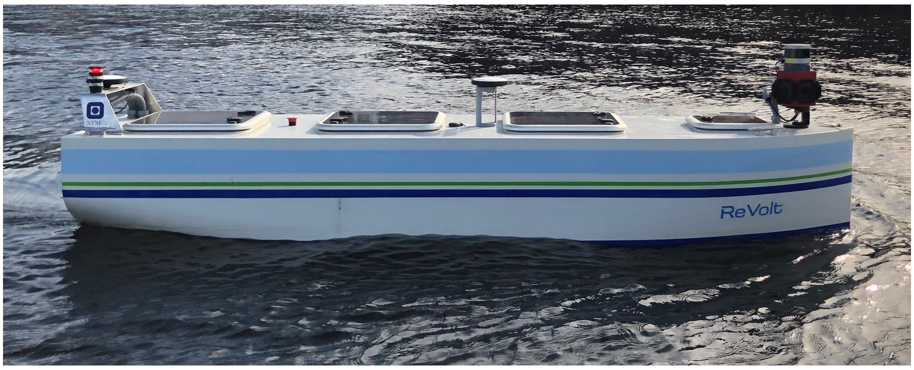
Knut Turøy

Closed-Loop Collision Avoidance System for the Revolt Model-Scale Vessel

Master's thesis in Cybernetics And Robotics
Supervisor: Edmund Brekke, Geir Hamre, Vegard Kamsvåg
July 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics



NTNU
Norwegian University of
Science and Technology

Knut Turøy

Closed-Loop Collision Avoidance System for the Revolt Model-Scale Vessel

Master's thesis in Cybernetics And Robotics
Supervisor: Edmund Brekke, Geir Hamre, Vegard Kamsvåg
July 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Preface

This thesis concludes the author's degree of Master of Science in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). It is written in collaboration with Det Norske Veritas-Germanisher Lloyd (DNV GL), and the work has been carried out under the supervision of Edmund Førland Brekke from the Department of Engineering Cybernetics (NTNU), and co-supervisors Geir Hamre and Vegard Kamsvåg from DNV GL.

This thesis is a continuation of the author's project thesis [1] from the fall semester of 2019. Some sections are partially re-used or influenced from the project thesis, as it is not published work available for the public. These sections include:

- Parts of Section 3.1 - Section 3.2
- Parts of Chapter 4
- Parts of Section 6.1 - Section 6.2

The work reported in this thesis build on previous advances made by several PhD and MSc students at NTNU. The tracking framework used in this thesis was developed by Erik Wilthil and Andreas Flåten in [2, 3], implemented on ReVolt by Håkon Norbye in [4], and further adapted by the author in [1]. The real-time sensor fusion pipeline on ReVolt was developed and implemented in [4]. The SBMPC algorithm was implemented on ReVolt by Tonje Midjås in [5], and the further development of the SBMPC was based on [6]. The simulator environment has been provided by DNV GL, along with the physical scale model ReVolt. Illustrations and figures have been created by the author unless stated otherwise.

The COVID-19 pandemic has had a significant impact on the progress of this thesis, where access to ReVolt has been restricted or limited from early March to late May. This drastically reduced the possibility of performing experimental sea trials. As a consequence, experiments for the closed-loop CAS were prioritized, and no separate experiments for verifying the tracking system's performance beforehand were conducted. Additionally, significant problems with the hardware, especially the azimuth thrusters, on ReVolt, drastically reduced the testing time of the conducted experimental sea trials.

Acknowledgement

I want to thank my supervisors for valuable discussion and guidance throughout this thesis. A special thanks must also be given to Håkon Gjertsen Norbye for his help and support with the sensor system and calibration. Further, I want to thank Simen Sem Øvereng, Magnus Oanes Knædal, Tom Arne Pedersen, and the summer interns at DNV GL for helping me carry out the experiments. Attention must also be given to Torleiv Håland Bryne for aiding GNSS equipment and Stefano Bertelli for the transportation of ReVolt.

I would also like to thank my good friend Åsmund Eek for help with structuring and proofreading this thesis. Finally, I would like to thank my wife and family for their unconditional love and support. They have stood by me throughout this journey and provided me with the freedom to pursue my interests in life. For this, I am forever thankful, and this thesis is dedicated to them as a token of my gratitude.

Abstract

In this thesis, a closed-loop collision avoidance system (CAS) based on the scenario-based model predictive control (SBMPC) is developed and tested for autonomous surface vessels (ASVs) using the 1:20 scale model ReVolt. The CAS is of great importance to make robust, safe, and predictable decisions in compliance with the international regulations for preventing collisions at sea (COLREGS) for ASVs.

The CAS has the ability to use either an integrated probabilistic data association (IPDA) camera-LiDAR sensor fusion system, or an automatic identification system (AIS) tracking system based of the missing measurement Kalman filter (MMKF). Prediction of the own ship trajectory in the SBMPC has been improved by considering time-varying heading reference, a cross-track error dependent lookahead distance, and the ability to change course during the prediction horizon. Furthermore, an improved sensor setup on ReVolt has been employed by moving the LiDAR, and a camera-LiDAR calibration has been performed.

The collision avoidance (COLAV) method, with the AIS tracking system, has been implemented and tested in a variety of real-time simulations. The results show that the SBMPC CAS can avoid the obstacles in compliance with COLREGS in all performed scenarios. Furthermore, the use of change points improved the efficiency of the evasive maneuvers as well as making the maneuvers more predictable to other vessels.

Real-life experiments with ReVolt using IPDA LiDAR tracking was then performed to verify the behavior of the COLAV method. The results show promising real-time capabilities, successfully avoiding the obstacle in all scenarios. The evasive maneuvers suffered from slow response due to the low bandwidth of the maneuvering controllers. The tracking results were verified by ground truth data and showed consistent and accurate estimates. The verification of the camera-LiDAR sensor fusion system remains as future work.

Sammendrag

I denne oppgaven har et antikollisjons-system i lukke sløyfe basert på scenariobasert modellprediktive regulator (SBMPC) blitt utviklet og testet for autonome overflatefartøy ved å bruke 1:20 modellskalabåten ReVolt. Antikollisjons-system er vesentlig for å utføre robuste, trygge og forutsigbare avgjørelser som samsvarer med de internasjonale forskriftene for forebygging av kollisjon på sjøen (COLREGS).

Antikollisjonssystem kan velge mellom å bruke et integrert probabilistisk datassosiasjonsfilter (IPDA), kamera og LiDAR-sensorfusjonssystem, eller et automatisk identifikasjonssystem (AIS) trackingsystem basert på et modifisert versjon av Kalmanfilter. Prediksjonen av trajektoren til eget skip i SBMPCen har blitt forbedret ved å ta hensyn til tidsvarierende headingreferanser, en lookahead-distanse som er avhengig av cross-track-feil, samt muligheten til å endre kursreferanse underveis i prediksjonshorisonten. Dessuten har et nytt sensorsystem blitt satt opp på ReVolt i samarbeid med ansatte i DNV GL, og en kamera-LiDAR-kalibrering har blitt utført.

Antikollisjonsmetoden, sammen med AIS trackingsystemet, har blitt implementert og testet i en rekke sanntidssimuleringer. Resultatene viser at det SBMPC-baserte antikollisjonssystemet klarer å unngå dynamiske hindringer i samsvar med COLREGS i alle utførte scenarier. Videre viste det seg at å endre kursreferanse iløpet av prediksjonshorisonten effektiviserte unnamanøvrene i tillegg til å gjøre unnamanøvrene mer forutsigbare for andre fartøy.

Det ble utført sjøeksperiment med ReVolt, som brukte IPDA LiDAR-tracking, for å verifiserte oppførselen til antikollisjonsmetoden. Resultatene viser lovende santidsegenskaper, der hindringen ble unngått i alle utførte kollisjonsscenarier. Unnamanøvrene led av treg respons grunnet lav bånnbredde på manøvreringsregulatorene. Trackingresultatene ble verifisert av GNSS-data og viste konsistente og nøyaktige estimater. Verifikasjonen av kamera-LiDAR sensorfusjonssystemet gjenstår som videre arbeid.

Table of Contents

List of Abbreviations

AEE Avarage Eucledian Error

AIS Automatic Identification System

APF Arteficial Potential Field

ASV Autonomous Surface Vehicles

CAS Collision Avoidance System

COG Course Over Ground

COLAV Collision Avoidance

COLREGS International Regulations for Preventing Collisions at Sea

CPA Closest Point of Approach

CV Constant Velocity

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DNV GL Det Norske Veritas-Germanisher Lloyd

DOF Degrees of freedom

DW Dynamic Window

EKF Extended Kalman Filter

GNC Guidance Navigation and Control

GNSS Global Navigation Satellite System

IMM Interactive Multiple Model

IPDA Integrated Probabilistic Data Association

KF Kalman Filter

LiDAR Light Detection And Ranging

LOS Line Of Sight

MHT Multiple Hypothesis tracking

MMKF Missing Measurement Kalman Filter

MPC Model Predictive Control

MSE Mean Square Error

NED North East Down

PDAF Probabilistic Data Association Filter

RADAR RAdio Detection And Raning

RMSE Root Mean Square Error

ROS Robot Operating System

SBMPC Scenario Based Model Predictive Control

SNAME Society of Naval Architects and Marine Engineers

SOG Speed Over Ground

VO Velocity Obstacle

Chapter 1

Introduction

1.1 Motivation

Collisions at sea can cause fatal consequences for the humans involved and significant environmental and structural damages. The main contributor can be attributed to human error, which is responsible for $75\% - 96\%$ of maritime casualties. According to [7], lack of operational skills and situational awareness are the primary reasons for humans' accidents. This encourages the development of unmanned autonomous ships, effectively reducing human involvement in maritime operations. Additionally, autonomous shipping gives environmental and economic benefits, where a 2016 study [8] projected over seven millions dollars could be saved per autonomous vessel over 25 years.

One of the main challenges of employing autonomous surface vehicles is the collision avoidance system problem. Simply avoiding an obstacle is not sufficient, as the maneuvers in maritime navigation should abide by the COLREGS, agreed upon by the International Maritime Organization in 1972 [9]. This adds another level of complexity to the problem. Moreover, achieving situational awareness using sophisticated sensors in a target tracking system is another demanding challenge. These challenges need to be handled with the utmost respect as they are critical in the development of a fully operational closed-loop ASV.

1.1.1 The ReVolt project

The classification company DNV GL launched the ReVolt project in 2013 [10]; a shipping concept envisioned to be a $60\mathrm{m}$ long, zero-emission, unmanned autonomous cargo ship. The vessel will have a range of 100 nautical miles, operating at a speed of 6 knots. For testing purposes, a 1:20 scale model has been built, where students in collaboration with DNV GL employees have been working on the vessel since 2016. The model boat, from now on referred to as ReVolt, is equipped with a dynamical position system [11], guidance and path following [12], a stand-alone CAS [5], and a LiDAR-camera sensor fusion system [1, 4, 13]. This thesis aims to use the LiDAR-camera sensor fusion system in a closed-loop CAS.

1.2 Previous work

Substantial research has been done within the field sensor fusion and target tracking for autonomous vehicles, as well as obstacle avoidance for ASVs. This section will present some examples of different approaches to these topics.

1.2.1 Previous work on sensor fusion

A collision avoidance system needs a way to acquire information about its surroundings, both static and dynamic obstacles. Measurements from the sensor system alone are not sufficient - the measurements need to be handled in a tracking system to be associated with an obstacle such an estimate of the obstacle trajectory is obtained.

A popular tracking method is the probabilistic data association filter (PDAF) [14], which calculates the association probabilities for a set of validated measurements to a target. These probabilities are subsequently used to weigh the Kalman gain in a Kalman filter. However, a disadvantage with the PDAF is that it assumes that a target exists, meaning that the probability of track existence is not available.

The integrated probabilistic data association filter (IPDA) [15] solves this issue by calculating the existence probability for each track, based on the difference between the measurements and their predictions. Additionally, the IPDA is significantly more efficient in track confirmation [16] as the PDAF produces a great amount of false tracks compared to IPDA with the same detection probability.

Both PDAF and IPDA are single-target tracking algorithms than can be used for multi-target tracking, given that the obstacles are sufficiently separated. They can be extended to handle situations where occlusion occur in the multi-target tracking algorithms Joint PDAF[17] and Joint IPDA[18] respectively. These algorithms will not be discussed further, and the reader is referred to the given citations for detailed explanations.

The multi hypothesis tracking algorithm (MHT) is another multi-target tracking algorithm and provides a systematical solution to the data association problem by building a tree of potential hypotheses for each candidate target. Then, the combination of most likely tracks is selected.

Using multiple sensors with complementary abilities can be beneficial to extract more nuanced information about the nearby environment. Combining the measurements from the different sensors is called sensor fusion and can be described as combining incomplete and imperfect pieces of mutually complementary sensor information so that a better understanding of an underlying real-world phenomenon is achieved Koch [19]. Stiller et al. [20] proposed a multi-sensor concept using differential global positioning system (DGPS), stereo vision, radio detection and ranging (RADAR) and LiDAR for an autonomous unsupervised vehicle. The sensory data were combined into a common obstacle map, achieving high accuracy validated by real-time evaluation of the sensor data. The estimation and tracking were performed using a Kalman filter (see Section 4.1.1), using point position estimates from the vision sensor and laser scanner.

Examples of sensor fusion in maritime environments have also been shown. Elkins et al. [21] published a paper on the Autonomous Maritime Navigation project, where they utilized multiple sensors in a sensor fusion engine to create a world map, including X-band RADAR, LiDAR. Obstacle tracking was performed using MHT with integrated multiple models (IMM) to create an obstacle map to be used for obstacle avoidance. The development in the field of visual object detection and computer vision has made it possible to extract more information from camera data. A 2015 study [22] presented an implementation of a real-time sensor fusion system using RADAR and camera in order to detect high-speed obstacles at sea. The results showed a significant increase in performance using the vision system.

Furthermore, Helgesen [23] demonstrated a JIPDA based sensor fusion system using LiDAR, RADAR, infrared and electro-optical camera data to detect objects at sea. Each sensor's detection capability was evaluated individually and in fusion with other sensors. It was shown that a fusion of RADAR, LiDAR, and infrared camera data gave promising results, providing better results than if used separately.

1.2.2 Previous work on collision avoidance

Numerous reviews of collision avoidance systems for manned and unmanned ships have been given over the last two decades. Although the objectives for manned and unmanned ships are different, some scholars believe that the conducted research can be mutually beneficial for the two ship types [24]. In the review articles [25] and [26], different approaches from the perspective of supporting the human in collision avoidance (COLAV) are presented, while [27] and [28] presented approaches from the perspective of developing ASVs. A comprehensive literature review of the state-of-the-art collision avoidance methods for ships was presented in [29], where they compared the strengths and weaknesses of different collision prevention techniques based on three steps; motion prediction, conflict detection, and conflict resolution, which are the processes of determining evasive maneuvers for collision prevention [29]. Also, they identified potential synergies and mutual benefits between research on manned an unmanned ship and outline a possible path towards developing fully autonomous ships.

In general, the collision avoidance approaches can be divided into two groups: path-planning methods, whose objective is to find a collision-free path considering static obstacle, and reactive algorithms, and reactive COLAV methods, whose aim is to consider obstacles unknown in prior. The path-planning algorithms might not be applicable for real-time applications with limited processing power [30]. The further focus will, therefore, be on reactive collision avoidance methods. Even though collision avoidance for manned and unmanned ships share similarities, the respective methods differ in their objectives. While collision avoidance methods for manned ships focus more on alerting the navigator of potential collision risks, the methods for unmanned ships focus on finding a navigational solution avoiding the collision(s). Thus, the collision prevention methods presented in this thesis will be from the perceptive of developing ASVs.

A much-used reactive method is the artificial potential field (APF), which generates repulsive virtual force fields around an obstacle and attractive virtual force fields around the destination. The algorithm does not guarantee a collision-free path but generates control inputs to move the vehicle in the direction of motion specified by the virtual force field. This approach is not designed for a dynamic environment but was addressed in [31] by considering the velocity and the maximal deceleration of the obstacle. The disadvantage with APF methods is that they can trap the ASV in a local minimum and that the ship dynamics are assumed holonomic. Also, the solution is not guaranteed to be optimal.

A hybrid approach of the A* and the dynamic window (DW) method was demonstrated in [32]. The A* algorithm was used for path planning, and the DW predicted the movements of the vehicle, including both acceleration and lateral speeds. However, DW assumed that the static state is always a safe state, which might not hold in a real-life situation [29].

The Velocity obstacle (VO) algorithm [33] is a simple and popular algorithm that deals with moving obstacles. A set of velocities that will lead to a collision is calculated assuming constant speed, meaning that a velocity outside this set will be collision preventive. This method does not require any information about the vessel's dynamics, and promising results were made by both [34] and [35]. However, the assumption that the obstacle travels with constant speed, along with the assumed circular shape for obstacles, are some of the disadvantages with the algorithm.

According to Johansen et al. in [36], many of the COLAV methods generally struggle with handling scalability and managing multiple obstacles in a dense traffic situation while also taking the ship dynamics, steering and propulsion system, and environmental disturbances into account. This motivates the use optimization-based control, more specifically the scenario-based model predictive control (SBMPC)[36]. Trajectories of the dynamical ship model are predicted using a finite set of control behaviors, and the trajectories are evaluated in a cost function based on collision risk, hazard, and compliance with COLREGS and computational cost. The algorithm was extended and improved in [37] and [6] and showed promising results in real-life experiments. The latter used an IPDA RADAR tracking method, where the results showed that the IPDA was a suitable tracking method for collision avoidance purposes. These papers [36, 37, 6] are the origin of the collision avoidance system used in this thesis and will be described in detail in Chapter 5.

1.2.3 Previous work on ReVolt

This thesis is a continuation of previous work done on ReVolt. Kamvsåg [13] developed a multi-target tracking framework using the multi-target version of the IPDA namely the Joint IPDA, using camera and Light detection and ranging (LiDAR) as primary sensors. A modified version of the density-based spatial clustering of applications with noise algorithm (DBSCAN) was used to process the LiDAR measurements, resulting in satisfactory results. The object detection algorithm used on the camera-data was the Faster R-CNN, previously trained by Tangstad [38]. The results showed that the LiDAR performed satisfactorily within the range of 10-50 meters, while the camera struggled with ranges exceeding 20 meters. The camera data was not used to fuse the measurements in the tracking system.

In [4], the old camera was replaced by a Ladybug5+ camera with the ability to provide high-quality images with a 90% spherical view, a significant improvement from the previous installation. Moreover, Norbye developed a real-time sensor fusion pipeline as well as implementing PDAF and IPDA as single-target tracking methods. Object detection was performed using YOLOv3. The results showed that the IPDA performed unsatisfactory behavior due to slow initialization, while the PDAF performed better. The camera measurement model and sensor fusion were not implemented due to time synchronization issues.

A solution to that problem was presented by the author of this thesis in [1], where the Ladybug5+ was integrated as a secondary sensor in the existing tracking system. The single target tracking system was extended by using both LiDAR and camera measurements to track targets in real-time. A comparison of LiDAR only tracking and sensor fusion tracking was made with promising results, but to technical failures in the conducted experiments, the results could not be verified by ground truth data.

Contributions with regards to COLAV have also been implemented on ReVolt. Midjås implemented a COLREGS compliant collision avoidance system using SBMPC in conjunction with an anti-grounding system in [5]. The obstacles were detected using an automation identification system (AIS), and the collision avoidance system was tested in both simulator a real-life with promising results.

1.3 Contributions

The primary contribution in this thesis is to combine the sensor fusion tracking system [13, 4, 1] and the SBMPC COLAV system [5] in a closed-loop COLREGS compliant CAS. To the author's knowledge, this is the first time real life experiments have been demonstrated using a LiDAR-based tracking system in a closed-loop CAS. Moreover, the work performed in this thesis entails a set of secondary contributions and can be summarized as:

- Plan and rearrange a new sensor setup on ReVolt in collaboration with DNV GL employees.
- Performing a new LiDAR-camera calibration.
- Solving the initialization problem on the IPDA algorithm.
- Implementing a cross-track error dependent lookahead distance in the guidance system.

- Implementing a time-varying heading reference throughout the prediction horizon in the SBMPC.
- Implementing the ability to change course during the prediction horizon, influenced by Giorgio Kufoalor in [2].
- Implementing the ability to add noise to the simulated AIS messages, making the simulated scenarios more realistic.
- Developing a tracking-to-COLAV system pipeline, influenced by Erik Wilthil in [2].
- Results from real-time closed-loop CAS simulations with single and multiple obstacles, using several configurations in the SBMPC.
- Demonstrating the closed-loop CAS on ReVolt using the IPDA tracker to obtain obstacle information, successfully conducting COLREGS compliant maneuvers in real-time at slow speeds.

1.4 Outline

This thesis is divided into nine chapters, described below:

Chapter 2 presents the maneuvering models used to describe the motion of marine, along with the heading and speed controllers on ReVolt. Finally, the line-of-sight guidance system is presented.

Chapter 3 introduces the sensor system on ReVolt along with the results from the camera-LiDAR calibration.

Chapter 4 introduces the state estimation and tracking algorithms used for COLAV purposes. Additionally, the measurement models used in the tracking system are given.

Chapter 5 introduces the SBMPC CAS in detail and presents the new additions to the previous implementation.

Chapter 6 gives a brief explanation of software frameworks used on ReVolt and the simulation platform.

Chapter 7 presents the simulation study. The performance of the SBMPC is tested in numerous scenarios with varying complexity, and the results are discussed.

Chapter 8 presents the results from the conducted experimental sea trials. Finally, the results are discussed.

Chapter 9 gives a conclusion from the results of this thesis and suggests further work.

Chapter 2

Guidance, Navigation, and Control

This chapter presents the mathematical model that describes the equation of motion used for simulating collision avoidance scenarios. By having an accurate representation of the actual vessel behavior, one can replicate its behavior and use this to predict its future behavior. Additionally, a mathematical model of the vessel dynamics can be used to design maneuvering controllers.

First, the notation and the reference frames used in this chapter are given. Then, the 3-Degrees of freedom (DOF) maneuvering model is presented along with the constant velocity model (CV). Next, the maneuvering controllers on ReVolt used to control heading and surge speed are described. Finally, the guidance system is presented.

2.1 Notation

The notation in this chapter is the same as in [39], which utilizes the Society of Naval Architects and Marine Engineers (SNAME)[40]. The SNAME notation can be seen in table 2.1 and is used to describe forces, moments, velocities, positions and angles in 6-DOF.

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

Table 2.1: SNAME notation.

The reference frames used in this thesis are given as:

NED The north-east-down NED coordinate system $\{n\} = (x_n, y_n, z_n)$ is located with the origin o_n, where o_n is decided by a given latitude and longitude. The frame is defined as the tangent plane on the Earth's surface where the x-axis points towards the true North, y-axis towards East, and z-axis downwards. Vessels that operate at local areas where one can assume approximately constant latitude and longitude use this frame for navigational purposes and is referred to as flat Earth navigation. In this case, the frame is assumed inertial such that Newton's law applies.

BODY The body-fixed reference frame $\{b\} = (x_b, y_b, z_b)$ is fixed with origin o_b in the center of origin of ReVolt, meaning that the frame moves with the vessel. The x-axis points forward, y-axis starboard, and z-axis down, which makes them coincide with principal axes of inertia.

2.2 Maneuvering models

2.2.1 3-DOF model

The maneuvering model used to describe the motion of Revolt is a standard 3-DOF with motion components in surge, sway and yaw. Here, the SNAME notation seen is used to express the position vector $\boldsymbol{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T$ in $\{n\}$, and the velocity vector $\boldsymbol{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}^T$ in $\{b\}$. The 3-DOF model assumes motion only in the horizontal plane, neglecting the motion in heave, roll and pitch, resulting in $w = p = q = 0$.

By using the state vectors $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ the equation of motion can be represented as [39]

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.1a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{2.1b}$$

where $\mathbf{R}(\psi)$ is the rotation matrix transforming the velocities in $\{b\}$ to $\{n\}$, defined as

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

From Table 2.1 one can see that $\mathbf{R}(\psi)$ makes a rotation about the z axis.

$\mathbf{M}$ is the mass matrix, consisting of the rigid body mass matrix $\mathbf{M}_{RB}$ and the hydrodynamic added mass matrix $\mathbf{M}_A$. The latter represents the inertia added to the system when the accelerating vessel displaces the water surrounding the vessel while it moves through the water. The mass matrix $\mathbf{M}$ is given by $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$, where

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{2.3}$$

and

$$\mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \tag{2.4}$$

Here, m is the mass of the vessel, x_g the distance from the center of origin (CO) to the center of gravity (CG) in x-direction. I_z is the moment of inertia about the z-axis. $X_{\dot{u}}$ represent the force in x direction caused by an accelerating $\dot{u}$ in surge. This force can also be written as [41]

$$X_{\dot{u}} \triangleq \frac{\partial X}{\partial \dot{u}} \tag{2.5}$$

The matrix $\mathbf{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix cause by rotation of $\{b\}$ about $\{n\}$. The the matrix is given as $\mathbf{C} = \mathbf{C}_{RB} + \mathbf{C}_A$, where

$$\mathbf{C}_{RB} = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & mu & 0 \end{bmatrix} \tag{2.6}$$

and

$$\mathbf{C}_A = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v_r + Y_{\dot{r}} r \\ 0 & 0 & -X_{\dot{u}} u_r \\ -Y_{\dot{v}} v_r - Y_{\dot{r}} r & X_{\dot{u}} u_r & 0 \end{bmatrix} \tag{2.7}$$

For lower speed motion, that is speed below $2\mathrm{m/s}$, the non-linear damping effects can be neglected, resulting in linear damping only (see Figure 2.1). This can also be backed up by the simulations performed in [42] where linear damping was valid approximation for speeds below $7\mathrm{m/s}$. Since the maximum speed of ReVolt is about $1.5\mathrm{m/s}$ only linear damping is considered in this thesis, resulting in a damping matrix $\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D_L}(\boldsymbol{\nu})$ that is given by

$$\mathbf{D_L} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \tag{2.8}$$

which follows the same convention as (2.5).

Lastly, the generalized force vector $\boldsymbol{\tau}$ is given by

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \begin{bmatrix} F_X \\ F_Y \\ l_r F_Y \end{bmatrix}, \tag{2.9}$$

where F_x and F_y are the forces in the x-and y-direction respectively coming from the vessel's actuators. l_r is the moment arm the force is acting on.

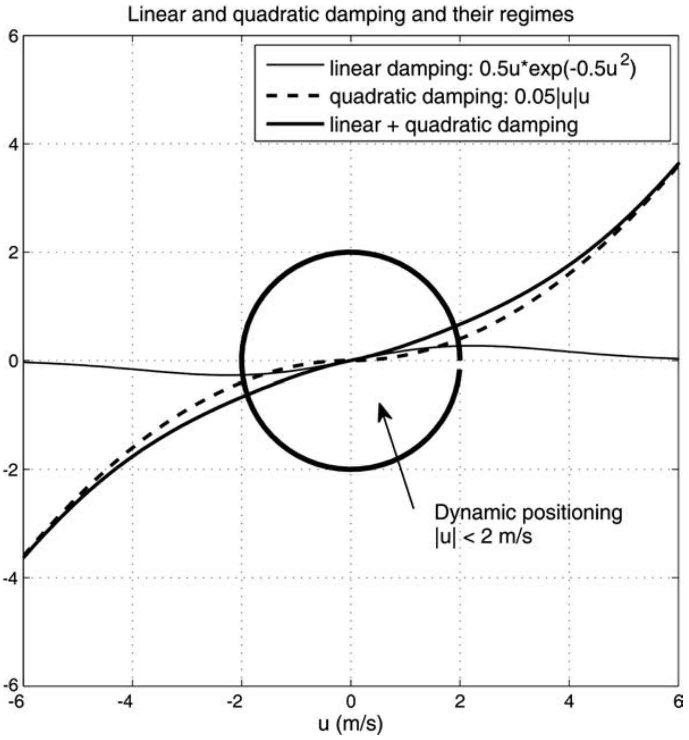


Figure 2.1: Speed regimes for linear and quadratic damping, found in [39, p. 138].

2.2.2 Constant Velocity model

Acquiring all the elements of the rigid-body kinematics of another vessel at sea is close to impossible. Thus, one cannot use (2.1) to describe the motion of the vessel. However, the estimated position and velocity of the vessel can be acquired using exteroceptive sensors and a tracking system (See Section 4.2). The constant velocity (CV) motion model is a motion model that uses only the position and velocity information to predict the future position of the object as a straight line. As the name suggests, the motion model assumes that the objects holds a constant velocity and consequently a constant heading. Let $\mathbf{x} = [N, V, V_N, E_N]$ be the state vector describing the position and velocity in North and East of an object. The CV model describes the motion of this object at time step $k+1$ as

$$\mathbf{x_{k+1}} = \mathbf{F_{cv}} \mathbf{x_k}, \tag{2.10}$$

where $\mathbf{F_{cv}}$ is the transition matrix given by

$$\mathbf{F_{cv}} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.11}$$

with T being the time step into the future.

2.3 Controllers

The controllers implemented on ReVolt were implemented and tuned by Havnegjerde [12] and is briefly repeated here for convenience. One important detail is that the heading controller was tuned for a cruise speed of $1\mathrm{m/s}$, meaning that it may not perform optimally for other surge speeds. The reader is referred to [12] for more details.

2.3.1 Speed Controller

The control objective of the speed controller is to minimize the surge speed error $\tilde{u}(t) = u_d(t) - u_f(t)$ where u_d is the time-varying desired surge speed and u_f is the low-pass filtered velocity measurement of ReVolt. The desired surge speed is given by the second order reference filter

$$\ddot{u}_d + 2\zeta\omega_n\dot{u}_d + \omega_n^2 u_d = \omega_n^2 u_{ref} \tag{2.12}$$

where u_{ref} is the reference surge speed, σ the damping ratio and ω_n the natural frequency. The reference filter's objective is to give a smooth reference surge speed to avoid large and unnecessary control efforts when the reference makes a step change.

The control law is given by

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_{m,FF} + \boldsymbol{\tau}_{m,FB} \tag{2.13}$$

where $\tau_{m,FF}$ is the feedforward term and $\tau_{m,FB}$ is the feedback term. The feedforward term is given by

$$\boldsymbol{\tau}_{m,FF} = M\dot{u}_d + \sigma(u_d) \tag{2.14}$$

where $M\dot{u}_d$ is the inertia term and $\sigma(u_d)$ is a function giving the steady-state damping ratio corresponding to the surge speed u_d. The feedback term is a PI-controller given by

$$\boldsymbol{\tau}_{m,FB} = K_p\tilde{u}(t) + K_i\int_0^t \tilde{u}(t)d\tau \tag{2.15}$$

where K_p is the proportional constant and K_i is the integral gain.

2.3.2 Heading controller

To make the heading controller on ReVolt, the 1.order Nomoto model was used, which is given by

$$\frac{r}{\delta}(s) = \frac{K}{1 + Ts} \tag{2.16}$$

and describe how the yaw rate, r, acts when a change in rudder angle (or thruster angle command δ) occurs. The elements K and T are the gain and time constants respectively.

The control objective of the heading controller is to minimize the heading and yaw error states, given by

$$\tilde{\psi} = \psi_d - \psi \tag{2.17}$$

$$\tilde{r} = r_d - r \tag{2.18}$$

where ψ_d and r_d are the desired yaw and yaw rate respectively. As for the speed controller, the desired values are supplied by a reference filter, and a 3.order reference filter is used to obtain the desired states ψ_d, r_d and $\dot{r}_d$. The reference model is given by

$$\frac{\psi_d}{\psi_{ref}}(s) = \frac{\omega_n^3}{(s+\omega_n)(s^2+2\zeta\omega_n s+\omega_n^2)} \tag{2.19}$$

where the elements are as previously described.

Further, the control law is formulated as

$$\boldsymbol{\tau}_\delta = \boldsymbol{\tau}_{\delta,FF} + \boldsymbol{\tau}_{\delta,FB}, \tag{2.20}$$

consisting of a feedback term ,$\boldsymbol{\tau}_{\delta,FF}$, and a feedforward term ,$\boldsymbol{\tau}_{\delta,FB}$. The feedforward term is given by

$$\boldsymbol{\tau}_{\delta,FF} = \frac{T}{K}(\dot{r}_d + \frac{1}{K}r_d) \tag{2.21}$$

where K and T are as described in (2.16). The feedback term is given by

$$\boldsymbol{\tau}_{\delta,FB} = -(K_p\tilde{\psi}(t) + K_i\int_0^t \tilde{\psi}(t)d\tau + K_d\tilde{r}) \tag{2.22}$$

and is a PID controller with the proportional, integral and derivative controller gains K_p,K_i and K_d respectively.

2.4 Line-Of-Sight guidance

Line-of-sight guidance (LOS) is a method frequently used for path following [39]. The objective of the method is to find a desired heading ψ_d for the heading autopilot, which forces the vessel to follow a predetermined path of waypoints. There are essentially two guidance approaches that can be used to find the desired course, namely lookahead-based and enclosure-based steering. In this case, the lookahead-based steering was used due to the fact that lookahead-based steering has multiple advantages over enclosure-based steering, e.g., computational complexity, and region of validity. A more thorough and detailed explanation is given in [39, Chapter 10.3].

In lookahead-based steering the desired course is defined to be

$$\chi_d(e) = \alpha_k + \chi_r(e), \tag{2.23}$$

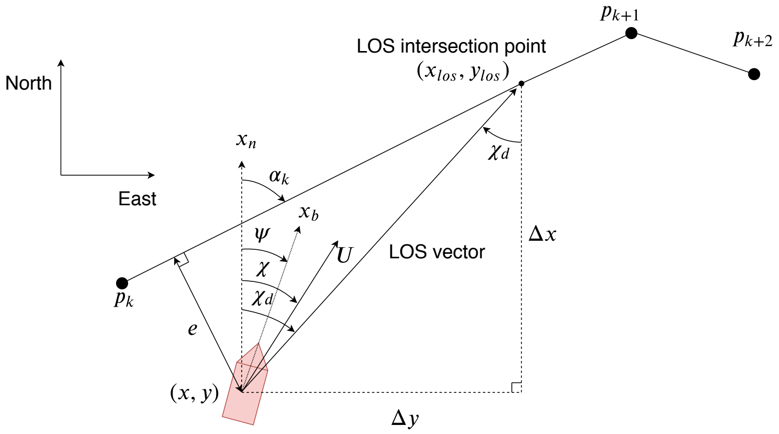


Figure 2.2: Simple illustration of the LOS guidance. The desired course χ_d points towards the LOS intersection point in the path.

where α_k is the path tangential angle and χ_r the velocity-path relative angle. The desired path consists of waypoints connected with a straight line between them where the waypoints in are coordinates given in the NED frame, seen in Figure 2.2. The waypoints are denoted as

$$\text{wpt} = \{p_1, p_2, ..., p_n\} \quad \text{where } p_i = [x_i, y_i]^T, i \in \{1, 2, \dots, n\}. \tag{2.24}$$

where n is the number of waypoints. Given two of these waypoints, p_k and p_{k+1}, one can find the path-tangential angle α_k by

$$\alpha := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k), \tag{2.25}$$

where k and $k + 1$ represents the current and next waypoint the vessel shall reach respectively. By using this angle one can find the cross-track error

$$e(t) = [x(t) - x_k]\cos(\alpha_k) + [y(t) - y_k]\sin(\alpha_k), \tag{2.26}$$

which is the shortest distance from the vessel to the desired path, as seen in Figure 2.2. The path following objective is fulfilled if the vessel is controlled such that the cross-track error converges to zero, as this implies that the vessel is on the desired path. Moreover, the velocity-path relative angle χ_r is defined as

$$\chi_r(e) = \arctan\left(\frac{-e}{\Delta}\right), \tag{2.27}$$

and is used to force the vessel towards a point on the desired path decided by the lookahead distance Δ. The lookahead distance used in this thesis is based on the [43], who proposed a solution where Δ was set small when the cross-track error was large, and vice versa. Mathematically, this is given by

$$\Delta(e) = (\Delta_{max} - \Delta_{min}) \exp(-\gamma|e|) + \Delta_{min}. \tag{2.28}$$

where Δ_{max} and Δ_{min} are maximum and minimum values for Δ, and γ the convergence rate towards the path. This is a new addition to the guidance system and results in a more aggressive angle towards the path when $e(t)$ is larger, and less aggressive when $e(t)$ is small. The values for the parameters are listed below

Δ_{min}	60
Δ_{max}	120
γ	1/300

Table 2.2: Parameters used in the time varying lookahead distance.

Chapter 3

Sensor system

The purpose of this chapter is to introduce the relevant sensors on ReVolt used in target tracking for COLAV purposes. ReVolt is equipped with a LiDAR and a camera as exteroceptive sensors, which is used in a sensor fusion system to extract information about targets of interest in the vessel's vicinity. Additionally, an artificial AIS is implemented in the Cybersea simulator (see Section 6.3) to obtain the ship-status of present obstacles. Due to the new sensor setup, a camera-LiDAR recalibration is needed to transform sensor measurements into a common reference frame.

In Section 3.1, the LiDAR on ReVolt is presented with necessary specifications. Then, the camera and object detection algorithm used to detect nearby boats are presented. Next, a brief introduction of the AIS is given. Then, the new sensor setup on ReVolt is presented along with the LiDAR-camera calibration and its results. Finally, the transformations that connect the sensor measurements to a common world frame are presented.

