Tonje Midjås

SBMPC Collision Avoidance for the ReVolt Model-Scale Ship

Master's thesis in Cybernetics and Robotics
Supervisor: Edmund Førland Brekke
June 2019

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

DNV·GL

Tonje Midjås

SBMPC Collision Avoidance for the ReVolt Model-Scale Ship

Master's thesis in Cybernetics and Robotics
Supervisor: Edmund Førland Brekke
June 2019

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

NTNU
Norwegian University of
Science and Technology

Problem Description

Autonomous surface vessels are dependent on a collision avoidance system for safe navigation. The scope of this thesis is to improve the performance of the simulation-based model predictive control (SBMPC) algorithm on ReVolt model-scale ship. A simple version of the algorithm does already exist on ReVolt, implemented during a 5$^{\text{th}}$ year specialization project, written in fall 2018. Where proof of concept was achieved, showing that ReVolt is able to perform collision avoidance. The system should adhere to the International Regulations for Preventing Collision at Sea (COLREGS) as well as avoid collision in a safe and predictable manner. This masters thesis will address the following tasks:

- Replace the ship model with one representing the actual vessel dynamics of ReVolt.
- Enhance the SBMPC algorithm to address oscillatory behaviour.
- Develop an anti-grounding system based on electronic navigational charts.
- Add simulation of automatic identification system (AIS) data and the necessary drivers for utilizing it.
- Perform more realistic simulations including environmental forces, more dynamical obstacles changing behaviour during scenarios and multi obstacle cases.
- Conduct real life experiments.

Preface

This thesis concludes my Master of Science degree in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). It is written in collaboration with DNV GL and as a continuation of the pre-project submitted during the fall semester of 2018. In the pre-project the SBMPC algorithm was introduced into ReVolt's existing code base, based on the implementation from the master thesis of I. B. Hagen [28], and proof of concept for collision avoidance was demonstrated. For this thesis, the more developed SBMPC library from the Autosea project (www.ntnu.edu/autosea) was adapted to the ReVolt platform. The simulator environment has been provided by DNV GL, with assistance from T. A. Pedersen. The physical scale model test platform was provided by DNV GL as well.

I would like to thank my supervisors Edmund F. Brekke at NTNU and Tom Arne Pedersen at DNV GL for their general guidance, help and support during the work of this thesis. An extra thanks to Tom Arne Pedersen for providing help with the simulator, arranging necessary equipment, always assisting with experiment and just generally answering questions at any time of the day. Thanks are also due to my co-supervisor Giorgio K. M. Kufoalor and Inger B. Hagen for help with implementation, fixing bugs and good discussions about acquired results. I am especially grateful for the help with structuring, proofreading and suggestions on the report from Edmund F. Brekke. Also thanks to Andreas B. Martinsen for help with carrying out experiments.

Lastly, I would like to thank my family for all help and support during my studies.

Tonje Midjås
Trondheim, June 2019

Abstract

Autonomy is the future of maritime shipping, and in order to be a viable solution for commercial use it has to be as safe or safer than conventional shipping. Fully autonomous surface vessels rely on a predictable collision avoidance system (CAS) to meet this criteria. The CAS should adapt as surroundings change and prepared for unexpected occurrences while adhering to the International Regulations for Preventing Collision at Sea (COLREGS). This thesis addresses a COLREGS compliant CAS using Scenario-Based Model Predictive Control (SBMPC) in conjunction with an anti-grounding system on the ReVolt model-scale ship.

A broad specter of realistic simulated scenarios, including static obstructions, unpredictable obstacles and environmental disturbances are evaluated as a preparation for real life testing. Necessary improvements to the method are implemented to address oscillatory behaviour. A more fitting ship model ensures good predictions along the way and a transitional cost help better comply with COLREGS. To avoid grounding, a map restriction based on electronic navigational charts is implemented and to detect approaching obstacles exploitation of automatic identification system (AIS) data is added. The improved CAS is tested both in simulator and in real life. The tests present promising results with regards to avoiding collision in a smooth predictable manner. To pursue optimal performance in all scenarios, future research may include improved tuning and additional complexity of the algorithm.

Sammendrag

Fremtiden innen maritim varetransport består av autonome skip, og for å bli en levedyktig kommersiell løsning må de lages like trygge eller tryggere enn bemannede farkoster. Fullstendig autonome fartøy som oppfyller dette kriteriet, er avhengig av et forutsigbart kollisjonsunngåelsessystem. Det må være et system som tilpasser seg ettersom omgivelsene endres, er forberedt på uventede hendelser og samtidig overholder de internasjonale forskriftene for forebygging av kollisjon på sjøen (COLREGS). Denne masteroppgaven omhandler et COLREGS-kompatibelt kollisjonsunngåelsessystem basert på Scenario-Based Model Predictive Control (SBMPC) og et anti-grunnstøtingssystem, begge installert på skalamodellen av ReVolt.

Et bredt spekter av realistiske simulerte situasjoner med statiske hindringer, uforutsigbare objekter og forstyrrelser fra omgivelsene har blitt evaluert som en forberedelse til virkelige sjøtester. For å håndtere oscillatorisk oppførsel har noen nødvendige utbedringer av metoden blitt implementert. En mer passende skipsmodell sikrer gode prediksjoner underveis, og transisjonskostnaden bidrar til å bedre overholde COLREGS. For å unngå grunnstøtingsulykker er det lagt inn restriksjoner basert på elektroniske sjøkart, og for å oppdage objekter som nærmer seg utnyttes AIS-data. Det utbedrede kollisjonsunngåelsessystemet er testet både i simulator og i virkeligheten, og gir lovende resultater med tanke på å unngå kollisjon på en forutsigbar måte. For å komme nærmere optimalt resultat i alle situasjoner kan fremtidig forskning omfatte justering av parametere, samt øke kompleksiteten i algoritmen.

Table of Contents

Abbreviations

AIS	=	Automatic Identification System
API	=	Application Programming Interface
ASCII	=	American Standard Code for Information Interchange
ASV	=	Autonomous Surface Vessel
AUV	=	Autonomous Underwater Vehicle
CAS	=	Collision Avoidance System
CG	=	Center of Gravity
CO	=	Center of Origin
COG	=	Course Over Ground
COLAV	=	Collision Avoidance
COLREGS	=	International Regulations for Preventing Collisions at Sea
CPA	=	Closest Point of Approach
DOF	=	Degrees Of Freedom
DW	=	Dynamic Window
ENC	=	Electronic Navigational Charts
ESC	=	Electronic Speed Controller
GPS	=	Global Positioning System
IHO	=	International Hydrographic Organization
IMO	=	International Maritime Organization
LIDAR	=	Light Detection And Ranging
LOS	=	Line of Sight
MPC	=	Model Predictive Control
NED	=	North East Down
NMEA	=	National Marine Electronics Association
RMC	=	Remote Control Station
ROS	=	Robot Operating System
ROT	=	Rate Of Turn
SBMPC	=	Simulation Based Model Predictive Control
SNAME	=	Society of Naval Architects and Marine Engineers
SOG	=	Speed Over Ground
TCP	=	Transmission Control Protocol
TSQ	=	Test Sequence
UDP	=	User Datagram Protocol
VHF	=	Very High Frequency
VO	=	Velocity Obstacle

Nomenclature

X, Y	=	Force in x and y direction
N	=	Moment in z direction
u, v, w	=	Linear velocity in x, y and z direction
r	=	Angular velocity in z-direction
x, y	=	Position in x and y direction
ψ	=	Heading (Yaw)
$\boldsymbol{\eta}$	=	Position and orientation vector
$\boldsymbol{\nu}$	=	Body-fixed linear and angular velocity vector
$\boldsymbol{R}$	=	Rotation matrix from body-frame to world frame
$\boldsymbol{M}$	=	Total system inertia matrix
$\boldsymbol{C}(\boldsymbol{\nu})$	=	Total coriolis and centripetal matrix
$\boldsymbol{D}(\boldsymbol{\nu})$	=	Damping matrix
$\boldsymbol{\tau}$	=	Generalized force vector
$\boldsymbol{I}_{xy}$	=	Product of inertia
y_g	=	Distance from center of origin to center of gravity
m	=	Vessel mass
I_z	=	Moment of inertia about the z-axis
χ	=	Course
χ_d	=	Desired course
α_k	=	Rotation between north and the desired path
e	=	Cross-track error
Δ	=	Lookahead distance
β	=	Sideslip angle
ψ_d	=	Desired heading
U	=	Speed over ground
$\hat{x}_i$	=	Estimated position in x-direction
$\hat{y}_i$	=	Estimated position in y-direction
$\hat{u}_i$	=	Estimated velocity in x-direction
$\hat{v}_i$	=	Estimated velocity in y-direction
t_0	=	Current time
ΔT	=	Step size
T	=	Prediction horizon
$\hat{\psi}_i$	=	Estimated heading angle
$\hat{r}_i$	=	Estimated yaw rate
u_d	=	Desired speed

K_{p_u}	=	Control parameter affecting u
K_{p_ψ}	=	Control parameter affecting heading
K_{d_ψ}	=	Control parameter affecting yaw
rudder$_d$	=	Distance from the rudder to center of gravity
$k^*(t_0)$	=	Selected best control behaviour
$\mathcal{H}^k(t_0)$	=	Cost function
$\mathcal{C}$	=	Cost associated with collision
$\mathcal{R}$	=	Collision risk factor
$\mathcal{M}$	=	COLREGS violation cost
$\mathcal{T}$	=	COLREGS transitional cost
$f(\mathcal{P}, \chi_{ca})$	=	Maneuvering cost
κ	=	Tuning parameter affecting $\mathcal{M}$
λ	=	Tuning parameter affecting $\mathcal{T}$
Δ_χ	=	Course penalty function
χ_s	=	Size of course penalty function
Δ_P	=	Speed penalty function
$K_i^{coll}(t)$	=	Weight on $\mathcal{C}$
d_{safe}^i	=	A chosen minimum distance between a vessel and an obstacle
d_o, i^k	=	Current distance between own-ship and obstacle
$\vec{v_o}^k(t)$	=	Velocity of own-ship
$\vec{v_i}(t)$	=	Velocity of obstacle
$\vec{L_i}^k(t)$	=	Unit vector in line-of-sight direction from own-ship to obstacle
d_i^{cl}	=	Largest distance where COLREGS apply
$O_i^k(t)$	=	Binary indicator of overtaking
$Q_i^k(t)$	=	Binary indicator of being overtaken
$X_i^k(t)$	=	Binary indicator of crossing
$S_i^k(t)$	=	Binary indicator showing if the obstacle is on starboard side of the vessel or not
$\mathcal{P}^k$	=	Speed offset
χ_{ca}^k	=	Course offset

Chapter 1

Introduction

1.1 Motivation

Autonomous shipping is the future of the maritime industry [48] and will play a substantial part in the future of Norwegian industry. Norway has a long and proud history in shipping, leading all the way back to the vikings [3]. For the shipping industry to further evolve in a sustainable manner, it will have to keep up with new developments and adapt to become more efficient, economical and environmental friendly. As seaborne trade is responsibly for around 90% of all global trade [7], an autonomous surface vessel (ASV) like ReVolt has great potential. ReVolt is an autonomous shipping concept developed by the classification society DNV GL [47], and is designed for short sea shipping along the Norwegian coast. Although the majority of international trade is done by sea, road transport is still the leading form of transportation for short distances [32]. ReVolt is designed with the aim to help move freight from the road to the sea in a cost effective manner.

Automation will play a huge part in reducing cost and increasing efficiency, safety and reliability of short sea shipping, as well as shipping in general. Removing human interaction reduces the risk of accidents drastically as about 80 % of all accidents are caused by human error [49]. Keeping in mind that autonomy may introduce new potential sources of failure, state of the art computer systems are typically superior to humans [46]. Humans are restricted in how much they are able to comprehend at the same time, whereas computers are able to take every different factor affecting the scenario into account simultaneously which is essential for safe automation. For the purpose of testing such autonomous systems a 1:20 scale model of ReVolt has been built. It has previously been fitted with systems such as dynamic positioning [10], guidance and path following [30] working towards complete automation. Hereafter, if not otherwise specified, will ReVolt refer to the physical scale model (more information about ReVolt is found in chapter 5).

Autonomous ships have come a long way, and good guidance, navigation and control systems are in use on ships today. Autopilots and dynamical positioning systems are becoming more and more common as well. A natural next step towards a fully autonomous systems is collision avoidance (COLAV) and anti-grounding, both crucial parts for ASVs to be able to operate in waters with other vessels and navigating close to the coastline. To be a viable solution for commercial use ASVs have to be as safe or safer than conventional shipping. Before a big shift can happen in the industry, more extensive simulations and real life testing has to be done removing all doubt that the ASV will be able to avoid collision. Achieving this is a significant step towards autonomous shipping and the first fully autonomous shipping vessels will likely be here before we know it.

1.2 Previous work

Ship navigation has been performed entirely as a human endeavour for many years. As new technology entered the picture in the decade after World War II, the discussion on optimal strategies for evasive maneuvers in close range encounters was started [55]. Studies on determining the point of collision based on data from radars was performed and as ship collision became an increasing problem through the sixties, more and more collision warning systems were designed. This was also the reason for the Convention on the International Regulations for Preventing Collisions at Sea (COLREGS), a set of rules to be followed by ships and other vessels created in 1972 [16]. The action taken to avoid collision was still dependent on the captains evaluation of the situation, and it was evident that in scenarios including three or more ships these actions became unpredictable. At this time the International Maritime Organization (IMO) introduced traffic separation reducing the number of collisions between ships heading in the opposite direction, especially in heavily trafficked areas around ports. This of course only partly solved the problem and there was still need for more sophisticated ways of avoiding collision in all thinkable situations [55].

In 1975 Goodwin proposed a way of discretely dividing the ship domain into three zones: give way, stand on and overtaking dependent on the angle of approach [27]. This concept was a building stone for several other studies like [19] in 1980 where an action domain was added, a circle where the necessity of an evasive maneuver was considered and Coldwell [17] in 1983 described a scenario-based ship domain for navigating in restricted waters. Around this time, more an more advanced approaches to collision avoidance surfaced, and the area of researched could be divided into two main categories: “Mathematical models and algorithms” and “Soft computing - Evolutionary algorithms, neural networks and fuzzy logic”. The focus will further be placed on the first one, including a mathematical description of the ship and a measuring algorithm that solved the collision problem [52].

The mathematical approach can further be split into global and local methods. In [50] collision avoidance methods based on both local and global path planning algorithms are discussed. The global methods can only handle static environments and hence the local method should be used in the dynamic environment representing the ocean, as it only focuses on a small area doing the necessary processing in a short enough amount of time. As real life scenarios include both static and dynamic obstacles, a combination of the two methods will ensure collision free travel from start to finish. Velocity Obstacle (VO) is a purely local and simple method first introduced in robot motion planning by Fiorini and Shiller [22]. Several modifications has since been introduced and Kuwata [38] presented VO for use in collision avoidance on ASVs with COLREGS compliance, tested in real life experiments. Stenersen [53] also obtained promising results with full-scale testing of the VO COLAV system. 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 [39]. It too performed well in both simulations and the full-scale experiments. Another modified DW algorithms used on autonomous underwater vehicles (AUVs) is presented in [21]. VO and DW are only two out of numerous examples of algorithms able to perform maritime collision avoidance.

Most currently existing COLAV methods lack the ability to handle dense traffic with multiple dynamical obstacles, whilst considering both the ship model and the environmental disturbances according to Johansen et. al. [35]. Furthermore including such complex scenarios into already existing algorithms would be non-trivial. Instead Johansen is proposing a possible solution based on model predictive control (MPC), where a ship model is used to predict the ship's trajectories. The method is called scenario based MPC. The predictions are done with a finite set of possible control behaviours, which are evaluated based on collision risk, hazard, compliance with COLREGS, operational constraints and objectives. [29] describes an improved version of the algorithm proposed by Johansen, showing promising result during real life experiments.

Considerable amounts of previous work have been performed on ReVolt increasing the level of automation and making it ready for implementation of a collision avoidance system. H. Alfheim and K. Muggerud performed several tests to improve knowledge of the parameters describing Revolt, as well as implementing dynamic positioning (DP) [10]. P. Minne implemented a framework for automatic testing in the simulator associated with ReVolt [42]. E. Henriksen further used this framework to test obstacle tracking and collision avoidance using simple implementations of VO and scenario-based MPC, all tested only in simulations [31]. A digital-twin for ReVolt was designed by A. Danielsen [18] and V. Kamsvåg worked on a sensor fusion system meant for sense-and-avoid purposes [36]. Finely A. Havngjerdet designed a guidance system enabling ReVolt to follow a predefined path [30].

1.3 Contributions

The author's specialization project leading up to this master thesis introduced a simple version of the SBMPC method onto ReVolt. This method was tested with purely virtual obstacles, in simple test scenarios, both in simulator and real life experiments, giving a first impression of how the ReVolt would handle collision avoidance. Level of performance was quite low, especially at sea, creating the opportunity for a wide variety of improvements to be added. The main contributions of this thesis are:

- Enhancements of the SBMPC algorithm to alleviate oscillatory behaviour. The main improvement is the added transitional cost, adapted from [29].
- A new ship model representing ReVolt's slow vessel dynamics.
- Implementation of a receiver and decoder for simulated AIS data, enabling it to be used while testing.
- Development of an anti-grounding system based on relevant extracted data from electronic navigational charts.
- Results from realistic simulations including environmental forces, map restriction, more dynamical obstacle and multi-obstacle cases.
- Results from real life collision avoidance experiments in Trondheimsfjorden.

1.4 Outline

Chapter 2 provides the necessary theoretical background needed for this thesis. The SB-MPC algorithm and the implementation of it are described in **Chapter 3** and **Chapter 4** explains the map extraction and anti-grounding system. In **Chapter 5** ReVolt, the experimental platform, is presented together with the accompanying simulator. The simulation study is presented in **Chapter 6**, showing both simple and more advanced test cases. Results from the real life experiments conducted are given in **Chapter 7**. A discussion of the results is provided in **Chapter 8**, whilst further work is proposed in **Chapter 9**. Finally followed by a conclusion in **Chapter 10**.

Chapter 2

Theory

The theory chapter is heavily based on the theory chapter in the author's specialization project [40], but this material is included here to make the master thesis self-contained. The speed, heading and LOS controllers are all from a previous master thesis [30]. The ship model has been advanced, removing most simplifications leading to a much more accurate model.

2.1 3DOF Ship model

This section presents a 3 degrees of freedom (DOF) maneuvering model, reduced from the complete 6 DOF general ship model. For the purpose of this thesis will motion in the horizontal plane be the main concern, meaning dynamics associated with roll p, pitch q and heave w will be neglected, giving $p = q = w = 0$. The motion components making up the resulting 3DOF model are surge, sway and yaw presented in equation 2.1 below. This model will be used during predictions in the collision avoidance method. Notation utilized is 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	ψ

Table 2.1: The notation of SNAME (1950) for marine vessels [24]. Only showing the relevant 3DOF.

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

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. $\boldsymbol{\tau}$ is the generalized force vector. $\boldsymbol{R}(\boldsymbol{\psi})$ is the rotation matrix transforming the body-fixed velocities into the world-fixed frame given by:

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

$\boldsymbol{\tau}$ 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 τ_X and τ_X are the forces along x- and y-axis, and τ_N is the moment about 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 be set in the center-line of the ship at the center of origin (CO) point, such that $y_g = 0$ where y_g is the distances from CO to center of gravity (CG) in y-direction.

Based on these assumptions we can define matrices $\boldsymbol{M} = \boldsymbol{M_A} + \boldsymbol{M_{RB}}$ and $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C_A}(\boldsymbol{\nu}) + \boldsymbol{C_{RB}}(\boldsymbol{\nu})$. 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. $\boldsymbol{M}$ is given as:

$$\boldsymbol{M} = \boldsymbol{M_A} + \boldsymbol{M_{RB}} = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} + \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \tag{2.4}$$

where m is the vessel mass and I_z is the moment of inertia about the z-axis and x_g is the distance from CO to CG in x-direction. The Coriolis matrix $\boldsymbol{C}(\boldsymbol{\nu})$ is also constructed from a rigid-body part and an added mass part.

$$
\begin{aligned}
\boldsymbol{C}(\boldsymbol{\nu}) &= \boldsymbol{C_A}(\boldsymbol{\nu}) + \boldsymbol{C_{RB}}(\boldsymbol{\nu}) \\
&= \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix}.
\end{aligned} \tag{2.5}
$$

The damping matrix is constructed by a linear and a nonlinear part and is defined by $\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{D_L}\boldsymbol{\nu} + \boldsymbol{D_{NL}}(\boldsymbol{\nu})\boldsymbol{\nu}$. The linear and nonlinear matrices are defined respectively:

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

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

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 feedforward term and a reference model supplying desired velocity. Both controllers implemented on ReVolt are designed by Albert Havnegjerde in [30]. ReVolt is an underactuated vessel as only surge and yaw can be controlled directly, this is a consequence of the actuator setup of two rear thrusters also acting as a rudder. This will impact how the controllers are designed. There are also two controllers in the SBMPC algorithm used during prediction, a speed controller and a heading controller additionally explained in this section.

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 ψ_d is desired heading and r_d is desired yaw rate. These are both time-varying and supplied by a reference model, while ψ 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 consists 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.

Controllers in SBMPC

The two controllers used during predictions in SBMPC are a feedback-linearizing controller for speed, given by:

$$F_x = (-mv + Y_{\dot{v}}v + Y_{\dot{r}}r)r - (X_u + X_{|u|u}|u| + X_{uuu}u^2)u + K_{p,u}m(u_d - u)$$

where $K_{p,u}$ is the proportional gain of the controller and the remaining parameters are described in table 2.1 and section 2.1 above. The second controller is a conventional heading PD controller given by:

$$F_y = \frac{K_{p,\psi}I_z}{l_r}((\psi_d - \psi) - K_{d,\psi}r)$$

where l_r is the arm the yaw moment is acting on, I_z is the moment of inertia about the z-axis and $K_{p,\psi}$ and $K_{d,\psi}$ are the proportional and derivative gains respectively.

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 to 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 x_n which is the relative North axis. α_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 Δ is the lookahead distance. $\Delta > 0$ and a rule of thumb is to set it to 1.5-2.5 ship lengths [24]. From (2.16) it is evident that a small Δ yields a big $\chi_r(e)$, and hence a more aggressive convergence to the desired path. Taking ocean current into consideration we need to account for the sideslip-angle β. 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 β 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 [13].

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 [23].

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 [23].

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 gradient-based MPC unsuitable for COLAV. Therefor model formulation, discretization, control trajectory parameterization, constraints and objectives need to be considered carefully, along with issues like dependability [34]. 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 [12].

One approach to robust MPC is utilizing 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, see e.g. [11], but most approaches incorporates optimization over control parameters to enhance the performance. Using this method numerical optimization is completely avoided, 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 Preventing 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 ones we follow today have emerged from years of development and came in to effect in 1972. They are called the International Regulations for Preventing 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 [16]. In 2016 a Part F was supplemented, called verification of compliance with the provisions of the convention [9]. Part B is the most relevant for this thesis and more specific the rules 6, 8, 13, 14, 15, 16 and 17. The rest of this section provides an overview of these rules [8].

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 two 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

Collision Avoidance

For a vessel to become fully autonomous, a sophisticated collision avoidance and anti-grounding system is required. The system must be designed to produce safe and predictable maneuvers, 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 execution of predictable actions. COLREGS is the basis for designing a maritime collision avoidance algorithm, and a necessity for the ASV to coexist with other manned vessels.

Over the last few decades several approaches to this problem have been explored, including methods with and without COLREGS compliance, with different level of complexity. The approaches can mainly be split into reactive and proactive methods, where reactive respond to an event after it occurs, while proactive try to anticipate possible challenges in the future. Velocity obstacle (VO) [38] [53] and dynamic window (DW) [21] are short term, reactive methods that have a harder time directly including COLREGS. Short term methods only looks for the immediate best solution, while long term find a solution viable further into the future. As a side note, these algorithms can be modified to perform behaviour that resembles COLREGS compliance, but it is not an optimal approach. Nevertheless, algorithms like VO and DW works well as reactive methods, but with the larger problem at hand a proactive method is the goal. Nothing will ever be fully proactive as the future can not be predicted perfectly, but instead deliberate methods is what to aim for. Long term, deliberate algorithms have the potential to be more comprehensive, including COLREGS compliance with less fundamental challenges and should be a preliminary focus. Path planning with COLREGS compliance, where the path is biased when encountering an obstacle [43], is an example of this type if method. Soft programming based algorithm is another, that also have the potential to handle all these challenges, they are based on artificial intelligence and some examples are fuzzy logic and neural networks [52].

In addition to the minimum requirement of COLREGS compliance the algorithm should be able to take all thinkable features, objectives, hazards and disturbances into consideration. This way of looking at the COLAV problem have existed for a long time and T. Miloh and S.D Sharma realised it already in 1975, when they designed a maritime collision avoidance algorithm based on Differential Games [41], which showed that the powerful analytical theory of differential games could be applied to determine mathematically optimal evasive maneuver. They stated that the simplified model they used was not good enough and also recognised that the collision avoidance problem is dependent on so much more than just the relative bearing to the target, for instance tuning rate, speed ratio, range and objectives. Hence there is a need for an algorithm capable of taking all of these factors into account when calculating the optimal evasive maneuver, while simultaneously handling unexpected elements. Replacing the Differential Game theory with MPC is one way of achieving this.

3.1 COLAV based on model predictive control

Model predictive control (MPC) is a versatile tool as it incorporates the possibility to utilize mathematical models of both the systems and its surroundings. All known and measurable disturbances can be modeled and integrated into the system, increasing its performance. MPC have been employed in several different collision avoidance scenarios e.g. collision free UAV formation flight [15], adaptive cruise control with collision avoidance [51] as well as stabilization and collision avoidance during emergency scenarios [26]. The amount of articles where MPC is used for COLAV in e.g. aerial and automotive domains increase the belief in the method's capabilities [25] [14].

