian belief network. However, these values are continuous in reality and should be implemented with continuous probability distribution in order to improve the accuracy of the analysis. However, the simulation results in Section 4.6, and the decision algorithm behaved as desired.

Chapter 5

Experimental setup

5.1 Experimental Setup

This chapter is written in cooperation with Ina Bjørkum Arneson. The Chapter covers background information on the vessel used in experiments, CS Enterprise I, as well as information on the hardware and software necessary to perform the HIL-simulations and model experiments. The chapter is mainly based on information from [24]. All experiments were performed in the Marine Cybernetics laboratory (MC lab) basin at NTNU.

5.1.1 CS Enterprise I

The CS Enterprise I is 1:50 scale tug boat, equipped with one bow thruster (BT) and two Voith Schneider propellers (VSP). The vessel is illustrated in Figure 5.1, with corresponding dimensions. The model ship is used during model scale testing of the manoeuvring control design, PID controller and online risk model.

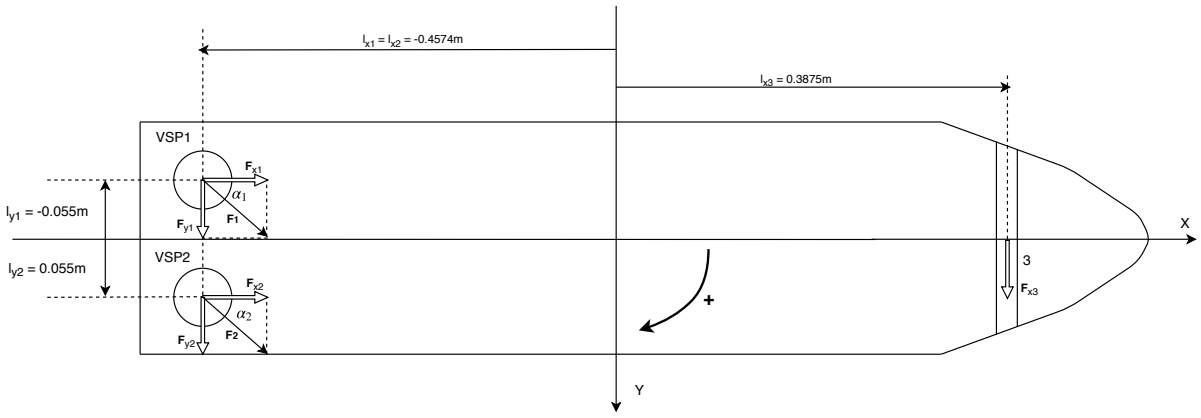


Figure 5.1: Thruster placement on CS Enterprise I

LOA	1.105 [m]
B	0.248 [m]
Δ	14.11 [Kg]

Table 5.1: Dimensions of CS Enterprise I

The low-speed control design model of CS Enterprise I is given by

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} &= -\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\tau} \end{aligned} \tag{5.1}$$

where $\boldsymbol{\eta}$ is the surge, sway and yaw position vector and $\boldsymbol{\nu}$ is the surge, sway and yaw velocity vector. $\boldsymbol{R(\psi)}$ is the rotation matrix, M is the vessel inertia matrix, $\boldsymbol{C(\nu)}$ is the Coriolis and centripetal matrix and $\boldsymbol{D(\nu)}$ is the damping matrix. $\boldsymbol{M}$ and $\boldsymbol{C(\nu)}$ are dependent on the rigid body and hydrodynamic added mass parameters, as well as hydrodynamic damping parameters. $\boldsymbol{D(\nu)}$ is a function of the hydrodynamic damping parameters. The necessary coefficients are given in Tables 5.2 an 5.3.

