NTNU
Norwegian University of
Science and Technology

Collision Avoidance for ASVs Using Model Predictive Control

Inger Berge Hagen

Master of Science in Cybernetics and Robotics
Submission date: February 2017
Supervisor: Tor Arne Johansen, ITK
Co-supervisor: Giorgio D. Kwame Minde Kufolaor, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Background:

Model Predictive Control (MPC) is an advanced control technique which obtains desired control actions based on a cost function definition, predicted behavior of controlled variables, and a prescribed set of constraints. The desired control action is obtained through an optimization procedure over a finite prediction horizon. Predictions and optimization are key ingredients in the design of reliable and efficient collision avoidance control schemes, which typically include both physical constraints, e.g. input bounds and operational limits, and logic constraints, e.g. navigation rules.

Master thesis objectives and proposed tasks:

The objective of the thesis is to study, implement, and test a simulation-based MPC algorithm for ASV collision avoidance, considering COLREGS (see e.g. [8]). The study can be divided into a theoretical part, which entails the analysis and synthesis of the collision avoidance scheme, and a simulation part, which considers different collision scenarios.

The study should investigate how a large number of scenarios (or control behaviors) can be defined in order to improve performance and robustness of the collision avoidance scheme. Some scenarios should consider the case where the MPC reference changes within the prediction horizon and also when other vessels change their path within the horizon.

Investigation should be made on how the MPC's cost function and constraints should be weighted and tuned in order to meet the goals of collision avoidance according to COLREGS. The MPC scheme should be tested using simulations on a variety of collision scenarios, and the extent to which COLREGS are satisfied should be evaluated.

As part of the simulation study, a comparison can be made with a Velocity Obstacles approach. The comparison study should reveal the pros and cons of both the simulation-based MPC and VO method for collision avoidance.

The implementation part of the thesis aims at achieving an efficient C++ implementation of the MPC based collision avoidance scheme within the ROS framework. It is a goal to test an implementation of the collision avoidance MPC scheme on a USV in collaboration with Maritime Robotics. The test setup for the thesis assumes that a target/obstacle tracking system is available. The test should consider a selected set of collision scenarios based on the simulation study results.

Autosea:

The candidate will be associated with the AUTOSEA project, which is a collaborative research project between NTNU, DNV GL, Kongsberg Maritime and Maritime Robotics, focused on achieving world-leading competence and knowledge in the design and verification of methods and systems for sensor fusion and collision avoidance for ASVs. The project has access to supervision and physical test platforms through our industry partners.

Preface

This document is a Master's thesis written at NTNU as a part of the study program Cybernetics and Robotics, the work on this thesis was carried out during the autumn semester of 2016. The thesis is associated with the AUTOSEA project which is a collaborative research project between NTNU, DNV GL, Kongsberg Maritime and Maritime Robotics. The AUTOSEA project focuses on achieving world-leading competence and knowledge in the design and verification of methods for sensor fusion and collision avoidance for ASV's. The project has access to supervision and physical test platforms through its industrial partners.

I would like to express my gratitude to some of the people that have helped me realize this thesis. First of all, my supervisor Tor Arne Johansen and especially my co-supervisor D. Kwame Minde Kulfolaor (Giorgio). I would also like to thank Paal Kristian and Erik, with whom I've shared an office with the past semester, for the valuable discussions we have had and a for a very good work environment. Finally I would like to thank Thomas Ingebretsen, Arild Hepsø and Keenan Trnka at Maritime Robotics for their help with the full-scale testing. I would not have come this far on my own.

Inger Berge Hagen

Trondheim, 2017-01-17

Abstract

If an Autonomous Surface Vehicle (ASV) is to be operated at sea in the proximity of other vessels it must be equipped with a Collision Avoidance System (CAS). This system must comply with the rules for avoiding collision that govern the seas, the International Regulations for Preventing Collisions at Sea (COLREGS).

In this thesis a COLREGS compliant CAS using Simulation-Based Model Predictive Control (SB-MPC) has been implemented. The four main scenarios used to test the method's behavior and COLREGS compliance is: head-on, overtaking, and crossing from port and starboard. Several more complicated scenarios are also studied to see how the method deals with more complicated situations. The tests were performed within a Robotic Operating System (ROS) framework using a non-linear ship model to predict the ASV's movements. The tests were then repeated using a linear model. The behavior of the system is discussed and compared with the behavior of the ASV equipped with a CAS based on the Velocity Obstacle (VO) algorithm.

After the simulation study was completed the CAS was also tested on the On-board System Simulator (OBS) of Maritime Robotics to prepare for full-scale experiments. These simulations were using live Automatic Identification System (AIS)-data from ships in the Trondheimsfjorden.

The CAS performed well during the simulations, both within the ROS framework and in the OBS environment. However, finer tuning is still needed to get optimal performance.

Full-scale testing was also attempted, but not completed, mostly due to the lack of suitable obstacle vessels.

Sammendrag

Hvis autonome fartøy skal kunne operere til sjøs og i nærheten av andre fartøy er et fungerende kollisjonsunngåelsessystem uunnværlig. Dette systemet må overholde trafikkreglene som gjelder på sjøen slik de er fastsatt i de internasjonale reglene for kollisjonsunngåelse til sjøs (International Regulations for Preventing Collisions at Sea (COLREGS)).