In the scenarios indicated above, different variations of the MPC method were used. Regarding maritime collision avoidance, especially complying with COLREGS, there was less available content. One approach is a mid-level COLAV system using nonlinear programming designed by B. O. Eriksen [20]. Apart from that all articles found on the subject were descendants of Johansens article "Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment" [35]. The collision avoidance method used in this article is named Scenario-Based Model Predictive Control (SBMPC) and is the background for several MPC-based maritime COLAV algorithms. An implementation of the method was made by I. B. Hagen in her master thesis [28], which focuses on improving performance and robustness of maritime COLAV as well as reducing the dependability of knowing the exact guidance scheme. S. D. Sæter adapted this implementation to fit a system with less freedom to change propulsion [54]. D. K. M Kufoalor has further developed the SBMPC concept from [35] and [28] together with I. B. Hagen, E. F. Brekke and T. A. Johansen in [29] which focuses on making the maneuvers more predictable, thus more in compliance with COLREGS. Focus is also placed on a viable approach for incorporating COLAV strategies into existing guidance and control systems on marine vessels. This article further proposes some additional improvements to the SBMPC algorithm, which will be the base for some of the improvements implemented in this thesis. The latest addition is a paper by Kufoalor addressing the challenges related to maritime radar tracking with SBMPC [37], which will not be considered in this thesis, but is important for future development.

3.2 Scenario-Based Model Predictive Control

Scenario-Based Model Predictive Control (SBMPC), as the name suggest, is an MPC-based algorithm using simulation of a finite set of possible scenarios to calculate the optimal control output. The following section will describe the algorithm and is based on the pre-project [40] leading up to this thesis, featuring the same basis algorithm.

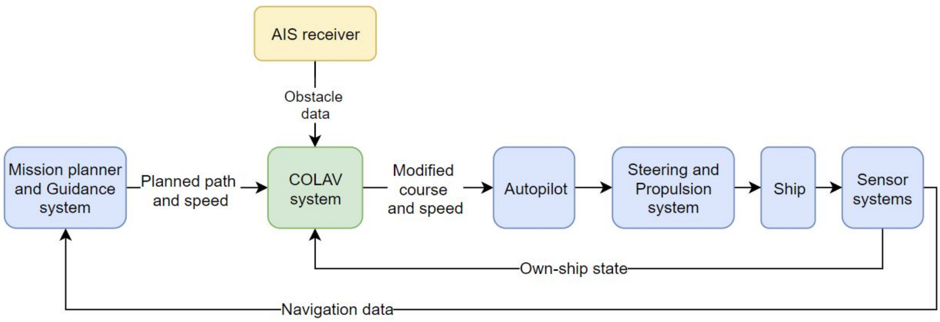


Figure 3.1: System architecture and information flow including SBMPC. Blue boxes represents the already existing code-base on ReVolt. Green is the added SBMPC structure and the yellow is the AIS receiver collecting obstacle data. Inspired by [35].

SBMPC is implemented as a module separate from both the mission planner and the autopilot. Figure 3.1 illustrates the overall structure of a system using SBMPC. The SBMPC module can be added to already existing systems, an important factor on the way to achieve fully autonomous surface vessels. Moreover we recognize from figure 3.1 that the module is dependent on a set of information to connect with the rest of the system. 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

The SBMPC algorithm is realized by a finite horizon and finite scenario minimization problem. Collision avoidance is complex and may yield non-convex optimization problems without a solution. This is where the finite scenario aspect takes effect, which makes the optimization problem deterministic and therefore always yields a result. To formulate the problem in this manner theory from robust MPC described in section 2.4 was applied, 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. The selection of the set of possible control offsets is a very important design decision.

Making SBMPC work for collision avoidance is in addition to the necessary available information, based on some other 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 to search 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 different paths to be evaluated are the product of a course offset and an adjustment to the speed, which together defines a scenario in SBMPC. All the scenarios are evaluated and a cost is calculated based on collision hazard, compliance with COLREGS and map restrictions. Then the scenario incurring 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 evaluate collision hazard. The general overview of the COLAV module is as follows:

1. For each pair of course and speed offset, the trajectory of the own-ship is predicted. Obstacle path is predicted once.

2. The cost function is applied to the predicted own-ship trajectories together with predicted obstacle paths, calculating the cost associated with the set of control behaviours.

3. Choose the control behaviour set corresponding to the lowest cost, and apply the first control input.

4. Repeat regularly.

3.2.1 Trajectory predictions

Predictions of the future path of both ReVolt and all obstacles are necessary for the algorithm to calculate cost. The prediction can be done in two ways: Linear- and Euler prediction. Euler prediction refer to prediction performed by using Euler integration. Linear prediction is given by the following equations:

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos(\hat{\psi}_i) & -\sin(\hat{\psi}_i) \\ \sin(\hat{\psi}_i) & \cos(\hat{\psi}_i) \end{bmatrix} \tag{3.1}$$

$$\hat{x}_{i+1} = \hat{x}_i + (t - t_0) \cdot (r_{11} \cdot \hat{u}_i + r_{12} \cdot \hat{v}_i) \tag{3.2}$$

$$\hat{y}_{i+1} = \hat{y}_i + (t - t_0) \cdot (r_{21} \cdot \hat{u}_i + r_{22} \cdot \hat{v}_i) \tag{3.3}$$

where R is a rotation matrix used to rotate the velocities into x- and y-direction. ψ_i is the heading, 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. $\hat{x}$ and $\hat{y}$ are predicted position in respectively x- and y-direction. These are simple calculations only taking the current speed and position into account, not relaying on an actual model of the boat. This is suitable for the obstacles, as we do not know the model describing them. This is a known weakness in the prediction of the obstacle paths, but since it is known we can account for it by for example weighting collision risk at the beginning of the predicted path heavier than further out, as the first part of the path will be most accurate. Further will linear prediction not be good enough for predicting ReVolt's future path, as the ship model is not taken into account and neither is the turning rate. It also assumes that change in heading happens from one time step to the next, which is unrealistic for ReVolt. ReVolt is a slowly moving ship with overall slower vessel dynamic, hence to improve performance of the algorithm a more accurate model in combination with Euler prediction was necessary.

One advantages of MPC is the possibility to include all the vessel dynamics, steering and propulsion systems as well as weather-, wind- and ocean current information. Having an accurate model of the own-ship is most important. The basis of the model used is described in theory section 2.1. The CyberSea simulator (described in section 5.3) contains a model of ReVolt obtained through different tests and measurements performed on ReVolt. Most measurements were completed by H. L. Alfheim and K. Muggerud during their master thesis [10], in addition to a towing tank test performed by the employees at DNV GL. This model is a 6DOF model, but was reduced to 3DOF before implemented into SBMPC, mostly done due to availability of data, but also for simplicity and an assessment of the cut variables (roll, pitch and heave) rating them less important for a good prediction. This resulted in the model from equation 2.1 with following values:

$$\boldsymbol{M} = \boldsymbol{M_A} + \boldsymbol{M_{RB}} = \begin{bmatrix} 7.432 & 0 & 0 \\ 0 & 55.84 & 4.05 \\ 0 & 3.728 & 22.45 \end{bmatrix} + \begin{bmatrix} 300 & 0 & 0 \\ 0 & 300 & -9 \\ 0 & -9 & 164 \end{bmatrix} \tag{3.4}$$

$$\begin{aligned} \boldsymbol{C}(\boldsymbol{\nu}) &= \boldsymbol{C_A}(\boldsymbol{\nu}) + \boldsymbol{C_{RB}}(\boldsymbol{\nu}) \\ &= \begin{bmatrix} 0 & 0 & 55.84v + 4.05r \\ 0 & 0 & -7.432u \\ -55.84v - 4.05r & 7.432u & 0 \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 & 9r - 300v \\ 0 & 0 & 300u \\ -9r + 300v & -300u & 0 \end{bmatrix} \end{aligned} \tag{3.5}$$

$$\boldsymbol{D} = \begin{bmatrix} 0.03074 & 0 & 0 \\ 0 & 0.1423 & 0 \\ 0 & 0 & 0.2193 \end{bmatrix}. \tag{3.6}$$

$\boldsymbol{M}$ and $\boldsymbol{D}$ are constant matrices where as $\boldsymbol{C}(\boldsymbol{\nu})$ changes depending on u, v and r.

With this model being a much better representation of ReVolt's slow dynamic, the improvement in performance of the algorithm will be evident during testing. Now the advantages of MPC could actually be exploited and the SBMPC algorithm will not be overconfident based on a bad prediction, believing ReVolt would be able to turn much faster than actually possible. A factor that might slightly reduce the newly improved performance is the skeg added onto ReVolt in 2018 (ReVolt is further described in chapter 5). It was added after the towing tank tests where performed, and is therefore not included in the model. The skeg further reduce ReVolt's already slow ability to maneuver. With the improved model, the next step is to use Euler prediction to predict the future path of ReVolt. It will include the vessel dynamic, but also the autopilot and guidance. An important assumption done by the prediction is that the guidance is constant. This assumption does not fit with the behaviour of a LOS guidance law and will affect how well the COLAV algorithm works. Predicted path will be more or less a straight line instead of the curve describing actually behaviour, this is kept in mind while examining the results later on. The equations for Euler prediction are as follows:

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos(\hat{\psi}_i) & -\sin(\hat{\psi}_i) \\ \sin(\hat{\psi}_i) & \cos(\hat{\psi}_i) \end{bmatrix} \tag{3.7a}$$

$$\hat{x}_{i+1} = \hat{x}_i + \Delta T \cdot (r_{11} \cdot \hat{u}_i + r_{12} \cdot \hat{v}_i) \tag{3.7b}$$

$$\hat{y}_{i+1} = \hat{y}_i + \Delta T \cdot (r_{21} \cdot \hat{u}_i + r_{22} \cdot \hat{v}_i) \tag{3.7c}$$

$$\hat{\psi}_{i+1} = \hat{\psi}_i + \Delta T \cdot \hat{r}_i \tag{3.7d}$$

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{C}(0) + \boldsymbol{D}(0) + K_{p,u} \cdot \boldsymbol{M}(u_d - \hat{u}_i) \\ (K_{p,\psi} \cdot I_z) \cdot ((\psi_d - \hat{\psi}_i) - K_{d,\psi} \cdot \hat{r}_i) \cdot \frac{1}{\text{rudder}_d} \\ (K_{p,\psi} \cdot I_z) \cdot ((\psi_d - \hat{\psi}_i) - K_{d,\psi} \cdot \hat{r}_i) \end{bmatrix} \tag{3.7e}$$

$$\boldsymbol{\mu} = \boldsymbol{M}^{-1} \cdot (\boldsymbol{\tau} - \boldsymbol{C} - \boldsymbol{D}) \tag{3.7f}$$

$$\hat{u}_{i+1} = \hat{u}_i + \Delta T \cdot \boldsymbol{\mu}(0) \tag{3.7g}$$

$$\hat{v}_{i+1} = \hat{v}_i + \Delta T \cdot \boldsymbol{\mu}(1) \tag{3.7h}$$

$$\hat{r}_{i+1} = \hat{r}_i + \Delta T \cdot \boldsymbol{\mu}(2), \tag{3.7i}$$

where $\hat{\psi}$ is the predicted heading of the vessel, ψ_d is the desired heading, $\hat{u}$ is predicted speed and u_d the desired speed the vessel is trying to reach. ΔT is the step size. rudder_d is the distance from the rudder to CG. $\boldsymbol{\mu}$ is a temporary variable only used during calculations. The first three equations are the same as for linear prediction, representing the kinematic part of the model. Then the heading angle is updated using the turning rate. $\boldsymbol{\tau}$ is a vector representing the control input and is updated at every time step, and is then used to update the future speed and rate of turn (ROT) estimates. This incorporates the kinetic part of the model. $\boldsymbol{\tau}$ contains two controllers used during predictions, a feedback-linearizing controller for speed and a conventional PD-controller for heading, also used to control yaw rate. The controllers are explained in section 2.2, and the the tuning parameters have the following values $K_{p,u} = 1$, $K_{p,\psi} = 1$ and $K_{d,\psi} = 5$. The improvement achieved with the new model is portrayed in figure 3.2.

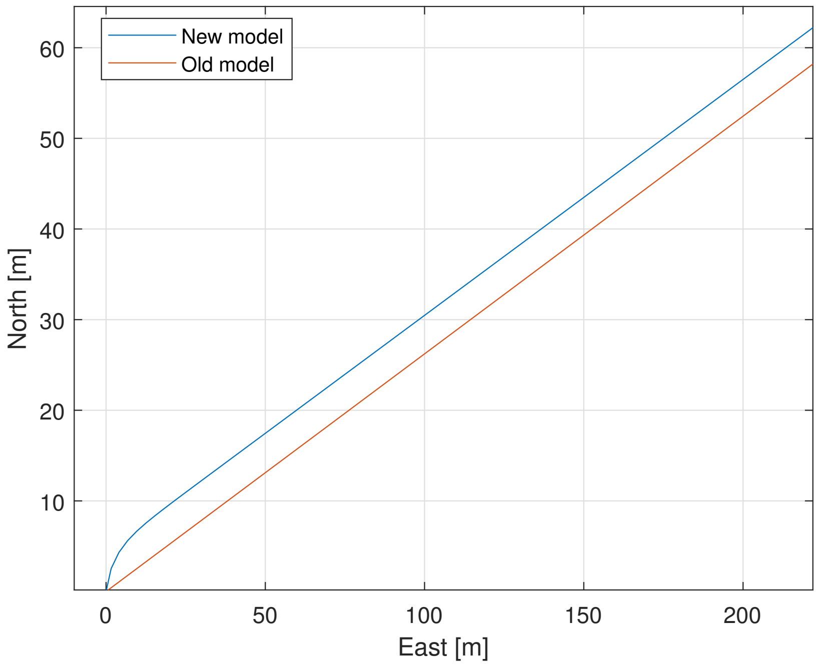


Figure 3.2: Improvement with new complete kinematic and kinetic model and Euler prediction, apposed to only kinematic model and linear prediction.

3.2.2 Selection of control law behaviour

To decide control behaviour a lot of different scenarios are evaluated. A scenario consists 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. [35] states a minimum of sets that should be evaluated, which is the base for the choices used in this thesis:

- Course offsets at $\{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$ in degrees.
- Speed factors are $\{\text{Keep speed, Slow forward, Stop}\}$, 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. Additionally will the model describing ReVolt not fit with the backwards motion as the vessel is controlled with propellers and a rudder. With all combinations of these control offsets 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. For this thesis the proposed minimum of course offsets are used and proven to be sufficient in several cases [28][29].

3.3 Cost function

The basis of the SBMPC cost function is adopted from [35]. Definitions of the different components in the cost function is from the author project thesis [40], but is for convenience repeated here. Additionally will newly introduced enhancements be elaborated in this section. The most important new addition is the COLREGS-transitional cost based on [29]. This combines to the following four main components used to evaluate the hazard of collision with other ships:

1. The cost of violating COLREGS
2. The cost of colliding with an obstacle
3. The COLREGS-transitional cost
4. The cost of changing your control offset

The most evident issue in the implementation of SBMPC in [28] and [40] is the oscillations that occur when an obstacle enter the evaluation area, but is on the outer perimeter of the COLREGS area, stepping in an out of it. Tuning the weights of the components in the cost function only partially attenuated the problem, prompting the need for a better solution. The oscillatory behaviour will be further explained in the COLREGS-transitional cost section, in combination with a solution to the problem.

The SBMPC objective is to evaluate all possible scenarios (k) at time t_0 and select the control behaviour that minimizes the cost function ($\mathcal{H}^k(t_0)$) as following:

$$k^*(t_0) = \arg\min_k \mathcal{H}^k(t_0), \tag{3.8}$$

where

$$\mathcal{H}^k(t_0) = \max_i \max_{t \in D(t_0)} (\mathcal{C}_i^k(t)\mathcal{R}_i^k(t) + \kappa_i \mathcal{M}_i^k(t) + \lambda_i \mathcal{T}_i^k(t)) + f(\mathcal{P}^k, \chi_{ca}^k) \tag{3.9}$$

where t_0 is the current time, t is a future time, κ and λ are tuning parameters and i represents the obstacles. Each scenario is evaluated at discrete sample times along the horizon T using the discretization interval ΔT, which defines $D(t_0)$ as $\{t_0, t_0 + \Delta T, \ldots, t_0 + T\}$. The remaining parameters will be described in the following sections.

3.3.1 COLREGS cost

The second element of the cost function consists of κ_i and $\mathcal{M}$, where as mentioned above κ is a tuning or weighting parameter and $\mathcal{M}$ is a boolean variable representing whether or not there is a COLREGS associated cost. This section describes how $\mathcal{M}$ is calculated.

As for the road, the sea has its own set of rules, called COLREGS. When performing COLAV maneuvers it is important for the autonomous surface vessel (ASV) to follow these rules, making the actions taken logical and predicable for operators of other vessels. The CAS uses the available information illustrated in figure 3.3 and specified in table 3.1 to evaluate the situational hazard with respect to COLREGS. Furthermore, the COLREGS compliant control action with the lowest risk and smallest deviation from the desired path is chosen.

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 from own-ship to the obstacle with index i in scenario k
$d_{o,i}^k(t)$	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 3.1: Parameter description for hazard calculations with respect to COLREGS [35].

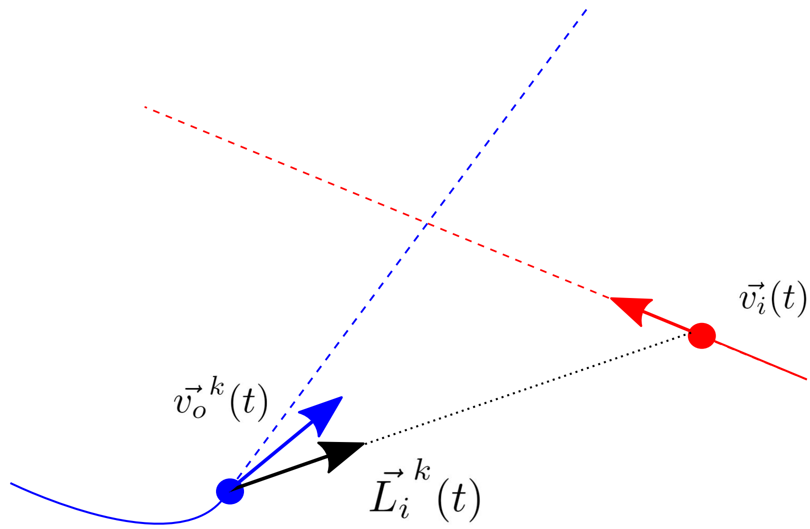


Figure 3.3: Main information for hazard evaluation in scenario k at time t [35].

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 area in which COLREGS apply, denoted d^{cl}_i. Utilizing these five parameters we can evaluate the risk at a time t in a scenario k. Which COLREGS rule that apply in a given situation is defined by different boolean parameters. These parameters are only valid when the obstacle is within COLREGS perimeters, meaning the distance to ReVolt is less than d^{cl}_i. All expressions below are obtained from [35] and the parameters used are described in table 3.1:

- **CLOSE:** When $d^k_{o,i}(t) \leq$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 ReVolt's 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 ϕ_{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 $\mathcal{M}_i^k(t) \in 0, 1$ is used. The rules taken into account here is mainly rules 14 (head-on) and 15 (crossing), resulting in the following expressions:

$$\mathcal{M}_i^k(t) = \textit{RULE 14 or RULE 15}$$

$$\textit{RULE 14} = \textit{CLOSE \& STARBOARD \& HEAD-ON}$$

$$\textit{RULE 15} = \textit{CLOSE \& STARBOARD \& CROSSING \& NOT OVERTAKEN}$$

Additionally is rule 13 (overtaking) implicitly taken into account through rule 14. This expression only consider rules 13, 14 and 15, whilst the remaining rules described in section 2.5 are relevant as well. These will be attempted followed by tuning the remaining parameters in the cost function.

3.3.2 Collision cost

The first element in the equation is composed of $\mathcal{C}$ and $\mathcal{R}$ representing the collision hazard. $\mathcal{R}$ is the collision risk factor given as:

$$\mathcal{R}_i^k(t) = \begin{cases} \frac{1}{|t - t_0|^p} \left(\frac{d_i^{safe}}{d_{o,i}^k(t)}\right)^q, & \textit{if } d_{o,i}^k(t) \leq d_i^{safe} \\ 0, & otherwise \end{cases} \tag{3.10}$$

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}, and d_i^{safe} is a chosen minimum distance between the ASV and the obstacles that should be adhered to. 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, 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 on close in time hazards.

Further will the short-term prediction be more accurate than the long-term, based on utilization of linear prediction of the obstacles future path. 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:

$$\mathcal{C}_i^k(t) = K_i^{coll}(t)|\vec{v_o}^k(t) - \vec{v_i}(t)|^2. \tag{3.11}$$

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 on several different conditions, like obstacle size, the right to stay on and/or responsibility to keep out of the way.

3.3.3 COLREGS transitional cost

The third part of the cost function is the COLREGS transitional cost represented by $\mathcal{T}$. This will penalize control behaviour that lead to aborting of a COLREGS maneuver, which among other thing addresses the problem of oscillations experienced in previous iterations of the algorithm [40][28]. When utilizing a line-of-sight (LOS) guidance scheme, there is a LOS vector pointing in the ASVs desired direction. When the ASV is on the desired path, this will point along the path. As obstacles arise, the ASV will avoid collision by altering its course and hence deviate from the desired path. In the next round of predictions will the LOS vector point back towards the desired path, and if it represents a collision free and COLREGS compliant path, the corresponding course of action is chosen. The path is collision free because the predictions use a constant value from the guidance law, not representative of where the vessel is headed in the long run, hence the returning predicted path will never approach the obstacle again. When returning to desired path the optimal action would again be to alter the course to avoid collision. This will repeat it self, leading to oscillation, until the vessels are close enough for the risk of collision to dominate the COLREGS cost. These oscillations are a significant problem that the other parts of the cost function are not sophisticated enough to handle. Hence the introduction of COLREGS transitional cost was a must, and together with some other improvements it solves the oscillation problem. Figure 3.4 shows the problematic behaviour.

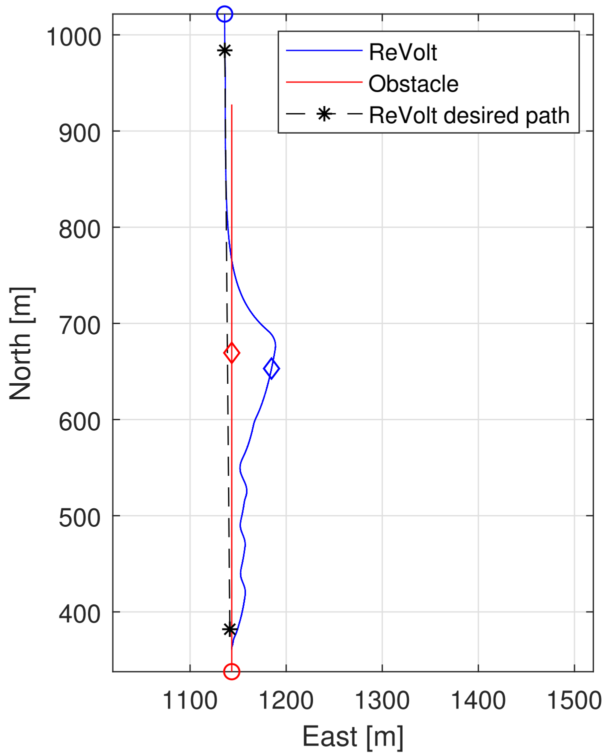


Figure 3.4: Plot of the positions and desired paths of ReVolt and obstacle, displaying unwanted oscillations. Diamond markers represent closest point of approach at 40 meters.

Similarly to the COLREGS compliance cost the COLREGS transitional cost i formulated using a binary indicator $\mathcal{T}_i^k \in \{0, 1\}$ which is weighted by the tuning parameter λ_i. The indicator is given by:

$$\mathcal{T}_i^k(t) = O_i^k(t) \vee Q_i^k(t) \vee X_i^k(t), \tag{3.12}$$

where $O_i^k(t)$, $Q_i^k(t)$ and $X_i^k(t)$ are binary indicators as well representing the ASV is overtaking a vessel, the ASV is being overtaken and the ASV is in a crossing situation, respectively. If $\mathcal{T}_i^k = 1$ it means that scenarios k will, at a future time, lead to the vessels passing each other on the opposite side of what is currently predicted. Hence action must be taken to avoid oscillation and indecisive behaviour. The rest of the section will define and explain the three different situation indicator.

Overtaking

There will be a transitional cost associated with a given control behaviour when the ASV is overtaking an obstacle, if the predicted position of the obstacle at a future time t is not on the same side of the ASV as at the current time. Writing this out as a mathematical formula results in the following:

$$O_i^k(t) = \begin{cases} O_i(t_0) \wedge S_i^k(t) & \text{if } \neg S_i^k(t_0) \\ O_i(t_0) \wedge \neg S_i^k(t) & \text{if } S_i^k(t_0) \end{cases} \tag{3.13}$$

where $S_i^k(t)$ and $S_i^k(t_0)$ represent whether or not the obstacle is on the starboard side of the ASV at future time t or the current time t_0 respectively. Hence if either parameter equals 1 the obstacles will at that time appear on starboard side of the ASV. $O_i(t_0) = 1$ means that the ASV is currently overtaking obstacle i, which necessary criteria are explained above in the COLREGS cost section (3.3.1).

Being overtaken

If the ASV is being overtaken by an obstacle, will there be a transitional cost associated with a maneuver if it causes the predicted position of the ASV to be on the opposite side of what it is currently. This case is evaluated similarly to the overtaking case above, as the only difference is whether or not the ASV is the give way vessel.

$$Q_i^k(t) = \begin{cases} Q_i(t_0) \wedge S_i^k(t) & \text{if } \neg S_i^k(t_0) \\ Q_i(t_0) \wedge \neg S_i^k(t) & \text{if } S_i^k(t_0) \end{cases} \tag{3.14}$$

where $Q_i(t_0) = 1$ implies that the ASV is currently being overtaken by another vessel.

Crossing

There are two different crossing scenarios, either the obstacle comes from port side or starboard side crossing the ASV path. According to COLREGS is the correct behaviour for the give way vessel a starboard maneuver leaving the obstacle on port side after the crossing is finished. Hence alternate control behaviour will have an associated transitional cost if they lead to the obstacle appearing on the starboard side of the ASV.

$$X_i^k(t) = X_i^k \wedge S_i^k(t_0) \wedge S_i^k(t) \wedge \text{turn to port} \tag{3.15}$$

where $X_i(t_0)$ represent whether or not the ASV is in a crossing situation at the current time, which necessary criteria are explained above in the COLREGS cost section (3.3.1) as well.