Table 5.2: CS Enterprise I rigid body and added mass parameters

Rigid body		**Added mass**	
Parameter	**Value**	**Parameter**	**Value**
m	14.11	$X_{\dot{u}}$	-2.00
I_z	1.76	$Y_{\dot{v}}$	10.00
x_g	0.04	$Y_{\dot{r}}$	-0.00
y_g	0.00	$N_{\dot{r}}$	-1.00

Table 5.3: CS Enterprise I damping parameters

Surge		**Sway**		**Yaw**	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
X_u	-0.66	Y_v	-1.33	N_v	0.00
X_{uu}	0.35	Y_{vv}	-2.78	N_{vv}	-0.21
X_{uuu}	-3.79	Y_{vvv}	-64.91	N_{vvv}	0.00
X_v	0.00	Y_r	-7.25	N_r	-1.90
X_{vv}	-2.44	Y_{rr}	-3.45	N_{rr}	-0.75
X_{vvv}	0.00	Y_{rrr}	0.00	N_{rrr}	0.00
–	–	Y_{rv}	-0.81	N_{rv}	0.13
–	–	Y_{vr}	-0.84	N_{vr}	0.08

5.1.2 Hardware architecture

The model ship is powered with a 12V12Ah battery on-board. The battery is mounted by connection of the positive and negative terminal to the vessel using wires. In addition, CS Enterprise I is equipped with an IMU, Inertial Measurement Unit from Analog Devices. The sensor type is ADIS16364, which includes a triaxis accelerometer and gyroscope. The coordinate frame of the IMU is left-hand orientation for linear accelerations and right-hand orientation for the angular rates. This is accounted for by multiplying the accelerations with -1. The control system onboard the scale model consists of four parts:

Compact re-configurable input/output (cRIO): cRio is an embedded controller provided by National Instruments. The ship model is equipped with the cRIO-9024 version, which is an embedded real-time controller, commonly used for advanced control and monitoring. The cRIO-90234 reads positioning data and is connected to 4 FPGA modules for I/O.

- NI-9215: Used for measuring voltage.
- NI-9263: Used for reading IMU measurements.
- NI-9401: Not used.
- NI-9474: Used for sending PWM signals.

Raspberry Pi (RPi): The RPi provides the communication with the Sixaxis controller. The Sixaxis controller transmits information from the joystick to the RPi. This is accomplished by Bluetooth communication. When the RPi is powered, it starts searching for a wireless controller. As soon as the RPi is successfully connected to a controller, the controller will output commands through the Ethernet to the cRIO.

Electronic Speed Control (ESC): The ESC controls the thruster motor speeds. These are controlled with Pulse-Width-Modulated (PWM) signals from the cRIO.

Four servos: The servos control the position of the VSP steering rods.

In addition to the on-board control system, a laptop is used in the communication system. The laptop reads simulated data and sends inputs to the cRIO based on outputs from the VeriStand Engine. The inputs are sent over MC Lab Wi-Fi. An illustration is displayed in figure 5.2. Additional information is found in Marine Cybernetics laboratory handbook [27].

