Master's thesis

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

Marthe Moengen

Autonomous Ships

Combining Hybrid Control and Machine Learning

Master's thesis in Marine Technology
Supervisor: Dong Trong Nguyen
June 2019

NTNU
Kunnskap for en bedre verden

Marthe Moengen

Autonomous Ships

Combining Hybrid Control and Machine Learning

Master's thesis in Marine Technology
Supervisor: Dong Trong Nguyen
June 2019

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

NTNU
Norwegian University of
Science and Technology

NTNU

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF MARINE TECHNOLOGY

MASTER THESIS

Autonomous Ships

Combining Hybrid Control and Machine Learning

Marthe Moengen

Supervised by

Dong Trong NGUYEN

Thursday 6th June, 2019

Abstract

This thesis investigates how one can enable a high level of autonomy for a ship moving the vessel from A to B without human intervention. A review of previous work has been conducted in order to get an overview of former implemented solutions, inspiring the conducted work presented in this thesis.

An advanced control system was developed based on hybrid control theory by developing a set of controllers and reference models for four different modes: Slow speed manoeuvring, Acceleration, Deceleration and Transit. A supervisory switch control was implemented in order to switch between the four modes while at the same time retain stability. A scale-independent hysteresis switching logic was chosen as this method guarantees stability also when switching between linear and nonlinear models.

In order to achieve a high level of autonomy, a deliberative level is needed for the system. Based on the definition of autonomy the vessel must be able to make justifiable choices in a challenging and complex environment. A path planning algorithm therefore was developed. This algorithm took advantage of the theory behind Artificial Intelligence (AI) and Machine Learning in order to calculate an optimal path for the vessel by producing a set of waypoints. The Reinforcement Learning algorithm Q-learning was used, taking into account the environmental forces from current in order to find the optimal path based on minimum time.

The overall hybrid control system and the path planner was tested and verified on the simulator MCSim developed in Matlab/Simulink and in the MCLab at the Department of Marin Technology, NTNU, using a 1:30 model of the supply ship Cybership 3. The resulting system manages to follow the calculated optimal path in a satisfactory manner, switching between the four modes without human intervention. The optimal waypoints found by the path planner also result in a faster path than the shortest distance path. Hence, the concept of the overall autonomous system was verified.

Sammendrag

Denne masteroppgaven utforsker hvordan man kan oppnå en høy grad av autonomi for et skip ved å å utvikle et kontrollsystem som fører skipet fra A til B uten menneskelig innblanding. Et litteraturstudie av tidligere forskning på feltet har blitt gjennomført for å gi en oversikt over prøvde løsninger og for å inspirere arbeidet i denne masteroppgaven.

Et hybrid kontrollsystem bestående av et sett av kontrollere og referansemodeller ble utviklet. Fire forskjellige modus ble definert for en seilas: Manøvreing i sakte fart, Akselerasjon, Deakselerasjon og Transit. En supervisory switch control ble implementert for å bevare stabiliteten til kontrollsystemet i det man bytter mellom de fire modusene. En scale-independent hysteresis switching logikk ble valgt da denne metoden garanterer stabilitet også når man bytter mellom lineære og ulineære modeller.

For å oppnå en høy grad av autonomi for systemet er et intelligent nivå nødvendig. Basert på defenisjonene for autonomi må et skip kunne ta begrunnede beslutninger i et utfordrende og komplekst miljø. En baneplanleggingsalgoritme ble derfor utviklet for å utgjøre dette nivået. Denne algoritmen tok utgangspunkt i kunstig intelligens og maskinlæring for å beregne en optimal bane for skipet ved å produsere et sett med rutepunkter. Reinforcement learning teknikken Q-learning ble brukt ved å ta inn miljøfaktoren strøm for å beregne den optimale ruten med hensyn på tidsbruk.

Det hybride kontrollsystemet og ruteplanleggeren ble testet med simulatoren MCSim utviklet i Matlab/Simulink og på MCLab på Marinteknisk senter, NTNU, ved å bruke en 1:30 skala av modellskipet Cybership 3. Det resulterende systemet klarer å følge den beregnede optimale ruten på en god måte og bytter mellom modusene samtidig som stabilitien i systemet holdes inntakt uten menneskelig innblanding. Resultatene fra testene viser at de optimale rutepunktene utgjør en raskere rute enn en rute basert på korteste vei. Konseptet til det autonome systemet kan derfor vurderes som vertifisert.

Preface

This master thesis is written spring 2019 at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU) in Trondheim. The thesis is a continuation of a project thesis written fall 2018 on the same subject and wraps up a five-year master program specialising within the field of Marine Cybernetics.

While being a master student in the field of Marine Technology, I have observed a slow and rooted maritime industry being forced to innovate. This development is very inspiring and I have actively pursued cybernetics and computer technology through my master to be able to participate in the digital shift that is taking place worldwide.

This master thesis is therefore inspired by the powerful combination of computer science and cybernetics in the maritime industry. Hence, the thesis combines the domain of artificial intelligence and machine learning with advanced control systems used for marine applications.

Acknowledgments

I would like to express my gratitude towards my supervisor Dong Trong Nguyen, for guidance and suggestions throughout the project period. I would also like to thank Torgeir Wahl for technical support during the lab experiments and my co-student Thomas Johansen for interesting academic discussions and collaboration in the lab. I must also to thank Erlend Andreassen for making sure I was never malnourished during the project period.

Trondheim, Thursday 6th June, 2019

Marthe Moengen

Table of Contents

List of Tables

List of Figures

Abbreviations

DP	Dynamic Positioning
NPO	Nonlinear Passive Observer
PM	thruster assisted Position Mooring
SPM	Self-Propelled Model
ND	Near field Diagram
AAWA	Advanced Autonomous Waterborne Applications Initiative
MUNIN	Maritime Unmanned Navigation through Intelligence in Network
PID	Proportional Integral Derivative
MCU	Motion capture unit
JONSWAP	Joint North Sea Wave Project
LOS	Line-of-Sight
LOA	Levels of Autonomy
AI	Artificial Intelligence
RL	Reinforcement Learning
Q-learning	Machine learning algorithm
MCSim	Marine Cybernetics Simulator
MCLab	Marine Cybernetics Laboratory

Symbols

$\{n\}$	=	Denotes the North-East-Down frame
$\{b\}$	=	Denotes the BODY-xed frame
$\{e\}$	=	Denotes the Earth-Centered Earth-Fixed frame
$\boldsymbol{\eta}$	=	Position and attitude vector
$\boldsymbol{\nu}$	=	Linear and angular velocity vector
$\boldsymbol{\tau}$	=	Generalized forces and moments vector
$\boldsymbol{p}$	=	Position vector
$\boldsymbol{\omega}$	=	Angular velocity vector
x, y, z	=	Positions in $\{$e$\}$
N, E, D	=	North, East, Down positions in $\{$n$\}$
u, v, w	=	BODY-xed linear velocities
ϕ, θ, ψ	=	Euler angles about the x, y and z-axis
$\boldsymbol{M}$	=	Inertia matrix
$\boldsymbol{M}_{RB}$	=	Rigid-body inertia matrix
$\boldsymbol{C}_{RB}$	=	Rigid-body Coriolis-centripetal matrix
$\boldsymbol{M}_A$	=	Added mass Coriolis-centripetal matrix
$\boldsymbol{D}$	=	Damping matrix $\boldsymbol{\Delta}$ = Relative damping ratio matrix (boldface)
$\boldsymbol{\Omega}$	=	Natural frequency matrix
$\boldsymbol{x}_d$	=	Reference model desired state vector
$\boldsymbol{r}^b$	=	Operator input in $\{$b$\}$
p_k^n	=	Position of waypoint k in $\{$n$\}$
α_k	=	Path-tangential angle
ϵ	=	Along-track distance and cross-track error vector
e	=	Cross-track error
s	=	Along-track distance
χ_d	=	Desired course angle
χ_P	=	α_k
Δ	=	lookahead distance
ψ_{los}	=	Heading angle from LOS Guidance System
β	=	Sideslip angle
s_{k+1}	=	Distance from waypoint k to k + 1
R_{k+1}	=	Radius of acceptance of waypoint k + 1
$\boldsymbol{u}$	=	Control input to actuator(s)
α_i	=	Thruster angles in $\{$b$\}$
l_{xi}	=	Length to thruster i in x-direction in $\{$b$\}$
l_{yi}	=	Length to thruster i in y-direction in $\{$b$\}$
ω_n	=	Natural frequency
ζ	=	Damping ratio
K_P	=	Proportional gain
K_I	=	Integral gain
K_D	=	Derivative gain
K	=	Nomoto gain
T	=	Nomoto time constant

b	=	Slowly varying unknown bias
w	=	Model noise
v	=	Measurement noise
H_s	=	Signicant wave height
ω_p	=	Peak wave frequency
μ_p	=	Monitoring signal in scale-independent hysteresis switching
σ	=	Switching signal
h	=	Positive hysteresis constant
λ	=	Non-negative forgetting factor
$\boldsymbol{e}_p$	=	Error vector
γ	=	Class $\mathscr{K}$-function
γ_Q	=	Discount factor
α_Q	=	Learning rate

Chapter 1

Introduction

This chapter will introduce the master thesis. A general introduction to the main subjects and a presentation on the motivation behind this study will follow. Then the previous work on the main topics will be evaluated through a literature review. Finally, the objective and scope in addition to the main contribution of the thesis are presented, before the organization of the project is displayed.

1.1 Background

1.1.1 General Introduction

This master thesis is inspired by the ongoing change that has been seen over the last couple of years in the maritime industry. Digitalization is providing new opportunities at the same time as the industry is experiencing a push towards developing greener solutions and taking advantage of new technologies. One such technology is the newly expanding field of autonomous ships. Autonomous ships are regarded as vessels that can perform navigation on its own, without the interference of a human operator. In order to achieve this, both the combination of an advanced automatic system for the control execution layer and a level of intelligence for the mission planning layer is necessary.

One approach in making these vessels autonomous is by taking advantage of hybrid control theory for the automatic system in the control execution layer. This approach exploits a bank of controllers based on some switching logic. Hence, the control system can choose the best-suited controller based on observed measurements. Conventional control systems as DP will not provide the desired performance or robustness, as it will be optimized for only one environmental or operational condition. However, a hybrid control system will be qualified for a number of different environmental conditions and operations, moving the ship towards autonomy.

Additionally, in order to achieve the deliberate level for the mission planning layer the vessel must be able to make justifiable decisions in complex environments, tackling unforeseen situations. This is necessary in order to reduce the risk of collision and accidents when traveling without a human operator. An intelligent guidance system that generates the desired path based on input from the environment could be an approach to achieve this. There are numerous ways to implement such systems. A field that has harvested a lot of attention recently is Artificial Intelligence (AI) and Machine Learning. AI contains the study on intelligent agents that can perceive its environment and take actions to maximize the chance of reaching a goal or desired performance. Consequently, in order to realize an intelligent autonomous ship

will a hybrid control system be implemented together with a machine learning based algorithm for the generation of an minimum-time optimal path.

1.1.2 Motivation

The motivation for implementing intelligent advanced control systems in the marine industry are plentiful. The hybrid control system will ensure the performance and stability of the control system for different environmental conditions and through different parts of the operation. In addition, the intelligent guidance system will contribute to the reduction of risk and potentially reduced cost and emissions if fuel consumption also is taken into account. In a world struggling with greenhouse gasses and global warming the implementation of such systems can be important. Furthermore, the field of artificial intelligence poses unlimited possibilities that have yet not been fully exploited by the marine industry. The expected benefits of autonomous ships combined with an intelligent guidance system in comparison to conventional ships are therefore reduced risk to the operator, reduced cost, and increased energy efficiency.

1.2 Litterature Review

1.2.1 Autonomous ships

The interest in autonomous ships has increased during recent years. In order to understand this domain, it is necessary to define the concept of autonomy. A system is autonomous if an automatic system also has a level of deliberation. Autonomy demands systems to make justifiable choices in complex environments on its own, without human interference. This introduces the necessity to define to what degree such a system is autonomous; Levels of Autonomy (LOA). LOA defines the characteristic relationship between human and automation. A robot is never either autonomous or not, as a combination of the two is possible. There exsist numerous definitions of LOA. Sheridan and Verplank (1978) introduced a 10-level spectrum of autonomy ranging from the human performing the task completely, to the machine performing the task completely. Another 10-level definition was described in the book *Autonomous vehicles in support of naval operations*, National-Academies-Press (2005), defined by the US Army. Here, the systems awareness of the situation, decision-making ability and capability to steer the vessel was taken into account when defining the degree of autonomy. A four-level definition has also been defined by the US National Institute of Standards and Technology, described in Sørensen (2018):

1. Automatic operation (remote control)
2. Management by consent (teleoperation)
3. Semi-autonomous or management by exception
4. Highly autonomous

Hence, the levels of autonomy function as way of defining to what degree the robot or automatic system is working in combination with a human operator.

Pietrzykowski and Malujda (2018) defined the concept of autonomous ships as a ship navigating on its own, without human supervision or intervention. The concept of autonomy has been utilized for many years in the maritime industry. On such example is the Dynamic Positioning (DP) system of a ship, that through control systems on the actuators keeps the vessel at the desired set point. DP systems have been commercially available since the late 1960s, Hassani et al. (2018b). However, one could argue that the commercial DP systems are lacking the deliberate level, and should only be considered automatic.

The Advanced Autonomous Waterborne Applications Initiative (AAWA) project, Rolls-Royce (2016), is an example of an European project that was aiming to develop the technology needed to enable autonomous ships. They also looked into the safety and security aspects as well as social and legal acceptance. Another example of such a project is the Maritime Unmanned Navigation through Intelligence in Network (MUNIN) project, MUNIN (2016), that aimed to develop and verify the concepts of autonomous ships.

However, the competition on being the first to develop a complete, well performing and safe autonomous ship is tightening. 3rd of December 2018 Rolls Royce, TU.no (2018a), announce that they had completed the worlds first fully autonomous ferry operation, where the Finnish ferry Falco had sailed with an autonomous system for 400 hours. This system made decisions on course, speed and collision avoidance based on sensory information with the aid of a computer program that processes and analyses the environment and surrounding traffic. Another ongoing project is the launch of the autonomous vessel Yara Birkeland by Kongsberg Maritime, that is to carry the cargo equivalent to 40.000 trailers a year on the coast of Norway, TU.no (2018b). By looking into these examples there is no doubt that the interest for enabling such autonomous ships are growing in today's global maritime industry.

Tessier argue in Lawless (2017) that human operators result in larger risk in unforeseen situations as when facing challenging situations, humans can postpone the decision or ask for further information, while Thieme and Utne (2017) argue that the expected benefits of autonomous marine systems are reduced cost, reduced risk to operators, and increased efficiency of such systems. These arguments highlight the motivation for introducing autonomous ships.

1.2.2 Hybrid Control

Hybrid control is based on a control system that can switch between a bank of controllers based on some switching logic in order to achieve the optimal performance. This control system can be defined as a hybrid dynamical system by the theory found in Goebel (2012). There are a number of ways to define this switching logic. In the work conducted by Hunt et al. (1996) is the switching based on a smoothly interpolating controller for an autonomous vehicle. However, the smooth switching approach might not yield a stable system when switching between nonlinear controllers. Supervisory switch control is another alternative proposed by Hespanha (2001), Hespanha et al. (2003) and Hespanha and Morse (2002). Supervisory switch control ensures stability with the use of dwell-time for linear supervisory control and by scale-independent hysteresis switching for nonlinear supervisory control. In the estimator-based supervision, the supervisor compares a set of models to the actual process and decides which model is best to describe the ongoing process.

Nguyen et al. (2007) used the hysteresis switching method defined in Hespanha (2001) to switch between a set of controllers in order to improve the performance of a DP system in varying environmental conditions. A set of observers estimated the peak period ω_p of the experienced waves for a model ship and used this to switch between controllers tuned for a sea-state defined by ω_p. In the work conducted by Hassani et al. (2018b) is a Dynamic Hypothesis Testing (DHT) approach utilized in order to decide what controller should be chosen by the supervisor. For detection of mooring line breakage in a thruster assisted position mooring (PM) system, a set of hypotheses defined based on different scenarios of line breakage. At each sampling time, the inputs and outputs was used to generate a conditional probability of each hypothesis being true. The supervisor then chose the best suited controller for this scenario based on these probabilities.

Another approach in defining the switching logic, is to base it on the unfalsied control methodology. This switching concept is based on measured data to asses the performance of a set of controllers. The unfalsified control theory can be found in Stefanovic and Safonov (2011). This method evaluated the

performance of all the controllers in real time, also the ones not currently in the loop. This data driven control mythology was utilized by Hassani et al. (2018a) on a set of controllers for use in DP of marine vessels subjected to environmental forces. Here, the concept of virtual reference was used to calculate a reference point for the controllers not currently in the loop. The bank of controllers was constructed so that at least one controller should meet the desired performance specification. A cost function was implemented to evaluate the performance of each controller in real time where the controllers were falsified until ultimately one optimal controller remained.

Trong Dong Nguyen et al. (2008) described a methodology in hybrid control by combining dynamic positioning, manoeuvring and transit. The vessels operational condition was in this paper dependent on changes in the operational mode, speed and environment. Here the operation was divided into the five modes: transit, transformation from transit to station keeping, DP, transformation from DP to PM, and PM. A case study was conducted for a shuttle tanker going from transit to station keeping at a fixed environmental condition, varying speeds and modes. In transit mode, an autopilot PID controller for heading control was used while the DP mode used a PID controller controlling the position and heading of the vessel. For the transition from/to autopilot to/from DP, the DP and autopilot controller was combined using a weighting function. However, the switch was made manually by an operator.

Hybrid control was also used by Mirosław (2017) to obtain an autonomous ship by conducting experiments on a training ship in Poland. A hybrid control system consisting of four modes and five controllers was developed with the use of supervisory switch control. The ship was required to complete the operating modes of manoeuvring the ship at low speeds, steering the ship at different speeds in the course or along a trajectory, and stopping the ship on the route. To accomplish this, five controller was developed, where the used controller was chosen by the supervisory control system described by Hespanha (2001). For the low-speed mode, a multi-variable precision controller was developed, where the aim of the controller was to track the desired smooth trajectory defined by the operator. To stop the ship, a multivariable low-speed controller was designed using state feedback linearization. To steer the ship along a trajectory with different speeds, a track-keeping controller based on the method described in Holzhüter (1990) was used to control the rudder angle, and a surge speed PI controller was used to control the speed. Finally, the ship motion on the course was controlled by a course-keeping PID controller. For this setup, the supervisor operates based on a reference trajectory obtained by the system operator, where certain operation modes were assigned to each trajectory segment. The hybrid control system described in Mirosław (2017) is therefore not considered fully autonomous as the need for an operator is present to define the trajectory.

In Norway a hybrid control system for the Anda-Lote ferry crossing in Sogn og Fjordane was developed by Torben (2018). The article considers three modes for the hybrid system; transit, transition and docking. The transit mode is solved by using a PID controller to control the heading and a PI controller to control the speed of the vessel. For the transition model the goal is to decelerate the speed from service speed to low speed and position the ferry for docking. Also here a PID heading controller was used in combination with an Adaptive Line-of-sight guidance law and a cubic speed interpolation to determine the reference speed when going from service speed to low speed. In the docking mode, a nonlinear PID control law with reference feed-forward on velocity and acceleration was used. The system was tested in a high fidelity simulation environment with a model of the Anda-Lote ferry, resulting in a promising performance.

1.2.3 Path Planning

Path planning is the approach of finding the most suitable path for a vessel based on some input. To solve this issue, there exist numerous path planning algorithms. Fossen (2011) presents an approach for path following for straight lines path following. This method is using a waypoint database to generate a desired trajectory or path for the vessel to follow. The generation of these waypoint databases is usually conducted by algorithms based on different criteria like the mission, environmental data, collision avoidance, feasibility, etc.