3.3.4 Maneuvering cost

The last element of the cost function is $f(\mathcal{P}^k, \chi_{ca}^k)$ which is given as:

$$f(\mathcal{P}, \chi_{ca}) = k_P(1 - P) + k_{\chi_s}\chi_s + \Delta_P(P - P_{last}) + \Delta_\chi(\chi_{ca} - \chi_{ca,last}) \tag{3.16}$$

where Δ_P, χ_s and Δ_χ are penalty functions and k_p and k_{χ_s} are positive tuning parameters that influence the priority of keeping nominal speed and course. This part of the hazard function is making sure there is not an unnecessary high offset from the nominal course and speed. It also assures that the ship gets back to the original path when the collision hazard is over. χ_s and Δ_χ 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.

Δ_χ 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:

$$\Delta_\chi = \begin{cases} K_{\Delta_\chi,port}(\chi_{ca} - \chi_{ca,last})^2, & \textit{if turn to port} \\ K_{\Delta_\chi,starboard}(\chi_{ca} - \chi_{ca,last})^2, & \textit{if turn to starboard} \\ 0, & \textit{otherwise} \end{cases} \tag{3.17}$$

where $K_{\Delta_\chi,port}$ and $K_{\Delta_\chi,starboard}$ are the tuning parameters, given different values forcing the favouring of turning to starboard side, as this is in compliance with COLREGS.

χ_s is similar to Δ_χ, but depends on the size of the offset instead of change in offset. The asymmetry works equally to Δ_χ, with weighting parameters $K_{\chi_s,port}$ and $K_{\chi_s,starboard}$.

$$\chi_s = \begin{cases} K_{\chi_s,port}(\chi_{ca})^2, & \textit{if turn to port} \\ K_{\chi_s,starboard}(\chi_{ca})^2, & \textit{if turn to starboard} \end{cases} \tag{3.18}$$

3.4 Other improvements

Most improvements to the algorithm are made with regards to the unwanted oscillations that have affected the algorithm in previous projects. To substantiate the implementation of transitional cost, some smaller improvements and necessary parameters were designed. This section describe implementations found in the SBMPC library that are improvements from the authors project thesis [40].

When the ASV is approaching an obstacle, cost calculations are not started until the distance is less then the parameter D_{CLOSE}. But when the obstacle has just entered the COLAV range there is a risk of it leaving it again the next time step, caused by minor changes in course or noise on the measurements. This will lead to oscillatory behaviour, hence the solution to this is a parameter called D_{INIT}. If the distance to one of the obstacles is less than the constant value D_{INIT}, then it is close to the COLAV area and $colav\_active$ variable is set to true. It is only set to false again when the distance is bigger than $D_{INIT} + D_{SAFE}$. This will avoid the problem of turning on and off the COLAV calculations when the obstacle is just entering dangerous area. If the value of $colav\_active = false$ then zero offsets are used, else the code runs as usual.

An additional improvement to avoid strange, unwanted and oscillatory behaviour are current scenario descriptors. These descriptors define what type of scenario the ASV is in at the current time and are used both in the transitional cost and independently. Typical scenario types are crossing, overtaking and head-on. The descriptor are calculated outside the cost function and passed in as an argument. The purpose of the parameters is to avoid the ASV drastically changing the desired course of action mid way through a maneuver, because it looks like it suddenly enter another situation. An example of this is after the crossing when the ASV is returning back to the path, it will then have close to same direction as the obstacle and it will reassemble an overtaking. Hence the algorithm could be lead to believe it needs to alter its course.

Together with the scenario descriptors is the parameter OBS_{PASSED} passed to the cost function as an argument, helping to avoid undesirable behaviour after the obstacle has passed. The hazard would be over after the obstacle have passed, even though the distance to it is still quite small and less than D_{SAFE}. The parameter is only used in crossing situations, helping the ASV to return back to nominal path smoothly.

A small improvement not connected to oscillations is the introduction of a dynamic value for D_{SAFE}. D_{SAFE} represents a desired minimum distance between the ASV and the obstacle. The distance between the ships are calculated from the center of the ships not correcting for the size of the ship. Hence if the ASV approach a large, 300 meters long shipping vessel, a minimum distance of 100 meters would mean you might crash into the back of the ship, but the safe distance of 100 meters is more than enough if you meet a small daycruiser. Hence the safe distance depends on the obstacle size like this: $d^i_{safe} = D_{SAFE} + \frac{OBS^k_{len}}{2}$. Where d^i_{safe} is the safe distance calculated for each time step i and OBS^k_{len} is the length of obstacle k.

3.5 Implementation

The implementation of SBMPC is based on the SBMPC library provided by Autosea, a knowledge-building project founded by the Research Council of Norway and owned by the department of Engineering Cybernetics at NTNU [1]. It is a object oriented library, based around the main class called ScenarioBasedMpc. This class handles all the parameters, the cost function and the overall choice of control offset. To integrate the collision avoidance system into ReVolt's existing code base only one function is to be called, named getBestControlOffset. It is called from the guidance controller every 5$^{\text{th}}$ second. The input to this function consist of desired speed and course from the guidance law, ASV state and a list of obstacles. The output is the desired heading and speed with potential offsets calculated in the function. An overview of structure and data flow is presented in figure 3.5.

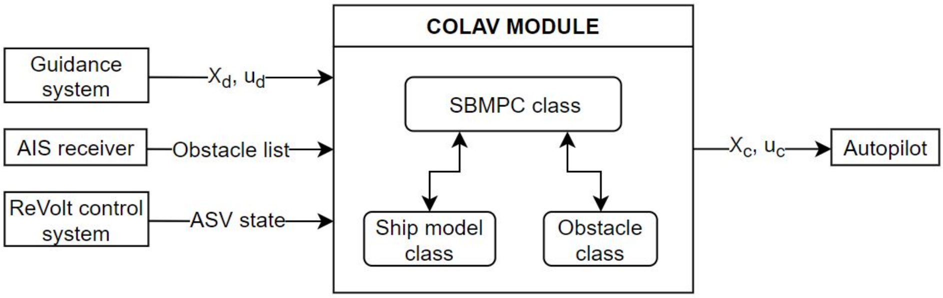


Figure 3.5: Data flow and structure of the collision avoidance module. χ_d and u_d are desired values from guidance and χ_c and u_c are modified values from the COLAV module.

In addition to the main class there is a ship model class and an obstacle class. The obstacle class itself was left untouched, just designing generic obstacle vessels and predicting their future path using linear prediction. Instead the input promotes the biggest change as it takes ship size and type information into account when designing the obstacle class, instead of using a default. This information was received over the automatic identification system (AIS), which is an addition to the overall system designed during this thesis. It is described under the simulation section (5.3.1) later on. The ship model on the other hand is fully changed out for a model actually fitting with the mathematical model of ReVolt. It is more or less based on the same model structure, but changing out all the parameters and removing most simplification previously done to the model.

The most challenging part of the implementations is getting the tuning right. There are a lot of parameters influencing the performance of the SBMPC algorithm, leaving many possible combinations to assess. As ReVolt is a slow and small vessel, the parameters are scaled thereafter, hence using a shorter distance on both D_{CLOSE} and D_{SAFE} compared to faster or lager vessels. All the tuning parameters and the values chosen in the end are listed below:

Parameter	**Value**	**Description**
T	600.0 [s]	Prediction horizon
DT	0.5 [s]	Time step used for trajectory prediction
pred_step	10	Reduces the number of prediction steps in obstacles
P	1	Weight on time to evaluation situation
Q	4	Weight on distance to obstacle at evaluation time
D_{INIT}	300.0 [m]	Distance where COLAV is activated
D_{CLOSE}	300.0 [m]	Distance where COLREGS comply
D_{SAFE}	100.0 [m]	Distance to obstacles that is considered safe
K_{COLL}	0.1	Weight on collision cost
ϕ_{AH}	68.5 [deg]	Angle specifying if obstacle is ahead
ϕ_{OT}	68.5 [deg]	Angle specifying if an obstacle overtaking the ship
ϕ_{HO}	22.5 [deg]	Angle specifying if an obstacle head on the ship
ϕ_{CR}	68.5 [deg]	Angle specifying if an obstacle crossing the ship
κ	3.0	The cost of not complying with COLREGS
κ_{TC}	10.0	Transitional cost
K_P	100	Cost of having a speed offset from nominal speed
$K_{\chi_s,starboard}$	1.5	Cost of course offset from nominal course to starboard side
$K_{\chi_s,port}$	100	Cost of course offset from nominal course to port side
K_{Δ_P}	0.5	Cost of changing speed
$K_{\Delta\chi,starboard}$	0.5	Cost of changing course to starboard side
$K_{\Delta\chi,port}$	0.9	Cost of changing course to port side

Table 3.2: Final parameters in the COLAV algorithm, used during testing.

Chapter 4

Map Extraction and Anti-Grounding

During short sea shipping, especially in Norway, there is a long list of possible hazards represented by narrow fjords, land, shallow waters, islands, underwater skerries and more. Hence an anti-grounding system will be a necessary addition to the full collision avoidance system (CAS). A natural way of detecting hazards is with light detection and ranging (LIDAR) equipment and cameras, but they are not optimal do measure water depth, detect underwater skerries and they are also weather dependent. Hence for safer travels along the coast, avoiding grounding based on map data is an essential addition to the CAS. Processing of the map data and the actual anti-grounding system implemented onto ReVolt will be explained in this chapter.

4.1 Electronic Navigational Charts

The implementation used in this thesis is based on Electronic Navigational Charts (ENC) which are electronic vector maps of the sea, containing all information necessary for safe sailing [6]. These charts are obtained according to the International Hydrographic Organisation's (IHO) standards called S57. As the charts include an extensive amount of information it will be necessary to extract the relevant information according to what kind of mission the own-ship is on. Hence sailing in open water is clearly different from sailing into a fjord and docking in a harbour. This leads to the necessity of choosing relevant information for a given mission. ReVolt will be sailing in Dorabassenget, Nidelven and close to Munkholmen for testing purposes in this thesis and will therefore only need data about the coastline, water depth, bouys and beacons. For actual short sea shipping additional information might be necessary. In ENC all features have a letter code, those who are relevant for this thesis are listed below (a full list of available features could be found here [33]):

Acronym	Object name	Description
BCNISD	Isolated Danger Beacon	A beacon placed on an isolated danger with water all around.
BCNLAT	Lateral Beacon	A beacon that indicates the port or starboard hand side of the route to be followed.
BCNSPP	Special Purpose Beacon	A beacon used to indicate an area or feature which is in reference to a chart, a Notice to Mariners or Sailing Directions.
BOYSAW	Safe Water Buoy	Indicate navigable water.
DEPARE	Depth Area	An area where the water depth is within a defined range of values.

Table 4.1: Selection of relevant S57 objects for this thesis [33].

4.2 Map Extraction

To extract the relevant information from the ENC essential to the CAS, the algorithm designed by O. S. Otterholm, in his masters thesis [44], was applied. The code is published and available on GitHub (https://github.com/olesot/map-extraction). The algorithm offers the opportunity to enter the wanted S57 features and a minimum water depth for your vessel to sail safely. The extracted information is returned in a new file of type ESRI Shapefile, including only the selected features, stacked in separate layers. This file represents the hazardous areas for the vessel to traverse. Next step is merging all features to only one final layer, containing of several polygons. This is a necessary step, formatting the map data for use in the anti-grounding system. The resulting format is chosen to be compatible with the libraries from GDAL/OGR, which are open source libraries designed to read and write vector geospatial data formats.

The extraction process happens offline, preparing the map data for the relevant area and scenarios. As ReVolt is being tested close to the shore the coastline is of course an important feature to extract. As the extraction algorithm depends on depth, it will mark all parts of land hazards areas. This is not optimal with regards to run-time, as there is more data than strictly necessary. Hence only a outline of the hazards land areas are included in the final file, reducing the amount of data need to be processed to a minimum. Figure 4.1 shows the original ENC data, along with the resulting reduced Shapefile in figure 4.2. Original ENC data was provided by Kartverket via email. The result from the offline code is passed on to the CAS to be applied in the anti-grounding system. During reduction the code was not able to keep the coastline consistent, creating gaps in the line separating the hazardous area from the safe. This is problematic as from ReVolt's standpoint those gaps look like passageways safe to sail through, which in reality causes a collision with land. The reason for these gaps is unknown, and due to limited time there was not made an effort to figure out what is causing the problem. A solution to still carry out realistic testing of the anti-grounding system where to conduct test around Munkholmen which does not demonstrate the problem.

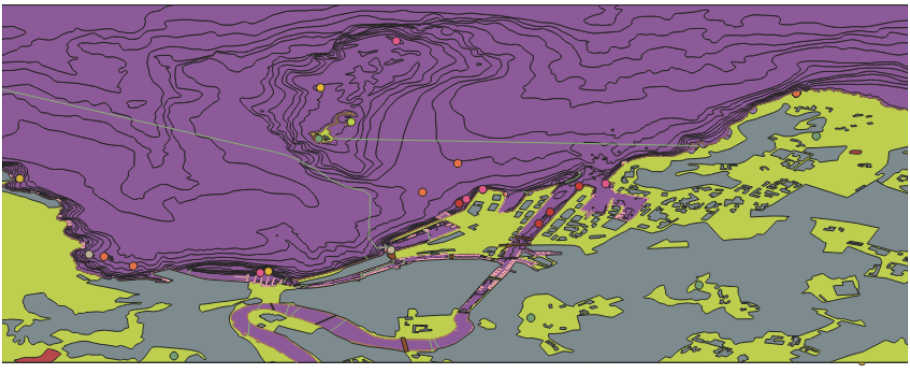


Figure 4.1: The original ENC with all features showing.

Figure 4.2: Reduced chart containing only the desired features. Purple outline the no-go zones.

4.3 Anti-Grounding System

The basic idea behind this anti-grounding system is to check whether ReVolt's predicted path is colliding with a hazardous area in the map. This is done by checking for intersections between the curve, representing ReVolt's path, and the polygons, presenting the no-go zones in the map. If the intersection function return true the future path of ReVolt is predicted to collide with land and/or the seabed and promotes the need to take action. To implement this functionality the GDAL/OGR library is used. Which further provides both a C++ and a Python application programming interface (API).

The anti-grounding code implemented for this master is inspired by the C++ code provided by Otterholm [44]. The first attempt was to implement an already existing solution into ReVolt's code base, for a quick way of improving the collision avoidance system. But due to substantial amounts of installation troubles of the GDAL/OGR library and mysteries unexplained behaviour from the built in functions, this idea was rejected. Instead a code, inspired by the existing C++ algorithm, was implemented using the Python API. As Python allowed for simpler ways of debugging, a working anti-grounding system was achieved with less amounts of problems along the way, designed as a standalone robot operating system (ROS) node. It was connected to the rest of the SBMPC algorithm using ROS messages.

The necessary input to the anti-grounding function are the 39 curves representing the predicted paths for ReVolt in all different combinations of control offsets. These values were published by the SBMPC code after the predictions were finished. The Python node subscribes to this data, using it to check for intersection and calculate risk of grounding costs for all scenarios. These costs are then published using a separate publisher, for the guidance law to subscribe to and pass on to the SBMPC function as an input argument. It is further added to the existing cost after the other cost calculations are completed. This is not optimal for simplicity and understanding of the code, neither with regards to run time. Thus with more time a C++ version of the function should be implemented and called internally in the SBMPC algorithm. This provides the opportunity to call the function for each scenario, not testing all scenarios at once. For the purpose of this thesis the implementation is sufficient, in that it runs real time during simulations and performs the desired task, but some simplification were needed to achieve this.

As there are 39 different scenarios to predict for ReVolt, and each is predicted with 1200 steps, where each step have an x- and y-coordinate, there were too much data to be published. This caused a slight problem with array size and run time, hence to not loose too much resolution in the line data, the solution was to only use the 13 cases where ReVolt is at full speed. This was justified as the SBMPC algorithm tries to keep full speed anyway and with ReVolt's low top speed of only $1.5\,\mathrm{m/s}$ there is not much difference between the scenarios. It would also be a reasonable simplification as with maximum speed it will hit such obstacles sooner than with a lower speed. To achieve acceptable performance 200 points were used to describe each predicted path.

A proposed task as future work in Otterholm's thesis, is considering the distance from the ASV to the hazards, using it to avoid motions close to hazardous areas. To achieve this the predicted line for ReVolt was split into three parts, each checked for intersection separately. Part one is the first 120 steps, which is 60 seconds and with nominal speed of $1\,\mathrm{m/s}$, it corresponds to 60 meters. Part two is the next 420 steps or 3.5 minutes, and part three is the remaining 600 steps equivalent of 5 minutes. If the first part, closest to ReVolt intersects with the hazardmap the cost returned will be higher than if it intersects in any of the other parts further away from the vessel. This allows for a smoother algorithm letting the vessel go within some distance of the hazard area, just not too close. What is considered ”too close” is a tuning question and will depend on size of the boat, how narrow the passage sailed in is and other relevant factors. It can be affected by both the split of the line as well as the cost of intersection. The costs used in this implementation are for part one 100, 40 for part two and 10 for part three.

Chapter 5

Experimental Platform

The physical experimental platform utilized in this master thesis is the same as in the author's specialization project [40], hence the general description is based on that. In addition will the provided simulator be described in this chapter, as it forms the basis for the extensive simulation testing performed in this thesis. Including the new feature, an AIS simulator.

5.1 ReVolt concept

Figure 5.1: Concept ReVolt [47].

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.

The concept ship has an optimal speed of 6 knots, cargo capacity at 100 TEU and an 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. [45]

5.2 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 5.2: ReVolt test platform.

ReVolt as it is today has a maximum speed of about $1.5\,\mathrm{m/s}$ and the thruster angle is restricted at $\pm 45°$ offset to each side during transit. These restrictions influence ReVolt's 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 ReVolt's 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.

5.2.1 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.

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	MIC-7700Q
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	-
LIDAR	On top front	Velodyne VLP-16
Camera	On top bow	Flir Ladybug5+

Table 5.1: Components on the ReVolt.

Summer 2018 two new batteries where installed which increased possible testing time. And in march 2019 a new more powerful embedded computer was fitted in ReVolt, increasing computational power, leaving the system able to do the real time calculations necessary for an autonomous surface vessel to function optimally. An AIS receiver was purchased in the beginning of 2019, but there has not been a priority to install it, hence it will not be included on ReVolt during real life testing.

5.3 Simulator

Simulations are a very important part of the development process, as testing directly on the real life ReVolt would never be sustainable. When the code is not tested properly in simulations before a sea trial the risk of unwanted, hazards or unpredictable behaviour is high, which can result in damage on the test platform or other surrounding objects. It is also a waste of time, when minor bug fixes could have been done in advance. The simulator framework called CyberSea was therefore provided by DNV GL for this master thesis. The simulator includes a digital twin of ReVolt, making simulations more realistic and close to real life testing. CyberSea provides a setup which gives a distinct separation between the different modules, as the control system is run on another computer hence lying entirely outside the simulator. Such a setup has the advantage that the control systems is not able to tell the difference between a marine vessel and a simulated one, leaving the gap between simulations and real life testing as short as possible. Testing in CyberSea is done using the exact same code as would be used for running the actual ReVolt.

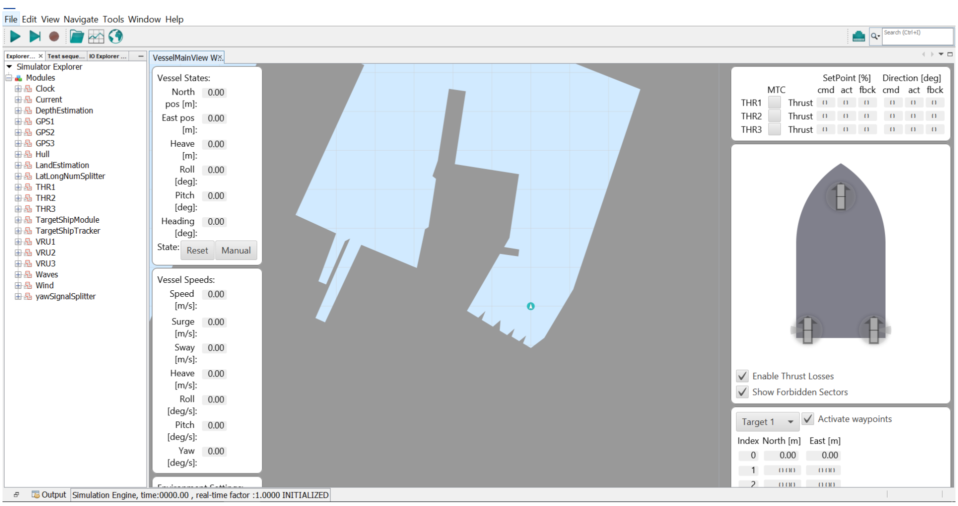


Figure 5.3: CyberSea simulator.

The simulator environment is shown in Figure 5.3 above. It includes an outline of a map, display of direction and throttle settings on all actuators, displays for different interesting values, list of all internal parameters as well as the parameters passed on to the control system. Out of frame in figure 5.3 is the environmental settings shown in figure 5.4. This panel allows to change the speed and direction of wind, the wave height and direction of them as well as the speed and direction of the ocean current. The pull down menu on top contains a list of predefined weather combinations that could be used instead of setting each factor individually. This is useful for more realistic testing, as there is never completely still waters around Nidelva and in Trondheimsfjorden.

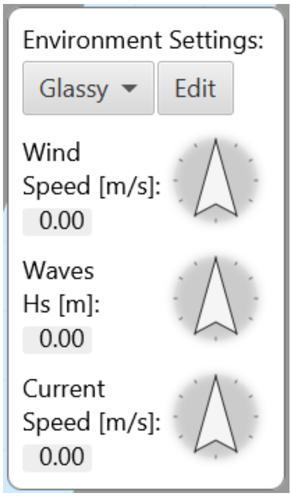


Figure 5.4: CyberSea environmental settings.

In addition to all the parameters controlling different parts of ReVolt, CyberSea have the possibility to add in up to 32 virtual obstacle ships. These ships could sail in straight lines, change course angle and speed at set times or follow waypoints. They can also run collision avoidance them selves, leading to more sophisticated and realistic testing. This is a new addition to the simulator for this master thesis, as well as the simulation of AIS data from the given obstacles. Behaviour of all 32 obstacles throughout a scenario can be designed in advance using a test sequence (TSQ) file. This increases simplicity of simulations as well as providing the opportunity to either test the same exact case several times, or make minor changes as you go without having to reenter all the obstacle information at run-time. All data from the simulated sensor are sent to the control system using modbus. Modbus is a serial communication protocol and enables communication among devices connected to the same network. Hence the test setup would require a network connection between the simulator computer and the control system computer. The exception from this is the AIS data, which is transmitted over User Datagram Protocol (UDP) for a more close to real life setup.

5.3.1 Automatic Identification System

As mentioned above the obstacle information is now simulated in the Automatic Identification System (AIS) message format. The data includes information from other ships about the type of ship as well as its position, speed, course and more. To access this information on physical ReVolt an AIS receiver is needed, which makes it possible to identify other ships in a near by proximity, given they have a AIS transmitter onboard. For all commercial vessels and bigger ships, an AIS transmitter is mandatory. AIS uses the same frequency as a very high frequency (VHF) radio and is an addition to the traditional radio onboard a boat. It makes it possible to look around headlands to detect radar target out of reach [2].

To use the simulated AIS data, published via UDP, a socket was needed to receive data through. Further the data have to be distributed to the rest of the system. This was done by designing a new ROS node called UDP receiver which would read the received data, decode it and publish it on a ROS publisher. The actual receiver would be the only part differing between simulations and real life VHF transmitting. Another driver would be necessary to read the received data, but when that is done, it could be passed to the same ROS node and the remaining part of the code, explained below, will be exactly the same.

All AIS data is encoded using the international standard format of National Marine Electronics Association (NMEA). There are 24 different AIS message types defined in this standard. A typical AIS message would look like this:

$$!AIVDM, 1, 1, , A, 100000P00DPgfi : TCI < G25` : P000, 0 * 29$$

The first part is the NMEA message type, where $!AIVDM$ mean data received from another vessel. Another type could be $!AIVDO$ which is your own vessel information. The next three are the number of sentences, the sentence number and the sentence ID (in case of multi-sentence messages). Next is the AIS channel and then comes the actual encoded AIS data. The last field starts with a 0 which represents the number of fill bits required for the data to fit the 6 bit boundary, and ends with *29 which is the NMEA 0183 data-integrity checksum [5].

The AIS message data is encoded using a 6-bits American Standard Code for Information Interchange (ASCII) mechanism, when received the message char is decoded and converted into binary, leaving a long string of zeros and ones. Further the binary message string is split according to the decoding specifications corresponding to the specific message type. Depending of which messages type is being sent the table presented under will differ, but for the purpose of this thesis only the Common Navigational Block has been implemented, which supports message types 1, 2 and 3 containing navigational information.

Field	Description
0-5	Message Type
6-7	Repeat Indicator
8-37	NMSI
38-41	Navigation Status
42-49	Rate Of Turn
50-59	Speed Over Ground
60-60	Position Accuracy
61-88	Longitude
89-115	Latitude
116-127	Course Over Ground
128-136	True Heading
137-142	Time stamp
143-144	Maneuver Indicator
145-147	Spare
148-148	RAIM flag
149-167	Radio status

Table 5.2: The Common Navigation block. Showing which part of the decoded AIS message represent what data. Relevant for message type 1, 2 and 3: Position Report Class A [5].

After the string is split into the different fields presented in table 5.2 there are still some additional information and decoding needed, to use this data as intended. The navigational status number has to be crosschecked with a table giving statuses like under way using engine, at anchor, not under command and more. Further the correct value for course over ground (COG), speed over ground (SOG), rate of turn (ROT), latitude and longitude have to be calculated using the following formulas respectively:

$$COG = \frac{COG_{AIS}}{10.0} \tag{5.1}$$

$$SOG = \frac{SOG_{AIS}}{10} \cdot 0.514444 \tag{5.2}$$

$$ROT = \text{sgn}(ROT_{AIS})(\frac{ROT_{AIS}}{4.733})^2 \tag{5.3}$$

$$\text{Latitude} = \frac{\text{Latitude}_{AIS}}{600000.0} \tag{5.4}$$

$$\text{Longitude} = \frac{\text{Longitude}_{AIS}}{600000.0}. \tag{5.5}$$

Chapter 6

Simulations

This chapter presents the completed simulation study, used to confirm that the system works as expected and test how more complex, realistic scenarios impact performance.