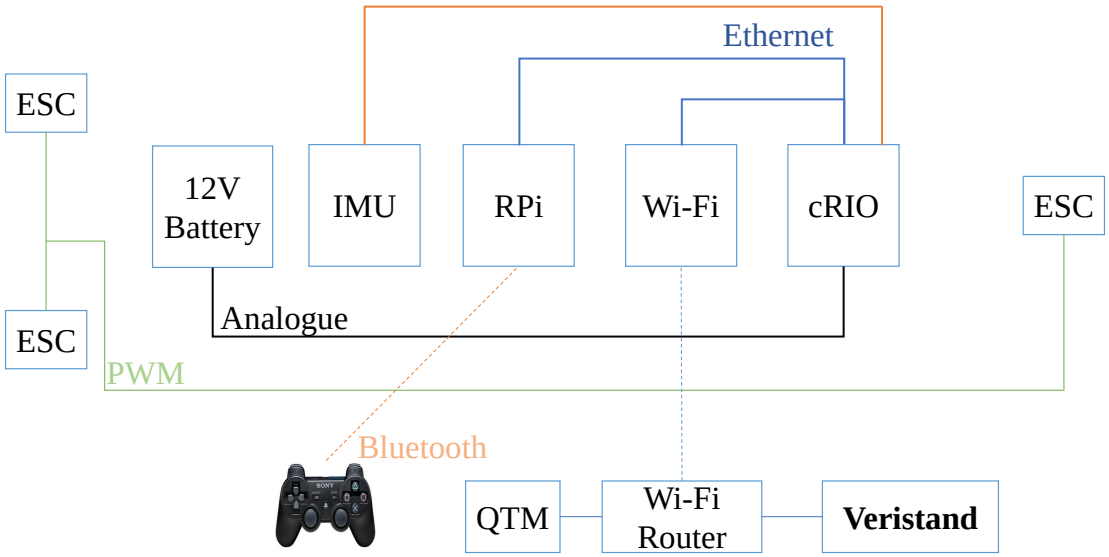


Figure 5.2: CS Enterprise hardware setup

5.1.3 Software architecture

Several software parts are needed in order to utilize the hardware architecture from section 5.1.2. This includes the MATLAB/Simulink system, the VeriStand software and the Qualisys Track Manager (QTM) software.

The MATLAB/Simulink system is developed at NTNU, and can be downloaded from GitHub. The different models include ctrl_custom, ctrl_DP, ctrl_sixaxis2thurster and u2.pwm, where ctrl_custom is the model used for this thesis. Figure 5.3 shows a block diagram of the elements in ctrl_custom, along with the elements added to facilitate for risk-based decision making. A Proportional Integral Derivative (PID) controller has been developed, which takes inputs from the observer and a reference model. The reference model smoothens the vessel setpoint to a realistic vessel path. The actuator system gets the commanded thrust from the PID, and communicates the allocated thrust to the plant. The plant models the vessel position, which is estimated by a sensor system. These noisy sensor measurements, along with the commanded thrust, are used in a Nonlinear Passive Observer (NPO) which produces estimated vessel positions.

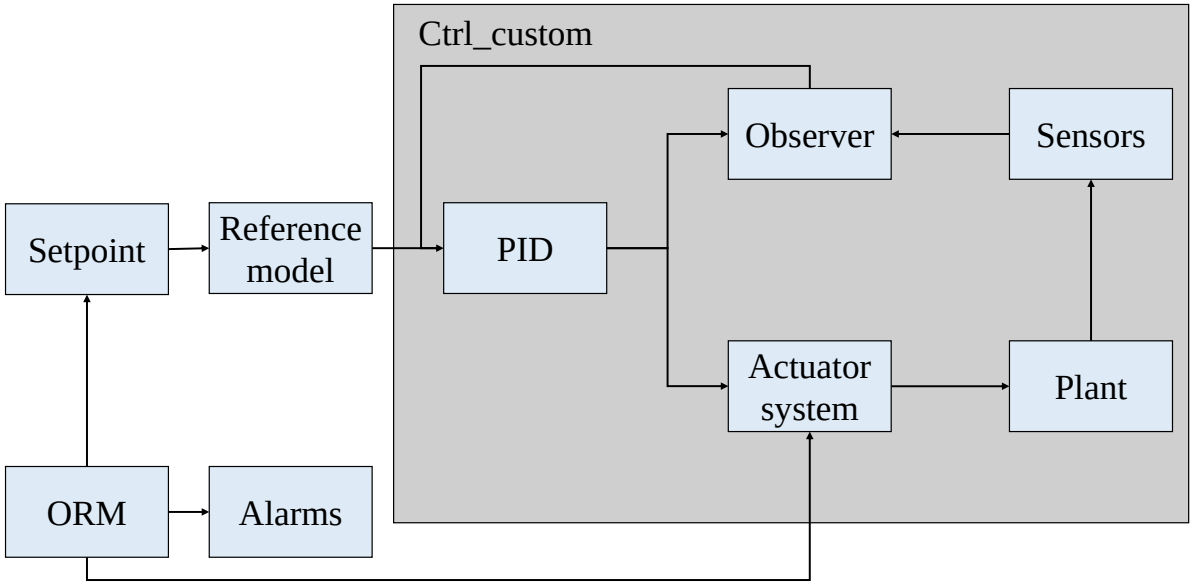


Figure 5.3: Representation of the ctrl_custom Simulink block diagram, along with elements added for the purpose of the thesis. ORM is short for Online Risk Model.