I denne avhandlingen har det blitt utviklet et kollisjonsunngåelsessystem som overholder COLREGS-reglene og som er baset på metoden «Simulation-Based Model Predictive Control (SB-MPC)». Fire hovedscenarier har blitt brukt for å teste systemets generelle oppførsel og dets evne til å følge COLREGS reglene, de er som følger: front-mot-front, forbikjøring og kryssning fra styrbord og babord. Flere, mer kompliserte scenarier ble også brukt for å teste algortimens evne til å takle mer komplekse situasjoner. Metoden ble implementert innen et «Robotic Operating System (ROS)»-rammeverk og gjør bruk av en ulinær skipsmodell for å forutsi det autonome fartøyets bevegelser. Testene ble også repetert med en linær modell. Systemets oppførsel blir diskutert og sammenlignet med oppførselen til et autonomt fartøy utstyrt med et kollisjonsunngåelsessystem som baserer seg på algoritmen «Velocity Obstacle (VO)».

Etter at simulasjonsstudien var fullført ble systemet også testet på en «On-board System Simulator (OBS)» tilhørende Marine Robotics som en forberedelse til full-skala testing. OBS-simulasjonene ble gjennomført ved bruk av «Automatic Identification System (AIS)»-data mottat directe fra ship i Trondheimsfjorden.

Kollisjonsunngåelsessystemet oppførte seg bra under simuleringene, både innen ROS-rammeverket og innen OBS miljøet. Men det er fremdeles rom for forbedring i rguleringen av simulajonsparametrene.

Full-skala tester ble forsøkt gjennomført, men ble ikke fullført, mye på grunn av vansker med å finne egnede fartøy å teste metoden mot.

Contents

List of Figures

List of Tables

Acronyms

AIS Automatic Identification System.
ASV Autonomous Surface Vehicle.

CAS Collision Avoidance System.
COLREGS International Regulations for Preventing Collisions at Sea.

DOF Degreees of Freedom.
DW Dynamic Window.

IMO International Maritime Organization.

LOS Line-of-Sight.

MPC Model Predictive Control.

NED North, East, Down.

OBS On-board System Simulator.

RBB Rigid Bouyancy Boat.
ROS Robotic Operating System.
RTTs Rapidly-exploring Random Trees.

SB-MPC Simulation-Based Model Predictive Control.
SNAME Society of Naval Architects and Marine Engineers.

USV Unmanned Surface Vehicles.

VO Velocity Obstacle.

Nomenclature

χ	Course
η	Position and orientation vector
$\mathbf{C}(\nu)$	Coriolis and centripetal matrix
$\mathbf{D}(\nu)$	Damping matrix
$\mathbf{M}$	Mass matrix
ν	Body-fixed linear and angular velocity vector
ψ	Heading (yaw)
τ	Force vector
x	position

Chapter 1

Introduction

1.1 Motivation

The automation of tasks previously performed by humans has been a crucial part in the evolution of modern society. From water mills exploiting the power of rivers and streams to for instance grind grains to the invention of the flying shuttle and the spinning jenny that started the industrial revolution in England, automation has allowed for the execution of tasks that earlier was limited by the physical power and skill of humans or animals.

While the motivation in the early days was to save time and ease the work burden of the people, it soon focused more on the speed of execution, cost-effectiveness and quality. Nowadays there is also an ongoing effort to conceive advanced machines that can perform tasks in environments hazardous to human beings, an example is the use of snake robots in search and rescue operations[3]. In other areas such as the development of high performance fighter-planes, it is the limits of the human physique that put bounds on the performance of the system.

While humans are both adaptable and have the ability to apply judgment, they can also be inconsistent and subject to errors[6]. Machines on the other hand are consistent and predictable, but are not able to apply judgment and it is not possible to program them for all eventualities. However, the potential savings makes it interesting to look into the automation of surface vehicles. The crucial point is that the automated system must be at least as safe as a manually controlled vessel.

An Autonomous Surface Vehicle (ASV) will have to be able to navigate in waters where other vessels are present. For automation to be a viable option a reliable and predictable Collision Avoidance System (CAS) is therefore necessary.

A project that literally brings these concepts close to home is the NTNU project that aims to have an autonomous ferry running between Ravnkloa and Vestre Kanalhavn ready for full-scale testing in 2018/2019 [19]. The central location of this ferry route, in the middle of Trondheim city, has attracted attention even outside the academic community and shows that autonomous vehicles is the future also at sea.

Also of interest, although on a completely different scale, is the MUNIN project [14]. This is a collaboration project co-founded by the European Commission and aims to develop and verify a concept for an autonomous ship. Their motivation is the increased need for maritime transport along with the decreasing interest among people to work at sea [15]. In addition to overcoming the purely technical challenges, this project also looks to integrate autonomous vessels into the governing regulations. If this succeeds, it will be a major step towards autonomous shipping.

1.2 Literature Review

In the review article [17] the development of collision avoidance algorithms is discussed and most of the published works on the subject from the previous two decades have been categorized. This is extended in [20] where several of the publications deemed significant to the subject's development are discussed further. The reviews include methods with and without International Regulations for Preventing Collisions at Sea (COLREGS) compliance and provide a comprehensive historical background on the subject of collision avoidance and path planning methods.

Path planning algorithms can be classified into static and dynamic types named after the type of obstacles they are supposed to avoid [16]. They can also be categorized as local or global. Methods that use previous knowledge of its environment to plan a path from start to finish are called global algorithms, this is also known as path planning. These methods can only deal with static obstacles. On the other hand, local methods have the ability to produce new paths as their environment changes. These methods can thus be used for collision avoidance. A third approach is to combine a local and a global method to try and achieve a system that can guarantee that it gets from A to B, but also have the ability to avoid dynamic obstacles as they appear.