All simulations are conducted in the simulator framework called CyberSea described in Section 5.3, in combination with ReVolt's remote control (RMC) station designed by A. Havnegjerde, further described in [30]. A test case contains a desired path for ReVolt to follow together with obstacles to interact with. In the RMC station a desired path can be designed by placing waypoints on to a map, transmitted to the control system via transmission control protocol (TCP). A snapshot of the path-designing window of the RMC station is portrayed in figure 6.1. The obstacles are designed in and simulated by CyberSea, and are detected by the control system using the simulated AIS data. There is access to a long list of parameters describing the obstacles, giving the opportunity to customise realistic test cases, fitting to your problem description.

In the simulations presented here only start position, speed and course will be changed. In addition ability to turn on and off the obstacle's own collision avoidance system is utilized in some selected cases. The obstacles are running velocity obstacle (VO) as their method of COLAV. It is a simpler algorithm, but it will help increase sophistication of the obstacles, leading to more realistic test cases. This VO algorithm is using current position and velocity to determine which velocities will result in collision. To chose a safe control offsets a grid of possible headings and speeds is defined, where a cost is assigned to each cell representing a velocity. The cost is affected by several tuning parameters, but the ones considered here are d_{CPA} and t_{CPA} which are the distance and time to closest point of approach (CPA) respectively. For a more thorough explanation of the method and its parameters see [42], which originally implemented it into CyberSea. The method has later been re-implemented and is currently not performing optimally in all scenarios, which is why it is only used selectively during this simulation study.

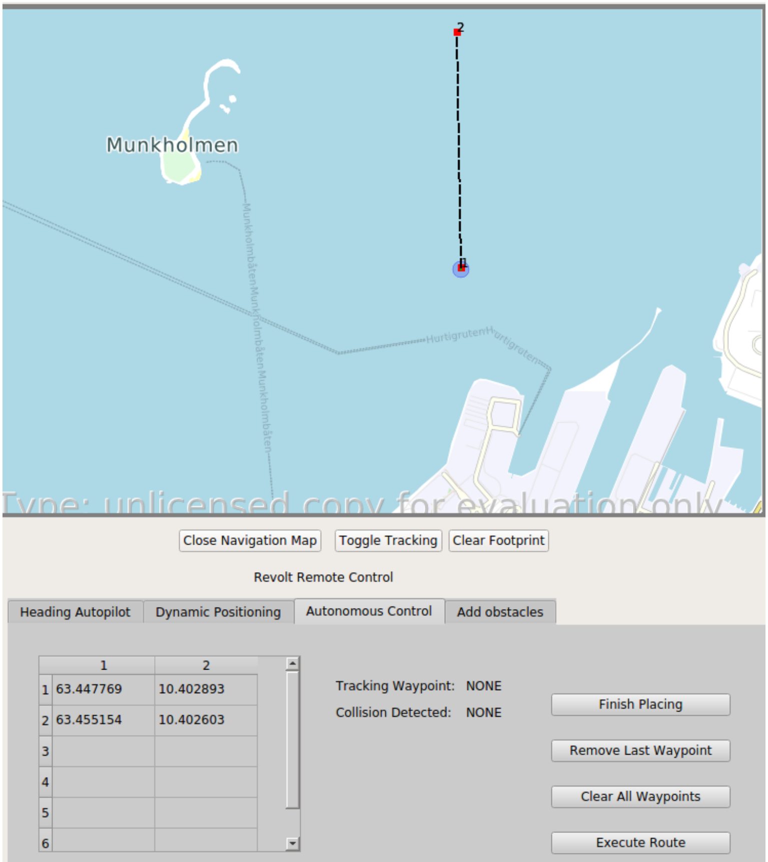


Figure 6.1: The remote control station. Showing how ReVolt's desired path is designed.

Extensive testing have been conducted during the simulation study, leaving the system in best possible condition to handle real life experiments. Striving to construct realistic test cases with regard to obstacle behaviour, number of obstacles, environmental disturbances and map restrictions. Results along the way have been used for tuning purposes, aiming for optimal behaviour of the SBMPC algorithm, in all different cases. The plots that will be displayed in this section are the final results after considerable amounts of tuning, showing how different scenarios are working with the "optimal" tuning combination. It should be pointed out that more emphasis was placed on tuning the simpler cases, as they were directly relevant for real life experiments. Tuning was performed systematically by examining how each separate part of the cost function was affected by each tuning parameter. This makes it easy to see which part of the cost function is dominating and possibly reduce the parameters which are responsible, if it is subject to unwanted behaviour. The final parameters are stated in section 3.5.

The collision avoidance system (CAS) is built upon the already existing heading controller, speed controller and guidance law. They all seem to be working adequately during simulations, given that there are no disturbances on the measurements. While testing with environmental disturbance turned on the effect of the guidance law being based on desired heading instead course became evident. How it effects the results is described in section 6.3.2 and the justification, which is based on poor course measurements during real life testing, will be further addressed in the next chapter. The LOS guidance law is based on a parameter called look-ahead distance, which affects how aggressively ReVolt will return to desired path. To support performance of SBMPC is it important with a look-ahead distance greater than the maximum distance from ReVolt to desired path. This will ensure that the LOS vector points forward towards the end goal, opposed to straight back onto the path. All simulations are conducted with look-ahead distance of 150 meters. The side effect of a large look-ahead distance is ReVolt using more time to proceed back to desired path after a COLAV maneuver. However following the exact path is not the scope of this algorithm, as long as the general direction is correct.

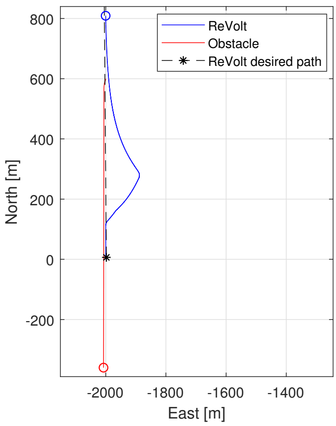
6.1 Simple simulations - single obstacle

The simulation study is introduced with some simpler simulations, confirming expected behaviour from ReVolt. These simpler cases will include only a single obstacle and there will be no disturbances. The five cases that will be tested in this part are head-on, crossing from starboard, crossing from port, overtaken and overtaking. These are the standard cases used for testing collision avoidance, as they are distinctly defined scenarios were the correct course of action is clear. The obstacles will be sailing in a straight line, with constant speed and course, resulting in ReVolt having to take action in all cases, to avoid collision.

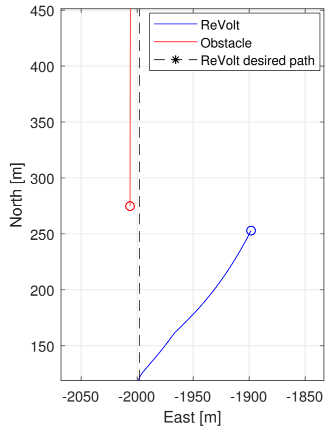
6.1.1 Head-on

A head-on scenario is played out when an obstacle is approaching ReVolt with nearly reciprocal course. According to COLREGS should both vessels, in this situation, initiate a maneuver to their starboard side, avoiding collision with the other vessel. As the obstacle will instead keep on, ReVolt would have to do all the maneuvering. The correct action is therefore a clear, predictable maneuver towards starboard making the other vessel appear on port side, with a sufficient distance, while passing. This test case is set up with initial distance of 600 meters north, where ReVolt is going south to north and the obstacle in the opposite direction. Both vessels sail at a speed of $1\,\mathrm{m/s}$.

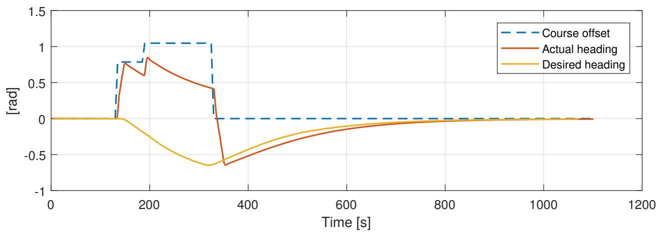
The results are presented in figure 6.2 and are quite satisfactory. Collision is avoided with a smooth evasive maneuver, passing the obstacle with closest point of approach (CPA) at 110 meters, showed in figure 6.2b. Any CPA over 100 meters is satisfactory with respect to the chosen D_{SAFE} value of 100. The maneuver is to starboard side, which is correct according to COLREGS. The first change in course offset is a bigger jump ensuring that ReVolt will be able to avoid collision with an adequate distance. Later the course is increased by a step which is expected behaviour 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



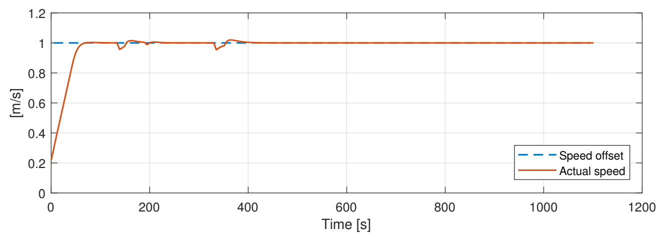
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

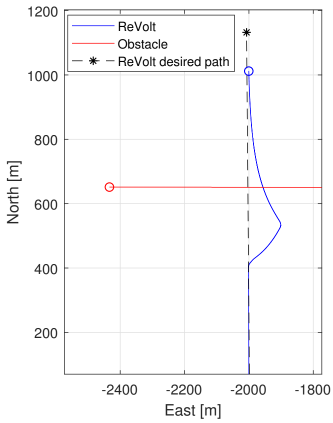


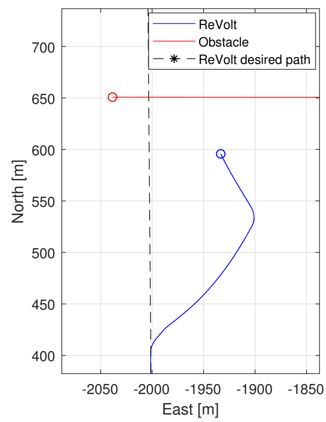
(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.2: Simple head-on scenario.

step-wise increase in course offset to keep following the predicted optimal path. After the obstacle is passed, a zero course offset is given for ReVolt to get back to the nominal path. This behaviour is portrayed in figure 6.2c, showing course offsets, desired heading and actual heading of ReVolt. The reason for showing desired course together with heading is because the guidance law assumes that course equals heading, and therefore apply the desired course offset directly to the desired heading from guidance. The speed was kept constant trough the entire scenario, which can be seen in figure 6.2d showing the speed offset and the actual speed. The chosen speed offset is multiplied with the desired speed from the guidance law, meaning a speed offset of 1 is nominal speed.

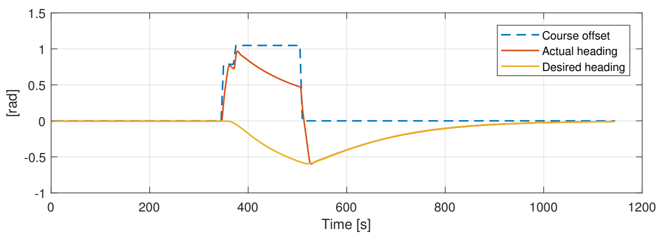
6.1.2 Crossing from starboard



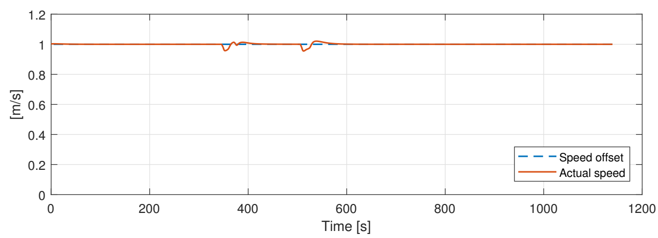


(a) Positions and desired path of ReVolt and obstacle.

(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.3: Simple crossing scenario from starboard side.

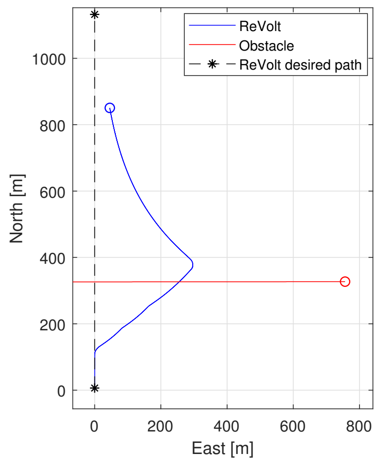
Next is a simple crossing situation with the one obstacle crossing in from starboard side. In this case COLREGS states that the ASV is the keep way vessel and the obstacle is stay on vessel. Hence ReVolt is the one supposed to alter his course to avoid collision. The correct way of doing so, according to COLREGS, is to make a maneuver to starboard side and pass behind the crossing obstacle. In the test setup ReVolt is sailing south to north with the speed of $1\,\mathrm{m/s}$ and the obstacle is passing from east to west with the same speed. Initial position of the obstacles is 600 meters east of ReVolt and 650 meters north. The resulting simulation is plotted above in figure 6.3.

The actions taken in the simulations are according to COLREGS and leads to a smooth evasive maneuver. The closest point of approach between the two vessels is 119 meters which is more then enough to be considered a safe distance to pass. The same step-wise behaviour from the head-on scenario occur in this case and will be present in all other cases, as it is how the SBMPC algorithm is designed. After a course offset to starboard side for some time the algorithm predicts that it is safe to set the offset to zero and start returning to desired path. The speed is not changed during the scenarios, as is the desired behaviour.

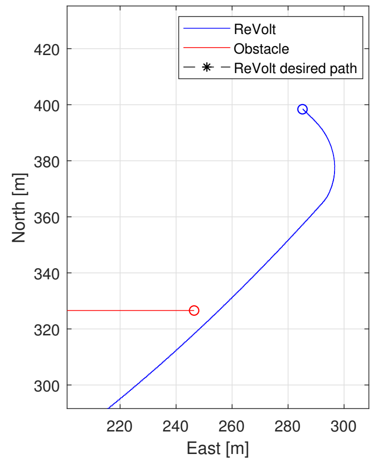
6.1.3 Crossing from port

The opposite crossing situation were the obstacle is approaching from port side is a more complex scenario. As these elementary simulations are using simple obstacles that will go on no matter what, it will always be up to ReVolt to move out of the way. According to COLREGS would the correct action for ReVolt in this case be to just keep on and it is the obstacles which is named give-way vessel, which should alter its course to pass behind ReVolt. But as the obstacle do not act ReVolt is forced to do something when the boat enters the hazards area on a collision course. As the SBMPC algorithm still favours starboard it should turn to starboard passing well in front of the obstacle. In this particular test case ReVolt is sailing from south to north with a nominal speed of $1\,\mathrm{m/s}$. The obstacles is passing in from the west going east with the same nominal speed of $1\,\mathrm{m/s}$. The initial position of the obstacle is 300 meters west and 300 meters north of ReVolt's initial position.

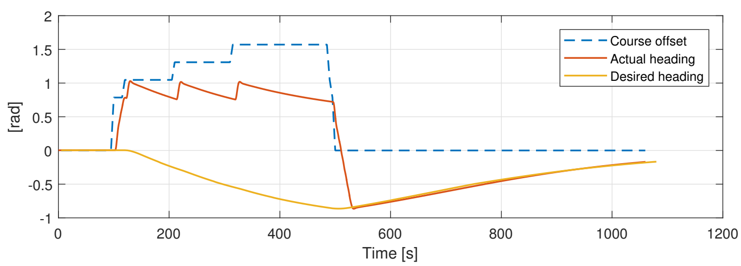
The results are satisfactory and the plot in figure 6.4 confirm the expected behaviour. ReVolt alter its course to starboard side passing in front of the crossing obstacle at a CPA of 85 meters, but when ReVolt is right in front of the obstacle the distance is even larger. This CPA is just smaller than the desired distance, and is a result of ReVolt acting at the last minute, since it is not the give way vessel, combined with need for even more accuracy in the tuning. Finding the best balance between all tuning parameters, to have a high enough CPA and no oscillations in absolute all cases is difficult, and there are other factor which come into play as well. Taking the guidance scheme into consideration is one such factor, that might better performance in this case. Further elaborated in section 6.1.4 below, as the effects of this weakness is more evident in the overtaken case. All things considered is the important part of this case for ReVolt to act as early as possible and make a predictable and clear course change to show the other crossing obstacle clearly what it is going to do. Hence if the other obstacle for some reason suddenly starts to act it can also turn to its starboard side still having the opportunity to pass behind ReVolt. After the crossing is over ReVolt returns to the desired path. The speed is kept the same through the whole scenario, which is good as a change of speed during this scenarios would just further complicate the situation.



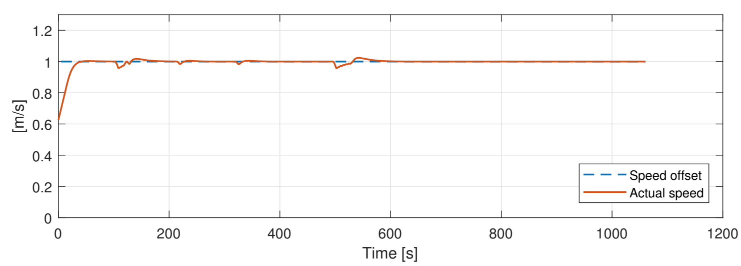
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

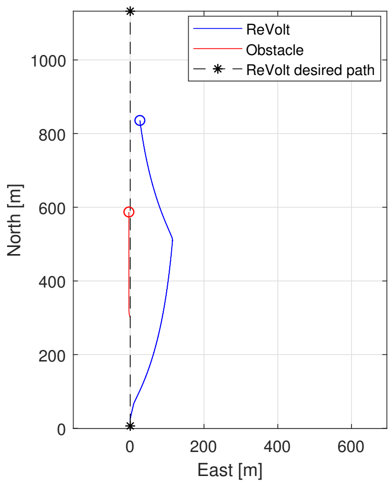


(d) COLAV speed offset and actual measured speed of Revolt.

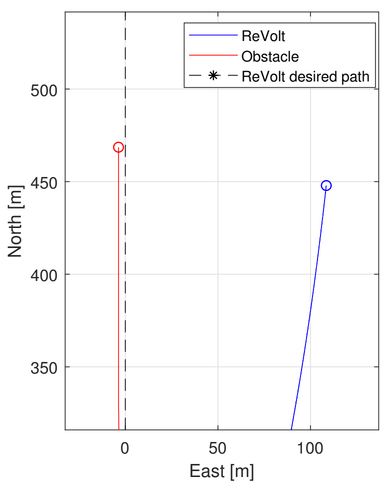
Figure 6.4: Simple crossing scenario from port side.

6.1.4 Overtaking

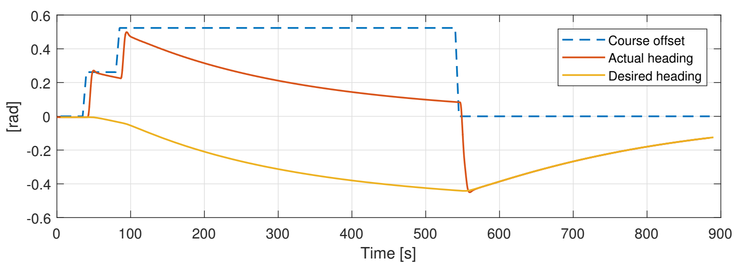
Overtaking can take place two ways, either ReVolt is overtaking another vessel or ReVolt is being overtaken by the other vessel. Both cases will be evaluated here.



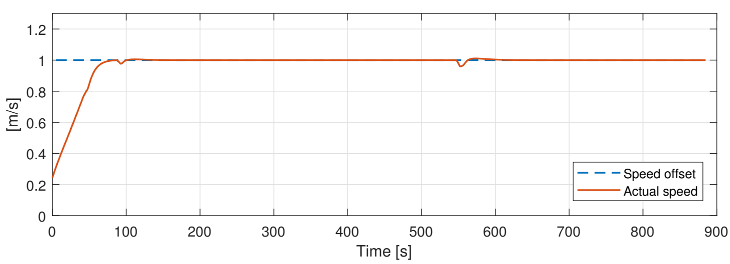
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

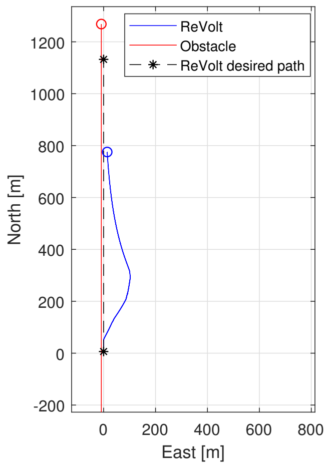
Figure 6.5: Simple overtaking scenario.

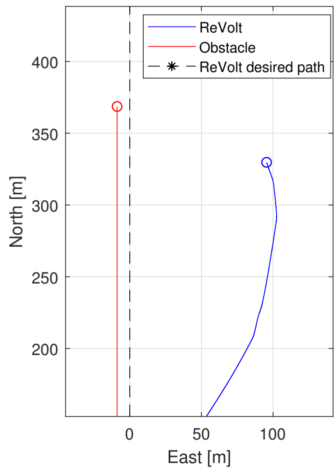
ReVolt is a slow vessel with a top speed of only $1.5\,\mathrm{m/s}$, ergo it has a hard time overtaking other vessels. Nevertheless to show that it is capable of preforming the necessary maneuver a test case where the obstacle is going south to north, same as ReVolt, starting 300 meters in front of it with the low speed of $0.3\,\mathrm{m/s}$ was designed. ReVolt is still keeping normal speed of $1\,\mathrm{m/s}$. According to COLREGS the vessel overtaking another vessel is the one who should keep out of the way, but it do not state which side of the vessel it should pass one. Hence this could be done on either side depending on several factors like other vessels, land or convenience. But as the SBMPC algorithm is built to favour starboard side for the other cases, it will do so also when overtaking. Hence the expected action would be for ReVolt to make an evasive maneuver to starboard side.

Results are plotted in figure 6.5 showing ReVolt taking the expected maneuver to starboard side. ReVolt deviates just enough from the desired path to pass the obstacle at a safe distance, before slowly returning to the desired path in front of the obstacle. The minimum distance between the vessels is 114 meters which is satisfactory. The course is clearly changed and in a step-wise manner, not having an unnecessarily large distance from the desired path. The speed is kept constant at $1\,\mathrm{m/s}$ which is natural as ReVolt would need to use full speed to be able to catch up with obstacle.

When the situations is the other way around, and ReVolt is being overtaken, it is the obstacle that should avoid collision by altering its course. As that will not be the case here, ReVolt will have to take action to avoid collision in this case as well. To show the different behaviour in this scenarios two slightly different test cases were set up. In both cases ReVolt go from south to north, with nominal speed of $1\,\mathrm{m/s}$. The initial distance between ReVolt and the obstacle is 400 meters south, but in the first case the obstacle is almost straight behind ReVolt, just shifted a few meters to port side. While in the second case the obstacle is shifted 40 meters to starboard side. The obstacle is sailing at $2\,\mathrm{m/s}$ speed in both cases.

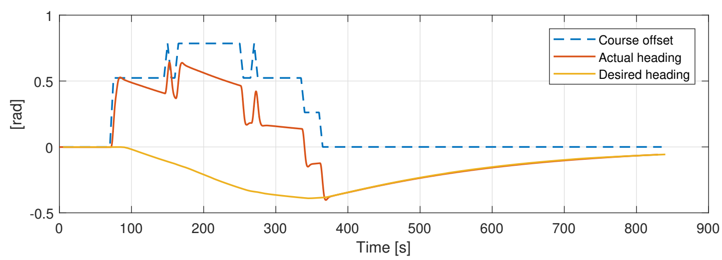
In the first version of the scenario ReVolt alters its course to starboard side to avoid being run into from behind, plotted in figure 6.6 below. This would be the expected course of action as the algorithm is in general built to favor starboard maneuvers, even though COLREGS do not state which way is correct in this case. The CPA is 109 meters, which is a large enough distance for a safe passing. The speed is kept at maximum the whole time, as usual. Even though the scenarios unfold in a safe manner, is ReVolt experiencing some undeceive behaviour when firstly changing to a new course offset. These spikes could result from poor balance in tuning between keeping a sufficient CPA and not deviating to much from the path. The over all impact of this behaviour is minor, but it symbolises a bigger weakness in the algorithm. It mostly results from the fact that the prediction takes a constant value as input from the guidance law. The predictions have a current desired course and speed, that only represent a snapshot of how the guidance law works. To remove the spike in control offset the prediction have to take the type of guidance and the desired waypoints into consideration, and hence be able to predict the actual curve ReVolt will follow back to desired path, instead of a straight line away from it. This represent the biggest problem with the current implementations, since if the algorithm knows were ReVolt is headed in the long run, it could choose a better suited course offset to avoid oscillations and ensure a large enough CPA. It helps the algorithm know when the COLREGS scenario is over and not believe it reenters it. SBMPC does also not know that desired heading from guidance will be changed if it changes course offset, thus it might return to path too quickly, causing the experienced spikes.



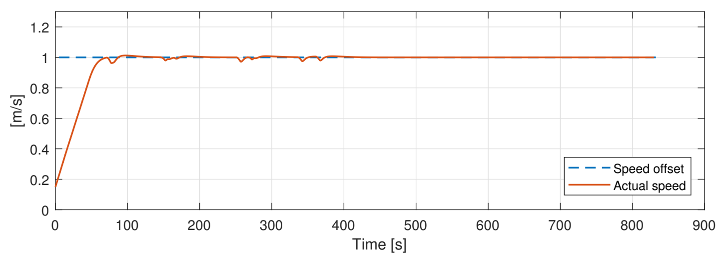


(a) Positions and desired path of ReVolt and obstacle.

(b) Closest point of approach between ReVolt and the obstacle.

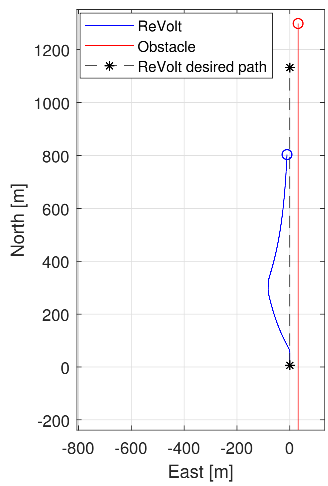


(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

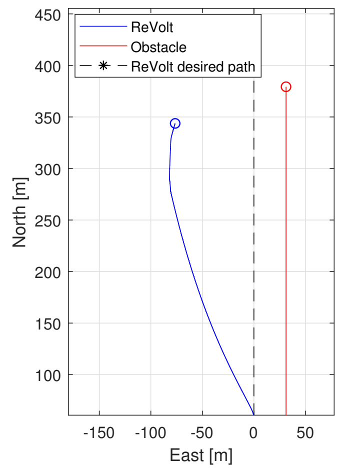


(d) COLAV speed offset and actual measured speed of Revolt.