VeriStand is a software which can import control algorithms, simulation models and other tasks from a third-party environment. CyberShip Enterprise I is equipped with Veristand 2017 which is compatible with MATLAB 2016b. All MATLAB files are connected together in VeriStand. The QTM software provides vessel date such as position and orientation of the model ship in the laboratory over Wi-Fi. Additional information is found in CyberShip I user manual[24].

5.1.4 GeNIE and jSMILE

GeNIe is used in order to create the BBN, which is a graphical user interface based on SMILE. SMILE, structural modelling inference and learning engine, is developed in C++ and performs all calculations in GeNIe. The online risk model and control system developed in this thesis are programmed in Simulink and MATLAB. BayesFusion provides several fusions between SMILE and different programming languages. Consequently, a fusion written in Java called jSMILE is implemented, functioning as a bridge between MATLAB and SMILE. A tutorial on the implementation process is given on BayesFusion's website [3]. The connection between the hardware described above and SMILE was however unsuccessful, so experiments were done with

pre-calculated values from the BBN.

Chapter 6

Conclusions and further work

The aim of the thesis was reduced to the two following research questions:

1. *How can hardware-in-the-loop simulations assist the control design process, in specific the backstepping control design?*
2. *Is it possible to develop an online risk model based on risk of collision, in order to increase the situation awareness of a continuously unmanned ship during a DP operation?*

This Chapter provides an conclusion and answer to the research questions stated, followed by a suggestion to further work.

6.1 Concluding remarks

In case study I two different control systems for CS Enterprise I was developed and demonstrated with simulation in Simulink, HIL and the Marine Cybernetics laboratory. The first control system presented was DP maneuvering control design with backstepping. The controller was tuned throughout trial and error with Simulink and HIL simulations. It was experienced that simulation in Simulink and HIL made the tuning process surprisingly less demanding than anticipated. The experimental results in the laboratory showed that the backstepping control designed worked as desired, however some deviations from the results obtained in HIL. The second control system demonstrated, was linear PID. The experimental results showed that the PID controller was stable and robust. Throughout a discussion regarding the applicability of linear conventional PID and backstepping control design, it can be concluded that both control designs are of great importance. Backstepping is considerably more advanced, however by investing time in HIL simulation satisfactory results were achieved. In the process towards autonomous shipping, safe and robust systems are needed. Following, it is illustrated that HIL simulations contribute to reducing the time during commissioning as well as reduces the risk of software bugs in the control system.

In case study II a framework of an online risk analysis model has been presented, simulated and demonstrated in the Marine Cybernetics laboratory. The model is developed to display a method for assessing the risk of collision for an autonomous ship, followed by decision-making based on risk mitigating measures. Three different scenarios were tested and resulted in three different outcomes. The scenarios covered loss of situation awareness on the vessel, inadequate power supplied to the thrusters and harsh weather and unsatisfactory maintenance. Throughout the simulation and experimental testing, the results showed that the vessel successfully made decisions based on the probability of collision as desired.

6.2 Further work

In the process towards autonomous shipping, safe and robust systems are needed. A significant challenge is to combine the software and hardware to achieve reliable systems. In this thesis, the importance of hardware-in-the-loop simulation has been addressed. In addition, a framework for increased safety during decision-making. However, during the development of the thesis, several topics for further investigation emerged. The main proposal for further work are described in the following.

Suggestion for further work related to case study I:

- For the backstepping control design it was noticed a constant deviation in the heading. Hence, further work would be implementation of heading control.

Suggestion for further work related to case study II:

- For the Bayesian belief network, it would be interesting to including a larger amount of relevant risk influencing factors. It follows that the model would reflect the actual situation with increased accuracy.
- It would also be interesting to extend the possible risk mitigating measures. This suggests that the model would be more applicable and realistic.
- Situation awareness for autonomous ships involves uncertainty. Consequently, by testing the proposed framework on a ship with object detection, the online risk model would yield increased applicability of the model as the results obtained would be more realistic.
- It would also be interesting to include sea state estimation algorithms into the Bayesian belief network.

Bibliography

[1] Muhammad Juned Akhtar and Ingrid Bouwer Utne. "Human fatigue's effect on the risk of maritime groundings–A Bayesian Network modeling approach." In: Safety science 62 (2014), pp. 427–440.