In [1] a review is given on the development needs of unmanned marine vehicles in order to achieve an increased level of autonomy and coherence with the COLREGS. The article's main focus is not on collision avoidance, an Artificial Potential Field approach is the only local method considered([9], [11]). A summary of the main COLREGS rules is given and how to implement these rules for multiple cooperating Unmanned Surface Vehicles (USV)s is also discussed. The paper does however give a good overview of the challenges to consider when developing a USV.

In [13] a hybrid approach is proposed. The method uses the A* algorithm as a guide for the Rapidly-exploring Random Trees (RTTs) ([10]) which provides global path planning while a Dynamic Window (DW) ([5]) takes care of local collision avoidance. The original DW method predicts vehicle movements along constant-radius arcs with constant velocities and includes some vehicle dynamics which prevents the demand for impossible control input. In [13] this is extended with lateral dynamics (sway-motion) and acceleration constraints to better comply with the behavior of a sea faring vessel. Modifications were also proposed to achieve COLREGS adherence. The system showed good performance both in the simulations and in the full-scale experiments, but because of technical problems the testing was not completed.

Another local method is implemented by [18], the Velocity Obstacle (VO) algorithm. The approach is to check all possible velocities, excluding the set, called the VO, that will lead to collision with another vessel. This is a very simple method which does not require any information about the vessel's dynamics. The method performed in accordance with the COLREGS during the simulated tests, but full-scale testing was completed.

In [8] it is argued that the methods currently available generally do not scale very well to handle dense traffic and multiple highly dynamic obstacles while considering the dynamics of the ship and environmental disturbances. The authors continue to point out that the incorporation of such complex situations into the existing algorithms is not likely to be uncomplicated.As a possible solution the article proposes a receding horizon Model Predictive Control (MPC) where a dynamic ship model including environmental disturbances is used to predict the ship's trajectory for a finite set of control behaviors. The trajectories are then evaluated by a cost function that includes risk, hazard, operational constraints and objectives. Re-optimization is then performed at regular intervals as new information becomes available. The paper continues with a system overview where the overall concept and information flow of the system along with the proposed architecture of the CAS. Details are then given on the implementation and formulation of the COLREGS and hazard evaluation criterion. The simulation study that follows show that the method complies with the main rules of COLREGS and can handle situations with multiple obstacles, also when they exhibit random behavior. This article is the origin of this thesis and further details will be described in chapter 3.

1.3 Assumptions

Even though the final goal is a robust SB-MPC algorithm that can handle any situation, some limitations has to be set during the development phase. The work presented in this thesis therefore assumes the following:

- All other vessels are power-driven, i.e. situations where less maneuverable vessels are encountered have not been studied.
- The necessary sensor data about other vessels is always available.
- The necessary sensor data about own vessel is always available.
- There are no disturbances to the measurements.
- Only COLREGS apply, i.e. no location specific rules or traffic separation schemes have been taken into account in the design of the CAS.
- The potential collisions occurs in open sea.

1.4 Contribution

The main contributions of this thesis are:

- The implementation as a stand-alone C++ library of a COLREGS compliant CAS found in the simulation-based MPC algorithm described in [8].
- A comparison between the use of a linear and a non-linear ship model in the SB-MPC.
- A comparison between the SB-MPC method and the VO algorithm described in [18].
- Results from full-scale simulations based on live Automatic Identification System (AIS) data.
- Results from collision avoidance experiments in Trondheimsfjorden.

1.5 Thesis Structure

Chapter 2 provides necessary background information on the subjects of vessel modeling, MPC and the COLREGS. In chapter 3 the SB-MPC algorithm and its implementation is explained. Chapter 4 presents the results of the Robotic Operating System (ROS) simulations along with discussions of the system's behavior in the different scenarios. Results from the On-board System Simulator (OBS) simulations and the full scale then follow in chapter 5. A discussion of the findings along with strengths and weaknesses of the method is given in chapter 6 followed by suggestions for future work in chapter 7. Finally, conclusions are drawn in chapter 8.

The appendix contain two short chapters. First a source code example, then further details on the implementation.

Chapter 2

Theoretical Background

2.1 Surface Vessel Modeling

A part of the motivation for this thesis was to study the difference between a CAS using a SB-MPC and one using VO. In 2015 Thomas Stenersen implemented a VO based CAS along with a simulator within a ROS framework [18]. It was therefor natural to reuse this simulator in this work. The ship-model is based on the *Viknes 830*, as described in [13] and [18]. This is also the model used for ASV trajectory prediction in the SB-MPC and is described in the following section.

The Equations of Motions

The model used in the SB-MPC is a 3-Degreees of Freedom (DOF) model of a surface-vessel where six states describes the motion of the vessel. The notation used is the Society of Naval Architects and Marine Engineers (SNAME) notation as shown in table 2.1. The general motion of the vessel is described by the vectors $\eta = [x\ y\ \psi]^\top$ giving the generalized coordinates of the vessel, $\tau = [X\ Y\ N]^\top$ giving the forces working on the vessel, both in the North, East, Down (NED) frame, and $\nu = [u\ v\ r]^\top$ giving the velocities of the vessel in the body-fixed frame.

Table 2.1: The SNAME notation (1950) for marine vessels. Only 3-DOF variables are shown.

DOF	Forces and moments	Linear and angular velocities	Positions and Euler angles
Motions in the x-direction (surge)	X	u	x
Motions in the y-direction (sway)	Y	v	y
Motions about the z-axis (yaw)	N	r	ψ

The equations of motion are given as [4]:

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

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

For the SB-MPC it is preferable to keep the model as simple as possible, the center of gravity is therefore assumed to coincide with the body-origin of the vessel. To further simplify the model the environmental disturbances, $\boldsymbol{w}$, has also been left out. This will influence the accuracy of the predicted trajectories when the physical vessel's path is affected by for instance current or waves. For the simulation study this will not cause any problems, but it should be taken into account when deciding the safety margins in the controller.