3.1 LiDAR

LiDAR is an active sensor that emits laser pulses at a given a time, and measures the parameters of the reflected pulse. The time of flight principle is then used to calculate the distance to the reflected point, given by

$$R = \frac{c}{2}(t_r - t_t). \tag{3.1}$$

Here, R is the distance, c the speed of light, t_r and t_t the reflection time and transmission time respectively. The laser pulse is transmitted with a given angle, namely the elevation ω and azimuth α, as shown in Figure 3.1. By using (3.2) one can convert the point to a Cartesian coordinated frame, which is the LiDAR reference frame. Using the same notation as in section 2.1, the LiDAR reference frame will from now on be referred to as $\{l\}$.

$$
\begin{aligned}
X &= R\cos(\omega)\sin(\alpha) \\
Y &= R\cos(\omega)\cos(\alpha) \\
Z &= R\sin(\omega).
\end{aligned}
\tag{3.2}
$$

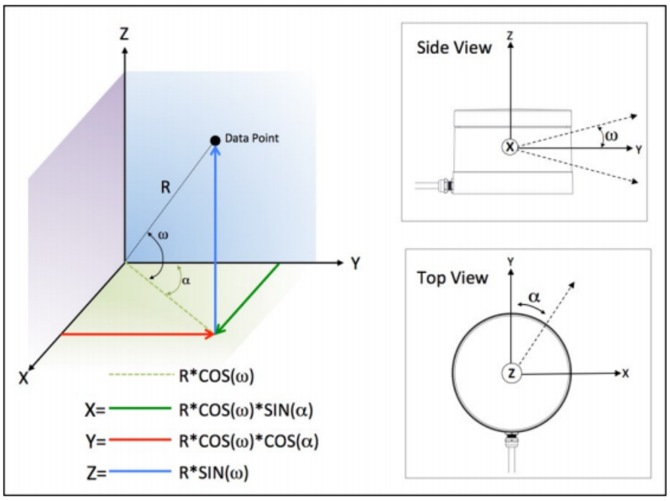


Figure 3.1: The coordinate system used by the LiDAR [44].

The LiDAR used on ReVolt is a Velodyne VLP-16 Puck, providing a full $360°$ horizontal view and a vertical field view of $\pm 15°$. It has a range of $100\mathrm{m}$ with accuracy of $3\mathrm{cm}$. The lasers are separated with a $2°$ angle vertically, meaning that the vertical distance between the LiDAR measurements are $d_v = R\sin(2°)$.

Reducing LiDAR points

A common assumption in target tracking algorithms is that a target produces at most one measurement per scan [3]. The LiDAR returns a point cloud containing thousands of points each second, where a potential target could be the source of a few hundred of them. To reduce the number of points that are passed on to the tracking system, a set of reduction methods are performed.

On the open sea, the reflected points will mostly originate from nearby targets, but as the vessel approaches land, there will be a drastic increase in reflected points originating from land. Therefore, land masking is performed to remove points originating from land using data from The Norwegian Mapping Authority (Kartverket). The map data is pre-processed and represented as a binary grid in $\{n\}$, such that if LiDAR points are within a cell that contains land, the points are discarded. The land masking algorithm was implemented on ReVolt in [4].

The remaining LiDAR points are still of a considerable amount, and to drastically reduce the number of points to be analyzed, clustering is performed using the DBSCAN-algorithm implemented by Kamsvåg in [13]. The reader is referred to

[13, 4] for a more in-depth explanation of both the clustering and land masking algorithm.

3.2 Camera

ReVolt is equipped with the FLIR Ladybug5+, providing high quality and accurate imaging in a close to 90% field of view of a full sphere using six cameras in different directions. The images are received individually by the Ladybug-driver, where each image is 2464×2048 with the image-quality set to 60%. A higher-quality would increase the data amount significantly, and the selected image quality was deemed sufficient. Only three of the cameras are used, namely, the cameras pointing towards the bow, port, and starboard, as they provide the necessary field of view to capture the nearby surroundings.

The camera model used to map a 3D-scene onto a 2D-image is the Pinhole camera model, where the intrinsic and extrinsic parameters were found by the calibration performed in [4]. The intrinsic parameters relate the camera coordinates to the image plane while the extrinsic parameters relate the camera coordinates to the real-world coordinate system. Using an algorithm capable of detecting objects in an image, one can, therefore, transform the detected object to a real-world coordinate system. Computer vision advancements and deep learning have produced a number of algorithms capable of performing visual object detection. The algorithm used on ReVolt is You Only Look Once version 3 (YOLOv3) [45] due to its real-time capabilities. YOLOv3 on ReVolt was implemented in [4] with close to 100 % detection of labeled boats in the test set.

As the main focus of this thesis is collision avoidance, the reader is referred to the author's specialization report [1] for more details concerning the Ladybug5+, pinhole camera model, and YOLOv3.

Ladybug5+ coordinate frame

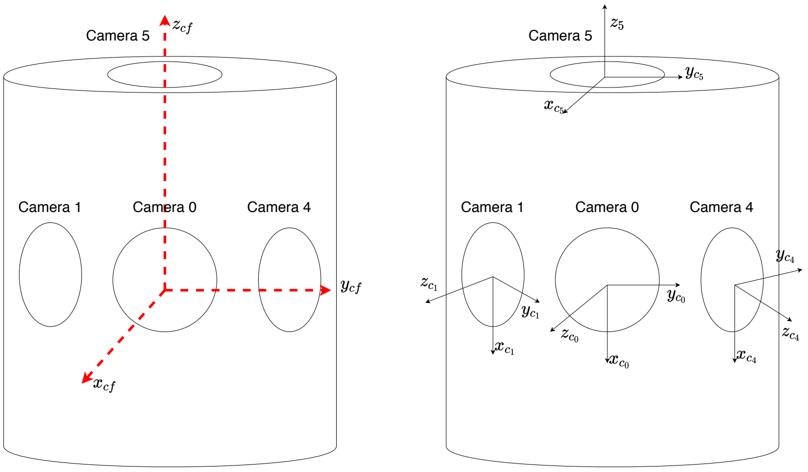
The Ladybug camera has its own coordinate system with origin in the center of the camera, each of lens have their own coordinate system, shown in Figure 3.2. Using the same notation as in section 2.1, the following reference frames are added:

Ladybug5+ main-frame The body-fixed reference frame $\{c_f\} = (x_{c_f}, y_{c_f}, z_{c_f})$ is located with origin o_{c_f} in the center of the camera structure. The x-axis points through the lens of camera 0, the z-axis through the lens of camera 5, and y-axis to complete the right hand rule.

Camera n-frame The body-fixed reference frame $\{c_n\} = (x_{c_n}, y_{c_n}, z_{c_n})$ is located with origin in the lens of camera n, where $n \in \{0, 1, 2, 3, 4, 5\}$. The z-axis points out of the lens, the x-axis points downwards, and y-axis to complete the right hand rule.

Perceived camera 0-frame The body-fixed reference frame $\{c\} = (x_c, y_c, z_c)$ is equal to $\{c_0\}$ but rotated $-90°$ about z_{c0}, resulting in the x-axis pointing to starboard side, and y-axis pointing downwards to complete the right hand rule.

The latter is also known as the standard camera frame and is used in the LiDAR-camera calibration, and is added now for convenience.



(a) The Ladybug5+ main-frame relative to the different cameras.

(b) The different camera frames on the Ladybug5+.

Figure 3.2: Simple illustration of the camera frames of the Ladybug5+.

3.3 Automatic Identification System

Automatic Identification System (AIS) is an automated tracking system used in the maritime world to exchange navigational information between vessels. The data is transmitted every $2 - 10$ seconds depending on the vessel's speed, and the data message consists of the the vessels size, position, position accuracy, speed over ground (SOG), course over ground(COG), and a unique identification number called MMSI, amongst other things [46]. The position accuracy of the AIS data is limited by the accuracy of the sensor system on the transmitting vessel.

ReVolt is not equipped with an AIS receiver now but is planned to be installed in the future. However, the simulator used as the simulation platform for ReVolt (see Section 6.3) is capable of simulating AIS messages.

3.4 Sensor setup

The sensor system on ReVolt is planned to be expanded by implementing RADAR and an AIS receiver in the future. Due to the small size of ReVolt, there were limited possibilities to mount the new sensors, and therefor a rearranging of the previous sensor setup, seen in Figure 3.3, was needed. Additionally, the previous setup was prone to lose heading as the metal structure above one of the GNSS receivers blocked many of the incoming measurements, resulting in only one functional receiver. The method used by the GNSS on ReVolt uses method 5 in [47] to find the heading, which requires two GNSS antennas with a sufficient known distance between them.

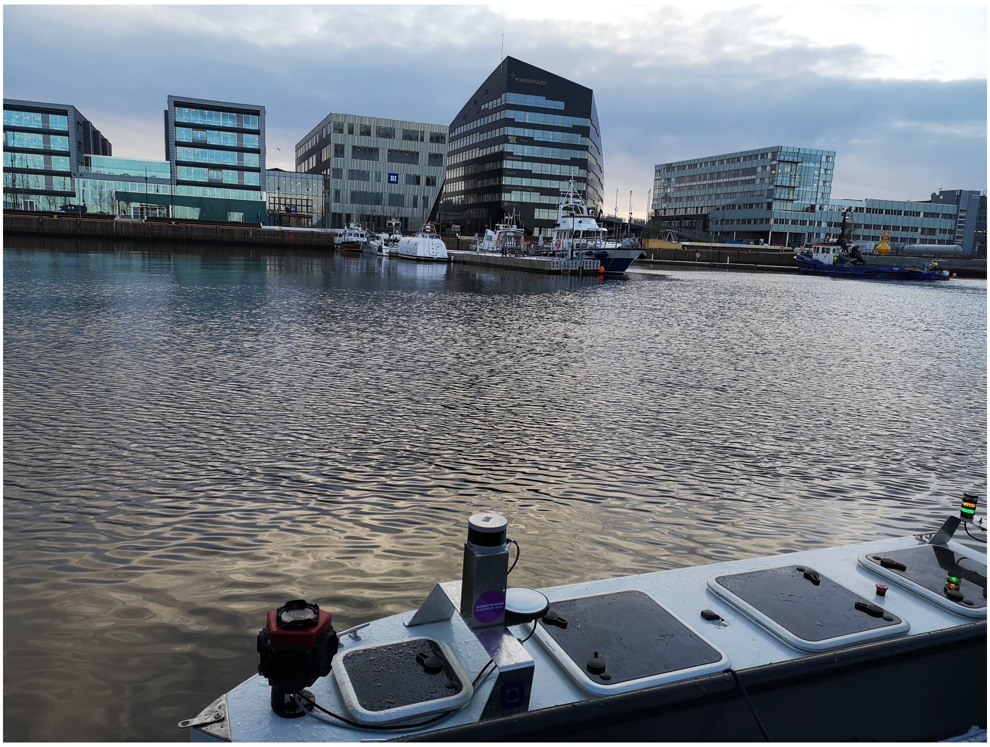


Figure 3.3: Old sensor setup where the LiDAR is mounted on top of a metal structure above the GNSS antenna, causing problems to find the heading of ReVolt.

The new setup can be seen in Figure 3.4 and was planned and installed in cooperation with DNV GL employees. The old structure was removed, and the LiDAR was placed on top of the Ladybug5+, while the GNSS antenna was moved to the center of the vessel. This resulted in a baseline of $1.51\mathrm{m}$. Where the previous setup struggled to receive heading measurements, the new setup achieved heading results almost momentarily after startup. This solution was a major bug fix on ReVolt as the heading-problem has been a prevalent problem.

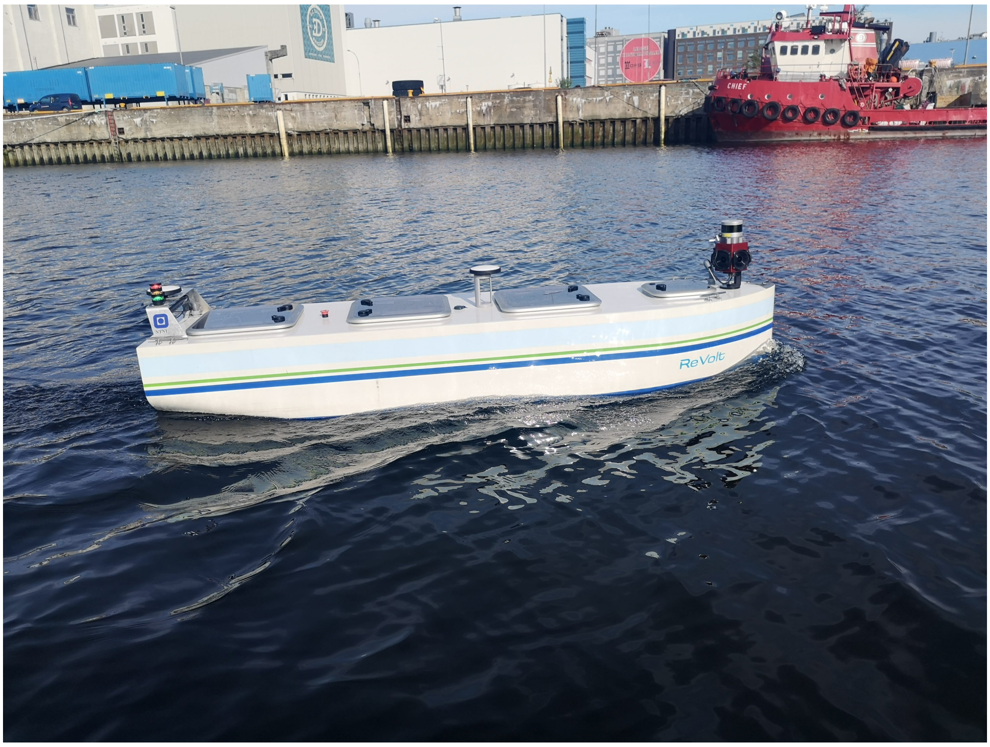


Figure 3.4: New sensor setup where the LiDAR is mounted on top the camera, and the GNSS antenna on a stand-alone structure in the middle of ReVolt.

3.5 Camera-LiDAR calibration

To fuse the LiDAR and camera measurements in the tracking system, the measurements must be transformed into a common world frame. This transformation depends on the transformation between the sensors. One solution could be to measure the placement of each sensor relative to $\{b\}$ manually. However, this might result in significant errors due to inaccurate measurements and is also time-consuming and cumbersome. Instead, a semi-automatic lidar-camera calibration method, see [48], was used to estimate the rigid-body transformation matrix between the two frames using 3D-3D correspondences between points in the $\{l\}$ and the perceived camera frame $\{c\}$.

3D point correspondences in the camera frame

The 3D point correspondences in the camera frame are found by using ArUco Markers, which are special encoded patterns that facilitate the detection and error correction of the tags themselves [49]. The markers are printed and attached to a cardboard (or any planar surface), seen in Figure 3.5, both with known dimensions. Given the marker's location, the corners of the cardboard can be calculated, resulting in a rotation and translation, $[\mathbf{R}|d]$, between the camera and the center of the marker. Thus, the corners can be transformed from the marker frame to $\{c\}$.

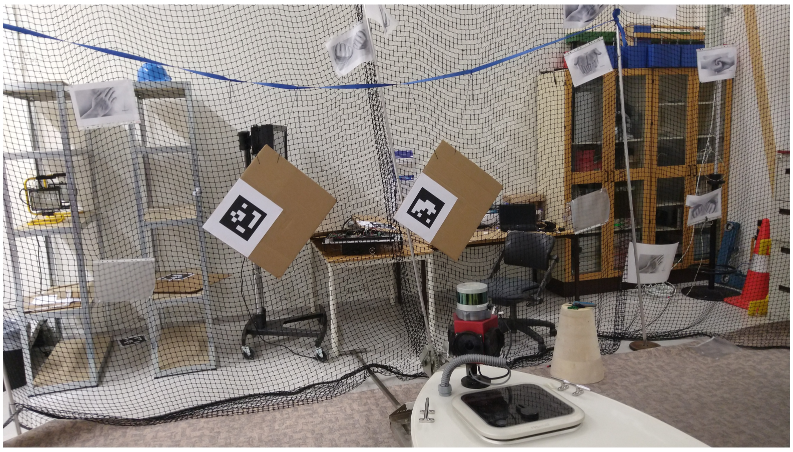


Figure 3.5: The calibration setup. The LiDAR and the camera may be seen at the bow of ReVolt.

3D point correspondences in the LiDAR frame

The 3D point correspondences in the LiDAR frame are found by detecting edges of the cardboard. For each edge of the cardboard, the user marks a polygon containing all the edge points, as illustrated in Figure 3.6. Then, Random Sample Consensus (RANSAC) is used to fit lines on the points in each of the four polygons. The corners are then found by using the intersection point of each of the line segments. However, if the intersection point does not exist, the midpoint of the shortest line segment between the two lines is approximated to be the corner point. This is done for both AruCo markers.

When the point correspondences for both camera and LiDAR are obtained, the rotation and translation between the two frames can be estimated using the Kabsch algorithm [50]. The derivation of this algorithm will not be explained further, and the reader is referred to [48] for a detailed explanation.

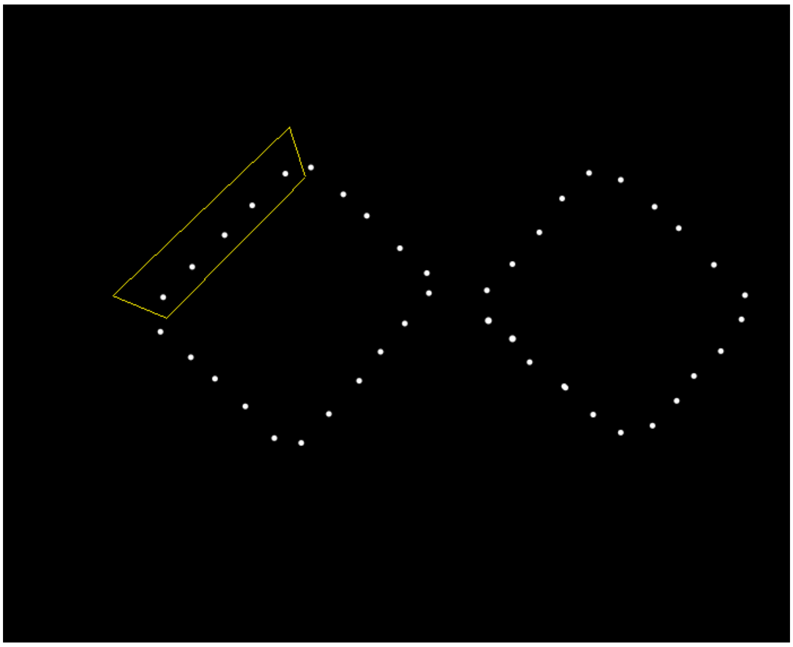


Figure 3.6: Marking line segments in the LiDAR 3D pointcloud.

Calibration results

The resulting average rotation and translation between the camera and LiDAR, $[\mathbf{R}_l^c|\mathbf{d}_{\mathbf{cl}}^{\mathbf{c}}]$ were estimated to be

$$\mathbf{R}_{\mathbf{l}}^{\mathbf{c}} = \begin{bmatrix} 0.999441 & 0.00632021 & 0.0328271 \\ -0.006991174 & 0.99815 & 0.0179375 \\ -0.0327077 & -0.0181543 & 0.9993 \end{bmatrix} \tag{3.3}$$

$$\mathbf{d}_{\mathbf{cl}}^{\mathbf{c}} = \begin{bmatrix} 0.0115954\,\mathrm{m} \\ -0.193193\,\mathrm{m} \\ -0.039949\,\mathrm{m} \end{bmatrix}. \tag{3.4}$$

where (3.3) consists of rotations in yaw, pitch and roll found to be

$$\psi = 0.489951\,\mathrm{rad} \tag{3.5}$$

$$\theta = -1.53268\,\mathrm{rad} \tag{3.6}$$

$$\phi = 1.07342\,\mathrm{rad}. \tag{3.7}$$

The average transformation matrix may then be constructed as

$$\mathbf{T}_l^c = \begin{bmatrix} \mathbf{R}_{\mathbf{l}}^{\mathbf{c}} & \mathbf{d}_{\mathbf{cl}}^{\mathbf{c}} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{3.8}$$

and the RMSE on the average transformation was $1.30748\mathrm{cm}$ according to the calibration method. This is a slight improvement of the calibration results from [4], which was performed with a similar setup at approximately the same distances.

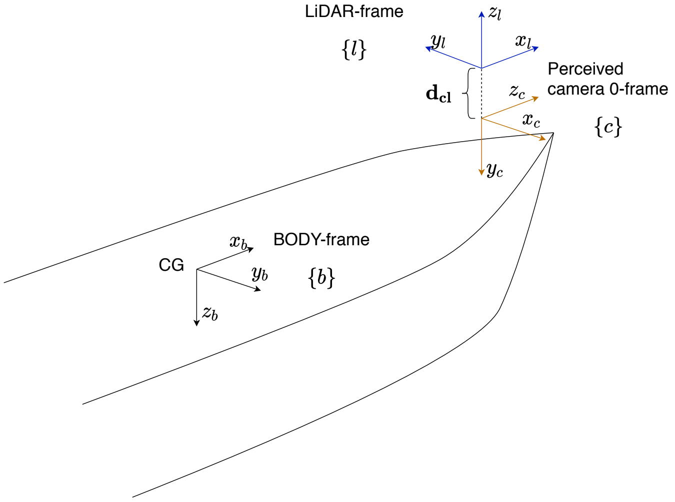


Figure 3.7: Overview of the reference frames on ReVolt, where $\mathbf{d_{cl}}$ is the distance from $\{\mathrm{c}\}$ to $\{\mathrm{l}\}$.

3.6 Transformation to a common world

In order to use sensor measurements in the tracking system they must be transformed from their respective frame to a common world frame, which in this case is $\{n\}$. For the camera n, a given measurement $\mathbf{z_{c_n}^{c_n}}$ is transformed from $\{c_n\}$ to $\{n\}$ by

$$\mathbf{T}_{c_n}^{n} = \mathbf{T}_{b}^{n}\mathbf{T}_{c_f}^{b}\mathbf{T}_{c_n}^{c_f}, \tag{3.9}$$

where $\mathbf{T}_{b}^{n}$, $\mathbf{T}_{c_f}^{b}$ and $\mathbf{T}_{c_n}^{c_f}$ are given in [4])

For the LiDAR one can transform the measurement, z_l, to $\{c_n\}$ and use (3.9) to transform the LiDAR measurement z_l to $\{n\}$. Recall from section 3.2 that $\{c_n\}$ differs from $\{c\}$, so by using (3.8) obtained from the calibration and the transformation matrix $\mathbf{T}_{c}^{c_n}$, one can form the transformation matrix

$$\mathbf{T}_{l}^{n} = \mathbf{T}_{c_n}^{n}\mathbf{T}_{c}^{c_n}\mathbf{T}_{l}^{c}. \tag{3.10}$$

The matrices (3.9) and (3.10) make it possible to map the LiDAR and camera measurements in the real world. By feeding these measurements to a tracking system, one can extract information about targets of interest in the nearby environment.

Chapter 4

Tracking and state estimation

Autonomous vehicles rely on measurements from multiple sensors to navigate safely in their environment. Each sensor has limited precision and uncertainty associated with each measurement. By combining various sensors, one can estimate the *state* of the vehicle, filtering out the uncertainties to the best of its ability. The state is a quantity that describes the vehicle's internal behavior, e.g., position, orientation, velocity, and acceleration. *Tracking* is the estimation of the motion of one or multiple moving objects using sensors that detect them. In any system where control theory is applied, state estimators are critical to be able to function properly.

First, the state estimators Kalman Filter and Extended Kalman filter will be discussed. Then, the two single-target tracking methods will be presented, namely the PDAF and IPDA. Finally, the measurement models for each of the sensors on Re-Volt are given. Section 4.1 and 4.2 are inspired by the the textbook that recently was introduced in the sensor fusion source at NTNU [16].

4.1 State estimation

This section will focus on the *state estimators*, or *filters* as they also are called. The two terms are often used interchangeably, even though there are some differences. In electronics, a filter is a device that removes unwanted characteristics on a signal, not necessarily on a dynamic system. A state estimator estimates the internal states of a system using sensory data and filters out unwanted noise, thus making it a filter. In other words, a state estimator is a filter since it removes unwanted noise to replicate the true state, but a filter is not necessarily a state estimator. Due to the uncertainty of the sensory data, a state estimator is essential for an autonomous vehicle to estimate both its own and surrounding objects' position. This section introduces the state estimators Kalman filter (KF) and its nonlinear extension extended Kalman filter (EKF). [16].

4.1.1 Kalman filter

The Kalman filter was first proposed in 1960 [51] by Robert Kalman and is a linear state estimator. If the following assumptions are satisfied, the Kalman filter is the optimal minimum mean square error estimator [52]:

- The dynamic system model is linear,
- the process and measurement noise are uncorrelated, white, Gaussian noise sequences with zero mean and known covariances,
- the system is observable,
- the initial state $\mathbf{x_0}$ is Gaussian.

A state space model that satisfies the assumptions above can be written on the form

$$\mathbf{x_k} = \mathbf{F}\mathbf{x_{k-1}} + \mathbf{v_k}, \qquad \mathbf{v_k} \sim \mathcal{N}(0, \mathbf{Q}) \tag{4.1a}$$

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} + \mathbf{w_k}, \qquad \mathbf{w_k} \sim \mathcal{N}(0, \mathbf{R}) \tag{4.1b}$$

$$\mathbf{x_0} \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P_0}), \tag{4.1c}$$

where (4.1a) describes the propagation of the state from time $k-1$ to k by applying the transition matrix $\mathbf{F}$ to the previous state $\mathbf{x_{k-1}}$, and (4.1b) describes how the measurement $\mathbf{z_k}$ relates to the state $\mathbf{x_k}$through the measurement matrix $\mathbf{H}$. The additive process and measurement noises are given by $\mathbf{v_k}$ and $\mathbf{w_k}$ respectively.

The Kalman filter is a two-step process consisting of a prediction step and a update step:

Predict

In the prediction step the KF predicts the state and covariance for the next time step, namely $\hat{\mathbf{x}}_\mathbf{k|k-1}$ and $\hat{\mathbf{P}}_\mathbf{k|k-1}$. They are estimated using the state estimate of the previous time step $\hat{x_{k-1}}$, the transition matrix $\mathbf{F}$ from (4.1) and the noise covariance matrix $\mathbf{Q}$, starting at the initial state $\mathbf{x_0}$ with initial state covariance $\mathbf{P_0}$. The prediction step can be summarized as

$$\hat{\mathbf{x}}_\mathbf{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_\mathbf{k-1} \qquad \text{Predicted state estimate} \tag{4.2a}$$

$$\hat{\mathbf{P}}_\mathbf{k|k-1} = \mathbf{F}\hat{\mathbf{P}}_\mathbf{k-1}\mathbf{F^T} + \mathbf{Q} \qquad \text{Predicted state covariance.} \tag{4.2b}$$

.

Update

In the update step the latest measurement is used to update the state, and can be summarized as

$$\hat{\mathbf{z}}_{\mathbf{k|k-1}} = \mathbf{H}\hat{\mathbf{x}}_{\mathbf{k|k-1}} \qquad \text{Predicted measurement} \qquad (4.3a)$$

$$\boldsymbol{\nu}_{\boldsymbol{k}} = \mathbf{z}_{\mathbf{k}} - \hat{\mathbf{z}}_{\mathbf{k|k-1}} \qquad \text{Innovation} \qquad (4.3b)$$

$$\hat{\mathbf{S}}_{\mathbf{k}} = \mathbf{H}\mathbf{P}_{\mathbf{k|k-1}}\mathbf{H}^{\mathbf{T}} + \mathbf{R} \qquad \text{Innovation covariance} \qquad (4.3c)$$

$$\mathbf{W}_{\mathbf{k}} = \mathbf{P}_{\mathbf{k|k-1}}\mathbf{H}^{\mathbf{T}}\mathbf{S}_{\mathbf{k}}^{-\mathbf{1}} \qquad \text{Kalman gain} \qquad (4.3d)$$

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k|k-1}} + \mathbf{W}_{\mathbf{k}}\boldsymbol{\nu}_{\mathbf{k}} \qquad \text{Posterior state estimate} \qquad (4.3e)$$

$$\mathbf{P}_{\mathbf{k}} = (\mathbf{I} - \mathbf{W}_{\mathbf{k}}\mathbf{H})\mathbf{P}_{\mathbf{k|k-1}} \qquad \text{Posterior covariance.} \qquad (4.3f)$$

. Since both process and measurement noise is present in the system, the state estimate cannot solely rely on either the measurement or the predicted state. Therefore, the effects of the measurement and the state estimate in (4.2a) is weighted by the Kalman gain (4.3d) to form the posterior state estimate in (4.3e). The Kalman gain is dependent on the innovation covariance (4.3c) and the predicted covariance (4.2b), and can be seen as a difference between having confidence in the predicted state versus confidence in the measurement. From (4.3d), one can see that when the predicted covariance is large, meaning that the predicted state is uncertain, the measurement will be weighted more in the posterior state estimate. The opposite occurs when the innovation covariance is large, meaning that the innovation is uncertain, resulting in that the predicted estimate will contribute more to the posterior state estimate. Lastly, the posterior state covariance (4.3f) is calculated, considering the blending of the predicted state estimate and the measurement.

4.1.2 Extended Kalman filter

One of the assumptions the aforementioned Kalman filter makes is that the system needs to be linear. However, more often than not this assumption is strictly impossible to hold [53] and the need for a non-linear state estimator arises. The Extended Kalman filter is a suboptimal state estimator for non-linear system that linearizes the non-linear dynamic and/or measurement equations about the latest state estimates [17]. As for the Kalman filter, the noises are assumed to enter additively. The EKF model can be written as

$$\mathbf{x}_{\mathbf{k}} = \mathbf{f}(\mathbf{x}_{\mathbf{k-1}}) + \mathbf{v}_{\mathbf{k}}, \qquad \mathbf{v}_{\mathbf{k}} \sim \mathcal{N}(0, \mathbf{Q}) \qquad (4.4a)$$

$$\mathbf{z}_{\mathbf{k}} = \mathbf{h}(\mathbf{x}_{\mathbf{k}}) + \mathbf{w}_{\mathbf{k}}, \qquad \mathbf{w}_{\mathbf{k}} \sim \mathcal{N}(0, \mathbf{R}) \qquad (4.4b)$$

where $\mathbf{f}$ and $\mathbf{h}$ are non-linear functions that represent the state transition model and measurement model functions respectively. As with the Kalman filter, $\mathbf{v}_{\mathbf{k}}$ and $\mathbf{w}_{\mathbf{k}}$ are the process and measurement noise respectively with associated covariance ($\mathbf{Q}$ and $\mathbf{R}$).

The algorithm is very similar to the Kalman filter, with the main difference being that the Jacobian of state transition and measurement equations are used in both the predict and update step to calculate the predicted state covariance, predicted measurement (and consequently the innovation) and the Kalman gain, as seen in (4.7) and (4.8). The Jacobians of the state transition matrix and the measurement matrix is given by

$$\mathbf{F_k} = \frac{\partial \mathbf{f_k}}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_\mathbf{k}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_j} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_i}{\partial x_1} & \frac{\partial f_i}{\partial x_2} & \cdots & \frac{\partial f_i}{\partial x_j} \end{bmatrix} \tag{4.5}$$

and

$$\mathbf{H_k} = \frac{\partial \mathbf{h_k}}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_\mathbf{k}} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_j} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_i}{\partial x_1} & \frac{\partial h_i}{\partial x_2} & \cdots & \frac{\partial h_i}{\partial x_j} \end{bmatrix} \tag{4.6}$$

respectively.

The prediction step of the EKF is quite similar to (4.2) with the addition of the transition Jacobian.

Predict

$$\hat{\mathbf{x}}_{\mathbf{k|k-1}} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k-1}}) \qquad \text{Predicted state estimate} \tag{4.7a}$$

$$\mathbf{F_k} = \frac{\partial \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k-1}})}{\partial \hat{\mathbf{x}}_{\mathbf{k-1}}} \qquad \text{Transition Jacobian} \tag{4.7b}$$

$$\hat{\mathbf{P}}_{\mathbf{k|k-1}} = \mathbf{F}\hat{\mathbf{P}}_{\mathbf{k-1}}\mathbf{F}^{\mathbf{T}} + \mathbf{Q} \qquad \text{Predicted state covariance} \tag{4.7c}$$

.

The same goes for the update step, where the measurement Jacobian is added, with the rest being identical to (4.3)

Update

$$\hat{\mathbf{z}}_{\mathbf{k|k-1}} = \mathbf{H}\hat{\mathbf{x}}_{\mathbf{k|k-1}} \qquad \text{Predicted measurement} \tag{4.8a}$$

$$\boldsymbol{\nu}_{\boldsymbol{k}} = \mathbf{z}_{\mathbf{k}} - \hat{\mathbf{z}}_{\mathbf{k|k-1}} \qquad \text{Innovation} \tag{4.8b}$$

$$\mathbf{H}_{\mathbf{k}} = \frac{\partial \mathbf{h}(\hat{\mathbf{x}}_{\mathbf{k-1}})}{\partial \hat{\mathbf{x}}_{\mathbf{k-1}}} \qquad \text{Measurement Jacobian} \tag{4.8c}$$

$$\hat{\mathbf{S}}_{\mathbf{k}} = \mathbf{H}\mathbf{P}_{\mathbf{k|k-1}}\mathbf{H}^{\mathbf{T}} + \mathbf{R} \qquad \text{Innovation covariance} \tag{4.8d}$$

$$\mathbf{W}_{\mathbf{k}} = \mathbf{P}_{\mathbf{k|k-1}}\mathbf{H}^{\mathbf{T}}\mathbf{S}_{\mathbf{k}}^{-\mathbf{1}} \qquad \text{Kalman gain} \tag{4.8e}$$

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k|k-1}} + \mathbf{W}_{\mathbf{k}}\boldsymbol{\nu}_{\mathbf{k}} \qquad \text{Posterior state estimate} \tag{4.8f}$$

$$\mathbf{P}_{\mathbf{k}} = (\mathbf{I} - \mathbf{W}_{\mathbf{k}}\mathbf{H})\mathbf{P}_{\mathbf{k|k-1}} \qquad \text{Posterior covariance} \tag{4.8g}$$

.

The use of the series expansion of the nonlinear functions $\mathbf{f}$ and $\mathbf{h}$ can cause the filter to diverge if the evaluation point of the Jacobian is poor. This usually occurs at the initial starting point as the EKF is sensitive to inaccurate initialization. In general, if the initial errors and noises are not too large, the EKF performs well [17].

4.2 Target tracking

Being able to detect, track, and estimate moving and stationary obstacles, or targets, in its vicinity, is critical for an autonomous vehicle to operate properly. When these estimates are acquired, they can be used in a collision-avoidance pipeline.

The Kalman filter assumes an ideal sensor where all measurements received originates from the target. However, in a real-world situation, this assumption does not necessarily hold. In other words, the problem of determining which measurement originating from which target arises and is known as the data association problem. Multiple methods exist to solve this problem, both for single and multiple targets. This section presents a high-level overview of two single-target tracking algorithms, its assumptions, and its requirements. These algorithms are PDAF and IPDA, where the latter is the tracking algorithm used during the conducted experiments. The section is based on chapter 7 and 8 in [16], [54] and [17]. A more detailed explanation is given there.

4.2.1 PDAF

In target tracking, several signals are received from the sensors, and the goal is to find out which measurements are target originated. A multidimensional gate, or region, is set up to avoid searching the entire measurement space for the measurement from the target of interest [54]. A measurement within this gate is a valid association candidate, and the gate is called a *validation region*. Any measurement outside this region is ignored, meaning that they are ”too far” away to be considered a likely target originated measurement.

As Figure 4.1 shows, multiple measurements can occur in the validation gate even though only one of them is target originated, leaving the remaining measurements clutter or false-alarm originated. Thus, the data association uncertainty arises. There are two choices when it comes to the data association:

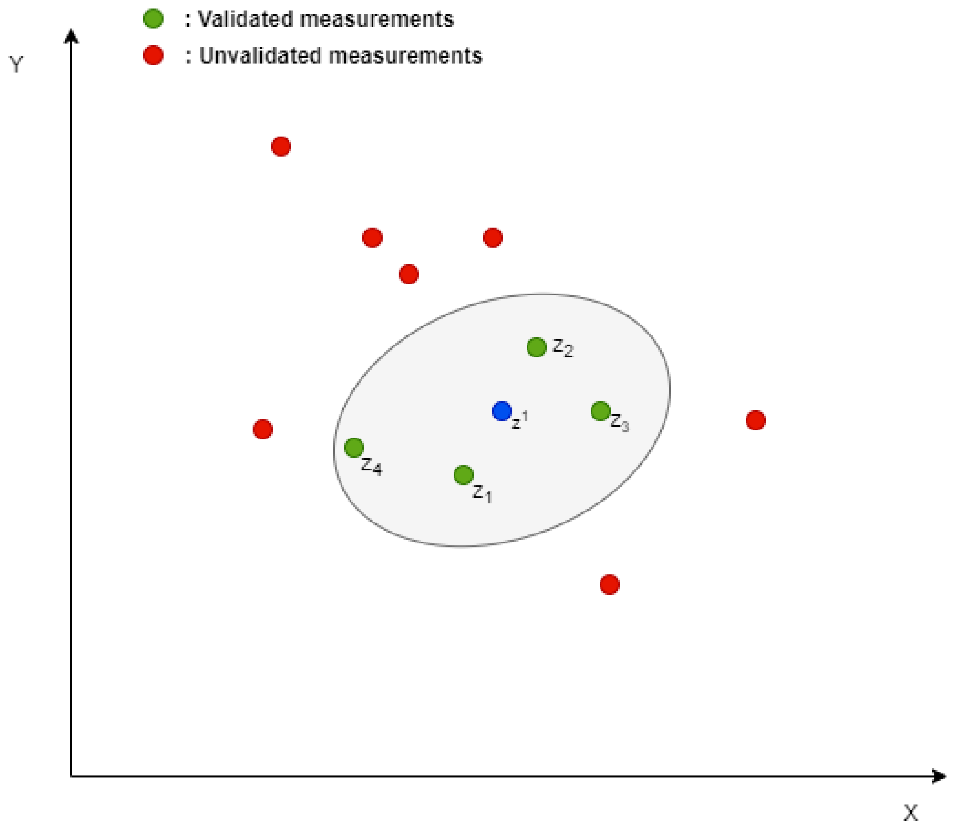


Figure 4.1: Validation gate with four validated measurements, given as green dots. The red dots are discared measurements.

