# Collision Avoidance for the ReVolt Model-Scale Ship

Tonje Midjås,
Department of Engineering Cybernetics

December 18, 2018

NTNU

# Summary

Autonomous surface vessels (ASVs) are the future of maritime shipping. But to be able to act within the vicinity of other vessels a collision avoidance system is necessary. This system must adhere to the International Regulations for Preventing Collision at Sea (COLREGS).

In this project a COLREGS compliant collision avoidance system using simulation-based model predictive control, have been implemented on ReVolt model-scale ship. The system was tested using four different scenarios: head-on, crossing from starboard, crossing from port and overtaking. Results from simulations were satisfactory, yielding predictable COLREGS compliant behaviour in all four cases. All simulations were done using liner prediction and ideal conditions.

Real life testing of the collision avoidance system on ReVolt was also conducted. The vessel showed promising results taking action to avoid collision, but with mixed performance on acting reliable and complying to COLREGS. Improvements to increase performance would be to perform more realistic simulations before further sea-trials, in combination with a need for further tuning.

# Table of Contents

# Abbreviations

AIS = Automatic Identification System
ASV = Autonomous Surface Vessel
CAS = Collision avoidance system
COLAV = Collision avoidance
COLREGS = International regulations for avoiding collisions at sea
DOF = Degree Of Freedom
DW = Dynamic Window
ENU = East North Up
ESC = Electronic Speed Control
GPS = Global Positioning System
IMU = Inertial Measurement Unit
LOS = Line of Sight
MPC = Model Predictive Control
NED = North East Down
ROS = Robot Operating System
SBMPC = Simulation based model predictive control
SIL = Software In the Loop
SNAME = Society of Naval Architects and Marine Engineers
TCP = Transmission Control Protocol
VO = Velocity Obstacle

# Chapter 1

# Introduction

## 1.1 Motivation

A lot of research have been done to come up with solutions that would reduce the amount of work needed to be done by humans, to save time and ease the workload. After the industrial revolution, machines started taking over a lot of the hard work previously done by humans. In modern times automated systems, nearly independent of humans interaction, have been invented. In contrast to earlier times the new motivation for automated systems is speed of execution, high quality work, environmental friendliness and cost-effectiveness. One of many fields where this technology will make immense improvements is the marine shipping industry.

At least 75% of all marine accidents are caused by human error [20], which leaves big room for improvement. A possible solution is the introduction of autonomous surface vessels (ASVs). ASVs will be important in the shipping industry, helping to move some of the load off the roads. In 1960, two thirds of the domestic shipping was done by ships, but now the amount is reduced substantially [23]. At the same time the road transport has increased. In 2017 the total shipping performed with trucks in Norway was 21.4 billions ton-kilometers [24]. The increase is due to the easy, reliable and cheap availability of road transport. For these reasons there is a need to make domestic short distance shipping cheaper and more reliable at sea, hopefully leading to reduced shipping by trucks.

The ideal autonomous ship has no crew, hence removing the man-made risk of accident, but at the same time introducing other kinds of hazards that need to be assessed. The no crew feature allows for optimization of the boat structure by removing the crew facilities freeing more space for payload. This will also lead to economical savings, as there is no need to pay crew salary, as well as the added income from additional payload. Furthermore there is fewer people interested in working at sea, making it hard to hire the necessary crew for shipping purposes. This issue is understandable as it is unappealing to be bounded inside a ship for two weeks straight, with minimal contact with friends and family. Hence the need for autonomous ships will emerge regardless of all other benefits [36].

For autonomous surface vessels to be accepted they need to be at least as safe as a conventional vessel. The first step towards that is a good guidance, navigation and control systems. And secondary, as the ASV will be navigating in waters with other ships present, a crucial part is a reliable collision avoidance system, to avoid dangerous situations and act predictable for other manned ships.

## 1.2 Background and previous work

A comprehensive historical background of work done within collision avoidance and path planning is given in the review articles [29] and [31]. Including methods both with and without compliance to the International Regulations for Preventing Collision at Sea (COLREGS). In [27] collision avoidance methods based on both local and global path planning algorithms are discussed. The global methods can only handle static environments, hence the local methods are the way to go for avoiding collision in the dynamical changing environment experienced at sea. By combining the two methods an algorithm guaranteeing that it gets from A to B without colliding with dynamical obstacles appearing along the way could be achieved.

A hybrid approach based on a modified version of the Dynamic Window (DW) algorithm with path planning provided by the Rapidly-Exploring Random Tree algorithm is proposed in [21]. The original DW method predict vehicle movement along a constant-radius ache, designing a space of velocities called the dynamic window. This space is further reduced by including vehicle dynamic, limiting control offsets to only velocities possible to reach within the next time interval and that are safe with regards to collision. The system proposed in [21] with its enhancements preformed well in both simulations and the full-scale experiments they where able to complete. Another purely local method is Velocity Obstacle (VO) implemented by [30]. This algorithm designs a velocity space including all velocities eventually leading to collision. It is a simple algorithm not dependent on the vessel dynamic. Full-scale testing was conducted, showing promising result for collision avoidance in compliance with COLREGS.

The methods currently existing do not scale very well according to [18]. Lacking the ability to handle dense traffic with multiple dynamic obstacles, while considering both the ship model and the environmental disturbances. Further pointing out that including such complex scenarios into already existing algorithms would be non-trivial. [18] proposes a possible solution based on model predictive control, where a ship model in concurrence with environmental disturbances is used to predict the ship's trajectories. The predictions are done with a finite set of possible control behaviour, which are evaluated based on collision risk, hazard, operational constraints and objectives. The paper describes the over all structure of the collision avoidance systems as well as including COLREGS compliance into the cost calculations. The simulations performed shows promising result also handling multiple obstacle scenarios. This articles is the background for this project, further explored in [15].

## 1.3 Problem Description

The scope of this project is to implement and test collision avoidance on the ReVolt model-scale ship. Carried out by firstly implementing Simulation-Based MPC into ReVolts existing code base. For testing, virtual objects will be used, consequently a way of adding them will be implemented using the Qt software platform. Then simulation testing should be conducted, using DNV GLs simulator called CyberSea. When the performance of the algorithm is confirmed, real life testing on the physical ReVolt will be executed. To enhance the algorithm further, collision avoidance with land and ground should be added to the system.

## 1.4 Contribution

The main contributions of this project are the implementation and testing of the simulation-based MPC algorithm. The code have been added into ReVolt's already excising code-base and satisfactory behaviour confirmed in simulations. Furthermore, results from collision avoidance experiments in Trondheimsfjorden are obtained.

## 1.5 Outline

**Chapter 2** provides the necessary theoretical background needed for this project. In **Chapter 3** ReVolt, the experimental platform, is presented. **Chapter 4** describes the SB-MPC algorithm and the implementation of it. The simulations are presented in **Chapter 5** and the real life experiments in **Chapter 6**. A discussion of the results is given in **Chapter 7**, whilst further work is proposed in **Chapter 8**. Finally followed by a conclusion in **Chapter 9**.

Chapter 2

# Theory

## 2.1 3DOF Ship model

This section presents the 3-Degrees of Freedom (3DOF) model using the notation used in Society of Naval Architects and Marine Engineers (SNAME), shown in table 2.1.

| **DOF** | **Forces and moments** | **Linear and angular velocities** | **Positions and Euler angles** |
|---|---|---|---|
| Motion in the x direction (surge) | $X$ | $u$ | $x$ |
| Motion in the y direction (sway) | $Y$ | $v$ | $y$ |
| Rotation about the z axis (yaw) | $N$ | $r$ | $\psi$ |

**Table 2.1:** The notation of SNAME (1950) for marine vessels. Only showing the relevant 3DOF. [14]

For the purpose of this project I will only be concerned with the motion in the horizontal plane described by the motion components in surge, sway and yaw. Meaning I neglect the dynamics associated with roll p, pitch q and heave w, giving $p = q = w = 0$. This results in the following 3DOF model [14]:

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

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

where $\boldsymbol{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}^T$ is velocity of the vessel and $\boldsymbol{\eta} = \begin{bmatrix} N & E & \psi \end{bmatrix}^T$ is the pose given in the earth-fixed North-East-Down (NED) reference frame. $\boldsymbol{M}$ is the system inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix and $\boldsymbol{D}(\boldsymbol{\nu})$ is the damping matrix. $\tau$ is the generalized force vector. The kinetic equation of motions is reduced to one principal rotation about the z-axis presented in equation 2.1, where $\boldsymbol{R(\psi)}$ is the rotation matrix transforming the body-fixed velocities into the world-fixed frame:

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

$\boldsymbol{\tau}$ contains the forces resulting from the vessel's actuator and is in this case given as:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ l_r F_y \end{bmatrix}, \tag{2.3}$$

where $\tau_X$ and $\tau_X$ are the forces along x- and y-axis, and $\tau_N$ is the moment around the z-axis. $F_x$ and $F_y$ represent forces in x- and y-direction respectively, and $l_r$ is the arm the moment is acting on.

In addition to assuming no vertical motion we assume that the craft has homogeneous mass distribution and symmetry about the xz-plane such that $\boldsymbol{I}_{xy} = \boldsymbol{I}_{yz} = 0$, where $\boldsymbol{I}_{xy}$ and $\boldsymbol{I}_{yz}$ are products of inertia. Also letting the body-frame coordinate origin coincide with the center of the ship(CO), such that $y_g = x_g = 0$ where $x_g$ and $y_g$ are the distances from CO to center of gravity (CG) in respectively x- and y-direction.

Based on these assumptions in addition to some added simplifications we can define matrices $\boldsymbol{M = M_A + M_{RB}}$ and $\boldsymbol{C(\nu) = C_A(\nu) + C_{RV}(\nu)}$. The subscript RB stands for rigid-body and A stand for added mass, hence the inertia matrix is built up of the rigid-body mass of the vessel as well as the added mass. Added mass comes from the water displacement when accelerating or decelerating. Setting $\boldsymbol{M_A} = 0$ to keep the model simple, resulting in:

$$\boldsymbol{M} = \boldsymbol{M_{RB}} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \tag{2.4}$$

where $m$ is the vessel mass and $I_z$ is the moment of inertia about the z-axis. The Coriolis matrix $\boldsymbol{C(\nu)}$ is also constructed from a rigid-body part and an added mass part. For simplicity also setting the hydrodynamic Coriolis and centripetal matrix $\boldsymbol{C(\nu)_A} = 0$, leaving the following matrix:

$$\boldsymbol{C(\nu)} = \boldsymbol{C_{RB}(\nu)} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix} \tag{2.5}$$

The damping matrix is contructed by a linear and a nonlinear part and is defined by: $\boldsymbol{D(\nu)\nu = D_L\nu + D_{NL}(\nu)\nu}$ to keep it simple for this thesis. The linear and nonlinear matrices are defined repectivly:

$$\boldsymbol{D_L} = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \tag{2.6}$$

$$\boldsymbol{D_{NL}}(\boldsymbol{\nu}) = -\begin{bmatrix} X_{|u|u}|u|u + X_{uuu}u^3 \\ Y_{|v|v}|v|v + Y_{vvv}v^3 \\ N_{|r|r}|r|r + N_{rrr}r^3 \end{bmatrix}, \tag{2.7}$$

where all parameters are defined in table 2.1 above.

## 2.2 Controllers

The autopilot consist of two controllers, one for heading and one for speed. The heading controller is a PID controller with feedforward term based on the first order Nomoto model, and the speed controller is a PI controller with feed forward term and a reference model supplying desired velocity. Both controllers implemented on ReVolt are designed by Albert Havnegjerde in [16].

### Heading controller

The objective of the heading controller is to track both heading and yaw rate, which error states are given by $\tilde{\psi} \triangleq \psi_d - \psi$ and $\tilde{r} \triangleq r_d - r$, where $\psi_d$ is desired heading and $r_d$ is desired yaw rate. These are both time-varying and supplied by a reference model, while $\psi$ and $r$ are the associated measurements. The heading control law is formulated as:

$$\tau_\delta = \tau_{\delta,FF} + \tau_{\delta,FB}, \tag{2.8}$$

which consist of a feedforward term ($\tau_{\delta,FF}$) and a feedback term ($\tau_{\delta,FB}$) respectively given as:

$$\tau_{\delta,FF} = \frac{T}{K}\left(\dot{r}_d + \frac{1}{T}r_d\right), \tag{2.9}$$

where $K$ and $T$ are the gain and time constant from the Nomoto model, which the feedforward term is based on.

$$\tau_{\delta,FB} = -\left(K_p\tilde{\psi}(t) + K_i\int_0 t\tilde{\psi}(\tau)d\tau + K_d\tilde{r}(t)\right) \tag{2.10}$$

$K_p$, $K_i$ and $K_d$ are the proportional, integral and derivative controller gains.