A normal problem formulation is to find the shortest path on a given map. The map might be gridded to define a set of nodes the agent can visit. For a marine application, this agent can be a ship or ferry. Wang et al. (2011) propose the conventional grid-based search algorithm Dijkstra in their paper for solving the shortest path of a maze robot. Another approach is the widely used A* algorithm that also includes the cost to reach a node. Thus, if the goal is to find the cheapest path from a node to the goal, this algorithm presents a good alternative. The algorithm and underlying theory are described further in Mohammed et al. (2016).

However, these algorithms only suggest a solution for a rigid environment based on an observable map. Garau et al. (2005) extended the application of the A* algorithm by including the effects of currents in order to make it account for the environmental forces to find the minimum-time path for an AUV. Another advancement of the A* algorithm is the extension Theta* algorithm that was used by Kim et al. (2012) to plan the most desirable path. Most path planning algorithms for surface vessels have been developed only in an (x,y) plane. This makes it difficult to follow the path, as the vessel's heading angle ψ is not considered. Kim et al. (2012) expands the field of path planning as their article incorporate the focus of the heading angle for a surface vessel in an ocean environment.

Many path planning algorithms are based on machine learning techniques when searching for optimal solutions. The A* algorithm is, for instance, a type of informed (heuristic) search strategy. Hence, the field of machine learning and artificial intelligence is pushing the performance of these algorithms and AI is playing an increasingly important role.

1.2.4 Machine Learning

Machine learning is one of the branches of Artificial Intelligence (AI) that is used to improve the performance of computer systems, giving them the ability to learn without programming the algorithms explicitly. Machine learning has strong bonds to computational statistics and AI makes use of mathematical optimization and prediction, Russell (2016). By utilizing the power of machine learning tools in the maritime industry one could enable new solutions.

Yoo and Kim (2016) has combined the theory of path planning and machine learning to develop an algorithm that takes the ocean current into account when generating the desired path. The path planner is based on the reinforcement learning algorithm, which is a technique within the field of machine learning. This method calculates the optimal control sequence from interacting with the environment during the learning process. The motion of the vehicle is modelled in the two-dimensional plane, taking into account the constraints of the turning rate. The position is discretized into tiles on the map, while the heading of the vehicle is discretized into finite intervals. The action space was also discretized to going straight, turning right, or turning left. The cost to be minimized in this paper is the time to reach the goal position. The penalty is defined as the amount of time the agent remains in each tile. The algorithm was tested and the results were compared to a known analytical solution, and the effectiveness of the algorithm was confirmed.

Yang et al. (2014) wanted to explore the possibilities of taking advantage of the Q-learning method to improve safety when developing a path planning algorithm while taking into account environmental forces as well as various port facilities, navigation facilities, and the ships nearby. A large-scale self-propelled model (SPM) in a real marine environment was used to conduct the experiment. The paper introduces offline learning to a collision avoidance algorithm based on the local information perception called the near field diagram (ND) method. The Q-learning algorithm is adapted by making the action selection strategy random by adding a chaos operator to achieve semi-randomness behaviour. The Q-learning algorithm is used to compensate for the effect of the environmental forces by giving penalties for large deviations in the desired heading angle. The chaos algorithm helps explore the state space, but after a period of learning is the chaos algorithm replaced by a $epsilon$-greedy selection strategy to exploit the information from the exploration phase. Simulations and sea tests concluded that the algorithm performs well for collision avoidance under the interference of winds.

1.3 Objective

This literature review displays some of the work done within the field of autonomous ships, hybrid control and path planning combined with machine learning. Advanced and automatic DP systems has been developed and big companies within the maritime industry are working to push these systems towards more intelligent solutions. However, the utilization of hybrid control system for autonomous crossing and docking of a vessel is limited, and there exists little literature on a full implementation for a complete operation that fully removes the need of a human operator. The main purpose of this master thesis will, therefore, be to narrow the gap between the automatic hybrid control system and the autonomous intelligent ship. The work committed by Mirosław (2017) will be extended by making the supervisor, switching logic and choice of optimal trajectory independent of a human operator. In addition, an intelligent path planner will be developed inspired by the work described in Yoo and Kim (2016), to remove the need for an operator when defning the waypoint database for the path.

The objective of this project thesis is, therefore, to push the control structure of ships towards higher levels of autonomy by exploiting the stability and robustness of a hybrid controller, together with the intelligence of a machine learning based path planner in order to optimize the performance and fully remove the need for human interference.

1.4 Scope

The scope of this project thesis will be to set up mathematical representations of the vessel dynamics (surge, sway and yaw) that can be used for autocrossing and autodocking of a vessel. For the implementation of the hybrid control system, a set of controllers will be developed to manoeuvre the vessel from port A to B, using supervisory switch control.

In addition, a Q-learning algorithm will be developed to find the optimal path for the vessel based on environmental data and to evaluate the concept and potential of AI in marine applications.

The combined intelligent system will be validated and tested through simulations in MCSim and experiments in the MCLab by using a model of the supply vessel Cybership 3. Both systems and the overall implementation will be developed using Matlab/Simulink.

1.5 Main Contribution

The main contribution from this master thesis is to showcase the potential that lies within combining the domain of machine learning with advanced control systems in order to achieve high levels of autonomy for ships. This work is verified by both simulations and lab experiments.

Additionally, the tests that were conducted in the MCLab required environmental forces from current and a full exploitation of the basin length. In order to achieve this, a set of new methods were developed to enable these features in the lab. Hence, the contribution of these methods provides the MCLab with a wider area of application for testing and verification of a variety of systems in the future.

1.6 Organization of Project

This master thesis presents an introduction to the background concepts and a collection of papers in Chapter 1 together with the motivation, objective, scope and main contribution of the thesis. Chapter 2 contains the theory, explaining the underlying mathematical background for the following implementations. Chapter 3 presents the methods used to develop the hybrid control system and path planner in addition to the setup for the test in simulation and lab. Chapter 4 presents the simulation results for both the hybrid controller and the machine learning based path planning algorithm, while Chapter 5 presents the experimental results from the lab. Chapter 6 will discuss these findings and the implementation of the autonomous system, while Chapter 7 concludes the project, summing up the major findings and suggesting future work.

Chapter 2

Theory

This chapter explains the mathematical background and theory used in the work of this thesis. A mathematical description of the vessel with high fidelity and a simplified model will be presented. Then an observer and a set of controllers used for both manoeuvring and DP operations are described before three guidance laws are explained. The theory behind hybrid system modelling with supervisory switch control will also be described. Finally, a machine learning based path planning algorithm will be presented.

2.1 Process Plant Model

The process plant model is a high fidelity model of the marine vessel used for simulation purposes. It is an extensive description of the process plant model and is describing the real motion of the vessel. The purpose of the model is to provide a simulated environment for testing of implemented control systems, Sørensen (2013). The 6 DOF body-fixed nonlinear low-frequency vessel model can be written as

$$\boldsymbol{M}\dot{\nu} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}_{A}(\boldsymbol{\nu_r})\boldsymbol{\nu_r} + \boldsymbol{D}(\kappa, \boldsymbol{\nu}_r) + \boldsymbol{G}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{thr} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{waves} + \boldsymbol{\tau}_{cur} \tag{2.1}$$

where $\boldsymbol{M}$ is the mass including added mass, $\boldsymbol{C}_{RB}$ and $\boldsymbol{C}_{A}$ are the rigid body and added mass coriolis and centripetal matrices, $\boldsymbol{D}$ is the generalized damping matrix consisting of both a nonlinear and linear component such that $\boldsymbol{D} = \boldsymbol{D_L} + \boldsymbol{D_{NL}}$, and $\boldsymbol{G}$ is the generalized restoring matrix. $\boldsymbol{\eta}$ is the generalized position, $\boldsymbol{\nu}$ and $\boldsymbol{\nu_r}$ is the generalized velocity and relative velocity respectively and $\boldsymbol{\tau}_{thr} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{waves} + \boldsymbol{\tau}_{cur}$ represents the external environmental and thruster forces acting on the vessel. More detailed description on these matrices can be found in Sørensen (2013).

2.2 Control Plant Model

The control plant model is a simplification of the process plant model, and is used to describe the motions of the vessel in 3 DOF; surge, sway and yaw. For the purpose of the model-based observer and controller, it is sufficient to use this simplified mathematical model as it describes the main physical characteristics of the system, Sørensen (2013). In addition, as the model is used in the model-based observers and controllers it needs to be computationally efficient. As the environmental forces, operational objectives and vessel types vary, the control plant model takes different forms. A description of a control plant model for a DP operation and an autopilot transit operation will follow.

2.2.1 DP Model

The control plant for a DP model with 2nd-order mean and slowly varying wave, current and thruster forces is defined in equation 2.2 as in Sørensen (2013). This is a simplification of the Process Plant Model, adding a bias term, the wave frequency model and transforming the positions to the NED-frame. This results in a state-space model $\mathbf{x} = [\boldsymbol{\xi}^T, \boldsymbol{\eta}^T, \mathbf{b}^T, \boldsymbol{\nu}^T] \in \mathbb{R}^{15}$, $\boldsymbol{\tau} \in \mathbb{R}^3$ and $\mathbf{y} \in \mathbb{R}^3$.

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_w \boldsymbol{\xi} + \mathbf{E}_w \mathbf{w}_w \tag{2.2a}$$
$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.2b}$$
$$\dot{\mathbf{b}} = -\mathbf{T}_b^{-1}\mathbf{b} + \mathbf{E}_b \mathbf{w}_b \tag{2.2c}$$
$$\dot{\boldsymbol{\nu}} = -\mathbf{M}^{-1}\mathbf{D}\boldsymbol{\nu} + \mathbf{M}^{-1}\mathbf{R}^T(\psi)\mathbf{b} + \mathbf{M}^{-1}\boldsymbol{\tau} \tag{2.2d}$$
$$\mathbf{y} = \boldsymbol{\eta} + \mathbf{C}_w \boldsymbol{\xi} + \mathbf{v} \tag{2.2e}$$
$$\tag{2.2f}$$

The wave frequency model represents the 1st-order wave induced response of the vessel. $\mathbf{A}_\omega$, $\mathbf{E}_w$ and $\mathbf{C}_w$ is modelled as in equation 2.3 where ω_p is the peak wave frequency and λ is the damping ratio, typically in the range 0.05-0.1, Sørensen (2013). These parameters are chosen according to the sea state of the operation of the DP system. In the North Sea, such sea states often are described by the JONSWAP wave spectrum. The wave frequency model is also used in model-based estimators where the goal is to filter out the effect of the wave frequency vessel motion. This is done to avoid wear and tear on the actuators that will try to compensate for this oscillatory first order wave motion. Such observers will be presented in Section 2.3.

$$\begin{aligned}
\mathbf{A}_w &= \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \\ -\boldsymbol{\Omega}^2 & -2\boldsymbol{\Lambda}\boldsymbol{\Omega} \end{bmatrix} \in \mathbb{R}^{6x6} \\
\mathbf{C}_w &= \begin{bmatrix} \mathbf{O}_{3x3} & \mathbf{I}_{3x3} \end{bmatrix} \in \mathbb{R}^{3x6}, \mathbf{E}_w = \begin{bmatrix} \mathbf{0}_{3x3} \\ \mathbf{I} \end{bmatrix} \in \mathbb{R}^{6x3},
\end{aligned} \tag{2.3}$$

$$\begin{aligned}
\boldsymbol{\Omega} &= \text{diag}\{\omega_1, \omega_2, \omega_3\} \\
\boldsymbol{\Lambda} &= \text{diag}\{\lambda_1, \lambda_2, \lambda_3\}
\end{aligned}$$

The bias b represents unmodelled slowly varying force dynamics as current, wave drift and modelling errors. In equation 2.2c, the bias is modelled as a first-order Markov process where $\mathbf{T}_b \in \mathbb{R}^{3x3}$ is a diagonal matrix of bias time constants and $\mathbf{w}_b \in \mathbb{R}^3$ is zero-mean Gaussian white noise.

Equation 2.2d is a simplification of the low-frequency Process Plant Model in equation 2.1. Here the damping is simplified to only hold a linear damping term in $\mathbf{D}$. As there is no mooring considered for this control plant model and hence no stiffness, $\mathbf{G} = 0$. Additionally is $\mathbf{C}_{RB}(\nu)\nu$ and $\mathbf{C}_A(\nu_\mathbf{r})\nu_\mathbf{r}$ neglected as the velocities ν and $\nu_\mathbf{r}$ are considered small for DP.

Equation 2.2b represents the 3DOF kinematics where the velocities are transformed from body-fixed frame to NED-frame using the rotation matrix $\mathbf{R}(\psi)$ defined as

$$\boldsymbol{R(\psi)} = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Equation 2.2e represents the measurements of the system where $\mathbf{C}_w$ is defined as in equation 2.3. $\mathbf{y}$ is the sensor output where $\boldsymbol{\eta}$ is the vessel position, $\boldsymbol{\eta}_w = \mathbf{C}_w\boldsymbol{\xi}$ is the position and orientation measurement vector and $\mathbf{v}$ is the zero mean Gaussian measurement noise.

2.2.2 Heading Autopilot

For autopilot operation, the Nomoto model can be used as the control plant model. The Nomoto model is a manoeuvring model that represents the yaw dynamics of a marine craft. The model describes the yaw rate r response due to the change in rudder angle δ. The model is decoupled in sway-yaw and is derived from the following yaw subsystem described in Fossen (2011) as

$$\mathbf{M}\dot{\nu} + \mathbf{N}(u_0)\nu = \mathbf{b}\delta \tag{2.5}$$

The 1. order Nomoto model in equation 2.8 is obtained from the 2. order Nomoto model described in equation 2.6 by defining the equivalent time constant in equation 2.7.

$$\frac{r}{\delta} = \frac{K(1+T_3 s)}{(1+T_1 s)(1+T_2 s)} \tag{2.6}$$

$$T := T_1 + T_2 - T_3 \tag{2.7}$$

$$\frac{r}{\delta} = \frac{K}{(1+Ts)} \tag{2.8}$$

Setting $\dot{\psi} = r$ in equation 2.8 yields the transfer function that is most used in commercial autopilot systems. For more information on the Nomoto model, see Fossen (2011).

2.3 Observer

As seen from the previous section, the models are dependent on state measurements. However, these measurements are not necessarily available. A reconstruction of non-measured data is, therefore, necessary and can be done by implementing observers. Observers are used for filtering of noisy signals to reduce wear and tear on machinery and to estimate states that for some reason are not measured. All equipment will eventually fail according to some failure rate, Fossen (2011). Hence, the observers provide reduced risk as the model-based filters can replace the lost measurements by a model prediction for some time. In order to apply a state estimator, the system needs to be observable. A system is observable if it is possible to determine the behaviour of the entire system by considering the output of the system, Fossen (2011). In this section, the observer design for the Nonlinear Passive Observer will be presented.

2.3.1 Nonlinear Passive Observer

The research on Nonlinear Passive Observers (NPO) is motivated by the large number of tuning parameters needed for the alternative and more known Kalman Filter that needs to be determined through experimental testing, Fossen (2011). The Kalman Filter can potentially consist of 15 states and 120 covariance equations, while the NPO only require 15 tuning parameters that additionally are easier to

relate to physical quantities. The NPO can be modelled to copy the dynamics of the control plant model defined in equation 2.2 as Fossen (2011):

$$\dot{\hat{\xi}} = \mathbf{A_w}\hat{\xi} + K_1(\omega_0)\tilde{y} \tag{2.9a}$$
$$\dot{\hat{\eta}} = \mathbf{R}(y_3)\hat{\nu} + K_2\tilde{y} \tag{2.9b}$$
$$\dot{\hat{\mathbf{b}}} = -\mathbf{T}^{-1}\hat{\mathbf{b}} + \mathbf{K}_3\tilde{\mathbf{y}} \tag{2.9c}$$
$$\mathbf{M}\dot{\hat{\nu}} = -\mathbf{D}\hat{\nu} + \mathbf{R}^\top(y_3)\hat{\mathbf{b}} + \tau + \tau_{wind} + \mathbf{R}^\top(y_3)\mathbf{K}_4\tilde{y} \tag{2.9d}$$
$$\hat{\mathbf{y}} = \hat{\eta} + \mathbf{C_w}\hat{\xi} \tag{2.9e}$$

where $\tilde{y} = y - \hat{y} \in \mathbb{R}^3$ is the estimation error, $\mathbf{K}_1 \in \mathbb{R}^6$, and $\mathbf{K}_2, \mathbf{K}_3, \mathbf{K}_4 \in \mathbb{R}^3$ are the observer gains. There are two types of bias models to choose from. One is a simplified version based on a *Wiener Process*. In this simplified version, $\mathbf{T} = 0$ and the bias only handles the white noise term, $\dot{\hat{\mathbf{b}}} = \mathbf{K}_3\tilde{\mathbf{y}}$. Alternatively, the bias can be modeled as the more advanced *Markov Process* in Equation. Here, $\mathbf{T} > 0$ and the filter has a low pass filter that gives exponential stability, Fossen (2011).

Fossen (2011) propose the following formulas for estimating the observer gains K_1 and K_2,

$$\mathbf{K}_{1i}(\omega_i) = -2(\zeta_i - \lambda_i)\frac{\omega_{ci}}{\omega_i} \tag{2.10}$$
$$\mathbf{K}_{1(i+3)}(\omega_i) = -2\omega_i(\zeta_i - \lambda_i) \tag{2.11}$$
$$\mathbf{K}_{2i} = \omega_{ci} \tag{2.12}$$

where $\omega_{ci} > \omega_i$ are the wave spectrum peak frequency and filter cutoff frequency respectively, and $\zeta_{ni} > \lambda_i$. Typically $\zeta_{ni} = 1.0$ and $\lambda = 0.1$.

2.4 Controller

Controllers are used to calculating the desired thrust τ from the actuators in order to move the system to the desired reference state. In this section PID controllers developed for DP in addition to a heading and speed controller suited for transit operations will be presented.

2.4.1 DP Controller

A PID controller is a widely used controller in the industry. It is easy to implement as prior knowledge of the model is unnecessary, and it is developed to control the horizontal motions in surge, sway and yaw. The control law for a PID controller is defined as

$$\boldsymbol{\tau} = K_p\boldsymbol{R}^T(\psi)\tilde{\boldsymbol{\eta}} + K_d\tilde{\boldsymbol{\nu}} + K_i\int_0^t \boldsymbol{R}^T(\psi)\tilde{\boldsymbol{\eta}}(\tau)d\boldsymbol{\tau} \tag{2.13}$$

where $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta_d} - \boldsymbol{\eta}$ and $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu_d} - \boldsymbol{\nu}$ is the error. The controller gains $\mathbf{K_p}$, $\mathbf{K_i}$ and $\mathbf{K_d}$ are tuned to achieve the desired performance.

Reference feedforward can be added to the PID controller as

$$\boldsymbol{\tau} = K_p \boldsymbol{R}^T(\psi)\tilde{\boldsymbol{\eta}} + K_d \tilde{\boldsymbol{\nu}} + K_i \int_0^t \boldsymbol{R}^T(\psi)\tilde{\boldsymbol{\eta}}(\tau) d\boldsymbol{\tau} + \underbrace{Ma_d + D\nu_d}_{\text{Reference feedforward}} \tag{2.14}$$

where a_d is the acceleration reference in the BODY-frame, M is the mass matrix and D is the damping matrix of the vessel.

Even though the PID controller is easy to implement, it is only tuned for one environmental condition and one specific operation. The same PID controller will therefore not be applicable for both manoeuvring and dynamic positioning operations. This motivates the implementation of a hybrid system which will be explained in Section 2.6.