In (2.1a) the $\mathbf{R}(\psi)$ variable is a 3×3 rotation matrix that transforms the body-fixed velocities into the world-fixed frame:

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

In (2.1b) the variables $\mathbf{M}$, $\mathbf{C}(\boldsymbol{\nu})$ and $\mathbf{D}(\boldsymbol{\nu})$ are also 3×3 matrices. The $\mathbf{M}$ is the inertia matrix counting for the rigid-body mass of the vessel and also the added mass of the vessel: $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$. The added mass can be seen as a virtual mass added to the system as the acceleration or deceleration of the vessel must also displace some of the surrounding fluid.

The rigid-body inertia matrix of the vessel is given as:

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

where m is the mass of the vessel, and I_z is the moment of inertia about the z-axis of the vessel. To keep the model simple the added mass was set to zero. Thus giving:

$$\mathbf{M_A} = \mathbf{0} \tag{2.4}$$

$\mathbf{C}(v)$ is the coriolis and centripetal matrix which also has a term for added coriolis: $\mathbf{C}(\boldsymbol{v}) = \mathbf{C}(\boldsymbol{v})_{RB} + \mathbf{C}(\boldsymbol{v})_A$. As the added mass has been set to zero the added coriolis $\mathbf{C}(\boldsymbol{v})$ will also be zero. The rigid-body coriolis and centripetal matrix is given as:

$$\mathbf{C}(\boldsymbol{v})_{RB} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix}. \tag{2.5}$$

The $\mathbf{D}(\boldsymbol{v})$ matrix represents the damping effects in the system. This can be modeled in many different ways, the simplified version chosen for this thesis is defined as: $\mathbf{D}(\boldsymbol{v})\boldsymbol{v} = \mathbf{D}_L\boldsymbol{v} + \mathbf{D}_{NL}(\boldsymbol{v})\boldsymbol{v}$, where $\mathbf{D}_L$ represents the linear damping defined as:

$$\mathbf{D}_L = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}. \tag{2.6}$$

The nonlinear effects are represented by $\mathbf{D}_{NL}(v)$, which is given by:

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

The forces acting on the vessel are found on the left side of (2.1b). The $\boldsymbol{w}$ represents the environmental forces in the equation, but as the goal is to keep the model as simple as possible this will be set to zero for now. $\boldsymbol{\tau}$ is the generalized force vector containing the resultant forces of the vessel's actuators.

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ l_r F_y \end{bmatrix}, \tag{2.8}$$

where τ_X and τ_Y represents the forces along the x and y axes of the vessel, and τ_N the moment around the z-axis. F_x and F_y are the forces produced in x and y-direction respectively and l_r is the moment arm.

Controller

There are two controllers used in the SB-MPC, a feedback-linearizing controller for speed and a conventional PD-controller for heading. The speed controller is given by

$$F_x = (-mv + Y_{\dot{v}}v + Y_{\dot{r}}r)r - (X_u + X_{|u|u}|u| + X_{uuu}u^2)u + K_{p,u}m(u_d - u), \tag{2.9}$$

and the yaw rate controller

$$F_y = \frac{K_{p,\psi}}{l_r}((\psi_d - \psi) - K_{d,\psi}r). \tag{2.10}$$

Table 2.2: Ship model parameters

(a) Vessel parameters

Parameter	Value	unit
m	3980.0	kg
I_z	19703.0	kg/m^2
X_u	-50.0	kg/s^2
$X_{\|u\|u}$	-135.0	$\text{kg}/\text{m}^\text{s}$
X_{uuu}	0.0	$\text{kg}/(\text{m}\cdot\text{s})^2$
Y_v	-200.0	kg/m^2
$Y_{\|v\|v}$	-2000.0	kg/s^2
Y_{vvv}	0.0	$\text{kg}/(\text{m}\cdot\text{s})^2$
N_r	-3224.0	$\text{kg}\cdot\text{m}^2/2$
$N_{\|r\|r}$	0.0	$\text{kg}\cdot,^2$
N_{rrr}	-3224.0	$\text{kg}\cdot\text{m}^2\text{s}$
N_v	0.0	$\text{kg}\cdot\text{m}/\text{s}$
Y_r	0.0	$\text{kg}\cdot\text{m}/\text{s}$
$F_{x,max}$	13100.0	N
$F_{x,min}$	-6550.0	N
$F_{y,max}$	645.0	N
$F_{y,min}$	-645.0	N
l_r	4.0	m

(b) Controller parameters

Parameter	Value	unit
$K_{p,u}$	0.1	1/s
$K_{p,\psi}$	5.0	1/s
$K_{d,\psi}$	1.0	s

2.2 Model Predictive Control

MPC bases itself on iterative, finite horizon optimization using a plant model. A cost function is used to evaluate the system's behavior according to the constraints and objectives of the plant. At the current time t the system is sampled and a cost-minimizing control strategy for the prediction is computed for the interval $[t, t+T]$, where T is called the prediction horizon of the system. When the first action of the control strategy has been applied the process is repeated.

MPC can be used to compute an optimal trajectory for a vessel based on predictions of obstacle movements. In addition to operational constraints and objectives the uncertainties with regard to the obstacles' motions and environmental disturbances can be taken into account along with the risk and hazard involved for each trajectory.