**Speed Controller**

The control objective is to track the desired surge reference speed $u_{ref}$. This is done by minimizing surge speed error $\tilde{u}(t) \triangleq u_d(t) - u_f(t)$, where $u_d(t)$ is the time-varying, desired surge speed given by a second order reference filter and $u_f(t)$ is the low-pass filtered measurement of velocity. The control law is given by:

$$\tau_m = \tau_{m,FF} + \tau_{m,FB}. \tag{2.11}$$

$\tau_{m,FF}$ is the feedforward term and is given by:

$$\tau_{m,FF} = M\dot{u}_d + \sigma(u_d), \tag{2.12}$$

where $M\dot{u}_d$ is the inertia term and $\sigma(u_d)$ is the steady-state polynomial damping term. Further is $\tau_{m,FB}$ the feedback term which is given by:

$$\tau_{m,FB} = K_p\tilde{u}(t) + K_i \int_0^t \tilde{u}(\tau)d\tau, \tag{2.13}$$

where $K_p$ and $K_i$ are the proportional and integral gain of the controller.

## 2.3 Line-of-sight guidance

Line-of-sight (LOS) guidance is a path-following algorithm which calculates a desired course angle. The control objective is minimize the cross-track error which is the shortest distance from own-ship to the desired path. In this case a lookahead-based steering is used giving desired course angle as:

$$\chi_d(e) = \chi_p + \chi_r(e) \tag{2.14}$$

where

$$\chi_p = \alpha_k = \arctan 2(y_{k+1} - y_k, x_{k+1} - x_k) \tag{2.15}$$

$x$ and $y$ are the coordinates of the waypoints in the desired path. Both the desired path and the actual path taken are rotated relative to the $x_n$ which is the relative North axis. $\alpha_k$ represents the rotation between North and the desired path. $\chi_r(e)$ is given by:

$$\chi_r(e) = \arctan\left(\frac{-e}{\Delta}\right), \tag{2.16}$$

where $e$ is the cross-track error and $\Delta$ is the lookahead distance. $\Delta > 0$ and a rule of thumb is to set it to 1.5-2.5 ship lengths [14]. From (2.16) it is evident that a small $\Delta$ yields a big $\chi_r(e)$, and hence a more aggressive convergence to the desired path. Taking ocean current into consideration we need do account for the sideslip-angle $\beta$. Sideslip is the difference between course and heading angle. When having velocity measurements available the new output from the algorithm is:

$$\psi_d = \chi_d - \beta \tag{2.17}$$

where $\beta$ can be calculated as:

$$\beta = \arcsin\left(\frac{v}{U}\right) \tag{2.18}$$

$v$ is sway velocity and $U$ is the speed over ground.

## 2.4 Model Predictive Control

Model Predictive Control (MPC) is one of the most widely accepted modern control strategies because of its sensible compromise between speed of computation and optimality. An overall description of MPC would be that it predicts future behaviour using a model and a given hypothetical future input. Then only the first input of the predicted optimal control sequence is applied to the actual system [8].

```
for $t = 0, 1, 2, ...$ do
    Compute an estimate of the current state $\hat{x}_t$ based on the measured data up until time t.
    Solve a dynamic optimization problem on the prediction horizon from $t$ to $t + N$ with $\hat{x}_t$ as the initial condition.
    Apply the first control move $u_t$ from the solution above
end
```

**Algorithm 1:** Output feedback MPC procedure [13]

The algorithm solves an optimization problem over and over at each time step. It has a moving horizon, meaning that the prediction horizon will move one step at each time step. An important part of the algorithm is how to find the initial value $x_t$. This can be done using the predicted value $x_{t+1}$, predicted at time t or by doing state estimate based on available measurements. The reason for doing the latter is that our prediction will not account for disturbances and modeling error, hence it can easily give a bad estimate of the actual initial value [13].

The main challenges of MPC relates to computational complexity and convergence. As collision avoidance (COLAV) scenarios can get extremely complex, it may lead to non-convex optimization problems. Such problems can exhibit local minimums and be hard to solve, making conventional MPC not optimal for COLAV. Therefor model formulation, discretization, control trajectory parameterization, constraints and objectives need to be considered carefully, along with issues like dependability [17]. In the basic MPC algorithm presented above we assume that the plant used for prediction is the same that is to be controlled. This is generally not a valid assumption as there will be unmeasured noise in the system. We need to have a guaranty for feasibility and convergence, as not getting a result is unacceptable. Robust MPC is a solution to this [5].

Robust MPC utilizes the concept of optimizing over a finite set of possible control behaviour, and can be as simple as picking between a discrete number of outputs based on cost comparison e.g. [4], but most approaches incorporates optimization over control parameters to enhance the performance. Using this method we completely avoid numerical optimization, assuring feasibility and a resulting system that could be able to perform in real-time. This approach will reduce the degrees of freedom possible to control, hence imposing responsibility of performance on to picking a decent set of possible control behaviour.

## 2.5 International regulations for avoiding collisions at sea (COLREGS)

Rules for preventing collision at sea have been in existence for several hundred years, but they have only been of statutory force in the last century. The once we follow today have emerged from years of development and came in to effect in 1972. They are called the international regulations for avoiding collision at sea or COLREGS. COLREGS is divided into five parts, Part A - general, Part B - steering and sailing rules, Part C - lights and shapes, Part D - sound and light signals and lastly Part E - exemptions [9]. Part B is the most relevant for this project and more specific rules 6, 8, 13, 14, 15, 16 and 17. The rest of this section provides an overview of these rules [3].

#### 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. Visibility, traffic density, weather, water depth and more must be taken into account when determining a safe speed.

#### Rule 8 - Action to avoid collision

For a vessel in risk of collision, action to avoid it shall be taken with accordance to the rules of part B. Any alteration of course and/or speed to avoid collision shall be large enough to be readily apparent and made in ample time. The action shall result in passing at a safe distance.

#### Rule 13 - Overtaking

Any vessel overtaking any other shall keep out of the way of the vessel being overtaken. It is deemed a overtake vessel when it comes up with another vessel from a direction more than 22.5 degrees abaft her beam.

#### Rule 14 - Head-on situation

When to power-driven vessel are meeting on reciprocal courses we have a head-on situation and each shall alter her course to starboard, so they both pass on port side.

### Rule 15 - Crossing situation

There is a crossing situation if two power-driven vessels are crossing in a way that involve risk of collision. Then the vessel which has the other on her starboard side, is deemed the give-way vessel and should avoid collision as well as try to avoid passing in front of the other vessel.

### Rule 16 - Action by give-way vessel

A give-way vessel shall as far as possible take a substantial and early action to avoid collision.

### Rule 17 - Action by stand-on vessel

The stand-on vessel should keep her speed and course. However when it becomes apparent that the give-way vessels actions alone is not enough to avoid collision, the stand-on vessel should take action as well. This rule do not relieve the give-way vessel of her obligations.

Chapter 3

# Experimental Platform

## 3.1 ReVolt concept

ReVolt, an efficient, safe and environmentally friendly short sea shipping concept, designed by DNV GL to help explore how much improvement is possible by utilizing state of the art technology. It is designed to be an unmanned, zero emission ship for the future, transporting cargo along the Norwegian coast at a low cost. As well as lowering costs ReVolt is set to reduce the number of fatalities caused by human error, introducing opportunity for an autonomous system. Such a system requires sensors for situational awareness, guidance, navigation and control as well as collision avoidance systems.

**Figure 3.1:** Concept ReVolt [33]

The concept ship has an optimal speed of 6 knots, cargo capacity at 100 TEU and and operational range of 100 nautical miles. The hull was designed with a straight vertical bow to minimizing resistance and optimize ship efficiency. At the low cruising speed of the vessel, the only resistance to overcome will be hull friction and some environmental forces like waves and wind. The propulsion system is fully electrical and consists of two stern pods for the main propulsion and one retractable bow thruster for manoeuvring. [22]

## 3.2 ReVolt test platform

In 2014 a 1:20 scale model of the ReVolt was built, delivered by Stadt Towing Tank. The model has the same thruster configuration and hull design as the concept vessel. September 2018, a skeg was added in the aft of ReVolt. The skeg is an additional fin which will aid the directional stability, which was strongly lacking. The enhancements resulted in ReVolt being considerable less difficult to control.

**Figure 3.2:** ReVolt test platform

ReVolt as it is today has a maximum speed of about $1.5\frac{m}{s}$ and the thruster angle is restricted at $\pm 45^\circ$ offset to each side during transit. These restrictions influence ReVolts maneuverability. It classifies as a slow system, and because of the restrictions the turning radius is quite vast. There will also take some time from the offset in desired course angle is given until it is reached. All factors to be considered while designing the control systems.

The vessel is a scale-model, testing in a full scale environment, obviously leading to some challenges. Waves, wind and ocean current have 20 times the effect on the test platform compared to the concept model, as the configuration of the boat is design to be full scale. There are probably many unknown effects of this problem, but the most evident and known difficulty is weather conditions. With ReVolts low maximum speed, there is not much extra power to be used when sailing in rough waters, hence the propulsion is close to zero when sailing up against bigger waves or large ocean current. Also immense waves from the sides increase the risk of ReVolt actually capsizing. Based on this assessment ReVolt is best tested in calm water, where environmental factors have minimum effect.

### 3.2.1 Components

This section presents the existing components on the ReVolt. The main components are the embedded computer, motors with associated electronic speed controllers (ESC), the global positioning system (GPS), the Xsens and the two Arduinos controlling the motors. This year new batteries where inserted to increase possible testing time. There is also need for a more powerful embedded computer, in order to do the real time calculations necessary for an autonomous surface vessel.

| **Name** | **Placement** | **Model** |
|---|---|---|
| Motor controller | Bow | Robbe NavyControl535R |
| DC-motor | Bow | Robbe Roxxy Starmax 48 |
| Linear actuator | Bow | Firgelli L16 |
| Servo | Bow | HiTEC HS-5485HB |
| H-bridge | Bow | L293NE |
| Motor controller (ESC) | 2 x Stern | Robbe Roxxy Control 900 |
| AC-motor | Stern | Robbe Roxxy BL-outrunner 5055-45 |
| Stepper motor | 2 x Stern | Nanotec PD2-N41 |
| Current measurment sensor | 2 x Stern, 1 x bow | Phidgets 1122_0 |
| Inductive sensor | 2 x Stern | XS618B1PAL2 |
| Xsens | Top middle | Xsens MTi-G-710 |
| Vector | Middle | Hemisphere VS330 |
| Antenna | 2 x Stern | Hemisphere A45 |
| Water sensor | Under batteries | Homemade |
| Embedded computer | Middle port side | Tank 720 |
| Hard drive | Middle port side | Verbatim 500GB |
| 4G router | Stern port side | TP-Link MR200 |
| Arduino Uno | Bow | Arduino Uno R3 |
| Arduino Mega | Stern | Arduino Mega |
| Battery | 2 x Middle | Exide EP650 |
| Relay | Middel starboard side | - |
| RC remote | - | Spektrum DX6i |
| RC receiver | Stern | Spektrum AR610 |
| Light beacon | Stern | - |

**Table 3.1:** Main components on the ReVolt

Others are working on adding both Lidar and video camera systems to ReVolt, making it possible to design an obstacle detection system in the future. With the current equipment available, ReVolt do not have whats necessary for doing object detection and tracking, both with regards to sensors and the computational power of the embedded computer. An automatic identification system (AIS) receiver would also be a possible wanted component to increase reliability on the object detection system.

Chapter 4

# Maritime collision avoidance

To achieve a fully autonomous system we need a way of avoiding collision with other ships, obstacles and land. This have to be done in a safely manner, guided by the International Regulations for Preventing Collisions at Sea (COLREGS). These rules specify actions to be taken when ships operate in near proximity of other vessels. Preventing dangerous situations with a high risk of collision, and ensuring reliable and predictable actions for everyone around. A good amount of approaches to this problem have been presented, both with and without COLREGS compliance. Simple collision avoidance (COLAV) algorithms like velocity obstacle [12] and dynamic window [11] do not incorporate COLREGS directly. This is not sufficient for a ship to be fully autonomous among other manned ships. Examples of methods capable of incorporating COLREGS compliance are fuzzy logic [19], evolutionary path planning based algorithms [10], interval programming [6], 2D grid maps [32] and presumably many others.

## 4.1 COLAV based on model predictive control

Collision avoidance based on model predictive control (MPC) have been used in several different settings like autonomous vehicles [28], cars [35], or in autonomous aerial vehicles [25] [34]. Further some elements of optimization and optimal control have been used at sea before [21], but the use of MPC for ship collision avoidance in compliance with COLREGS was first proposed by Johansen et al. [18], and that method is called simulation based model predictive control (SBMPC). Afterwards the concept has been utilized in other COLAV methods. In [7] they further develop the method by implementing a MPC algorithm for mid-level COLAV using nonlinear programming, while [15] uses the SBMPC with COLREGS compliance to improve performance and robustness of COLAV, and reduce dependability on knowing the exact guidance method. Further in [26] the algorithms is adapted to a system with less freedom to change its propulsion.