2.4.2 Heading Controller

For a transit manoeuvring problem, the PID controller described in the previous section can be simplified to

$$\tau_\psi = -K_p\tilde{\psi} - K_d\dot{\tilde{\psi}} - K_i \int_0^t \tilde{\psi} d\tau \tag{2.15}$$

where $\tilde{\psi} = \psi - \psi_d$, and ψ_d is the desired heading angle. For manoeuvring is only the heading angle of importance and hence this is the only state used for the control law. However, the vessel's lateral distance to the path will depend on the magnitude of the environmental forces for this control law, Fossen (2011). The environmental forces will induce a sideslip angle β modelled as in equation 2.16. A way to counteract this effect is to take the sideslip angle into account when defining the desired heading ψ_d with a guidance law. This will be explained in Section 2.5.3.

$$\beta = \frac{v}{U} \qquad U = \sqrt{u^2 + v^2} \tag{2.16}$$

2.4.3 Speed Controller

In order to control the speed of the vessel, a PI speed controller can be utilized, calculating the force in the x-direction of the vessel body. The reason for using a PI controller when controlling the surge speed instead of a PID controller is due to the often poor velocity measurements. Derivating these measurements would result in undesirable noisy signals. Hence, the PI speed controller is defined as

$$\tau_x = -K_p\tilde{u} - K_i\tilde{u} \tag{2.17}$$

where $\tilde{u} = u - u_d$, and u_d is the desired surge speed, Fossen (2011).

2.5 Guidance

In order to ensure desired performance for the motion controllers, guidance systems are often used. These provide smooth reference signals to the controllers, avoiding unwanted wear and tear on the actuators in addition to account for physical limitations of the craft. In addition, these smooth signals will reduce the effect of integral wind-up for the controller. Integral wind-up is an unwanted accumulation of integral effect which can lead to massive overshoot.

In this section, a physically motivated model to generate the desired reference for position, velocity and surge speed will be presented in addition to a Line-of-sight guidance law used for path following.

2.5.1 Position and Attitude Reference Model

When manoeuvring a vessel through a set of waypoints, the desired state of the vessel will experience large setpoint changes. The reference model provides a smooth trajectory between these setpoints. A common way of generating these smooth trajectories η_d is to use a physically motivated model.

For a marine vessel the reference model can be motivated by the dynamics of a mass-damper-spring system to generate the desired state trajectories, as it applies well for marine crafts, Fossen (2011). A third order reference model for position and attitude can be defined as

$$\eta_d + (2\mathbf{\Delta} + \mathbf{I})\mathbf{\Omega}\ddot{\eta_d} + (2\mathbf{\Delta} + \mathbf{I})\mathbf{\Omega}^2\dot{\eta}_d + \mathbf{\Omega}^3\eta_d = \mathbf{\Omega}^3\mathbf{r}^n \tag{2.18}$$

where η_d is the desired position reference, $\mathbf{r}^n$ is the setpoint, $\mathbf{\Delta}$ is a diagonal matrix of relative damping ratios as defined in equation 2.19 and $\mathbf{\Omega}$ is a diagonal design matrix of natural frequencies as defined in equation 2.20. If the system is set to be critically damped, the reference model will not overshoot the desired setpoint value. For a critically damped system is $\zeta = 1$ leaving $\mathbf{\Delta} = \mathbf{I}$.

$$\mathbf{\Delta} = diag\{\zeta_1, \zeta_2, \ldots, \zeta_n\} \tag{2.19}$$

$$\mathbf{\Omega} = diag\{\omega_{n_1}, \omega_{n_2}, \ldots, \omega_{n_n} \tag{2.20}$$

In order to ensure that the dynamical limitations of the craft are included in the reference model, one can add saturation elements for the velocity and acceleration. The third order reference model is displayed in Figure 2.1.

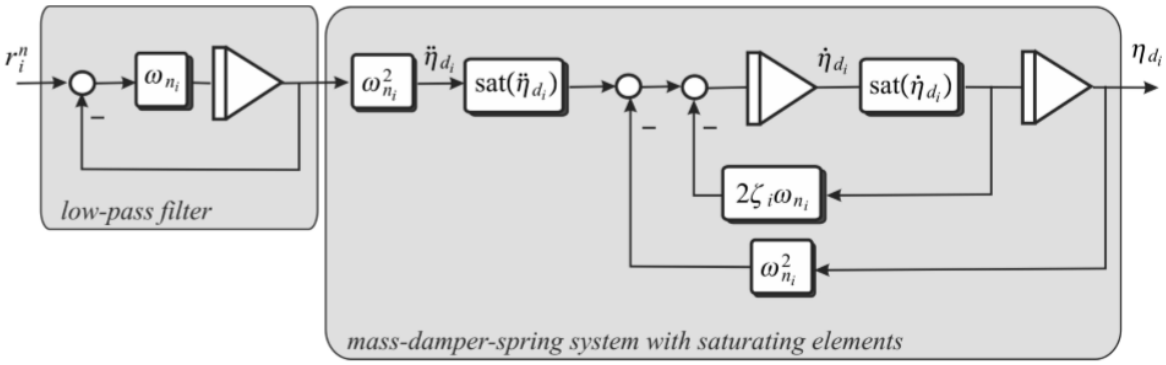


Figure 2.1: Third order reference model including saturating elements. Source: Fossen (2011)

2.5.2 Velocity Reference Model

A velocity reference model should at least be of order two in order to obtain smooth reference signals for the desired velocity ν_d and acceleration $\dot{\nu}_d$. For a speed reference model to be used in manoeuvring operations where the surge speed is assumed much larger than the sway speed $u >> v$ one can define a second order surge speed filter for the surge speed as

$$\ddot{u_d} + 2\zeta\omega\dot{u}_d + 2\zeta\omega^2 u_d = \omega^2 u_r \tag{2.21}$$

where $\omega > 0$ is the natural frequency $\zeta > 0$ is the reference model damping ratio and u_r is the setpoint specifying the desired surge speed.

2.5.3 Line-of-sight

For manoeuvring applications, path following often is the main objective of the operation. A common way of defining the desired path for a vessel is to define a set of desired waypoints. A *Line-of-sight* (LOS) guidance law generates a straight line path by connecting this set of waypoints in the $\{\mathrm{n}\}$-frame, and defines the desired heading for the vessel when moving along these straight lines. The set of waypoints are defined as in equation 2.22 and 2.23 where k is the enumerator for each waypoint, for a path consisting of n number of waypoints.

$$\mathbf{p_k^n} = [\mathbf{x_k}, \mathbf{y_k}]^\mathbf{T} \in \mathbb{R}^\mathbf{2} \tag{2.22}$$

$$\mathbf{p_{k+1}^n} = [\mathbf{x_{k+1}}, \mathbf{y_{k+1}}]^\mathbf{T} \in \mathbb{R}^\mathbf{2} \tag{2.23}$$

Figure 2.2 displays a straight line defined by the two waypoints $\mathbf{p_k}$ and $\mathbf{p_{k+1}}$. If one considers a path fixed reference frame with origin in $\mathbf{p_k}$ with the x-axis rotated by the positive angle

$$\alpha_k = atan2(y_{k+1} - y_k, x_{k+1} - x_k) \in [-\pi/2, \pi/2] \tag{2.24}$$

the vessels coordinates in the path fixed reference frame can be computed as

$$\varepsilon(t) = [s(t)e(t)]^T = \mathbf{R}_p(\alpha_k)^T(\mathbf{p^n}(t) - \mathbf{p}_k^n \tag{2.25}$$

where

$$\mathbf{R}_p(\alpha_k) = \begin{bmatrix} cos(\alpha_k) & -sin(\alpha_k) \\ sin(\alpha_k) & cos(\alpha_k) \end{bmatrix} \in SO(2) \tag{2.26}$$

$SO(m)$ it the special orthogonal group of order m and states that the matrix is orthogonal and has determinant = 1. $s(t)$ represents the along track distance tangential to the path while $e(t)$ represents the cross-track error normal to the path. For path following purposes, only this cross track error is of importance. The control objective

$$e(t) \to 0 \tag{2.27}$$

i.e. minimizing the cross-track-error, will make the craft converge to the desired path asymptotically.

The LOS vector define the projection from the marine craft and the intersection point on the straight line segment (x_{los}, y_{los}), see Figure 2.2. The aggressiveness of the steering law is decided by the *Lookahead-distance* Δ as how far from the along-track distance $s(t)$ the LOS vector intersect.

Two different guidance principles can be used; Enclosure-based steering and Lookahead-based steering. The latter is less computationally expensive and also valid for all cross-track errors. This steering law is therefore implemented for this project.

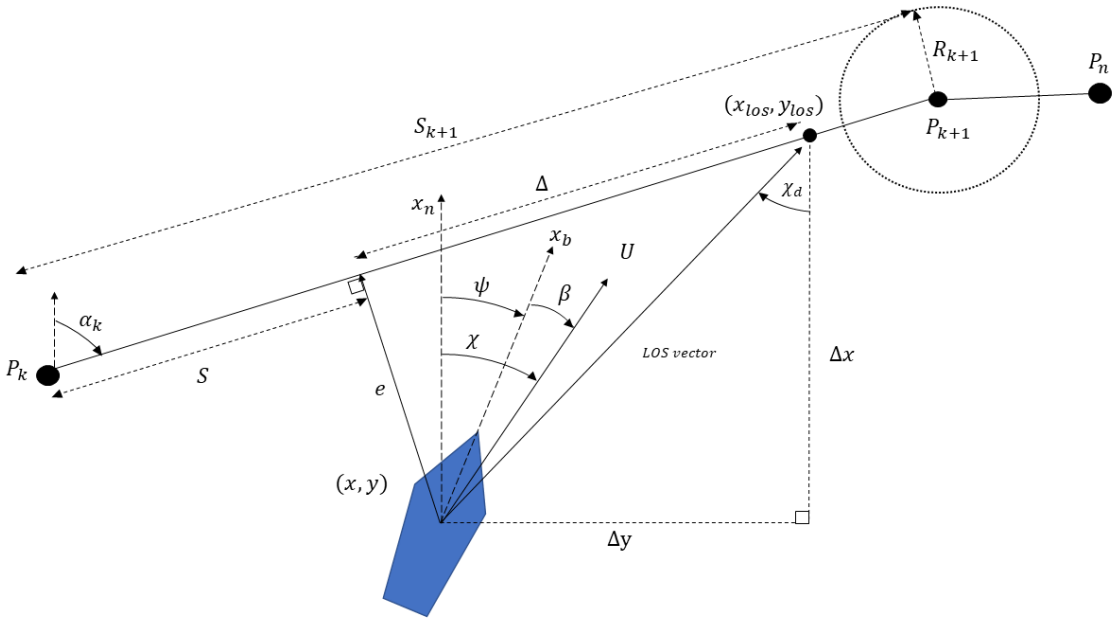


Figure 2.2: LOS guidance illustration. Figure adapted from Fossen (2011)

Lookahead-Based Steering Law

A lookahead-based steering calculates the desired course angle for the vessel, $\chi_d(e)$. This angle is calculated by considering a straight line trough the waypoints defined in equation 2.22 and 2.23, see Figure 2.2.

The desired course angle consists of two parts defined in Fossen (2011) as

$$\chi_d(e) = \chi_p + \chi_r(e) \tag{2.28}$$

where χ_p is the path tangential angle defined as

$$\chi_p = \alpha_k \tag{2.29}$$

and $\chi_r(e)$ is a velocity-path relative angle. α_k can be calculated as stated in equation 2.24 and $\chi_r(e)$ is defined as

$$\chi_r(e) := arctan\left(\frac{-e}{\Delta}\right) \tag{2.30}$$

where e is the cross-track error and $\Delta > 0$ is the lookahead distance. From equation 2.30 it is evident that a small Δ yields a large $X_r(e)$ causing a more aggresive convergence to the path.

In the presence of environmental forces as ocean currents, the controller must account for the *sideslip*-angle β. Hence, course angle χ_d needs to be transformed to a heading angle ψ_d as

$$\psi_d = \chi_d - \beta \tag{2.31}$$

The sideslip angle β is computed by $\beta = arcsin\frac{v}{U}$ where $U = \sqrt{u^2 + v^2}$. However, this requires knowledge of the velocities. If these are not measured, guidance laws of PI can be utilized to counteract the sideslip-angle by treating β as a slowly varying unknown disturbance satisfying $\dot{\beta} \approx 0$. The velocity-path relative angle can then be written as

$$\chi_r = arctan\left(-K_p e - K_i \int_0^t e(\tau)d\tau\right) \tag{2.32}$$

Circle of Acceptance

A switching mechanism when moving along the straight line segments connected by the waypoints is necessary. One way to implement such a mechanism is by using a *circle of acceptance* around the waypoints. When the vessels position at time t satisfies

$$[x_{k+1} - x(t)^2] + [y_{k+1} - y(t)]^2 \leqslant R^2 \tag{2.33}$$

is the next waypoint (x_{k+1}, y_{k+1}) selected. As a guideline is often R chosen equal to two ship lengths, $R = 2L_{pp}$, Fossen (2011).

2.6 Hybrid System Modeling and Control

A hybrid system is a system combining both continuous and discrete dynamics. This is especially profound in systems where logic decision making and embedded control actions are combined with physical processes. An introduction in the underlying theory for hybrid systems and an example will follow before theory on supervisory switch control is presented.

Controlling a continuous-time system can pose challenges. Most control systems are developed for one operational condition, but what happens if these conditions change? This issue motivates the use of hybrid control as it opens up for the use of different controllers through different parts of the operation and allows changes in parameters for both the plant model and environment.

A hybrid system can be divided into flows and jumps as defined in equation 2.34 where C is the flow set, F is the flow map, D is the jump set and G is the jump map, Goebel (2012).

$$\begin{cases} x \in C & \dot{x} \in F(x) \\ x \in D & x^+ \in G(x) \end{cases} \tag{2.34}$$

Based on this representation, the state x will change according to the differential relation $\dot{x} = F(x)$ when in the set C, and the state will jump according to the difference relation $x^+ \in G(x)$ when in the set D. x^+ represents the value of a state x after a jump. Defining the hybrid system with such flow

and jump sets avoid rapid successive jumping called chattering. Another advantage of this definition is its applicability to a number of different types of hybrid models, i.e systems with computer sampling, impact affected mechanical systems or systems with logic variables. The simple bouncing ball example will follow below to further shed light on the modelling of hybrid systems. For more information on the modelling of hybrid systems, see Goebel (2012).

2.6.1 Example: Bouncing ball

The bouncing ball is described as an example of a hybrid system in Goebel (2012). When a bouncing ball is falling towards a table surface, it flows inbetween impacts. The state of the bouncing ball can be described as $x = [x_1, x_2]^T \in \mathbb{R}^2$, where x_1 represents the height above the surface and x_2 represents the vertical velocity. The flow set C can therefore be defined as the state when the ball is above the surface, or when it is at the surface and the velocity points up, $C = x = \mathbb{R}^2 : x_1 > 0 \quad \text{or} \quad x_1 = 0, x_2 \leq 0$. The flow map can be defined as the trajectory of the bouncing ball, where the flow map is noncontinuous at the surface, $f(x) = [x_2, -\gamma]^T$ when $x_1 > 0$ or $x_1 = 0, x_2 > 0$ where γ is the acceleration due to gravity. The jump set D is defined as the instant where the ball hits the surface and the velocity is negative, $D = X \in \mathbb{R}^2 : x_1 = 0, \quad x_2 < 0$. Hence, the jump map can be defined as $g(x) = [0 - \lambda x_2]^T$, where $\in (0, 1)$.

In this example, every jump is followed by a periodic flow and there are no consecutive jumps. However, consecutive jumps might happen for other systems and can impose issues with chattering for the system. For more examples on hybrid dynamical systems, see Goebel (2012).

2.6.2 Supervisory Switch Control

Supervisory control is a way of implementing hybrid control. The supervisory switch control framework is described in Hespanha (2001). This framework is considered for the control of complex systems where traditional control methods do not provide satisfactory performance and are developed by defining a bank of candidate controllers one switch between based on measurements. Hence, this system can be considered a hybrid system as the continuous and discrete dynamics are combined.

The hybrid controller DP system combines two main blocks; a supervisor and a bank of controllers and observers. The estimator and controller are designed in pairs where each estimator is designed together with at least one controller. The set of estimators, multi-estimator, is defined as

$$\mathscr{M} := \bigcup_{p \in \mathscr{P}} \mathscr{M}_p \tag{2.35}$$

while the set of controllers is denoted as

$$\mathscr{C} := \bigcup_{q \in \mathscr{L}} \mathscr{C}_q \tag{2.36}$$

where $\mathscr{P}$ and $\mathscr{L}$ are a small set of admissible observers and controllers respectively, while q and p are the qth and pth model respectively.

The main idea is to switch between a bank of controllers with a switching signal σ provided from a supervisor, see Figure 3.1. σ is defined in the supervisor based on the available measurements, choosing the controller that seems most promising based on some switching logic.

Hespanha (2001) define two important properties for the switched system; *matching* and *detectability*. In addition, the switching logic will guarantee the *small error* and *non-destabilization* properties. The

matching property states that each estimator should be designed such that each $\hat{y}$ provides a good approximation to the output y. This results in a small error $\tilde{y} = y - \hat{y}$ and is called the small-error property. The detectability property is preserved by the non-destabilization property, which is satisfied if the switching is slow. For a linear model and controllers, this can be achieved by implementing dwell-time switching logic. In the case of a nonlinear model, scale-independent hysteresis switching logic can be utilized to satisfy the property. The scale-independent hysteresis switching logic was chosen for our system. This will be further discussed in Section 3.1.

The matching and detectability properties are important, as the small error property guarantees that the selected estimator is the best. When the system is at the border of the control/estimator design, instability and fast switching (chattering) may occur. This is avoided by the use of the non-destabilization property. More details on this can be found in Hespanha (2001).

The scale-independent hysteresis switching logic for the monitoring signal μ_p is defined in Nguyen et al. (2007) as

$$\dot{\mu}_p = -\lambda\mu_p + \gamma(\|\mathbf{e}_p\|), p \in \mathscr{P} \tag{2.37}$$

where λ is a constant non-negative forgetting factor used to tune the switching system, γ is the class $\mathscr{K}$ function, $\mu_p(0) > 0$, and $\|\dots\|$ denotes any norm. This switching logic slows down switching based on the growth of the error $\boldsymbol{e}_p$, and the switching procedure is displayed in Figure 2.3, where h is a positive hysteresis constant and $argmin\mu_p$ returns the index of the minimum value of the μ_p vector.

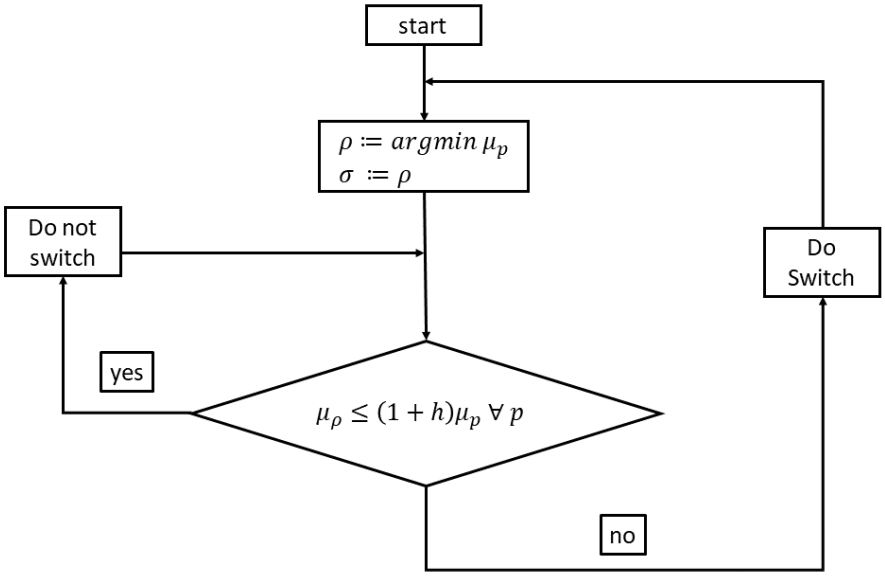


Figure 2.3: Scale-independent hysteresis switching logic. Figure adapted from Nguyen et al. (2007)

2.7 Machine Learning

The implementation of a controller combined with a guidance law and a set of static waypoints cover the automatic level in autonomy. However, the deliberate or intelligent level is still missing. What if the conditions that lie behind the predefined waypoints change? In order to improve the level of autonomy, the domain of machine learning could be used to generate an optimal set of waypoints by taking into account the environmental conditions.