Numerical methods are often used to solve the optimization problem, but the complex situations that can occur in a collision avoidance setting can cause problems related to convergence and the computation time required, it is therefore not well adapted for implementations in real-time systems such as a CAS. These problems can

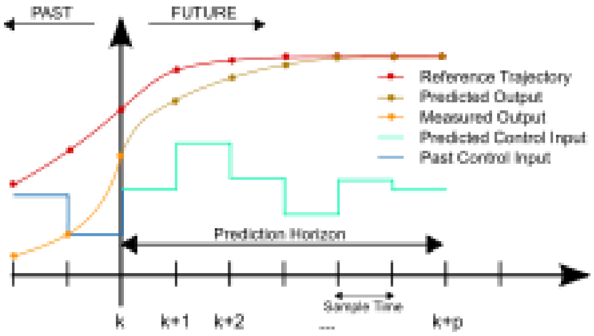

Figure 2.1: Discrete MPC [22]

however be avoided by a using a reduced set of possible control behaviors in a scenario based MPC framework. A scenario or simulation based MPC uses simulation to predict a plant's output for a given control input. Simulations are run for each possible control input and each scenario is evaluated and assigned a cost by a cost function. The control behavior that induces the lowest cost can then be applied to the system.

MPC is a very powerful method and a simulation based version sidesteps the uncertainties related to computation time associated with numerical optimization. This rater simple approach is consequently the most appropriate for this implementation.

2.3 COLREGS

The history of the COLREGS goes back to 1840 [21] when the London Trinity House drew up a set of regulations with the purpose of preventing collisions between marine vessels. Since then the regulations has developed with the increased activity at sea and advances in technology, notably the invention of the Radar. The rules governs conduct, but also regulates the use of light, shapes and sound signals and technical details related to the former. The current COLREGS was established by the International Maritime Organization (IMO) in 1972 and came into force on the 15th of July 1977 [12] and is now an internationally recognized collision avoidance scheme [7].

The COLREGS were constructed with human operators in mind and can be open to interpretation. This becomes evident when implementing a COLREGS compliant control algorithm and the necessity of deciding such critical values as "safe distance", "early action", etc. appears. However, following the COLREGS is important for an ASV as this will produce predictable behaviors when encountering other vessels.

The next section contains an overview over the requirements from COLREGS that are most relevant to this application.

2.3.1 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.

2.3.2 Rule 8 - Actions to avoid collision

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

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

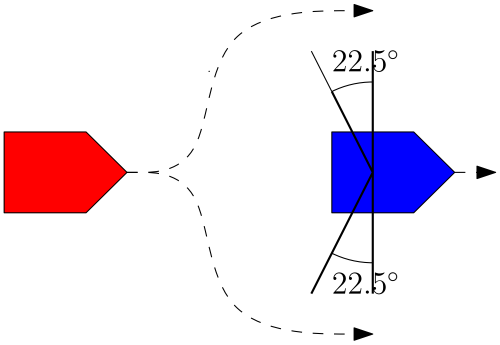

Figure 2.2: The overtaking vessel can pass either starboard or port to the vessel being overtaken.

2.3.3 Rule 13 - Overtaking

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

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

2.3.4 Rule 14 - Head-on situation

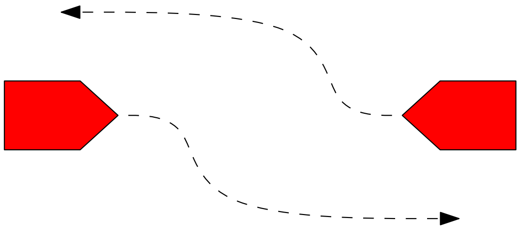

Figure 2.3: Correct collision avoidance behavior in head-on situation.

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

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

2.3.5 Rule 15 - Crossing situation

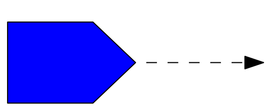

Figure 2.4: Crossing situation showing give-way vessel in red and stand-on vessel in blue.

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

2.3.6 Rule 16 - Actions by give-way vessel

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

2.3.7 Rule 17 - Actions by stand-on vessel

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

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

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

2.3.8 Other considerations

In addition to the rules cited above the convention also regulates the responsibilities between vessels, in general a power-driven vessel shall keep out of the way of less maneuverable vessels. There are also special regulations for narrow channels and low-visibility Area-specific rules also exist, for instance in harbors where the local rules set by the harbor authority have precedence over the COLREGS. In this thesis only situations with power-driven vessels where COLREGS apply are considered.

The underlying rule that applies to all situations is *Rule 8 - Actions to avoid collision*. Taking early and substantial action to keep well clear of other vessels is a base criterion for any CAS. With this in mind, the most important scenarios to consider are the ones where *Rule 14 - Head-on* and *Rule 15 - Crossing* apply. In these cases the COLREGS prescribe a clear course of action and is thus easy to test. The scenario described by *Rule 13 - Overtaking* does not demand any actions not already covered by *Rule 8* and will therefore not be studied further.

Chapter 3

Implementation

The CAS implemented in this thesis is based on the concept described in the article *Ship Collision Avoidance using Simulation-Based Control Behavior Selection with Predictive Hazard Assessment* [8]. The following sections describe this method. Where significant changes have been made to the original method this will be mentioned. The last part of the chapter is dedicated to an overview of the `C++` implementation.