## 4.2 SBMPC architecture

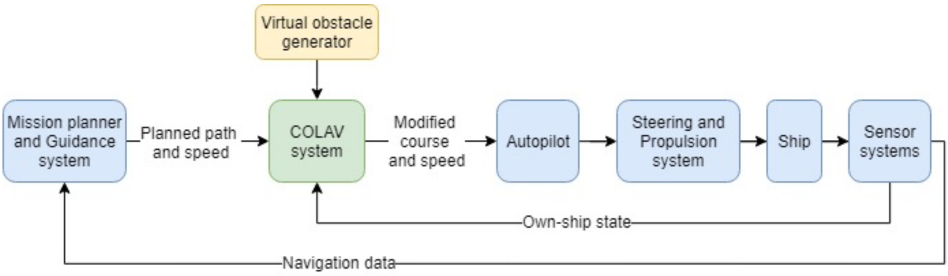


**Figure 4.1:** System architecture and information flow including SBMPC. Blue boxes represents the already existing code-base on ReVolt. Green is the added SBMPC structure and the yellow is the additional necessities for testing. Inspired by [18].

The proposed architecture for SBMPC in [18] allows for a highly modular system, as the COLAV algorithm is separate from both the mission planer and the autopilot. This makes SBMPC easy to add on to already existing systems, increasing opportunity for real life testing on already existing ships. Figure 4.1 illustrates the overall structure of a systems using SBMPC. The algorithm's input is the sensor data and planned trajectory in form of a list of waypoints. Using this information the COLAV system searches for a collision free and COLREGS compliant path to take. This path should be as close to the nominal path as possible, without being hazardous. The COLAV system then outputs control offsets in form of a course offset and a adjustment to the speed. This course and speed adjustment combination is what defines a scenario in SBMPC.

The SBMPC algorithm is realized by a finite horizon and finite scenario minimization problem. The objective functions is defined by the ship plant model, and the constraints are formed by environmental forces, model uncertainty, weather, obstacle trajectory uncertainty, hazard and operational cost. Collision avoidance could be a complex problem, which may yield non-convex optimization problems without a solution. The key here is the finite scenario part, which makes the optimization problem deterministic and therefor always yields a result. To formulate our problem in this manner we use theory from robust MPC described in section 2.4, which is carried out using a discrete set of possible control offsets and speed adjustments. Having a deterministic problem is important as not getting a result would be unacceptable in real life situations. A drawback of the proposed solution is the reduction in degrees of freedom, making the selection of the set of possible control offsets a very important design decision.

Moreover we recognize from figure 4.1 that the systems is dependent on a set of information. In order to support collision avoidance we have to assume availability of the following:

- List of obstacle's position and velocities
- List of waypoints for desired path and desired speed

- Own ship's state
- Mathematical model of own-ship

## 4.3 COLAV with SBMPC

Making SBMPC work for collision avoidance is based on some assumptions. It is assumed that the collision avoidance system (CAS) is working in real-time, meaning it is able to compute the best control input faster than real time. Solving the optimization problem is done using a receding horizon, with re-optimization every 5 seconds, utilizing new information from the sensors. All the scenarios are evaluated and a cost is calculated based on collision hazard and compliance with COLREGS. Then the scenario related to the lowest cost is chosen and the associated control offsets sent to the autopilot. The cost function uses velocities and line-of-sight vectors to express the COLREGS rules, as well as distance and speed to evaluated collision hazard. The general overview of the COLAV module is as follows:

1. For each pair of course and speed offset, the trajectory of both the ship and the obstacles are predicted.
2. The cost function is applied to the same set of control behaviours, calculating the cost using the associated ship and obstacle trajectories.
3. Choose the control behaviour set corresponding to the lowest cost, and apply the first control input.
4. Repeat when new sensor information is available.

### 4.3.1 Prediction of trajectories

To predict the trajectory of the own-ship both linear- and Euler prediction can be utilized. Both will be discussed here, but Euler prediction with the 3DOF model explained in section 2.1 is the preferred method. However, for simplicity only linear, straight line prediction is used in this case. Also taking into consideration that this would most likely also be necessary for the current ReVolt on-board computer to be able to compute the control offsets in real-time. Even though one of the advantages of MPC is the possibility to include all the vessel dynamics, steering and propulsion systems and weather-, wind- and ocean current information, we will still get a satisfactory result using a simplified system. The reason for this is our small, some what easy to maneuver test platform.

On the other hand, when applying this method to a bigger ship with slow dynamics, like the ReVolt concept ship described in section 3.1, the 3DOF model is not even good enough. We would need to take all the environmental forces into consideration and use a 6DOF model with less or none simplifications because of the slower dynamics. For the purpose of this project, the linear prediction will be sufficient. Keeping in mind that it do not take ReVolts slow turning into the calculations, expecting the course changes to happen instantaneously. Euler prediction is a first order numerical procedure that calculates the shape of an unknown curve given a initial value and differential equation. Hence it tries to take the model dynamics into account when calculating the predicted path of the own-ship, resulting in a much more accurate prediction.

Predicting our own path has its challenges, but at least we know the ship dynamics and the input. It is a much greater challenge to predict the path of the obstacles, which is one of the significant concerns when implementing collision avoidance. Knowing only current position, heading and speed the easiest approach is to use linear prediction for the obstacles as well, given no information what so ever on the vessel model. As we update our prediction every 5 seconds in the SBMPC method this solution will provide enough accuracy, and combined with our safety margins the overall systems will work sufficiently for this purpose.

The linear prediction is given by the following equations:

$$\hat{x}_{i+1} = \hat{x}_i + (t - t_0) \cdot \hat{u}_i \tag{4.1}$$

$$\hat{y}_{i+1} = \hat{y}_i + (t - t_0) \cdot \hat{v}_i \tag{4.2}$$

where $t_0$ is the time of the measurements and $t$ is some future point in time. $\hat{u}_i$ is the assumed speed in x-direction and $\hat{v}_i$ in y-direction.

### 4.3.2 Selection of control law behaviour

To decide control behaviour a lot of different scenarios are evaluated. A scenario consist of the current state of the own ship, its desired control behaviours and the predicted path of the obstacles. The control behaviour is either constant on the prediction horizon or change $p$ number of times. Johansen et. al. [18] states a minimum of sets that should be evaluated, which is the base for the choices used in this project:

- Course offsets at $\{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$ in degrees.
- Keep speed, slow forward and stop commands. Represented as $\{1, 0.5, 0\}$.

The original suggestion of the speed set included backwards propulsion, which is omitted here to reduce computations as driving backwards is unwanted behaviour. With all combinations of these we have $13 \cdot 3 = 39$ possible control behaviours, assuming that they are kept constant over the prediction horizon. If you choose to change control behaviour on the horizon the number of possibilities quickly increase, and with one change you already have $39^2 = 1521$ different possible control behaviours. Hence the computational cost increase equally rapid. If there is access to enough computational power, allowing change in course on the horizon would likely increase performance, if not, a bigger set of control values could be an alternative enhancement. For this project the proposed minimum of course offsets are used.

### 4.3.3 COLREGS compliance

As for the road the sea have its own set of rules, called COLREGS - International Regulations for Preventing Collisions at Sea. When performing COLAV it is important for the autonomous surface vessel (ASV) to follow those rules, making the actions taken logical and predicable for operators of other vessels. The CAS uses the available information illustrated in figure 4.2 and specified in table 4.1 to evaluate the situational hazard with respect to COLREGS. Furthermore, the COLREGS compliant control action with the lowest risk is chosen.

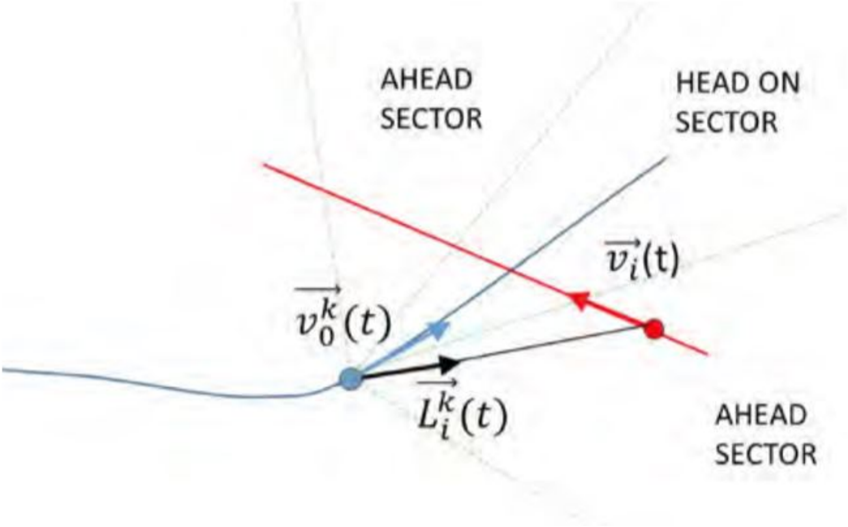


**Figure 4.2:** Main information for hazard evaluation in scenario $k$ at time $t$ [18]

| **Parameter** | **Description** |
|---|---|
| $\vec{v_o}^k(t)$ | Predicted velocity of own-ship in scenario $k$ |
| $\vec{v_i}(t)$ | Predicted velocity of obstacle with index $i$ |
| $\vec{L_i}^k(t)$ | Unit vector in LOS direction for own-ship to the obstacle with index $i$ in scenario $k$ |
| $d^k_{o,i}(t)$ | Predicted distance between own-ship and obstacle with index $i$ at time $t$ in scenario $k$ |
| $d_i^{cl}$ | The largest distance where COLREGS apply |

**Table 4.1:** Parameter description for hazard calculations with respect to COLREGS [18]

The blue curve is the predicted path of ReVolt, and the red for the obstacle, both based on the most recent measurements. The blue and red dots denote the predicted position at some future point in time, $t$. The vectors attached to them represent the predicted velocities at that time, denoted $\vec{v_o}^k(t)$ and $\vec{v_i}(t)$. The black vector is the Line-of-sight (LOS) vector, denoted $\vec{L_i}^k(t)$. It is a unit vector which represents the direction from ReVolt to the obstacle. Along with these vectors the distance between the ships is needed, denoted $d^k_{o,i}(t)$, as well as a distance representing the COLREGS "area", denoted $d^{cl}_i$. Utilizing these five parameters we can evaluate the risk at a time $t$ in a scenario $k$. Descriptions of how COLREGS connect to the scenarios, could be described by different boolean parameters. These parameters are valid when the own-ship and the obstacle is within COLREGS perimeters, meaning the distance between them is less than $d^{cl}_i$. All expressions below are obtained from [18] and the parameters used are described in table 4.1:

- **CLOSE:** When $d^k_{o,i}(t) \leq \mathrm{d}^{cl}_i$, ReVolt is said to be CLOSE to the obstacle $i$.

- **OVERTAKEN:** ReVolt is said to be OVERTAKEN by obstacle $i$ if
$$\vec{v_o}^k(t) \cdot \vec{v_i}(t) > \cos(68.5°)|\vec{v_o}^k(t)||\vec{v_i}(t)|$$

- **STARBOARD:** Obstacle is STARBOARD of ReVolt if the bearing angle of $\vec{L_i}^k(t)$ is larger than ReVolts heading.

- **HEAD-ON:** Obstacle $i$ is HEAD-ON if it is CLOSE to ReVolt and:
$$|\vec{v_i}(t)| > 0.05$$
$$\vec{v_o}^k(t) \cdot \vec{v_i}(t) < -\cos(22.5°)|\vec{v_o}^k(t)||\vec{v_i}(t)|$$
$$\vec{v_o}^k(t) \cdot \vec{L_i}^k(t) > \cos(\phi_{ahead})|\vec{v_o}^k(t)|$$
where the angle $\phi_{ahead}$ is to be selected.

- **CROSSING:** ReVolt is CROSSING the obstacle if it is CLOSE and:
$$\vec{v_o}^k(t) \cdot \vec{v_i}(t) < \cos(68.5°)|\vec{v_o}^k(t)||\vec{v_i}(t)|$$

To indicate a violation of COLREGS the boolean parameter $\mu^k_i(t) \in 0, 1$ is used, as in [18]. The rules taken into account here is mainly rules 14 (head-on) and/or 15 (crossing), resulting in the following expressions:

$$\mu^k_i(t) = \textit{RULE 14 or RULE 15}$$
$$\textit{RULE 14} = \textit{CLOSE \& STARBOARD \& HEAD-ON}$$
$$\textit{RULE 15} = \textit{CLOSE \& STARBOARD \& CROSSING \& NOT OVERTAKEN}$$

Additionally rule 13 is implicitly taken into account by requiring ReVolt to be NOT OVERTAKEN, as the overtaking vessel is the one that should keep out of the way. This expression only consider rules 13, 14 and 15, whilst the remaining rules described in section 2.5 are relevant as well. Hence these will be attempted followed by tuning the parameters described in the next section.