[2] A.H Brodtkorb. "Hybrid Control of Marine Vessels." In: 2 (2017).

[3] John P Burkett. "A Bibliographic Guide to Methods for Causal Inference: Theory, Software, and Applications." In: (2017).

[4] W. C. Burmeister H. C.and Bruhn, Ø. J Rødseth, and T. Porathe. "Can unmanned ships improve navigational safety?" In: Proceedings of the Transport Research Arena, TRA 2014, 14-17 April 2 2014.

[5] "Definitions for Autonomous Merchant Ships." In: Norwegian Froum For Autonomus Ships (NFAS) (2017), pp. 1–21.

[6] D. Floreano and R. J Wood. "Science, technology and the future of small autonomous drones." In: Nature 521.7553 (2015), p. 460.

[7] T. I. Fossen. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, 2011.

[8] Martial H Hebert, Charles E Thorpe, and Anthony Stentz. Intelligent unmanned ground vehicles: autonom Vol. 388. Springer Science & Business Media, 2012.

[9] Jeevith Hegde et al. "A Bayesian approach to risk modeling of autonomous subsea intervention operations." In: Reliability Engineering & System Safety 175 (2018), pp. 142–159.

[10] HIL Testing Concept Explanation. URL: `https://www.dnvgl.com/services/hil-testing-concept-explanation--83385` (visited on 04/06/2019).

[11] E. Hiltunen and K. Hiltunen. "Technolife 2035: How Will Technology Change Our Future?" In: (2015).

[12] Finn V Jensen et al. An introduction to Bayesian networks. Vol. 210. UCL press London, 1996.

[13] S. K Jha. "Emerging technologies: Impact on shipbuilding." In: Maritime Affairs: Journal of the National M 12.2 (2016), pp. 78–88.

[14] Adit Joshi. Powertrain and Chassis Hardware-in-the-Loop (HIL) Simulation of Autonomous Vehicle Platf Tech. rep. SAE Technical Paper, 2017.

[15] H. K. Khalil. Nonlinear Control, Global Edition. Pearson, 2015.

[16] E. Lethovaara and K. Dr.Tervo. B0- a conditionally and periodically unmanned bridge. US Patent 8,346,578. 2018.

[17] Martin Ludvigsen and Asgeir J Sørensen. "Towards integrated autonomous underwater operations for ocean mapping and monitoring." In: Annual Reviews in Control 42 (2016), pp. 145–157.

[18] Robert Nilsson, Tommy Gärling, and Margareta Lützhöft. "An experimental simulation study of advanced decision support system for ship navigation." In: Transportation research part F: traffic 12.3 (2009), pp. 188–197.

[19] M. Rausland. Risk Assessment. 1st. WHILY, 2011.

[20] Ø.J Rødseth. Control centre for unmanned ships. 2014.

[21] Kailan Shang and Zakir Hossen. "Applying fuzzy logic to risk assessment and decision-making." In: Canadian institute of actuaries (2013).

[22] Roger Skjetne. "The maneuvering problem." In: NTNU, PhD-thesis 1 (2005).

[23] Roger Skjetne and Thor I Fossen. "On integral control in backstepping: Analysis of different techniques." In: Proceedings of the 2004 American Control Conference. Vol. 2. IEEE. 2004, pp. 1899–1904.

[24] Tran N.D Valle E. Bjørne E. Mykland A. Skåtun H.N. CyberShip Enterpise I User Manual. Faculty of Engineering Science and Technology Department of Marine Technology.

[25] Øyvind Smogeli. Marine Cybernetics. 2014.

[26] A. J. Sørensen. TMR4240 Marine Control Systems: Propulsion and Motion Control of Ship and Ocean St Department of Marine Technology, NTNU, 2013.

[27] L. Trevor. Marine cybernetics laboratory. 2017.

[28] I. B. Utne, A. J Sørensen, and I. Schjølberg. "Risk management of autonomous marine systems and operations." In: ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engi (2017).

[29] Skjetne R.. Vrn S.A. "Observer for simplifed DP model: Desgin and proof." In: NTNU 1 (2017).