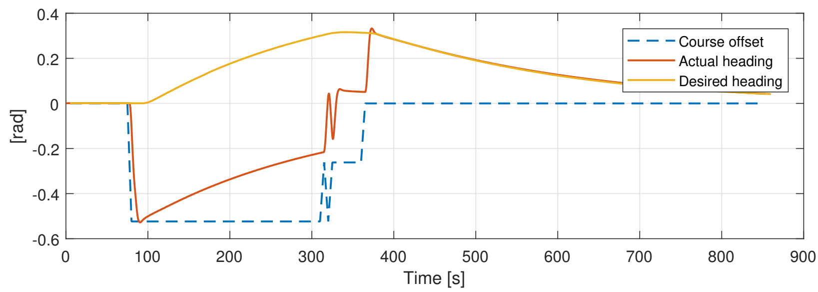
Figure 6.6: Simple overtaken scenario, with obstacle approaching straight from behind.



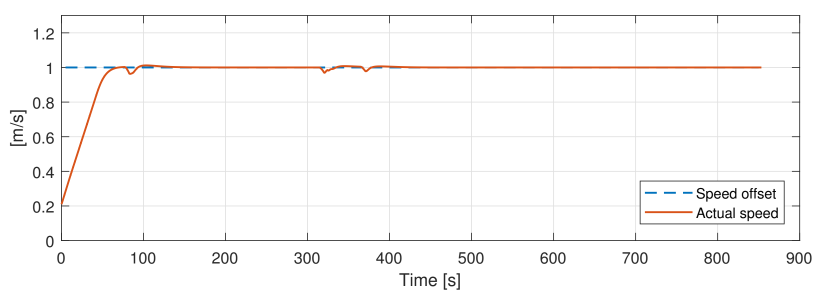
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.7: Simple overtaken scenario, with obstacle shifted to starboard side.

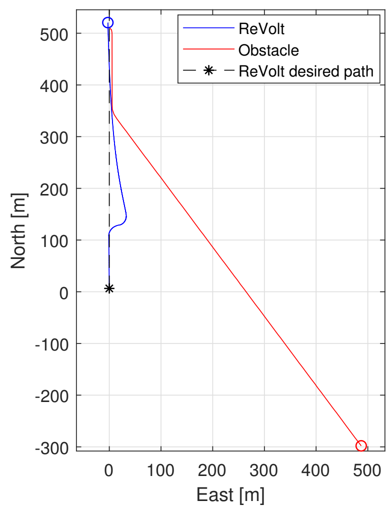
In the starboard shifted case, ReVolt chose to alter its course to port side to avoid collision. The resulting plot is shown in figure 6.7 above. According to COLREGS this is an approved approach as long as the change in course is clear and predictable for the surrounding vessels, which is the case here. The reason why it chose port side is because of the transitional cost. When the obstacles initially enters COLAV area on starboard side, a starboard maneuver from ReVolt will result in transitional cost and is therefore not preferred. The CPA is 111 meters, which is a safe distance. Additionally if the overtaking obstacle suddenly choose to alter its course, it most likely would do so to starboard side, resulting in this being a good way to maneuver. The speed is kept at maximum here as well. There is a small jump in course offset just before ReVolt returns to the desired path. This is not desired behaviour, but does not have a huge impact on the overall course of ReVolt, as it only keeps the offset in 5 seconds before returning to zero offset. One reason for this behaviour might be the asymmetric tuning of course offsets, which favours starboard maneuvers. But most likely is this as well a result of the assumed constant guidance in the prediction. With only the one spike, tuning might be a sufficient solution in this case and increasing the look-ahead distance might help as the desired course from guidance will point towards the end goal instead of close to straight back towards desired path. On the other hand will introducing the full guidance law into the predictions reduce the over all need for tuning, and will likely be a more sustainable solution.

6.2 Complex simulation cases and multiple obstacle scenarios

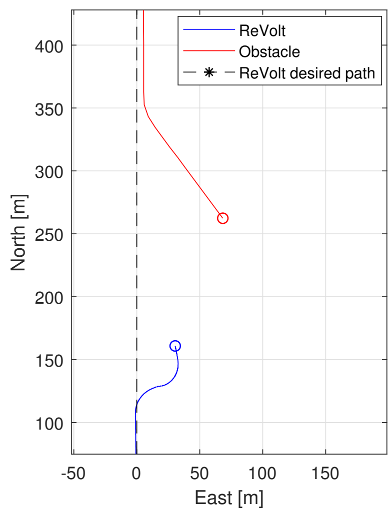
After the general behaviour of the collision avoidance system is confirmed it is time to see how it handles more complex situations. The simple cases above had predefined optimal actions to be taken, but that is not the case in this section and there might be several viable options. Nevertheless the same overall concepts still stands, hence the actions will resemble the once in the previous section, always striving to complying with COLREGS. This section will explore simulations with obstacles that changes course during the COLAV scenario, obstacles running their own COLAV algorithm and multiple obstacles scenarios. With enough computational power is there in theory no limit to how many obstacles the algorithm can handle at once. But for performance reasons the number is limited to four obstacles in this thesis, feeling this will be sufficient for testing many different complex cases.

6.2.1 Single obstacle with course change

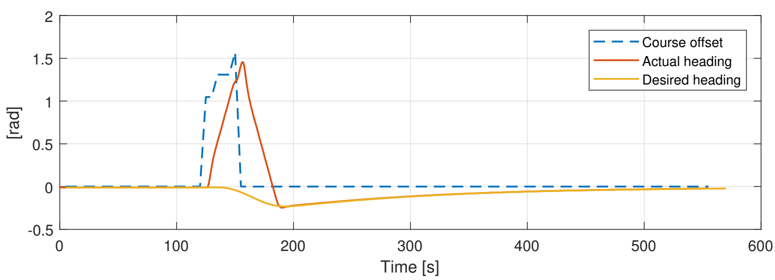
No real life obstacle keep the same exact course at all time. Due to i. e. waves, ocean current, distraction or change of plans the obstacles will suddenly change course mid-scenario. Hence ReVolt should be able to either increase the course angle offset to accommodate the change in course from the obstacle, or be able to abort a COLREGS maneuver, when it is not the safest, most effective action to take anymore. The test case plotted in figure 6.8 is set up with an initial distance of 500 meters north. ReVolt is sailing from south to north as usual, with nominal speed of both vessels at $1\,\mathrm{m/s}$. After 2 minutes the obstacle increase speed to $2\,\mathrm{m/s}$, and after 2.5 minutes it alters course from 180 degrees to 145 degrees.



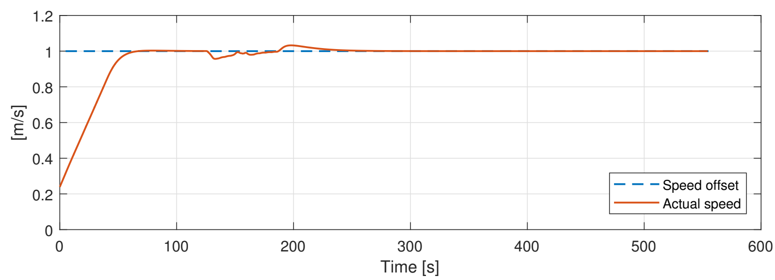
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

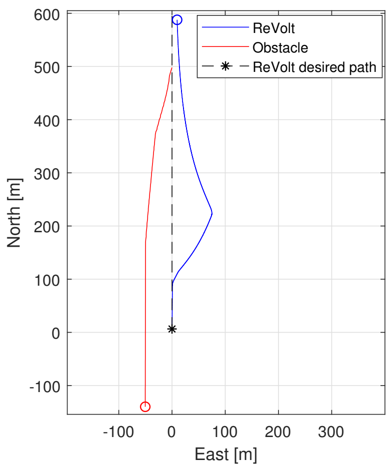
Figure 6.8: Head-on scenario with obstacle changing speed and course.

The beginning of the scenario is equal to a normal head-on scenario, hence ReVolt starts to alter its course to starboard side as expected. Then when the obstacle increases its speed ReVolt still keep maneuvering to starboard side increasing the offset to maximum of 90 degrees to get out of the way as fast as possible, as the obstacle is now approaching at double speed. Then when the obstacle change course as well, SBMPC reevaluates the situation, and abort the started COLREGS-compliant maneuver due to the drastic change in situation. This is an important property of the SBMPC algorithm, showing the adaptability of the method. Avoiding collision and staying close to desired path are prioritised in contrast to strictly following the rules of the sea.

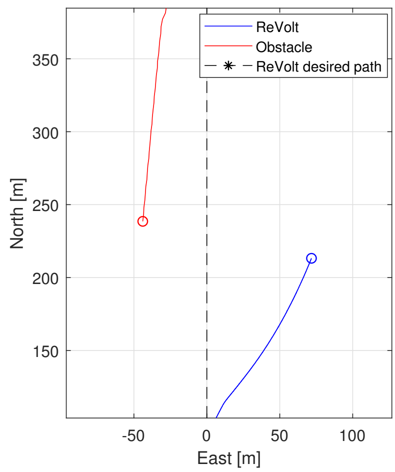
6.2.2 Single obstacle with VO

Another, more realistic way of introducing more dynamical movements from the obstacles, is for them to run their own collision avoidance algorithm. Optimally that would be the same SBMPC algorithm, making comparison easier, but the algorithm implemented into the simulator is VO. Nevertheless this case show how the SBMPC algorithm handles the more realistic head-on case, where the obstacle also alters it course according to COLREGS. The case is set up with a desired path for ReVolt to follow going north and the obstacle initially positioned 500 meters in front, going south. Both vessels sail at $1\,\mathrm{m/s}$.

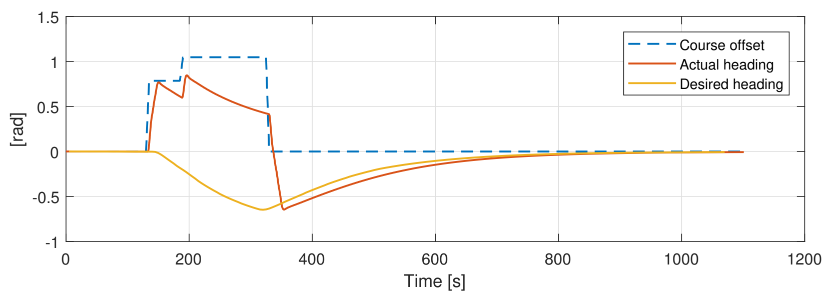
Figure 6.9 presents the result of activating collision avoidance on the obstacle. The obstacle starts altering its course right away, which is a result of the VO tuning parameters. The obstacle detection distance is set quite high as well as the desired CPA. The reason for this is the algorithm not working optimally to begin with, hence starting the evasive maneuver earlier improves performance, resulting in a more realistic case for ReVolt. Regardless of the performance of the VO, is the necessary course offset ReVolt have to take reduced from 60 degrees to 45 degrees. Still keeping the large CPA around the same distance at 118 meters. Showing that the SBMPC algorithm will adjust how aggressive the evasive maneuvers are depending on how the scenarios evolve. Full speed is kept throughout the scenario, as usual.



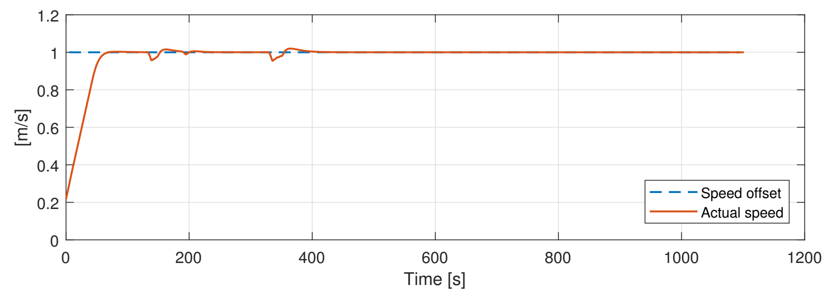
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.9: Head-on scenario with obstacle changing speed and course.

6.2.3 Head-on - Multiple obstacles

The rest of this section will encompass multi obstacle scenarios. Both with several obstacles approaching from the overall same direction and combinational cases where they approach from all different direction. Over all should the expected behaviour still resemble the one from the corresponding simple cases in section 6.1 above, although some deviations are to be expected.

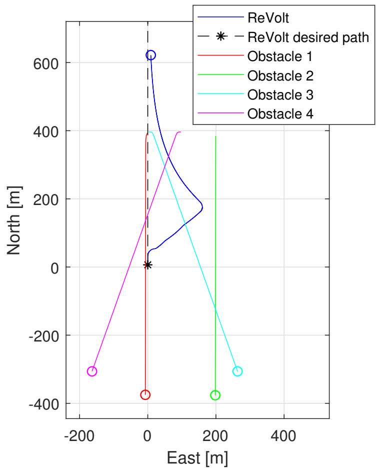
First is a head-on case where all four obstacles are approaching ReVolt from approximately the same direction. Three are close to the desired path and the last is shifted 200 meters east. The obstacles details are presented in table 6.1. ReVolt's desired path goes from south to north and the speed is $1\ \mathrm{m/s}$. Results are plotted below in figure 6.10, where ReVolt's path is blue as before and the obstacles are plotted in different colors.

Ship	North init. pos. [m]	East init. pos. [m]	Course [deg]	Speed [m/s]
1	400	0	180	1
2	400	200	180	1
3	400	0	160	1
4	400	100	-161	1

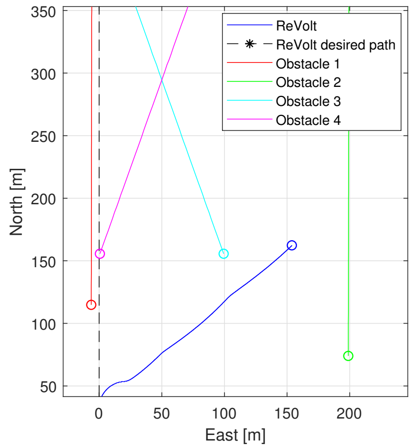
Table 6.1: Obstacle specifications in multi head-on scenario.

The over all performance of the algorithms in this scenario is satisfactory as ReVolt avoids collision with all four obstacle and return to the desired path. The closest point of approach for obstacle 1 is 125 meters, for obstacle 2 it is 75 meters, 53 meters for obstacle 3 and finely 123 meters for obstacle 4. These are all acceptable distances, even though they are not all above 100, since when there is more than one obstacle apparent the algorithm have to compromise to ensure ReVolt travel safely through the whole scenario. Figure 6.10b show a close up of the positions midway through the scenario. This shows that ReVolt pass with obstacle 1, 3 and 4 on port side, while obstacle 2 is on starboard side. Hence it mainly follows COLREGS, but it do not go out of its way to follow it if another action is safer and reduces the distance traveled away from desired path. Like in this scenario, there is no reason for ReVolt to go around obstacle 2, as that would generate a more hazards situation.

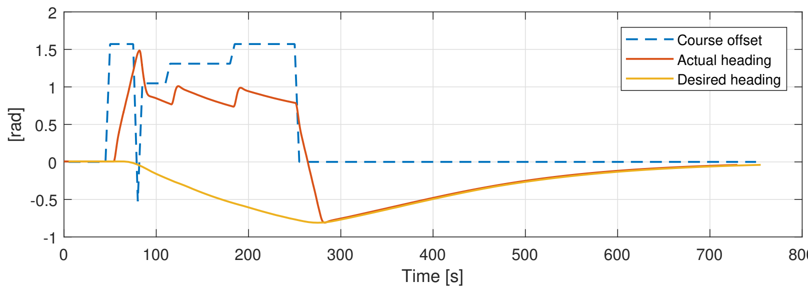
Figure 6.10c presents the course offset chosen by the SBMPC algorithm together with the desired heading from the guidance law and the actual measured heading angle. And figure 6.10d present chosen speed offset and the actual speed of ReVolt. The first change in course results from the first obstacles that enter the COLREGS area. Then the downwards spike happens when the third obstacle enters the area, hence it slightly alters it course back towards the desired path before it recalculates that it is safe to continue in the same direction to avoid collision. This spike is undesired behaviour, even though it did not affect ReVolt's over all path as it was only kept for 5 seconds. More tuning in multi obstacle cases would most likely be a solution to the problem, as it should be a little less aggressive when just discovering new obstacles. The weakness that predictions do not take the full guidance scheme into consideration, not knowing where ReVolt is headed in the end, is also a reason for the jumpy behaviour.



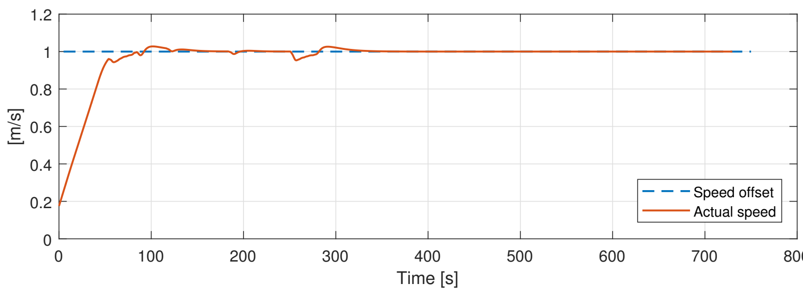
(a) Positions and desired paths of ReVolt and obstacles.



(b) Midway positions of ReVolt and obstacles.



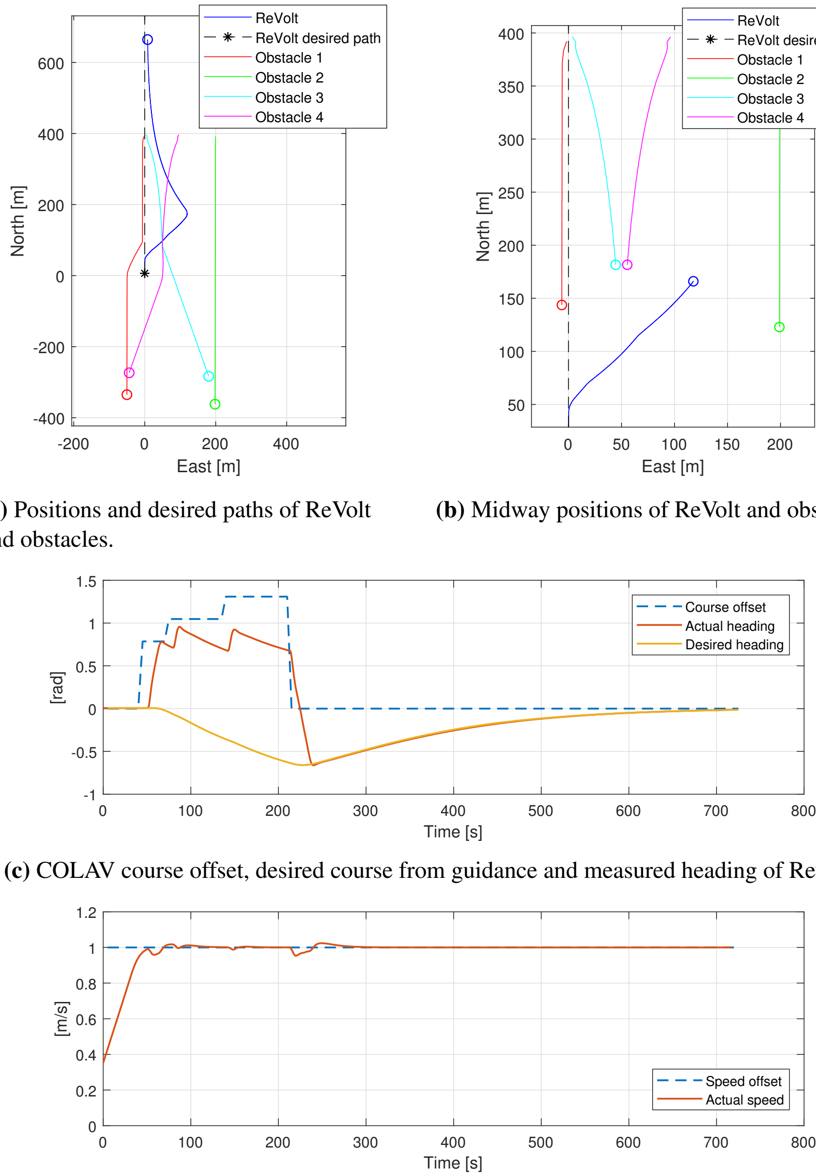
(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.10: Head-on scenario with multiple obstacles.

COLREGS states that in a head-on situation should both vessels alter their course to avoid collision. Hence simulating with obstacles that do not run COLAV, is not as realistic compared to real life. Therefore COLAV is turned on for all the obstacles, and the same multi obstacle head-on case is run once more.



(a) Positions and desired paths of ReVolt and obstacles.

(b) Midway positions of ReVolt and obstacles.

(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.11: Head-on scenario with multiple obstacles. Obstacles running VO.

The results are plotted above in figure 6.11. As can be seen in the plots are the three left most obstacles altering their course to avoid collision. This leads to a much smoother and easier maneuver for ReVolt as it do not have to alter its course as drastically. The close up plot also confirms that Revolt do not deviate as much from the desired path, resulting from the obstacles helping to avoid collision. From figure 6.11c the smooth step-wise offset change can be recognised, similar to the response on the single obstacle head-on case. ReVolt easily avoids collision with all four obstacles in this case as well, with closest point of approach on obstacle 1 at 107 meters, obstacle 2 at 90 meters, obstacle 3 at 74 meters and obstacle 4 at 64 meters.

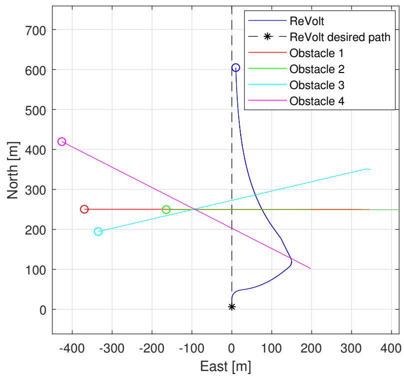
6.2.4 Crossing - Multiple obstacles

The next case is a multi obstacle crossing scenarios, where all obstacles approach ReVolt from starboard side. This was chosen instead of the more advanced crossing from port, because of the poorly functioning VO implemented in the simulator would lead to a far from realistic case, and hence this starboard crossing scenarios was regarded more interesting. The test case is set up with ReVolt sailing south to north with nominal speed of $1\,\mathrm{m/s}$. All the obstacle details are listed in table 6.2. As with the simple crossing from starboard situation is ReVolt expected to perform a starboard evasive maneuver, avoiding collision.

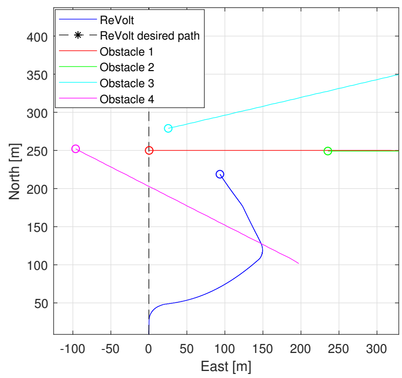
Ship	North init. pos. [m]	East init. pos. [m]	Course [deg]	Speed [m/s]
1	250	350	-90	1
2	250	550	-90	1
3	350	350	-104	1
4	100	200	-64	1

Table 6.2: Obstacle specifications in multi crossing scenario.

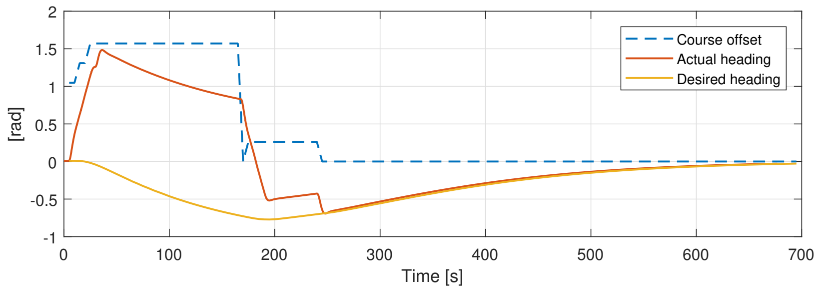
ReVolt behave very nicely in this scenario, avoiding collision and keeping well clear of all four obstacles. Closest point of approach are 97 meters to obstacle 1, 89 meter to obstacle 2, 84 meter from obstacle 3 and 101 meters from obstacle 4. Obstacle 4 enters the case close to ReVolt hence the need for a quite big course offset right away. After obstacle 4 has passed, ReVolt returns to nominal path, just making a small adjustment as a new obstacle enters the COLREGS area at about 180 seconds into the case, shown in figure 6.12c as the second little course alteration bulk.



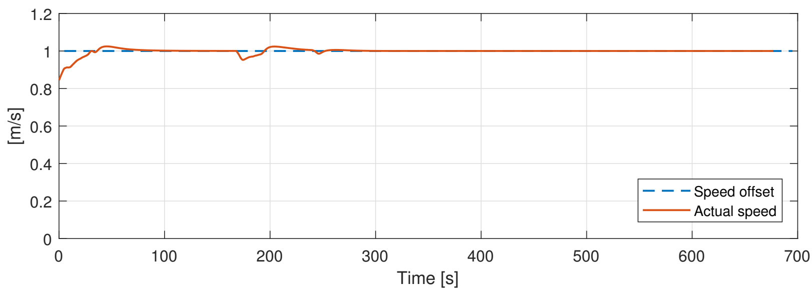
(a) Positions and desired paths of ReVolt and obstacles.



(b) Midway positions of ReVolt and obstacles.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

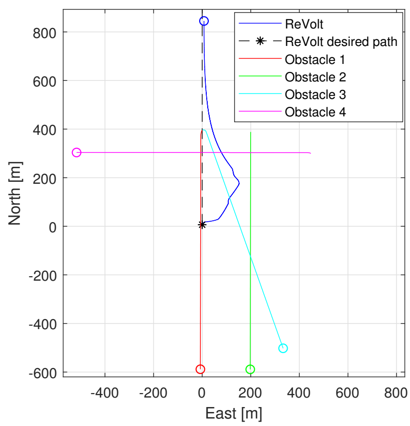


(d) COLAV speed offset and actual measured speed of Revolt.