### 4.3.4 Hazard Evaluation Criterion

The hazard calculations is mostly based on three parameters. There are two collision hazard parameters taking different parts of the available information into considerations. The third is the COLREGS part of the hazard. The combination of these parameters can be expressed as the following cost function:

$$\mathcal{H}^k(t_0) = \max_i \max_{t \in D(t_0)} (\mathcal{C}_i^k(t)\mathcal{R}_i^k(t) + \kappa_i \mu_i^k(t)) + f(\mathcal{P}^k, \chi_{ca}^k) \tag{4.3}$$

where $t_0$ is the current time and $\kappa$ is a tuning parameter. The remaining parameters will be further described below.

The first part of the equation is composed of $\mathcal{C}$, $\mathcal{R}$ and $\mu$ representing the risk of collision. $\mathcal{R}$ is the collision risk factor given as:

$$\mathcal{R}_i^k(t) = \begin{cases} \dfrac{1}{|t - t_0|^p}(\dfrac{d_i^{safe}}{d_{o,i}^k(t)})^q, & if\, d_{o,i}^k(t) \leq d_i^{safe} \\ 0, & otherwise \end{cases} \tag{4.4}$$

where $t_0$ is the current time, and $t > t_0$ is the time of the prediction. The risk factor is only calculated when the ships are inside the perimeter of $d_i^{safe}$. The value of $d_i^{safe}$ together with $q \geq 1$ must be chosen carefully to make the system comply with rule 16 of COLREGS. Implying that ReVolt will have to take actions preventing collision, along with staying well clear of the obstacles. $d_i^{safe}$ also incorporate staying away from ships that are fishing, sailing or appear to not be under command. The risk factor is additionally dependent on time, and will reduce the cost of risk appearing further into the future unlike more close in time hazards. The time dependence is weighted by $p \geq \frac{1}{2}$. Factoring time into the cost function is important as there is less time to act at close in time hazards. Further will the short-term prediction be more accurate than the long-term, based on utilization of a linear prediction function. Hence there should be put less emphasis on hazards further into the future, because of possible uncertainties, which is taken into account by the time dependent factor $\frac{1}{|t-t_0|^p}$.

The cost associated with collision, denoted $\mathcal{C}$, is the next part of the equation, and is calculated as:

$$\mathcal{C}_i^k(t) = K_i^{coll}(t)|\vec{v_o}^k(t) - \vec{v_i}(t)|^2 \tag{4.5}$$

This cost factor is scaled by the relative kinetic energy of ReVolt and an obstacle, and is most important if collision with all obstacles is unavoidable. It is weighted by $K_i^{coll}(t)$ which may depend of several different conditions, like obstacle size, the right to stay on and/or responsibility to keep out of the way.

Lastly $\mu_i^k$ represents the cost connected to COLREGS. It is a boolean variable weighted proportionally by $\kappa_i$.

The second part of the hazard equation is $f(\mathcal{P}^k, \chi_{ca}^k)$ which is given as:

$$(P, \chi_{ca}) = k_P(1 - P) + k_\chi \chi + \Delta_P(P - P_{last}) + \Delta_\chi(\chi_{ca} - \chi_{ca,last}) \tag{4.6}$$

where $\Delta_P$, $\chi$ and $\Delta_\chi$ are penalty functions and $k_p$ and $k_\chi$ are positive tuning parameters that influence the priority of keeping nominal speed and course. Hence this part of the hazard function is making sure there is not an unnecessary high offset from the nominal course and speed. It also make sure the ship get back to the original path after the collision hazard is over. $\chi$ and $\Delta_\chi$ are asymmetric to ensure compliance with COLREGS rules 14, 15 and 17, and are presented in more detail below. $f(\mathcal{P}^k, \chi_{ca}^k)$ favors a straight line drive with constant cruising speed, making actions taken more predictable for others that might be in near proximity of the ship. This favouring is utilized in the two last terms of $f(\mathcal{P}^k, \chi_{ca}^k)$ which make sure to not change control offset too often. Hence there have to be a significant change in cost for the ship to take action. This will also prevent oscillations.

$\Delta_\chi$ is called the course penalty function and forces the ship to favour turning to starboard side, fulfilled by having an asymmetric cost. This ensures an algorithm working towards COLREGS compliance. The function is stated below:

$$\Delta_\chi = \begin{cases} K_{\Delta_\chi,port}(\chi_{ca} - \chi_{ca,last})^2, & \textit{if turn to port} \\ K_{\Delta_\chi,starboard}(\chi_{ca} - \chi_{ca,last})^2, & \textit{if turn to starboard} \\ 0, & \textit{otherwise} \end{cases} \tag{4.7}$$

$\chi$ is similar to $\Delta_\chi$, but depends on the size of the offset instead of change in offset. This function started out symmetric, simply penalizing large offsets from nominal control. During implementations it was proven that this function sometimes overpowered other terms in the cost function, leaving us with strange and unwanted behaviour. Hence the need for the asymmetric function stated below. The added asymmetry works equally to $\Delta_\chi$, and improved the behaviour significantly.

$$\chi = \begin{cases} K_{\chi,port}(\chi_{ca})^2, & \textit{if turn to port} \\ K_{\chi,starboard}(\chi_{ca})^2, & \textit{if turn to starboard} \end{cases} \tag{4.8}$$

### 4.3.5 Implementations

The implementations in this project are based on the code written by Inger B. Hagen in her MSc thesis Collision Avoidance for ASVs Using Model Predictive Control [15]. The code is reused, and changed to fit with the already existing system on ReVolt. There were some challenges along the way as the original code is written with respect to the coordinate system east-north-up (ENU) while the ReVolt system is based on north-east-down (NED), hence some modifications was needed to join the two code bases correctly. Moreover the need to send correctly formatted information into the CAS, was a challenge.

As virtual objects are used in this project, instead of relying on working sensory systems, their path must be decided. The only input available is the initial values of heading, speed and position. For simplicity a straight line path was chosen for all obstacles, and the new position of the obstacle is calculated every time step using equations 4.1 and 4.2.

There are a lot of parameters affecting the COLAV algorithm, and to get the desired behaviour with respect to both risk and COLREGS they need to be tuned carefully. The parameters are presented in table 4.2 below, and are the ones chosen after a substantial amount of trial and error. They give an acceptable behaviour of the algorithm, which will be discussed more excessively later on.

| **Parameter** | **Value** | **Description** |
|---|---|---|
| $T$ | 300.0 [s] | Prediction horizon |
| $DT$ | 0.5 [s] | Time step used for trajectory prediction |
| $P$ | 1 | Weight on time to evaluation situation |
| $Q$ | 4 | Weight on distance to obstacle at evaluation time |
| $D_{CLOSE}$ | 200.0 [m] | Distance where COLREGS comply |
| $D_{SAFE}$ | 40.0 [m] | Distance to obstacles that is considered safe |
| $K_{COLL}$ | 0.1 | Weight on collision cost |
| $\phi_{AH}$ | 68.5 [deg] | Angle specifying if obstacle is ahead |
| $\phi_{OT}$ | 68.5 [deg] | Angle specifying if an obstacle overtaking the ship |
| $\phi_{HO}$ | 22.5 [deg] | Angle specifying if an obstacle head on the ship |
| $\phi_{CR}$ | 68.5 [deg] | Angle specifying if an obstacle crossing the ship |
| $\kappa$ | 3.0 | The cost of not complying with COLREGS |
| $K_P$ | 10 | Cost of having a speed offset from nominal speed |
| $K_{\chi SB}$ | 2.5 | Cost of course offset from nominal course to starboard side |
| $K_{\chi P}$ | 10 | Cost of course offset from nominal course to port side |
| $K_{\Delta_P}$ | 0.5 | Cost of changing speed |
| $K_{\Delta_{\chi SB}}$ | 0.5 | Cost of changing course to starboard side |
| $K_{\Delta_{\chi P}}$ | 0.9 | Cost of changing course to port side |

**Table 4.2:** Parameters in the COLAV algorithm used during testing

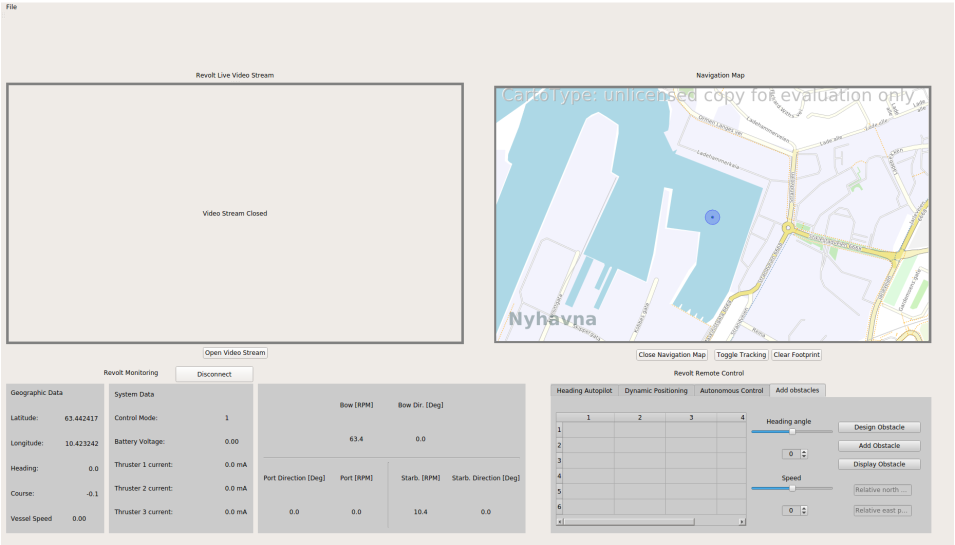
Chapter 5

# Simulation

All simulations were conducted in a simulator framework called CyberSea, provided by DNV GL. It provides a setup which gives a distinct separation between the different modules, letting the control system lie entirely outside the simulator. The advantage with this structure is that the control system should not be able to tell the difference between the real marine vessel and the simulated one. Testing is done using the exact code that will be run on actual ReVolt, together with the same Remote control system.

To set up the different test cases and send necessary initial information to CyberSea, the Remote control station, made with Qt [2], is used. The system was originally designed by Albert Havnegjerde in his master thesis Remote Control and Path Following for the ReVolt Model Ship [16] and is reused and enhanced to accommodate the needs of this project. The change done to the system is adding the possibility to design and display dynamic obstacles, making Qt usable for setting up COLAV test cases. Figure 6.5 show the user interface for the remote control station. A test case is designed by first drawing up the path for ReVolt to follow, done by right-clicking the map at the desired waypoint positions. Next the 'Design Obstacle' button is pushed, seen in figure 6.5, and the desired initial values are set. For this project only single obstacle cases were tested, but the system is designed to easily increase to an unlimited number of obstacles. All necessary scenario information is transferred to the control system via transmission control protocol (TCP).

The obstacles used for this project are virtual, removing the need for sensors like camera and lidar. This decreases the complexity of the problem, releasing time to focus on the COLAV algorithm instead of sensor fusion, object detection and tracking. Initial position is set relative to ReVolt using distance in meters in North and East directions. Speed and heading are also chosen, and kept constant throughout the scenario. The obstacles behave in a straightforward manner and go in a straight line, keeping constant speed and heading, no matter what. Hence the objects will not perform evasive maneuvers in cases where they are designated keep-way vessel, thus not following COLREGS. This leaves ReVolt with the full responsibility of avoiding collision.



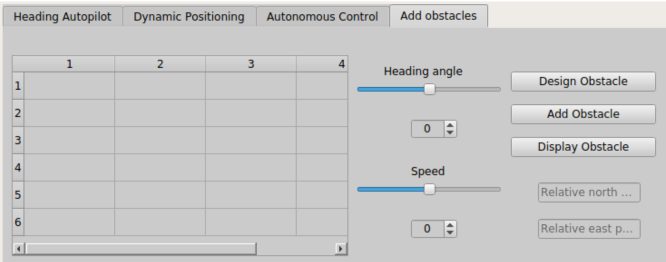
Geographic Data
Latitude: 63.442417
Longitude: 10.423242
Heading: 0.0
Course: -0.1
Vessel Speed 0.00

System Data
Control Mode: 1
Battery Voltage: 0.00
Thruster 1 current: 0.0 mA
Thruster 2 current: 0.0 mA
Thruster 3 current: 0.0 mA

| Bow [RPM] | Bow Dir. [Deg] |
|---|---|
| 63.4 | 0.0 |

| Port Direction [Deg] | Port [RPM] | Starb. [RPM] | Starb. Direction [Deg] |
|---|---|---|---|
| 0.0 | 0.0 | 10.4 | 0.0 |

**(a)** Overview of the remote control station.



Heading angle 0

Speed 0

**(b)** Closer look at the “Add obstacle” tab.

**Figure 5.1:** Remote control station designed in Qt.

As widely known, simulations will never be accurate enough to compete with real world testing. Hence there will always be some difference in behaviour, that is hard to predict. The simulations performed in this thesis is conducted in optimal conditions, with no weather, noise or other disturbances. The simulators allows for adding such conditions, but for the purpose of this project the most straightforward simulations were utilized, to focus on getting the desired behaviour. Bearing this in mind when examining results from testing on actual ReVolt.