[30] Anna Witkowska and Roman Smierzchalski. "Nonlinear backstepping ship course controller." In: International journal of automation and computing 6.3 (2009), p. 277.

[31] Alexander M Wyglinski et al. "Security of autonomous systems employing embedded computing and sensors." In: IEEE micro 33.1 (2013), pp. 80–86.

[32] Eray Yağdereli, Cemal Gemci, and A Ziya Aktaş. "A study on cyber-security of autonomous and unmanned vehicles." In: The Journal of Defense Modeling and Simulation 12.4 (2015), pp. 369–381.

Appendix A

Calculations

A.1 Thrust allocation in rectangular vector coordinates:

Explicit solution to the thrust allocation for f_{cmd}:

$$\begin{aligned} &\min_{f_{cmd}} && {f_{cmd}}^T f_{cmd} \\ &\text{subject to} && Bf_{cmd} = \tau_{cmd} \end{aligned}$$

By using lagrange multiplies and assuming that BB^T is non-singular, the explicit solution of the least-squares optimization problem is derived as follows:

$$\begin{aligned} L(f_{cmd}, \lambda) &= {f_{cmd}}^T f_{cmd} + \lambda^T(\tau_{cmd} - Bf_{cmd}) \\ \frac{\partial L}{\partial f_{cmd}} &= 2f_{cmd} - B^T\lambda = 0 \\ f_{cmd} &= \frac{1}{2}B^T\lambda \\ \tau_{cmd} &= Bf_{cmd} = \frac{1}{2}BB^T\lambda \end{aligned}$$

$$\begin{aligned} \lambda &= 2(BB^T)^{-1}\tau_{cmd} \\ f_{cmd} &= B^T(BB^T)^{-1}\tau_{cmd} \\ f_{cmd} &= B^\dagger\tau_{cmd} \end{aligned}$$

A.2 Observer without bias: Error dynamics

To derive the error dynamics, the observer dynamics (**??**) is subtracted from the vessel dynamics (3.15).

$$\begin{aligned} \dot{\hat{\boldsymbol{\eta}}} - \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)(\boldsymbol{\nu} - \hat{\boldsymbol{\nu}}) - \boldsymbol{L_1}\bar{\boldsymbol{\eta}} \\ \boldsymbol{M}(\dot{\boldsymbol{\nu}} - \dot{\hat{\boldsymbol{\nu}}}) &= -\boldsymbol{D}(\boldsymbol{\nu} - \hat{\boldsymbol{\nu}}) - \boldsymbol{R}^T(\psi)\boldsymbol{L_2}\bar{\boldsymbol{\eta}} + \boldsymbol{\tau} - \boldsymbol{\tau} \end{aligned}$$

Defining $\bar{\boldsymbol{\eta}} := \boldsymbol{\eta} - \hat{\boldsymbol{\eta}}$ and $\bar{\boldsymbol{\nu}} := \boldsymbol{\nu} - \hat{\boldsymbol{\nu}}$ give the error dynamics:

$$\begin{aligned} \dot{\bar{\boldsymbol{\eta}}} &= \boldsymbol{R}(\psi)\bar{\boldsymbol{\nu}} - \boldsymbol{L_1}\bar{\boldsymbol{\eta}} \\ \boldsymbol{M}\dot{\bar{\boldsymbol{\nu}}} &= -\boldsymbol{D}\bar{\boldsymbol{\nu}} - \boldsymbol{R}(\psi)^T\boldsymbol{L_2}\bar{\boldsymbol{\eta}} \end{aligned}$$

A.3 Control design

A.3.1 Kinematic model control design: differentiating V_1

$$\dot{V}_1 = z_1^T \dot{z}_1 = z_1^T \left(\dot{R}(\psi)^T (\eta - \eta_d(s)) + R(\psi)^T (\dot{\eta} - \eta_d^s(s)\dot{s}) \right)$$

$$\dot{V}_1 = z_1^T \left(rR(\psi)^T S(\eta - \eta_d(s)) + \nu - R(\psi)^T \eta_d^s(s)\dot{s} \right)$$