1. Choose the closest measurement and ignore all other, or
2. find a weighted average of the relevant measurements.

The weights in this case represents the association probabilities to the target, that is, how likely it is that a measurement within the validation region is target originated. The PDAF algorithm uses the latter approach to obtain this probabilistic information, and the algorithm is based on either the KF or EKF depending on whether the state and measurement equations are linear or non-linear. By using a set of assumptions a state estimation scheme almost as ”simple” as the KF can be made. These assumptions are [54]:

A.1 There is only one target of interest, which is modeled by a Markov process.

A.2 The track has been initialized.

A.3 The past information at time $k-1$ is approximated by

$$\mathbf{p}(\mathbf{x_{k-1}}|\mathbf{z_{k-1}}) \sim \mathcal{N}(\mathbf{x_{k-1}}; \hat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{P_{k-1}}). \tag{4.9}$$

A.4 A measurement of the target is detected with probability P_D.

A.5 A measurement validation region is set up at each time around the predicted measurement, described in (4.10).

A.6 At most one measurement in the validation region can be originated from the target.

A.7 The remaining measurements are assumed to be from clutter or false alarm.

It should be noted that even though **A.1** states that there is only one target of interest, the PDAF (and consequently IPDA) can be used to detect multiple targets, given the targets are sufficiently separated. However, if two targets are too close to each other, measurements from one of them can be associated with the other, and one of the tracks may be lost. This problem is handled by JPDA (and JIPDA) but is not the scope for this thesis.

One cycle of the PDAF algorithm is shown in Figure 4.2 and is pretty similar to the state estimators (KF and EKF). In addition to the predict and update step, the PDAF adds the *measurement validation* and the *data association* to its algorithm.

Prediction

The PDAF starts in a similar manner as the KF and uses the (4.2a), (4.2b) and (4.3a) to predict the state, state covariance and the measurement respectively. In addition the innovation covariance is calculated as in (4.3c).

Measurement Validation

The next step in the process is measurement validation, where the measurement is validated if it is within the validation gate given by

$$\mathcal{V} = \{\mathbf{z} : \text{such that } \boldsymbol{\nu}_{\boldsymbol{k}}^{\boldsymbol{T}} \mathbf{S_k}^{-1} \boldsymbol{\nu}_{\boldsymbol{k}} < \gamma\}. \tag{4.10}$$

Here,γ is the gate threshold corresponding to the *gate probability* P_G, which is the probability that the gate contains the target-originated measurement. Normally, this is set high [54]. The gate threshold γ is obtained from the χ^2 tables as the quadratic form in (4.10) defines the region as χ^2 distributed with degrees of freedom being the same as the dimension of the measurement vector [17]. $\boldsymbol{\nu}_{\boldsymbol{k}}$ and $\mathbf{S_k}$ are the innovation and innovation covariance respectively, as described in (4.3). Since the innovation is assumed to be Gaussian, the validation region takes an elliptical shape.

The volume of the validation gate in (4.10) is

$$V_k = c_{n_z}|\gamma \mathbf{S_k}|^{\frac{1}{2}}, \tag{4.11}$$

where c_{n_z} is the volume of the hyperspace with a dimension equal to the dimension of the measurement space n_z. The validation region is located around the predicted measurement (4.3a) and the set of all validated measurements is

$$\mathbf{z}_k = \{z_i^k\}_{i=1}^{m_k}, \tag{4.12}$$

where m_k is the amount of measurements inside the validation region.

Data association

The association probability describe how likely it is that a measurement $\mathbf{z^i}$ is target originated. Let the association hypothesis be defined as

$$\theta_i^k = \begin{cases} z_i^k \text{ is the target oriented measurement,} & i = 1, 2, ..., m_k \\ \text{none of the measurements are target originated,} & i = 0 \end{cases} \tag{4.13}$$

with the events being mutually exclusive for $m_k \geq 1$ [17]. The association probabilities can then be written as

$$\beta_i^k = P(\theta_i^k | \mathbf{Z}^k). \tag{4.14}$$

The form of (4.14) depends on the probability mass function $\mu_F(m)$ of the number clutter measurements and can be split into two clutter models [17]:

1. Poisson model with clutter density λ, given as

$$\mu_m = e^{-\lambda V} \frac{(\lambda V)^m}{m!}. \tag{4.15}$$

2. Diffuse prior model gives as

$$\mu_m = \mu_{m-1} = \text{const.} \tag{4.16}$$

The association probability (4.14) using the Poisson clutter model can then be expanded to be

$$\beta_i^k = \begin{cases} \frac{\mathcal{L}_i^k}{1-P_D P_G + \sum_{i=1}^{m_k} \mathcal{L}_j^k}, & i = 1, 2, ..., m_k \\ \frac{1-P_D P_G}{1-P_D P_G + \sum_{j=1}^{m_k} \mathcal{L}_j^k}, & i = 0 \end{cases}, \tag{4.17}$$

where the likelihood ratio of the measurement z_i^k is target originated is given by [54]

$$\mathcal{L}_i^k := \frac{\mathcal{N}(z_i^k; \hat{z}_{k|k-1}, \mathbf{S_k})}{\lambda}. \tag{4.18}$$

If the clutter density, λ, is substituted with m_k/V_k, where V_k is the volume of the validation region from (4.10), the nonparametric PDAF is obtained. This means that the number of clutter measurements are equally likely, and since the amount of clutter measurements can be either m_k or $m_k - 1$ using **A.1**, the substitution makes sense.

State Estimation

The posterior state estimation is almost identical to the posterior state estimation in the KF, and is given by

$$\mathbf{\hat{x}_k} = \mathbf{\hat{x}_{k|k-1}} + \mathbf{W_k}\nu_\mathbf{k}, \tag{4.19}$$

where $\mathbf{W_k}$ is the Kalman gain from (4.3d). The difference between (4.19) and (4.3e) is the innovation term. Here, $\nu_\mathbf{k}$ is the combined innovation

$$\boldsymbol{\nu_k} = \sum_{i=1}^{m_k} \beta_k^i \boldsymbol{\nu_k^i} \tag{4.20}$$

where the higher association probabilities will contribute more with their innovations.

The covariance associated with the state update is

$$\mathbf{P_{k|k}} = \beta_k^0 \mathbf{P_{k|k-1}} + (1 - \beta_k^0)\mathbf{P^c} + \mathbf{\tilde{P}_k} \tag{4.21}$$

where

$$\mathbf{P^c} = \mathbf{P_{k|k-1}} - \mathbf{W_k S_k W_k^T} \tag{4.22}$$

is the covariance of the state updated with the correct measurement [54]. The term β_k^0 represents the probability that none of the measurements is correct, meaning that $1 - \beta_k^0$ gives the probability of the correct measurement being available. The spread of the innovation terms is given by

$$\mathbf{\tilde{P}_k} = \mathbf{W_k}\Big(\sum_{\mathbf{i=1}}^{\mathbf{m_k}} \beta_\mathbf{k}^\mathbf{i} \boldsymbol{\nu}_\mathbf{k}^\mathbf{i} {\boldsymbol{\nu}_\mathbf{k}^\mathbf{i}}^\mathbf{T} - \boldsymbol{\nu}_\mathbf{k}\boldsymbol{\nu}_\mathbf{k}\Big)\mathbf{W_k^T} \tag{4.23}$$

This term is positive semi-definite and is added to increase the posterior state covariance due to the measurement origin uncertainty [54].

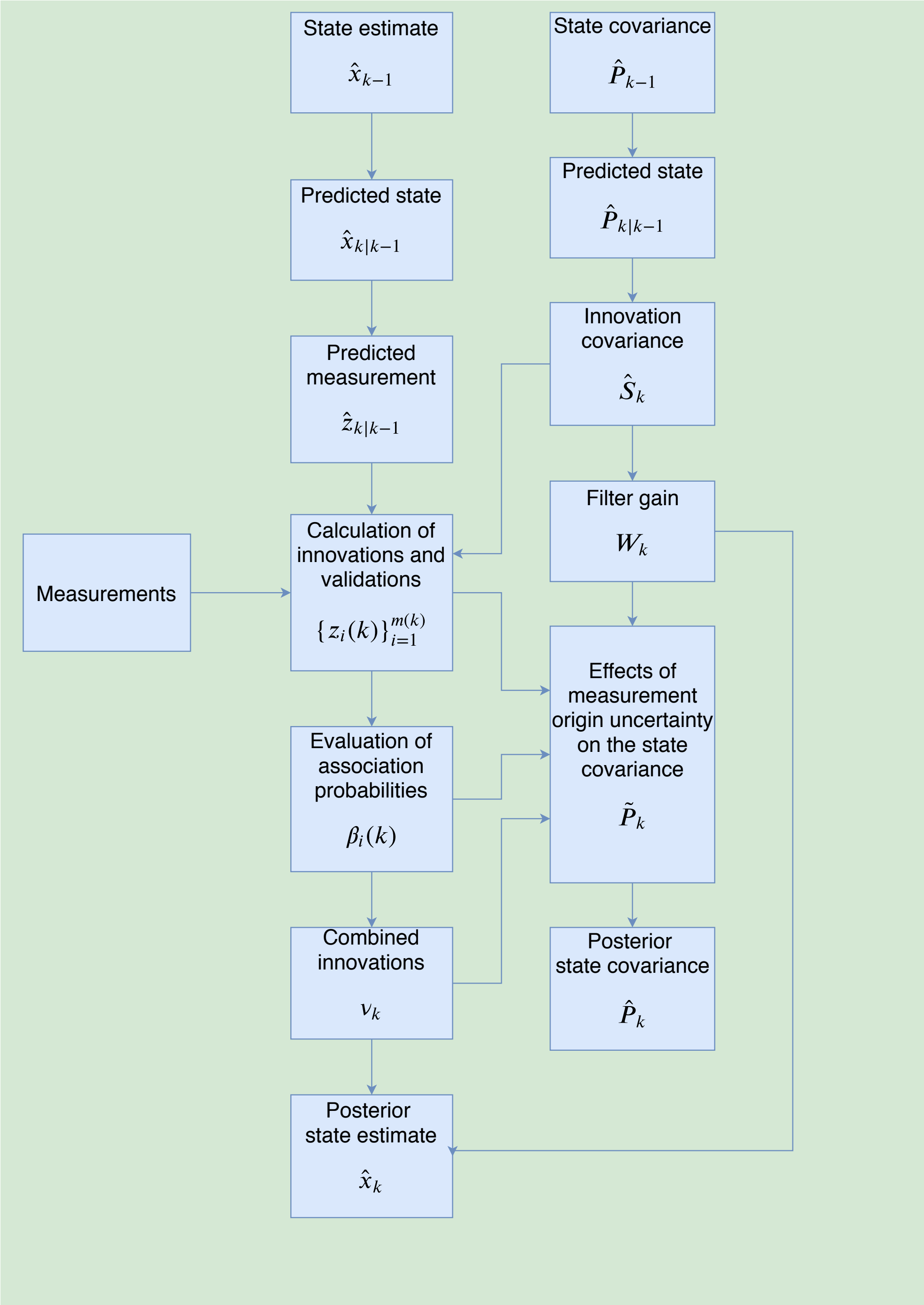

State estimate $\hat{x}_{k-1}$
State covariance $\hat{P}_{k-1}$
Predicted state $\hat{x}_{k|k-1}$
Predicted state $\hat{P}_{k|k-1}$
Predicted measurement $\hat{z}_{k|k-1}$
Innovation covariance $\hat{S}_k$
Filter gain W_k
Measurements
Calculation of innovations and validations $\{z_i(k)\}_{i=1}^{m(k)}$
Effects of measurement origin uncertainty on the state covariance $\tilde{P}_k$
Evaluation of association probabilities $\beta_i(k)$
Combined innovations ν_k
Posterior state covariance $\hat{P}_k$
Posterior state estimate $\hat{x}_k$

Figure 4.2: One cycle of the PDAF algorithm adapted from [54].

4.2.2 IPDA

The Integrated Probabilistic Data Association filter [15] is an extension of the PDAF that assumes that the probability of target existence can be calculate at each time instance. The PDAF assumption **A.1** is therefore altered to that the target exists with probability ϵ_k at time k, and does not exists with probability $1 - \epsilon_k$. Moreover, if the target exists at time k the target continues to exists at time $k + 1$ with a probability of P_S, and seizes to exist with probability $1 - P_S$. This infers that a target that dies cannot be reborn again. The remaining assumptions of the PDAF continues to hold for the IPDA, conditioned on target existence.

The tracking process of the IPDA algorithm is very similar to the PDAF, but the calculation of the target existence is added as a part of the process. This is done in the same two-step process, namely the prediction en update process. First, the existence probability is predicted. Then the state prediction, calculation of association weights, and the state update are performed in the same manner as in the PDAF. Finally, the posterior existence probability is calculated. For simplicity, only the first and last step of this process will be presented in this section.

Existence probability

The existence probability prediction is designed to be a Markov chain and is given

$$\begin{bmatrix} \epsilon_{k|k-1} \\ 1 - \epsilon_{k|k-1} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \epsilon_{k-1} \\ 1 - \epsilon_{k-1} \end{bmatrix} \tag{4.24}$$

where ϵ_{k-1} is the posterior existence probability at time $k - 1$. The coeffisients of the 2×2 transition matrix satisfies $p_{11} + p_{12} = p_{21} + p_{22} = 1$.

Existence update

Using

$$\delta_k = P_D P_G \Big(1 + \frac{1}{\lambda} \sum_{i=1}^{m_k} p(z_i^k | \mathbf{Z^{k-1}})\Big) \tag{4.25}$$

one can form the posterior existence probability for Markov chain one as

$$\epsilon_k = \frac{1 - \delta_k}{1 - \delta_k \epsilon_{k|k-1}} \epsilon_{k|k-1}. \tag{4.26}$$

where $p(z_i^k | \mathbf{Z^{k-1}}) = \mathcal{N}(z_i^k; \hat{z}_{k|k-1}, \mathbf{S_k})$ is the likelihood of the measurement z_i^k being target originated. From this, the association probabilities can be expressed as [15]

$$\beta_i^l = \begin{cases} \frac{P_D P_G p(z_i^k | \mathbf{Z^{k-1}})}{\lambda(1-\delta_k)}, & i = 1, ..., m_k \\ \frac{1 - P_D P_G}{1-\delta_k}, & i = 0. \end{cases} \tag{4.27}$$

4.3 Measurement models

4.3.1 LiDAR Measurement Model

The raw measurement from the LiDAR is given in spherical coordinates and converted into Cartesian coordinates using (3.2), given in the LiDAR frame {l}. The state vector $\mathbf{x} = \begin{bmatrix} N & E & V_N & V_E \end{bmatrix}^T$ is given in $\{n\}$, and by using the transformation $\mathbf{T}_l^n$, given by (3.10), the NED frame is obtained from the LiDAR measurements. The measurement model is then given by

$$\mathbf{z_{k,l}} = \mathbf{H_l x_{k,l}} + \mathbf{w_{k,l}} \quad p(\mathbf{w_{k,l}}) \sim \mathcal{N}(\mathbf{w_{k,l}}; 0, \mathbf{R_l}), \tag{4.28}$$

where $w_{k,l}$ is the measurement noise and H_l the measurement matrix given by

$$\mathbf{H_l} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.29}$$

Note that the subscripts k, l denotes the time step k in {l}. Moreover, the measurement noise covariance matrix $\mathbf{R_l}$ is given by

$$\mathbf{R_l} = \sigma_l^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{4.30}$$

where $\sigma_l^2 = 0.5$. This can be justified by the results found in [4] where the LiDAR centroids had a mean square error (MSE)of $0.39\mathrm{m}^2$ at $50\mathrm{m}$. As the MSE and σ_l^2 are closely related it is reasonable to believe that σ_l^2 is in the same order of magnitude.

4.3.2 Camera measurement model

As the camera image is in 2D, extracting 3D coordinates is difficult without knowing additional information, as illustrated in Figure 4.3. A possibility is to assume that all relevant detections are at sea level, meaning that $z = 0$, and estimating a 3D position with sufficient camera elevation [23]. However, due to the small size of ReVolt, the camera elevation is not sufficient for such a georeferenced approach.

Instead, the ReVolt sensor fusion system processes camera data through a bearing-only measurement model. YOLOv3 outputs a set of bounding boxes of detected objects in the pictures, where each bounding box has an associated confidence score. The center of these bounding boxes is projected to the image plane of camera n, resulting in the line-of-sight angle ψ_{c_n}, which describes the direction of where the target is located in $\{c_n\}$.

Let then the predicted position of the target in $\{n\}$ and $\{c_n\}$ be denoted as

$$\mathbf{\bar{x}_t} = [x_t, y_t, z_t]^T \tag{4.31}$$

$$\mathbf{\bar{x}_{c_n}} = [x_{c_n}, y_{c_n}, z_{c_n}]^T \tag{4.32}$$

respectively. The line-of-sight angle ψ_{c_n} relative to the z_{c_n}-axis is given by

$$\psi_{c_n} = \arctan\left(\frac{-y_{c_n}}{z_{c_n}}\right), \tag{4.33}$$

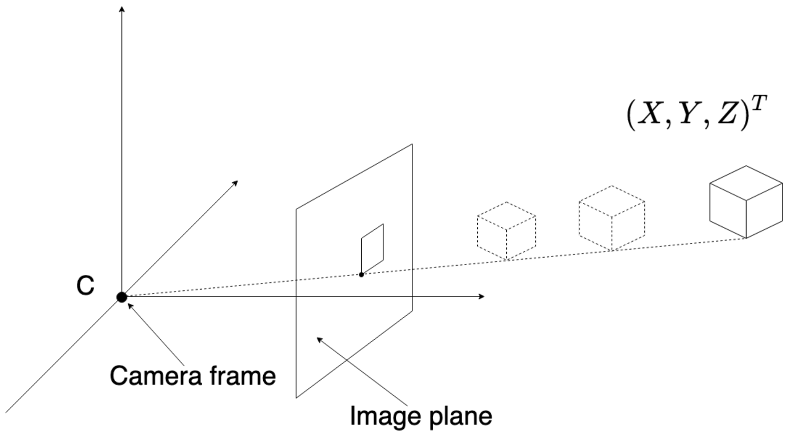


Figure 4.3: Illustration of how a box can be projected on to the image plane without knowing the world coordinate. From the image plane's perspective, all three boxes are the same, even though the cube with solid lines is the correct one.

as $\{c_n\}$ is rotated -90° about the z-axis relative to $\{c\}$. In terms of $\mathbf{\bar{x}_t}$, the line of sight angle can be expressed as a nonlinear function $\mathbf{h_c}(\mathbf{x_k})$ using the inverse of (3.9), given by

$$\mathbf{T^{c_n}_n} = \begin{bmatrix} \mathbf{R^{c_n}_n} & \mathbf{d^{c_n}_{c_n,n}} \\ \mathbf{0_{1\times3}} & 1 \end{bmatrix}, \tag{4.34}$$

resulting in

$$\psi_{c_n} = \arctan\left(\frac{r_{21}x_t + r_{22}y_t + d_2}{r_{31}x_t + r_{32}y_t + d_3}\right) = h_c(x_k) \tag{4.35}$$

where r_{ij} is the ij^{th} entry of the rotation matrix $\mathbf{R^c_n}$, d_i is the i^{th} entry of $d^{c_n}_{c_n,n}$. The nonlinear function (4.35) describes the relationship between the camera measurement and the target state, and can be used to form the nonlinear measurement model

$$z_{k,c} = h_c(x_k) + w_{k,c} \quad p(w_{k,c}) \sim \mathcal{N}(w_{k,c}; 0, \sigma_c^2). \tag{4.36}$$

Here, $\sigma_c^2 = 0.1 \approx \frac{6\pi}{180}$ and corresponds to a measurement uncertainty of about 6° and was found to through trial and error in the author's project thesis [1].

Furthermore, the Jacobian of (4.35) evaluated at the target state needs to be calculated in order to update the innovation covariance update (4.8), and is given by

$$\mathbf{H_k} = \begin{bmatrix} \frac{\partial h}{\partial x_t} & \frac{\partial h}{\partial y_t} & \frac{\partial h}{\partial \dot{x}_t} & \frac{\partial h}{\partial \dot{y}_t} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{\bar{x}_t}}. \tag{4.37}$$

Now, let the denominator and numerator of the $\arctan$-function in (4.35) be denoted as $a = r_{21}x_t + r_{22}y_t + d_2$ and $b = r_{31}x_t + r_{32}y_t + d_3$ respectively. Then the elements in (4.37) can written as

$$\frac{\partial h}{\partial x_t} = \frac{r_{21}b - r_{31}a}{a^2 + b^2} \tag{4.38}$$

$$\frac{\partial h}{\partial y_t} = \frac{r_{22}b - r_{32}a}{a^2 + b^2} \tag{4.39}$$

$$\frac{\partial h}{\partial \dot{x}_t} = \frac{\partial h}{\partial \dot{y}_t} = 0. \tag{4.40}$$

4.3.3 AIS measurement model

Both the LiDAR and camera produce measurements where the origin of the measurement is unknown. This is not a problem for the AIS, as the transmitted AIS message contains a unique identification number called MMSI. Thus, only a standard Kalman filter is required to track the corresponding ship. The AIS measurement model used in the Kalman filter is based on the work done in [3].

The AIS messages also give position data in the form of latitude and longitude and can be converted to the local NED reference frame by following the formulas in [39, p. 38-39]. The NED velocities can be found by using the SOG and COG from the AIS message as

$$\begin{bmatrix} V_N \\ V_E \end{bmatrix} = \begin{bmatrix} V\cos\chi \\ V\sin\chi \end{bmatrix}, \tag{4.41}$$

where V and χ are the SOG and COG respectively. As these data are obtained from the GNSS on board the transmitting vessel it is sensible to assume GNSS accuracy on the measurements. Large position errors may occur due to sampling time quantization when the sampling time is not an integer multiple of the UTC seconds, and is obviously dependent on the SOG and COG as well as the resolution of the time stamp. The worst case rounding error can be assumed to be $\pm 0.5\text{s}$. This error can be taken in consideration in the measurement covariance matrix by adding a position covariance proportional to the NED velocity, scaled by a variance related to the quantization error. Wilthil et al. [3] suggested a moment matched Gaussian distribution with zero mean and standard deviation $\sigma = \frac{1}{\sqrt{12}}$. The resulting measurement covariance matrix can therefor be written as

$$\mathbf{R_A} = \mathbf{R_{GNSS}} + \frac{1}{12}\mathbf{R_V} \tag{4.42}$$

where

$$\mathbf{R_{GNSS}} = \text{diag}[0.5^2, 0.1^2, 0.5^2, 0.1^2] \tag{4.43}$$

and

$$\mathbf{R_V} = \text{diag}[V_N^2, 0, V_E^2, 0]. \tag{4.44}$$

Using (4.42) one define the measurement model to be

$$\mathbf{z_{k,A}} = \mathbf{H_A}\mathbf{x_{k,A}} + \mathbf{w_{k,A}} \quad p(\mathbf{w_{k,A}}) \sim \mathcal{N}(\mathbf{w_{k,A}}; 0, \mathbf{R_A}), \tag{4.45}$$

where $\mathbf{H_A} = \mathbf{I_4}$ and the state vector $\mathbf{x_{k,A}} = \begin{bmatrix} N & V_N & E & V_E \end{bmatrix}^T$. Note that the state vector $\mathbf{x_{k,A}}$ differs from $\mathbf{x_{k,l}}$. This is due to implementation differences and should be fixed to have consistency in the models. If an AIS is to be installed on ReVolt to integrate a fusion between AIS and the other exteroceptive sensors, this certainly needs to be addressed. However, as the AIS messages are strictly used for tracking in simulations and not for real life experiments, the measurement models do not interfere with each other, and the problem was not pursued in this thesis.

Chapter 5

Collision Avoidance

This chapter will present the collision avoidance system (CAS) on ReVolt. First, the international guidelines for conduct at sea are presented, namely the COLREGS rules. Then, a short introduction to optimization-based control using model predictive control (MPC) is given. Finally, the algorithm used for the CAS on ReVolt, SBMPC, is presented in more detail, including the contributions made of the author of this thesis.

5.1 International Regulations for Preventing Collisions at Sea

For a fully autonomous vessel to function properly and safely, it must follow some guidelines to prevent collisions at sea. The International Maritime Organization (IMO) has published a set of navigational rules for ships and vessels called International Regulations for Preventing Collisions at Sea (COLREGS), where the objective is to prevent collision between vessels at sea. The COLREGS gives the ”rules for the road” and encourages ”good seamanship”, which makes the task of managing collision avoidance more complex and challenging since it is not enough only to prevent collision, but one has to do it in a particular manner. They are divided into six parts, covering different topics, namely:

- PART A - General
- PART B - Steering and sailing
- PART C - Lights and shapes
- PART D - Sound and light signals
- PART E - Exemption
- PART F - Verification of Compliance with the provisions of the Convention

However, the rules from part B about Steering and sailing, are the most relevant ones for this thesis. More specific, the rules 6,8,13,14,15,16 and 17 will be described in the following section, and are direct citations from [9].

5.1.1 Rules

Rule 6 - Safe speed

Every vessel shall at all times proceed at a safe speed so that she can take proper and effective action to avoid collision and be stopped within a distance appropriate to the prevailing circumstances and conditions.

Rule 8 - Actions to avoid collision

(a) Any action to avoid collision shall be taken in accordance with the Rules of this Part and shall, if the circumstances of the case admit, be positive, made in ample time and with due regard to the observance of good seamanship.

(b) Any alteration of and/or speed to avoid collision shall if the circumstances of the case admit, be large enough to be readily apparent to another vessel observing visually or by radar; a succession of small alterations of course and/or speed should be avoided.

(d) Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance. The effectiveness of the action shall be carefully checked until the other vessel is finally past and clear.

Rule 13 - Overtaking

(a) Notwithstanding anything contained in the Rules of part B, sections I and II, any vessel overtaking any other shall keep out of the way of the vessel being overtaken

(b) A vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam that is, in such a position with reference to the vessel she is overtaking, that at the night she would be able to see only the stern light of that vessel but neither of her sidelights.

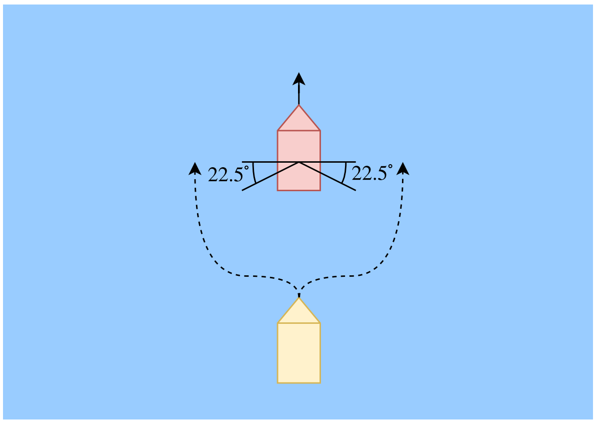


Figure 5.1: Simple illustration of the overtaking situation.

Rule 14 - Head-on situation

(a) When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses so as to involve risk of a collision, each shall alter her course to starboard so that each shall pass on the port side of the other.

(b) Such a situation shall be deemed to exist when a vessel sees the other ahead or nearly ahead and by night she could see the masthead lights of the other in a line or nearly in a line and/or both sidelights and by day she observes the corresponding aspect of the number of other vessels.

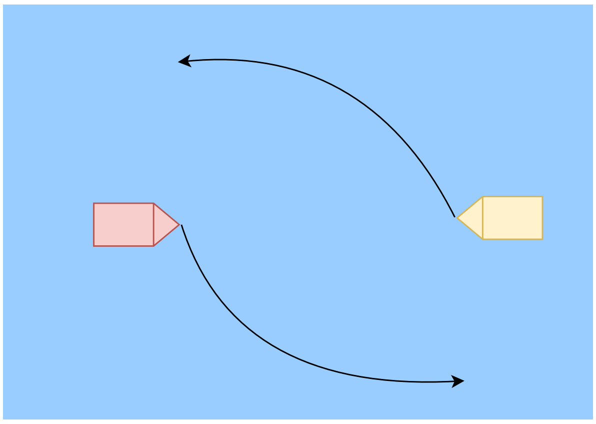


Figure 5.2: Simple illustration of a head-on situation showing the correct behavior.

Rule 15 - Crossing situation

When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

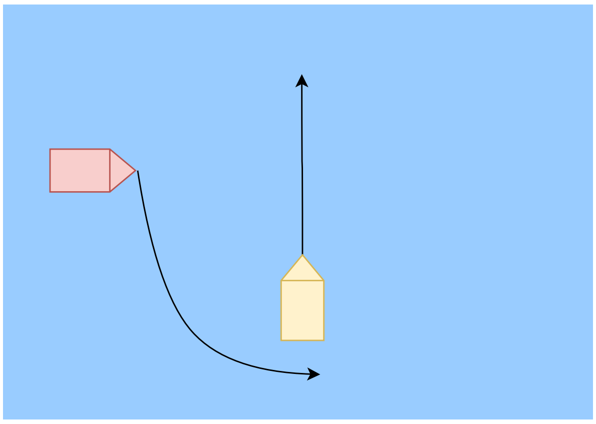


Figure 5.3: Simple illustration of a crossing situation where the correct behavior is for the vessel that has the other vessel on its starboard side to keep way.

Rule 16 - Actions by give-away vessel

Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.

Rule 17 - Actions by stand-on vessel

(a) (i) Where one of two vessels is to keep out of the way, the other shall keep her course and speed.

(ii) The latter vessel may, however, take action to avoid collision by her maneuver alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules.

(b) When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid a collision.

5.2 Model Predictive Control

Model predictive control (MPC) is an advanced optimization-based control method that can employ a nonlinear vehicle model to predict the optimal future behavior of the system, based on predictions of obstacles' motion. The uncertainty regarding the obstacle predictions can robustly be accounted for along with environmental forces. In addition, risk, hazard, and operational constraints and objectives can be formalized in a cost function, which is used in the optimization problem. The optimization will result in an optimal control input sequence, where only the first control input is applied to the system.

```
for $t = 0, 1, 2...$ do
    Get an estimate of the current state, $\hat{x}_t$, using the measurements up until
      the current time step.
    Solve a dynamic optimization problem on the prediction horizon, using
      $\hat{x}_t$ as the initial condition.
    Apply the first control move, $u_t$, from the solution above.
end
```

Algorithm 1: Output feedback of the MPC procedure [55].

A basic MPC algorithm is described in Algorithm 1. At each time step, a dynamic optimization problem is solved over a given time into the future, called the prediction horizon. This means that MPC uses a moving horizon, where the horizon changes from $t, ..., t+N$ to $t+1, ...t+N+1$. The selection of the initial state of the vehicle model is an important choice. One of the options is to choose either the state estimate $\hat{x}_{t+1}$ or the predicted state x^*_{t+1} computed at time t. The latter does not account for model errors and disturbances, which can result in a bad estimate of the initial state. Thus, the state estimate $\hat{x}_{t+1}$ is a more sensible option.

5.3 Scenario-based model predictive control

The Scenario-based model predictive control is an MPC-based algorithm that evaluates a finite set of scenarios in a cost function to compute an optimal control output. This algorithm can be applied in a CAS where the scenarios consist of the current state of the ownship, the predicted trajectory of the ownship generated using a pair of control behaviors from a finite set of predetermined control behaviors, and the predicted trajectories of the obstacles. Each of the scenarios is evaluated in a cost function designed to express the hazard level of the scenario, meaning that the scenario with the minimum associated cost is the optimal one. The control behaviors used to generate the optimal scenario are then applied as the optimal control output and transmitted to the autopilots. The algorithm is in turn repeated every t_{period}[s] to produce a new optimal control output. The following section will describe the SBMPC CAS on ReVolt in detail, which is a continuation of [5], and is based on the work done in [36], [37]and [6].

Architecture

The system architecture of ReVolt is showed in Figure 5.4 and is a modification of what Midjås [5] implemented in her thesis. Note that the SBMPC CAS is decoupled from both the Mission Planning and Guidance modules, which enables the possibility to use the collision avoidance method in a variety of control system architectures, as well as other collision avoidance methods. A pipeline has been developed by the author to enable the CAS to receive state estimates and associated variances of observable obstacles provided by the tracking system on ReVolt, and is further explained in Section 6.2. In addition, an obstacle management interface has been implemented with the purpose of determining the impact of tracks based on the duration of their lifetime and is heavily based on [6]. This means that a long-lasting track that seizes to exist should gradually have less impact in a collision avoidance situation. This obstacle information is one of four main inputs to the SBMPC algorithm, where the other three are; the current state of the ownship, references for course and speed, χ_d and u_d respectively, and a list of waypoints giving the desired path for the ASV.

The resulting output of the CAS is a course offset and a speed factor, χ_m and u_m respectively, that are used to form the modified references $\chi_c = \chi_d + \chi_m$ and $u_c = u_d \cdot u_m$. These modified references are forwarded to the autopilots (see Section 2.3) which guides the ASV in the direction of predicted collision free path. When there are no obstacles present the original reference is used, meaning that the initial condition of modified references are $\chi_{m0} = 0$ and $u_{m0} = 1$.

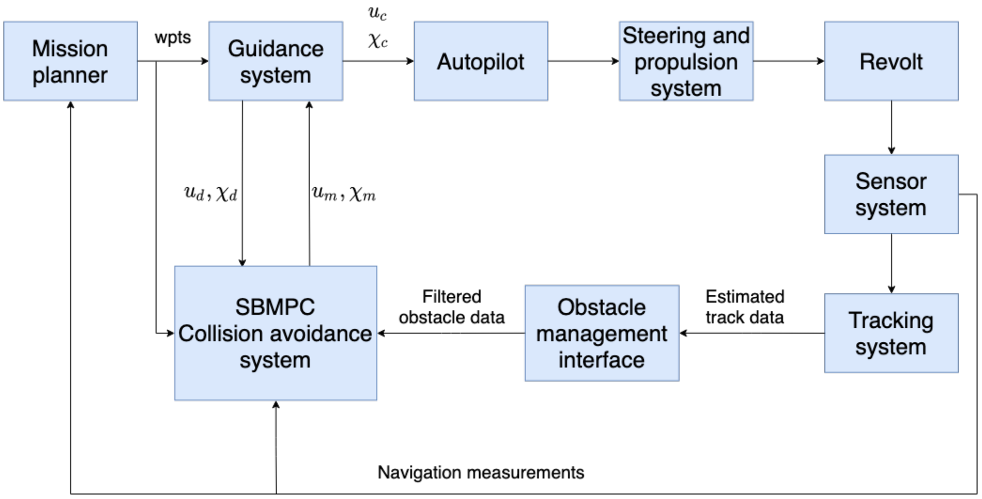


Figure 5.4: Flow chart of the system architecture.

Prediction of ownship trajectory

To accurately predict the ship dynamics over the prediction horizon the 3-DOF maneuvering model, explained in Section 2.2.1, in conjunction with the first order Euler method is used. The 3-DOF model enables the possibility to capture the vessel dynamics as well as environmental disturbances as well, even though the latter was not implemented in this thesis and is therefor left for future work. The CyberSea simulator (see Section 6.3) uses a 6-DOF maneuvering model of ReVolt with values acquired from Alfheim and Muggerud in [11] as well as some values from a towing tank test performed by DNV GL. The 3-DOF model in the SBMPC is based on this model and uses its values. The following numerical values are inserted in the mass, coriolis and damp matrices from (2.3) - (2.8):

Parameter	Value	Unit
m	300	kg
I_z	298	kg·m^2
x_g	-0.03	m
$X_{\dot{u}}$	6.93	kg
$Y_{\dot{v}}$	49.44	kg
$Y_{\dot{r}}$	7.007	kg·m
$N_{\dot{v}}$	7.028	kg·m
$N_{\dot{r}}$	24.556	kg·m^2
X_u	0.03074	kg/s
Y_v	0.1423	kg/s
Y_r	0	kg/s
N_v	0	kg/s
N_r	0.2193	kg·m/s

Table 5.1: Numerical values used in the system matrices in the 3-DOF model.

The first order Euler method is a simple method where the estimated state $\hat{\mathbf{x}}$ at time $k + 1$ is approximated by moving a small step along the tangent line of the estimated state at the previous time step k. Mathematically this is given by

$$\hat{\mathbf{x}}_{\mathbf{k+1}} = \hat{\mathbf{x}}_{\mathbf{k}} + hf(\hat{\mathbf{x}}_{\mathbf{k}}, t_k) \tag{5.1}$$

where $f(\hat{\mathbf{x}}_{\mathbf{k}}, t_k)$ is the derivative of $\hat{\mathbf{x}}_{\mathbf{k}}$ and h is the time step such that $t_{k+1} = t_k + h$. The idea behind the method is that if h is chosen small enough the approximation will not be that different from the actual state. Applying the Euler prediction in the SBMPC CAS is done by setting $\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\boldsymbol{\eta}}_{\boldsymbol{k}}$ and $f(\hat{\mathbf{x}}_{\mathbf{k}}, t_k) = \mathbf{R}(\hat{\psi_{\mathbf{k}}})\hat{\boldsymbol{\nu}}_{\mathbf{k}}$, and solving the system equations (2.1). Note that $\hat{\boldsymbol{\eta}}_{\boldsymbol{k}}$ is in the NED-frame while $\hat{\boldsymbol{\nu}}_{\boldsymbol{k}}$ is in the BODY-frame. This results in the following equations

$$\hat{\boldsymbol{\eta}}_{\boldsymbol{k}\mathbf{+1}} = \hat{\boldsymbol{\eta}}_{\boldsymbol{k}} + h\mathbf{R}(\hat{\psi_{\mathbf{k}}})\hat{\boldsymbol{\nu}}_{\mathbf{k}} \tag{5.2a}$$

$$\hat{\boldsymbol{\nu}}_{\boldsymbol{k}\mathbf{+1}} = \hat{\boldsymbol{\nu}}_{\boldsymbol{k}} + h\dot{\hat{\boldsymbol{\nu}}}_{\boldsymbol{k}} \tag{5.2b}$$

$$\dot{\hat{\boldsymbol{\nu}}}_{\boldsymbol{k}\mathbf{+1}} = \mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{C}(\boldsymbol{\nu}_{\boldsymbol{k}})\boldsymbol{\nu}_{\boldsymbol{k}} - \mathbf{D}(\boldsymbol{\nu}_{\boldsymbol{k}})\boldsymbol{\nu}_{\boldsymbol{k}}) \tag{5.2c}$$

$$\tag{5.2d}$$

where $\boldsymbol{\tau}$ is an artificial control input applied to the system given by

$$\boldsymbol{\tau} = \begin{bmatrix} c_1 + d_1 + K_{p,u}\tilde{u}_k \\ (K_{p,\psi}I_z)(\tilde{\psi}_k - K_{d,\psi}\hat{r}_k)\frac{1}{l_r} \\ (K_{p,\psi}I_z)(\tilde{\psi}_k - K_{d,\psi}\hat{r}_k) \end{bmatrix}, \tag{5.3}$$

where a feedback-linearizing controller and two PD controllers (see Section 2.3) are used to control speed, heading and yaw rate respectively. They have their own associated tuning parameters that were selected to be $K_{p,u} = 1$, $K_{p,\psi} = 1$ and $K_{p,\psi} = 5$.