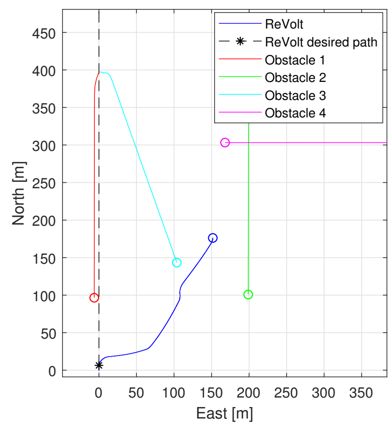
Figure 6.12: Crossing scenario from starboard with multiple obstacles.

6.2.5 Combo - Multiple obstacles

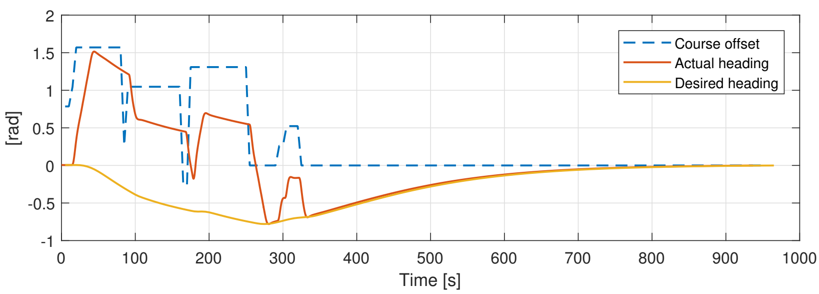
The next two cases are combinational cases, where the obstacles enter from different directions. Firstly a simpler one with mostly head-on obstacles, and one crossing from starboard side is conducted. The next will be more complex, with obstacles entering from 3 different sides. Both test cases is set up with the standards path for ReVolt going north with speed $1\ \mathrm{m/s}$. The obstacles from the first case is described in table 6.3 and the results are presented in figure 6.13.



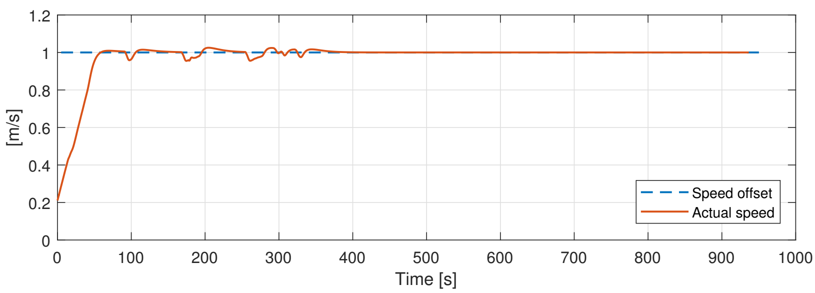
(a) Positions and desired paths of ReVolt and obstacles.



(b) Midway positions of ReVolt and obstacles.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.13: Combinational scenario with multiple obstacles. Approaching head-on and from starboard.

Ship	North init. pos. [m]	East init. pos. [m]	Course [deg]	Speed [m/s]
1	400	0	180	1
2	400	200	180	1
3	400	0	160	1
4	300	450	-90	1

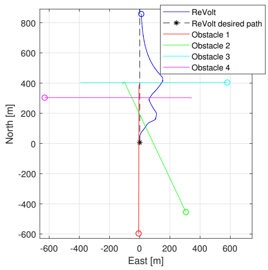
Table 6.3: Obstacle specifications multi combo scenario, with obstacles mostly head-on.

Over all performance is OK. Collision is avoided by an acceptable distance of 129 meter, 64 meter, 47 meter and 48 meters for obstacle 1, 2, 3 and 4 respectively. The maneuver performed by ReVolt follows COLREGS as the head-on scenarios are handled by passing with the obstacle on port side, and passing behind the crossing obstacle. The resulting control offsets are jumping a bit more up and down than desired, and it is clear that some fine tuning is needed. By the looks of it, is there a spike in control offset for each obstacle as they enter the hazards area. A way of fixing this, and smoothing out the behaviour would maybe be to increase the D_{CLOSE} parameter, so ReVolt can calculate cost on the obstacle at a further distance, and know earlier that returning to path is not safe yet, as more obstacles approach. Also the need to add guidance to prediction would have helped this problem, and somewhat reduced the rapidly changing course offsets and reduce need for extensive tuning.

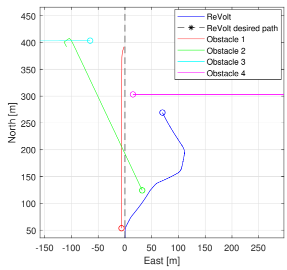
Last of the more complex test cases including multiple obstacles is a combination of obstacles approaching from head-on, starboard and port side. This is quite a chaotic situation, but could easily happen in ports and crowded places, maybe especially relevant for smaller vessels. This case would be simplified if the obstacles were running collision avoidance themselves. But that was tested out, and there were close to no difference in behaviour of the obstacles. This results from the VO algorithm not working properly, hence the test case with VO activated was left out and only the case where ReVolt would have to do all the maneuvering to avoid collision is presented here. Same as the other cases ReVolt follow a path from south to north with speed fo $1\,\mathrm{m/s}$ and the obstacles are presented in table 6.4 below.

Ship	North init. pos. [m]	East init. pos. [m]	Course [deg]	Speed [m/s]
1	400	0	180	1
2	400	-100	154	1
3	400	-400	90	1
4	300	350	-90	1

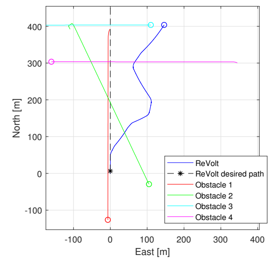
Table 6.4: Obstacle specifications multi combo scenario, with obstacles from all different directions.



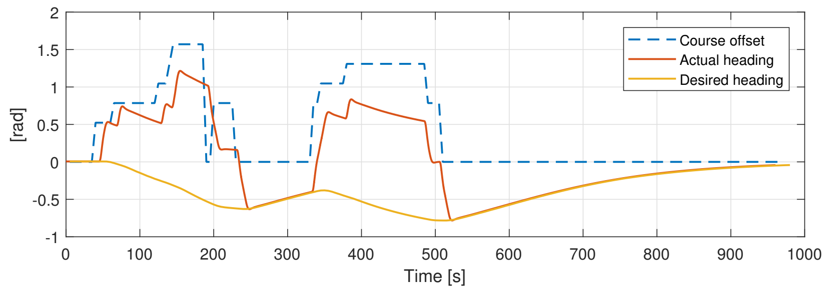
(a) Positions and desired paths of ReVolt and obstacles.



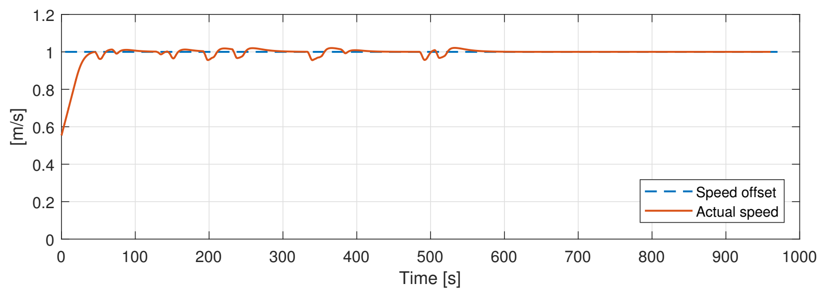
(b) Midway positions of ReVolt and obstacles.



(c) Late midway positions of ReVolt and obstacles.



(d) COLAV course offset, desired course from guidance and measured heading of Revolt.



(e) COLAV speed offset and actual measured speed of Revolt.

Figure 6.14: Combinational scenario with multiple obstacles. Approaching head-on, from starboard and from port.

Results from the scenario is plotted are figure 6.14. This is a very complex situation where ReVolt is getting no help to avoid collision, thus a somewhat wavering behaviour would be expected. The distance between the obstacles are quite large in comparison to D_{CLOSE} as explained above, increasing it would probably smooth out the behaviour somewhat, but likely not all the way. As obstacle 3 is approaching the scenario much later than obstacle 1, would ReVolt want to proceed back to the desired path in the mean time, to not deviate too much from the desired path when it again have to alter its course to avoid collision. In addition is it essentially not ReVolt's responsibility to avoid collision in the last part of the case, hence the offset taken will be a last minute decision as it becomes evident that the crossing obstacle will not alter its course. In figure 6.14c it becomes evident that ReVolt barely passes obstacle 3, and the CPA is 32 meters, which is a bit to close for comfort, especially when passing in-front of an obstacle. The CPA of the remaining obstacles are 90 meters to obstacle 1, 102 meters to obstacle 2 and 76 meters to obstacle 4, which are all sufficient.

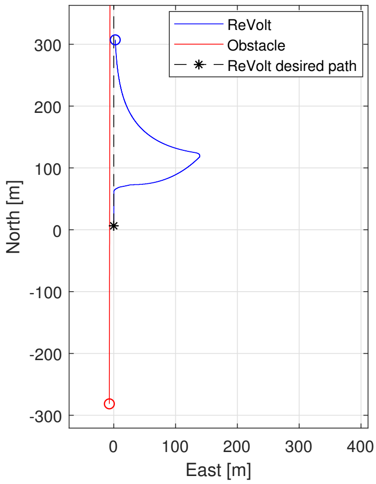
6.3 Simulations with weather disturbances

An important step towards realistic simulations is to simulate with realistic environmental forces like wind, waves and ocean current. CyberSea have the opportunity to simulate these forces and they are used by setting their speed and direction parameters. This results in a large set of possible weather conditions. Though as mentioned in section 5.2 ReVolt has a low top speed and do not handle strong environmental forces very well. Therefor will these simulations only feature ocean current from two different directions to see how the algorithm acts without overpowering ReVolt, leaving it with no power to perform the necessary evasive maneuver. There are no modeling of weather conditions included in the SBMPC algorithm at this time, hence it will have to compensate only based on the input from sensors.

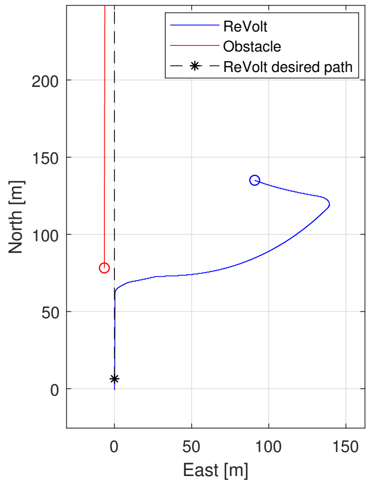
It is important to notice that the inflicted ocean current only affects ReVolt, not the obstacle. It was simulated this way to be closer to the planned real life experiments in this thesis, as they would use purely virtual obstacle not affected by any weather conditions either. Apart from being altered to fit with experiments will this simulation setup be a good way of simulating unknown disturbances on ReVolt. There are several possibilities of factors that might only influence ReVolt. A realistic one could be that one of the two thrusters is stuck and do not turn as it is supposed to, as they also work as the rudder on ReVolt this would affect the direction traveled in an unknown way. Simulating with external influence on ReVolt only will therefore fit with such a scenario.

6.3.1 Current from north

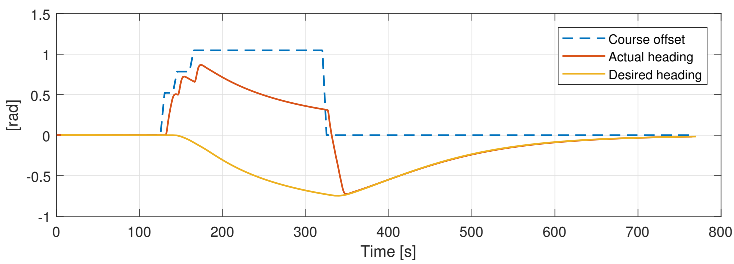
First test case will be a head-on scenario with an ocean current coming directly from north at the speed of $0.9\,\mathrm{m/s}$. The standard test setup is used, where ReVolt follows a path going north at nominal speed of $1\,\mathrm{m/s}$, and the obstacle is approaching head-on from north going south with the same speed. The results are shown in figure 6.15 and are more or less equal to the simple head-on simulation in section 6.1.1 except the whole motion is compressed due to the strong current opposite of the direction traveled. Except for that, which do not lead to any problems, the performance is very good and as expected. The CPA is 114 meters and is more than enough to pass the obstacle safely.



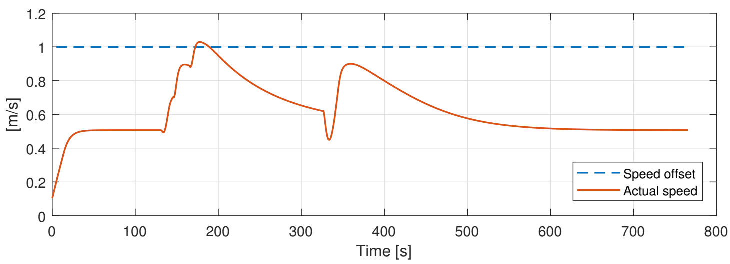
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

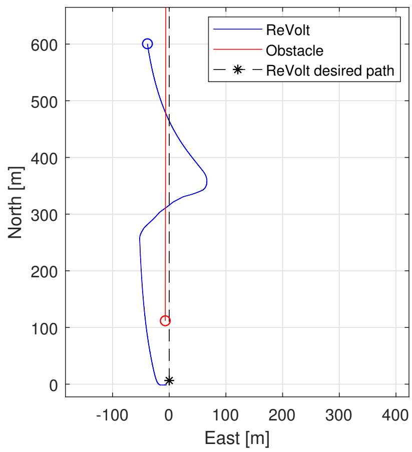


(d) COLAV speed offset and actual measured speed of Revolt.

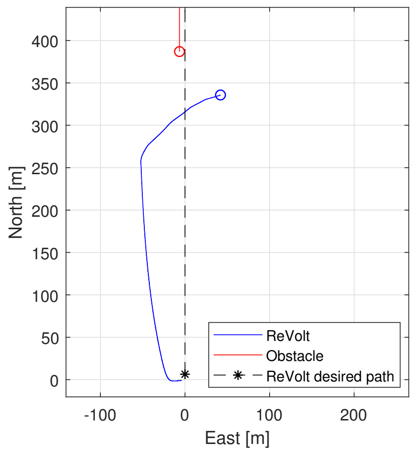
Figure 6.15: Head-on scenario with ocean current from the north.

6.3.2 Current from east

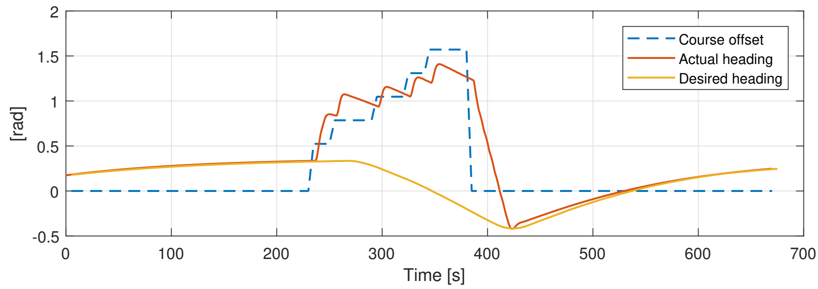
The second test case was affected by ocean current coming in from the east. This time at a much lower speed of only $0.4\,\mathrm{m/s}$.



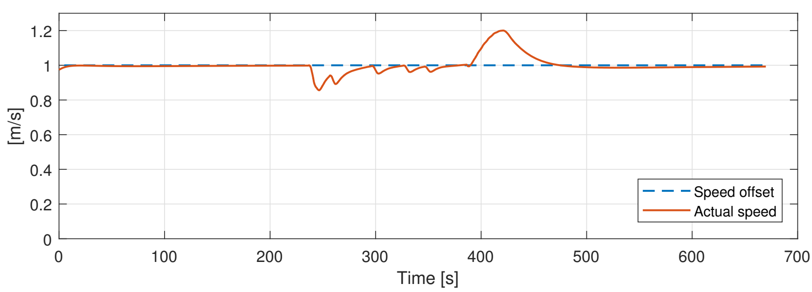
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 6.16: Head-on scenario with ocean current from the east.

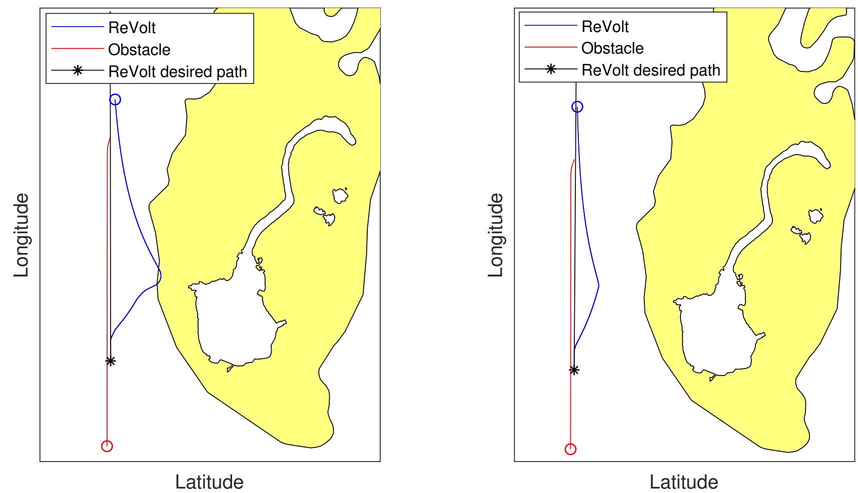
This scenario revealed the disadvantage of using desired heading to follow the path instead of desired course. The reason why this just becomes a problem in this section is that without environmental forces heading equals course. When introducing a current from the side, we get something called a sideslip. The sideslip angle is the angle from the body x-axis to the velocity vector of the vehicle [24]. Hence when the decomposed speed over ground is different from zero in y-direction, here caused by the ocean current, heading does not equal course. The equations describing this phenomenon are given in section 2.3 of the theory chapter. Hence the outcome of this is ReVolt keeping desired heading, but the as course is different there will be an offset from the desired path, seen in figure 6.16a.

Apart from the evident problem of sideslip the test case is still valid to show how ReVolt handles environmental forces. In this test case ReVolt is trying to follow a path from south to north with nominal speed of $1\,\mathrm{m/s}$. The obstacle is approaching head-on with initial distance 800 meters north and speed of $1\,\mathrm{m/s}$ as well. Results in figure 6.16 show that even though ReVolt has drifted quite a lot off the desired path when the obstacle enters hazards area, it is still able to cross over and do the COLREGS compliant maneuver of adjusting its course to starboard side. The CPA is 69 meters, which is OK as a larger distance is not possible in this case since ReVolt is already utilizing the maximum allowed course offset of 90 degrees, and can therefor not do anything else to increase CPA. The change in course is smooth and step-wise as desired and the actions taken are predictable for the other vessel. On the other hand, if the current was measured and taken into account during predictions the SBMPC algorithm might have figured out that it would need a bigger offset right away to be able to avoid collision. The speed is kept at maximum through the whole scenario which will be necessary to overcome the ocean current, to move fast enough for the situation to be safe.

6.4 Simulation avoiding collision with land

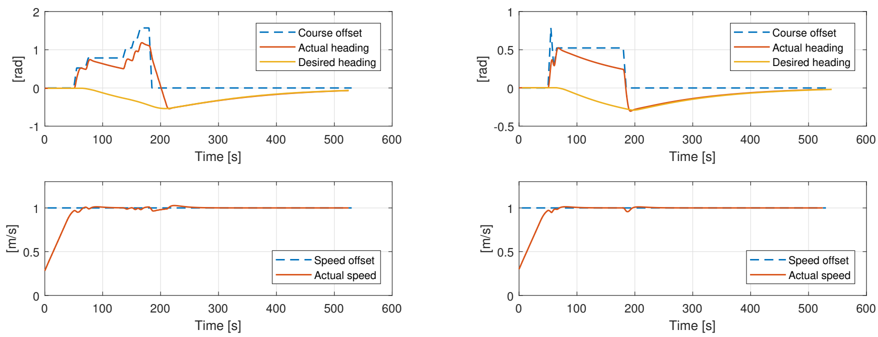
All cases above have tested collision avoidance with no reference to land or possibility of grounding. By adding in the anti-grounding system described in chapter 4, we can run even more realistic test cases not only avoiding collision with the obstacles, but taking the coastline, shallow waters, buoys and beacons into account. This is an essential addition to the overall collision avoidance system. There is an added cost to a set of control offsets if they lead to a collision with land or enters shallow water. Hence ReVolt would have to alter the initial optimal path accordingly. To test out the anti-grounding system a head-on test case close to Munkholmen is designed. Depending on how close to the hazards area around Munkholmen ReVolt is initially placed, it will act differently. Two different initial distances will be used here.

In the first part of the case the initial distance to the hazardous area around Munkholmen is about 100 meters. ReVolt will be sailing from south to north, with speed of $1\,\mathrm{m/s}$ and the obstacle is approaching head-on with initial distance to ReVolt at 400 meters and the same speed. For comparison the case is run twice, once without the map cost activated and once with. The resulting plots are displayed in figure 6.17 where the white area inside the yellow is Munkholmen, and the yellow is outlining the hazards zone to keep away from.



(a) Positions plot without map constraint.

(b) Positions plot with map constraint.



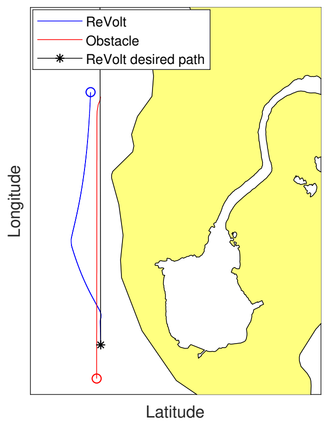
(c) Plots of desired and actual course and speed without map constraint.

(d) Plots of desired and actual course and speed with map constraint.

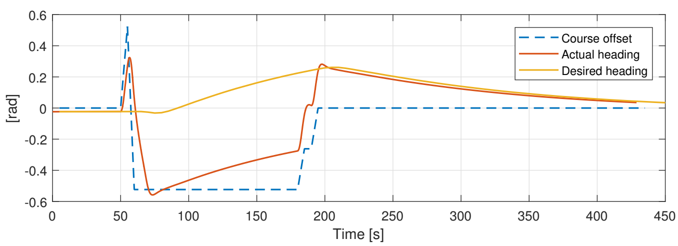
Figure 6.17: Head-on scenario with and without map constraints.

The first plot in figure 6.17a show how ReVolt will enter the hazards area if there is no map restriction. This is much closer to shallow water and land than anyone would like to sail in real life. Figure 6.17c shows the normal course of action taken by ReVolt in a head-on scenario. It will usually have a course offset of up to 90 degrees, smoothly avoiding collision with the obstacle with CPA of 100 meters. Comparing to figure 6.17b it can easily be seen that ReVolt deviates much less from the desired path when avoiding collision with the obstacle, but still adhering to COLREGS making a maneuver to starboard side. From the offset plot in figure 6.17d it is also clear that Revolt use a way smaller offset of only 30 degrees throughout the case. It still have a CPA of 50 meters which is a good distance when needing to compromise and is approximately placing ReVolt right in between Munkholmen and the obstacle. There is a small spike in the beginning, which is from before land is detected, but when starting with the 45 degrees offset it quickly predicts collision with land and alters its course.

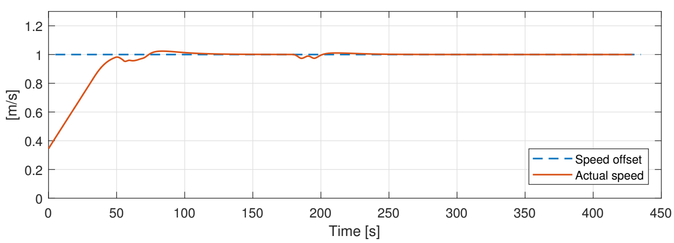
Avoiding collision will always be of the highest priority in the SBMPC algorithm. To prove this, the same test case was set up again, this time closer to the hazardous area around Munkholmen at only 50 meters distance.



(a) Positions plot with map constraint.



(b) COLAV course offset, desired course from guidance and measured heading of Revolt.



(c) COLAV speed offset and actual measured speed of Revolt.

Figure 6.18: Head-on scenario with map constraints closer to Munkholmen.

Figure 6.18 show the results when there is not enough space between the obstacle and Munkholmen to pass safe with a starboard maneuver. The algorithm quickly finds out that an evasive maneuver to starboard side will not be safe, as it will either pass too close to the obstacle or too close to hazards area. Hence to avoid collision a maneuver to port side is performed which is not desired, and could lead to an even more dangerous situation. The optimal action would be for ReVolt to go as close as comfortable towards Munkholmen and then reduce speed to zero waiting for the obstacle to pass. But since the implementation only allow maximum speed, and there is not an option for SBMPC to chose full stop, the algorithm will chose to do a port maneuver as it is the only option. Even though collision is avoided, this maneuver will in most cases risk causing a more dangerous situation than just waiting for the obstacle to pass. Introducing different speed options will be the obvious solution to this, in combination with some presumed need for tuning.

Chapter 7

Experiments

This chapter presents the experiments conducted with the physical ReVolt. Include fixing and tuning of the heading controller, validation of the guidance system as well as the actual COLAV testing.

7.1 Test plan

A simple test plan was drafted before experiments were conducted, with the main goal of validating the performance achieved in simulator. The first priority was to test the five simple cases from section 6.1 of the simulation study, including head-on, crossing from starboard, crossing from port, overtaking and being overtaken. Obstacle information would come from virtual obstacles designed in ReVolt remote control station and updated by the guidance law.

From previous experience with testing physical ReVolt, there was an awareness that not all existing code works as intended. Hence a part of the test plan was to confirm the performance of the necessary supporting systems, including heading controller and the guidance law.

Testing was planned to begin in Dorabassenget where the confirmation of the supporting system would be done. After the initial verification testing would be moved outside the breakwater besides Munkholmen. The area around Ilabassenget was used as a backup testing space, when waves made transport of ReVolt to Dorabassenget risky. The workboat onboard Gunnerus was rented as a following boat.