3.1 System Overview

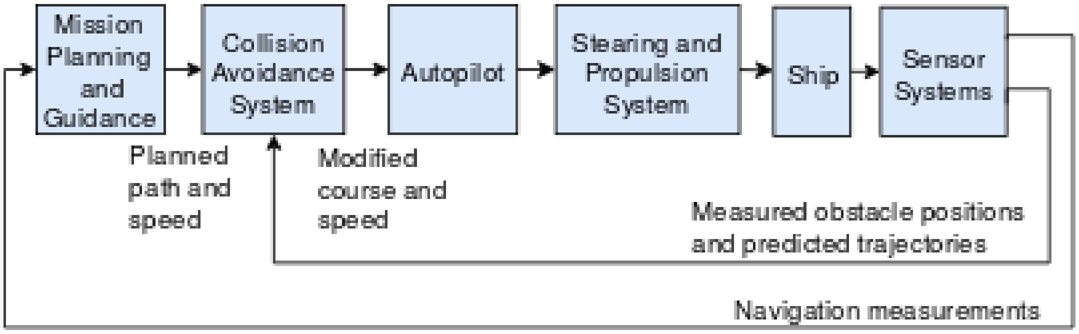

Figure 3.1: System overview and information flow between modules.

Figure 3.1 gives an overview over the main modules of a navigation system and the information flow between them. The Mission Planner produces a desired course and speed based on a set of way-points. The Sensor Systems provides navigational measurements for the ship and any present obstacles. Based on these measurements the collision avoidance module will make a prediction of the obstacles' trajectories and do a series of simulations of the ship's trajectory with a finite set of offsets to the nominal course and speed. The smallest offset-pair that produces a collision-free and COLREGS-compliant trajectory is then chosen and the new course and speed references are sent to the autopilot.

In [8] the Guidance module is incorporated in the Autopilot-block. In this case the desired path is given as a set of waypoints and the CAS must therefore also contain a Guidance module. A prediction of the input to the Autopilot can then be made and the offsets calculated. An advantage of this implementation is that it allows the detection of a missed waypoint. Compared to the SB-MPC the extra computational power required is insignificant, but knowledge of the Guidance algorithm in the Autopilot is required as discrepancies between the predicted and actual set-point can lead to undesired behavior. In projects where the system is developed as a whole this is not a problem. But it, as in the case of this thesis, the CAS is to be implemented in an already existing system getting access to the guidance algorithm can be difficult.

For this collision avoidance scheme to be applicable the following information must be available:

- List of obstacle's positions and velocities
- A desired path and speed
- A mathematical model of the ship
- The ship's own state

3.2 Collision Avoidance System

The main architecture of the CAS-module from [8] can be seen in figure 3.2. This structure have been somewhat simplified in this implementation in that the weather scenarios and sensor uncertainty has been removed. This implies that the information about the obstacles is assumed to be correct, that the weather conditions must be calm and that there is no current present.

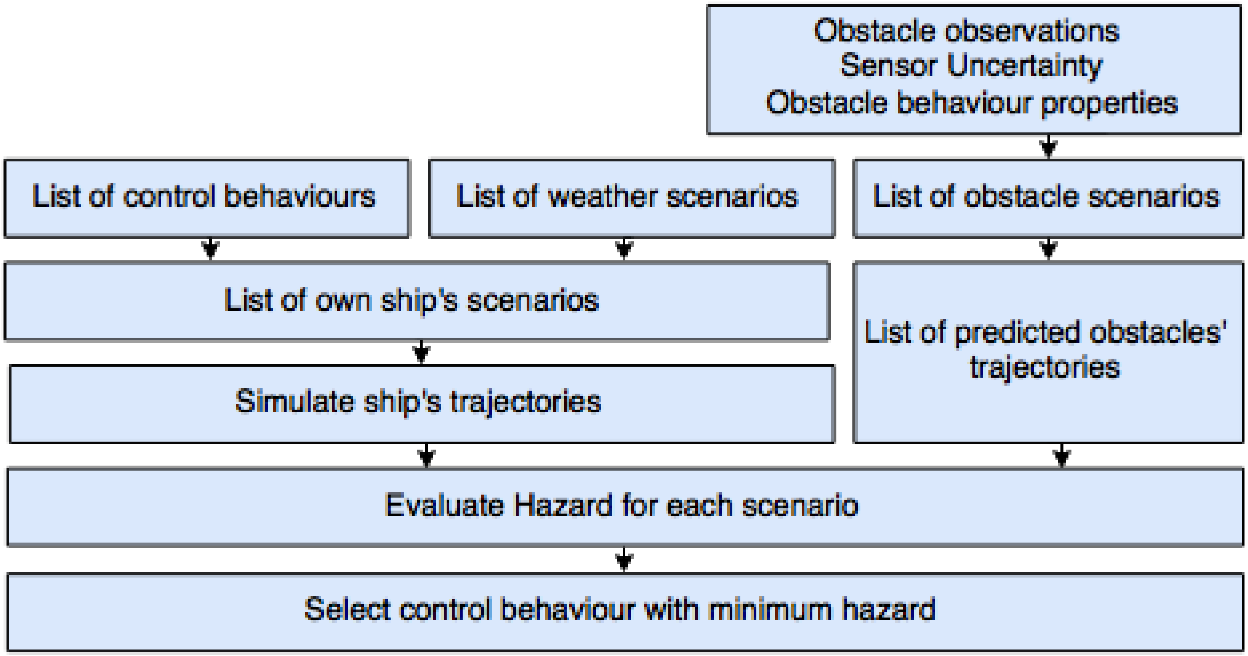

Figure 3.2: Collision avoidance algorithm

3.2.1 Obstacle Trajectory Prediction

There are considerable uncertainties with regard to the prediction of the future trajectories of the obstacles. But as the situation is re-evaluated with regular intervals using a straight-line prediction is maybe the easiest solution. The

$$\bar{x}_i(t) = \hat{x}_i + \hat{u}_i(t - \tau_i) \tag{3.1}$$

$$\bar{y}_i(t) = \hat{y}_i + \hat{v}_i(t - \tau_i) \tag{3.2}$$

Where t is a future point in time and τ_i is the time of the last observation.