The elements d_1 and c_1 are the first elements of the vectors $\boldsymbol{C}(\boldsymbol{\nu_k})\boldsymbol{\nu_k}$ and $\boldsymbol{D}(\boldsymbol{\nu_k})\boldsymbol{\nu_k}$ respectively, l_r is the distance from CG to the rudder, and $\tilde{u}_k = (u_{d,k} - \hat{u}_k), \tilde{\psi}_k = (\psi_{d,k} - \hat{\psi}_k)$. Note here that the desired heading reference $\psi_{d,k}$ is time dependent and not constant on the prediction horizon. This improvement is motivated by the fact that the assumption of keeping guidance constant does not fit with the behaviour of LOS guidance [5]. The time dependent heading reference is calculated in the same manner as in Section 2.4 for every time step on the prediction horizon using the predicted states.

In the cases where the ASV performs evasive maneuvers that deviates from the desired path, one can see from (2.27) that the increase in the cross-track error e_k will cause a larger value for $\chi_{r,k}$ and subsequently a more aggressive $\psi_{d,k}$ towards the desired path. The resulting optimal trajectory will, therefore, be less conservative as the inclusion of LOS guidance in the prediction horizon will result in a smoother trajectory compared to the predicted trajectory from [5].

Prediction of obstacle trajectory

The future motion of the obstacle is highly uncertain and is one of the main challenges in the collision avoidance problem. A simple short-term prediction of the obstacle trajectory is to use the CV model described in Section 2.2.2. By using the CV model, it follows that the obstacle is assumed to have both constant velocity and heading, resulting in a straight-line trajectory on the prediction horizon. This assumption is obviously a major simplification of reality, but in many cases, it is sufficient to avoid collision [36, 37, 6].

The obstacle information is acquired from the tracking system on ReVolt and is a new addition to the code base. This change was motivated by the fact that the previous solution obtained obstacle information directly from a simulated AIS sensor, which is not realistic for an ASV. The position and velocity estimate obtained from the tracking system is used as the initial state, and the trajectory is predicted using (2.10) for every time step on the prediction horizon.

In [5] the obstacle information was acquired directly from a simulated AIS sensor, giving ground truth measurements with no disturbances or uncertainties. However, in this thesis, the incoming sensor messages are passed through the tracking system, as a realistic closed-loop CAS would have. The Cybersea simulator can only produce AIS messages, meaning that simulated camera and LiDAR measurements are not available, as would be the case for a real-life test. However, by adding white noise to the simulated AIS messages, an acceptable replication of the LiDAR can be made. An example of how this is done is shown in Listing 5.1.

```python
import numpy as np
def addWhiteNoise(measurement,maxError):
    mean = 0
    numSamples = 1
    whiteNoise = np.random.normal(mean, maxError, numSamples)
    return measurement + whiteNoise
```

Listing 5.1: Example of how to add white noise in Python.

Control behaviors

The set of control behavior alternatives used in the prediction of the ownship trajectories are

- Course offset (χ_m): $\{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$
- Speed factor (u_m): $\{1, 0.5, 0\}$ corresponding to {keep speed, slow down, stop},

resulting in a total of $13 \cdot 3 = 39$ possible scenarios to be evaluated assuming a fixed control behavior on the prediction horizon. This has been the case for the SBMPC in [36], [37], [6] and [5] which this thesis is based on. To the author's knowledge, there has not been published any work where there is a possibility to change control behavior on the prediction horizon. In this thesis, there has been implemented the possibility to change the course offset n_{cp} number of times, resulting in an increase in scenarios being evaluated. The course offset is changed $\text{sgn}(\chi_m) \cdot 15°$ every t_{cp} seconds on the prediction horizon for a total of n_{cp} times, resulting in a total of $39 \cdot n_{cp}$ scenarios to be evaluated. This was motivated by the fact that an expansion of the trajectory space may lead to a more optimal trajectory [36][2, Section 5.1]. It was found that the configuration in Table 5.2 resulted in a more efficient trajectory in all of the tested collision scenarios. Choosing these parameters is situation-dependent as they should depend on the length of the prediction horizon, the distances d_{safe} and d_{init}, as well as the velocities of both the ownship and the obstacles. Finding a general expression for the time and number of change points is not straight forward and was not further investigated in this thesis. Using the parameters described in this scenario it was found that choosing $t_{cp} = 25$s and $n_{cp} = 3$ showed promising results. Little to no increase in performance was shown if the number of change points allowed surpassed 3. With that being said, the author cannot take full credit for the implementation as it was adapted by implementation by Giorgio Kufoalor in the Autosea project [2].

Parameter	Value
n_{cp}	3
t_{cp}[s]	25

Table 5.2: Change points parameters.

Cost function

The SBMPC cost function used in this thesis evaluates hazard criteria from a collision avoidance strategy point of view, and is based on [36] and the extensions made in [37] and [6]. The cost function consists of five main components, and can be described to be

1. the cost of colliding with an obstacle,
2. the cost of violating COLREGS,

3. the COLREGS transitional cost,
4. the cost of maneuvering, and
5. the cost of colliding with land,

making the SBMPC a COLREGS-compliant decision method. These components can be formalized mathematically in the following optimization problem

$$k^*(t_0) = \arg\min_k \mathcal{H}^k(t_0) \tag{5.4}$$

where

$$\begin{aligned} \mathcal{H}^k(t_0) = \max_i \max_{t \in D(t_0)} \big(& l_i(t_{lost}) \cdot c_i^k(u_m^k, \chi_m^k, t) \\ & + \kappa_i \mu_i^k(t) + \lambda_i \tau_i^k(u_m^k, \chi_m^k, t)\big) \\ & + f(u_m^k, \chi_m^k) + g(u_m^k, \chi_m^k) \end{aligned} \tag{5.5}$$

Here, the SBMPC evaluates all possible scenarios (k) and chooses the control behavior which minimizes $\mathcal{H}^k(t_0)$, where the subscript i denotes an obstacle, t_0 the current time and t the future time. $D(t_0)$ is the set that defines the discrete time period where each scenario is evaluated, and is defined to be $D(t_0) = t_0, t_0 + T_s, ..., t_0 + T$, where T_s is the sampling time and T the prediction horizon. The following sections will give more detailed descriptions of the cost function components $l_i, c_i, \kappa_i, \mu_i, \tau_i, f, g$, and is influenced by [36] and [6].

Collision cost

The first element of the cost function is $l_i(t_{lost}) \cdot c_m^k$, where $l_i(t_{lost})$ represents the track-loss factor and will be discussed later in the section. c_m^k represents collision hazard and is defined by

$$c_i^k(u_m^k, \chi_m^k, t) = \mathcal{C}_i^k(t)\mathcal{R}_i^k(t), \tag{5.6}$$

where $\mathcal{R}$ is the collision risk factor and $\mathcal{C}$ the cost associated with collision. The risk factor is defined as

$$\mathcal{R}_i^k(t) = \begin{cases} \frac{1}{|t-t_0|}^P \left(\frac{d_i^{safe}}{d_{0,i}^k(t)}\right)^q, & \text{if } d_{0,i}^k(t) \leq d_i^{safe} \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

where $t \geq t_0$ is the time of the prediction. The risk factor is only calculated when the predicted distance between the ownship and an obstacle, $d_{0,i}^k$ is lower than a chosen minimum allowed distance d_i^{safe}, that is $d_{0,i}^k \leq d_i^{safe}$. This means that a predicted trajectory that goes through the circular area with radius d_i^{safe} that encloses the obstacle will have a higher collision cost than a trajectory that falls outside said area, as illustrated in Figure 5.5. The exponent $q \geq 1$ and distance d_i^{safe} must be chosen with care, and should be large enough to comply with COLREGS rule 16 (See Section 5.1.1), taking early and substantial action to keep well clear of another vessel. In addition, COLREGS rule 18 must be accounted for when choosing these parameters, ensuring sufficient distance to vessels that are sailing, fishing, restricted in their ability to manoeuvre or that appear to not be under command. The collision risk is time dependent and the exponent $p \geq \frac{1}{2}$ weights the time until collision, prioritizing collisions that are closest in time.

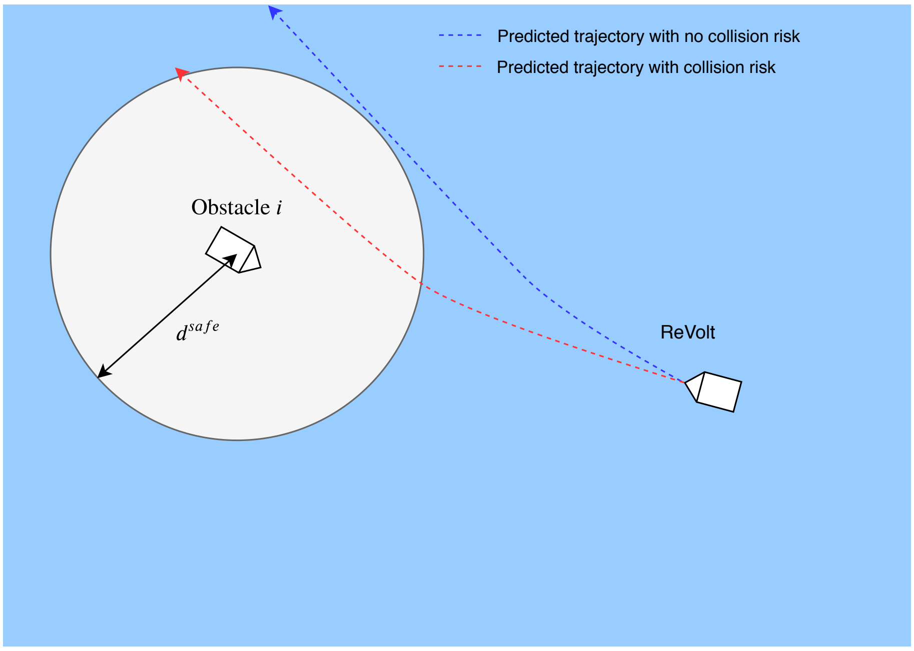

Predicted trajectory with no collision risk
Predicted trajectory with collision risk
Obstacle i
d^{safe}
ReVolt

Figure 5.5: Illustration of trajectory with no collision cost.

Furthermore, the cost associated with collision with obstacle i at time t is given by

$$\mathcal{C}_i^k = K_i^{coll} \left| \mathbf{v_0^k(t)} - \mathbf{v_i^k(t)} \right|^2 \tag{5.8}$$

and scales with the relative kinetic energy of both ReVolt and the obstacle. This cost is most important if collision with any obstacle is unavoidable. Here, K_i^{coll} is the scaling factor and may depend on multiple properties, like the size and type of the obstacle, the right to stay on or the responsibility to keep out of the way [36].

Finally, the track-loss factor $l_i(t_{lost})$ is introduced, and is a new addition to the cost function. It was first introduced in [6] and then adapted to the existing code on ReVolt. The track-loss factor is given by

$$l_i(t_{lost}) = \frac{T_s}{(t_{lost})^{q_1}}, \quad t_{lost} \geq T_s, \tag{5.9}$$

and reduces the collision cost of obstacle i when the tracking system terminates its track. The track-loss duration is given by t_{loss} and reduces the track-loss factor the larger it is, and additionally weighted by the tuning parameter $q_1 \geq 1$. A track is discarded if t_{lost} exceeds some short-duration $\bar{t}_{lost}$, or if the estimate of the track error covariance exceeds a predefined threshold. This can be done since it is observed that real tracks that are terminated quickly will be reassigned with a new track-ID, while false tracks or tracks that leave the tracking system's sensing range may never return again [6]. When obstacles are in close range, t_{lost} must be kept as small as possible since (falsely) terminated tracks that are reassigned as new tracks can deviate substantially from the lost track. This may occur in the event of a sharp turn. The effect of the lost track will, however, quickly be reduced, with the new track being given greater priority.

COLREGS violation cost

The second element of the cost function is the cost of violating the "rules for the roads", namely the COLREGS rules (see Section 5.1.1). An ASV performing collision avoidance maneuvers must satisfy these rules, and in order to determine whether or not they have been satisfied or violated, one must determine the different types of collision situations one can encounter. Thus, the boolean variables CLOSE, OVERTAKEN, STARBOARD, HEAD-ON and CROSSING are defined to describe the rules concerning the collision situation, namely rule 14 and 15. These variables are set by the following inequality tests at time t in scenario k, using the parameters described in Table 5.3:

Parameter	Description
$\mathbf{v}_o^k(t)$	Predicted velocity of the ownship
$\mathbf{v}_i^k(t)$	Predicted velocity of obstacle i
$\mathbf{L}_o^k(t)$	Unit vector in LOS direction from ownship to the obstacle i
$\mathbf{d}_{o,i}^k(t)$	Predicted distance between ownship and obstacle i
$\mathbf{d}_i^{cl}(t)$	The largest distance where COLREGS apply

Table 5.3: Parameter description of the COLREGS violation cost.

- **CLOSE:** An obstacle i is said do be CLOSE to ReVolt if

$$d_{o,i}^k \leq d_{close} \tag{5.10}$$

- **OVERTAKEN:** A vessel is said to be OVERTAKEN by an obstacle i if

$$\mathbf{v_o^k(t)^T v_i^k(t)} > \cos(68.5°)|\mathbf{v_o^k(t)}||\mathbf{v_i^k(t)}| \tag{5.11}$$

- **STARBOARD:** An obstacle i is said to be on STARBOARD side of ReVolt if

$$\angle\mathbf{L_i^k(t)} \geq \psi^k(t) \tag{5.12}$$

- **HEAD-ON:** An obstacle i is said to be HEAD-ON if it is close to ReVolt, and

$$|\mathbf{v_i^k(t)}| > 0.05 \tag{5.13}$$

$$\mathbf{v_o^k(t)^T v_i^k(t)} < -\cos(22.5°)|\mathbf{v_o^k(t)}||\mathbf{v_i^k(t)}| \tag{5.14}$$

$$\mathbf{v_o^k(t)^T L_i^k(t)} > \cos(\phi_{ahead})|\mathbf{v_o^k(t)}| \tag{5.15}$$

  where the ϕ_{ahead} is to be chosen.

- **CROSSING:** An obstacle i is said to be CROSSING if is close to ReVolt and

$$\mathbf{v_o^k(t)^T v_i^k(t)} < \cos(68.5°)|\mathbf{v_o^k(t)}||\mathbf{v_i^k(t)}| \tag{5.16}$$

As previously mentioned, the cost of violating COLREGS is represented as the term $\kappa_i\mu_i$, where κ_i is a weighting parameter and $\mu_i \in \{0, 1\}$ is a binary indicator representing whether or not COLREGS rule 14 or 15 have been violated. By using the boolean variables defined above, rule 14 and 15 can be defined in terms of:

RULE14 = CLOSE & STARBOARD & HEAD-ON
RULE15 = CLOSE & STARBOARD & CROSSED & NOT OVERTAKEN

and the binary indicator as

$$\mu_i^k(t) = \text{RULE14 or RULE15} \tag{5.17}$$

Note that rule 13, which states that the overtaking vessel shall keep out of the way, is incorporated implicitly in rule 14, making the term compliant with COLREGS rules 13, 14 and 15.

COLREGS transitional cost

The third element of the cost function is the COLREGS-transitional cost given by the term $\lambda_i\tau_i^k$. The objective of the cost is to penalize control behaviours that will lead to maneuvers violating COLREGS rules. The term itself consist of the binary indicator $\tau_i^k \in \{0, 1\}$ given by

$$\tau_i^k(t) = O_i^k(t) \vee Q_i^k(t) \vee X_i^k(t), \tag{5.18}$$

and the tuning parameter λ_i. The elements $O_i^k(t), Q_i^k(t)$ and $X_i^k(t)$ are also binary indicators representing which situation the ASV is in. Here, assigning the value 1 to each elements represent that

$$\begin{aligned} O_i^k(t) = 1 &\iff \text{the ASV is overtaking the obstacle } i \\ Q_i^k(t) = 1 &\iff \text{the obstacle } i \text{ is overtaking the ASV} \\ X_i^k(t) = 1 &\iff \text{a crossing situation} \end{aligned}$$

The following section will explain each scenario k at time t in more detail:

- **Overtaking** If the predicted location of obstacle i is not on the same side as the ASV at time t as observed at time t_0, a control behavior in scenario k is associated with a transitional cost. This is given by

$$O_i^k(t) = \begin{cases} O_i(t_0) \wedge S_i^k(t) & \text{if } \neg S_i^k(t_0) \\ O_i(t_0) \wedge \neg S_i^k(t) & \text{if } S_i^k(t_0) \end{cases} \tag{5.19}$$

  where $S_i(t) = 1$ represents that the obstacle i is on starboard side at time $t \in \{t_0 + T_s, ..., t_0 + T\}$. In addition, $O_i(t) = 1$ represents that the obstacle i is overtaken by the ASV, as explained in 5.3.

- **Being overtaken** Here, the same mathematical expression as in (5.19) can be used to describe whether or not the ASV is being overtaken by obstacle i, given by

$$Q_i^k(t) = \begin{cases} Q_i(t_0) \wedge S_i^k(t) & \text{if } \neg S_i^k(t_0) \\ Q_i(t_0) \wedge \neg S_i^k(t) & \text{if } S_i^k(t_0) \end{cases} \tag{5.20}$$

  The difference between (5.19) and (5.20) is from which perspective the situation is seen from. Here, $Q_i(t) = 1$ represents that the ASV is being overtaken by another vessel at time t, opposed to that the ASV is overtaking obstacle i.

- **Crossing** According to COLREGS, the vessel crossing from starboard side has the right of way, as seen in Figure 5.3. The correct behavior is for the give way vessel (vessel that has another vessel on its starboard side) is to make an starboard maneuver, resulting in the crossing vessel to be on the port side when the crossing is finished. Any control behavior leading to the crossing vessel being on the starboard side after a crossing will therefor have an associated transitional cost. The cost is given by

$$X_i^k(t) = X_i(t_0) \wedge S_i(t_0) \wedge S_i^k(t) \wedge \text{turn to port} \tag{5.21}$$

  where $X_i(t) = 1$ indicates that the ASV is in a crossing situation at time t, where the criteria for the situation is explained in 5.1.1.

Maneuvering effort and grounding cost

The final element of the cost function is the cost of maneuvering, with the purpose of favoring predictable and straight paths,penalizing high offsets from the nominal course and speed. The term is given by

$$\begin{aligned} f(u_m, \chi_m) =& K_{u_m}(1 - u_m) + K_\chi(\chi_m) + \Delta_{u_m}(u_m, u_{m,last}) \\ &+ \Delta_\chi(\chi_m, \chi_{m,last}) \end{aligned} \tag{5.22}$$

where $\Delta_{u_m}, \Delta_{\chi_m}$ and K_χ are penalty functions given by the following equations:

$$K_\chi(\chi_m) = \begin{cases} K_{\chi,port}\chi_m^2, & \text{if turn to port} \\ K_{\chi,starboard}\chi_m^2, & \text{if turn to starboard} \end{cases} \tag{5.23}$$

$$\Delta_{u_m}(u_m, u_{m,last}) = K_{\Delta u_m}|u_m - u_{m,last}| \tag{5.24}$$

$$\Delta_{\chi}(\chi_m, \chi_{m,last}) = \begin{cases} K_{\Delta\chi,port}(\chi_m, \chi_{m,last})^2, & \text{if turn to port} \\ K_{\Delta\chi,starboard}(\chi_m, \chi_{m,last})^2, & \text{if turn to starboard} \end{cases} \tag{5.25}$$

Here, $K_{u_m}, K_{\chi}, K_{\Delta u_m}$ and $K_{\Delta\chi}$ are tuning parameters. K_{χ}, (5.23) is the penalty function that penalizes the size of the offset, using the tuning parameters $K_{\Delta\chi,port}$ and $K_{\Delta\chi,starboard}$. Δ_{χ}, given in (5.25), on the other hand penalizes the change in the offset and favours turning to starboard side, working towards COLREGS compliance. Note that both (5.23) and (5.25) have a squared term, making the cost asymmetric. The reason for this is to ensure compliance with the COLREGS rules 14,15 and 17 (See Section 5.1.1).

The grounding cost $g(\cdot)$, which gives a penalty based on where the ownship is in relation to its nearby environment, based of electronic map data, can also be included. However, this was not opted for in this thesis.

Tuning the parameters

The amount of parameters in the SBMPC makes the tuning job tedious and difficult. The logic used to tune the parameters used in this thesis was to classify the hazard level of each cost of the cost function. The hazard level classification was set to: $l_i(t_{lost}) \cdot c_i^k > \lambda_i \tau_i^k > \kappa_i \mu_i^k(t) > f$ Obviously, the collision cost $l_i(t_{lost}) \cdot c_i^k(u_m^k, \chi_m^k, t)$ should contribute to the greatest cost in the cases where it is calculated, as the end goal is to avoid a collision. The COLREGS transitional cost was prioritized over the cost of violating the COLREGS rules as aborting COLREGS-compliant maneuvers was considered more hazardous than just violating COLREGS. The cost with the least hazard level was decided to be the maneuvering cost as this cost penalizes guidance deviations, which is not a crucial criterion to consider in a hazardous situation. The maneuvering cost contributes when the threat of colliding decreases, e.g., moving away from each other, ensuring that the ASV returns to its desired path.

Chapter 6

Implementation

This chapter will give a brief introduction to the implementation aspect of this thesis. First, the framework used to run the already existing control system will be presented, namely the Robot Operating System (ROS). Then, the framework used to implement the track system will be presented. Finally, the CyberSea simulator used for testing the closed-loop system on ReVolt will be presented.

6.1 Robot Operating System

ROS is an open-source framework designed to simplify developing complex and robust robotics systems. The system is structured as several small programs communicating by passing messages to one another, enabling the creation of generic modules that can be applied to numerous applications and classes of robotic systems. The creators of ROS chose this structure to encourage the users to create, share, and reuse code to the global robotics community. ROS programs, called *nodes*, communicate with each other asynchronously using *topic* with a subscribe/publish logic, and synchronously using *services* with a request/response logic. The nodes can be written in C++, Python, and Lisp, making the communication between programs independent of the programming language used.

6.2 Autosea framework

The main objective of the Autosea project [2] was to develop closed-loop target tracking and collision avoidance systems using sensor fusion for autonomous ships. The tracking system framework used on the Autosea experiments was developed by Erik Wilthil and Andreas Flåten [3], and consists of a python library named *autoseapy* and a cluster ROS-packages called *autosea_ros*. This framework was implemented and adapted on ReVolt in [4], and further extended in work done by the author.

Enabling the tracking system

The tracking system used in real-life experiments on ReVolt is enabled by launching the *lidar_camera_tracker* node using a certain amount of parameters depending on whether the PDAF or IPDA algorithm is used. These parameters include initialization values for the tracking-algorithm, measurement covariances for the camera and LiDAR, and specified topics and services to be used by the node. The subscribers in the node listen to measurements from their respective sensors and trigger a callback function that handles the incoming measurement. Both the LiDAR and camera measurements are stamped at acquisition time and stored in their respective measurement message.

Similar logic is applied to the AIS tracker used in simulations, where the *ais_tracker node* is launched, which processes the incoming AIS messages and passes it to the tracker. The tracking algorithm used in the simulator is discussed further in Section 7.3

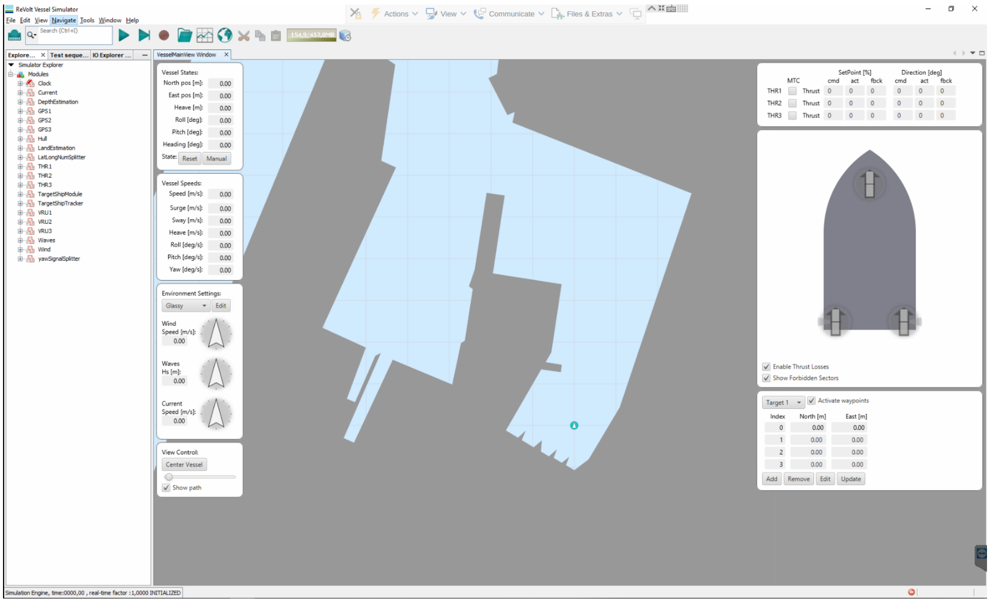
Connecting the tracking system and CAS

A pipeline has been developed for the ReVolt system to connect the tracking system, and CAS and is a new addition to the code base. The pipeline consists of a node that collects and processes state information about the ownship and obstacles before passing it to the SBMPC algorithm every t_{period} seconds. Instead of listening continuously to the track system, the node calls a ROS service that requests tracks at a given timestamp, in which the track system responds to, handling the prediction and the interpolation. This solution was originally developed by the authors of autoseapy and further adapted for ReVolt. Moreover, the node can handle raw AIS measurements from the simulation platform if desired (see Section 6.3). The node publishes the optimal control input obtained from the SBMPC behavior to the GNC system, meaning that the node is decoupled from the GNC system and can be used in a variety of control system architectures. Thus, the new pipeline provides modularity and usability to the code base, which are important attributes for complex control systems.

6.3 Cybersea simulator

A realistic simulation platform is critical for developing and testing a complex control system such as ReVolt. It enables a risk-free and efficient analysis that gives an early insight into potential conflicts in the code. The simulation platform used in this thesis is the Cybersea simulator provided by DNV GL. It includes a digital twin of ReVolt, which allows for the same code in both the simulation environment as in a real-life experiment. The Cybersea simulator is run separately from the control system and provides the same input to the control system as the real-life ReVolt would do. This makes it impossible for the control system to distinguish between simulated and real-life data, which shortens the gap between simulations and real-life testing.

The Cybersea environment, depicted in Figure 6.1, utilizes a detailed map of



Vessel States:

North pos [m]:	0.00
East pos [m]:	0.00
Heave [m]:	0.00
Roll [deg]:	0.00
Pitch [deg]:	0.00
Heading [deg]:	0.00

State: Reset Manual

Vessel Speeds:

Speed [m/s]:	0.00
Surge [m/s]:	0.00
Sway [m/s]:	0.00
Heave [m/s]:	0.00
Roll [deg/s]:	0.00
Pitch [deg/s]:	0.00
Yaw [deg/s]:	0.00

Environment Settings: Glassy Edit

Wind Speed [m/s]: 0.00

Waves Hs [m]: 0.00

Current Speed [m/s]: 0.00

View Control: Center Vessel

☑ Show path

			SetPoint [%]			Direction [deg]		
	MTC		cmd	act	fbck	cmd	act	fbck
THR1	□	Thrust	0	0	0	0	0	0
THR2	□	Thrust	0	0	0	0	0	0
THR3	□	Thrust	0	0	0	0	0	0

☑ Enable Thrust Losses

☑ Show Forbidden Sectors

Target 1 ☑ Activate waypoints

Index	North [m]	East [m]
0	0.00	0.00
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00

Simulation Engine, time:0000,00 , real-time factor :1,0000 INITIALIZED

Figure 6.1: Cybersea environment.

Trondheimfjorden, which is the local test environment for autonomous maritime vehicles at NTNU. The simulator displays a multitude of sensory data, allowing remote monitoring of the system while operating. This includes all motion and forces in all 6 degrees of freedom, thrust forces, and environmental forces. Additionally, the user can set both the magnitude and direction of environmental forces in the simulation environment.

The Cybersea simulator also supports the use of up to 32 virtual obstacle ships in the environment, where the user decides the vessel's initial state and further behavior. The obstacles follow waypoints using the same navigation as in Section 2.4, as well as using the VO collision avoidance if desired. Although not tuned perfectly, using the collision avoidance method allows for more realistic collision situations to be simulated. The data from the simulator is transmitted to the control system using Modbus, a communication protocol that enables communication amongst devices connected to the same network, except for the obstacle information, which is transmitted over UDP to the control system as an AIS message (see Section 3.3).

Chapter 7

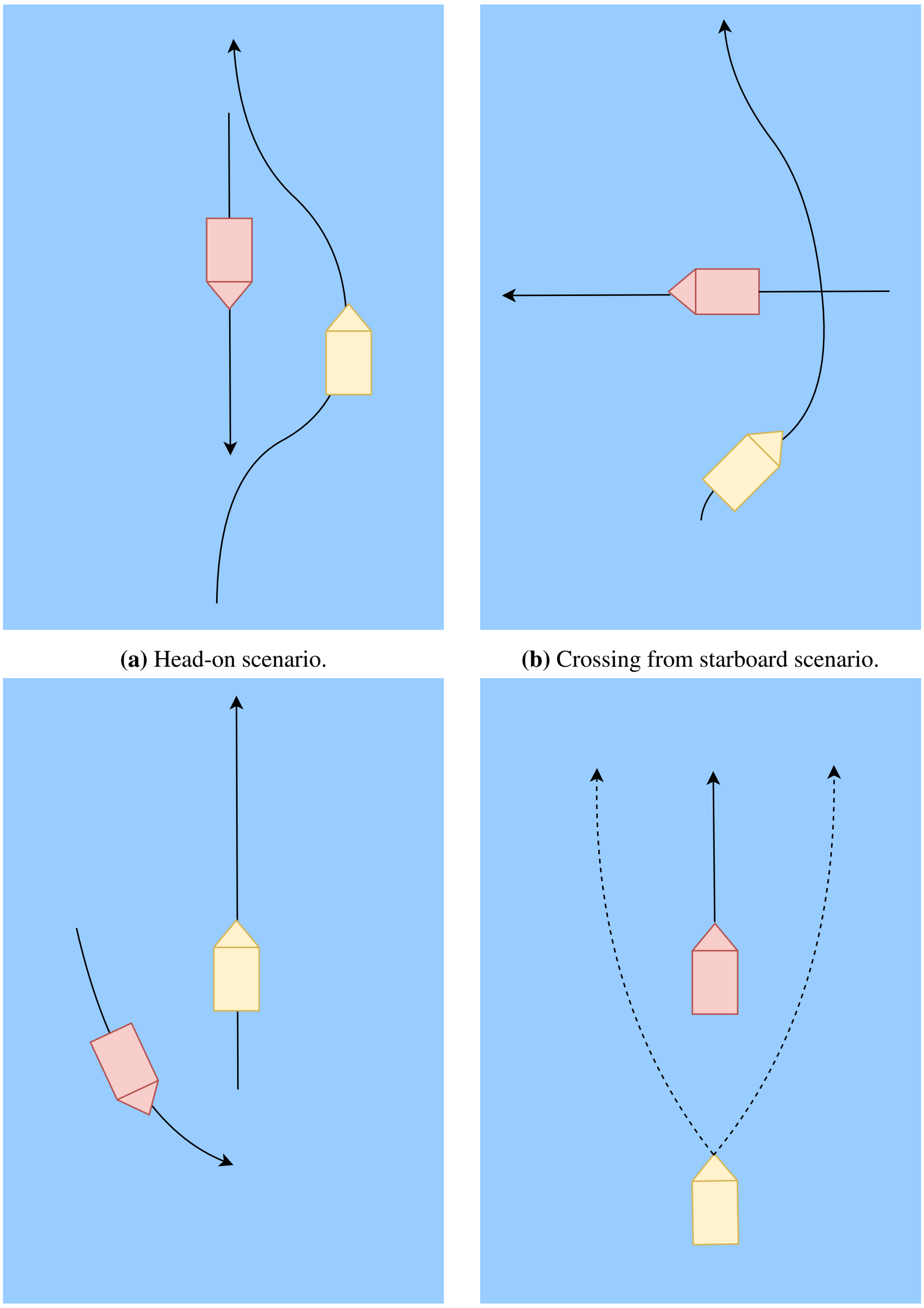
Simulation study

This chapter presents the closed-loop real-time simulation studies of the tracking and COLAV systems implemented on ReVolt. First, the evaluation metrics used to evaluate the performance of the AIS tracking system and CAS will be presented. Next, simulations using the cruise speed of ReVolt while ignoring the range restrictions of the tracking system are displayed. The purpose is to isolate the performance of the SBMPC where the tracking range and GNC system are not the restricting factors. Finally, simulations with tracking system restrictions at a lower cruise speed are presented to replicate the scenarios conducted in real-life experiments.

7.1 Setup

The first simulated scenarios are standard scenarios where the correct actions are clear to verify that the new contributions perform as intended. These scenarios are performed with a single obstacle and include the scenarios head-on, crossing from port and starboard, overtaking, and being overtaken. Then, some complex situations using multiple obstacles to replicate a more realistic scenario are tested.

The SBMPC parameters used in both Section 7.4 and Section 7.5 are given in Table 7.1 and was found using the tuning principles from Section 5.3. As the range of the two cases are different, given by d_{init}, the prediction horizon is reduced in the realistic to reduce unnecessary computation, which in turn reduces the computational cost.



(a) Head-on scenario.

(b) Crossing from starboard scenario.

(c) Crossing from port scenario.

(d) Overtaking scenario.

Figure 7.1: Overview of the scenarios to be simulated in the simulation study. Here, ReVolt is depicted as the yellow vessel, while the red vessel is the obstacle. The arrows indicate the expected behavior according to COLREGS.

Parameter	Ideal case	Realistic case	Description
T [s]	300	150	Predction horizon
DT [s]	0.5	0.5	Time step
t_{period}[s]	3	3	Release period of the SBMPC task
P	1	1	Weight on time to evaluation instant
Q	4	4	Weight on distance to evaluatino instant
d_{init}[m]	150	50	Distance where COLAV is activated
d_{close}[m]	150	50	Distance where COLREGS are said to apply
d_{safe}[m]	50	10	Minimal distance that is considered safe
K_{coll}	10	10	Collision cost
κ	3	3	Cost of not complying with the COLREGS
λ	10	10	Cost of changing a COLREGS behavior
K_{u_m}	2.5	2.5	Cost of deviating from nominal speed
$K_{\Delta u}$	0.5	0.5	Cost of changing the speed offset
$K_{\chi,port}$	1.8	1.8	Cost of deviating from nominal course to port
$K_{\chi,starboard}$	1.5	1.5	Cost of deviating from nominal course to starboard
$K_{\Delta\chi,port}$	0.5	0.5	Cost of changing course offset to port
$K_{\Delta\chi,starboard}$	0.9	0.9	Cost of changing course offset to starboard
ϕ_{AH} [°]	68.5	68.5	Angle which an obstacle is said to be ahead
ϕ_{OT} [°]	68.5	68.5	Angle which an obstacle is said to be overtaking the ship
ϕ_{HO} [°]	22.5	22.5	Angle which an obstacle is said to be head-on
ϕ_{CR} [°]	68.5	68.5	Angle which an obstacle is said to be crossing

Table 7.1: SBMPC parameters used in the ideal and realistic case.

7.2 Evaluation metrics

7.2.1 Tracking metrics

In order to evaluate the performance of the AIS tracking system a set of metrics are used: RMSE, average euclidean error (AEE) and average normalized innovation error (ANIS):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\nu_i^T \nu_i}} \tag{7.1}$$

$$\text{AEE} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{\nu_i^T \nu_i})^{1/2} \tag{7.2}$$

$$\text{ANIS} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\nu_i^T S_i^{-1} \nu_i}. \tag{7.3}$$

The first two are used to evaluate the deviation of the state estimate. The ANIS is used to test the consistency of the filter by considering whether the filter noise is consistent with the actual measurement noise. As the ANIS should follow a χ^2-distribution a confidence interval between α and $1 - \alpha$ can be made in order to see the percentage of measurements that are inside this interval.

It should be noted that these metrics by themselves cannot be used to evaluate whether the performance of the tracking system is sufficient. As the main objective is to create a COLREGS compliant CAS, the magnitude of the tracking metrics must be seen in this context to evaluate the performance of the tracking system.