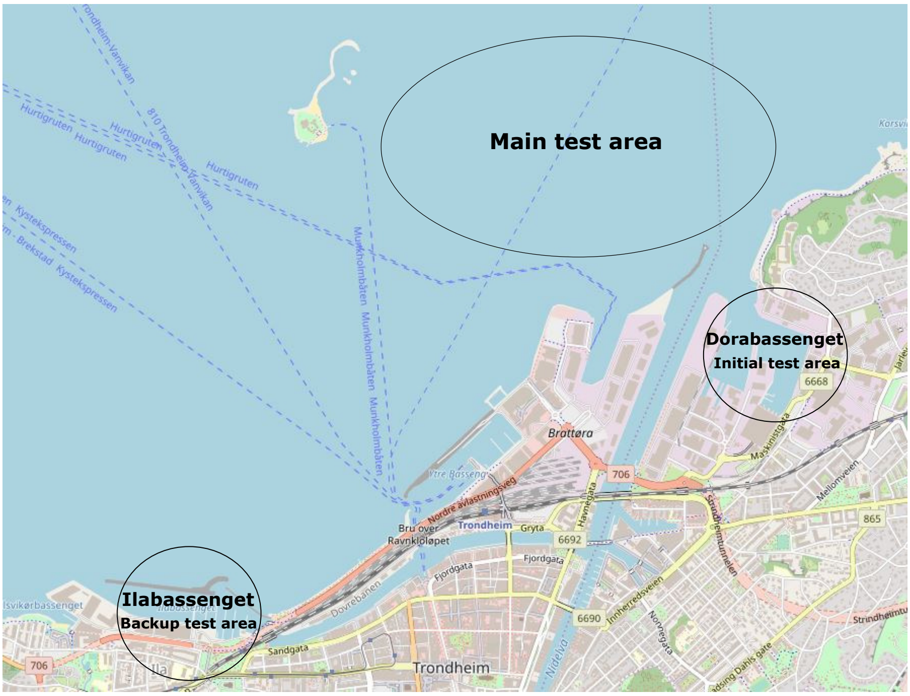


Figure 7.1: Test areas used during experiments [4].

Figure 7.2: The R/V Gunnerus Workboat, used as following boat during experiments.

Figure 7.3: Tom Arne Pedersen assisting experiments as captain and discussion partner.

Figure 7.4: Me handling the control system and the remote control station on separate computers. Confirming behaviour on the guidance system.

7.2 Supporting systems

The two first days of testing were used to get the supporting systems working satisfactory enough for COLAV testing to be performed.

7.2.1 Heading controller

The heading controller onboard ReVolt is designed by another student [30] and was assumed working as there are no problems when testing it in the simulator. The first day of testing was May 8th, after experiments were cancelled on May 3rd because of too much wind and waves. On this first day experiments started by testing the heading controller described in section 2.2, which additionally includes a reference filter. The filter is supposed to slowly change the reference to the desired heading, used when the desired heading makes a bigger jump. This is a necessary part of the heading controller especially when used in combination with the guidance law. In theory it is removing possibly large change in control offset which would leading to maximum offsets on the actuators, and instead it gradually increase the desired heading keeping the actuator offset smaller. In real life on the other hand this was not the experienced behaviour. The plot in figure 7.5 below shows the strange, unwanted behaviour leading to conclude that the reference filter was not working as expected. In addition was it way to slow to reach actual desired heading, making it unfit for COLAV purposes. Problems could not be solved while testing, hence the day was cut short, going back on land to fix the problem.

The temporary solution to the problem was removing the reference filter, which was

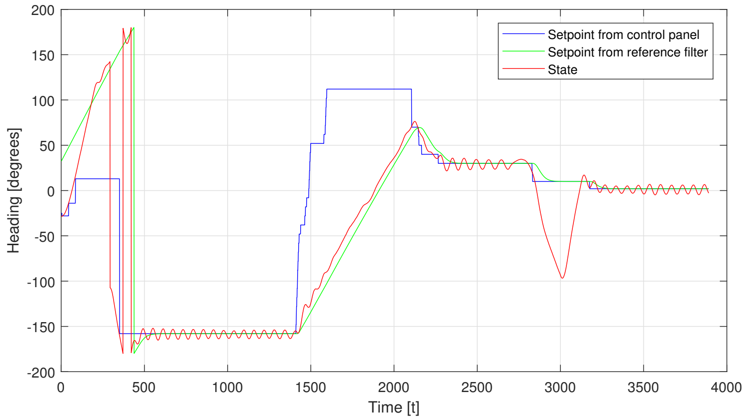


Figure 7.5: Heading controller reference and state. Including reference filter.

the most time efficient fix to be ready for testing the next day. May 9th tests were continued and this day the heading controller worked much better overall. Results are showed below in figure 7.6. It is still oscillating slightly , but is able to follow the desired heading overall. A lot of tuning was attempted just resulting in a more poorly working controller. There is clearly something not working as it is supposed to, but due to lack of time the resulting slightly under damped heading controller was deemed acceptable and the oscillations will be taken into consideration while evaluating the results of the COLAV tests. The spike in state in the plot appear as there is a gap in the state data tracked by the heading controller, because the radio controller had to take over as ReVolt was driving into a buoy.

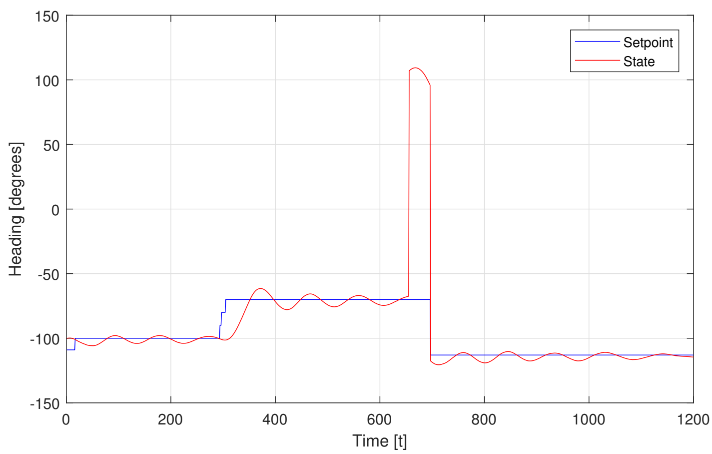


Figure 7.6: Heading controller reference and state.

7.2.2 Guidance system

The next element the CAS is dependent on is the guidance system, also this made by another student in a previous master thesis [30]. The second half of May 9th was used to test guidance, and as there was some waves this day we used the backup testing area at Ilabassenget. To test the guidance system a small route for ReVolt to follow was designed. Tests were performed inside the breakwater, hence there were not a lot of testing space, which should be taken into consideration. Additionally is the same under damped heading controller used, hence the oscillations will appear during guidance as well. The result is plotted in figure 7.7.

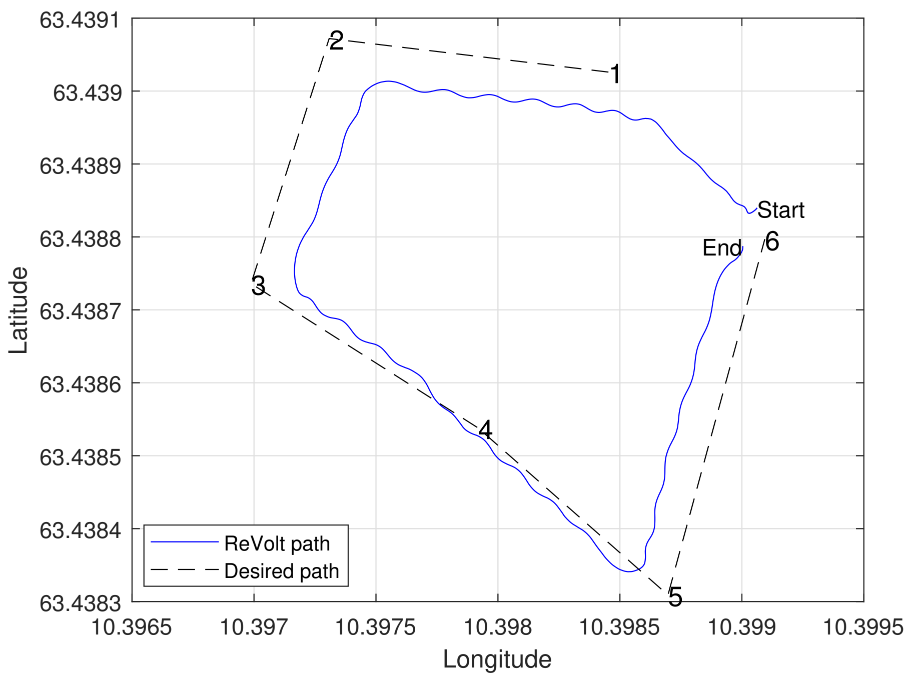


Figure 7.7: Path following using the LOS guidance scheme.

ReVolt is clearly following the given path overall. The offset is a consequence of the look-ahead distance being 100 meters. Hence ReVolt will use more time and space than available in this test, to reach the path exactly. However since the point of this thesis is not to accomplish perfect behaviour of the guidance law, it was good enough to be confident in the overall performance and accepted for use in the collision avoidance testing. Reaching this point took the rest of the day, leaving no more time or battery for collision avoidance testing. However the system was ready to perform the main goal of the test plan during the next test day.

7.3 Collision avoidance tests

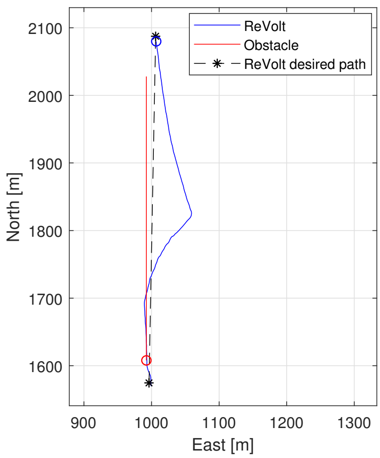
The third and last day of testing was performed on May 14$^{\text{th}}$, assisted by PhD student Andreas Bell Martinsen. ReVolt was launched into Nidelva and transported out to the main test area next to Munkholmen. About 4 hour of continuous testing was conducted, getting through all the planned test cases. The weather conditions during the day was variable as usual in Trondheim, but overall there was only smaller waves and varying amount of low speed wind. The results will be presented in the remaining part of this chapter.

Figure 7.8: Place and method for launching and retracting ReVolt from the water. Here assisted by Andreas Bell Martinsen.

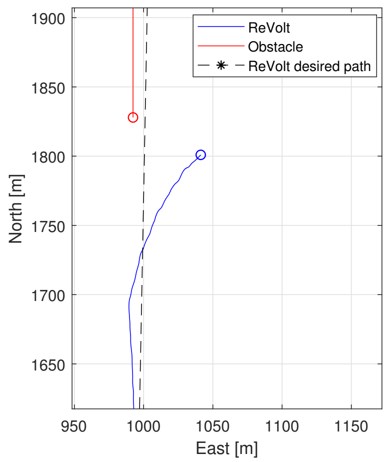
Figure 7.9: Observing ReVolt's behaviour during testing, as well as controlling all parameters on the computer.

7.3.1 Head-on

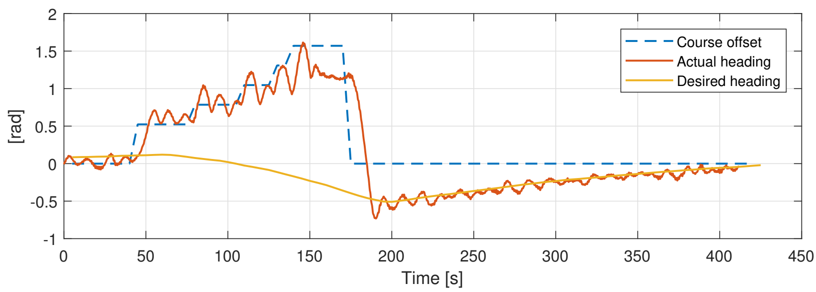
A head-on scenario was constructed designing a path for ReVolt to follow going from south to north, with the virtual obstacle going in the opposite direction. ReVolt sails at nominal speed of $1\,\mathrm{m/s}$, same speed for the obstacle. The initial distance to the obstacle is 400 meters. Figure 7.10a show the desired path of ReVolt together with the actual path, and the position of the obstacle. Figure 7.10b is the section of the scenario representing closest point of approach (CPA), which in this case is 56 meter. This is not a large enough CPA to satisfy the restrictions in the SBMPC algorithm, but since ReVolt is using maximum course offset it is not physically able to achieve a larger minimum distance. Furthermore are the actions taken in compliance with COLREGS, hence the overall performance of the scenario is satisfactory.



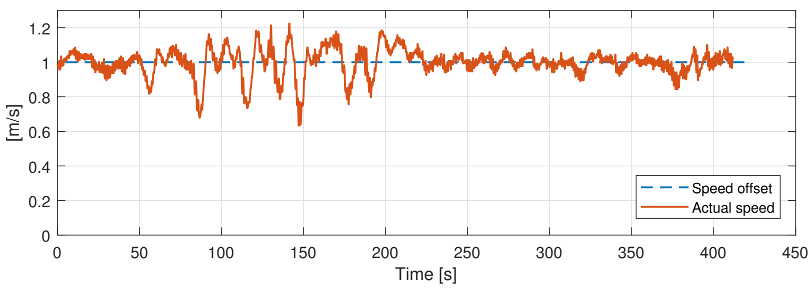
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

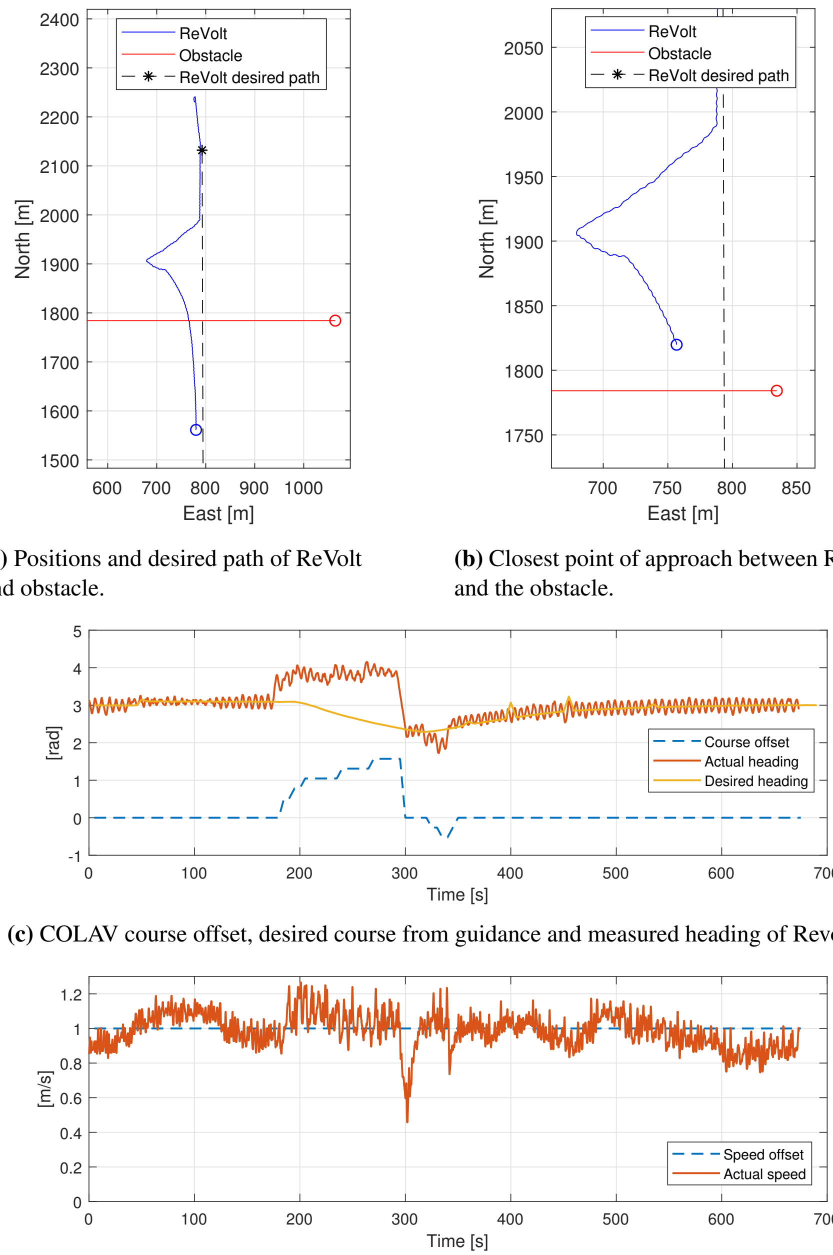
Figure 7.10: Head-on scenario.

Figure 7.10 show the course and speed offsets chosen through the scenario in combination with the desired heading from the guidance law and actual heading. It is smooth and step-wise as would be expected comparing it to the simple head-on simulation from section 6.1.1. However the offsets are overall larger in an attempt to reach an appropriate CPA, which results from the environmental forces reducing the maneuverability, as well as the oscillating heading controller making it harder to follow desired course. The speed is kept at maximum all the way, most likely necessary for ReVolt to be able to perform the needed evasive maneuver, as reducing speed would just increase effect of the environmental forces.

7.3.2 Crossing from starboard

Next scenarios is a crossing situation where the obstacle is crossing in from starboard side. In this scenarios ReVolt is following a path from north to south, and the obstacle is sailing from west to east. Both vessels sail at $1\,\mathrm{m/s}$ and the initial position of the obstacle is 400 meters west and 400 meters south of ReVolt. Comparing to the simulated simple crossing from starboard case is the correct action to alter the course to starboard side, passing behind the crossing obstacle. Figure 7.11a shows ReVolt performing the expected action and returning to the nominal path.

From figure 7.11a it can be seen that while returning to path SBMPC suddenly chooses a control offset of -30 degrees, leading ReVolt to move even faster back to the desired path. This is unwanted behaviour, but only lasts a short amount of time and do not affect the path taken by ReVolt that much, hence the overall impact on the performance is minimal. The main reason for this behaviour is a consequence of the prediction assuming constant guidance, as discussed in the simulation section as well. When the guidance law and desired waypoints are unknown to the prediction, it will for a little while seem like ReVolt enters an overtaking case on the way back to desired path. This is where the negative course offset is applied, trying to avoid collision in the new COLREGS scenarios that appears. This effect could be reduced by increasing the look-ahead distance even more, so the LOS vector always point towards the desired position, instead of more directly in towards the desired path. The CPA is still at 87 meters and the crossing can be granted as safely conducted with close to desired minimum CPA. Nominal speed was kept through the entire scenario.



(a) Positions and desired path of ReVolt and obstacle.

(b) Closest point of approach between ReVolt and the obstacle.

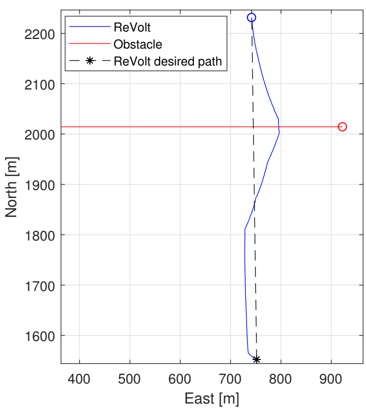
(c) COLAV course offset, desired course from guidance and measured heading of Revolt.

(d) COLAV speed offset and actual measured speed of Revolt.

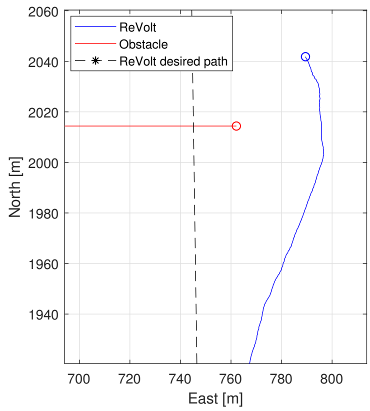
Figure 7.11: Crossing scenario from starboard side.

7.3.3 Crossing from port

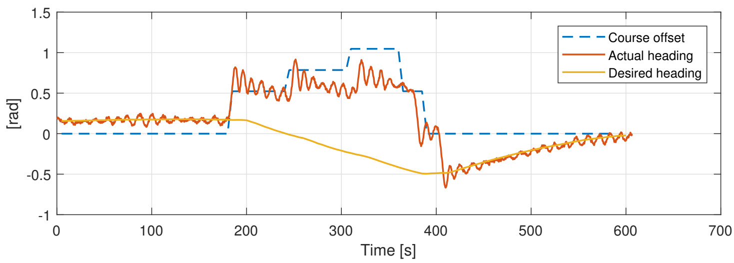
The opposite crossing scenario, where the obstacle cross in from port side, was tested next. ReVolt is following a path going north and the obstacle is coming from west going east. Both sail at nominal speed of $1\,\mathrm{m/s}$. The initial distance between the vessels are 400 meters north and 400 meters west. Figure 7.12 presents the results of this test case.



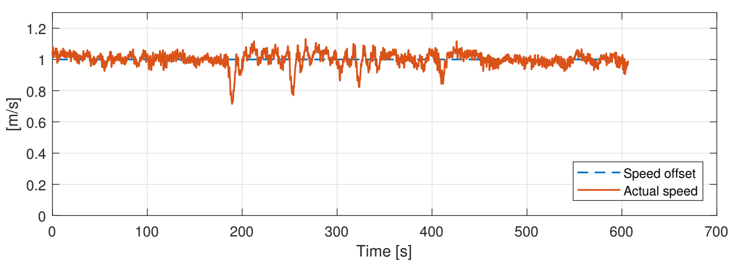
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 7.12: Crossing scenario from port side.

ReVolt is performing an evasive maneuver to starboard side as the obstacle is not followings its duty to give way. This is the expected behaviour when comparing to the simulated case. The CPA of this case is only 39 meters, which is a bit too close, but could likely be increased by punishing collision even harder and increasing look-ahead distance. Figure 7.12c show ReVolt not using maximum course offset, only staying at 60 degrees offset. This, in combination with the influence of the environmental forces, impacts the performance of the algorithm. Figure 7.12d shows that nominal speed is kept all the way, as expected.

This scenario also showcases the problem with the guidance law depending on the heading instead of the course. From the start of the scenario does ReVolt have a deviation from the desired path, it is not able to remove. The mathematical reason for this problem is described in section 6.3.2 and in theory section 2.3, but the justification behind the implementation originates from the poor course estimate on ReVolt. The vector onboard ReVolt estimate the course of the vessel while sailing, but with low speed will the estimate never be any good. Hence for the guidance system to work at all, it is necessary to use the heading as reference instead of the course. Figure 7.13 show the measured course angle compared to measured heading angle. The impact of ReVolt's low speed is most evident in the beginning were ReVolt is gradually increasing its speed from $0\,\mathrm{m/s}$ to $1\,\mathrm{m/s}$.

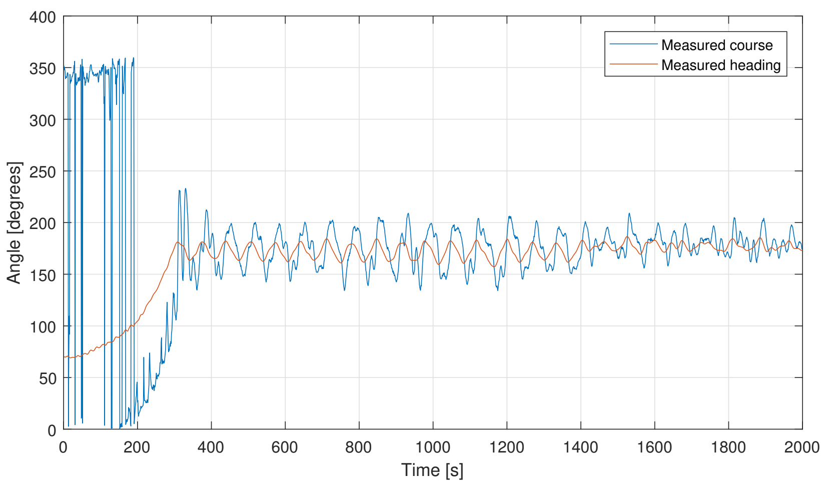
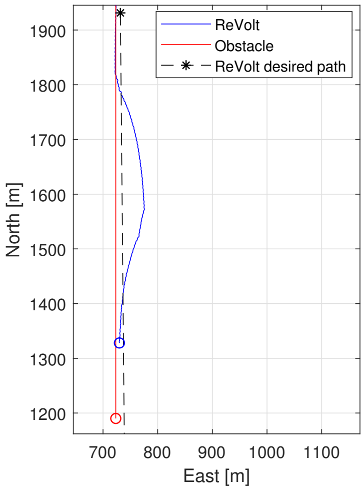


Figure 7.13: Measured course and heading during real life testing.

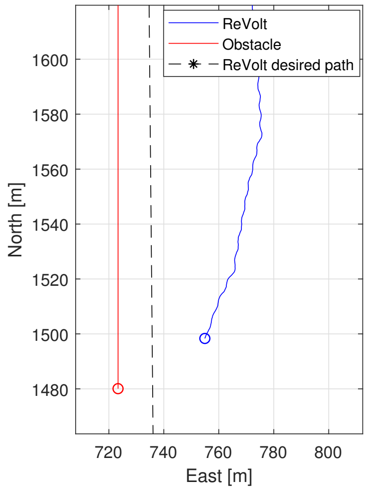
7.3.4 Overtaken

The last two scenarios are overtaking situations. The first one is of ReVolt being overtaken by another obstacle, approaching from behind at a higher speed. ReVolt is sailing south at $1\,\mathrm{m/s}$ and the obstacle is approaching with speed of $2\,\mathrm{m/s}$. The obstacle is starting out

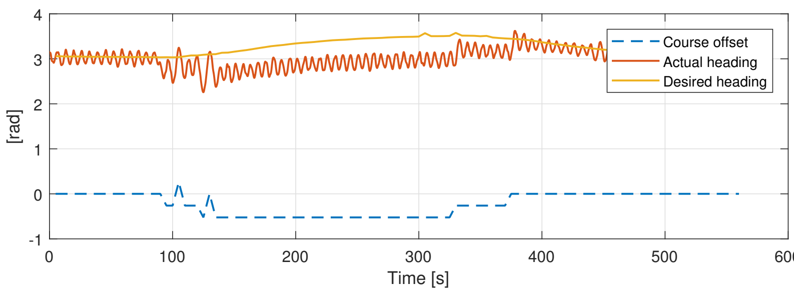
200 meters behind ReVolt.