Machine learning is a subset of Artificial Intelligence (AI), containing an interdisciplinary field with contributions from computer science, mathematics, statistics and others. A machine learning algorithm is recognized by using sample data (or training data) to build a mathematical model for making decisions or predictions without being explicitly programmed, Bishop (2006). However, domains within machine learning developed to avoid the need for sample data or a model of the environment also exist. Reinforcement learning is such a domain containing the Q-learning algorithm. Hence, a Q-learning algorithm will be presented to showcase the potential of machine learning for marine applications.

2.7.1 Q-learning Algorithm

The Q-learning algorithm is a part of a domain within machine learning called reinforcement learning. Reinforcement learning is recognized by using observed rewards to learn an optimal policy for the environment, Russell (2016). As mentioned in Section 1.2.4 the article Yoo and Kim (2016) used this algorithm to generate an optimal path while taking into account the environmental conditions from current. The Q-learning algorithm devloped in this thesis therefore is inspired by the work conducted in this article. By taking the factor of current forces into account, safety and fuel consumption could be improved.

The task of reinforcement learning is to use observed rewards to learn an optimal policy for the environment. The problem is assumed fully observable and stochastic with additive rewards and therefore named a Markov decision process (MDP) that consist of an agent, states, actions and rewards. The agent can change the state by completing an action in the environment, and will then be rewarded based on that specific condition, see Figure 2.4. The goal for the agent is to maximize its total reward by learning the optimal policy π^* for each state.

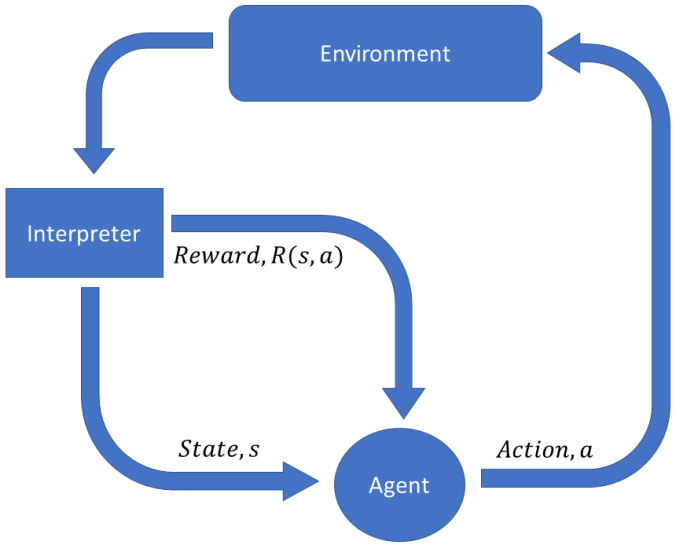


Figure 2.4: Reinforcement learning flow chart

Reinforcement learning is powerful in the way that it does not need a model or data to learn the desired path. The approach is based on simulated trial and error, where the algorithm test different actions in a simulated environment in order to reach the goal state. The agent then finds the optimized path based on the rewards gathered during the simulation. This analysis can then be made prior to departure of the vessel, or be repeated at specific time instants during the operation if alternations to the path need to be made based on changing conditions.

The pseudo code for the algorithm is provided in listing 2.1. Q(s,a) denote the quality of doing action a in state s and is initialized to an empty matrix. The R matrix represents the reward for taking an action from a given state, Russell (2016). γ_Q represents the discount factor and determine the importance of future rewards. A factor of 0 will make the agent short-sighted, while a factor of 1 will make it strive for a long-term high reward. α_Q represents the learning rate and determines how newly gained information overrides old. A way to implement α_Q is by defining functions to produce an exploratory or exploitable approach when the agent chooses it's action. Yang et al. (2014) defines an chaos algorithm used for the initial period, before it is replaced by a $epsilon$-greedy selection strategy to exploit the information from the exploration phase. For more information on the Q-learning algorithm and its implementation, see Russell (2016).

Listing 2.1: Pseudo code Q-learning algorithm

```
% s: state, a: action
Q = initialize;
R = initialize;
for a number of itterations
    for each state
        for each available action
            Q(state,action) =  Q(s,a)(1-α_Q)
            + α_Q (R(s,a) + γ_Q max(Q(nextS,a));
        end
    end
end
π* = max(Q);
```

Chapter 3

Method

This chapter presents the methods used to implement an autonomous system. First, the methods behind the hybrid control system are described in addition to the implementation of the Q-learning path planner. Then the simulation and lab setup is presented for both the performance tests of the hybrid controller and the verification of the Q-learning path planner.

3.1 Hybrid Control

The scale-independent hysteresis switching logic described in Section 2.6 was chosen, as this model is well suited for switching between linear/nonlinear models and controllers. Hence, this model will ease future implementation of more advanced nonlinear models and controllers to the system. The supervisor determines the next desired waypoint based on the hysteresis switching logic and the distance from the current position to all the waypoints. The waypoint then decides what mode should be active.

The supervisory switch was implemented for the hybrid control system using the theory described in Section 2.6.2, using equation 3.1 and 3.2. The error vector $\boldsymbol{e}_p$ was set to the distance from the vessel to each of the waypoints. The crossing was divided into a DP phase and a manoeuvring phase, that again was divided into modes with different active controllers.

The index of the waypoint with the smallest error is represented by ρ. If the smallest error goes below value h_{man} for the manoeuvring modes or below h_{DP} for the DP modes, the desired point is marked as reached. Subsequently, the minimum error in the $\mathbf{e_p}$ vector is multiplied with a large gain to ensure that this waypoint won't be rechosen by the supervisor. h_{DP} was tuned to a lower value than h_{man} as the DP phase of the crossing produce a trajectory with smaller errors to the desired path compared to the manoeuvring phase. λ, h_{man}, h_{DP} and h was tuned together to make the switch happen at an adequate distance from the desired waypoint. Consequently, the active mode is determined based on the next desired waypoint.

$$\dot{\mu}_p = -\lambda\mu_p + \gamma(\mathbf{e}_p), p \in \mathscr{P} \tag{3.1}$$

$$\mu_\rho <= (1+h)\mu_p \tag{3.2}$$

Figure 3.1 displays the block diagram of the hybrid control system. The NPO observer is used for all the modes, while the resulting force τ from the controllers are chosen based on the switching signal σ, defining the active mode.

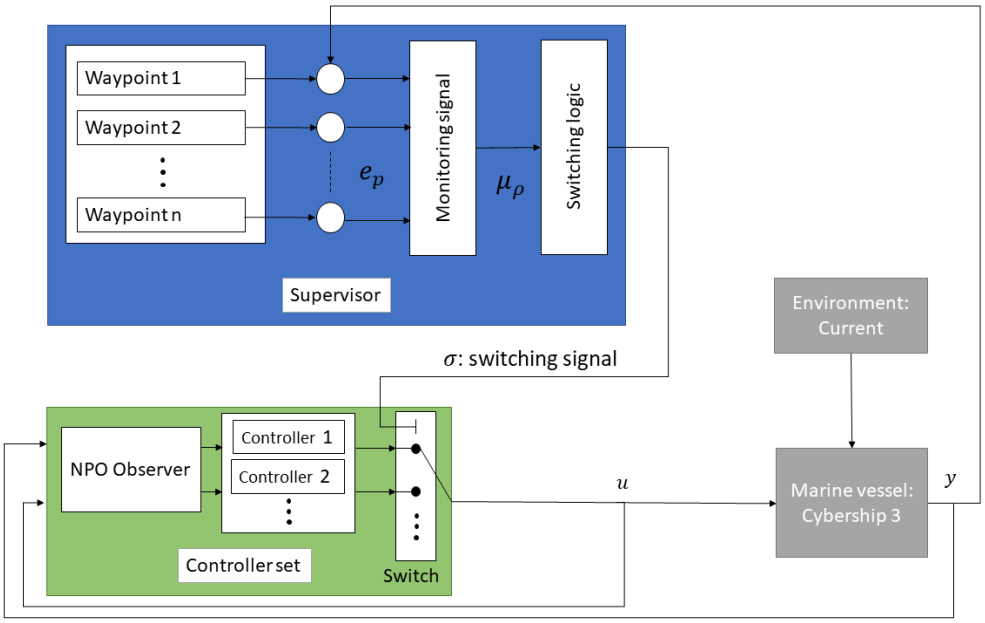


Figure 3.1: Block diagram of hybrid control system

3.2 Path Planner

For the Q-learning path planner, a planar motion model was chosen, as the motion in the horizontal plane is the major concern for surface ships. In addition, the number of states was kept to a minimum to improve the computational efficiency of the path planner. The chosen model is displayed in equation 3.3. The equation represents the equations of motion for a vessel affected by ocean currents in the 2D plane. x and y are the position coordinates, V is the forward velocity and a is the acceleration. The yaw dynamics are disregarded as the feasibility of the Q-learning problem much depends on the simplicity of the model in regards to computational time. This choice will be further debated in Section 6. The path planner was developed to calculate the most optimal waypoints in terms of minimum-time for the vessel during the manoeuvring phase of the operation.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V cos\psi \\ V sin\psi \\ a \end{bmatrix} \tag{3.3}$$

A mesh grid map was defined as displayed in Figure 3.2 in order to discretize the position state variables. The figure demonstrates an example of how a map can be drawn, where the size of the mesh can be extended in both positive x- and y-direction. The size of each tile can also be defined arbitrary by l. The initial state is set to one by default, presented by the blue colour in Figure 3.2. The orange coloured state represents the goal state, as the one furthest away from the initial state in both x and y-direction. However, an extension to define this goal state arbitrary on the map is possible with minor edits to the code, but with a payoff on the computational-efficiency. This will be further discussed in Section 6.

The available actions for the vessel in each state were limited to $[-90, -45, 0, 45, 90]$ degrees. Hence, the vessel was not allowed to move backwards, except when at the edge of the map to ensure the feasibility of the Q-learning algorithm. A positive constant forward force was also applied. This was done to mimic a more realistic dynamic for the vessel with the yaw rate of marine vehicles often being constrained and as transit operations essentially move in the positive x-direction for the BODY-frame. The actions were chosen through the **randi()** Matlab function, to ensure an exploratory approach.

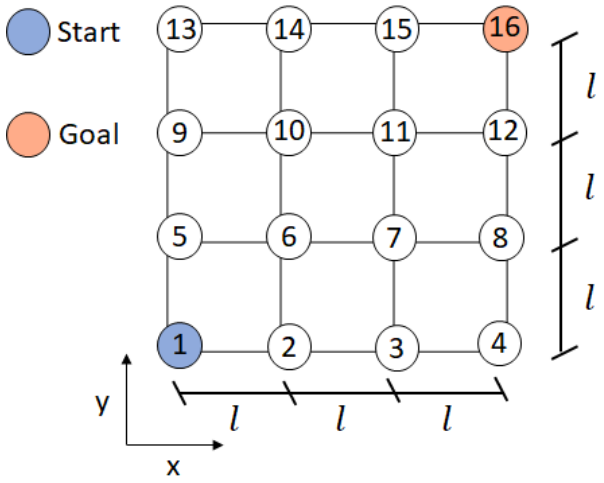


Figure 3.2: Meshgrid example of map for Q-learning algorithm

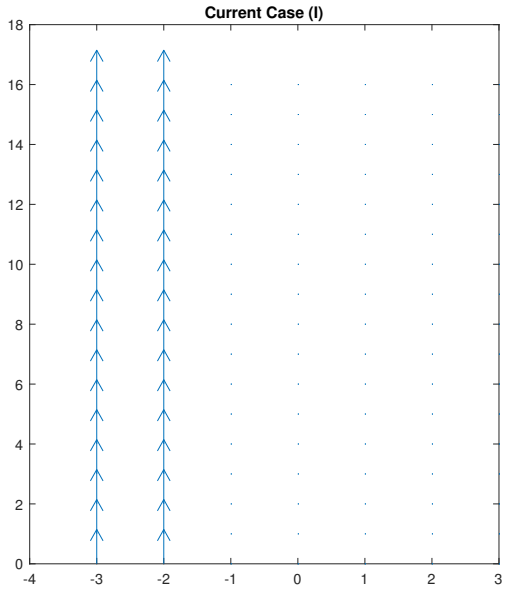
The optimization problem for the Q-learning path planner was time. Hence, the time spent going from the initial state to the goal state was to be minimized. The amount of time spent in each tile on the mesh grid was therefore defined as a penalty or negative reward, inspired by Yoo and Kim (2016). The summarized penalty for each completed episode then represents the total amount of time spent on travelling from the initial state to the goal state. This total penalty was then multiplied by -1 to produce a minimum-time path found by maximizing the total reward.

The current is modelled as speed in the x and y-direction of the mesh grid map. Current maps was developed from the Matlab function **quiver()**, see Figure 3.3. The arrows display the direction of the current while the length of each arrow represents the velocity. Case (I) was generated to represent the lab setup when the Q-learning path planner was verified in the MCLab. Here the length of l for each tile was 2 meters. A current of 0.1 m/s in the positive y-axis was initiated in the area of $x \in [-2, -3]$. Case (II), however, is produced to represent an environment more true to a real-world domain. This current case was inspired by tidal currents emerging between tight passages, e.g two islands or a fjord. The current velocity for case (II) varies from $0.01\ m/s$ and $0.15\ m/s$, and was enabled when testing the hybrid control system in the simulation environment. The size of l for each tile for this case was set to 6 meters. Table 3.1 summarise which of the two current cases that was used for verification of the path found by the Q-learning algorithm in simulation and lab.

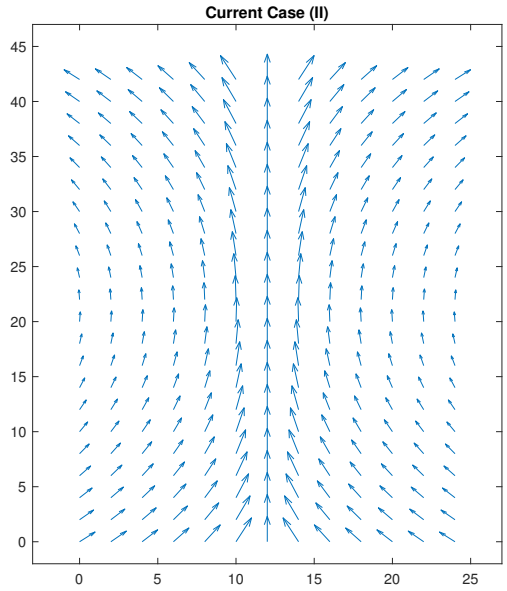
The resulting path of the path planner is represented through a list of waypoints that represents the optimal path. This list comprise the desired waypoints that are fed into the control system.

Table 3.1: Overview of current cases used when verifying path planner in simulation and lab

	Case (I)	Case (II)
Simulation	×	×
Lab	×	



Case (I): Current map for lab



Case (II): Current map for ferry crossing

Figure 3.3: Current maps

3.3 Simulation Setup

Both the hybrid control system and the path planner was tested in a simulation environment. The Marine Cybernetics Simulator (MCSim) was used, developed in Matlab/Simulink with the latest changes provided by Astrid H. Brotkorv. The model used for the simulations was the model ship Cybership 3, displayed in Figure 3.4 and 3.5. MCSim contained a 6 DOF high fidelity model, captured in equation 2.1. The simulator was divided into a vessel dynamic module containing a LF and WF model, an environmental module containing wave-, surface current-, ice- and wind models, a thruster and shaft module with thrust allocation and thruster dynamics together with a local thruster control, and finally a vessel control module. The control module had a nonlinear passive observer (NPO) and a DP controller that was used for the simulation. In addition, was a heading controller, a speed controller and reference models developed and added to the control module. The MCSim was set to manoeuvring mode, meaning that the nonlinear damping term in $\boldsymbol{D = D_L + D_{NL}}$ was used. For more information on the MCSim, see Hassani et al. (2012).

3.3.1 Hybrid Control

As mentioned in Section 3.1, the operation with the hybrid controller was divided into modes. For the simulation setup, the following modes were defined:

1. Low speed maneuvering (DP)
2. Acceleration
3. Deceleration
4. Transit

Mode 1: Low-speed manoeuvring is used initially in the crossing operation when leaving the dock, and finally when docking at the destination. The PID controller with reference feedforward for DP described in Section 2.4.1 was used when this mode was active, and the mode makes up the DP phase of the crossing. In addition, the third order reference model described in Section 2.5.1 was used to provide a smooth reference signal to the controller. When switching back to mode 1 at the end of the operation, the reference model was reset to the state of the vessel at that time instant. This was done to avoid unnecessary jumps in the reference signal and the unwanted effects from integral windup.

Mode 2: Acceleration is used for the acceleration phase of the crossing, where the vessels surge speed is increased rapidly into the desired surge operation speed. For this mode, the PID heading controller described in Section 2.4.2 was used in combination with the PI speed controller described in Section 2.4.3. The LOS guidance law, with a PI law to counteract the sideslip-angle described in Section 2.5.3, was used to provide a smooth reference signal for the heading. The second order reference model described in Section 2.5.1 was used to provide a smooth reference signal for the speed controller. The reference model for the speed controller for this mode was tuned aggressively in order to accelerate the vessel towards the desired transit speed as quickly as possible.

Mode 3: Deceleration does the opposite of mode 2 as it decreases the surge speed of the vessel in order to prepare it for the docking phase of the crossing. Similarly to mode 2 was the heading controller from Section 2.4.2 and the speed controller from Section 2.4.3 active for this mode, in addition to the LOS guidance law and speed reference model. Mode 3 takes advantage of the hydrodynamic damping when decreasing the speed in order to minimize the use of thrust force. Consequently, the reference model for the speed controller was tuned to slowly reduce the speed of the vessel.

Finally, mode 4: Transit represents the phase of the crossing when the vessel is holding a constant surge speed. Also here the speed and heading controller were used to achieve the desired heading and surge speed for the mode. Only the LOS guidance reference model for the heading controller is active during this mode, as there are no changes in the desired speed when in transit. Mode 2, 3 and 4 compound the manoeuvring phase of the crossing operation.

When switching between the modes, the integral terms in the controllers was reset to zero to avoid the effects of the integral windup.

A summary of the used controllers and refernce models for each mode can be found in table 3.2 and 3.3.

Table 3.2: Controllers used for the modes in simulation

Controller	**Mode 1**	**Mode 2**	**Mode 3**	**Mode 4**
DP Controller	×			
Heading Controller		×	×	×
Speed Controller		×	×	×

Table 3.3: Reference models used for the modes in simulation

Reference Model	**Mode 1**	**Mode 2**	**Mode 3**	**Mode 4**
Position and Attitude	×			
LOS guidance law		×	×	×
Spped - acceleration		×		
Speed - deceleration			×	

3.3.2 Path Planner

The Q-learning path planner was tested and verified in the simulation environment MCSim. This was done by comparing the time spent on travelling the path calculated by the Q-learning path planner, to the shortest distance path, with the given environmental conditions. The current maps in Figure 3.3 was used as environmental input. Simulations of both the experiment to be conducted in the lab, and a crossing operation more true to a real-world environment, were conducted using MCSim. The simulation of the lab experiment used the current map of case (I) displayed to the left Figure 3.3, and the simulation of the real-world operation used the current map of case (II) displayed to the right in Figure 3.3.