All simulations are completed using the linear prediction, instead of Euler prediction as discussed in section 4.3.1. This simplification was mostly done to avoid the possible risk of ReVolt not being able to perform the heavier computations. As there were little time to test this theory, I chose linear prediction based on Inger B. Hagens conclusion in

[15]. She had concerns about run-time on the on-board computer, and therefor chose to evaluate linear prediction in comparison to Euler. In her simulation results for the simpler test cases, there are little to no difference between the two. Using Euler prediction is reserved for future work. Even though linear prediction is less accurate, it will be sufficient for the small scale testing conducted in this thesis, but would likely not be for full scale testing with many quickly moving obstacles.

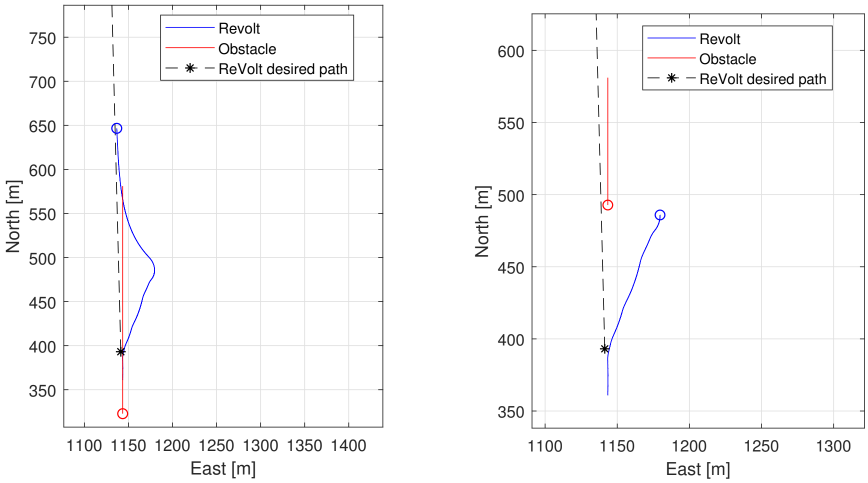
## 5.1 Head-on

Head-on situations are likely the simplest scenarios. To comply with COLREGS ReVolt should, as early as possible, make an evasive maneuver to starboard side. Both vessels are supposed to turn to their starboard side, but as the obstacle is trivial it will not act and continue straight forward. Two scenarios were constructed with different initial distance, respectively 300 and 600 meters. The purpose of this is to study the different phases of the scenario, seeing how it will react both close to collision and when there is no immediate risk. In this particular situation ReVolt move from south to north and the obstacle from north to south. Both ships keep a constant speed of $1\,\mathrm{m/s}$.

The results from the simulation with 300 meters initial distance are presented in figure 5.2. The over all performance is quite satisfactory, ReVolt avoids collision with minimum distance of about 37 meters. The change in course offset is step-wise until the obstacle have passed, then no offset is given to get back to nominal path immediately. This behaviour is expected as ReVolt get further away from nominal path the desired course angle from the LOS guidance (described in section 2.3) will increase accordingly. As this is not accounted for during calculations there is a need for a step-wise increase in course offset to keep following the predicted optimal path. The reason for the slight offset to the desired path at the end is the quite big look-ahead distance of 40 meters causing it to take some time before getting all the way back on track. The speed is kept constant through the entire scenario.

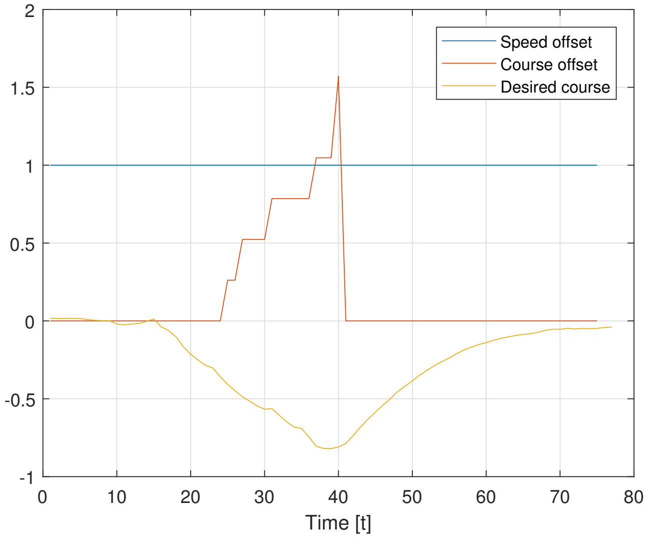
Figure 5.3 shows the results from the other head-on scenario starting at 600 meters distance. ReVolt avoids collision with a safe distance of about 43 meters, which is adequate. In the beginning of the case there are oscillations between zero and 30 degrees control offset, causing unwanted behaviour. When the ships gets close enough together the oscillations stops and the case is pretty much the same as the previous one, with 300 meters initial distance. After this the behaviour is as expected.

The reason for these oscillations originates in tuning. There have to be a balance between the prediction horizon and the parameter $d_{close}$ for this to work. If they are unbalanced ReVolt will start avoiding collision without taking COLREGS into account, and when the predicted path get within the perimeter of where COLREGS apply it might have to change its mind to comply with COLREGS. When the distance between the vessels is large this will happen on and off until they are so close that the unbalance between the parameter do not lead to a disturbance in the system.



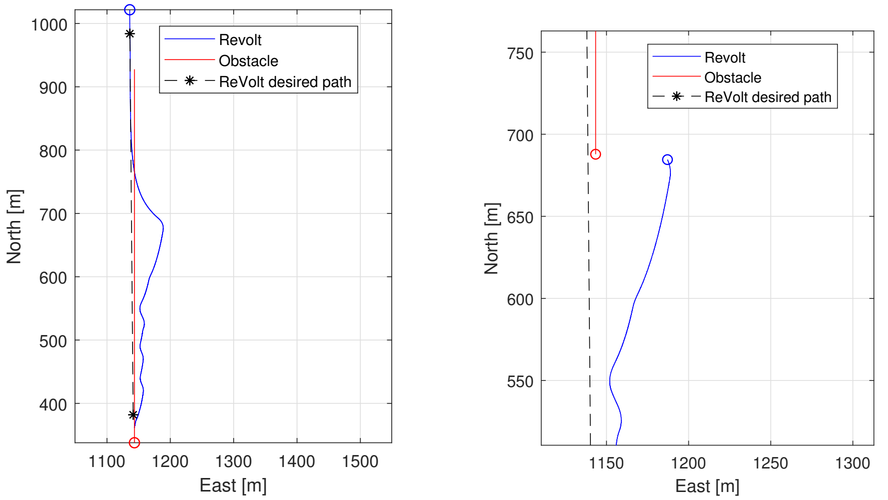
**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



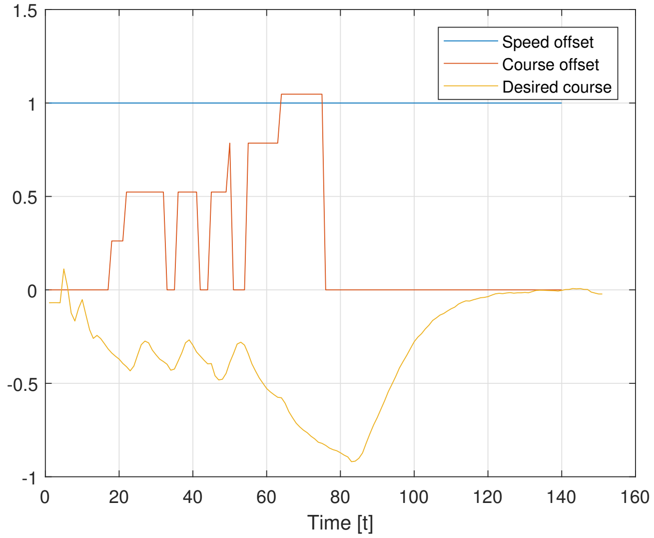
**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 5.2:** Head-on scenario with initial distance of 300 meters.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.

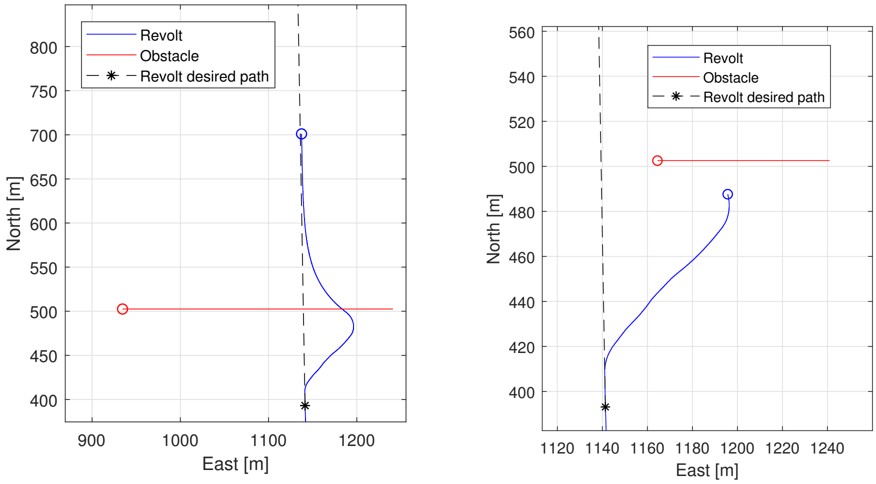


**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 5.3:** Head-on scenario with initial distance of 600 meters.

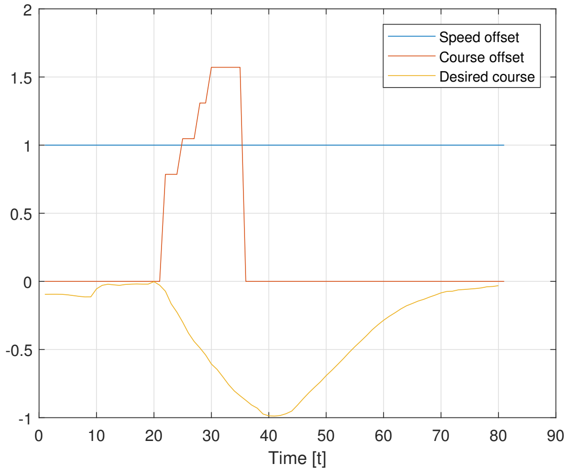
## 5.2 Crossing from starboard

In a crossing situation when the obstacles approaches ReVolt from starboard side, ReVolt will be named keep-way vessel and the obstacle stay-on vessel. ReVolt is sailing from south to north and the obstacles is coming from east going west. Both ships have a speed of $1\,\mathrm{m/s}$.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

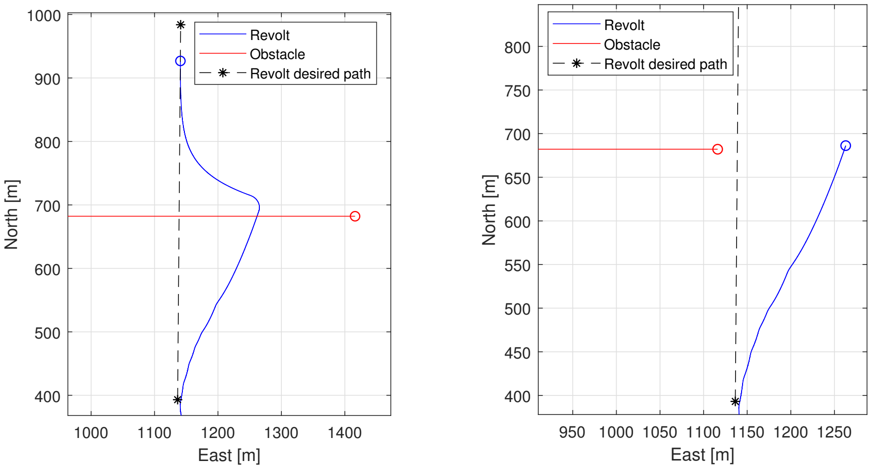
**Figure 5.4:** Crossing scenario from starboard side.

The correct actions to be taken in this situation according to COLREGS is that the keep-way vessel should pass behind the crossing vessel. Meaning ReVolt have to make an evasive maneuver to its starboard side. From figure 5.4 it is clear that that is exactly what is executed in the simulations, leading to a smooth change in path avoiding the collision with minimum distance of 34 meters. The course changes are predictable and ReVolt returns quickly back to nominal path, giving an overall satisfactory behaviour.