7.2.2 CAS metrics

A common way of evaluating the performance of COLAV algorithms is to use the performance metrics developed by Woerner in [56], which quantifies the performance of the COLAV algorithms in terms of safety and COLREGS compliance. These metrics are useful when comparing different COLAV algorithms in a variety of scenarios. However, as only a single COLAV algorithm is evaluated here, a more simplistic metric is deemed sufficient for this thesis's scope.

To evaluate the performance of the SBMPC the closest point of approach (CPA) is used. Using the safety distance d_{safe} is obviously the natural choice that the CPA is compared with. This leads to the following definition:

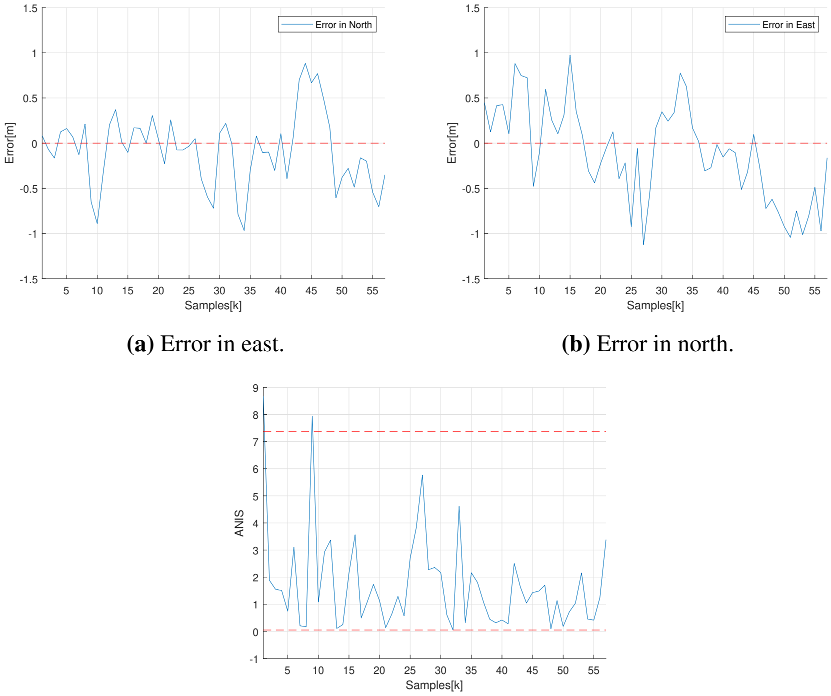
Definition 7.2.1 (Satisfactory manever)**.** A satisfactory maneuver of the CAS is when the maneuver does not violate the COLREGS and the CPA between an obstacle and the ownship is larger than the safety distance d_{safe}. Consequently, a CPA closer than d_{safe} is deemed as an unsatisfactory behavior.

To compare the performance of the SBMPC with and without change points, the same scenario will be performed by both cases and plotted over an equal period, revealing the difference ineffectiveness. The effectiveness will be quantified in the euclidean distance between them and the maximum cross-track error during the maneuver. The reason being that it is desirable to minimize the time usage and cross-track error during evasive maneuvers while still complying with the COLREGS. This way, the evasive maneuver will have minimal impact on the path-following objective.

7.3 AIS tracking system

The tracking system in the simulation study uses measurements from simulated AIS messages in the Cybersea simulator (see Section 6.3) giving ground truth measurements. The current simulator does not support the use of camera or LiDAR. However, to replicate the behavior of the LiDAR the AIS messages are contaminated with white noise, as described in Listing 5.1. Recall from Section 3.5 that the RMSE of the average LiDAR transformation was within the same order of magnitude as the results from [4]. Due to the limited access to ReVolt it was not possible to verify calibration results from Section 3.5. Hence, the mean square error (MSE) found in [4], 0.39m^2 at 50m, is used as it is reasonable to assume that the MSE from the new calibration will be in the same order of magnitude as well. Assuming zero bias, one can use that MSE $= \sigma^2$ to form the white noise $w \sim \mathcal{N}(0, k_{MSE}\sigma_l^2)$, where $k_{MSE} = d_{init}/50$

is a scaling parameter to make the MSE proportional to the range of the tracking system.



(a) Error in east.

(b) Error in north.

(c) Plot of the ANIS where 96% of the measurements were inside the 95% confidence interval.

Figure 7.2: Performance of the MMKF-tracking algorithm used in the ideal case with $k_{MSE} = 3$.

The tracking algorithm used in the simulation study was the Missing Measurement Kalman filter (MMKF) adapted from [3]. AIS messages are received at different rates depending on the speed and status of the ship [46], and it is convenient to use a fixed sampling rate 1s for all ships [3]. The MMKF works as a standard Kalman filter (see Section 4.1.1), but performs dead reckoning using the last predicted value for the iterations when new measurements have not been received.

The tracking system was tested by tracking an obstacle over a 285s period, where the AIS measurements were transmitted every 5s. The added white noise was set to $w \sim \mathcal{N}(0, 3\sigma_l^2)$ as the majority of the simulations were performed with a tracking range of $d_{init} = 150\text{m}$. The results are shown in Figure 7.2 and Table 7.2 where 96% of the measurements were inside the confidence interval. The track system shows a low average error in both north and east, while the maximum absolute error in north and east was 1.1210m and 0.9657m, respectively. Similar results were found over multiple tests, and the results are therefor not repeated.

The tracks do have a noisy behavior with multiple spikes occurring. However, the magnitude of the spikes is relatively low and does not constitute any real threat.

Metric	Value
$\text{RMSE}_N[\text{m}^2]$	0.4046
$\text{RMSE}_E[\text{m}^2]$	0.5323
$\text{AEE}_N[\text{m}]$	0.3085
$\text{AEE}_E[\text{m}]$	0.4316
ANIS $[\text{m}^2]$	1.7226

Table 7.2: Metric results.

This will be further discussed in Section 7.6. The tracker completed its computations within 0.09s for each received AIS measurement, showing sufficient real-time capabilities. Thus, the performance of the MMKF is deemed sufficient for COLAV purposes in the simulation study.

7.4 Results from the Ideal case

To evaluate the performance of the SBMPC the ideal case uses the cruise speed of 1m/s in order to get the optimal performance of the GNC system. This is also the cruise speed of the obstacles in the ideal case. Furthermore, the range restrictions of the track system on ReVolt are ignored, allowing $d_{init} = 150\text{m}$. This was the shortest distance at which the CAS performed satisfactorily while also showing proactive behavior in all scenarios. The remaining SBMPC parameters can be seen in Table 5.3. These parameters are tuned differently than in the previous SBMPC on ReVolt[5], where K_{u_m} and $K_{\chi,port}$ are reduced from 100 to 2.5 and 1.8 respectively, enabling ReVolt to slow down and turn to port if necessary.

7.4.1 Head-on

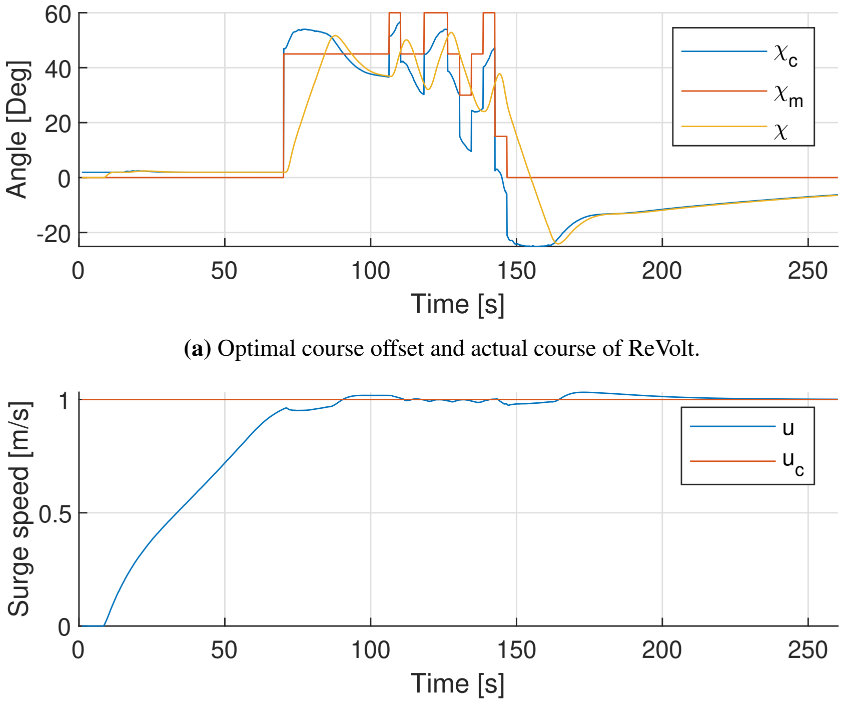
In a head-on situation, the correct course of action is to alter the course to starboard such that the vessels pass on the port side of each other, according to COLREGS rule 14 (see Section 5.1.1). As the obstacle is unable to do any maneuvers, it is expected that ReVolt must do a starboard maneuver such that the vessels pass at a safe distance according to rule 8 (Section 5.1.1). The obstacle started 250m north of the origin heading southbound, while ReVolt started in the origin heading northbound. This initial state of ReVolt will be kept constant during all scenarios.

The course of actions can be seen in Figure 7.4 where the results are satisfactory according to Definition 7.2.1 as the CPA was 45.6m. In the beginning, Revolt performs a hard starboard maneuver to avoid a collision, as the vessels are moving straight towards each other. Hence, a large course offset is needed in the early stages of the scenario, as can be seen in Figure 7.3a where $\chi_m = 45°$. As the vessels pas each other, a smaller offset is needed to avoid entering the safety perimeter, resulting in the optimal trajectory in Figure 7.4b that approaches the nominal path, opposed to the trajectory in Figure 7.4a. The optimal trajectory approaches the nominal path if χ_d dominates χ_m, and the optimal trajectory deviates from the nominal path if χ_m

dominates χ_d as in Figure 7.4a. Once the obstacle is passed, the collision is avoided, and ReVolt can return to its nominal path.

The evasive maneuver performed in the head-on scenario was both satisfying and smooth, where the cruise speed was maintained during the maneuver, as seen in Figure 7.3b. The SBMPC chose $n_{cp} = 1$ as the optimal solution 10 out of 25 SBMPC cycles before the obstacle was passed, where the remaining optimal solutions were without change points.

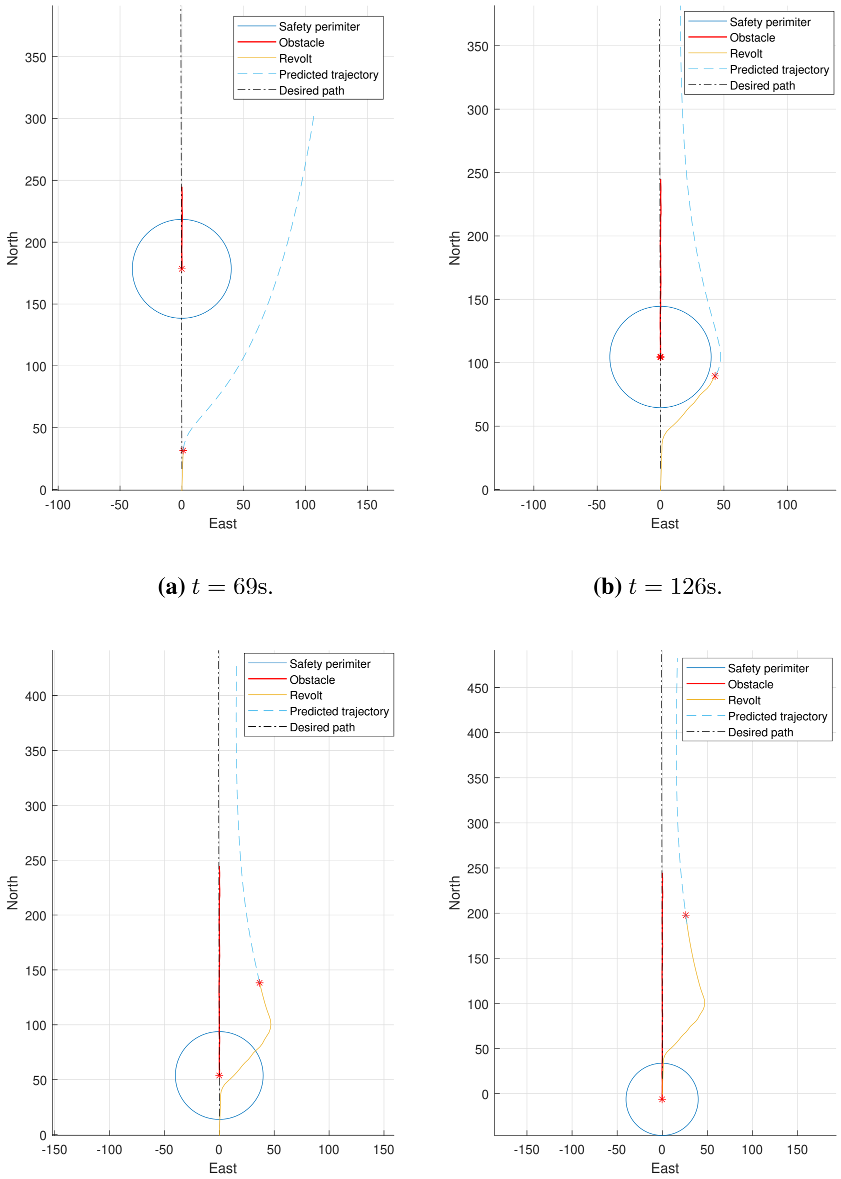
When active, the SBMPC showed promising real-time capabilities, where the average computational time was 0.9747s with a standard deviation of $\sigma = 0.0532$s. Similar results were found in all single target scenarios, and computational time will therefor not be repeated in the oncoming scenarios.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and speed course of ReVolt.

Figure 7.3: Optimal course and speed offsets in the head-on scenario.



(a) $t = 69$s.

(b) $t = 126$s.

(c) $t = 165$s.

(d) $t = 213$s.

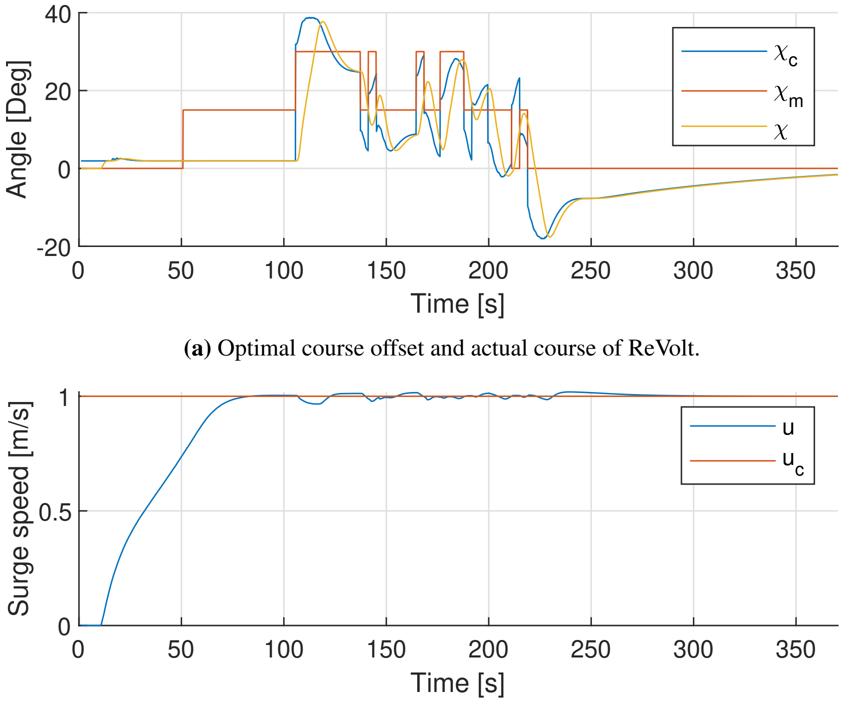
Figure 7.4: Course of action in the head-on scenario. The CAS was activated at $t = 69$s when $d_{o,1} <= d_{init}$. The CPA was 45.60m and is shown in Figure 7.4b.

7.4.2 Crossing from port

The next scenario is the crossing from port scenario. As ReVolt starts in the origin heading northbound, the obstacle has to start on the port side of ReVolt going from west to east. According to COLREGS rule 15 (Section 5.1.1), the vessel that has the other vessel on starboard side shall keep way, which in this case will be the obstacle. However, as the obstacle is passive, no attempt to follow COLREGS or perform evasive maneuvers will be attempted, and ReVolt is forced to do the maneuver in compliance with rule 17. The crossing from port situation will, therefore, be presented in two different scenarios: (1) The obstacle starts $150\mathrm{m}$ north and $220\mathrm{m}$ west of the origin such that ReVolt can pass the obstacle with the obstacle being on its port side during the maneuver, and (2) the obstacle starts at $150\mathrm{m}$ north, and $150\mathrm{m}$ west of the origin such that ReVolt is not able to pass the obstacle with the cruise speed of $1\mathrm{m/s}$ and must keep way until the obstacle has passed. ReVolt will, consequently, pass the obstacle with the obstacle being on ReVolt's starboard side.

Scenario 1

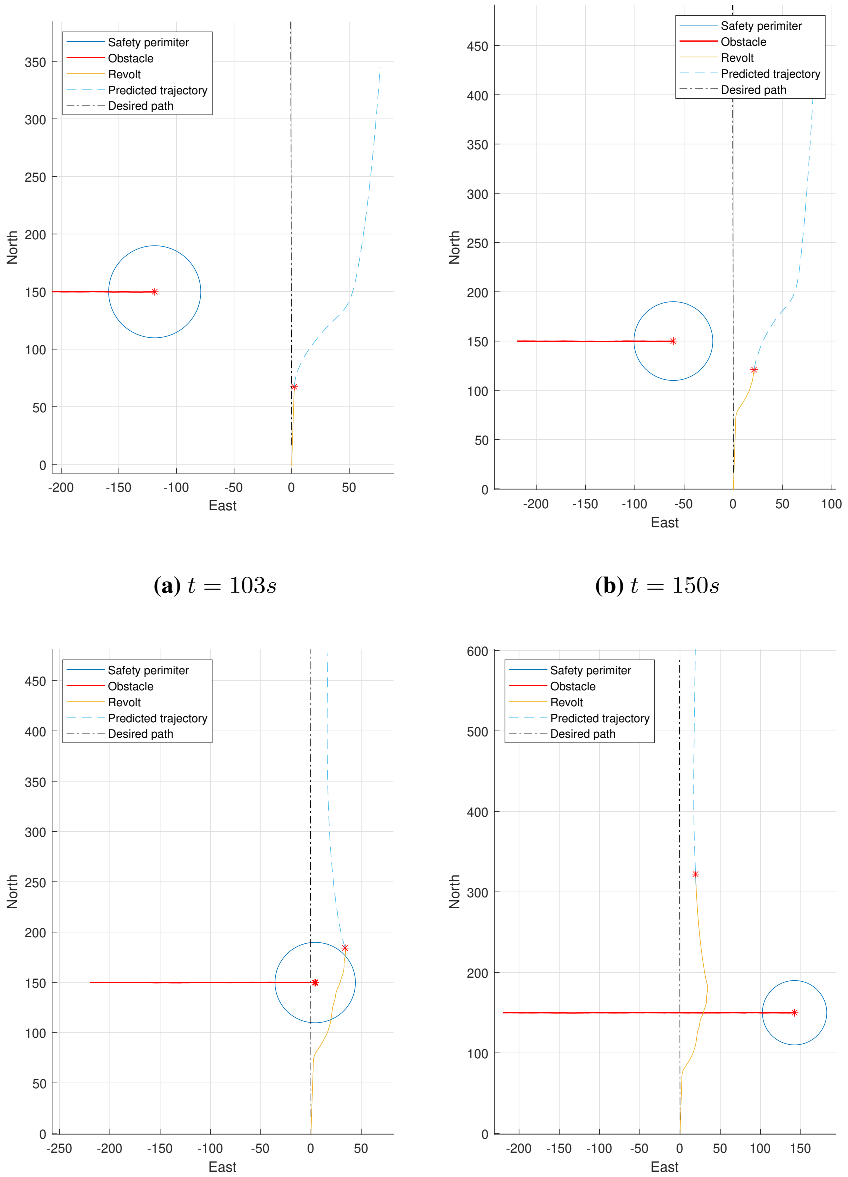
The course of action in the first crossing from port scenario is shown in Figure 7.6, showing that ReVolt performs the correct behavior according to the COLREGS. The keep away vessel shows no attempt to alter its course, resulting in that ReVolt must take action to avoid a collision. The speed was kept constant during the whole scenario, giving a smooth evasive maneuver without unnecessary stops. The CPA was $45.44\mathrm{m}$ meaning that the evasive maneuver was satisfactory according to Definition 7.2.1. The SBMPC chose $n_{cp} = 3$ as the optimal solution 19 out of 23 SBMPC cycles before passing the obstacle, while the remaining four optimal solutions were chosen to be $n_{cp} = 2$. The use of change points can also be seen in the trajectories in Figure 7.6a and Figure 7.6b respectively as the trajectories turn harder towards starboard at a later time step, which is expected as the course offset increases during the prediction horizon.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and actual speed of ReVolt.

Figure 7.5: Optimal course and speed offsets in the crossing from port scenario 1.



(a) $t = 103s$

(b) $t = 150s$

(c) $t = 201$

(d) $t = 312$

Figure 7.6: Crossing from port scenario 1 where ReVolt is able to pass with the obstacle on port side. The CPA was 45.44m and is shown in Figure 7.6c.

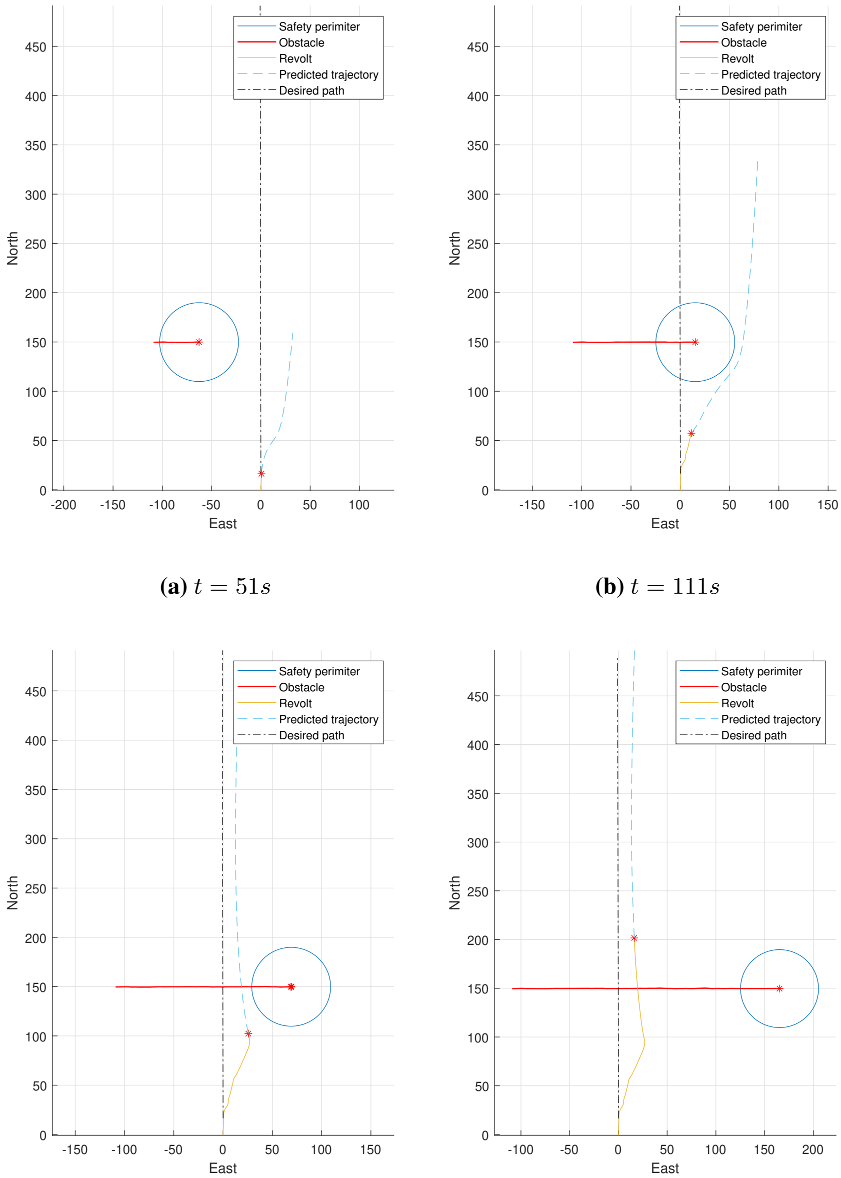
Scenario 2

The course of action in second crossing from port scenario is shown in Figure 7.7. The correct maneuver in this situation is that the obstacle shall keep way as ReVolt is on starboard side of the obstacle. However, the obstacle keeps its course, and it forces ReVolt to slow down to avoid collision since crossing in front of the obstacle is impossible given the current cruise speed. This can be seen in Figure 7.8b where the speed offset is set to $u_m = 0.5$ for a long period of time, as well as in Figure 7.7a where the optimal trajectory is shorter due to the low speed. It can be observed from Figure 7.7b that as long as the vessels are in a crossing situation, a starboard maneuver is preferred. However, when the crossing situation is over, ReVolt returns to its nominal path.

The evasive maneuver is well within safety distance requirements of Definition 7.2.1, where the CPA was $64.44\mathrm{m}$. The large CPA implies that the maneuver complied with COLREGS rules 8 and 16, as the action is taken in ample time, passing at a safe distance. Even though ReVolt had to slow down, the speed offset was kept constant at 0.5, resulting in a smooth behavior. During this scenario, the optimal control behavior included the use of change points 33 out of 35 SBMPC cycles before the vessels passed each other. The overview of the change points can be seen in Table 7.3.

Change points	#Selected
0	2
1	7
2	15
3	11

Table 7.3: Overview over the use of change points in the crossing from port scenario 2 before the vessels passed each other.



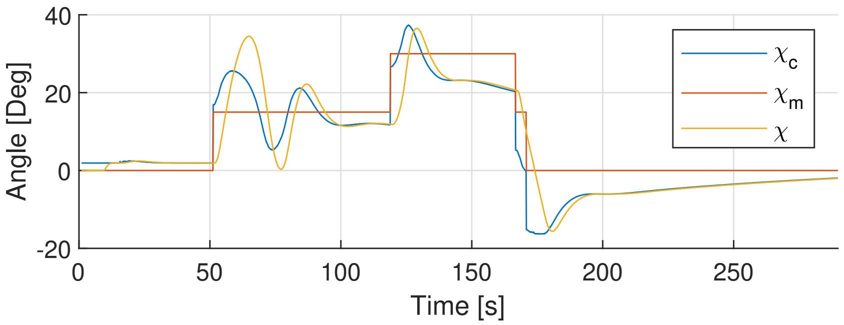
(a) $t = 51s$

(b) $t = 111s$

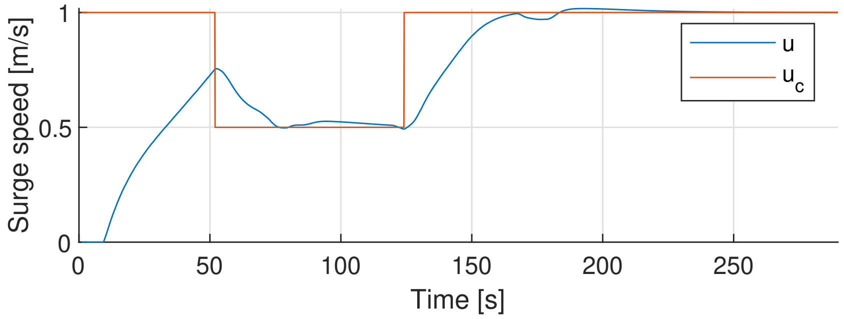
(c) $t = 153s$

(d) $t = 231s$

Figure 7.7: Crossing from port scenario 2 where ReVolt must stop and pass with the obstacle on starboard side. The CPA $= 64.44\text{m}$ and is shown in Figure 7.7c.



(a) Optimal course offset and actual course of ReVolt.



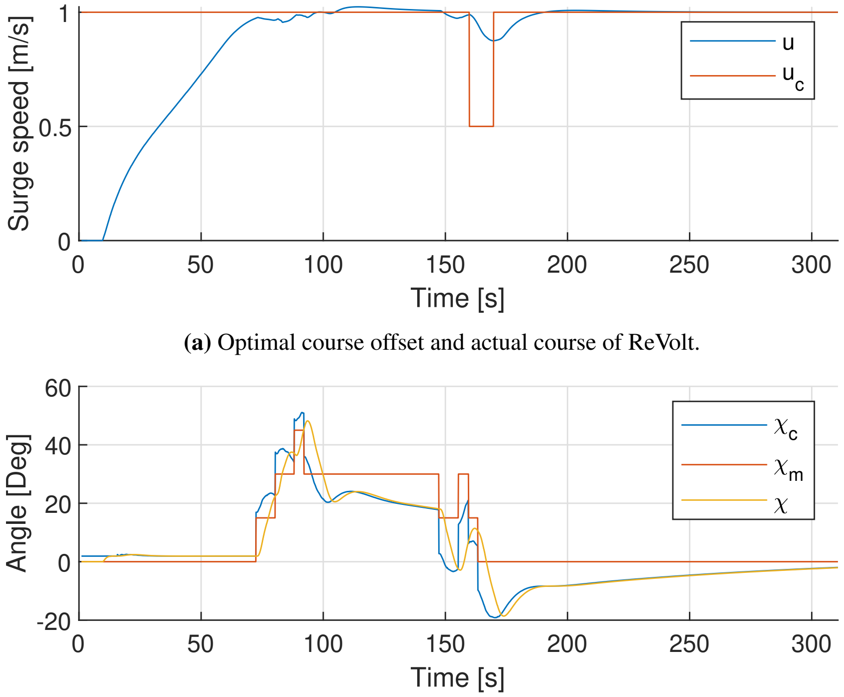
(b) Optimal speed offset and actual speed of ReVolt.

Figure 7.8: Optimal control behaviors in the crossing from port scenario 2. Observe the long duration where the optimal speed offset is $u_m = 0.5$, causing the ownship to slow down and let the obstacle pass.

7.4.3 Crossing from starboard

When the obstacle is crossing from starboard, ReVolt shall keep away while the obstacle can pass freely according to rule 15. Compared to the last scenario, this is a more realistic crossing situation as the keep away vessel can keep away. Thus, the expected maneuver is to let the obstacle pass ReVolt before returning to the nominal path. The COLREGS does not state how the keep away ReVolt should keep away, meaning that a stopping or starboard maneuver is acceptable. However, since the cost of deviating from the desired speed is greater than changing course to starboard it is expected that the latter maneuver is to be performed.

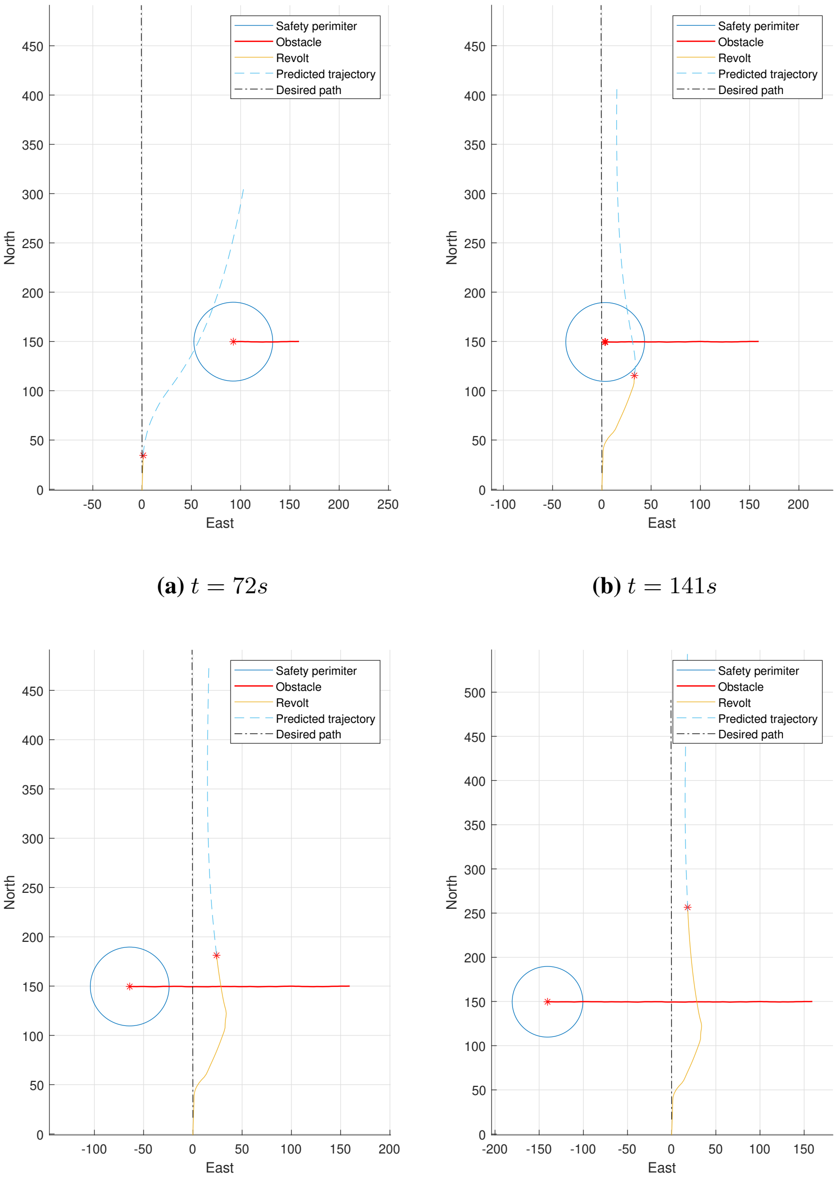
As expected, the starboard maneuver was chosen as the optimal maneuver, as shown in Figure 7.10. With a CPA of $45.11\mathrm{m}$ the CAS performed satisfactory according to Definition 7.2.1 in this scenario as well. The speed was kept constant nearly the whole time except for a 10 second period at about $t = 160\mathrm{s}$, where it slowed down momentarily. Nevertheless, the evasive maneuver is both smooth and efficient as a smaller CPA is almost impossible. In this scenario, there were fewer occurrences of change points being the optimal solution, with only 2 out of 19 SBMPC cycles. These were the first two SBMPC cycles with $n_{cp} = 2$ and 1 respectively in the two cycles.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and actual speed of ReVolt.

Figure 7.9: Optimal course and speed offsets in the crossing from starboard scenario.



(a) $t = 72s$

(b) $t = 141s$

(c) $t = 195s$

(d) $t = 255s$

Figure 7.10: Crossing from starboard scenario where the CPA was $45.11\mathrm{m}$ and is shown in Figure 7.10b.

7.4.4 Overtaking

Next is the overtaking scenario where the obstacle starts $200\mathrm{m}$ north of ReVolt heading northbound with a cruise speed of $0.3\mathrm{m/s}$. The COLREGS rule 13 (see Section 5.1.1) states that the overtaking vessel shall keep out of the way of the overtaken vessel, which in this case means that ReVolt is supposed to make the evasive maneuver while the obstacle maintains its nominal course. The rule does not state which side the overtaking shall take place, meaning that both sides are acceptable. However, as the cost of turning to port is higher than turning to starboard, it is expected that ReVolt shall pass the obstacle on its starboard side.

The results are displayed in Figure 7.12, where the maneuver was performed on the starboard side of the obstacle with a CPA of $44.52\mathrm{m}$. The maneuver was performed in compliance with rule 13 and 16, taking early and substantial action such that the overtaken vessel can keep its course. However, Figure 7.15a shows some oscillatory behavior when passing the obstacle at about $270\mathrm{s} - 310\mathrm{s}$, where the course varies between $10°$ and $20°$. These oscillations should be avoided, and can be viewed as a violation of COLREGS rule 8b. The oscillations probably occur due to the optimal trajectory being tangential to the safety perimeter of the obstacle, whose position varies due to noisy measurements (see Section 7.3). This will be further discussed in Section 7.6. However, the oscillations are not of large magnitude, and the maneuver is not in conflict with the COLREGS, resulting in a satisfactory behavior according to Definition 7.2.1.

The use of change points was prevalent in this scenario, as seen in Figure 7.11. Observe how the use of change points decreases as the vessels approach each other, implying that the use of change points are more efficient at greater distances.