For this test, the heading controller was used in combination with the LOS guidance controller. The waypoints found from the Q-learning path planner was used as input to the control system. A circle of acceptance as described in Section 2.5.3 was used to implement a switching mechanism when moving along the straight line segments made up by the waypoints. A constant force of 1N was used for τ_x to produce a constant force in the positive x_{body} direction.

3.4 LAB Setup

The model test tank Marine Cybernetics Laboratory (MCLab) at the Department of Marine Technology, NTNU, was used to verify the path planner and hybrid control system. The MCLab is equipped with a motion capture unit (MCU), consisting of motion capturing cameras attached to the towing carriage and markers mounted to the vessel. The markers reflect the infrared light emitted from the cameras and provide earth-fixes position and heading of the vessel. However, the cameras can only capture the position of an area of approximately $7 \times 6m^2$, while the basin actually measures $40 \times 6.5m^2$.

The vessel used for conducting the tests was the model ship Cybership 3, displayed in Figure 3.4 and 3.5. This is a 1:30 scale model of a supply vessel, weighing 75 kg with an overall length of 2.275 m and breadth of 0.4 m. The ship has two azimuth thrusters installed at the rare end of the vessel and one in the bow. The hardware is controlled by an onshore computer with NI Veristand software that communicates with an onboard computer through WLAN. The computer is used by the operator to regulate the control system, enable thrusters and log measurements.

A Matlab/Simulink diagram was provided by Astrid H Brotkorv, developed to communicate with NI Veristand 2014. The code included among other components a DP controller, a NPO estimator and a Thrust allocation block. These blocks were reused with approval from Brotkorv. The DP controller together with the thrust allocation was developed for Cybership 3 with fixed azimuths. The bow thruster was fixed in 90 degrees, mimicking a tunnel thruster, while angle α_1 and α_2 in Figure 3.5 was fixed to 30 and -30 degrees, respectively.

3.4.1 Challenges in the Lab

Both the Q-learning path planner and the manoeuvring modes in the hybrid control system were developed to work in transit. However, the space available for measurements by the infrared cameras was limited. In order to deal with this issue, the towing carriage with the attached cameras was moved in a constant speed in positive x-direction to follow the movement of the vessel. The input to the control system then was manipulated to produce a fixed coordinate system. This way, the basin's full length was exploited.

Another challenge was the environmental forces in the lab. The Q-learning path planner is developed to take into account the environmental forces of current. However, the MCLab is not equipped to produce

Figure 3.4: Photo of Cybership 3 from MCLab

this force. Hence, in order to add a current force to the tests, a virtual velocity in the positive x-direction was added for certain areas of the lab. When the vessel entered these areas, the coordinate system and velocity of the vessel was manipulated, resulting in the effects of surface current.

3.4.2 Hybrid Control

The hybrid control system was tested for the lab experiments using the same supervisory switch setup as was used for the simulations. The towing carriage was moved during the tests to take full advantage of the size of the basin. In the lab was the hybrid control system tested using only the DP controller described in Section 2.4.1 in addition to the heading controller described in Section 2.4.2. This was done due to the limited time available for the lab experiments. As each controller in combination with the supervisory switch system needs tuning, the modes of the hybrid control system was reduced to:

1. Low speed maneuvering (DP)

4. Transit

For mode 1: Low-speed manoeuvring was the DP controller provided by Astrid Brotkorv tuned and reused. The reference model from Section 2.5.1 was also used. However, the reset of the reference model when the switch was made back to mode 1 was not implemented for the experimental tests due to limited time. A discussion on how this affected the results will be presented in Section 6.

For Mode 4: Transit, the PID heading controller described in Section 2.4.2 was used to produce τ_ψ. The LOS guidance law from section 2.5.3 was also used. For the heading controller, the two rare azimuths was set to produce $\tau_x = 0.05$ each. The resulting τ_ψ from the guidance and heading controller was produced by the tunnel thruster as the only actuator controlling the heading of the vessel. The test was conducted moving the vessel in positive x-direction in the $\{\text{n}\}$-frame, alternating between mode 1 and mode 4 based on a set of waypoints.

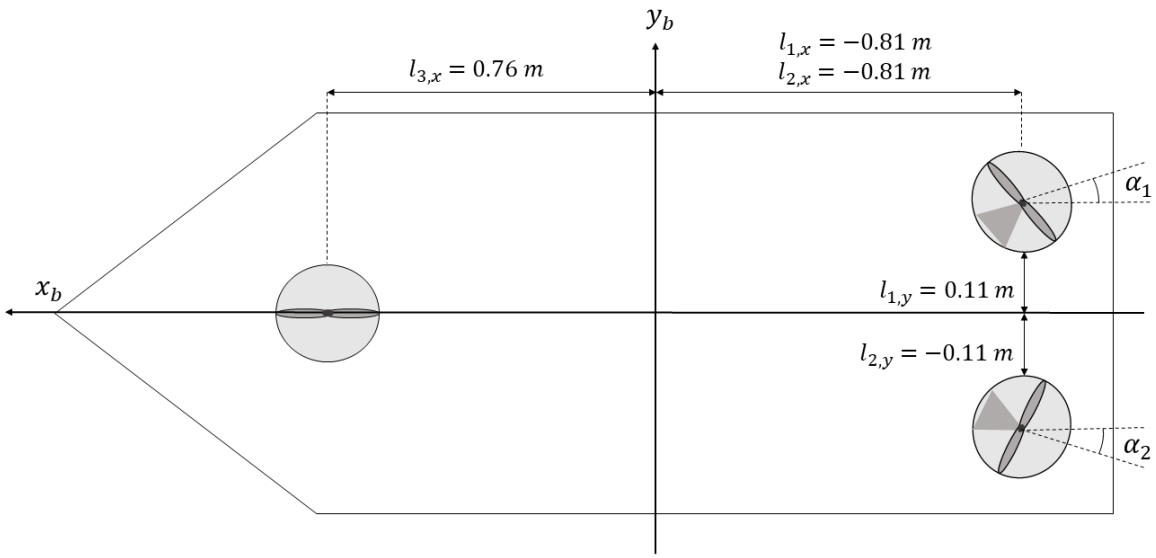


Figure 3.5: Cybership 3 thrust allocation setup

Table 3.4: Controllers used for the modes in lab

Controller	**Mode 1**	**Mode 4**
DP Controller	×	
Heading Controller		×

Table 3.5: Reference models used for the modes in lab

Reference Model	**Mode 1**	**Mode 4**
Position and Attitude	×	
LOS guidance law		×

3.4.3 Path Planner

The verification of the Q-learning path planner using current case (I) was also tested for the lab. The path found by the Q-learning path planner was tested against an calculated shortest distance path. The test was conducted using the heading controller while also moving the towing carriage during the test to exploit the full basin length. In addition, a virtual current was added to simulate the current case (I), as displayed in the left side of Figure 3.3. For this test, the control system used for the hybrid control system test was alternated to only include the guidance and heading controller.

The LOS guidance system was programmed to find the desired heading of the vessel and decide the desired waypoint, based on the lookahead-based steering law described in Section 2.5.3. A circle of acceptance was also implemented for this test with a radius of two meters. This is a tight radius, as two ship lengths often are used as a guideline. However, to mimic a trajectory more realistic to a transit operation, the whole length and width of the basin was used, leaving high demands on the accuracy when turning the vessel to avoid it crashing into the basin wall. Hence, a small radius of acceptance was chosen to force the vessel to follow the waypoints perfectly.

Chapter 4

Simulation Results

This chapter contains the simulation results. First, the results from the hybrid control system are presented, before the optimal path from the Q-learning path planning algorithm is displayed in addition to the verification on the optimal paths. The tuning parameters for the simulations can be found in appendix.

4.1 Hybrid Controller

The hybrid control system was tested with the environmental forces from current case (II) in simulation. The optimal waypoint-set found by the Q-learning path planner was used for the manoeuvring phase of the test. The simulation results from each of the controllers in addition to a tuning test of the LOS guidance law will follow in order to present the performance of each component. Then, the plots that demonstrate the overall performance and stability of the hybrid controller will be shown. The trajectory, waypoints, active modes and phase for each waypoint are summarised in Figure 4.1 and Table 4.1.

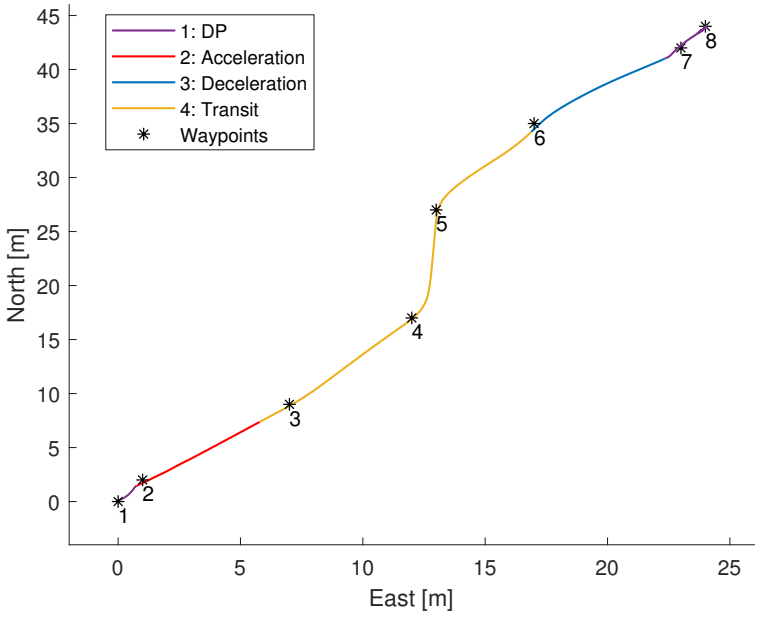


Figure 4.1: Summary of hybrid control simulation setup test

4.1.1 DP Controller

The performance of the DP controller can be seen in Figure 4.2. The trajectory in east and north, represented by the red line, is plotted against the desired position η_{des} when mode 1 and the DP controller is active. η_{ref}, the reference setpoint is fed to the reference model plotted in blue. This plot also demonstrate the task of the reference model as it feed the DP controller with a smooth η_{des} signal.

4.1.2 Heading Controller and Speed Controller

Figure 4.3 displays the performance of the heading and speed controller when activated during mode 2, 3 and 4. The desired reference signal from the reference models are plotted in comparison to the resulting heading angle, yaw rate and surge speed.

4.1.3 LOS Guidance Controller

As current case (II) was used during the tests of the hybrid control system, the PI law described in Section 2.5.3 was enabled for the LOS guidance controller in order to counteract the current induced side-slip angle β. In order to tune this PI law, the guidance controller, determining the desired course angle χ_d, was tested with different gains for K_I. The resulting trajectories for these gains can be viewed in Figure 4.4. The tuning parameter of $K_I = 0.05$ yields the best performance and will be further discussed in Section 6. This tuning parameter for the LOS guidance law was also used in the MCLab.

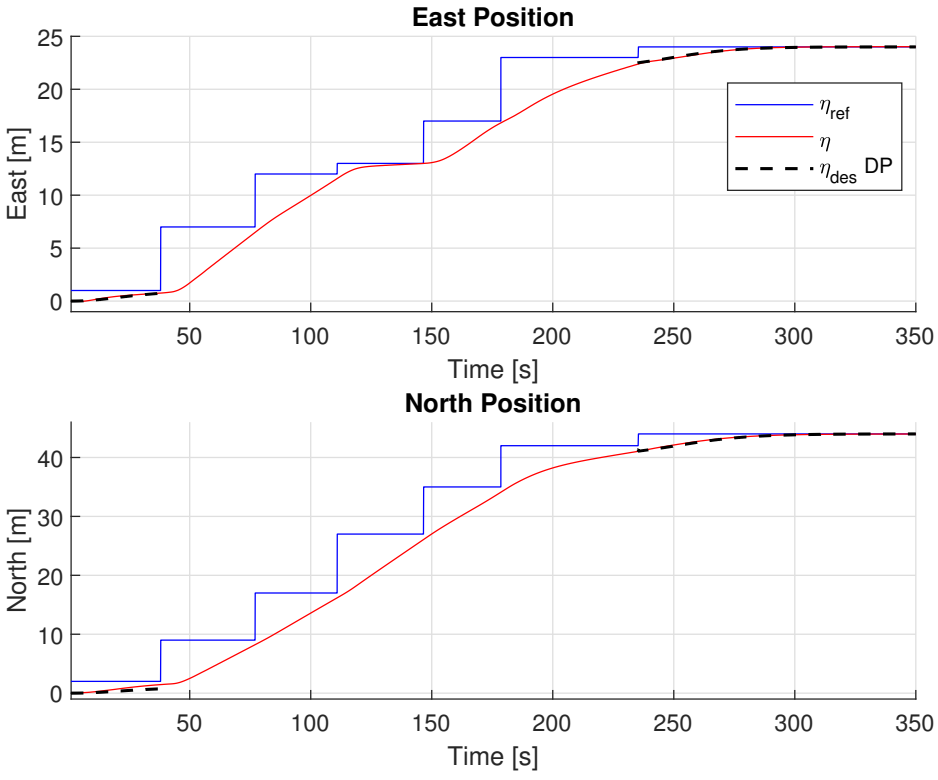


Figure 4.2: DP performance

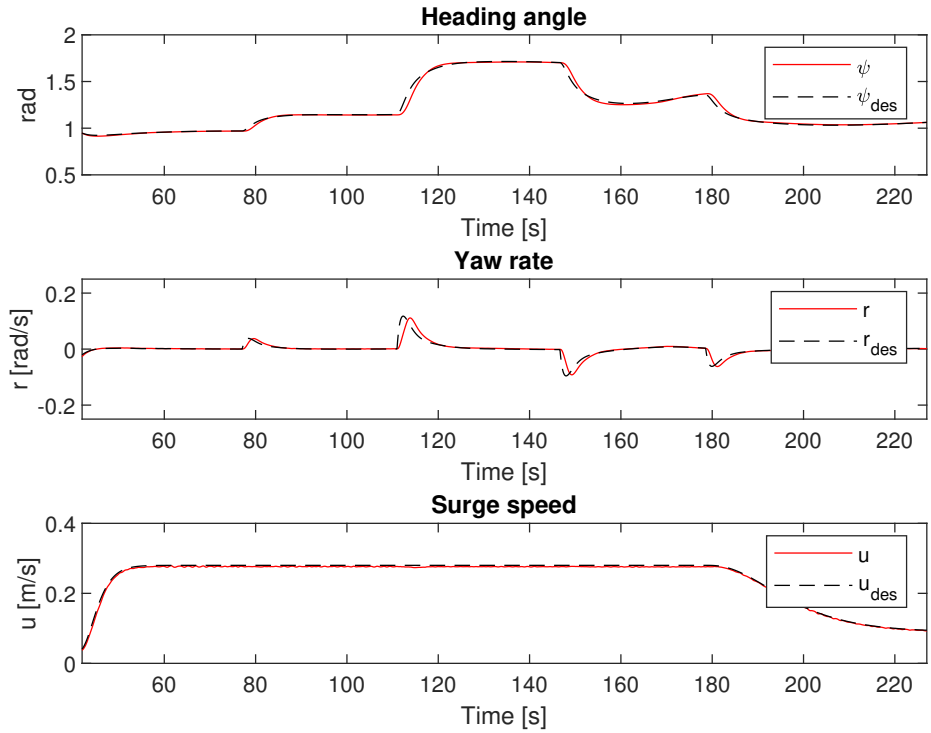


Figure 4.3: Heading performance

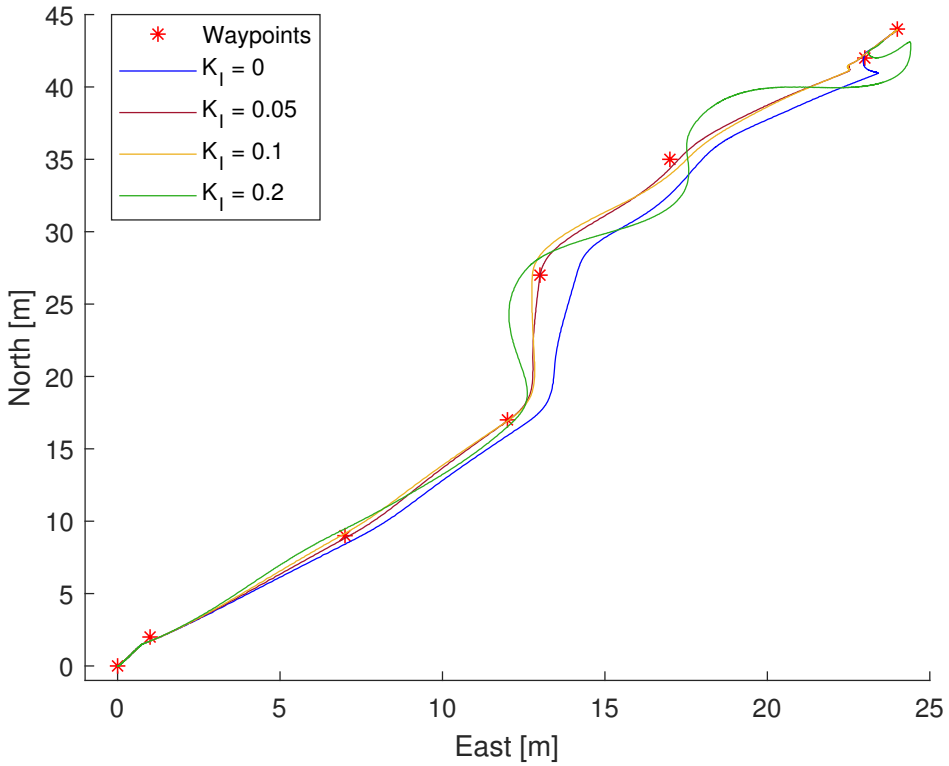


Figure 4.4: Trajectories comparing K_I gains for the guidance steering law

4.1.4 Supervisory Switch

Figure 4.5 displays the switch signal σ in blue. The value of σ define the active waypoint determining the mode of the operation. The lower figure displays the active mode. The value of one refers to DP mode, two is the acceleration mode, three represents the deceleration mode and four is transit mode. Table 4.1 present the phase of the crossing with regards to the waypoints.

The overall performance of the hybrid control system can be seen in Figure 4.6. Here the surge speed and trajectory in east and north are plotted with colors representing the different modes of the operation. Purple represents mode 1, red represents mode 2, blue represents mode 3 and yellow is mode 4. From this plot, the performance of the hybrid controller for the time instant of each switch also is observed.