## 5.3 Crossing from port

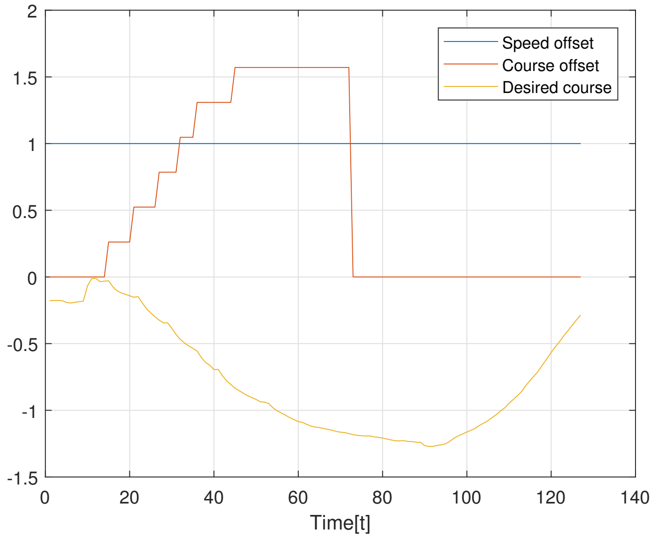
This scenario with the obstacle crossing in from port side is a more complex situation than crossing from starboard. The reason for this is that ReVolt now is the stand-on vessel and the obstacle should keep way, according to COLREGS. But as mentioned earlier the obstacles will just keep straight on, making no effort to avoid collision. Hence ReVolt has to take actions regardless. As the ships get closer to collision the cost of actually colliding will dominate the cost of obeying to COLREGS, forcing ReVolt to do an evasive maneuver.

In this particular case Revolt is sailing from south to north, and the obstacle is coming in from the west, going east. The nominal speed of both vessels are still $1\,\mathrm{m/s}$. The simulation results of this scenario is presented in figure 5.5 below. ReVolt end up avoiding collision by turning to starboard side and passing in front of the crossing obstacle. It is not trivial what the optimal course of action is here, but the one chosen by the algorithm is a good one as the minimal distance between the vessels is 147 meters, which is a safe distance. The most important part is that ReVolt take action early and stick with it. The act of being predictable is regarded less dangerous than for example crossing in front of the obstacle like carried out here. Furthermore, this particular action keeps opportunities open for the crossing obstacle to alter its course to starboard as it should have done in the first place. For ReVolt to change speed during this scenarios would just further complicate the situation.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

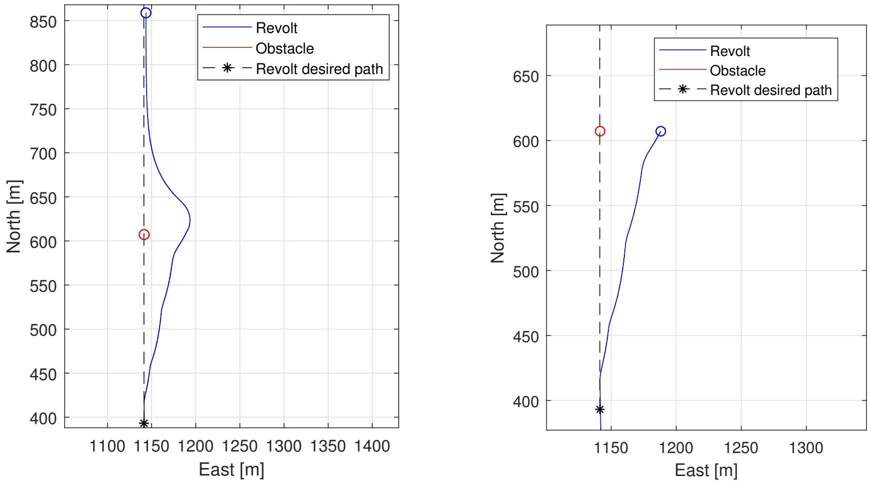
**Figure 5.5:** Crossing scenario from port side.

## 5.4 Overtaking

Overtaking can happen both ways, either ReVolt is overtaking the obstacle or the obstacle is overtaking ReVolt. Both cases will not be evaluated, due to ReVolt low top speed. Hence there might be more suitable to "overtake" a static object. According to COLREGS the vessel overtaking another vessel should be the one to keep well out of the way, and avoid collision. As the virtual obstacles still will not change course, ReVolt have to take actions in both cases. COLREGS do not specify which side of the obstacle passing should take place while overtaking, but as all other rules states to keep to starboard side, that would be the natural action here as well. The code is already implemented to favours offset to starboard side.

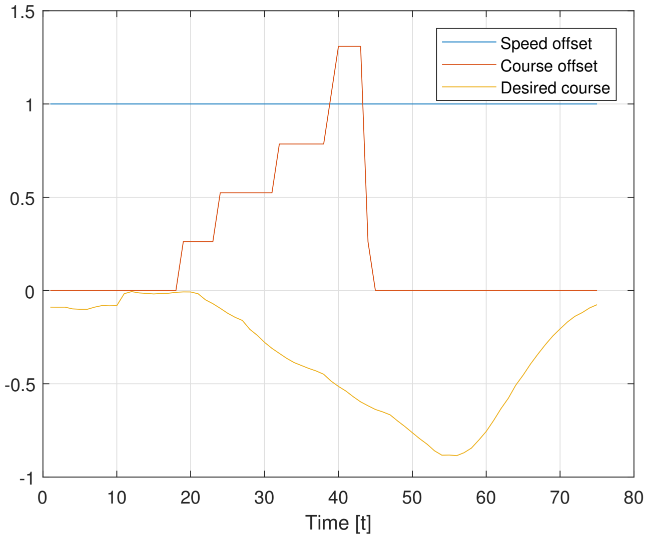
In this situation were ReVolt is overtaking the stationary obstacle ship, it is the one who should give way. ReVolt is going north with nominal speed around $1.4\,\mathrm{m/s}$, and the obstacle is staying put at an initial distance of 200 meters in front of ReVolt. Figure 5.6 shows the results where ReVolt takes action to starboard side keeping out of the way with minimum distance between the ships of about 40 meters. The result is as expected, and should easily translate into overtaking slowly moving obstacles.

When the situations is the other way around and the obstacle is overtaking ReVolt, the obstacle should avoid collision. But as that will not happen ReVolt once again have to take the necessary action to avoid collision. The scenario is equal to the last, but this time the speed of ReVolt is $1\,\mathrm{m/s}$ and the obstacle sails at $3\,\mathrm{m/s}$. This case is again split into two different scenarios with initial distance of respectively 300 and 600 meters. Figure 5.7 show the first case, where the behaviour is as expected and collision is avoided with a minimum distance of 45 meters. ReVolt also manages to get back to the original path. Figure 5.8 shows the larger case with 600 meters initial distance. Here as with the head-on situation there are initial oscillations until a certain distance between the ship is reached. From that point on it acts as in the smaller case, avoiding collision at at safe distance of 43 meters, and getting back on nominal path. This is caused by the same parameter unbalance described in the head-on section (5.1).



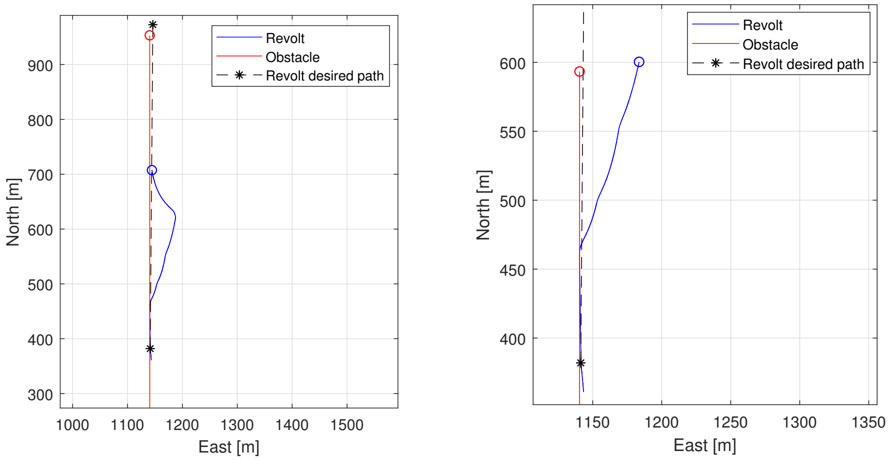
**(a)** Plot of the positions and desired paths of ReVolt and position of the obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



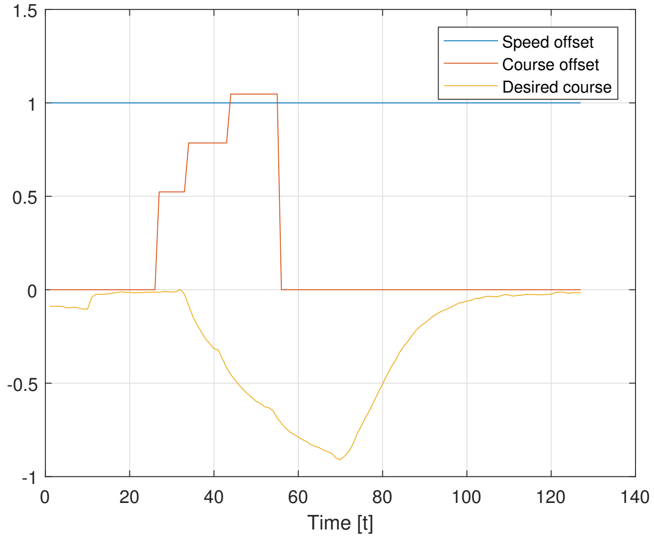
**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 5.6:** Overtaking scenario with obstacle initial position 200 meters in front of ReVolt.



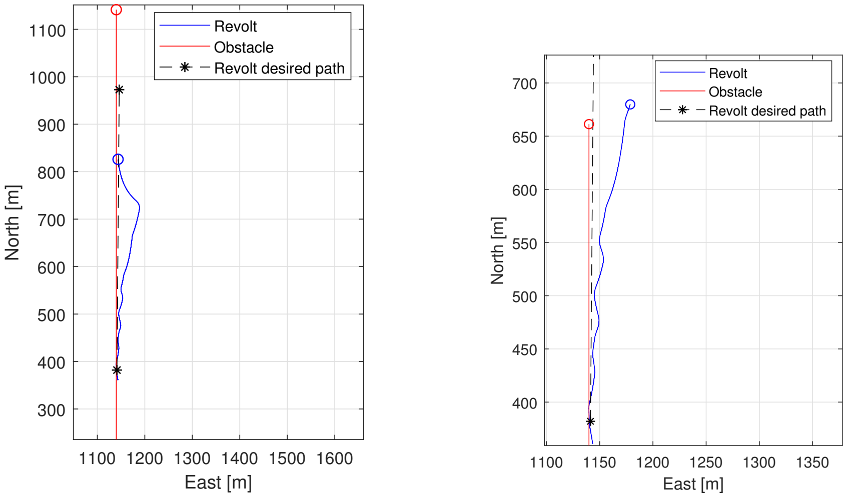
**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



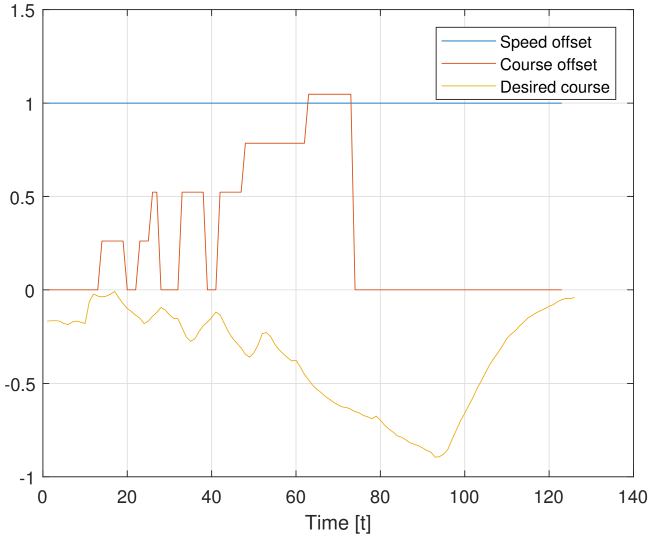
**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 5.7:** Overtaken scenarios with initial distance of 300 meters.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 5.8:** Overtaken scenarios with initial distance of 600 meters.

## Chapter 6

# Experiment

This section describes the experiments conducted with the physical ReVolt. This includes on-land preparations, tuning of heading controller, validation of the guidance system as well as the actual COLAV testing.

## 6.1 Test plan

The plan for testing was quite simple and the goal was to get proof of concept. Testing as many cases as possible to get validation of the obtained results from simulations. During testing performed prior to this project, problems with the heading controller was discovered, and hence it needed to be assessed before further testing could start. Satisfactory performance of the guidance system also had to be confirmed.

All test cases performed during the experiments were composed of one obstacle, sailing at constant speed and course. The prepared scenarios where head-on, crossing from both starboard and port side as well as overtaking. Testing was planned to begin in Dorabassenget (marked test area 1 in figure 6.1) and after initial verification moved outside the breakwater towards Munkholmen (marked test area 2 in figure 6.1) to have more room. To be able to interact and watch over ReVolt while testing a following boat borrowed from Gunnerus was utilized.