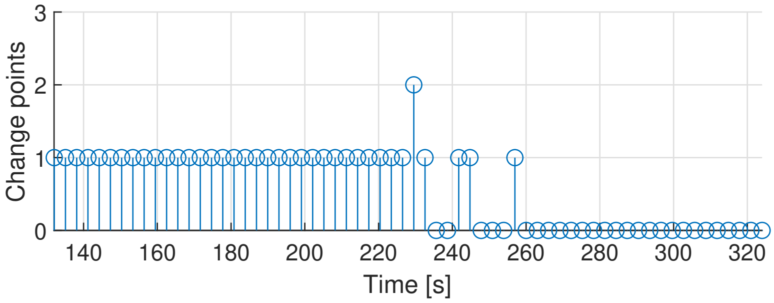
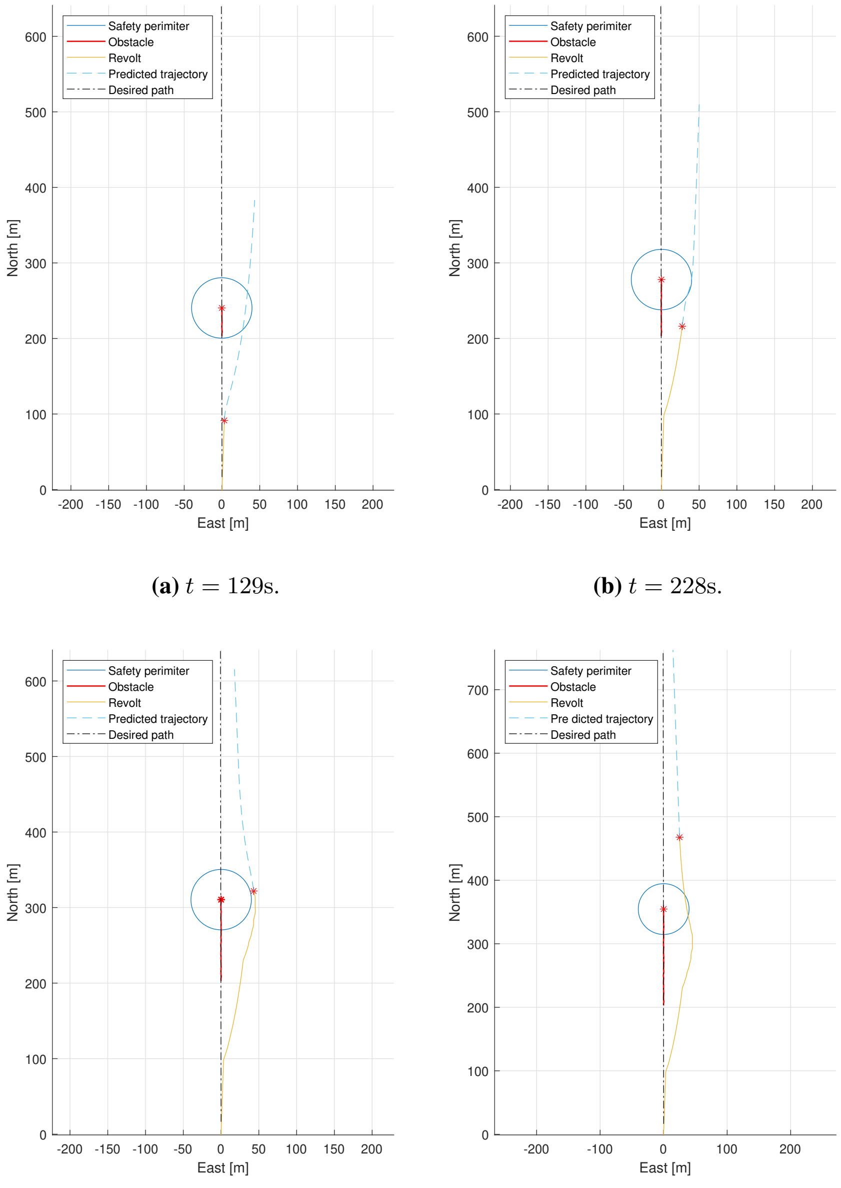


Figure 7.11: Distribution of the use of change points in the overtaking scenario before passing the obstacle. Note that the use of change points decreases when ReVolt approaches the distance d_{safe} to the obstacle.



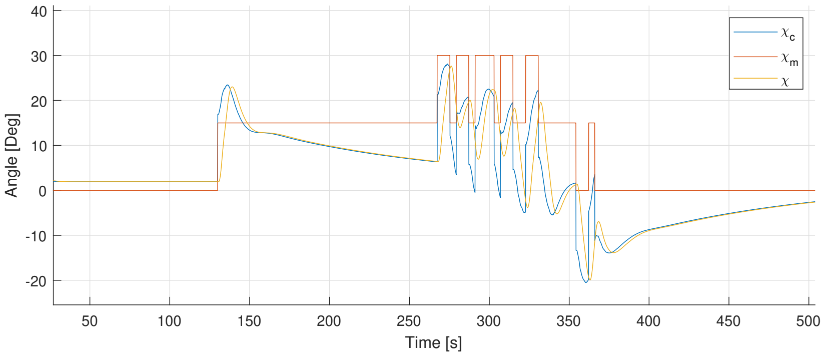
(a) $t = 129\text{s}$.

(b) $t = 228\text{s}$.

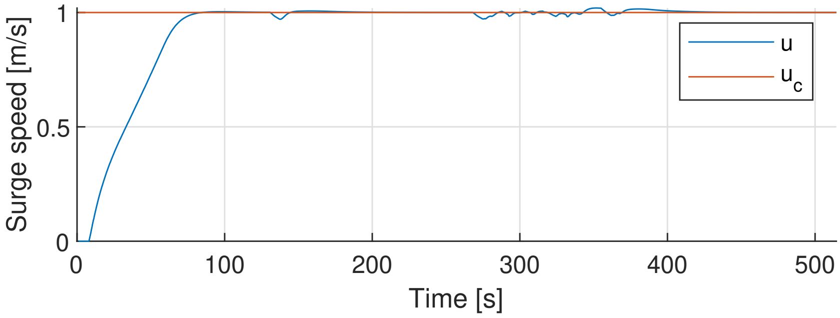
(c) $t = 312\text{s}$.

(d) $t = 426\text{s}$.

Figure 7.12: Course of action in the overtaking scenario. The CPA was 44.52m and is shown in Figure 7.12c.



(a) Optimal course offset and actual course of ReVolt.



(b) Optimal speed offset and speed course of ReVolt.

Figure 7.13: Optimal course and speed offsets in the overtaking scenario.

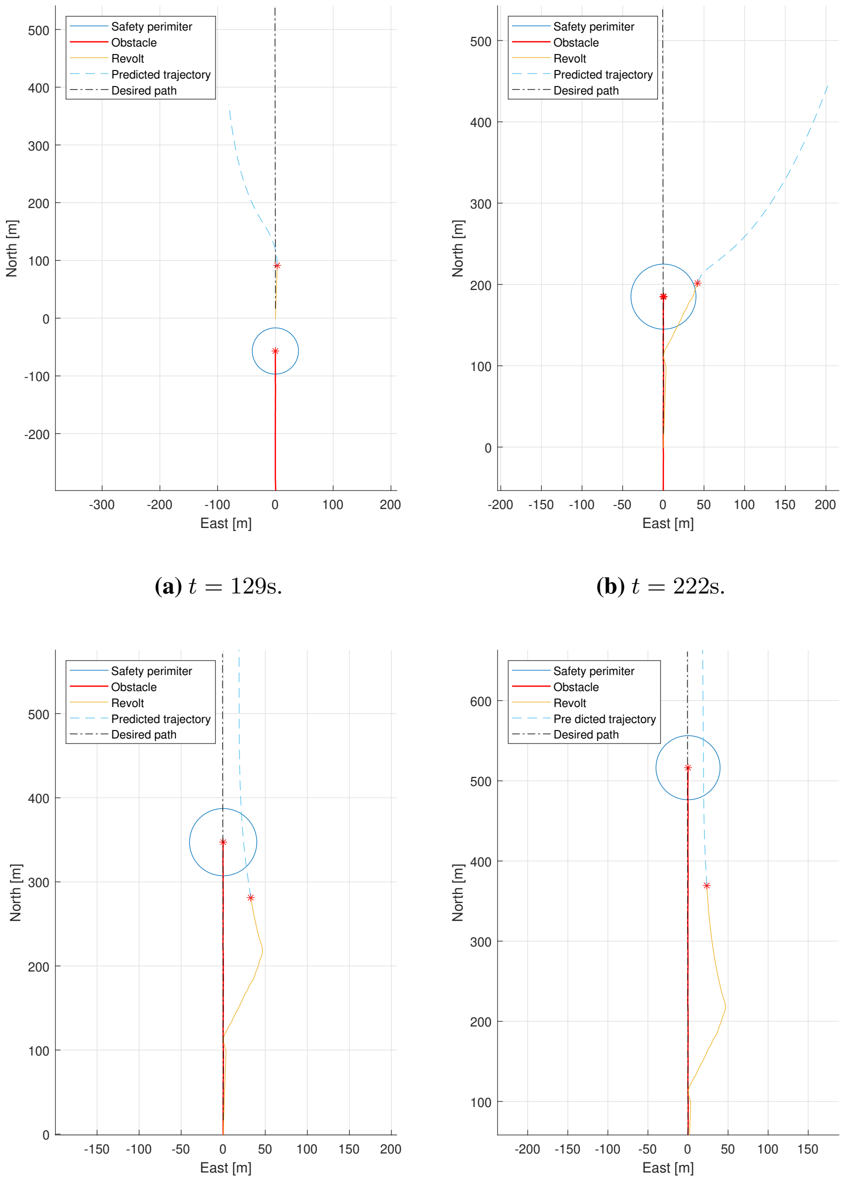
7.4.5 Being overtaken

The last standard scenario is the case where the obstacle is overtaking ReVolt. However, due to the passive nature of the obstacle, ReVolt is forced to take action to avoid a collision. The obstacle starts $300\mathrm{m}$ south of the origin heading northbound with a cruise speed of $2\mathrm{m/s}$. As before, a starboard maneuver is expected since the cost parameters are tuned to favor maneuvers to starboard. However, a slight alteration in the waypoints was added such that ReVolt started with an initial negative cross-track error (deviation on port side). This way, the maneuvering cost would increase when evaluating starboard maneuvers.

The results show that the starboard maneuver was performed with a CPA of $45.13\mathrm{m}$, seen in Figure 7.14. Note that the SBMPC chose a port maneuver as the optimal trajectory in Figure 7.14a, but changed to a starboard maneuver in Figure 7.14b. This can also be seen quite clearly in Figure 7.15a. The maneuver was smooth and efficient where the desired speed was maintained at cruise speed during the whole scenario. As in the overtaking scenario, the use of change points were frequent when the distance between the vessels was large, gradually increasing as they approached each other. The overview of the change points are presented in Table 7.4 below.

Number of change points	#Selected
0	19
1	18
2	4
3	0

Table 7.4: Overview over the use of change points in the being overtaken scenario before the obstacle passed ReVolt.



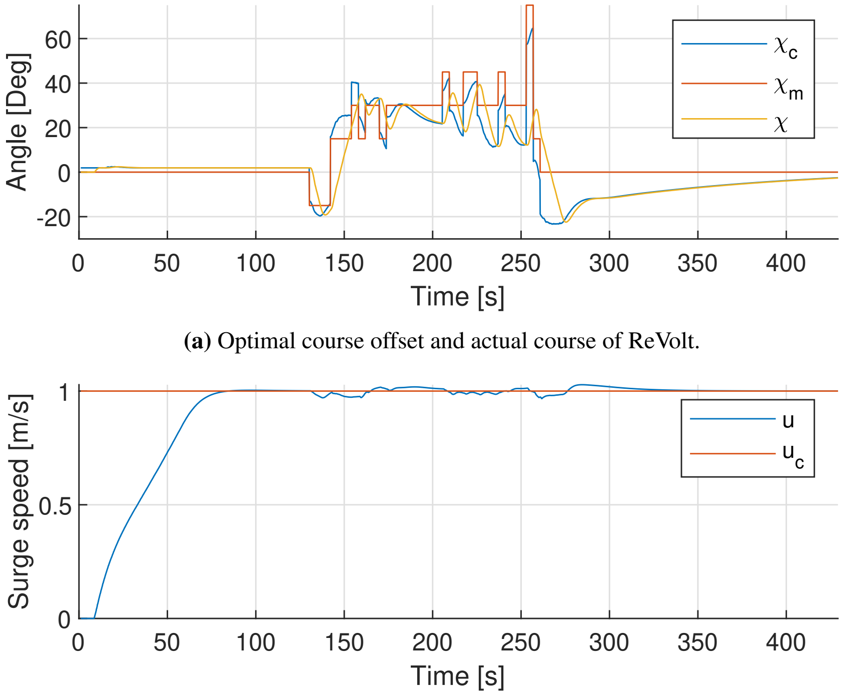
(a) $t = 129$s.

(b) $t = 222$s.

(c) $t = 285$s.

(d) $t = 354$s.

Figure 7.14: Course of action in the overtaken scenario. The CPA was 45.13m and is shown in Figure 7.14b.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and speed course of ReVolt.

Figure 7.15: Optimal course and speed offsets in the overtaken scenario.

7.4.6 Multiple obstacles

The next scenarios will test the SBMPC's ability to handle complex situations where multiple obstacles are involved. The scenarios include three obstacle that approaches ReVolt from different directions such that the three collision situations head-on, crossing from port and starboard occur. Listed below are the initial positions and course for each obstacle in the scenarios. The scenarios are performed in two cases, with and without range restrictions, where the range restriction is decided by d_{init}.

	North	East	Heading
Obstacle 1	250	0	180
Obstacle 2	150	-220	90
Obstacle 3	250	350	-110

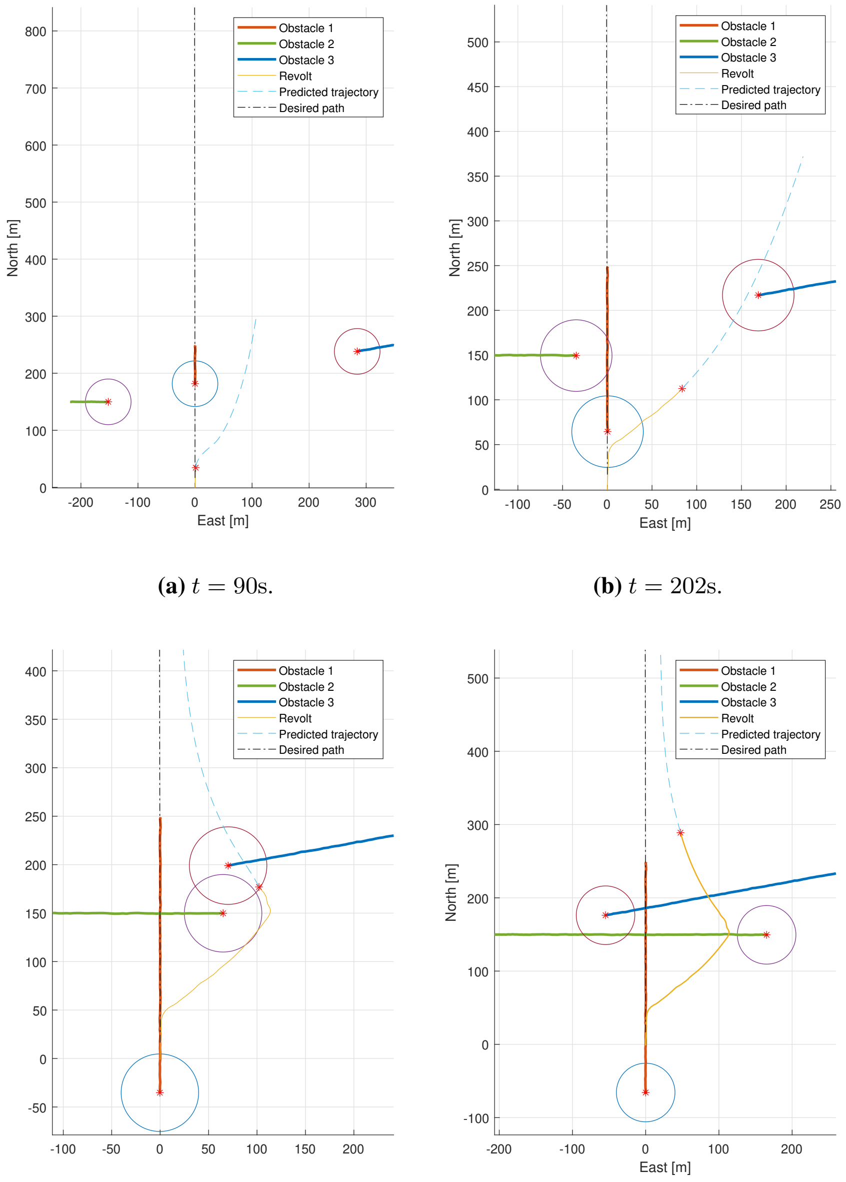
Range restrictions

The results can be seen in Figure 7.16 where the CPA to any obstacle was $38.72\mathrm{m}$, which is not satisfactory according to Definition 7.2.1. As the range of the tracking system is only $150\mathrm{m}$, the only obstacle that is considered in Figure 7.16a is Obstacle 1. Thus, a starboard maneuver similar to the head-on situation in Section 7.4.1 is performed, successfully avoiding the first obstacle, but not considering the other obstacles. Furthermore, Figure 7.16b shows that ReVolt continues the starboard maneuver to cross Obstacle 2 and keep away from Obstacle 3. However, as the initial starboard maneuver did not consider Obstacle 2 and 3 due to range restrictions, ReVolt is not able to avoid entering the safety perimeter of Obstacle 3, resulting in CPA of $38.72\mathrm{m}$, seen in Figure 7.16c. The passing distance is not satisfactory, according to Definition 7.2.1. However, the vessels are moving away from each other at the CPA, causing a minimal hazard.

No range restrictions

The next scenario has no range restrictions and considers all the obstacles when predicting the optimal trajectory. Therefore, it is expected to see a more proactive behavior as all obstacles are taken into consideration in the optimization problem. The course of events are shown in Figure 7.18. In this scenario, the SBMPC chooses to turn more to starboard than before. This allows ReVolt to cross Obstacle 2 with a larger CPA, seen in Figure 7.18c, while avoiding the safety perimeters of both obstacles. During this scenario, the CPA to any obstacle was $44.53\mathrm{m}$, which is classified as a satisfactory behavior.

Even though the number of obstacles increased, the SBMPC's real time capabilities was not significantly affected. In fact, with three obstacles present the average computational time for the SBMPC was $1.05\mathrm{s}$, only $0.1\mathrm{s}$ more than with one obstacle.



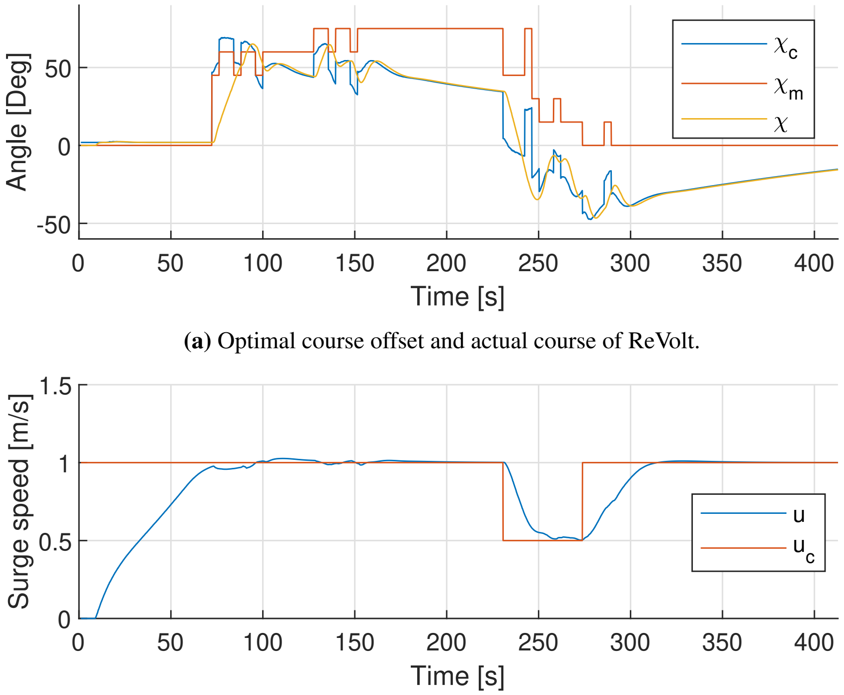
(a) $t = 90\text{s}$.

(b) $t = 202\text{s}$.

(c) $t = 311\text{s}$.

(d) $t = 423\text{s}$.

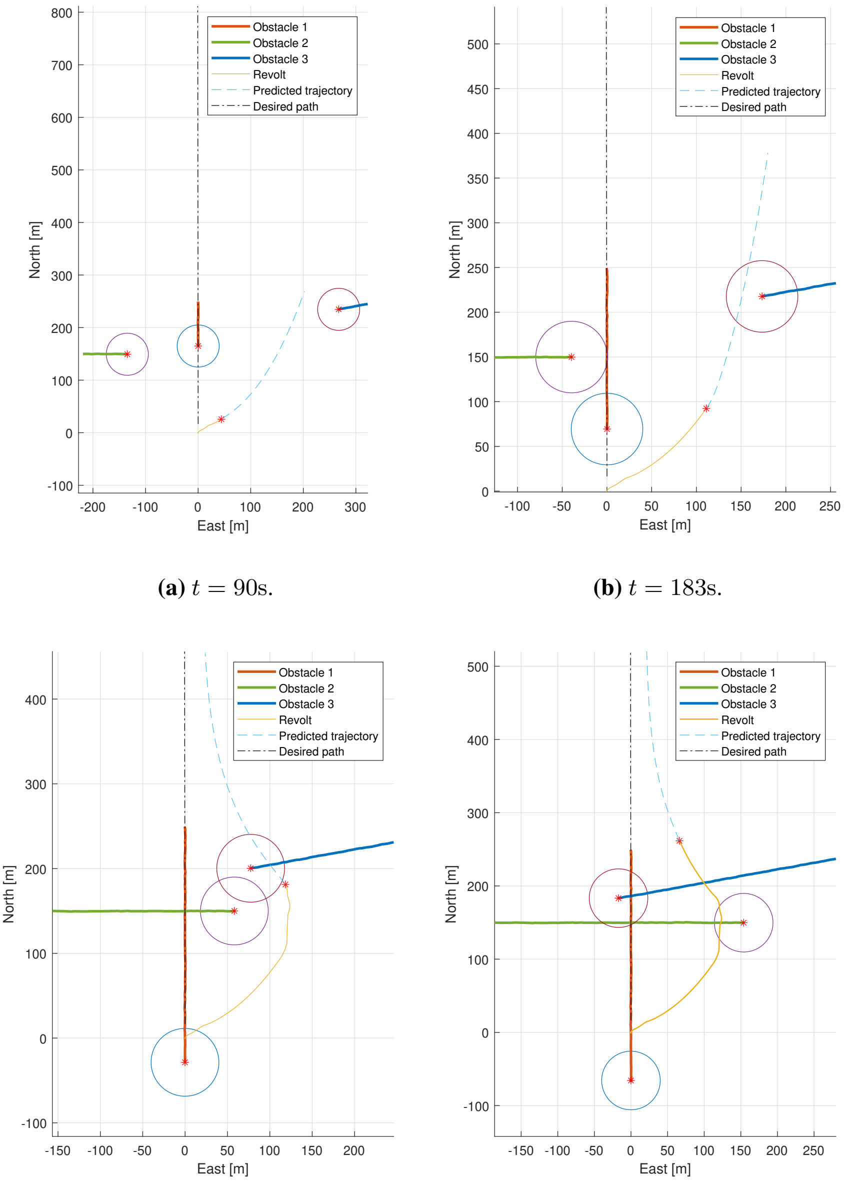
Figure 7.16: Course of action in the multiple obstacles scenario.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and speed course of ReVolt.

Figure 7.17: Optimal course and speed offsets in the multiple obstacles scenario.



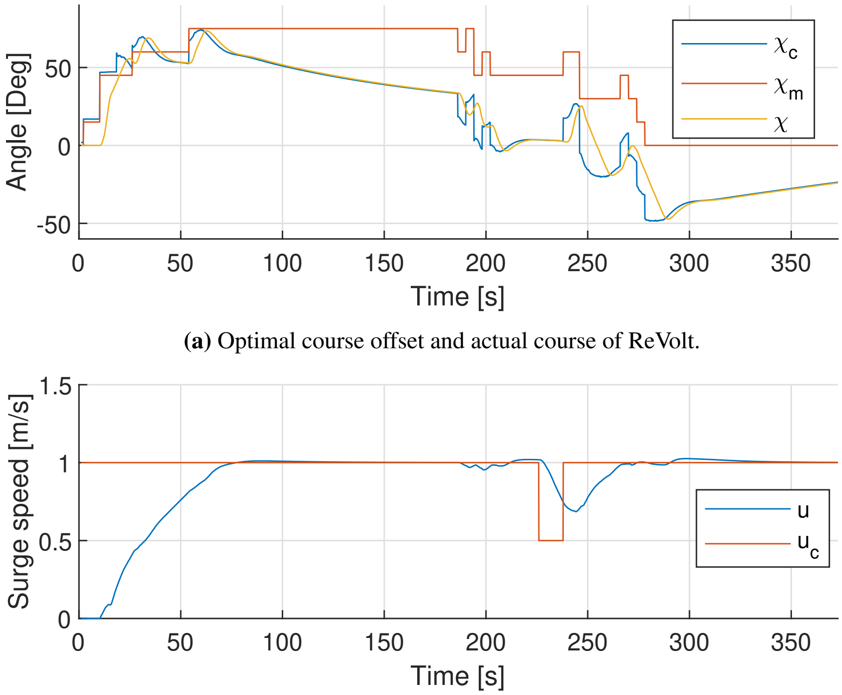
(a) $t = 90\text{s}$.

(b) $t = 183\text{s}$.

(c) $t = 280\text{s}$.

(d) $t = 374\text{s}$.

Figure 7.18: Course of action in the multiple obstacle scenario with no range restrictions.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and speed course of ReVolt.

Figure 7.19: Optimal course and speed offsets in the multiple obstacles scenario.

7.5 Realistic case

The cruise speed during these scenarios in this section were set as low as possible to replicate the slow dynamics of the full scale model. The operating speed of the the full scale model of ReVolt is 6 knots, which constitutes about $0.15\mathrm{m/s}$ for the 1:20 scale model. This operating speed was found to be unfeasible with the current GNC system, and the lowest cruise speed with acceptable behavior was found to be $u_0 = 0.5\mathrm{m/s}$, which is the cruise speed used in this section. This is also the cruise speed of the obstacles.

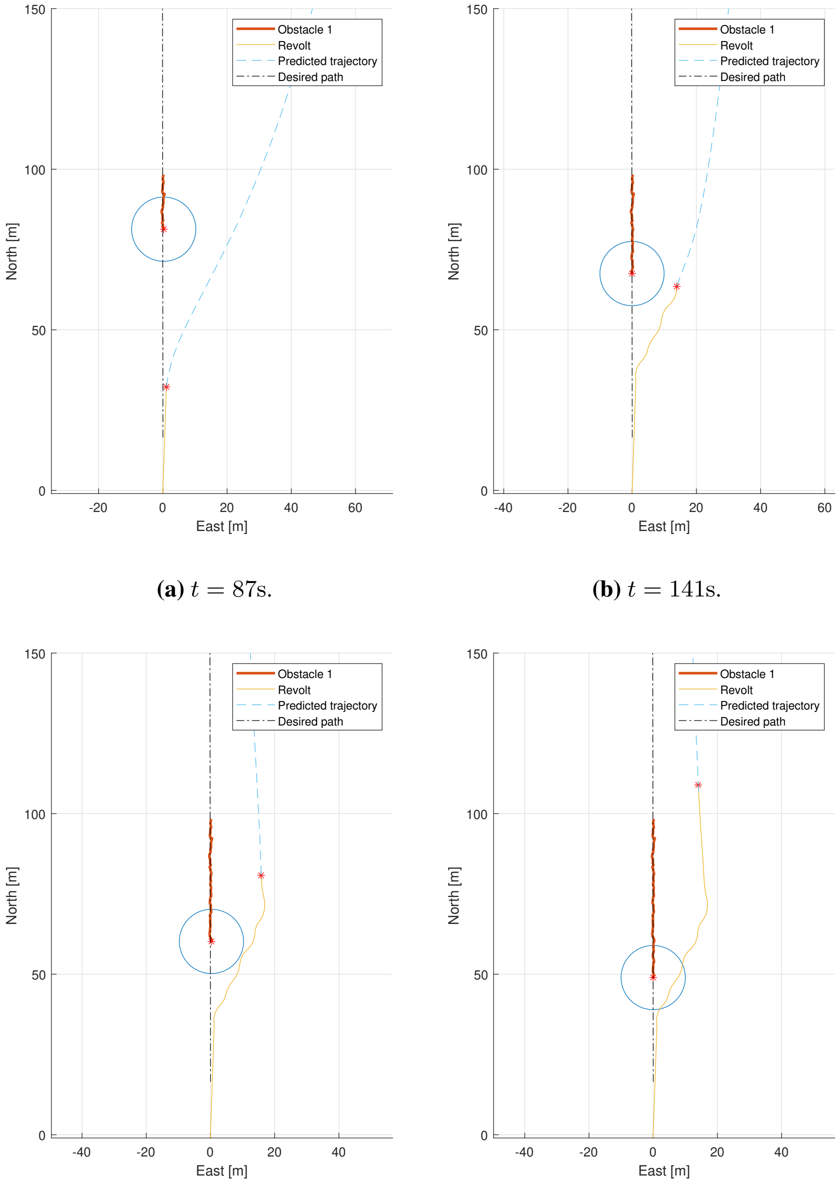
The range of the SBMPC is reduced to replicate the range of the current tracking system on ReVolt, which is $50\mathrm{m}$. As the range reduction is not scaled equally as the cruise speed it is expected that ReVolt is more reactive than in the ideal case. The range reduction also causes the scaling parameter k_{MSE} to be $d_{init}/50 = 1$, resulting in the noise $w \sim \mathcal{N}(0, \sigma_l^2)$ used in the MMKF. The tuning parameters used in the cost function are the same as in Section 7.4 and can be seen in Table 7.1. Note that the safety distance d_{safe} is changed, meaning that the CPA must be larger than $d_{safe} = 10\mathrm{m}$ to satisfy the safety distance requirements of Definition 7.2.1.

It should be noted that the real-time performance of the SBMPC shows similar results as the one presented Section 7.4.1 and is therefore not repeated here.

7.5.1 Head-on

The setup of this head-on scenario is almost identical to the ideal case, but with the obstacle starting at $100\mathrm{m}$ north. The results are shown in Figure 7.20 where the CPA was $14.49\mathrm{m}$, satisfying the safety distance requirements of Definition 7.2.1. However, the maneuver do experience some oscillatory behavior before passing the obstacle, seen in both Figure 7.20b and Figure 7.21a. This is likely to be caused by the GNC operating at a sub-optimal cruise speed, making the heading controller slow. Moreover, the speed is kept constant during the whole scenario, and ReVolt successfully avoids collision.

The use of change points was almost absent at this range, where the only occurrence happened in the first SBMPC cycle. This supports the observation from Section 7.4 that a sufficient range is necessary for change points to be the optimal solution.



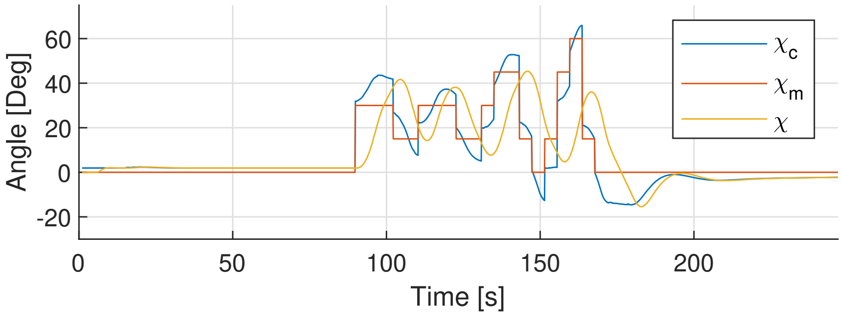
(a) $t = 87\text{s}$.

(b) $t = 141\text{s}$.

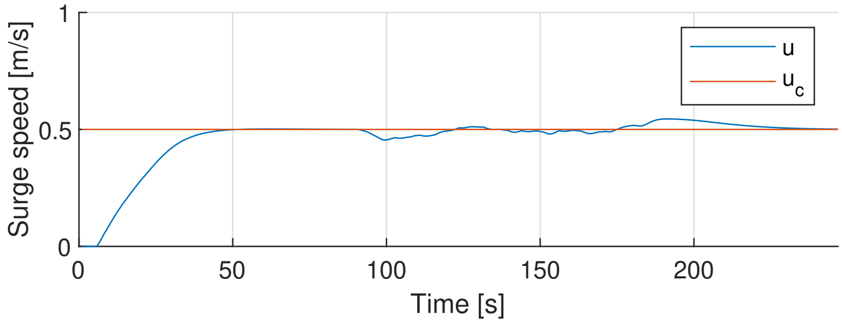
(c) $t = 168\text{s}$.

(d) $t = 210\text{s}$.

Figure 7.20: Course of action in the head-on scenario. The CAS was activated at $t = 87\text{s}$ when $d_{o,1} <= d_{init}$. The CPA was 14.49m and is shown in Figure 7.20b.



(a) Optimal course offset and actual course of ReVolt.



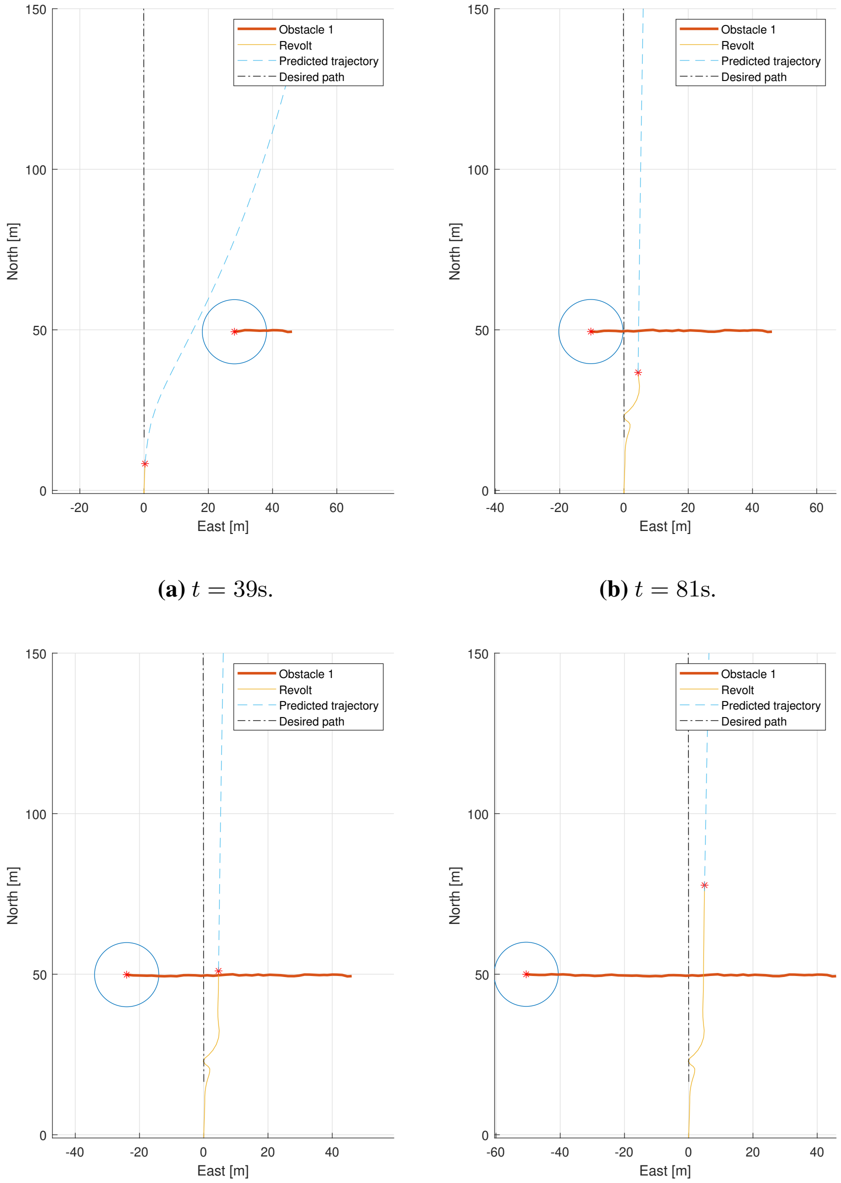
(b) Optimal speed offset and actual speed of ReVolt.

Figure 7.21: Optimal course and speed offsets in the crossing from port scenario 1.

7.5.2 Crossing from starboard

The final scenario is the crossing from starboard scenario where the obstacle started at 50m north and east from the origin heading westbound. The course of action can be seen in Figure 7.22, where ReVolt initially tries to take a starboard maneuver to avoid the obstacle, as expected. However, due to the slow dynamics of ReVolt and the fact at the cruise speed is sub-optimal with regards to the GNC system, the planned and executed maneuver do not coincide. Consequently, a slowing maneuver is performed as ReVolt was not able to maneuver itself to starboard in time. Moreover, ReVolt continues to wait patiently until the obstacle has passed before it returns to its nominal path. Although not optimal, the maneuver complies with the COLREGS rules, taking action in ample time while keeping away from the crossing obstacle. The CPA was at 19.44m, seen in Figure 7.22b, and the maneuver is therefore deemed satisfactory according to Definition 7.2.1.

As in the head-on scenario, the use of change points as the optimal solution occurred only in the first SBMPC cycle, continuing to strengthen the claim that the use change points are more frequent at greater distances.



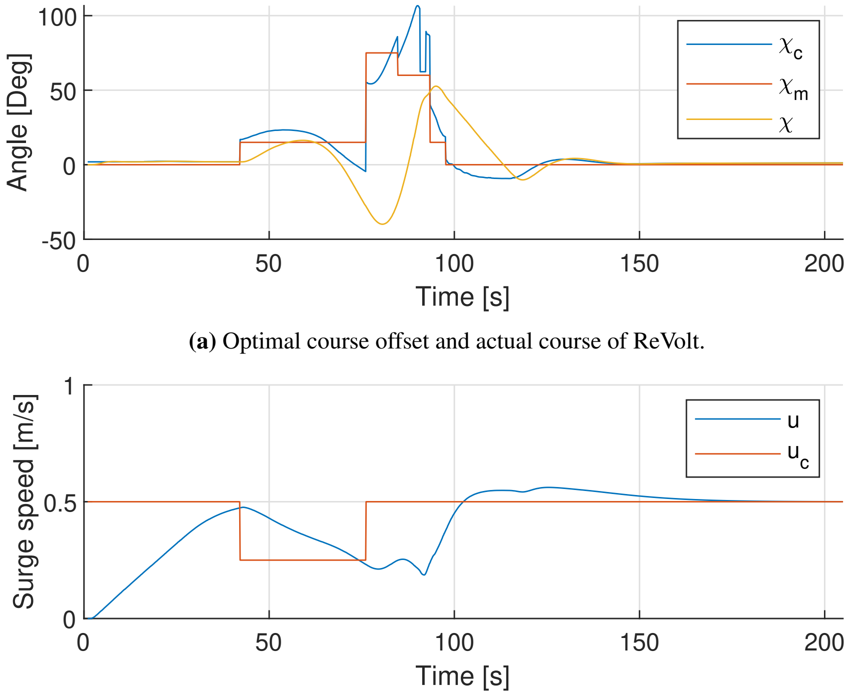
(a) $t = 39$s.