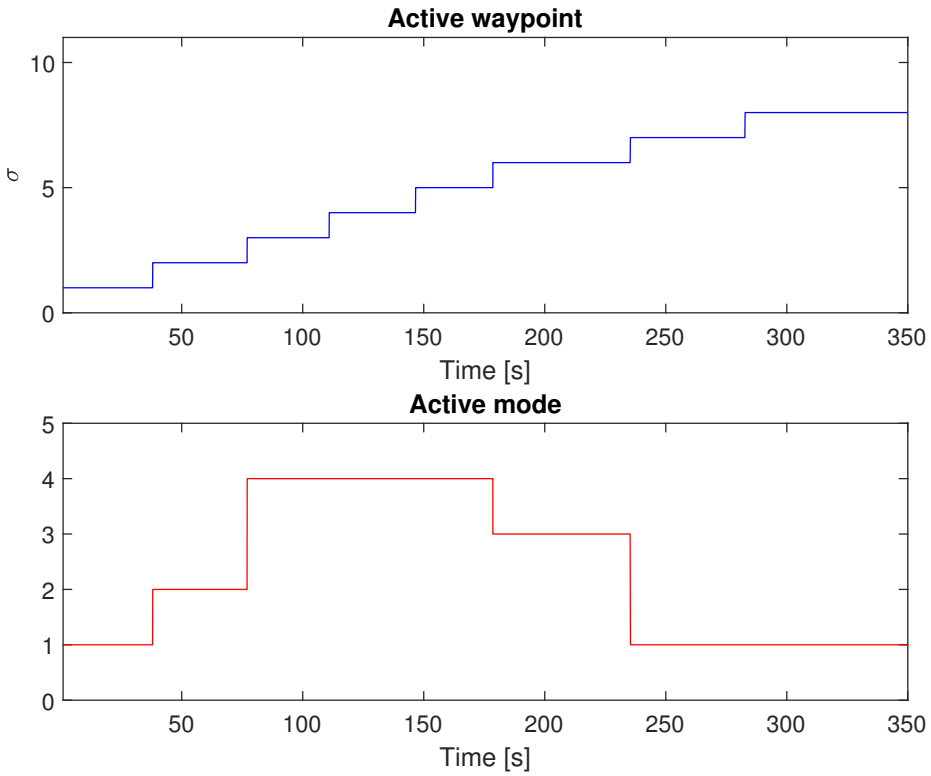


Figure 4.5: Switch performance

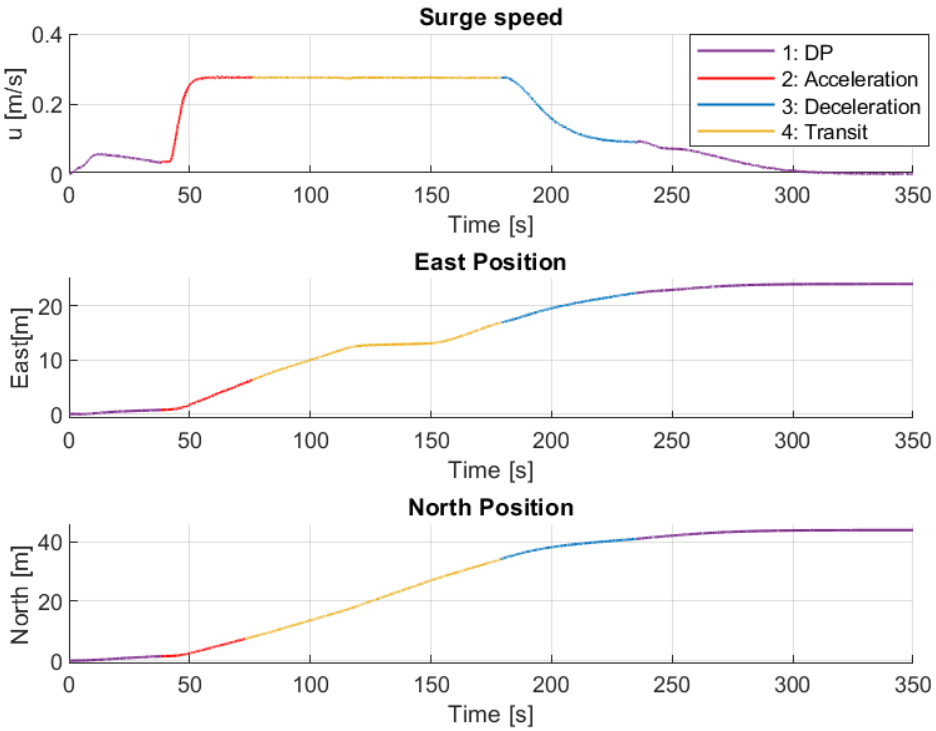


Figure 4.6: Surge speed, east position and north position for hybrid controller

Table 4.1: Phase and waypoint

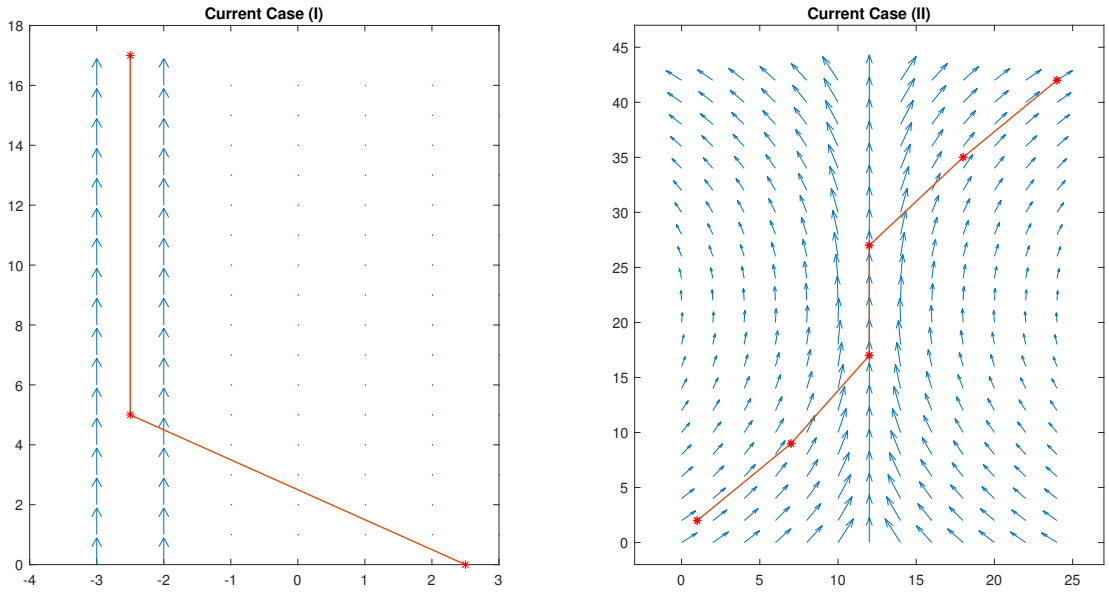
Waypoint	Phase
1-2, 7-8	DP
2-7	Manoeuvring

4.2 Path Planner

The Q-learning algorithm was tested on both current case (I) and (II) to find the optimal minimum-time path by generating a set of waypoints. The resulting optimal paths for the two cases will follow, before the paths are verified through tests with Cybership 3 using MCSim.

4.2.1 Optimal Path

The resulting minimum time paths found by the Q-learning path planner is displayed in Figure 3.3. The red stars represent the waypoints found by the path planner, while the red lines drawn between them is a way of highlighting the path a LOS guidance system would take. These results were found using the algorithm generated for the Q-learning path planner in Matlab with the current case (I) and current case (II) as environmental inputs.



Case (I): Current map for lab

Case (II): Current map for ferry crossing

Figure 4.7: Resulting minimum-time paths for current case (I) and (II)

4.2.2 Verification

The optimal path found from the Q-learning algorithm was verified through tests in MCSim using Cybership 3. The time spent traveling the Q-learning path was compared with the time spent on traveling the shortest distance path, with the given environmental current conditions.

Current case (I) represent the test setup used in the lab during the experimental tests. As preparation for the lab, the Q-learning path for this case was also verified in MCSim. The results of the test can be viewed in Figure 4.8. The path found from the Q-learning path planner is represented by the red waypoints and red trajectory, while the shortest distance trajectory is plotted in blue in the upper plot of Figure 4.8. The current and direction are represented by the light blue arrows, and the grey rectangle represents the basin walls. The lower plot in Figure 4.8 displays the x-position plotted against time for the two trajectories.

The green vertical lines mark the time instants of when the two trajectories reached the goal position, represented by the horizontal grey line. The test was conducted with the heading controller described in Section 2.2.2 calculating τ_{psi} with τ_x set to 2 N.

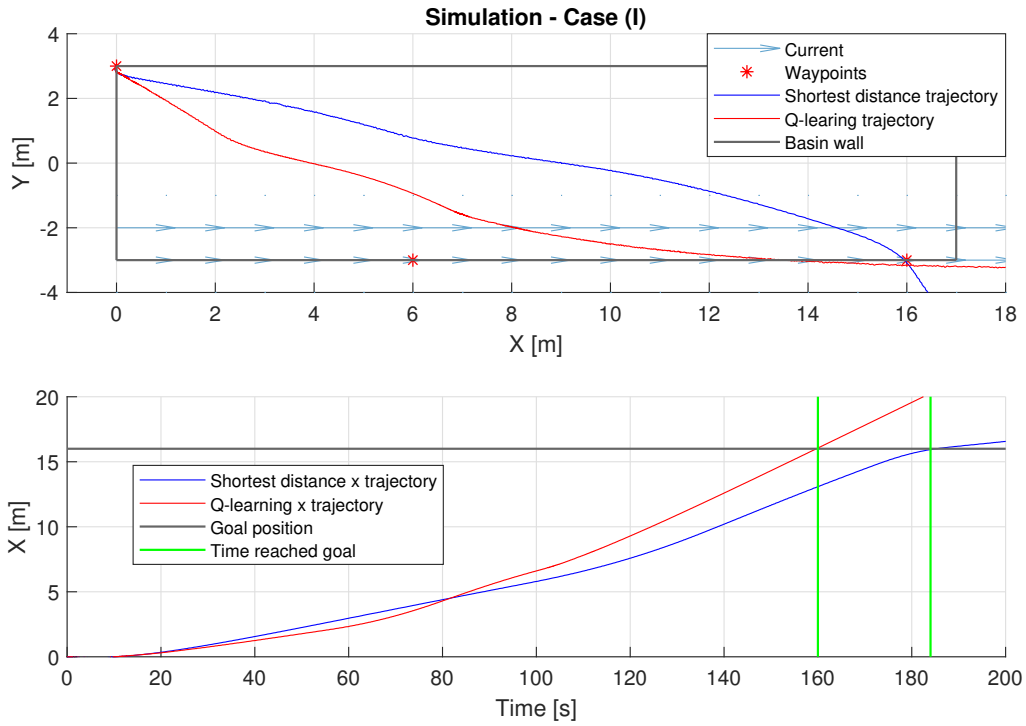


Figure 4.8: Verification of path planner - Case (I)

The optimal waypoints found for current case (II) was verified in Figure 4.9 and 4.10. Figure 4.9 display the Q-learning trajectory and shortest distance trajectory in east and north plotted over time. The vertical green lines represent the time instant for when the goal position, represented by the horizontal grey line, was reached. Figure 4.10 show the two trajectories, optimal waypoints and current map for case (II).

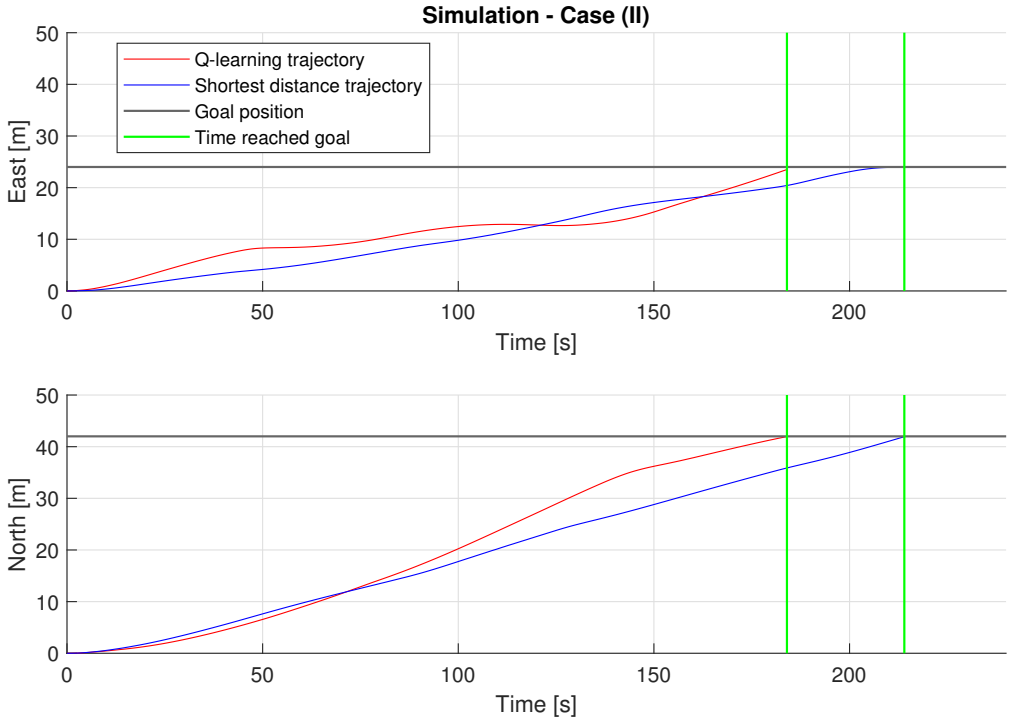


Figure 4.9: Verification of path planner - Case (II)

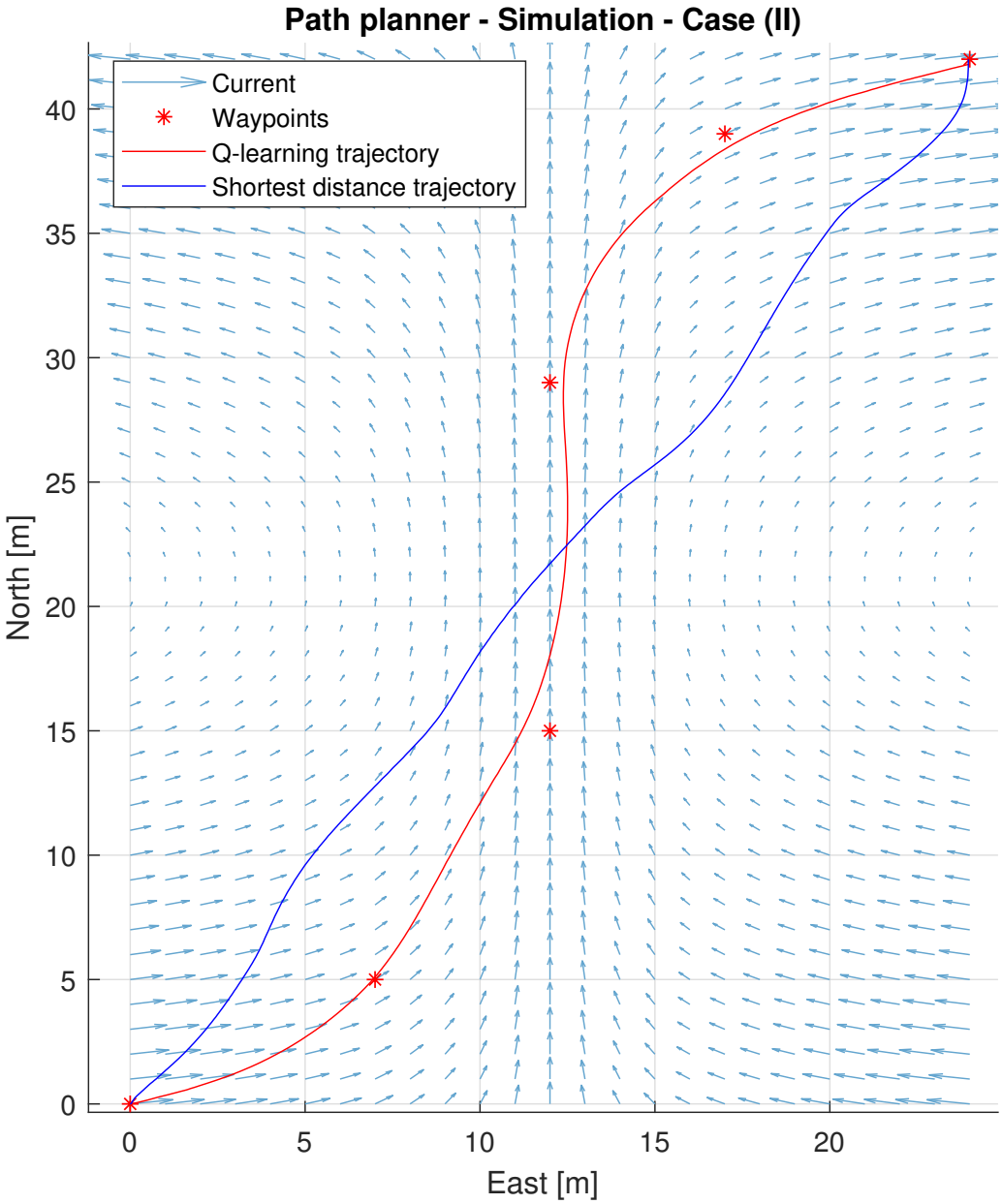


Figure 4.10: Verification of path planner - Case (II) - trajectory

Chapter 5

Experimental Results

This chapter contains the results from the lab experiments conducted in MCLab. First will the results from performance tests on each controller be presented, before the overall performance of the hybrid control system is displayed. Finally, is the results from verification of the path found by the path planner presented. The tuning parameters for the lab experiments can be found in appendix.

5.1 DP Controller

A separate DP controller test was conducted in the lab in order to evaluate the performance and robustness of the DP controller. The reference point was set to $[x, y] = [2, 0]$. When the vessel had reached its setpoint, a disturbance was induced. Figure 5.1 displays the result from the test, showing the position in both y- and x-direction. At approximately 190 seconds and 270 seconds, a force was induced in the negative and positive x-direction respectively. Then, at approximately 330 seconds and 420 seconds, a positive and negative force in the y-direction was induced respectively. This can be seen in the overshoot in the x and y-direction.

5.2 Heading Controller

The heading controller also was separately tested to evaluate and verify the performance of the controller before it was used in the hybrid control system and when testing the path planner. The controller was tested by moving in a zigzag trajectory along the x-axis. The heading angle and yaw rate were plotted together with the desired heading angle and yaw rate provided from the guidance controller, see Figure 5.2. In Figure 5.3 is the trajectory of the heading test plotted against the referenced straight line path made up by the waypoints. In addition, the circle of acceptance for each waypoint is plotted.