Using the definition of z_1:

$$\dot{V}_1 = z_1^T \left(rSz_1 + \nu - R(\psi)^T \eta_d^s(s)\dot{s} \right)$$

Since S is scew symmetric $z_1^T S z_1 = 0$:

$$\dot{V}_1 = z_1^T (\nu - R(\psi)^T \eta_d^s(s)\dot{s})$$

A.3.2 Kinematic model control design: implementing control law

By inserting $\alpha_1(\eta, s, t)$ (3.24) into equation (3.23), the following equation is obtained:

$$\dot{V}_1 = z_1^T (\nu - R(\psi)^T \eta_d^s(s)\dot{s})$$

$$\dot{V}_1 = z_1^T (-K_p z_1 + R(\psi)^T \eta_d^s(s) U_s(s,t) - R(\psi)^T \eta_d^s(s)\dot{s})$$

Using $V_1^s = -z_1^T R(\psi)^T \eta_d^s$ yields:

$$\dot{V}_1 = -z_1^T K_p z_1 - V_1^s (U_s(s,t) - \dot{s})$$

$$\dot{V}_1 \leq -\lambda_{min} K_p |z_1|^2 - V_1^s (U_s(s,t) - \dot{s})$$

A.3.3 Update Laws: Filtered gradiant update law

By inserting the speed assignment error, $\dot{\omega}_s$, into equation (**??**) and considering it as an additional state in the new CLF it is showed to be a solution to the maneuvering problem:

$$\dot{W}_1(\boldsymbol{\eta}, s, \omega_s) = \dot{V}_1 + \frac{1}{2\lambda\mu}(-2\omega_s\lambda(\omega_s - \mu V_1^s(\eta, s)))$$

$$\dot{W}_1(\boldsymbol{\eta}, s, \omega_s) = \dot{V}_1 + \frac{\omega_s^2}{\mu} - \omega_s V_1^s(\boldsymbol{\eta}, s)$$

$$\dot{W}_1(\boldsymbol{\eta}, s, \omega_s) \leq -\lambda_{min}(\boldsymbol{K}_p)|z_1|^2 - V_1^s(\boldsymbol{\eta}, s)(U_s - \dot{s}) - \frac{\omega_s^2}{\mu} + \omega_s V_1^s(\boldsymbol{\eta}, s)$$

$$\dot{W}_1(\boldsymbol{\eta}, s, \omega_s) \leq -\lambda_{min}\boldsymbol{K}_p|z_1|^2 - \frac{|\omega_s^2|}{\mu}$$

$$\dot{W}_1 \leq -\lambda_{min}\boldsymbol{K}_p|\boldsymbol{z}_1|^2 - V_1^s(\boldsymbol{\eta}, s)(U_s - \dot{s}) - \frac{|\omega_s^2|}{\mu} + \omega_s V_1^s(\boldsymbol{\eta}, s) \leq -\lambda_{min}\boldsymbol{K}_p|\boldsymbol{z}_1|^2 - \frac{1}{\mu}|\omega_s^2|$$

A.3.4 DP maneuvering control design: expressing $\dot{V}_1$

Inserting z_2 into equation (3.23) yields:

$$\dot{V}_1 = \boldsymbol{z}_1^T \boldsymbol{z}_2 - \boldsymbol{z}_1^T \boldsymbol{K}_p \boldsymbol{z}_1 - \boldsymbol{z}_1^T R(\psi)^T \boldsymbol{\eta}_d^s(s) \frac{\mu}{\boldsymbol{\eta}_d^s(s)} V_1^s(\boldsymbol{\eta}, s)$$

This is then inserted into the CLF along with the expression for V_1^s:

$$V_1^s = -\boldsymbol{z}_1^T \boldsymbol{R}(\psi)^T \boldsymbol{\eta}_d^s(s)$$

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

Inserting equation (3.28) into the expression the final CLF for the position error is expressed as:

$$\dot{V}_1 = \boldsymbol{z}_1^T \boldsymbol{z}_2 - \boldsymbol{z}_1^T \boldsymbol{K}_p \boldsymbol{z}_1 - \frac{\mu}{|\boldsymbol{\eta}_d^s(s)|} V_1^s(\boldsymbol{\eta}, s)^2$$