(b) $t = 81$s.

(c) $t = 120$s.

(d) $t = 162$s.

Figure 7.22: Course of action in the head-on scenario. The CAS was activated at $t = 39$s when $d_{o,1} <= d_{init}$. The CPA was 19.44m and is shown in Figure 7.22b.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and actual speed of ReVolt.

Figure 7.23: Optimal course and speed offsets in the crossing from starboard scenario.

7.6 Discussion

7.6.1 Tracking

The tracking system performed well in the simulation scenarios, showing close to ground truth estimates with satisfying real-time capabilities. Even though the AEE was below 0.5m in both north and east position, one cannot conclude how well the tracker performed using only this metric. One has to look at the worst-case scenarios and the consequences of that scenario to evaluate the track system in a COLAV situation. Using the results from Section 7.3, the worst-case scenario occurs when the maximum error occurs in both north and east at the same time, meaning that the obstacle is $\sqrt{(1.1210\text{m})^2 + (0.9657\text{m})^2} \approx 1.48\text{m}$ closer than estimated. When seen in a collision avoidance context where the safety perimeter $d_{safe} = 40\text{m}$, an error of this magnitude would constitute no real threat. Therefore it is reasonable to conclude that the MMKF performs satisfactorily for COLAV purposes in the simulation study with regards to the estimation results.

Moreover, the MMKF completes its calculations within 0.1s, a delay that is almost negligible at low speeds considering the vessels' size. If the obstacle were to travel at higher speeds, e.g., 10m/s, this delay would constitute an error of 1m, which would not be the determining factor in a collision situation. However, this does not mean that the delay should be ignored due to its low individual impact, but rather prioritized to a smaller extent than other contributing factors.

One of these factors could be the interpolation performed when the CAS requests tracks at a specified timestamp, and interpolation is performed to estimate the state at the given timestamp. Recall from Section 7.3 that the MMKF uses a fixed update rate of 1s and performs dead reckoning when no new measurements have been received. Then there is a possibility that the CAS requests tracks immediately before the MMKF has performed its latest computation, causing a delay of about 1s in the latest estimate. The accuracy of the estimate then depends on the interpolation performed. From the results in Section 7.3, one may argue that the interpolation performs well on average, given the tested scenarios, moving according to the CV model. If the obstacle were to perform a turning maneuver, the tracking system would likely perform significantly worse. One solution to this could be to implement an (IMM) that considers multiple motion models to describe the motion of an obstacle. This solution would require high computational power to perform satisfactorily in a real-time situation, as it is more computationally expensive.

The MMKF shows great consistency with 96% of the measurements being inside the 95%-confidence interval, which is expected as the MMKF is tuned with regards to the contaminated noise. Regardless, the tracking system performs satisfactorily in the simulated scenarios.

7.6.2 Collision avoidance

The simulation study shows that the CAS using the SBMPC algorithm avoids collision in all conducted scenarios, performing satisfactory results in a variety of realistic scenarios, including single and multiple obstacles. The new addition of a time-varying heading reference for ReVolt throughout the prediction horizon has improved the predicted trajectories compared to the former implementation, shown in [5]. As seen in Figure 7.4, the predicted trajectories considers the actual path of ReVolt, resulting in a curved and smooth trajectory compared to the former implementation in [5, Figure 3.2]. The addition of the tracking system with noisy measurements gives the simulations a more realistic touch. Regardless, the simulation environment may be improved using other simulation platforms as Gazebo, where a multitude of exteroceptive sensors may be used, including camera and LiDAR. This could drastically accelerate the development of a fully working sensor fusion tracking system using a camera and LiDAR.

Evaluating the use of change points

All of the scenarios in Section 7.4 showed a prevalent use of change points as the optimal solution. In order to quantify the effect of change points in the evasive maneuvers, the same scenarios are performed without the use of change points. It is worth mentioning that these maneuvers also performed satisfactory, according to Definition 7.2.1.

The results are presented in Figure 7.24 and Figure 7.25 where one can see that the use of change points makes the trajectory of ReVolt more smooth and efficient. Observe that the trajectories in the head-on scenario in Figure 7.24a are quite similar in shape, but the trajectories with no change points oscillate more and have a larger maximum cross-track error. These small adjustments are undesirable and should be avoided according to COLREGS rule 8, implying that the use of change points improves the COLREGS compliant maneuver. Additionally, the small adjustments accumulate and causes a relatively large positional difference of $36.87\mathrm{m}$ when compared at the same timestamp. This is also the case in the crossing from starboard scenario from Figure 7.24d and the overtaking scenario from Figure 7.25a. One may argue that the maneuver using no change points in Figure 7.25a complies more with COLREGS rule 16 as the maneuver turns more to starboard at an earlier time, signaling the indented maneuver. However, the maneuver performed using change points makes the starboard maneuver simultaneously, but chooses a less steep starboard maneuver. To the author's understanding of the COLREGS, both maneuvers take early and substantial action where the obstacle is passed at a safe distance, with the difference being that the use of change points proves to be the more efficient one.

The greatest difference is in the crossing from port scenarios, plotted in Figure 7.24b and Figure 7.24c, where the positional difference compared at the same timestamp are $84.47\mathrm{m}$ and $94.69\mathrm{m}$ apart respectively. These distances may not seem significant, but one has to remember that these scenarios are performed with a cruise speed of $1\mathrm{m/s}$, whereas the full-scale model is planned to operate at a cruise speed of $6\mathrm{m/s}$, meaning that the distances can be scaled up by 6. In the full-scale scenario, the difference between the two methods is thus much more significant. It is important to choose the most effective method, i.e., with change points, to minimize the required time and operation costs of the evasive maneuver.

Furthermore, the use of change points is prominent at large distances between the vessels. This is to be expected as the use of change points expands the maneuvering space which in turn increases the possibility to find a more optimal trajectory.

The use of change points increases the number of scenarios to be evaluated, which also increases the computational time. Without using change points the average computational time with one obstacle was $0.3884\mathrm{s}$, which is expected as the number of scenarios to be evaluated increases with a factor of $n_{cp} = 3$. The increase is without a doubt significant and causes further delay in the tracking to collision avoidance pipeline.

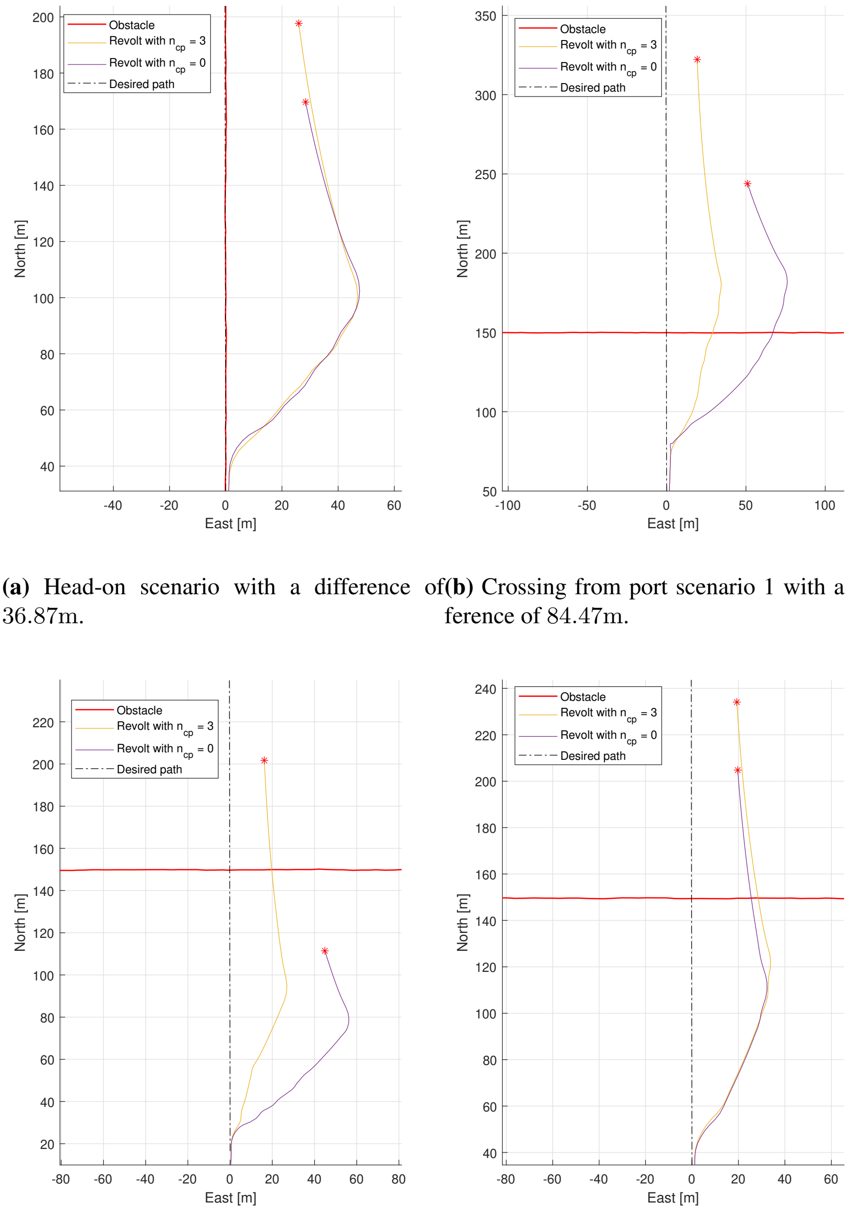
Evaluation of the Ideal case

The CAS performance has been somewhat discussed in the previous section, but a more detailed evaluation of the performed scenarios is needed.

The SBMPC performed satisfactory in all of the presented scenarios in Section 7.4, where the evasive maneuver never had a CPA closer than the permitted safety distance of d_{safe}. Nevertheless, an important factor when evaluating the maneuvers is to verify the maneuver's ability to comply with the COLREGS. The maneuvers began momentarily as the obstacle was within $d_{init} = 150\mathrm{m}$ of ReVolt, implying that rule 8 and 16 were followed. Additionally, the SBMPC chose the correct maneuver according to the COLREGS in each of the individual scenarios, including the crossing from port scenario where obstacle was the keep away vessel, complying with rule 17.

The overtaking scenario experienced some oscillation when passing the obstacle, which is undesirable and could potentially be a violation of rule 8b. The reason is most likely due to the optimal trajectory being located tangential to the safety perimeter of the obstacle, whose position varies due to noisy measurements. The predicted trajectory will therefor enter the safety perimeter, which causes a significant change in cost, and results in a change of course offset. It is not obvious where the line is drawn for which alterations in course are acceptable, as the alterations need to be seen in context. In this case, the alterations occurs when ReVolt is directly starboard of the obstacle, where the oscillations vary between $10° - 20°$. From the obstacle's point of view, the oscillations are hardly noticeable and cause little to no confusion for the intent of the maneuver, which implies that the maneuver complies with rule 8.

Scenarios involving multiple obstacles are difficult to determine the correct maneuver, but due to the tuning of the cost function parameters in Table 7.1, it is expected that starboard maneuvers are expected. An interesting observation is to see the impact of the range restrictions Section 7.4.6, where the maneuvers were seemingly similar but differed in performance. In the case with range restrictions, not all obstacles were considered when performing the evasive maneuver, and the maneuver began at a later stage, giving the SBMPC less time to react to the situation. This, in combination with the slow response of the heading controller, leads to an unsatisfac-



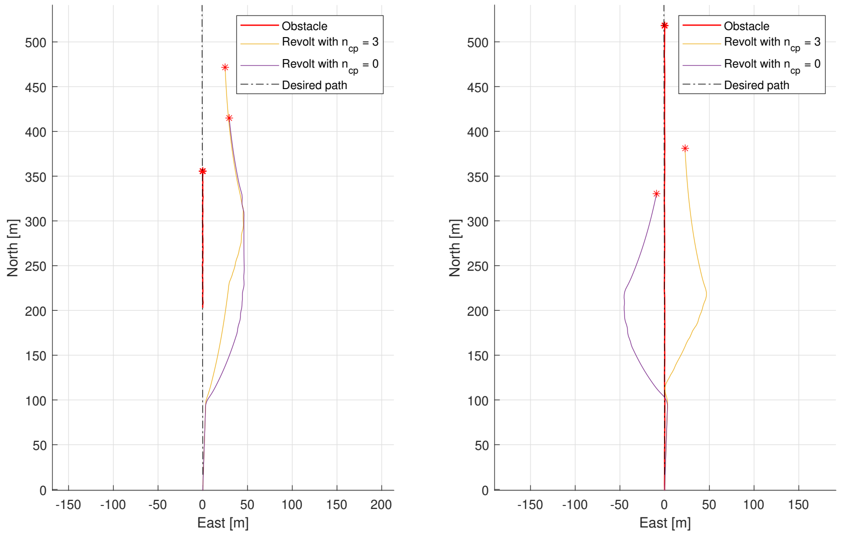
(a) Head-on scenario with a difference of $36.87\mathrm{m}$.

(b) Crossing from port scenario 1 with a difference of $84.47\mathrm{m}$.

(c) Crossing from port scenario 2 scenario with a difference of $94.69\mathrm{m}$.

(d) Crossing from starboard scenario with a difference of $29.31\mathrm{m}$ after.

Figure 7.24: Course of action in the overtaken scenario. The CPA was $45.13\mathrm{m}$ and is shown in Figure 7.14b.



(a) Overtaking scenario with a difference of $56.83\mathrm{m}$.

(b) Being overtaken with a difference of $59.97\mathrm{m}$.

Figure 7.25: Course of action in the overtaken scenario. The CPA was $45.13\mathrm{m}$ and is shown in Figure 7.14b.

tory result. In the case with no range restrictions, earlier action was taken where the need for fine maneuvering was minimal.

Realistic case

The scenarios in the realistic case suffered even more from the slow response of the heading controller. The sub-optimal performance of the GNC causes significant oscillations in the range of $10° - 45°$ in the head-on scenario. Compared to the same scenario in Section 7.4.1, one can observe that the oscillatory behavior, in this case, is more rapid and that the course error is larger. From a COLREGS perspective, this behavior is undesirable as COLREGS rule 8b states that the evasive maneuver should be apparent to another vessel, where a succession of small alterations in the course and speed should be avoided.

A solution to this could be to introduce gain scheduling, such that a desirable response is achieved at different cruise speeds. One might also consider the use of the bow thruster at lower speeds to increase maneuverability. The difference between the desired and performed maneuvers implies a mismatch between the models used in the SBMPC and GNC system. This could be fixed by constraining the yaw rate with the maximum yaw rate of ReVolt, and by tuning the controllers used in the SBMPC (see Section 5.3).

Another prominent behavior occurred during the crossing from starboard scenario where the heading controller was unable to perform at speeds close to zero.

In other words, the speed controller dominates the heading controller, causing lower maneuverability. This problem could also be solved using the bow thruster, such that both the speed and heading references are met.

Chapter 8

Experimental sea trials

In this chapter, results from the conducted experimental sea trials will be presented. First, a short introduction to the setup of the experiment is presented. Then, the tracking results will be given. Next, results from the closed-loop collision avoidance system will be presented. Finally, the results of this chapter will be discussed. To the author's knowledge, this is the first time real-life experiments have been conducted using a LiDAR-based tracking system in a closed-loop CAS.

8.1 Setup

The experimental sea trials were performed in Dorabassenget in Trondheim (See Figure 8.1) to minimize the effects of environmental forces such as waves and ocean currents. ReVolt is especially vulnerable to waves due to its small size, which could cause high frequent roll motions that affect the performance of the tracking system. However, current and wind forces going from northeast to southwest were observed during the sea trials, meaning that forward and starboard maneuvers were more challenging to perform.

The calm sea conditions do, however, come with a price. The urban environment causes the amount of reflected LiDAR points to increase substantially. The land masking (see Section 3.1) algorithms remove most of the land points, but not all. Additionally, a ferry and a tugboat were docked in Dorabassenget, seen in Figure 8.3, causing a significant amount of clutter measurements.

The target boat used in this thesis was Fjøset II, seen in Figure 8.2, and was mounted with a GNSS antenna to get ground truth data from the target. The GNSS data were post-processed using RTKlib [57], giving millimeter precision to the target boat's position. Fjøset II will be referred to as the *obstacle* from now on,

During the experiments, issues with the camera prevented it from being used in the tracking system. Hence, only the LiDAR was used to update the state estimates.

Due to time restrictions, only two scenarios were conducted, namely the head-on and crossing from starboard scenario. These scenarios were prioritized as the expected evasive maneuvers are predictable, making it easy to interpret the results. The scenarios were planned to resemble the scenarios in Section 7.5 as much as possible with ReVolt starting in the origin heading northbound. In the head-on scenario, the obstacle started at approximately $90\mathrm{m}$ north heading southbound. Correspondingly, in the starboard scenario, the obstacle started at approximately $30\mathrm{m}$ north and $50\mathrm{m}$ east heading westbound.

The main parameters used for the IPDA and SBMPC are shown in Table 8.2 and Table 8.1 respectively. Optimally one would like to test multiple configurations of the tracking parameters, but due to the circumstances, only one configuration was tested during the sea trials, seen in Table 8.2. The process noise covariance was set relatively low as the obstacle's motion was planned to conform with the CV-model in both scenarios. For the SBMPC, the same parameters from Section 7.5 were originally used, but as the heading controller did not perform as expected during preliminary testing of the GNC, some adjustments had to be made. The cruise speed of ReVolt was increased to $u_0 = 1\mathrm{m/s}$, as that is the cruise speed the heading controller was tuned for. Additionally the SBMPC update period was increased to $t_{period} = 5\mathrm{s}$ to reduce the number of changes in course reference.

SBMPC parameters	Value
T [s]	300
DT [s]	0.5
t_{period}[s]	5
d_{init}[m]	50
d_{close}[m]	50
d_{safe}[m]	10

Table 8.1: The main SBMPC parameters used during the experimental sea trials. Here, the same values were used for the cost function tuning parameters as in table 7.1.

Tracking parameters	Value
Survival probability	1.0
Init probability	0.2
Confirmation threshold	0.99
Termination threshold	0.1
Process noise covariance	0.1
Gate probability	0.99
Detection probability	0.9
LiDAR measurement covariance	$0.5\mathrm{m}^2$

Table 8.2: The main tracking parameters for IPDA used in the experimental sea trials.

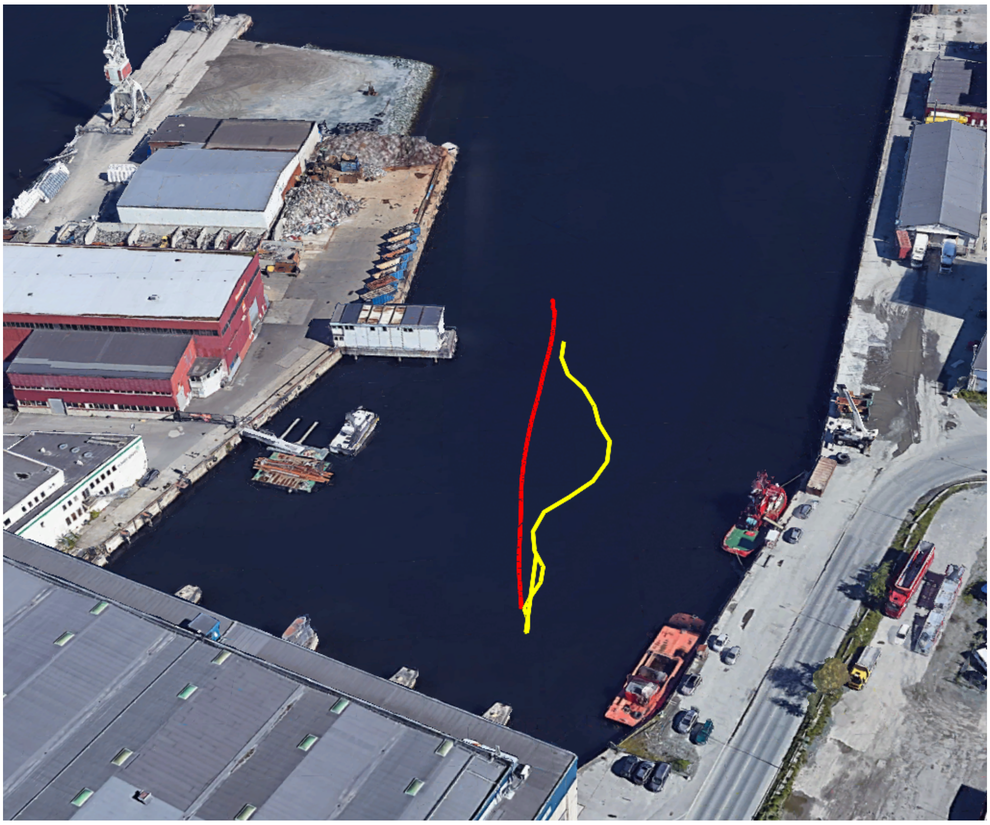


Figure 8.1: 3D photo of Dorabassenget where the experiments were conducted. The red and yellow line indicate the path of the obstacle and ReVolt respectively in the head-on scenario. The photo was obtained using Google Earth Pro and GNSS data from the experiment.

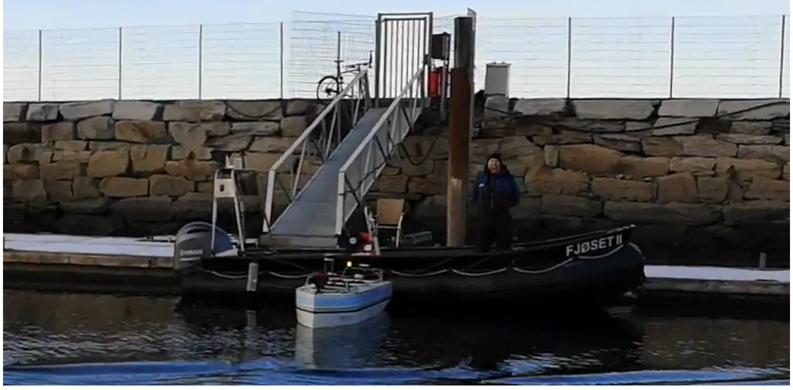
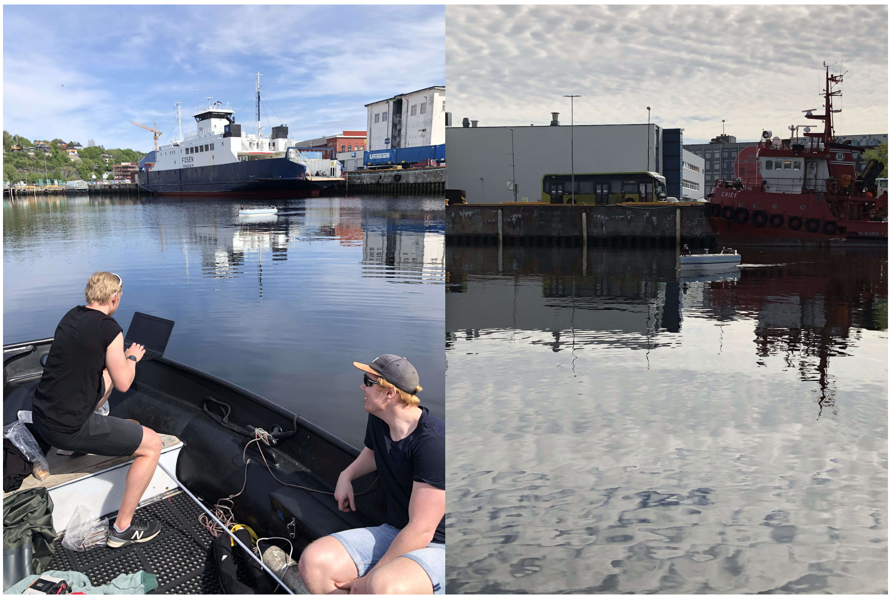


Figure 8.2: The target boat used as the passive obstacle in the experiments. The picture was taken during the experiments conducted in the author's project thesis [1].



(a) The ferry and ReVolt, with Simen Overeng to the left and the author to the right.

(b) The red tugboat in the background with ReVolt sailing in front.

Figure 8.3: A ferry and a tugboat were doecked in Dorabassenget during the sea trials, causing a substantially amount of reflected points from the LiDAR not removed by land masking.

8.2 Tracking

This section will first present the tracking results from both scenarios. When presenting the tracking results the color codes in Table 8.3 will be used to describe the plots. Here, the color of LiDAR centroids will gradually fade to black as time passes.

Red	Ground truth trajectory of the obstacle
Green	Tracks from the tracking system
Yellow	Trajectory of ReVolt
Grey/Black	LiDAR centroids from DBSCAN
Blue	Land border

Table 8.3: Description of the color codes used in the oncoming tracking plots. The land border is collected from Kartverket [58].

8.2.1 Results

The average track initiation time in both scenarios is presented in table 8.4 and shows promising results. Once a track is initiated, the average time used from measurement acquisition to tracking update is below $0.1\mathrm{s}$, causing a very low delay in the pipeline.

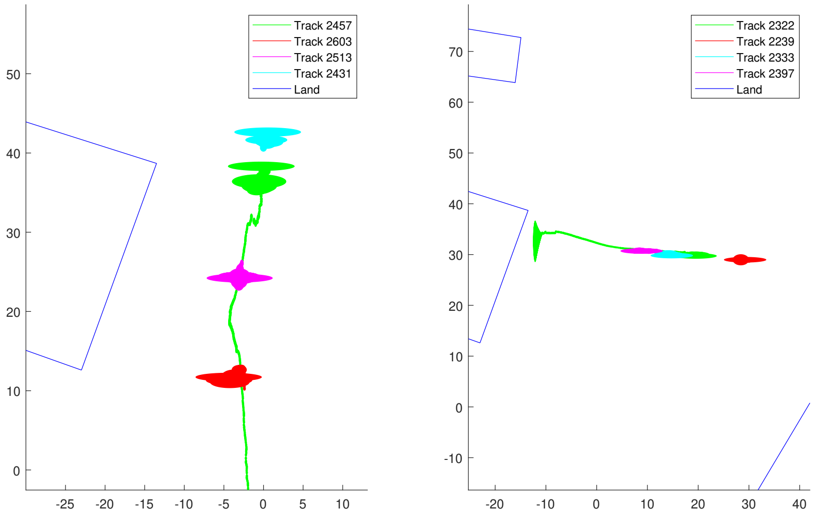
Track initiation head-on [s]	0.4031
Track initiation crossing from starboard [s]	0.3341
Acquisition to track time [s]	0.0765

Table 8.4: The average initiation time using IPDA and the average time used from the acquisition of LiDAR measurement to the estimated track.

The plots in Figure 8.5 and Figure 8.6 shows how the majority of the received LiDAR measurements and tracks originates from the ferry, located at about $100\mathrm{m}$ north and $70\mathrm{m}$ east, as well as the tugboat at about $35\mathrm{m}$ east of the origin. Apart from that, there are few occurrences of clutter, indicating that the land masking is working as expected.

In both scenarios, one can observe that the position of ReVolt conforms with estimated tracks. This is because the new LiDAR setup is closer to the surface of the ReVolt, causing more points originating from ReVolt to be reflected and subsequently tracked. This problem can be handled when processing the LiDAR points, but the problem was first discovered during the sea trial. These tracks disrupted the COLAV decisions in preliminary tests, resulting in the choice of ignoring all obstacles closer than $3\mathrm{m}$ by the SBMPC. This quick fix is sub-optimal and causes an unnecessary computational delay in the pipeline.

It was found that of the 91 established tracks during the head-on scenario, only 12 tracks originated from the obstacle. However, of the 12 tracks, only four were alive long enough to be used in the SBMPC, and only one survived over multiple SBMPC cycles, lasting about $50\mathrm{s}$. Similar results were found in the crossing from starboard scenario where only eight tracks were target originated, and only one survived multiple SBMPC cycles, with a total duration of 80s. One reason for the quick termination of the obstacle originated tracks may be their large covariances, which is visualized by in Figure 8.4. Here, only four tracks in each scenario are visualized for demonstration purposes, where the green tracks are the longest surviving ones. It can be observed that the covariances of the green tracks converge faster to a small magnitude than the non-green tracks, which may cause the existence probability to drop below the termination threshold for said tracks.



(a) Plot of covariance ellipses in the head-on scenario.

(b) Plot of covariance ellipses in the crossing from starboard scenario.

Figure 8.4: Covariance ellipses centered around the estimated position of the tracks. Only a selection of tracks were chosen for visualization purposes, where all were target originated.

Furthermore, it can be observed from Figure 8.5 and Figure 8.6 how the ground truth conforms with both the LiDAR centroids and the established tracks. It can be observed that the estimated and ground truth positions of the obstacle do not conform perfectly. However, the error may not necessarily be as large in reality as the estimated position is based on the LiDAR centroid, while the ground truth position is measured from the GNSS antenna, where the two points may not necessarily match. This will be discussed in more detail in Section 8.4. The calculated position errors in north and east can be seen in Figure 8.7a and Figure 8.7b, with the error metrics listed in Table 8.5a and Table 7.2. Here, the long-lasting tracks described above are the tracks that are compared with ground truth. It can be observed that errors are generally low and do not suffer from spikes, except at the end of the track in the crossing from starboard scenario.

Surge speed and heading estimates for both scenarios are shown in Figure 8.8 and Figure 8.9 respectively, and compared with ground truth. It can be observed that

Metric	Value
$\text{RMSE}_N[\text{m}^2]$	1.3323
$\text{RMSE}_E[\text{m}^2]$	1.1876
$\text{AEE}_N[\text{m}]$	1.1577
$\text{AEE}_E[\text{m}]$	1.0076

(a) Head-on scenario.

Metric	Value
$\text{RMSE}_N[\text{m}^2]$	1.6479
$\text{RMSE}_E[\text{m}^2]$	2.7413
$\text{AEE}_N[\text{m}]$	0.9842
$\text{AEE}_E[\text{m}]$	1.6473

(b) Crossing from starboard scenario.

Table 8.5: Track metric results from the position estimates.

the estimates from the head-on scenario were very accurate overall, where the largest errors occurred in the beginning of the tracks as expected. The estimates from the crossing from starboard scenario showed similar tendencies, but suffered from larger errors at the end of the track. This can also be seen in Section 8.2.1, where both the RMSE_ψ and AEE_ψ were very large. This may be due to the obstacle being very close to land, and will be discussed further in Section 8.4. It should be mentioned that the ground truth surge speed and heading were calculated using the position data in conjunction with basic equation of motions and trigonometry.

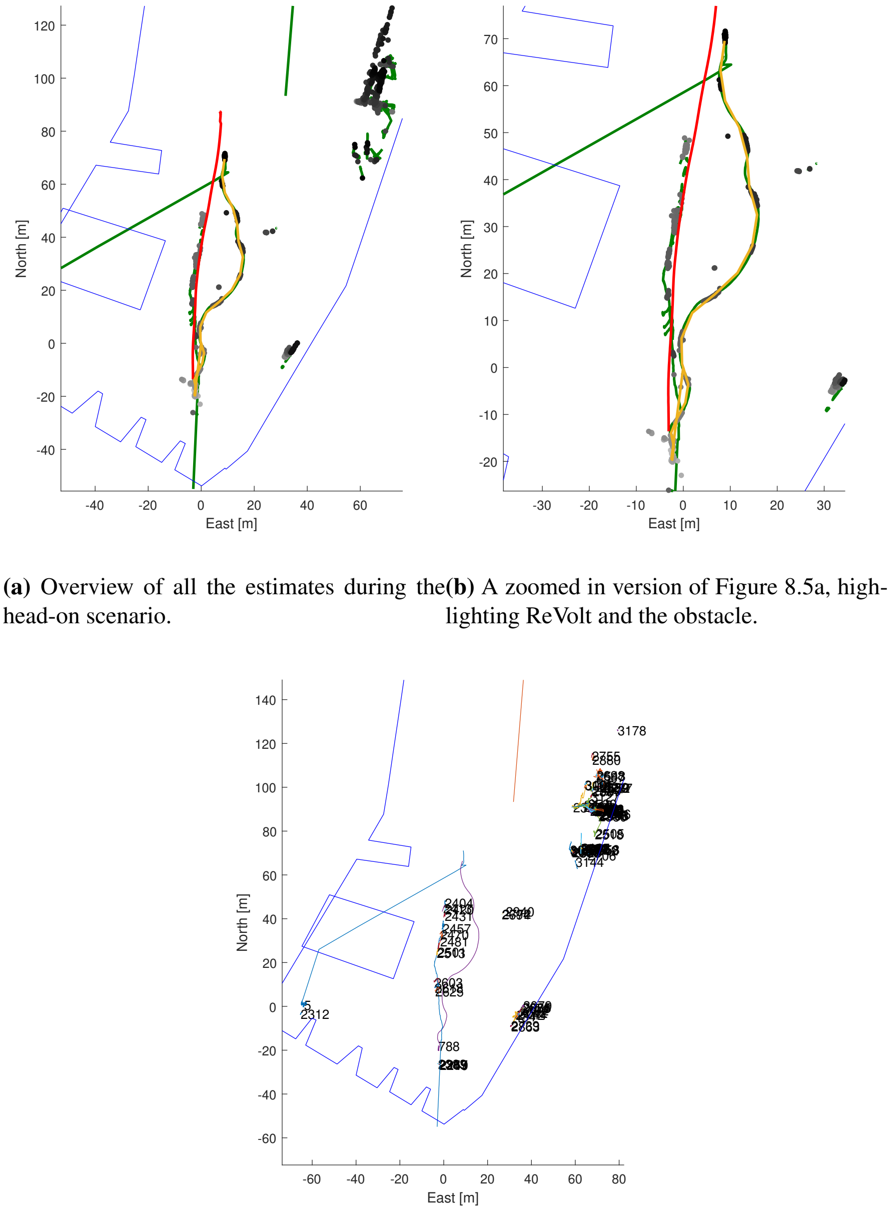
Metric	Value
$\text{RMSE}_U[\text{m}^2/\text{s}^2]$	0.5320
$\text{RMSE}_\psi[\text{deg}^2]$	9.4643
$\text{AEE}_U[\text{m}/\text{s}]$	0.2167
$\text{AEE}_\psi[\text{deg}]$	7.9922

(a) Head-on scenario.

Metric	Value
$\text{RMSE}_U[\text{m}^2/\text{s}^2]$	0.2014
$\text{RMSE}_\psi[\text{deg}^2]$	48.7292
$\text{AEE}_U[\text{m}/\text{s}]$	0.1258
$\text{AEE}_\psi[\text{deg}^2]$	23.0718

(b) Crossing from starboard scenario.

Table 8.6: Track metric results from the surge speed and heading errors.

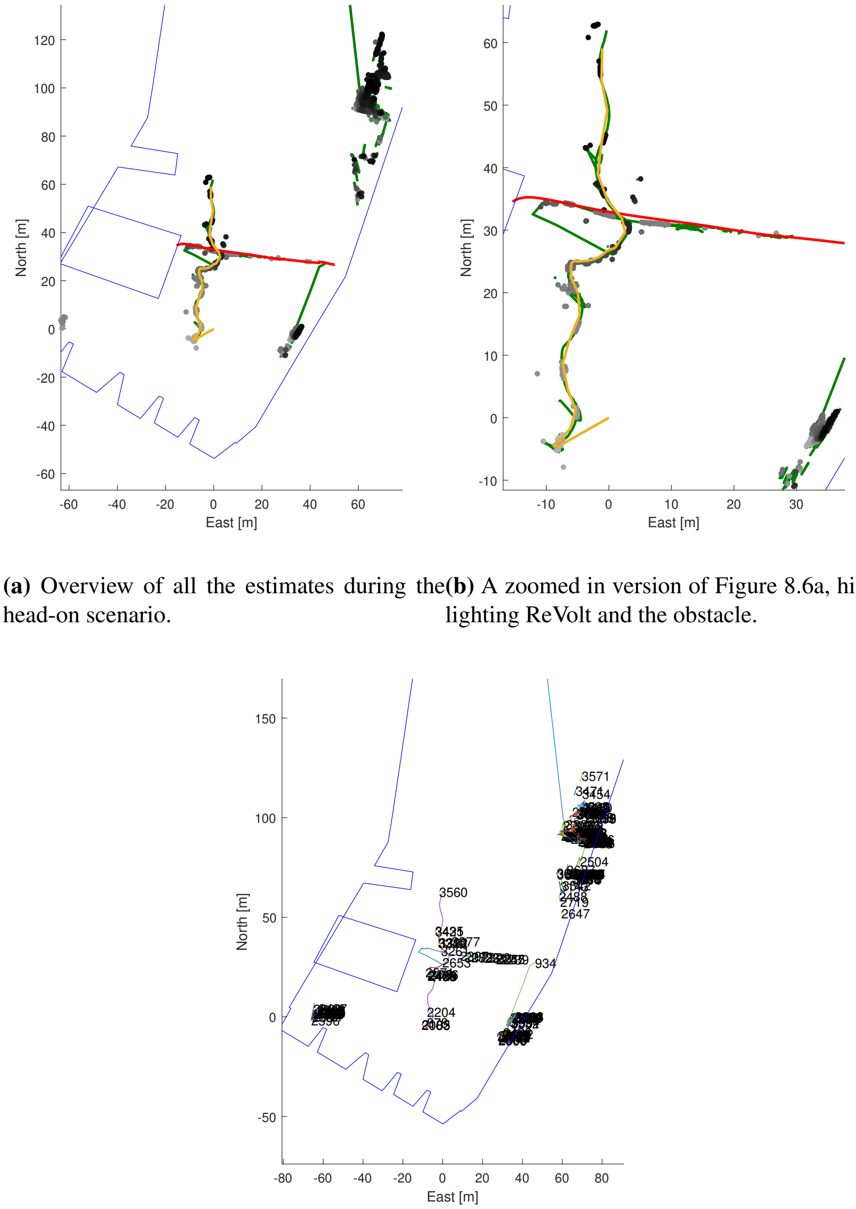


(a) Overview of all the estimates during the head-on scenario.