**Figure 6.1:** Map showing utilized test area in Trondheimsfjorden. [1]

**Figure 6.2:** Launched ReVolt ready for testing. Accompanied by the R/V Gunnerus Workboat, used as following boat.

## 6.2 Preparations

Before the actual sea trials could begin there was some work needed to be done on ReVolt. Last time ReVolt was tested was late August 2018 and since then some changes had been done. Several other students were also working on ReVolt simultaneously, making unknown changes. Hence we discovered trouble with both the GPS and IMU when performing system tests before launching ReVolt into the water. The IMU was not working at all, resulting in more troubles throughout the system. As the COLAV algorithm is not dependent on attitude, getting all other necessary data from the GPS, including yaw angle, the IMU was removed from the code. GPS troubles were connected to missing signals, solved by activating them manually. In addition to this the RC controller entered synchronization mode, which we were not able to get out of due to lack of equipment. For these reasons testing were delayed, but after fixing the controller all problems were solved and ReVolt was ready.

Experiments were conducted on November 15th 2018 at Dorabassenget and just outside the breakwater by Munkholmen. Before the actual COLAV tests could begin it was necessary to tune the heading controller as well as making sure the guidance system, designed by Albert Havnegjerde [16], was still working satisfactory. Tuning the heading controller was necessary as ReVolt got new motors this summer, and some other changes has been done both physically and in the code. After an extensive amount of time used on tuning, we had to move on due to the possible lack of daylight and battery power. The result was a slightly under damped heading controller, which with more time should be tuned to perfection for more accurate results.

**Figure 6.3:** Me programming changes to ReVolt, while its raining. Protecting the computer from water damage with a Jervenduk.

Then we moved on to validate that the Line-of-Sight guidance law was working properly. Firstly there were some troubles with ReVolt oscillating back and fourth over the path, which most likely mainly comes from the not perfectly tuned heading controller. It could also result from parameters needing to be tuned internally in the guidance law. We tried changing the look-ahead distance from 40m to 12m which gave a more satisfactory result, but still with some oscillations. With lack of time and knowledge about the guidance law system, we accepted it be and moved on to the main COLAV tests. It would have taken to much time to tune all parameters to perfection, and there were no major issues as ReVolt overall was following the desired path. The results are presented in figure 6.4.

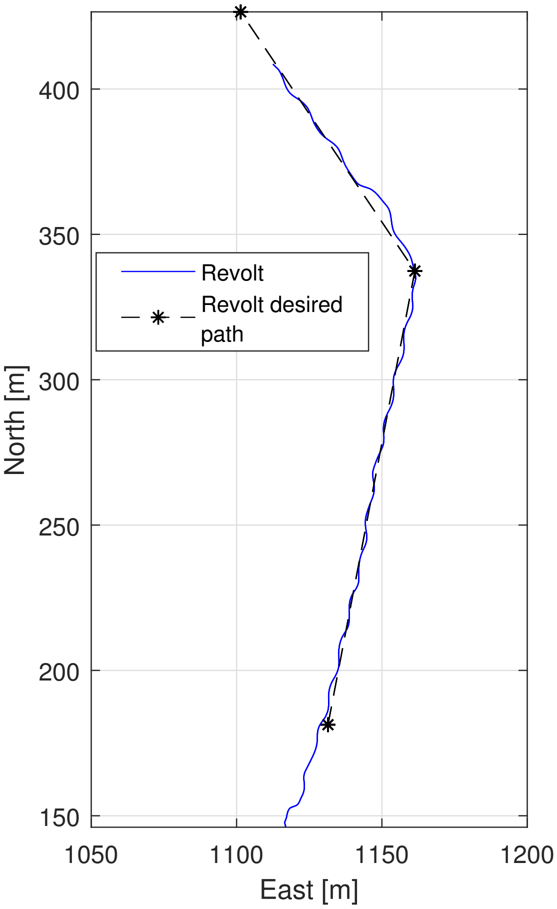


**Figure 6.4:** Resulting behaviour of the guidance system after reducing the look-ahead distance down to 12 meters.

## 6.3 Tests

Most of the experiments were conducted inside Dorabassenget which is sub-optimal as there is too little space to work with. The reason for this was the weather, there was too much waves outside the breakwater for ReVolt to sail safely. Until close to the end of the day we did not have the opportunity to test greater distances in our test cases. Getting towards Munkholmen there was only just enough battery left for three more scenarios.

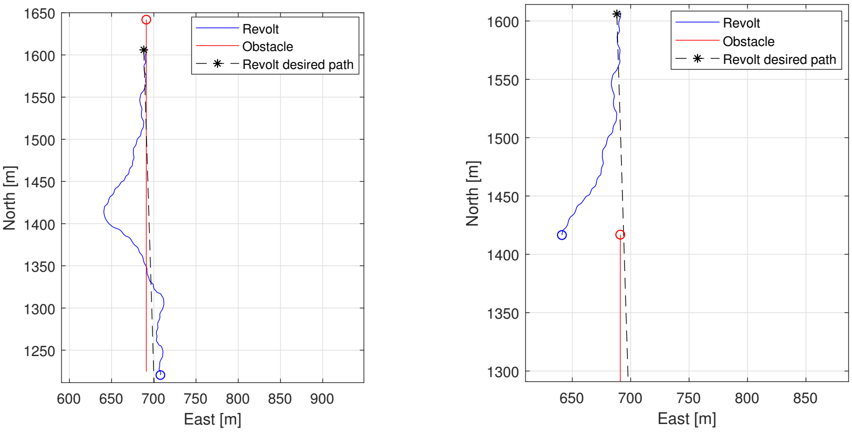
Even though we were able to test some scenarios outside the breakwater, there were still waves influencing ReVolt during sailing. When waves hit ReVolt from the front side, the forward speed get significantly reduced giving ReVolt less freedom of movement, hence collision avoidance could be hard to achieve.

**Figure 6.5:** Mid-testing, where ReVolt is running a COLAV scenarios carefully monitored in the remote control station.

### 6.3.1 Head-on

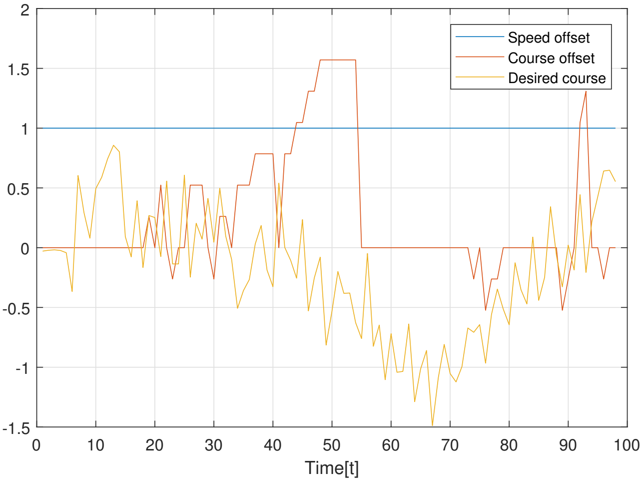
A head-on situation were ReVolt approximately wants to go from North to East and the obstacle in the opposite direction was constructed. ReVolt sails at nominal speed of $1\,\mathrm{m/s}$, same speed for the obstacle. Figure 6.6 show the desired path of ReVolt together with the actual path, and the position of the obstacle. There are some oscillations in the course all the way through this scenario, but the overall trend shows ReVolt following the path and avoiding collision with satisfactory distance of around 50 meters. From the offsets plotted in figure 6.6c there is a clear trend of positive course offset in the first half of the scenario.

Positive offsets corresponds to a starboard maneuver which is what we would expect for ReVolt to act in compliance with COLREGS. It is also evident from the plot that the desired course from the guidance law is oscillating. This will affect how the whole system works, and disguise how the SMBMP algorithm actually performs. After the collision risk has passed Revolt quickly returns to nominal path, apart from some oscillations in the aftermath of the scenario.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.

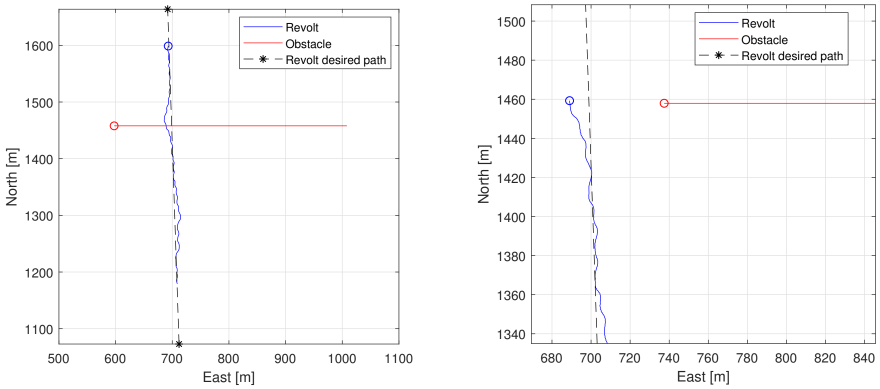


**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 6.6:** Head on scenario.

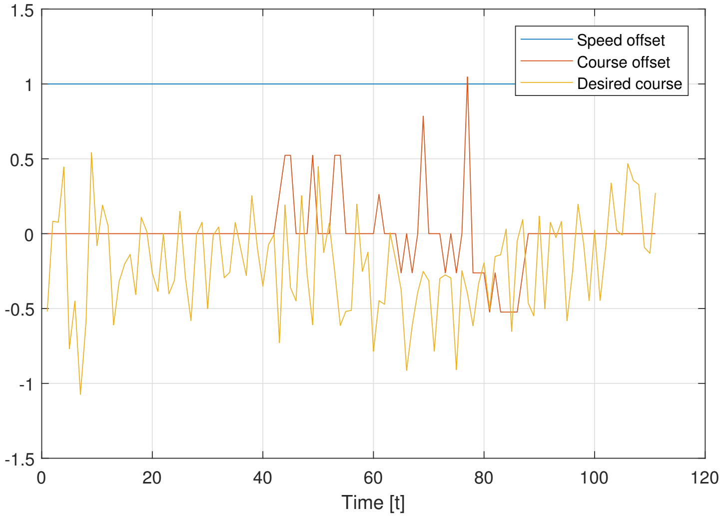
### 6.3.2 Crossing from starboard

Next scenario is a crossing situation where the obstacle approaches from starboard side. This makes ReVolt the give-way vessel and is therefor the one who should take action. COLREGS states that the give-way vessel should avoid risk by passing behind the crossing vessel. Revolt sails from south to north with speed of $1\,\mathrm{m/s}$ and the obstacle from east to west with the same speed.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 6.7:** Crossing scenario with obstacle approaching from starboard.

After the obstacle is detected there are more or less oscillations throughout the whole scenarios. The beginning trend is leaning towards starboard, but for some reason the vessel suddenly chooses to change directions and turn to port side instead. As the distance to collision decreases the vessel is really indecisive and change course offset back and forth between port and starboard side. All the way up until the risk of collision is so large that ReVolt is forced to pick a side, and hence picks port side crossing in front of the obstacle. Although the behaviour is unsatisfactory the minimum distance between the ships is no less than 45 meters, meaning ReVolt accomplishes the main goal of avoiding collision. On the other hand is the behaviour unexpected, unpredictable and not in compliance with COLREGS. After the hazard is over ReVolt returns to the desired path.

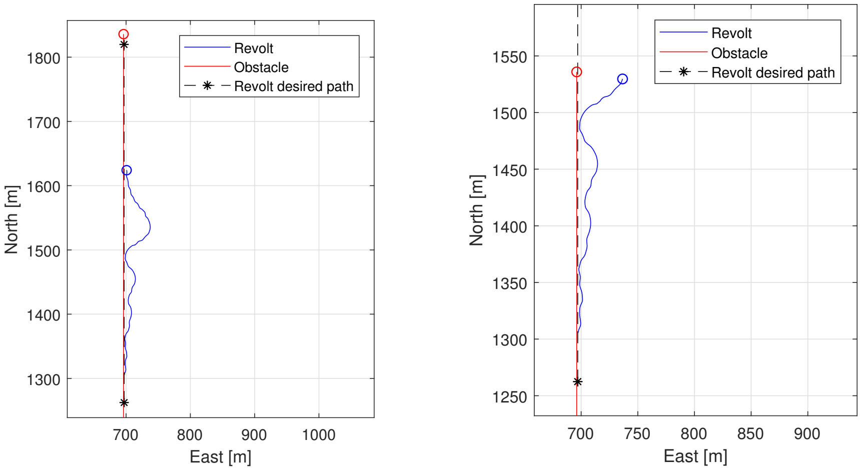
Because the batteries of ReVolt was running low there were not enough time to test crossing from port side. However in that case ReVolt would be the stand-on vessel and should in general not be the one taking action. But as the obstacle will not change path, ReVolt would have to act either way. The results would likely be somewhat similar to the crossing from starboard presented above. But as crossing from port side is a more complex situation, there is a likelihood that it would work even less satisfactory.