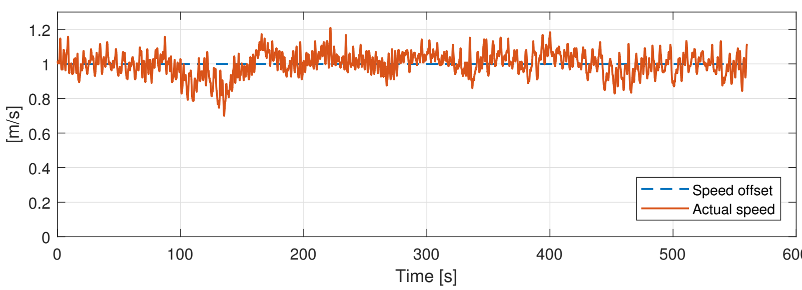
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

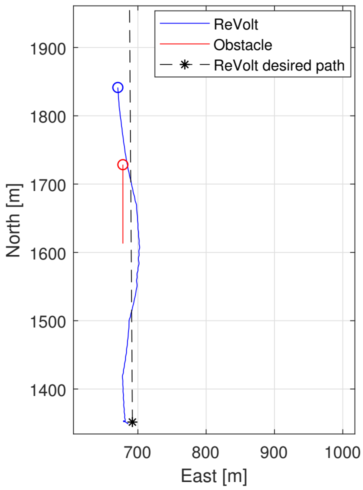
Figure 7.14: Overtaken scenario.

Looking at the over all performance ReVolt avoid collision with a minimum distance of 35 meters which is way too small of a distance, caused by the small course offset used all through the scenario. One reason why SBMPC do not chose to use a larger course offset than 30 degrees might be that the current measured heading input to the predictions is higher than the average caused by the oscillating heading controller. The low CPA can also result from when collision is not weighted heavily enough in the cost function, which with more time should have been adjusted live while testing. The action taken by ReVolt is to port side, happening because the obstacle is shifted a few meters starboard of ReVolt. Comparing to the simulations, the same behaviour is experienced there, which is a result of the added transitional cost. Figure 7.14c shows that even though the overall maneuver looks OK, there are some oscillations in the beginning of the maneuver. This behaviour is a consequence of the fact that the actual guidance law is not taken into consideration while predicting ReVolt's future path. Which created similar behaviour in several of the simulated cases as well. Furthermore is the speed kept at nominal speed throughout the scenario.

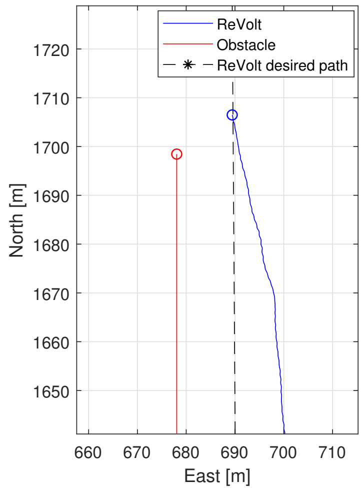
7.3.5 Overtaking

The final scenario tested during experiments was an overtaking situation where ReVolt would overtake another obstacle. As described in the simulations section performing the same scenarios, do ReVolt not have a lot of power to perform this maneuver which will be taken into account when examining the results. The case is set up with ReVolt going south to north and the obstacle starting 200 meters in front of ReVolt with a speed of only $0.3\,\mathrm{m/s}$. ReVolt sails at nominal speed of $1\,\mathrm{m/s}$ as usual.

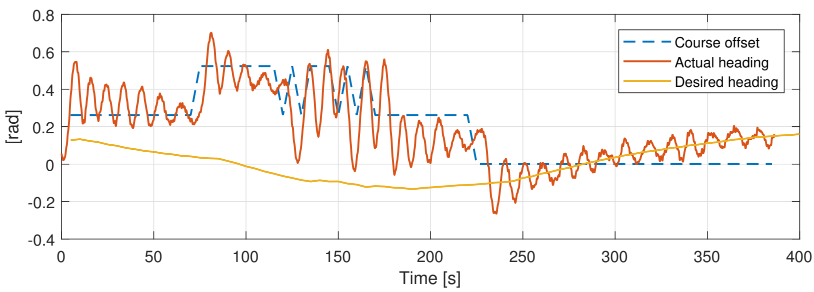
Figure 7.15a shows that ReVolt only executes a small maneuver to starboard side, only just avoiding collision with the obstacle it is passing. CPA is only 13 meters which is not acceptable and would almost be counted as a collision. One factor clearly affecting ReVolt is the environmental forces as it is not able to stay perfectly on the path. Further would it might have been beneficial to start the case with a larger distance between the vessels. This was not done as ReVolt is already sailing slowly, hence 200 meters was decided to be sufficient for ReVolt to make a maneuver. We were also running low on battery during the last test case so the distance was reduced a bit to not risk stopping mid-scenario. This case would also benefit from live tuning, increasing the weight on collision, forcing ReVolt to choose a larger course offset and further increase CPA. The plot of course offset in figure 7.15d show oscillatory behaviour when starting to return to the desired path. This is only partly a tuning problem, as the measurements are impacted by noise during real life testing and thus the tuning from simulations might not be robust enough in real life without including measurement filters. Further does the lack of knowledge about guidance and desired path during prediction increase the need for perfect tuning.



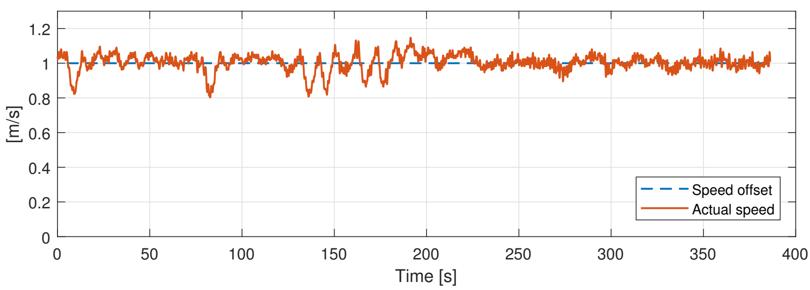
(a) Positions and desired path of ReVolt and obstacle.



(b) Closest point of approach between ReVolt and the obstacle.



(c) COLAV course offset, desired course from guidance and measured heading of Revolt.



(d) COLAV speed offset and actual measured speed of Revolt.

Figure 7.15: Overtaking scenario.

Chapter 8

Discussion

The simulation study in chapter 6 present a diverse set of scenarios in which the SBMPC algorithm performs well. The different cases tested represent a broad specter of realistic situations, starting out with the simple single obstacle scenarios. This section confirmed the overall performance of SBMPC, avoiding collision in all scenarios with no oscillations or indecisive behaviour, and a mostly sufficient CPA. When ReVolt starts maneuvering towards a specific side, proposed by the algorithm, it sticks with it throughout the scenario and do so in a predictable manner. Comparing to the oscillatory behaviour in figure 3.4, experienced in previous implementations of the algorithm, the introduction of COLREGS transitional cost has improved the performance remarkably ensuring better compliance with COLREGS.

In head on and crossing scenarios COLREGS states which direction each vessel should maneuver, but this is not the case for overtaking, where either side is accepted. The algorithm is designed to favour starboard side, as this help aid compliance with COLREGS. Hence in the ideal overtaking case, where the obstacle is approaching straight from behind and not acting to avoid collision, SBMPC would chose starboard side. To confirm the new behaviour introduced with transitional cost, the last simple simulation case in section 6.1.4 was designed as an overtaking case with the obstacle shifted 40 meters east. The obstacle is initially approaching ReVolt slightly on starboard side, prompting a port maneuver from ReVolt to avoid collision. The favoured starboard maneuver in this case will be associated with a transitional cost, due to the obstacle appearing on opposite side in future time, and is therefore weighted heavier than the equivalent port maneuver. As a side effect of doing a port maneuver is some minor spikes, partly resulting form the algorithm being asymmetrically tuned to favour starboard maneuvers. This will lead to an increased incentive to get back to path. The other part of the problem originates from the SBMPC not including the known guidance behaviour during predictions. Predictions are based on constant guidance and do not know the desired waypoints. If the guidance scheme was known to the prediction it could predict the actual path taken by ReVolt instead of the straight line resulting from a constant speed and heading input.

Once more dynamical obstacles are included, changing their speed and/or course during a scenario, there are several more aspects to take into consideration. The initial chosen course of action will not necessarily be the best throughout the whole scenario if an obstacle suddenly changes its course drastically. During simulations it is proven that SBMPC is able to abort a started COLREGS maneuver when the situation changes enough for it to be the better course of action. This demonstrate that avoiding collision is of highest priority in the SBMPC algorithm, not sacrificing safety to follow COLREGS. Another important factor to consider when encountering more dynamic obstacles are how utilization of linear prediction affects performance. The predicted straight line, based on current speed and course will poorly represent future path of an obstacle running collision avoidance. As the obstacles are helping to avoid collision it will likely have minimal impact on hazard levels in the simpler cases, but in multi obstacle scenarios it might not be as straight forward.

When evaluating the more complex test cases with multiple obstacles following the initial confirmation, defining satisfactory behaviour is not as straight forward, but a similar maneuvering pattern would be expected. The overall performance of the multi-cases is characterized by somewhat indecisive behaviour. The SBMPC algorithm has a tendency to reduce course offset all the way down to zero between obstacles, but then straight after have to apply a new course offset to avoid the next obstacle. One reason for this might be poor balance between the selected value of D_{CLOSE} and the prediction time, hence a better tuning combination could reduce the problem. Another possible reason for this behaviour is how multi obstacle cases are handled by the algorithm. The cost function calculates cost individually for each obstacle, choosing the maximum cost to represent the current case. Then cost is minimized over all different control offsets picking the best course of action. A drastic jump in optimal control offsets could occur when the obstacle representing the maximum cost changes from one obstacle to another. This behaviour could also be reduced by including the full guidance behaviour during predictions.

The improved ship model with Euler prediction is able to perform a more accurate prediction of ReVolt's future path. This have resulted in all crossing and head on cases's first course offset being at least 30 degrees. Based on the predictions will the smallest course offset at 15 degrees not be sufficient for avoiding collision at a safe distance, due to ReVolt's slow turning rate. Nominal speed of $1\ \mathrm{m/s}$ is kept throughout all tested scenarios, a behaviour suitable for ReVolt's slow dynamics. The minor fluctuations in speed during simulations result from the dependency between speed and heading, hence when heading changes speed is simultaneously reduced slightly. It will take a lot for the speed offset to change, as it is purposefully weighted heavily at $K_P = 100$. Only when there is no other way out of a hazardous situation will ReVolt alter its speed. The main reason behind this tuning scheme is ReVolt's already low maximum speed, since halving it will drastically reduce ReVolt's maneuverability and thus ability to avoid collision. Change in course is also easier for other ships to notice, consequently the algorithm prefers course change to avoid collision instead of change in speed. Further the algorithm strives to use as little course offset as necessary for a safe evasive maneuver, both concepts are reflected in the simulations results and experiments, especially apparent in the simpler cases.

In addition to avoiding obstacles ReVolt have to avoid grounding. The simulations performed in section 6.4 show promising results, avoiding both the approaching obstacle and Munkholmen simultaneously. The maneuvers performed when restricted by the map are still smooth and predictable, just reduced accordingly. ReVolt further proves the ability to abort a started maneuver and prioritize avoiding collision, completed by a port maneuver which is not the safest action to take. Collision is avoided, but the correct and safe action to take in such a scenario is to stay as close to Munkholmen as possible and stop propulsion until the obstacle has passed. With the current implementation this was not possible as only full speed is available, but even if a stop command was included it is a complex tuning case, which likely would need the map restriction to be a more integrated part of the SBMPC cost function, instead of a separate component added on at the end. Even though the simulations with map restrictions yields promising results when there is enough maneuvering space, there are known weaknesses to the system. There are gaps in the coastline present in the hazard map, which could cause problems if test were conducted close to them, which is not the case here. Further how the curve, representing ReVolt's predicted path, is split up will effect the behaviour in combination with how each part is weighted in the cost function. Further testing i. e. in a narrow passage would be beneficial to better challenge the performance, as well as to find a good combination of tuning parameters. The method of implementation is another known weakness, where an obvious improvement is for the algorithm to run inside the SBMPC algorithm which will increase accuracy as the full 1200 point curve could be utilized.

The stable, satisfying simulation results were highly reflected in the performance of the real life experiments. Overall the SBMPC algorithm performed very well with regard to the added uncertainties of noise, environmental forces and physical components. All planned fundamental test cases were conducted, resulting in a broad understanding of ReVolt physical behaviour during collision avoidance. Collision was avoided in all tested scenarios with more or less clear predictable maneuvers. A factor worth noticing is that the CPA was overall smaller during experiments, than in the associated simulated cases. The assumed main reason for this is the environmental forces, especially waves, affecting how precise ReVolt is able to follow the given speed and course offset. The larger fluctuations in speed during experiments, portrayed in the speed offset plots, are caused by the waves, and furthermore the impact of them varies from case to case. Hence it might not be able to maneuver as the predicted path suggests, not getting as far away from the approaching obstacle. ReVolt is also restricted in how large a CPA it is able to achieve by reaching maximum course offset in some of the cases.

The impact of environmental forces was studied during simulations, illustrating what to expected from the real life experiments where waves, wind and ocean current are unavoidable. Simulations mainly evaluated the effects of ocean current, which could easily be recognised during the experiments as well. The SBMPC algorithm compensate for the slight offset to the desired path, by using a larger course offset to perform a safe evasive maneuver. This positional offset result from the encountered challenge of the guidance law having to base control on heading instead of course, due to poor course measurements. As a result performance is somewhat reduced compared to zero-current simulation cases, but it is not crucial for overall ability to avoid collision. The main scope of the CAS is not to strictly follow a path, thus the results are still useful. During simulations it is evident that strong ocean current reduces ReVolt's ability to maneuver, as ReVolt do not have extra motor-power to compensate for the current. A solution to the problem would be to integrate models of the measurable environmental forces into SBMPC, allowing it to correct for waves, wind and ocean current during predictions. This, in combination with a guidance scheme able to compensate for the sideslip, will likely improve performance drastically. As by now ReVolt is most vulnerable to the environmental forces.

Even though overall performance during real life testing was satisfactory, there were some unexpected oscillations during the crossing from starboard scenario as well as both overtaking scenarios. In any scenario ReVolt is supposed to return slowly back to desired path after the collision hazard is passed, executed by setting the course offset to zero. Whilst in the experimental case it chose to alter the course more drastically, using a negative course offset, which is not nominal behaviour for the SBMPC algorithm. The reason for this might come from some bigger waves, from the larger shipping vessels operating in the area while testing, bad tuning, or noisy measurements, leading the algorithm to believe it was on a hazardous path. Furthermore this case most clearly portray the problems originating from lack of information about the full guidance behaviour during predictions. Specific to this case does the negative course offset result from the SBMPC algorithm believing ReVolt reenters a COLREGS scenario where it have to take action. This behaviour is not presented in the equivalent simulation case due to way less noise, but is visible in the more complex simulation cases. The speed is also severely reduced when the larger jump in heading reference take place, and could additionally contribute to the strange behaviour, reinforcing the belief in ReVolt reentering a hazardous situation. The oscillations experienced during overtaking seems like undeceive behaviour, and should strictly be a tuning problem. This behaviour is not present during simulations, consequently taking time to tune the SBMPC algorithm to be less sensitive in real life situation would be necessary to improve performance, as simulation will never be a perfect representation of real life.

Further, it is evident that measurement noise is affecting the experimental results. This is to be expected, but could be reduced by introducing noise filters on all measured values. The most evident performance issue is the oscillatory heading controller, clearly spotted throughout all scenarios in chapter 7. After a good amount of time spent trying to tune the heading controller while at sea, the conclusion was that either there is a minor bug in the code or the noise from measurements are too dominating for the controller to ever be fully stable. These oscillation are kept in mind when examining the results from the experiments. They may impact the SBMPC algorithm in unknown ways, since they were not present during simulations. SBMPC is built with robustness in mind, to avoid unnecessary change in control offsets. The algorithm need a significant change in cost before it decides to change course or speed, reducing chance of oscillations. This might be a contributing factor to the results still being promising, regardless of the oscillating heading controller.

Testing with virtual obstacles will not be as realistic as real obstacle ships. All the newly introduced uncertainties when performing experiment at sea do not affect the virtual obstacles used in this thesis. That is of course not ideal, as the obstacles sail in a perfect line not affected by the same weather conditions as ReVolt. Neither are they affected by the measurement noise on position, speed or course, overall resulting in less realistic test cases. To compensate for this, simulations were also conducted with perfect obstacles not affected by the same weather conditions as ReVolt. Even though this is not the preferred way of testing, did simulation at least lay a realistic foundation for real life experiments. The introduction of simulated AIS data and the ability to change the parameters of the obstacles has lead to more realistic simulations, which is an important factor in improving performance of the system. It also made it easier to construct more complex cases with several obstacles, obstacles changing speed and direction, as well as incorporating the more dynamical obstacles which are running VO themselves.

Chapter 9

Further work

The broad specter of simulations and the experiments performed during this thesis lay a good foundation for further work focusing on improving real life performance. A simple first step towards better performance is online tuning while performing experiments at sea. Even with realistic simulations, will the tuning never perfectly fit the actual system and some minor adjustments to the tuning parameters can go a long way in improving performance. The simple experiments yielded promising results, hence a next step would be to introduce more complex test scenarios with obstacles changing course, running its own COLAV algorithm as well as multi obstacle scenarios. To address the sometimes occurring minor oscillations portrayed on ReVolt, integrating the full known guidance behaviour into the SBMPC is necessary. That would result in correct prediction of ReVolt's actual future behaviour, with the destination from desired path in mind.

In addition to the experiments becoming more advanced, is it crucial that they become more realistic. The perfectly undisturbed obstacles are too fare from reality, hence a system able to detect real life obstacle should be installed on ReVolt. There are two ways to do this, AIS receiver or camera/LIDAR. Installing an AIS receiver would be the simplest solution, as the code is already designed around the simulated AIS data. The work needed to introduce it in real life is therefore minimal, only entailing a new driver to read received data. A more complex way of detecting obstacles are a senor fusion system between camera and LIDAR. This would also be an important addition to the system in combination with the AIS receiver, as only larger vessels have an AIS transmitter, hence detecting smaller vessels, kayaks, animals and other minor obstacles in the water can only be done with sensors. Introducing such a system onto ReVolt is a large project in itself, as object detection using computer vision methods in combination with object tracking would be necessary features. Merging such a system with the SBMPC algorithm on ReVolt would result in a stand-alone collision avoidance system well equipped for a wide range of realistic tests.

Measurement noise and environmental forces are deemed the two most impacting factors on ReVolt while performing real life tests. Both should be subject for further work, improving simulations to better understand how they impact the system, as well as implementing the necessary improvements. To reduce the effect of environmental forces should they be modeled and taken into account while predicting the path of both ReVolt and the obstacles. This will ensure the algorithm compensate for wind, waves and ocean current reducing the impact of ReVolt's lack of thruster power and maneuverability. With regards to measurement noise, should it be included in simulations as a first step, and that will possibly prompt the need for some measurement filters.

For the collision avoidance system to work at its optimal potential the underlying support systems on ReVolt would have to be improved. Future work will be to stabilize the heading controller, that might include filters, tuning and/or fixing bugs. If it is feasible, a guidance law based on course should be implemented, reducing the vertical offset to desired path caused by ocean current, also helping to reduce impact from environmental forces in general.

The map restriction and anti-grounding help improve the entirety of the system, but have potential for improvements. The algorithm should be implemented directly into SBMPC to improve accuracy, integrate map cost as a part of the existing cost function and enable the algorithm to use a full representation of ReVolt predicted path. When checking for intersection on the whole 1200 point curve in conjunction with a detailed map, there might might occur problems with computing in real time. The solution to this is extracting only the immediate necessary map area around ReVolt, moving it as ReVolt is moving. Further should focus be put on tuning and how to best split the predicted curve for a smooth evasive maneuver, always choosing the safe action.

Chapter 10

Conclusion

The implementation of scenario-based model predictive control for collision avoidance on the ReVolt model-scale ship have been improved significantly in this thesis. By adding a COLREGS transitional cost and a more accurate ship model, the previously experienced oscillatory behavior is eliminated. Minor occurrences of indecisive behaviour are still present in the more complex cases, but these are caused by the full guidance behaviour and desired waypoints being unknown to the prediction model. An anti-grounding system and utilization of AIS data have been added, contributing to a more complete collision avoidance system.

Considerable amounts of simulations and experiments were conducted in this thesis, providing a good understanding of how ReVolt behave during different collision avoidance scenarios. Results from simulations laid a good foundation for real life experiments, which furthermore exhibited great potential. The main performance issue experienced was the decreased CPA, which is a consequence of the largely reduced maneuverability of ReVolt while impacted by measurement noise and environmental forces. Introducing a disturbance model during prediction will further improve ReVolt's performance, pushing it one step closer to being a fully autonomous short sea shipping vessel.

Bibliography

[1] Autosea. https://www.ntnu.edu/autosea/about. Accessed: 2019-05-21.

[2] Hva er en ais? https://www.pronav.no/6419-2/. Accessed: 2019-01-31.

[3] The norwegian shipping industry. https://www.lifeinnorway.net/shipping-industry/. Accessed: 2019-05-04.

[4] Openstreetmap. https://www.openstreetmap.org/search?query=trondheim#map=14/63.4408/10.4034. Accessed: 2019-05-19.

[5] AIVDM/AIVDO protocol decoding. http://catb.org/gpsd/AIVDM.html#_types_1_2_and_3_position_report_class_a. Accessed: 2019-05-04.

[6] Sjøkart - elektroniske (ENC). https://kartkatalog.geonorge.no/metadata/kartverket/sjokart-elektroniske-enc/d030dffb-82c4-4eaa-a7c7-0e8cbbb515c5. Accessed: 2019-05-03.

[7] Shipping and world trade. http://www.ics-shipping.org/shipping-facts/shipping-and-world-trade. Accessed: 2019-05-04.

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

[9] COLREG - Convention on the International Regulations for Preventing Collision at Sea, Supplement. http://www.imo.org/en/Publications/Documents/Supplements%20and%20CDs/English/QB904E_012016.pdf, 2016. Accessed: 2019-06-04.

[10] H. Alfheim and K. Muggerud. Development of a dynamic positioning system for the revolt model ship. Master's thesis, NTNU, 2017.

[11] 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.

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

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

[14] G. Chaloulos, P. Hokayem, and J. Lygeros. Distributed hierarchical mpc for conflict resolution in air traffic control. In *Proceedings of the 2010 American Control Conference*, pages 3945–3950. IEEE, 2010.

[15] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang. Collision-free uav formation flight control based on nonlinear mpc. In *2011 international conference on electronics, communications and control (ICECC)*, pages 1951–1956. IEEE, 2011.

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

[17] T. Coldwell. Marine traffic behaviour in restricted waters. *The Journal of Navigation*, 36(3):430–444, 1983.

[18] A. Danielsen-Haces. Digital twin development - condition monitoring and simulation comparison for the revolt autonomous model ship. Master's thesis, NTNU, 2018.

[19] P. Davis, M. Dove, and C. Stockel. A computer simulation of marine traffic using domains and arenas. *The journal of Navigation*, 33(2):215–222, 1980.

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

[21] 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 *2016 IEEE Conference on Control Applications (CCA)*, pages 499–506. IEEE, 2016.

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

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

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

[25] H. Fukushima, K. Kon, and F. Matsuno. Distributed model predictive control for multi-vehicle formation with collision avoidance constraints. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 5480–5485. IEEE, 2005.

[26] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes. Collision avoidance and stabilization for autonomous vehicles in emergency scenarios. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216, 2017.

[27] E. M. Goodwin. A statistical study of ship domains. *The Journal of navigation*, 28(3):328–344, 1975.

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

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

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

[31] E. S. Henriksen. Automatic testing of maritime collision avoidance methods with sensor fusion. Master's thesis, NTNU, 2018.

[32] F. Hub. Modes of transportation explained: Which type of cargo and freight transportation is the best? `https://freighthub.com/en/blog/modes-transportation-explained-best/`. Accessed: 2019-05-11.

[33] IHO. S-57 Appendix A - IHO Object Catalogue. Available at `http://iho.int/iho_pubs/standard/S-57Ed3.1/31ApAch1.pdf`.

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

[35] 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.

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

[37] D. Kufoalor, E. Wilthil, I. Hagen, E. Brekke, and T. Johansen. Autonomous colregs-compliant decision making using maritime radar tracking and model predictive control.

[38] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. Safe maritime autonomous navigation with colregs, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, 39(1):110–119, 2013.

[39] Ø. A. G. Loe. Collision avoidance for unmanned surface vehicles. Master's thesis, NTNU, 2008.

[40] T. Midjås. Collision avoidance for the revolt model-scale ship, 2018. Project thesis, NTNU.

[41] T. Miloh and S. D. Sharma. Maritime collision avoidance as a differential game. Technical report, 1975.

[42] P. K. E. Minne. Automatic testing of maritime collision avoidance algorithms. Master's thesis, NTNU, 2017.

[43] W. Naeem, G. W. Irwin, and A. Yang. Colregs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics*, 22(6):669–678, 2012.

[44] O. S. Otterholm. Extracting mapped hazards from electronic navigational charts for asv collision avoidance. Master's thesis, NTNU, 2019.

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

[46] L. Whitney - TIME. Are computers already smarter than humans? 2017. Available at `http://time.com/4960778/computers-smarter-than-humans/`.

[47] S. Ø. Flæten - Teknisk 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.

[48] Rolls-Royce. Ship intelligence - autonomous ships the next step. `https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/rr-ship-intel-aawa-8pg.pdf`. Accessed: 2019-05-11.

[49] SAFETY4SEA. Human error the cause for most coastal vessels accidents in harbours. `https://safety4sea.com/human-error-the-cause-for-most-coastal-vessels-accidents-in-harbours/`. Accessed: 2019-05-21.

[50] 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.

[51] F. E. Sancar, B. Fidan, J. P. Huissoon, and S. L. Waslander. Mpc based collaborative adaptive cruise control with rear end collision avoidance. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 516–521. IEEE, 2014.

[52] 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.

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

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

[55] 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.

Tonje Midjås

NTNU
Norwegian University of
Science and Technology

DNV·GL