(b) A zoomed in version of Figure 8.5a, highlighting ReVolt and the obstacle.

(c) Plot of tracks only, where the number represents the track id.

Figure 8.5: Track results from the head-on scenario compared with the ground truth position of the obstacle. The tracks are plotted in the same color for visualization purposes.

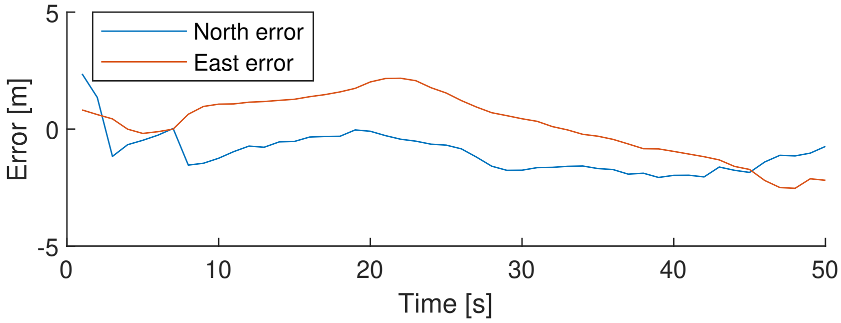


(a) Overview of all the estimates during the head-on scenario.

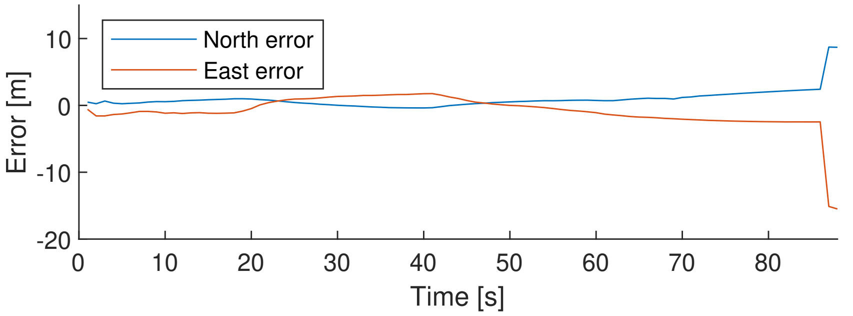
(b) A zoomed in version of Figure 8.6a, highlighting ReVolt and the obstacle.

(c) Plot of tracks only, where the number represents the track id.

Figure 8.6: Track results from the crossing from starboard scenario compared with the ground truth position of the obstacle. The tracks are plotted in the same color for visualization purposes.

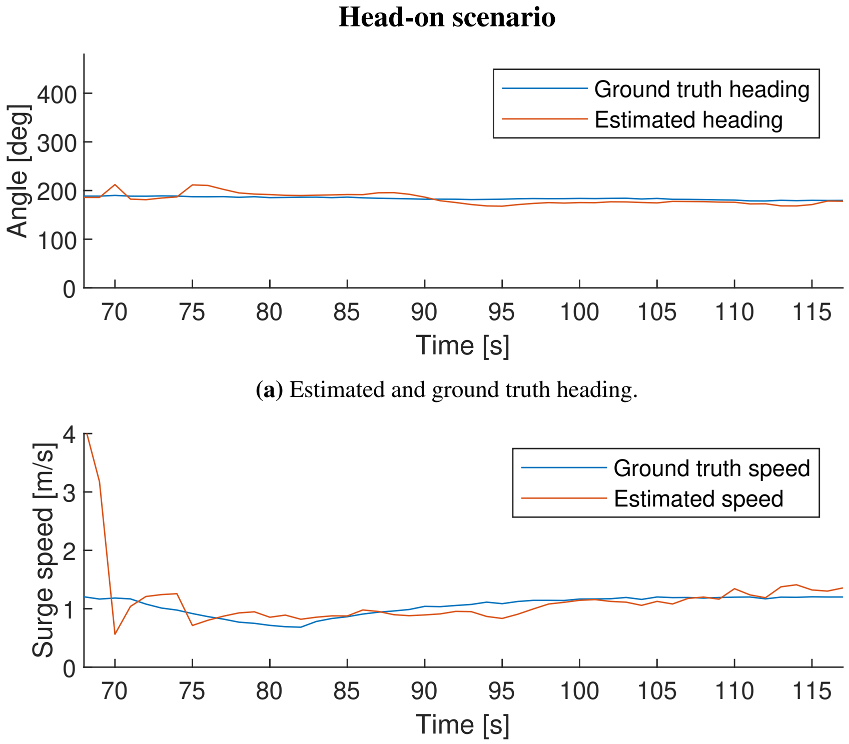


(a) Errors the crossing from the head-on scenario. Note how the errors are relatively smooth, with the major spikes occurring in start of the track.



(b) Errors the crossing from starboard scenario. The errors stay smooth with few spikes until the end of the track.

Figure 8.7: Difference between the obstacle track and the ground truth. However, when analyzing the error one have to consider that the track is based on the position of the LiDAR centroid, while the ground truth is based on the GNSS antenna mounted in the middle of the obstacle. The offset between these two will vary over time and is difficult to estimate.

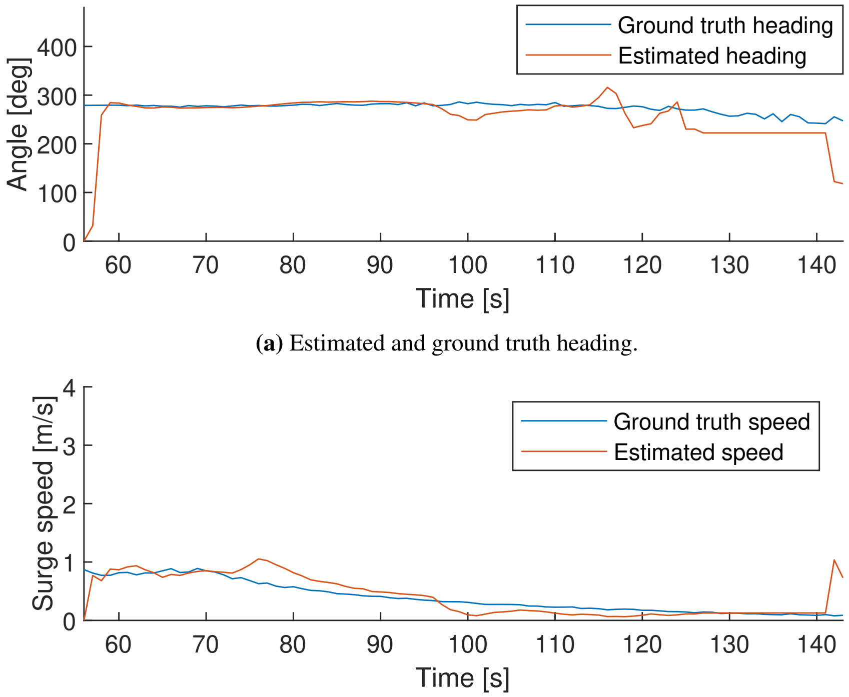


(a) Estimated and ground truth heading.

(b) Estimated and ground truth surge speed.

Figure 8.8: Plot of estimated vs ground truth heading and speed from the head-on scenario. The estimates are retrieved from the longest living track.

Crossing from starboard scenario



(a) Estimated and ground truth heading.

(b) Estimated and ground truth surge speed.

Figure 8.9: Plot of estimated vs ground truth heading and speed from the head-on scenario. The estimates are retrieved from the longest living track.

8.3 Collision Avoidance

In this section, the COLAV results will be presented, where the results from each scenario will be presented separately. The plots are presented in the same format as in Chapter 7, and obstacle information is retrieved from the long-lasting tracks, explained in Section 8.2.1.

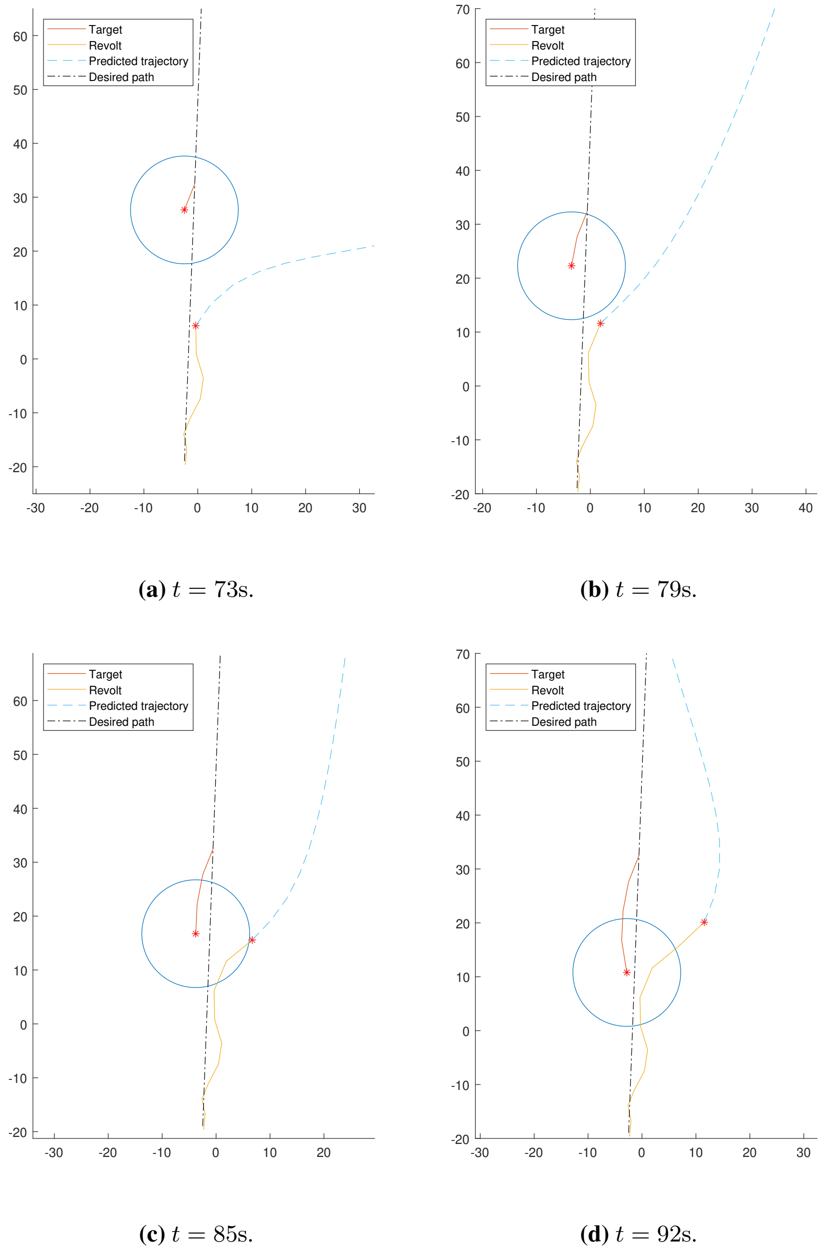
	Average # of obstacles	Average computational time
Head-on	6.90	1.25s
Crossing from starboard	7.55	1.26s

Table 8.7: Table showing the real-time performance of the SBMPC during both scenarios.

8.3.1 Results - Head-on scenario

The course of action in the head-on scenario can be seen in Figure 8.10, showing promising results. The evasive maneuver is taken in compliance with the COLREGS (see Figure 5.2) where a starboard maneuver is performed, leaving the obstacle at port side with a CPA of $10.45\mathrm{m}$. This is large enough to satisfy the CPA requirement of Definition 7.2.1, as $d_{safe} = 10\mathrm{m}$. One may argue that the maneuver is not taken in ample time (COLREGS rule 8a), but the action is taken at the earliest time possible, given the range restrictions of the tracking system. Hence, the maneuver complies with the COLREGS to its best ability, and the evasive maneuver is deemed satisfactory according to Definition 7.2.1.

Figure 8.11 shows how the GNC system struggles to follow the reference at lower speeds, as in Section 7.5.1. The performance is also affected by the environmental forces, causing even lower maneuverability. This forces the SBMPC to choose higher course offsets compared to the same scenario in the simulation study (see Section 7.5.1), in order to achieve a satisfactory CPA. The speed controller shows similar performance in the simulations where it uses a long time to reach the reference but maintains a smooth behavior during the whole scenario.



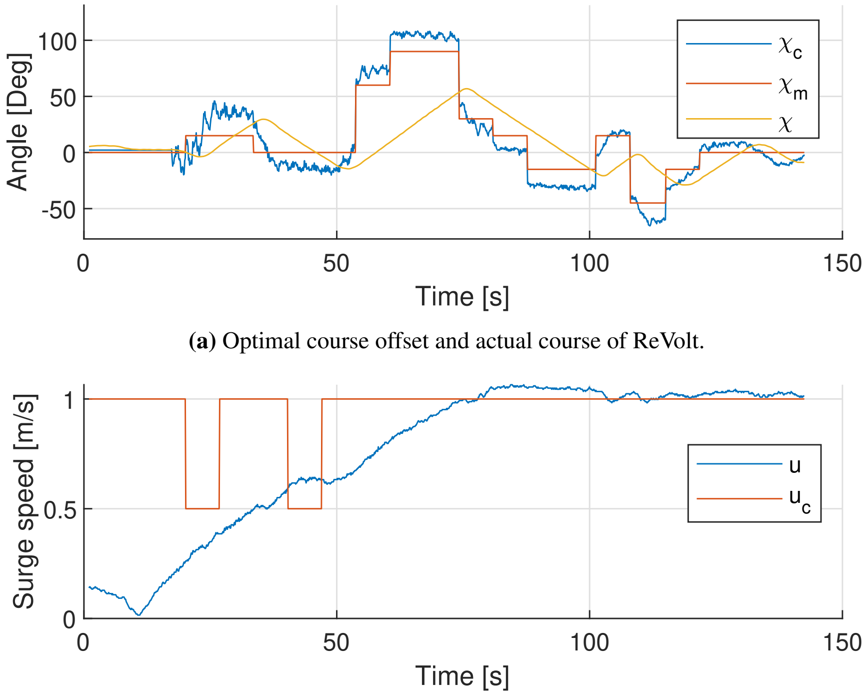
(a) $t = 73$s.

(b) $t = 79$s.

(c) $t = 85$s.

(d) $t = 92$s.

Figure 8.10: Course of action in the head-on scenario during the experimental sea trial . The CPA was 10.45m and is shown in Figure 8.10b.



(a) Optimal course offset and actual course of ReVolt.

(b) Optimal speed offset and speed course of ReVolt.

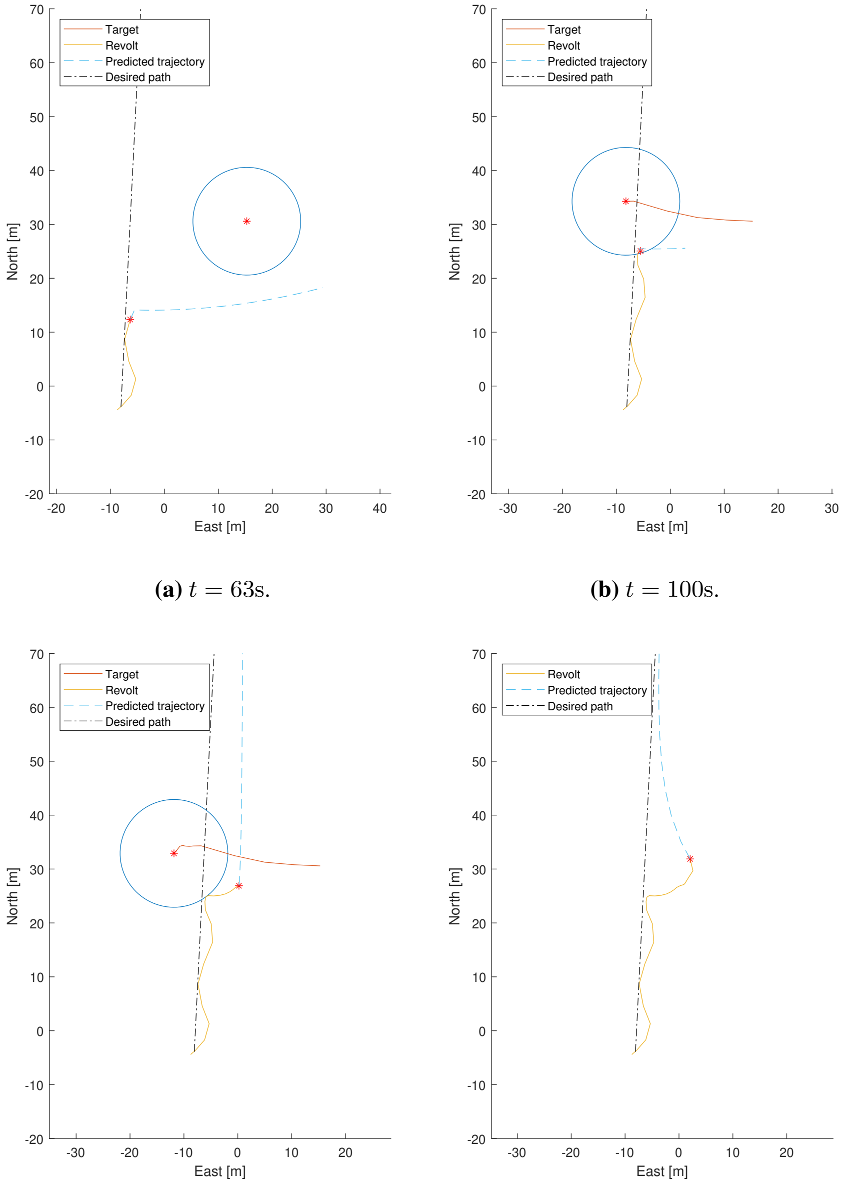
Figure 8.11: Optimal course and speed offsets in the head-on scenario.

8.3.2 Results - Crossing from starboard

The second scenario was the crossing from starboard scenario, where the course of action can be seen in Figure 8.12, with a CPA of 9.58m. This does not satisfy the safety distance requirement of Definition 7.2.1 and is not a satisfactory maneuver. However, from Figure 8.12 and Figure 8.13 one can observe that the SBMPC intends to slow down and make a starboard maneuver, but the speed controller does not slow down fast enough, causing ReVolt to drift towards the obstacle. The low speed restricts the maneuverability of ReVolt, which in turn slows and delays the starboard maneuver with about 80s. This can be seen by comparing Figure 8.12a and Figure 8.12c. After turning to starboard and passing the obstacle, ReVolt returns to the nominal path.

As mentioned in Section 8.2.1 the track of the obstacle is lost due to land masking, seen in Figure 8.12d. This was not desirable, but this did not affect the maneuver to a large extent as the obstacle would have continued westbound away from ReVolt, while ReVolt would return to its nominal path.

The execution of the evasive maneuver was not satisfactory according to Definition 7.2.1, but shows promise nevertheless. The actions are taken as early as possible with the intent of keeping away, in compliance with the COLREGS. However, the execution of the maneuver does not comply with COLREGS rule 8d) as ReVolt does not pass at a safe distance.



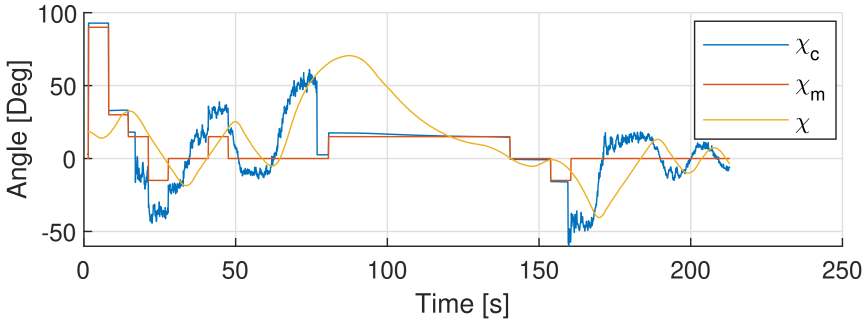
(a) $t = 63\text{s}$.

(b) $t = 100\text{s}$.

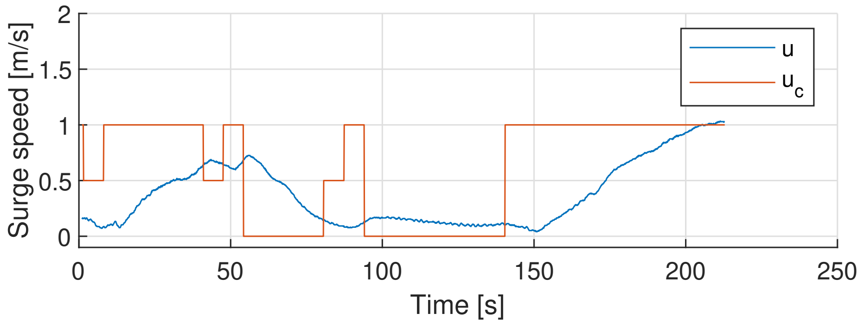
(c) $t = 144\text{s}$.

(d) $t = 175\text{s}$.

Figure 8.12: Course of action in the head-on scenario during the experimental sea trial . The CPA was 9.58m and is shown in Figure 8.12b. The track was lost at $t = 145\text{s}$ due to land masking as the obstacle was too close to the pier.



(a) Optimal course offset and actual course of ReVolt.



(b) Optimal speed offset and speed course of ReVolt.

Figure 8.13: Optimal course and speed offsets in the head-on scenario.

8.4 Discussion

8.4.1 Tracking system

The tracking system shows great real-time capabilities with low initiation time and processing time from measurement acquisition to track update. The performance of the IPDA is a great improvement when compared to the IPDA in Norbye's master thesis [4], where the lowest initiation time was 24.36s. It should be noted that the high initiation time of the IPDA in [4] was probably to be due to a bug in the implementation where the wrong state vector was used but was fixed in this thesis. It is also an improvement compared to the best performance of the PDAF, which was 0.728s. The average processing time from measurement acquisition to tracking update was 0.0765s, which is faster than the LiDAR point cloud frequency of 10Hz. This result is satisfactory.

The results in Figure 8.5a and Figure 8.6a do show a high amount of tracks originating from the ferry, as well as numerous tracks originating from the obstacle. This may imply that the DBSCAN clustering could be improved, as it is desired that only one measurement should be originated from each target. Additionally, tuning of the IPDA parameters could also be improved as only one configuration was tested.

When analyzing the position errors in Figure 8.7, one needs to consider that the LiDAR centroids would not necessarily conform with the position of the GNSS antenna mounted on the obstacle. This causes a time-varying offset depending on the pose of the obstacle relative to the LiDAR. This offset is difficult to find, and the actual position error is hard to quantify. It is believed that the use of the camera would aid this process as object detection can more reliably and accurately find the center of the obstacle, especially when it is perpendicular to the LOS angle.

It can be observed how the errors in the head-on scenario are relatively smooth with few spikes occurring. This can be verified by the RMSE and AEE being close to each other, where large errors would increase the RMSE values. The spikes that do occur are in the early stages of the track ReVolt, which is expected as the uncertainty is usually high in the preliminary stages of a track. Similar tendencies can be seen for the surge speed and heading estimates, where the estimates are close to the ground truth during the whole scenario. Speed and heading estimates are of great importance for the SBMPC, as these errors will accumulate when predicting the trajectory of the obstacle.

The position errors in the crossing from starboard scenario are also quite smooth, except for a significant spike at the end of the track. This can be seen in Table 8.5b where the RMSE is much larger than the AEE compared to the head-on scenario. The spikes may be caused by land masking due to the obstacle being too close to the pier. This causes the correct measurements to be discarded, resulting in a drastic covariance increase, seen in Figure 8.4. Consequently, this causes the existence probability to decrease, and the track is eventually terminated. The track uncertainties also affect the surge speed and heading estimates in Figure 8.9.

8.4.2 Collision avoidance

The closed-loop CAS shows very promising results in the two conducted scenarios. In the head-on scenario, the expected starboard maneuvers were performed in compliance with the COLREGS even though the first observed obstacle was about $35\mathrm{m}$ away, giving the CAS even less time to react. The low tracking range is probably due to the vertical separation of the LiDAR beams combined with the small obstacle. Recall from Section 3.1 that the LiDAR beams are separated with a 2° angle, meaning that the reflected points will be $35 \cdot \sin 2^\circ = 1.22\mathrm{m}$. Considering that the obstacle is a rigid inflatable boat with no cab, there are few reflected points at large distances, especially in a head-on scenario. Moreover, the performance of the GNC system, shown in Figure 8.11, was slower by a large margin when compared to Figure 7.21. The difference was also amplified by the environmental forces, which the current GNC system and SBMPC algorithm does not include, and remains as further work. Another solution would be to implement integral LOS guidance [42], effectively removing constant disturbances as currents and constant wind.

The crossing from starboard scenario shows promise when evaluating the CAS where the correct maneuvers are chosen. Regardless, the conflict between the heading and speed controller occurs in this scenario as well, but in this situation, the speed controller struggles to slow down in time. This causes ReVolt to move too close to the obstacle, which results in an unsatisfactory CPA. This is not, however, due to the track system, nor the SBMPC CAS.

The CAS shows great real-time capabilities with low computational time, even when considering multiple obstacles. The proactive nature of the SBMPC is yet to be shown due to the low range of the tracking system. The evasive maneuvers were more reactive, but the inclusion of the COLREGS makes the SBMPC valuable compared to more reactive methods as DW and VO. Results from the conducted experimental sea trials show that the closed-loop CAS can perform satisfactorily in urban environments, given that the vessels operate at low speeds.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

In this thesis, a closed-loop collision avoidance system based on the scenario-based model predictive control was implemented and tested in real-time simulations and experimental sea trials using ReVolt. A new sensor setup has been made in collaboration with DNV GL employees, and a camera-LiDAR calibration has been performed, giving an accurate transformation between the two sensor frames. A bug in the implemented IPDA algorithm has been fixed, and measurement models for the LiDAR, camera, and AIS have been formulated.

Improvements to the SBMPC algorithm in [5] have been presented and verified in closed-loop simulation studies. The results show that the introduction of a time-varying heading reference in the SBMPC prediction and the use of change points significantly improves the efficiency of the COLREGS compliant maneuvers.

Then, the closed-loop CAS was tested in experimental sea trails on ReVolt using an IPDA LiDAR tracking method. To the author's knowledge, there has not been conducted any real-life experiments using a LiDAR-based tracking system in a closed-loop CAS. The results are promising, but the performed maneuvers were suboptimal, probably due to the slow response of the maneuvering controllers on ReVolt.

9.2 Future work

In this section, the suggestions for future work on ReVolt will be presented.

- **Improve tuning of the maneuvering controllers:** The slow response of the maneuvering controllers was prevalent in the simulation studies and in the experimental sea trials, displaying the need for better tuning. Introducing gain scheduling could also improve the performance of the GNC at different cruise-speeds.

- **Implementing Integral Line-of-sight:** The environmental disturbances was a great contributor to the reduction of maneuverability during the experimental sea trials. One solution to this problem could be to implement integral line-of-sight guidance, which compensates for constant disturbances.

- **Implement RADAR and AIS:** Increase the range of the tracking system by implementing the exteroceptive sensors RADAR and AIS. Using a combination of RADAR, LiDAR and camera was shown to give promising results in [23].

- **Camera issue:** Investigate the camera issue that occurred during the experimental sea trials.

- **Improve the camera-LiDAR sensor fusion system:** Gather new data to verify and improve the performance of the camera-LiDAR fusion system. This includes tuning of the IPDA tracking system.

- **Multi-target tracking:** Implement a real-time multi-target tracking algorithm such as JIPDA or MHT.

- **Simulation platform:** Investigate the possibility to use a 3D-simulation platform for ReVolt such as Gazebo, where camera, LiDAR and RADAR plugins are supported. This may make the transition from simulations to real-life testing even shorter.

Bibliography

[1] Knut Turøy. Sensor fusion of camera-lidar for revolt. Unpublished, TTK4550 - specialization project, 2019.

[2] Edmund Førland Brekke, Erik Falmår Wilthil, BO H Eriksen, DKM Kufoalor, Ø K Helgesen, Inger Berge Hagen, Morten Breivik, and Tor Arne Johansen. The autosea project: Developing closed-loop target tracking and collision avoidance systems. In *Journal of Physics: Conference Series*, volume 1357, page 012020. IOP Publishing, 2019.

[3] Erik F Wilthil, Andreas L Flåten, and Edmund F Brekke. A target tracking system for asv collision avoidance based on the pdaf. In *Sensing and Control for Autonomous Vehicles*, pages 269–288. Springer, 2017.

[4] Håkon Gjertsen Norbye. Real-time sensor fusion for the revolt model-scale vessel. Master's thesis, Norwegian University of Science and Technology, 7 2019. Master's Thesis.

[5] Tonje Midjås. Sbmpc collision avoidance for the revolt model-scale ship. Master's thesis, Norwegian University of Science and Technology, 6 2019. Master's Thesis.

[6] D. K. M. Kufoalor, E. Wilthil, I. B. Hagen, E. F. Brekke, and T. A. Johansen. Autonomous colregs-compliant decision making using maritime radar tracking and model predictive control. In *2019 18th European Control Conference (ECC)*, pages 2536–2542, June 2019. doi: 10.23919/ECC.2019.8796273.

[7] Christine Chauvin, Salim Lardjane, Gael Morel, Jean-Pierre Clostermann, and Benoît Langard. Human and organisational factors in maritime accidents: Analysis of collisions at sea using the hfacs. *Accident Analysis & Prevention*, 59: 26–37, 2013.

[8] CML Fraunhofer et al. Maritime unmanned navigation through intelligence in networks. *Fraunhofer CML: Hamburg, Germany*, 2016.

[9] Alfred Norman Cockcroft and Jan Nanne Frits Lameijer. *Guide to the collision avoidance rules*. Elsevier, 2003.

[10] GL DNV. The revolt. *URL: https://www. dnvgl. com/technology-innovation/revolt/index. html*, 2018.

[11] Henrik Lemcke Alfheim, Kjetil Muggerud, Morten Breivik, Edmund Førland Brekke, Egil Eide, and Øystein Engelhardtsen. Development of a dynamic positioning system for the revolt model ship. *IFAC-PapersOnLine*, 51(29):116–121, 2018.

[12] Albert Havnegjerde. Remote control and path following for the revolt model ship. Master's thesis, NTNU, 2018.

[13] Vegard Kamsvåg. Fusion between camera and lidar for autonomous surface vehicles. Master's thesis, Master's thesis, NTNU, 2018.

[14] Yaakov Bar-Shalom and Edison Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, 1975.

[15] Darko Musicki, Robin Evans, and Srdjan Stankovic. Integrated probabilistic data association. *IEEE Transactions on automatic control*, 39(6):1237–1241, 1994.

[16] E. Brekke. *Fundementals of Sensor Fusion, Target tracking, navigation and SLAM, 1st edition*. 2019.

[17] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs Storrs, CT, 1995.

[18] Darko Musicki and Robin Evans. Joint integrated probabilistic data association: Jipda. *IEEE transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, 2004.

[19] Wolfgang Koch. *Tracking and sensor data fusion*. Springer, 2016.

[20] Christoph Stiller, Jochen Hipp, C Rössig, and A Ewald. Multisensor obstacle detection and tracking. *Image and vision Computing*, 18(5):389–396, 2000.

[21] Les Elkins, Drew Sellers, and W Reynolds Monach. The autonomous maritime navigation (amn) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles. *Journal of Field Robotics*, 27(6):790–818, 2010.

[22] Dan Hermann, Roberto Galeazzi, Jens Christian Andersen, and Mogens Blanke. Smart sensor based obstacle detection for high-speed unmanned surface vehicle. *IFAC-PapersOnLine*, 48(16):190–197, 2015.

[23] Øystein Kaarstad Helgesen. Sensor fusion for detection and tracking of maritime vessels. Master's thesis, Norwegian University of Science and Technology, 2019.

[24] Adan Lopez-Santander and Jonathan Lawry. An ordinal model of risk based on mariner's judgement. *The Journal of Navigation*, 70(2):309–324, 2017.

[25] Thomas Statheros, Gareth Howells, and Klaus McDonald Maier. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *The Journal of Navigation*, 61(1):129–142, 2008.

[26] CheeKuang Tam, Richard Bucknall, and Alistair Greig. Review of collision avoidance and path planning methods for ships in close range encounters. *The Journal of Navigation*, 62(3):455–476, 2009.

[27] Zhixiang Liu, Youmin Zhang, Xiang Yu, and Chi Yuan. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41:71–93, 2016.

[28] Riccardo Polvara, Sanjay Sharma, Jian Wan, Andrew Manning, and Robert Sutton. Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles. *The Journal of Navigation*, 71(1):241–256, 2018.

[29] Yamin Huang, Linying Chen, Pengfei Chen, Rudy R Negenborn, and PHAJM van Gelder. Ship collision avoidance methods: State-of-the-art. *Safety science*, 121:451–473, 2020.

[30] Yoonsoo Kim, Da-Wei Gu, and Ian Postlethwaite. Real-time path planning with limited information for autonomous unmanned air vehicles. *Automatica*, 44(3): 696–712, 2008.

[31] Hongguang Lyu and Yong Yin. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field. In *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, pages 351–357. IEEE, 2017.

[32] Øivind Aleksander G Loe. Collision avoidance for unmanned surface vehicles. Master's thesis, Institutt for teknisk kybernetikk, 2008.

[33] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[34] Thomas Stenersen. Guidance system for autonomous surface vehicles. Master's thesis, NTNU, 2015.

[35] Jia-yuan Zhuang, Lei Zhang, Shi-qi Zhao, Jian Cao, Bo Wang, and Han-bing Sun. Radar-based collision avoidance for unmanned surface vehicles. *China Ocean Engineering*, 30(6):867–883, 2016.

[36] T. A. Johansen, T. Perez, and A. Cristofaro. Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3407–3422, Dec 2016. ISSN 1558-0016. doi: 10.1109/TITS.2016.2551780.

[37] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen. Mpc-based collision avoidance strategy for existing marine vessel guidance systems. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7618–7623, May 2018. doi: 10.1109/ICRA.2018.8463182.

[38] Espen Johansen Tangstad. Visual detection of maritime vessels. Master's thesis, NTNU, 2017.

[39] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[40] Society of Naval Architects, Marine Engineers (U.S.). Technical, and Research Committee. Hydrodynamics Subcommittee. *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid: Report of the American Towing Tank Conference*. Technical and research bulletin. Society of Naval Architects and Marine Engineers, 1950. URL `https://books.google.no/books?id=VqNFGwAACAAJ`.

[41] Thor I Fossen and Ola-Erik Fjellstad. Nonlinear modelling of marine vehicles in 6 degrees of freedom. *Mathematical Modelling of Systems*, 1(1):17–27, 1995.

[42] Walter Caharija, Kristin Y Pettersen, Marco Bibuli, Pedro Calado, Enrica Zereik, José Braga, Jan Tommy Gravdahl, Asgeir J Sørensen, Milan Milovanović, and Gabriele Bruzzone. Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments. *IEEE Transactions on Control Systems Technology*, 24(5):1623–1642, 2016.

[43] Anastasios M Lekkas and Thor I Fossen. A time-varying lookahead distance guidance law for path following. *IFAC Proceedings Volumes*, 45(27):398–403, 2012.

[44] Velodyne LiDAR Manual. *Velodyne LiDAR VLP-16 User Manual*. Velodyne LiDAR, Inc, 2018.

[45] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL `http://arxiv.org/abs/1804.02767`.

[46] *CLASS A AIS POSITION REPORT (MESSAGES 1, 2, AND 3)*, 2019 (accessed May 22, 2020). URL `https://www.navcen.uscg.gov/?pageName=AISMessagesA`.

[47] Kenneth Gade. The seven ways to find heading. *The Journal of Navigation*, 69 (5):955–970, 2016.

[48] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna. LiDAR-Camera Calibration using 3D-3D Point correspondences. *ArXiv e-prints*, May 2017.

[49] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[50] Olga Sorkine. Least-squares rigid motion using svd. *Technical notes*, 120(3): 52, 2009.

[51] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[52] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[53] Subhash Challa, Mark R Morelande, Darko Mušicki, and Robin J Evans. *Fundamentals of object tracking*. Cambridge University Press, 2011.

[54] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6):82–100, 2009.

[55] Bjarne Foss and Tor Aksel N Heirung. Merging optimization and control. *Lecture Notes*, 2013.

[56] Kyle Woerner. *Multi-contact protocol-constrained collision avoidance for autonomous marine vehicles*. PhD thesis, Massachusetts Institute of Technology, 2016.

[57] Tomoji Takasu and Akio Yasuda. Development of the low-cost rtk-gps receiver with an open source program package rtklib. In *International symposium on GPS/GNSS*, pages 4–6. International Convention Center Jeju Korea, 2009.

[58] Map of dorabassenget, kartverket. `https://www.norgeskart.no/?_ga=2.226281818.1724638359.1594719376-258636982.1594719376#!?project=norgeskart&layers=1002&zoom=15&lat=7042887.53&lon=271758.81&markerLat=7042887.526855468&markerLon=271758.80786132807&panel=searchOptionsPanel&sok=Kobbes%20gate`, 2020. Accessed: 2020-07-14.

Knut Turøy

Closed-Loop Collision Avoidance System for the Revolt Model-Scale Vessel

NTNU
Norwegian University of
Science and Technology