### 6.3.3 Overtaken

Overtaking situation can happen both when the vessel is overtaking someone else and when it is being overtaken. ReVolt is slow, hence the first scenario tested was ReVolt being overtaken by the obstacle. Both ReVolt and the obstacle are going from south to north. ReVolt have nominal speed of $1\,\mathrm{m/s}$ and the obstacle is sailing at $3\,\mathrm{m/s}$.

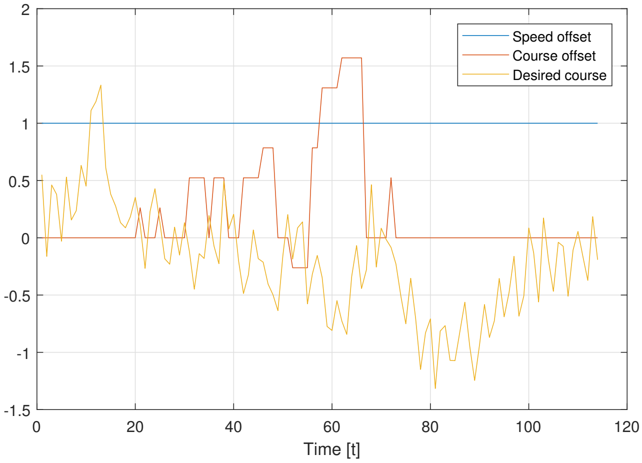
The obstacle's path is slightly west of the own-ship's path, as can be seen in figure 6.8. The reason for this is small inaccuracies in the manual setup of the test case, and it will not affect the case as they will still pass at quite a close distance of no action were taken. As the obstacle is the overtaking vessel, COLREGS states that it should keep out of the way on either side of ReVolt. However, the trivial obstacles in these experiments will not change course. This shows how ReVolt handles situations were it is not supposed to act, but the risk of collision gets too high.

Also this scenario causes some oscillations in ReVolts behaviour, but the overall tendency is course offset to ReVolts starboard side. And that behaviour is consistent through the scenarios except from one small spike the other way, but it is small enough for ReVolt to still stay om the starboard side of the vessel overtaking it. At the closest point the distance between the vessels is 42 meters. After the risk is over, ReVolt also comes back to the nominal path though it seems to have a small offset from nominal path.



**(a)** Plot of the positions and desired paths of ReVolt and obstacle.

**(b)** Plot of the closest position between ReVolt and the obstacle.



**(c)** Plot of ReVolts chosen offset in course (radians) and speed $(\mathrm{m/s})$, together with desired course (radians) from the guidance law.

**Figure 6.8:** Overtaking scenario where the obstacle is overtaking ReVolt.

Chapter 7

# Discussion

The simulation studies in Chapter 5 show that the SBMPC algorithm performs well on several different scenarios, by making a predictable and smooth evasive maneuver to avoid collision in all the cases tested. An issue that became evident during simulations was the appearance of oscillations when starting scenarios at greater distances. The reason for these oscillations are described in the simulations chapter, but is overall originating from poor tuning. Furthermore this leads to problems when the prediction is on the edge of where the COLREGS cost will activate, hence sequentially having true or false for $\mu$. This affects the cost too much, causing the algorithm to change course instead of keeping on. This behaviour will mostly be limited by better tuning, but to remove it completely there is need for a more complex cost function. This is due to that there might be other sources for oscillations as well, like ReVolt not knowing whether the obstacles has passed or not.

Although the simulation results presented seems to work quite perfectly, it should be taken into account that all simulations are done during perfect conditions, with no disturbances. For more realistic testing, simulations with non-ideal weather conditions added in should be performed. This will decrease the step from simulations to real world testing. More realistic testing is especially important for ReVolt, as the little scale model will be strongly affected by weather conditions. With a low maximum speed as low as $1.5\,\mathrm{m/s}$ ReVolts ability to avoid collision at all time should be discussed. When the bow is hit by waves ReVolt nearly moves forward, reducing the freedom of maneuverability. Furthermore reducing ReVolt ability to avoid collision, particularly with fast moving objects. This issue is only relevant for the scale model, as the full size concept ship will not be nearly as much affected by the environmental forces. Consequently testing in more calm waters is a reasonable adjustment to this challenge.

Moving on to the real life experiments described in Chapter 6, where the SBMPC algorithm clearly performs less optimal, as expected in advance. As a consequence of several obstructions along the way less testing than desired was actually carried out. There was only time for one full day of testing, leaving the end result as a first time trial of COLAV on

ReVolt, with no prior information of how it would respond. Factoring this into the equation are the obtained results quite satisfactory. The system worked mostly as expected, despite some abnormalities during the crossing scenarios in section 6.3.2. Oscillations occurring in all three test cases were expected, based on the result obtained during simulation. The amplification is caused by the existence of noise, weather conditions and the not perfectly tuned heading controller. Disregarding the oscillations, overall performance in both head-on- and overtaking scenarios are quite positive. The actions taken are clear, predictable and in compliance with COLREGS. Collision is also avoided with satisfactory safe distance in all three cases.

Change in course is easier for other ships to notice, consequently the algorithm prefers course change to avoid collision in stead of change in speed. This concept is proven in both simulation- and experimental results as the speed is kept constant in all cases.

During the experiments internal cost calculations in the optimization was not logged, which might be something worth looking into for further work. Logging calculated cost together with associated control offset, can help evaluate strange and unexpected behaviour. This would have been beneficial in order to conclude exactly on what went wrong during the crossing situation from section 6.3.2. However some theories would be that oscillations lead ReVolt to no other choice than turning port, as the COLREGS compliant action of turning starboard might have caused a collision. Other factors would be poor prediction on the basis of existence of noise and poor propulsion. The head-on scenario portrayed in figure 6.6 further present some strange behaviour, but this time after the collision hazards was supposed to be over. The reason for this is the cost function not taking into account whether the obstacle has passed the ship or not, leading to non zero collision cost in erroneous scenarios.

Another factor interfering with performance is the use of linear prediction instead of Euler. Not taking into account the fact that ReVolt do not turn instantaneously is causing the predicted paths to be off by more and more further into the prediction. This will further result in a smaller than expected offset relative to the path, as when desired heading increases ReVolt lean towards going parallel to the desired path. Keeping current control offset will hence not lead to a safe route, which will become evident as more predictions are done. Leaving ReVolt with less time to act and the need for more drastic change in course to avoid collision safely, enforcing unpredictable and unnecessary behaviour.

# Chapter 8

# Further work

The implementation of the SBMPC algorithm in this project is quite simple and there are a lot of potential for improvement, both with regards to tuning and the cost- and objective function. There are many issues to be resolved before the algorithm can be regarded ready for use outside the test-environment, starting with a more sophisticated cost function in cooperation with more advanced plant- and constraint models in the optimization problem.

First step towards an improved performance on real life experiments are simulation testing in more realistic environments, including environmental forces like waves, current and wind. This in conjunction with more extensive tuning will likely result in a more robust system. The development of a systematic method for parameter tuning will help with this. More experiments in general will moreover help map the challenges that need solving. Adding increased computational power to ReVolt also provide the opportunity to do optimization with Euler prediction, for more realistic results.

Further to increase the performance of the algorithm avoiding grounding and collision with land should be included, based on map data. Also navigational beacons should be considered, as well as other stationary obstacles. This will be crucial in short distance shipping, acting a lot in shallow water close to shore.

To be a stand-alone system there is of course a need for object detection, done by utilizing a sophisticated sensory system consisting of e.g. lidar and camera. This is another project in itself, and will also require the added computational power of a better on-board computer.

Chapter 9

# Conclusion

In this project a COLREGS compliant collision avoidance module have been implemented on the ReVolt test platform, using the simulation based model predictive control algorithm. The CAS have been tested through a number of different simulated scenarios, preforming well with short distance to collision, whilst experiencing some oscillations with greater distances caused by an unbalance in the tuning. All actions taken comply with COLREGS exploiting predictable and apparent behaviour.

The SBMPC algorithm has further been tested during one full day of sea trials. Revealing several stability and performance issues noted for future improvements. Temporary disregarding mentioned issues, do the results provide a proof of concept. Demonstrating that with future enhancements, will ReVolt be able to perform collision avoidance in a variety of situations, given acceptable weather conditions.

# Bibliography

[1] openstreetmap. `https://www.openstreetmap.org/export#map=15/63.4468/10.4114`. Accessed: 2018-12-16.

[2] Qt. `http://wiki.qt.io/About_Qt`. Accessed: 2018-09-17.

[3] *COLREGS - International Regulations for Preventing Collisions at Sea*. Lloyd's Register Rulefinder, 2005. Version 9.4.

[4] A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 2, pages 1384–1389. IEEE, 1998.

[5] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[6] M. Benjamin, J. Curcio, and J. Leonard. Navigation of unmanned marine vehicles in accordance with the rules of the road. Technical report, NAVAL SEA SYSTEMS COMMAND NEWPORT DIV RI, 2006.

[7] M. Breivik et al. Mpc-based mid-level collision avoidance for asvs using nonlinear programming. In *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, pages 766–772. IEEE, 2017.

[8] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[9] A. N. Cockcroft and J. N. F. Lameijer. *Guide to the collision avoidance rules*. Elsevier, 2003.

[10] J. Colito. *Autonomous mission planning and execution for unmanned surface vehicles in compliance with the marine rules of the road*. PhD thesis, University of Washington Washington, 2007.

[11] B.-O. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig. A modified dynamic window algorithm for horizontal collision avoidance for auvs. In *Control Applications (CCA), 2016 IEEE Conference on*, pages 499–506. IEEE, 2016.

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

[13] B. Foss and T. A. N. Heirung. Merging optimization and control. *Lecture Notes*, 2013.

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

[15] I. B. Hagen. Collision avoidance for asvs using model predictive control. Master's thesis, NTNU, 2017.

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

[17] T. A. Johansen. Toward dependable embedded model predictive control. *IEEE Systems Journal*, 11(2):1208–1219, 2017.

[18] 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, 2016.

[19] S.-M. Lee, K.-Y. Kwon, and J. Joh. A fuzzy logic for autonomous navigation of marine vehicles satisfying colreg guidelines. *International Journal of Control, Automation, and Systems*, 2(2):171–181, 2004.

[20] O. Levander. Autonomous ships on the high seas. *IEEE Spectrum*, 54(2):26–31, 2017.

[21] Ø. A. G. Loe. Collision avoidance for unmanned surface vehicles. Master's thesis, Institutt for teknisk kybernetikk, 2008.

[22] DNV GL. Revolt report. (2015-0170, Rev. 1), 2015.

[23] Miljødirektoratet. Klimagassutslipp fra transport. `http://www.miljostatus.no/tema/klima/norske-klimagassutslipp/utslipp-av-klimagasser-fra-transport/`. Accessed: 2018-11-29.

[24] Statistisk sentralbyrå. Økning i transportarbeidet. `https://www.ssb.no/transport-og-reiseliv/artikler-og-publikasjoner/okning-i-transportarbeidet`. Accessed: 2018-11-29.

[25] A. Richards and J. How. Decentralized model predictive control of cooperating uavs. In *43rd IEEE Conference on Decision and Control*, volume 4, pages 4286–4291. Citeseer, 2004.

[26] S. D. Sæter. Colregs compliant collision avoidance system for a wave and solar powered usv. Master's thesis, NTNU, 2018.

[27] M. Samadi and M. F. Othman. Global path planning for autonomous mobile robot using genetic algorithm. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on*, pages 726–730. IEEE, 2013.

[28] T. Shim, G. Adireddy, and H. Yuan. Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, 226(6):767–778, 2012.

[29] T. Statheros, G. Howells, and K. M. Maier. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *The Journal of Navigation*, 61(1):129–142, 2008.

[30] T. Stenersen. Guidance system for autonomous surface vehicles. Master's thesis, NTNU, 2015.

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

[32] K. Teo, K. W. Ong, and H. C. Lai. Obstacle detection, avoidance and anti collision for meredith auv. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–10. IEEE, 2009.

[33] S. F. T. Ukeblad. Dette skipet er utslippsfritt og har ingen mennesker ombord. `https://www.tu.no/artikler/dette-skipet-er-utslippsfritt-og-har-ingen-mennesker-ombord/231695`. Accessed: 2018-11-29.

[34] X. Wang, V. Yadav, and S. Balakrishnan. Cooperative uav formation flying with obstacle/collision avoidance. 2007.

[35] B. Yi, S. Gottschling, J. Ferdinand, N. Simm, F. Bonarens, and C. Stiller. Real time integrated vehicle dynamics control and trajectory planning with mpc for critical maneuvers. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 584–589. IEEE, 2016.

[36] Ørnulf Jan Rødseth. Skip uten kaptein bak roret. *SINTEF*, 2014.