3.2.2 Control Behaviors

The CAS simulates the ship's trajectory for a finite set of control behaviors. These are then compared to the predicted trajectories of any present obstacles and the collision hazard for each trajectory is calculated. The trajectories that do not comply with the COLREGS will also be assigned a penalty cost. The control behavior that result in the lowest cost is then chosen.

As a minimum set of alternative control behaviors [8] recommends:

- Course offset [degrees]: -90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90.
- Speed factor: [1.0, 0.5, 0.0, -1.0], which is equal to [nominal propulsion, slow forward, stop, full reverse]

3.2.3 ASV Trajectory Prediction

Two methods have been looked at for the prediction of the ASV's motion. The starting point was a 3-DOF as described in section 2.1, later the possibility of using a simple straight-line trajectory prediction was added. As one of the advantages of the SB-MPC approach is the possibility of including vessel dynamics this might seem counter productive, but because of computational limitations in the test platform it was necessary to find a less complex method. As the proposed target for the tests is a relatively small vessel with fast dynamics, see section 5.2, the discrepancy between the actual and predicted trajectory relative to the safety margins is acceptable.

The 3-DOF model is quite simple and environmental forces have not been taken into account, but it does include the main dynamics of a ship. This is sufficient for our purposes, but if the SB-MPC is to be implemented on a larger vessel with slower dynamics a more accurate model is necessary.

3.2.4 COLREGS Compliance

For the ASV to behave in a that is predictable and logical to the operators of other vessels it must adhere to the "rules of the road" as laid out by the COLREGS. This section lays out the fashion in which COLREGS compliance is evaluated by the SB-MPC.

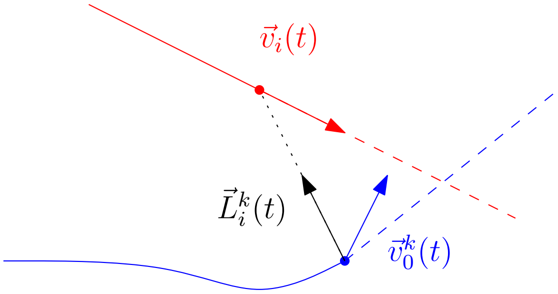

Figure 3.3: Vectors used for hazard evaluation in scenario k at a future time point t. The blue dot marks the ASV and the red dot marks the obstacle with index i, the vectors $\vec{v}_0^k(t)$ and $\vec{v}_i(t)$ represents their predicted velocities and the dashed lines signify the vessels' anticipated trajectories. The black vector marked $\vec{L}_i^k(t)$ is a unit vector in Line-of-Sight (LOS) direction from the ASV to obstacle i.

Table 3.1: Parameter description for evaluation of COLREGS compliance

Parameter	Description
$\vec{v}_0^k(t)$	Predicted velocity of the ASV at a future time instant t in scenario k.
$\vec{v}_i(t)$	Predicted velocity of obstacle with index i at a future time instant t in scenario k.
$d_{0,1}^k$	Predicted distance between the ASV and obstacle with index i at a future time instant t in scenario k.
d_i^{cl}	The distance within which the COLREGS is said to apply[1].

To evaluate the COLREGS compliance and collision hazard of a trajectory, position, velocity and LOS vectors are used, the main information used is shown in figure 3.3. Following is a description of how this evaluation is done.

- **CLOSE**: At a point in time t, the obstacle i is said to be close if $d_{0,i}^k \leq d_i^{cl}$.
- **OVERTAKEN**: At a point in time t, the ASV is said to be overtaken by the obstacle with index i if its is close, $|\vec{v}_i(t)| > |\vec{v}_0^k(t)|$ and
$$\vec{v}_0^k(t) \cdot \vec{v}_i(t) > \cos(\phi_{\text{ot}})|\vec{v}_0^k(t)||\vec{v}_i(t)| \tag{3.3}$$
- **STARBOARD**: At a point in time t, the obstacle i is said to be starboard of the ASV if the bearing angle of $\vec{L}_i^k(t)$ is larger than the heading angle of the ASV.

---

[1] This distance can be made to be specific to the obstacle with index i by making it dependent on the velocity of the ASV and the obstacle in question.

- **HEAD-ON**: At a point in time t, the obstacle i is said to be head-on if it is close and

$$\begin{aligned} |\vec{v}_i(t)| &> 0.05 \\ \vec{v}_0^k(t) \cdot \vec{v}_i(t) &< -\cos(\phi_{\text{ho}})|\vec{v}_0^k(t)||\vec{v}_i(t)| \\ \vec{v}_0^k(t) \cdot \vec{L}_i^k(t) &> \cos(\phi_{\text{ah}})|\vec{v}_0^k(t)| \end{aligned} \tag{3.4}$$

- **CROSSING**: At a point in time t, the obstacle i is said to be crossing if it is close and

$$\vec{v}_0^k(t) \cdot \vec{v}_i(t) < \cos(\phi_{\text{cr}})|\vec{v}_0^k(t)||\vec{v}_i(t)| \tag{3.5}$$

  It has been added that the obstacle must also be heading towards the ASV.

The angles used in the evaluation of the COLREGS situations can be found in table 3.2.