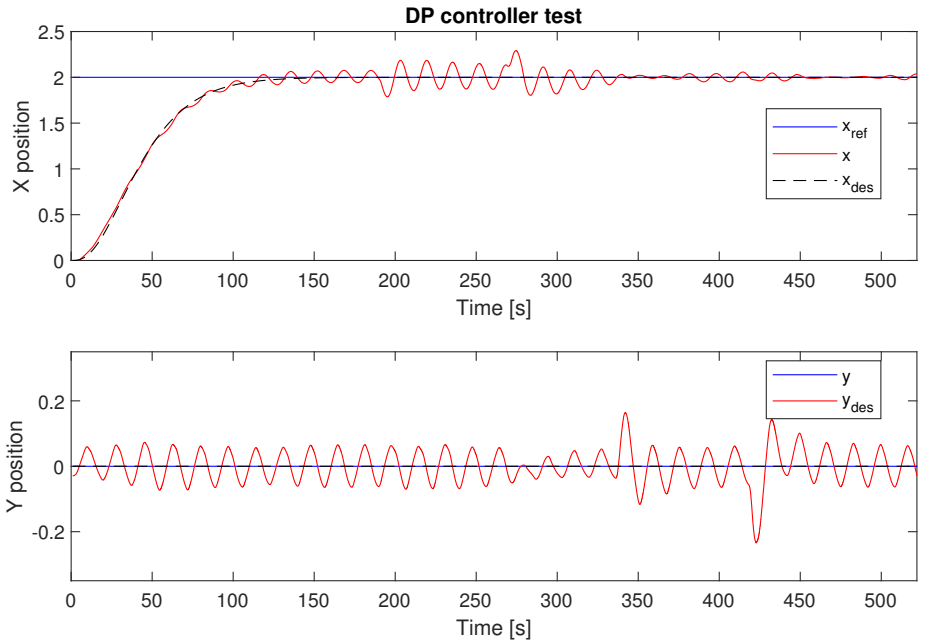


Figure 5.1: DP performance in lab

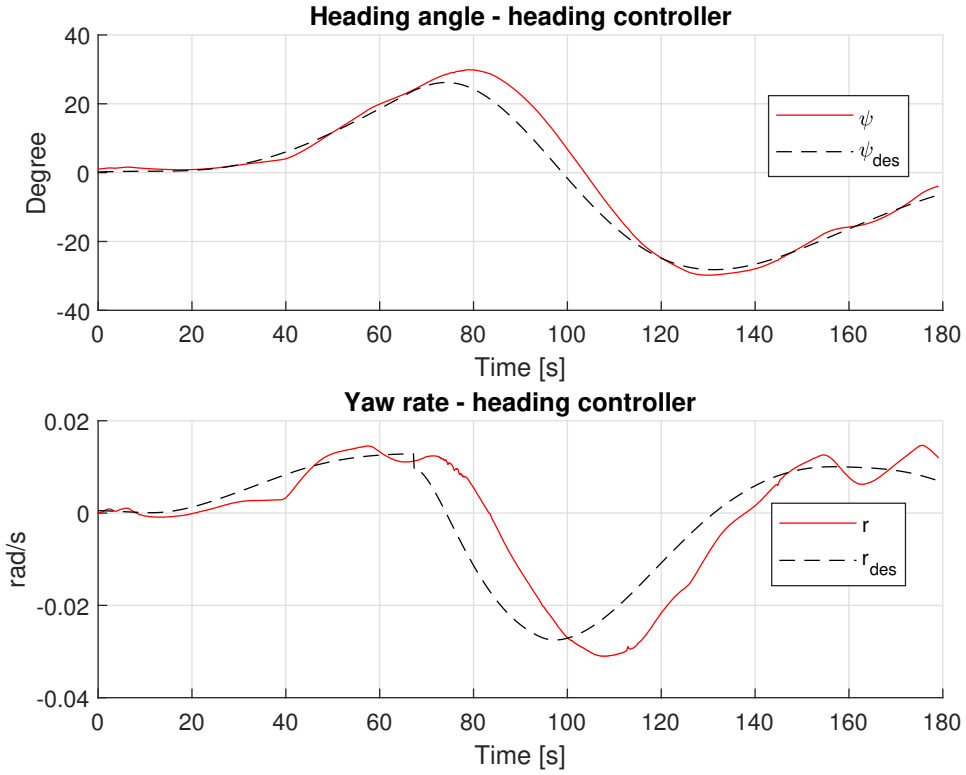


Figure 5.2: Heading performance in lab, heading angle and yaw rate

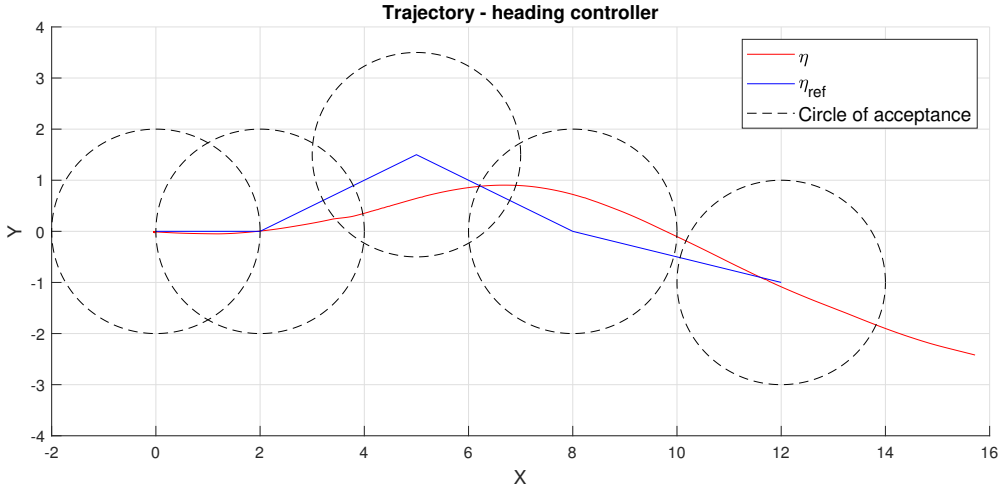


Figure 5.3: Heading performance in the lab, trajectory and circle of acceptance

5.3 Supervisory Switch

The same supervisory switch control that was used for the simulations was also used for the lab tests. Figure 5.4 displays the active waypoint in blue, represented by the σ signal that determines the next desired waypoint for the supervisory switch that consecutive determines the mode of the operation. The active mode is plotted in red, where 1 activates mode 1 and the DP controller and 4 activates mode 4 and the heading controller.

Figure 5.5 displays the referenced x and y-position of the vessel in blue together with the trajectory and desired position from the reference model when the DP controller is active. In addition, Figure 5.6 displays the surge speed of the vessel for the different modes of operation where purple represents DP mode 1 and yellow represents transit mode 4.

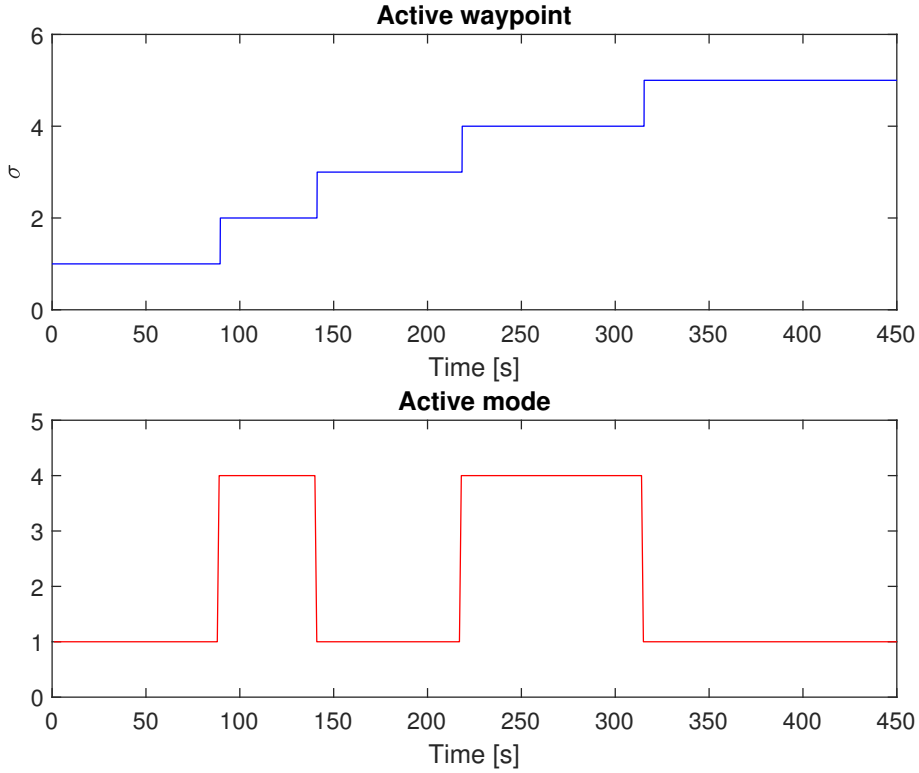


Figure 5.4: Switch performance in lab

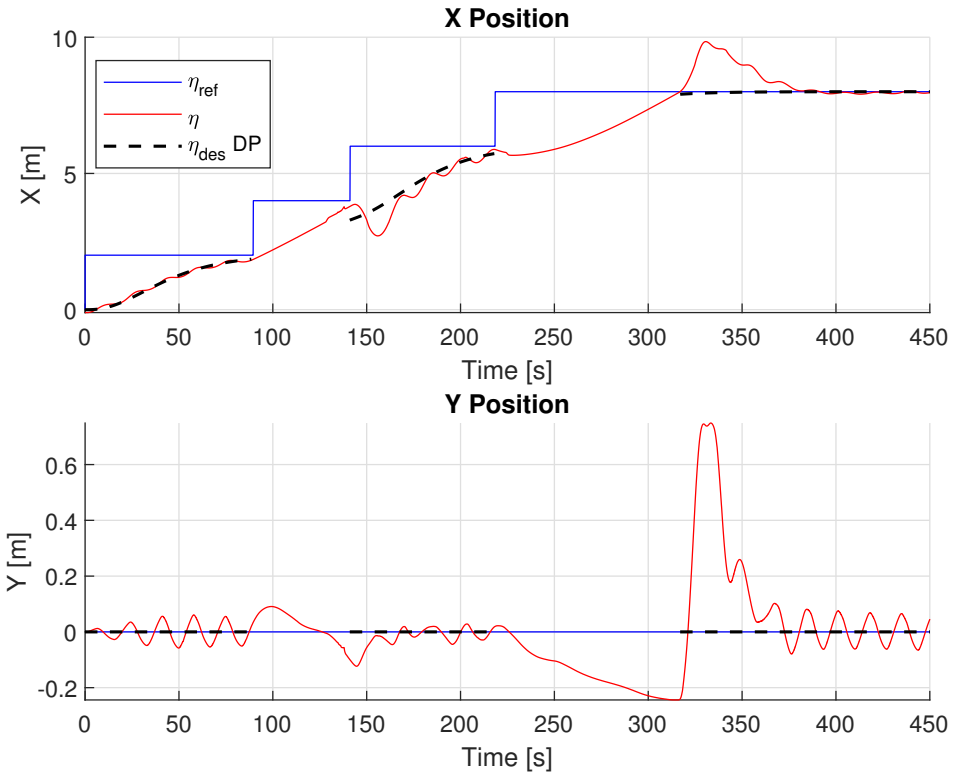


Figure 5.5: DP performance in lab

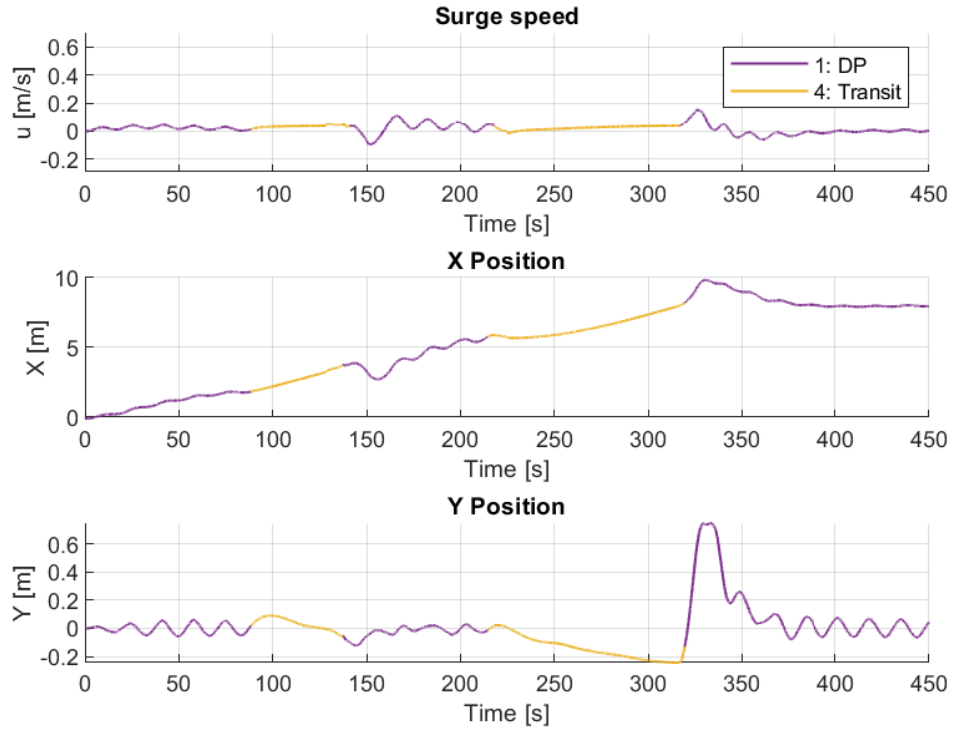


Figure 5.6: Surge speed, east position and north position for hybrid controller

5.4 Path Planner

The heading controller was used when verifying the Q-learning path planner in the MCLab. The stern thrusters were set to produce a 0.1 N each to move the ship at a constant surge speed. The tunnel thruster in the bow was used to control the heading. As in the simulation environment, a positive current in the x-axis was applied for $y \in [-2, -3]$. As seen in Figure 5.7, the performance of the heading controller was not perfect. Hence, an additional test was conducted by using a joystick to steer the heading of Cybership 3. The resulting plot from this test can be seen in Figure 5.8.

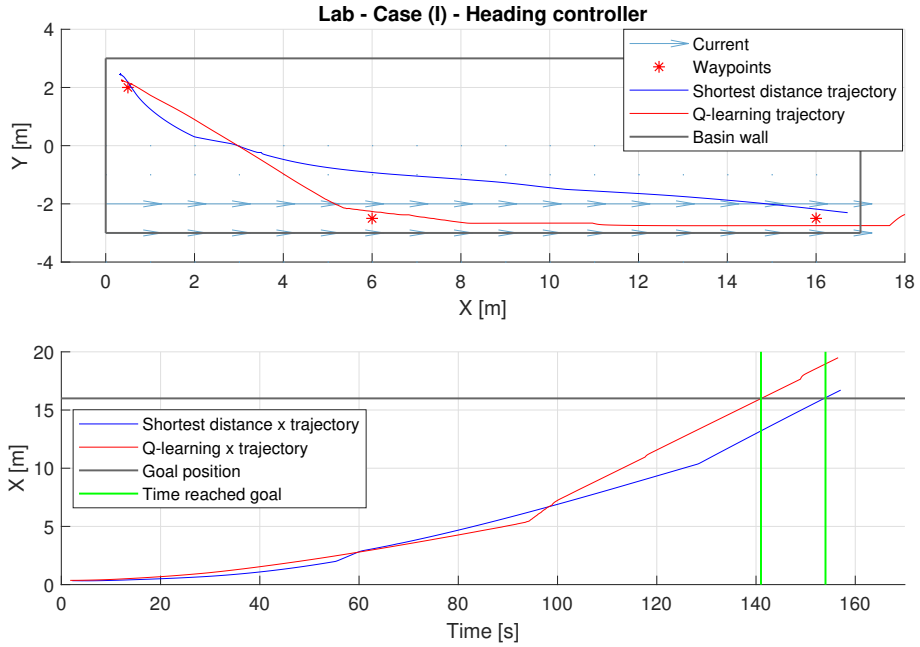


Figure 5.7: Verification of path planner in lab - Case(I) - heading controller

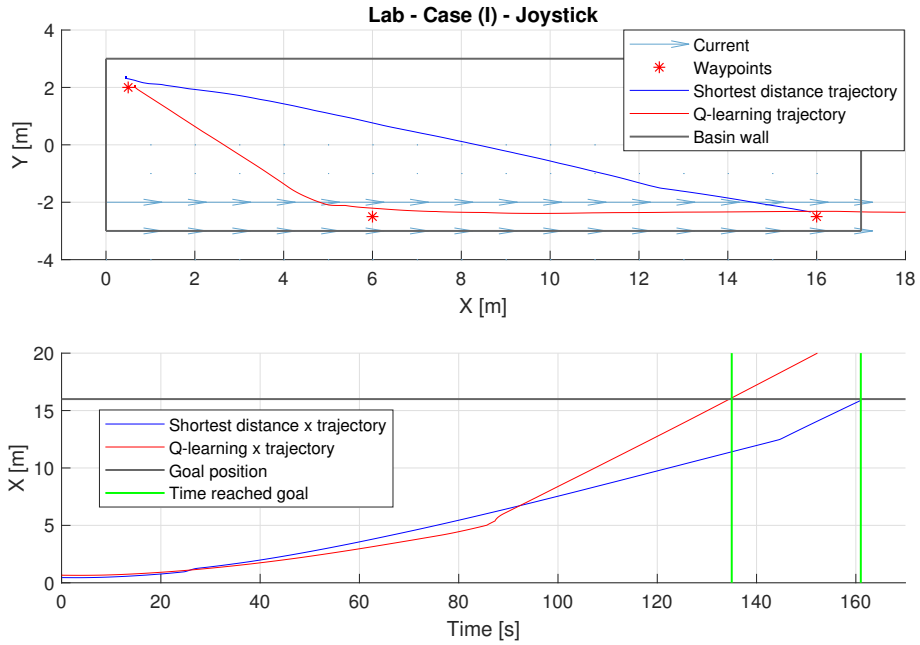


Figure 5.8: Verification of path planner in lab - Case(I) - joystick

Chapter 6

Discussion

This section presents a discussion of the work committed in this thesis and the resulting plots presented in the previous sections. First, the performance of the hybrid control will be reviewed for both the test conducted in simulations and lab before the performance of the path planner is evaluated. The new methods developed to better exploit the MCLab are also addressed.

6.1 Hybrid Control

The hybrid control was fully tested with all its components in the simulation environment. Figure 4.2 displays a good performance of the DP controller as it manages to follow the desired trajectory provided from the reference model in a satisfactory manner. It is also evident from this plot, that the reset of the reference model that was discussed in Section 2.4.1 produce a stable behaviour when switching from mode 3 back to mode 1. Hence, the DP controller combined with its reference model for the simulations performs well.

For the lab experiments, the hybrid control system was reduced only to include the DP controller with its reference model, and the heading controller with its LOS guidance law to switch between mode 1 and 4. The result from the DP controller test in Figure 5.1 shows that the DP controller manages to follow the desired position. When the disturbances were added in the x- and y-direction, the plots show that the controller manages to get back to the desired setpoint. Still, the plot also reveals an undesired oscillatory behaviour around the setpoint. There is reason to believe that the performance of the DP controller could have been improved through extensive tuning. However, due to limited time in the lab, this was not given priority.

Figure 4.3 display the simulated performance of the heading controller and guidance controller. The heading controller is showing good performance as it manages to follow the desired trajectory provided from the LOS guidance reference model. The surge speed is displayed in the bottom plot of Figure 4.3. Here it is evident that mode 2: acceleration, requires a rapid change in speed, with an aggressively tuned reference model, while mode 3: deceleration provides a slow gradual reduction of the vessel speed. However, the aggressive increase in speed for mode 2 could result in unnecessary wear and tear on the actuators. Even though the thrust allocation for Cybership 3 in the simulation environment had a saturation limit for τ_{max}, this limit was not investigated properly. Consequently, this needs to be taken into account if implementing the system for a real-life application.

The result from the test of the heading and LOS guidance controller in the lab is displayed in Figure 5.2. This figure shows that the heading controller manages to follow the desired heading angle and yaw rate, but the performance of the controller is not perfect. Extensive tuning would also here probably improve the performance of the heading controller. Figure 5.3 displays the trajectory of the vessel for the heading control, together with the circle of acceptance in the lab. From this plot, the issue regarding the size of the basin, when considering transit operations, become evident. The LOS guidance law is struggling with determining the current waypoint as the circle of acceptance is overlapping for some waypoints. Still, the circle of acceptance was set to a radius of only 2 meters, even though the recommended radius for the LOS guidance law is twice the size for Cybership 3. Reducing the circle of acceptance further is not recommended, as this could lead to an unfeasible problem for the guidance law. However, the problem could be solved by expanding the distance between the waypoints. Still, this would result in a limitation on the number of waypoints possible for a test in MCLab. The potential for testing transit operations in the MCLab therefore could be debated.

The results from tuning the LOS guidance law controller in the simulation environment MCSim was presented in Figure 4.4. This displays the effect of tuning the integral term for the controller, counteracting the sideslip angle β. As seen from the plot, it is evident that an integral gain of $K_I = 0.05$ yields the best performance in following a logical path between the waypoints. $K_I = 0$ is resulting in unsatisfactory behaviour as the vessel is drifting off, while a larger gain of $K_I = 0.2$ displays overshooting in the integral compensation.

The overall performance of the hybrid control system in the simulation environment can be viewed in Figure 4.5 and 4.6. From the plot in Figure 4.5 displaying the switch signal σ and the active mode for each time instant, it is evident that the chattering issue that might arise for supervisory switch control is avoided for our system. In Figure 4.6 the speed and position in north and east are plotted over time. These plots display smooth trajectories that show little effect on the switching.

The overall performance of the hybrid control for the lab setup is displayed in Figure 5.4 and 5.5. The stability for the lab set up is also kept intact as Figure 5.4 shows no chattering. Figure 5.5 also show that the vessel manages to follow the desired trajectory, even though the system experience some overshooting behaviour when switching. This might be due to the aforementioned effect of not resetting the reference model used for mode 1 when switching. One could, therefore, assume that this could have been resolved by implementing this also for the lab setup.

6.2 Developments in MCLab

The need for testing the system with environmental forces in transit operation inspired an innovative approach when preparing for the lab experiments. The result was a way of taking advantage of the towing carriage in the lab to fully exploit the whole length of the basin, in addition to virtually adding a surface current. However, as previously mentioned, one could argue that the basin still is too small for testing in transit manoeuvring, taking into account the results from the heading test with overlapping circles of acceptance. Additionally, the current force only results in an added velocity in the defined direction without taking into account the dynamic effect such a current would cause on the vessel. Nevertheless, the development of these approached expand possibilities for testing and use of the MCLab.

6.3 Path Planner

For the resulting optimal minimum-time paths presented in Figure 4.7 is the Q-learning algorithm displaying a promising performance. The resulting paths display a logic trajectory when taking into account the effect of the current. The verification tests that were performed in Figure 4.9 and 4.8 show that the path found by the Q-learning algorithm is, in fact, the most time efficient one compared to the shortest distance path for the given current cases in the simulation environment. Also, the tests conducted in the lab for the current case (I) verify the Q-learning path planner. Even the results from using the non-optimal heading controller in Figure 5.7 prove the performance of the algorithm. Hence, the verification of the paths in both the simulation environment and lab demonstrates that the path planner achieves to identify the minimum time paths with the given environmental conditions.

By evaluating the resulting desired path provided by the Q-learning algorithm, it is evident that the modelling of the problem is important for the result. An agent that learns a Q-function does not need a model, neither for learning or selection, and is therefore called a model-free method. This simplifies the learning problem, but potentially restricts the ability to learn in complex environments. However, our implementation ads a simplified model of the vessel to make the agent better mimic the dynamics of a real-world ship. Still, the heading dynamics of the vessel was not included. This results in a system not taking into account the effect of the heading angle when travelling through the current. This would undoubtedly affect the result, as information on the optimal heading for each waypoint could further optimize the path.

However, when finding the optimal paths in the Q-learning algorithm, the accuracy and performance of the path planner was somewhat varying for different environmental conditions. It was not even guaranteed to conclude on a suitable path, sometimes resulting in an unsolved problem. In addition, one current setup could result in different solutions for each run, not necessarily converging towards a desirable solution. Furthermore, the simulation time was a considerable issue. Small changes, as moving towards a more complicated system, resulted in an extensive increase in the time spent to achieve convergence for the path planner. Hence, the simplification of the vessel dynamics, resolution on the discretization of the map, placement of the goal state, and the number of available actions had to be weighted against the convergence rate of the path planner. The stability of the Q-learning path planner could therefore be questioned.

In addition, the resulting optimal path is affected by the coarse resolution of the discretization of the map. A finer mesh would result in a smoother and more accurate path. This would especially be preferable for the calculated path close to shore when inaccuracy in the path could be fatal. However, this would come with a trade-off on computational efficiency, as already mentioned. The computational efficiency could be an issue if the path planning algorithm was executed during the crossing to detect changes in the environmental conditions, demanding a fast computation time. A solution could be to have a varying resolution for the discretization of the map, with a finer mesh in areas with a high demand for accuracy.

Chapter 7

Conclusion and Future Work

In this master thesis, the performance of an autonomous control system for a ship has been evaluated for a crossing taking into account environmental forces as waves. This chapter concludes the thesis presenting the major results and making suggestions for future work.

7.1 Conclusion

In this thesis, a hybrid controller and a machine learning based path planner were developed in order to implement an autonomous control system for a vessel. The hybrid control system was divided into four modes: Low-speed manoeuvring, Acceleration, Deceleration and Transit. A DP controller, heading controller and speed controller were developed to control the vessel for the different modes. The modes and controller set has been modelled based on the framework proposed by Goebel (2012). A supervisory switch control was developed based on scale-independent hysteresis switching logic defined in Hespanha (2001) and Nguyen et al. (2007). This structure is beneficial due to perceived stability when switching between linear/nonlinear controllers and models. A path planning algorithm was developed based on the machine learning theory of reinforcement learning with a Q-learning algorithm. Q-learning is a powerful tool of reinforcement learning as it does not need a model of the environment to calculate the desired path. This algorithm took the effect of current into account when calculating the minimum-time optimal path for the vessel. Both the performance of the hybrid control system and verification of the path planner was tested both in simulation and through lab experiments.

Based on the major findings from the simulations and lab experiments, the overall implementation of the autonomous control system yield a satisfactory performance. The system manages to take the vessel from A to B through the optimal path calculated by the path planning algorithm without human intervention. The Q-learning path planning algorithm manages to find an optimal path for the vessel, and the modelling of the current is affecting the response of the algorithm as expected from a physical point of view. In addition, the stability of the hybrid control system is retained when switching between modes.

However, the implemented system poses some limitations with regards to the computational efficiency for the Q-learning algorithm. In addition, the controllers in the hybrid control systems could be further tuned to reduce wear and tear on the actuators and increase the performance of the system.

To summarize, as the combined system manages to find an optimal path and then proceeds to follow that path in a satisfactory manner, one can conclude that the concept of combining hybrid control and machine learning to achieve a high level of autonomy has been proved.

7.2 Future Work

Even though the resulting implementation of the hybrid control system and path planning algorithm works in a satisfactory manner, there still are improvements that could be made to the system. The hybrid control system is as of now divided into four modes. In order to improve the performance in a real-life implementation, the system could benefit from further dividing these modes. For instance, a DP system could be optimized to stay in one position while for instance a ferry is being loaded or offloaded. In addition, there might be areas in a fjord where the rules demand lower speeds than other areas.

A shortcoming for the path planning algorithm is the poor computational efficiency, in addition to it not always finding an optimal path. A way to investigate this issue could be to introduce a more accurate model of the agent, taking into account the aforementioned yaw dynamics, to see if this would benefit the success-rate of the path planner. The simulations performed in this master has been operated on what could be considered a standard performing personal computer. A solution to the computational efficiency issue could, therefore, be to simply run the algorithm on a stronger computer to reduce the computation time. It would also be interesting to explore the possibility of varying the mesh resolution when discretizing the map.

In order to further improve the autonomous system it could be extended to running the path planning algorithm multiple times during the operation in order to detect changes in the environmental conditions. However, this would demand a sensor system able to detect such changes in addition to improved computational efficiency for the Q-learning algorithm.

Bibliography

Bishop, C. M., 2006. Pattern recognition and machine learning.

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

Garau, B., Alvarez, A., Oliver, G., 2005. Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach. Vol. 2005. IEEE, pp. 194–198.

Goebel, R., 2012. Hybrid dynamical systems : modeling, stability, and robustness.

Hassani, V., Pascoal, A. M., Onstein, T. F., 2018a. Data-driven control in marine systems. Annual Reviews in Control 46, 343–349.

Hassani, V., Pascoal, A. M., Sørensen, A. J., 2018b. Detection of mooring line failures using dynamic hypothesis testing. Ocean Engineering 159, 496–503.

Hassani, V., Sørensen, A. J., Pascoal, A. M., 2012. Robust dynamic positioning of offshore vessels using mixed- synthesis part ii: Simulation and experimental results. IFAC Proceedings Volumes 45 (8), 183–188.

Hespanha, J. P., Dec. 2001. Tutorial on supervisory control. Lecture Notes for the workshop *Control using Logic and Switching* for the 40th Conf. on Decision and Contr., Orlando, Florida, available at `http://www.ece.ucsb.edu/~hespanha/published`.

Hespanha, J. P., Liberzon, D., Morse, A., 2003. Hysteresis-based switching algorithms for supervisory control of uncertain systems. Automatica 39 (2), 263–272.

Hespanha, J. P., Morse, A., 2002. Switching between stabilizing controllers. Automatica 38 (11), 1905–1917.

Holzhüter, T., 1990. A high precision track controller for ships. IFAC Proceedings Volumes 23 (8), 425–430.

Hunt, K., Haas, R., Kalkkuhl, J., 1996. Local controller network for autonomous vehicle steering. Control Engineering Practice 4 (8), 1045–1051.

Kim, H., Lee, T., Chung, H., Son, N., Myung, H., 2012. Any-angle path planning with limit-cycle circle set for marine surface vehicle. IEEE, pp. 2275–2280.

Lawless, W. F., 2017. Autonomy and artificial intelligence: A threat or savior?

Mirosław, T., 2017. Hybrid switching controller design for the maneuvering and transit of a training ship. International Journal of Applied Mathematics and Computer Science 27 (1), 63–77.
URL `https://doaj.org/article/67cfd0cb17fe4e369565fcb1f67734fd`

Mohammed, M., Khan, M. B., Bashier, E. B. M., 2016. Machine Learning, 1st Edition. CRC Press.

MUNIN, 2016. Maritime unmanned navigation through intelligence in networks, munin.
URL http://www.unmanned-ship.org/munin/about/the-autonomus-ship/

National-Academies-Press, 2005. Autonomous vehicles in support of naval operations.

Nguyen, T. D., SA[cedilla]rensen, A. J., Quek, S. T., 2007. Design of hybrid controller for dynamic positioning from calm to extreme sea conditions.(author abstract). Automatica 43 (5).

Pietrzykowski, Z., Malujda, R., 2018. Autonomous ship – uesponsibility issues. Vol. 897. Springer Verlag, pp. 395–410.

Rolls-Royce, 2016. Advanced autonomous waterborne applications initiative, aawa.
URL https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf

Russell, S. J. S. J., 2016. Artificial intelligence : a modern approach.

Sheridan, T. B., Verplank, W. L., 1978. Human and computer control of undersea teleoperators. Tech. rep.
URL http://handle.dtic.mil/100.2/ADA057655

Stefanovic, M., Safonov, M., 2011. Safe Adaptive Control: Data-Driven Stability Analysis and Robust Synthesis. Vol. 405 of Lecture Notes in Control and Information Sciences. Springer London, London.

Sørensen, A., 2013. Marine Conrol Sstems. Propulsion and Motion Control of Ships and Ocean Structures, 1st Edition. Department of Marine Technology, NTNU.

Sørensen, A., Fall. 2018. Autonomous marine operations and systems from space to ocean space. Lecture Notes for the course TMR4240 Marine Control Systems I, Department of Marine Technology, NTNU, available at ,.

Thieme, C. A., Utne, I. B., 2017. A risk model for autonomous marine systems and operation focusing on human–autonomy collaboration.
URL http://hdl.handle.net/11250/2450091

Torben, T., 2018. Hybrid control of autonomous ferries.

Trong Dong Nguyen, A., Sorensen, A., Ser Tong Quek, A., 2008. Multi-operational controller structure for station keeping and transit operations of marine vessels. Control Systems Technology, IEEE Transactions on 16 (3), 491–498.

TU.no, 2018a. Verdens første helt autonome fergeseilas gjennomført - teknologien er 100 prosent klar.
URL https://www.tu.no/artikler/verdens-forste-helt-autonome-fergeseilas-gjennomfort-teknologien-er-100-prosent-klar/452610

TU.no, 2018b. Yara birkeland.
URL https://www.tu.no/emne/yara-birkeland

Wang, H., Yu, Y., Yuan, Q., 2011. Application of dijkstra algorithm in robot path-planning. IEEE Publishing, pp. 1067–1069.

Yang, Y., Pang, Y., Li, H., Zhang, R., 2014. Local path planning method of the self-propelled model based on reinforcement learning in complex conditions. Journal of Marine Science and Application 13 (3), 333–339.

Yoo, B., Kim, J., 2016. Path optimization for marine vehicles in ocean currents using reinforcement learning. Journal of Marine Science and Technology 21 (2), 334–343.
URL http://search.proquest.com/docview/1791230251/

Appendix

Simulation Parameters

Controller Gains

Table 7.1: Controller parameters used in simulation

Description	Symbol	Value
Heading controller proportional gain	$K_{P-heading}$	10
Heading controller integral gain	$K_{I-heading}$	0
Heading controller derivative gain	$K_{D-heading}$	30
Speed controller proportional gain	$K_{P-speed}$	200
Speed controller integral gain	$K_{I-speed}$	0.2
DP controller proportional gain	K_{P-DP}	$diag(7, 30, 4)$
DP controller integral gain	K_{I-DP}	$diag(30, 30, 30)$
DP controller derivative gain	K_{D-DP}	$diag(1.5, 7, 0.9)$

Reference Models

Table 7.2: Reference model parameters used in simulation

Description	Symbol	Value
DP reference model omega	ω_{DP}	$[0.1, 0.097, 0.1]$
DP reference model zeta	ζ_{DP}	$[1, 1, 1]$
Speed reference model 1 omega	$\omega_{speed-mode2}$	0.62
Speed reference model 1 zeta	$\zeta_{speed-mode2}$	1
Speed reference model 2 omega	$\omega_{speed-mode3}$	0.15
Speed reference model 2 zeta	$\zeta_{speed-mode3}$	1
Heading reference model omega	$\omega_{heading}$	0.9
Heading reference model zeta	$\zeta_{heading}$	1
Heading guidance proportional gain	$K_{P-guidance}$	1
Heading guidance integral gain	$K_{I-guidance}$	0.05
Circle of acceptance radius	R_{sim}	5

Supervisory Switch Control

Table 7.3: Supervisory switch control parameters used in simulation

Description	Symbol	Value
Non-negative forgetting factor	λ	1
Hysteresis constant	h	0.1
Smallest value for μ_ρ when in DP mode	h_{DP}	0.2
Smallest value for μ_ρ when in Manoeuvring mode	h_{Man}	1.5

Q-learning Algorithm

Table 7.4: Q-learning algorithm parameters

Description	Symbol	Value
Discount factor	γ_Q	0.9
Learning rate	α_Q	1
Learning function	$\alpha_Q - function$	randi()
time-step	t	0.05

Lab Parameters

Controller Gains

Table 7.5: Controller parameters used in lab

Description	Symbol	Value
Heading controller proportional gain	$K_{P-heading}$	1
Heading controller integral gain	$K_{I-heading}$	0.001
Heading controller derivative gain	$K_{D-heading}$	3
DP controller proportional gain	K_{P-DP}	$diag(3.2714, 6.5319, 1.12)$
DP controller integral gain	K_{I-DP}	$diag(36.9495, 38.4064, 14.1977)$
DP controller derivative gain	K_{D-DP}	$diag(0.2068, 0.2024, 0.0746)$

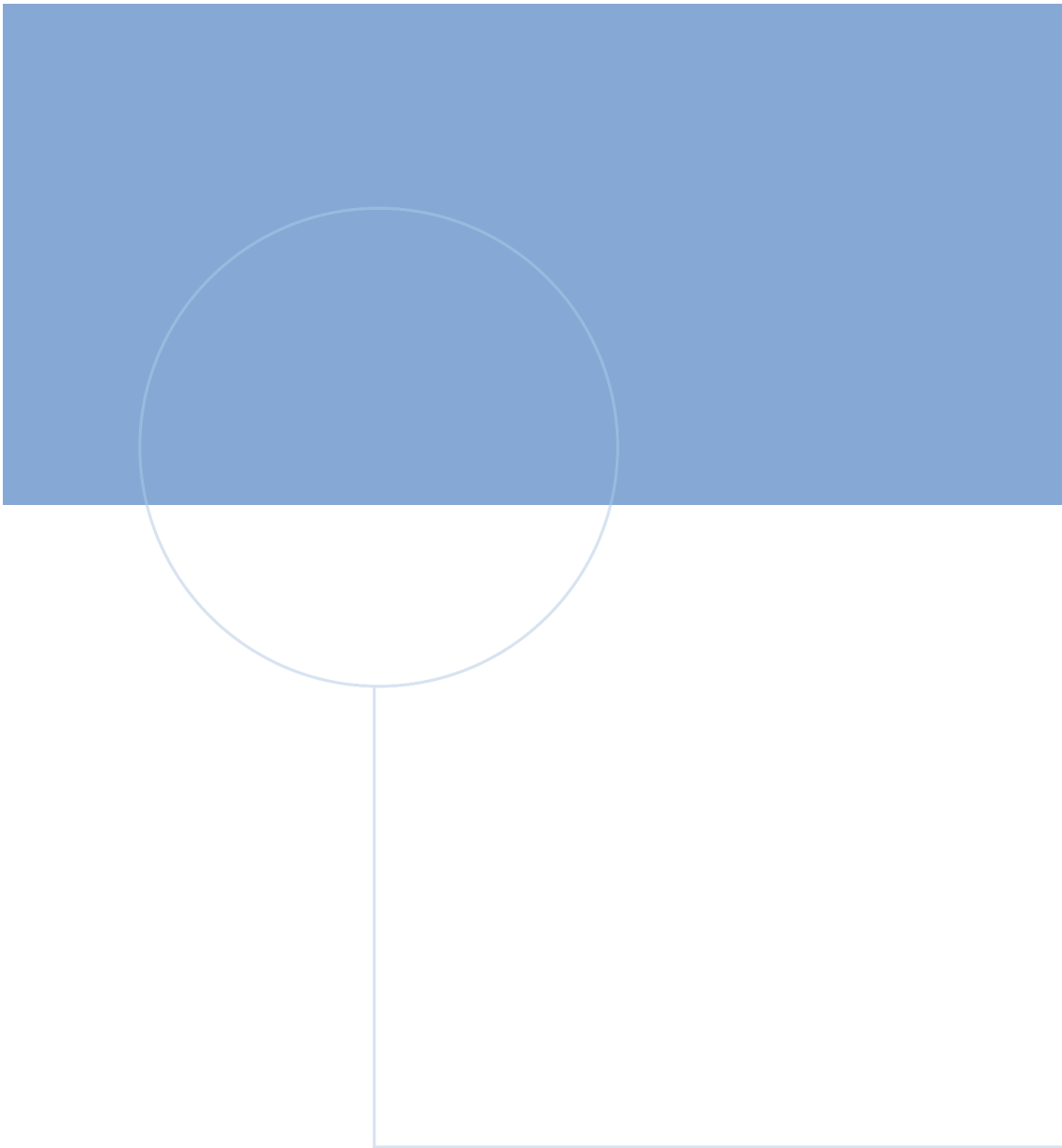
Reference Models

Description	Symbol	Value
DP reference model omega	ω_{DP}	$[0.065, 0.065, 0.055]$
DP reference model zeta	ζ_{DP}	$[1, 1, 1]$
Heading reference model omega	$\omega_{heading}$	0.1
Heading reference model zeta	$\zeta_{heading}$	1
Heading guidance proportional gain	$K_{P-guidance}$	1
Heading guidance integral gain	$K_{I-guidance}$	0.05
Circle of acceptance radius	R_{lab}	2

Supervisory Switch Control

Table 7.6: Supervisory switch control parameters used in lab

Description	Symbol	Value
Non-negative forgetting factor	λ	0.5
Hysteresis constant	h	0.1
Smallest value for μ_ρ when in DP mode	h_{DP}	0.08
Smallest value for μ_ρ when in Manoeuvring mode	h_{Man}	0.17



NTNU

Kunnskap for en bedre verden