Table 3.2: COLREGS angles

Parameter	Value
ϕ_{ot}	68.5°
ϕ_{ho}	22.5°
ϕ_{ah}[2]	15.0°
ϕ_{cr}[3]	68.5°

AS in [8] a binary indicator $\mu_i^k(t) \in 0, 1$ is used to indicate a violation of the COLREGS rules 14 (head-on) or 15 (crossing),described in section 2.3. This can then be expressed as:

$$\begin{aligned} \mu_i^k(t) &= \text{RULE 14 or RULE 15} \\ \text{RULE 14} &= \text{CLOSE \& STARBOARD \& HEAD-ON} \\ \text{RULE 15} &= \text{CLOSE \& STARBOARD \& CROSSING \& NOT OVERTAKEN} \end{aligned}$$

While rule 13 (overtaking) is implicitly taken into account via rule 14, the compliance with the other rules mentioned in section 2.3 is dependent on the tuning of the parameters described in the following section.

---

[2] Can if necessary be replaced by a more suitable angle.

[3] See footnote 2

3.2.5 Hazard Evaluation Criterion

The risk of collision is formalized as

$$R_i^k(t) = \begin{cases} \dfrac{1}{|t - t_0|^p} \left(\dfrac{d^{safe}}{d_{0,i}^k} \right)^q & , \quad \text{if} \quad d_{0,i}^k(t) < d_i^{safe} \\ \qquad\quad 0 & , \quad \text{otherwise} \end{cases} \tag{3.6}$$

Here, t_0 is the current time and $t > t_0$ is the prediction time. The exponent $p \geq 1/2$ weighs the distance in time until the risk occurs. As the inverse of this property is used in the function 3.6, occurrences that are close in time will be prioritized over more distant events.

The exponent $q \geq 1$ weighs the relation between the distance between the ASV and the distance considered as safe, d^{safe}. These two parameters affect the system's ability to comply with COLREGS rule 16, i.e. the duty to take early and substantial action to keep well clear of other vessels, and must be chosen accordingly.

The cost of an actual collision is given by

$$C_i^k(t) = K_i^{\text{coll}}(t) \left| \vec{v}_0^k(t) - \vec{v}_i^k(t) \right|^2 . \tag{3.7}$$

where it can be seen that the cost of a collision is influenced by the kinetic energy involved. In this implementation, the factor $K_i^{\text{coll}}(t)$ only depends on the size of the obstacle, but other factors such as the type of vessel might also be included. If, in a scenario with multiple obstacles, a situation occurs where collision is unavoidable, these considerations might help limit the negative consequences.

As reevaluation of the ASV's trajectory will be performed with regular intervals it is a possibility that the offsets will change between each optimization. To avoid the succession of small changes to the offsets two penalty functions are applied.

$$\Delta_\chi = \begin{cases} K_{\Delta\chi,\text{port}} \left(\chi_{ca} - \chi_{ca-last}\right)^2 & , \quad \text{if turn to port} \\ K_{\Delta\chi,\text{starboard}} \left(\chi_{ca} - \chi_{ca-last}\right)^2 & , \quad \text{if turn to starboard} \end{cases} \tag{3.8}$$

$$\Delta_P = K_{\Delta P} |P_{ca} - P_{ca,last}| \tag{3.9}$$

These equations are not specified in [8], but it is recommended that they are positive at the origin and that Δ_χ gives a larger penalty on course changes to port than to starboard. The latter helps assure compliance with COLREGS rule 14 and 15.

The hazard related to behavior scenario k can then be calculated as

$$H^k(t_0) = \max_i \max_{t\in D(t_0)} \left(C_i^k(t)R_i(t) + \kappa_i \mu_i^k(t)\right) + \Delta_\chi(\chi_{ca}, \chi_{ca,last}) + \Delta_P(P_{ca}, P_{ca,last}) + K_P(1-P) + K_\chi \chi_{ca}^2 \tag{3.10}$$

where t_0 is the current moment in time and $D(t_0) = \{t_0, t_0 + T_s, ..., T_0 + T\}$ is the set of discrete sample times, T_s being the discretization interval and T the prediction horizon. The κ_i parameter represents the cost of violating the COLREGS rule 13, 14 and 15. Finally, the last to terms penalizes deviating from the planned path. This assures that no change is made to the control behavior is made unless it significantly reduces the predicted hazard.

In [8] an additional element $g(\cdot)$ is also included in the hazard evaluation. This term is added to prohibit control behaviors leading to grounding of the ASV.

3.2.6 Control Decision

The previous section outlined a way of evaluating the hazard involved with each control behavior. For each scenario $k \in \{1, 2, ..., N\}$ at time t_0 the optimal behavior is chosen by

$$k^*(t_0) = \arg\min_k H^k(t_0). \tag{3.11}$$

Every time the optimization procedure is called the hazard of each scenario is evaluated, the control behavior involving the least hazard is then chosen and the offsets are sent to the autopilot. As new sensor data becomes available the process is repeated, [8] suggests that the intervals between each iteration should be of 5 seconds.

There are many considerations to be done while tuning the parameters of this algorithm and the process is not straightforwards. Following is listed the main points that have been considered in this thesis.

- Avoid collision
- Adhere to the COLREGS
- Keep nominal speed
- Keep nominal course

3.3 C++ Implementation

The algorithm was first implemented as a separate node within the ROS framework of the simulator developed in [18]. It was later separated out as a stand-alone library to facilitate the testing of the method on other systems. The main components of the code are the same, but in this section the focus will be on the latter as this is the most versatile.

3.3.1 Structure

The code is divided into three classes: *sb_mpc*, *ship_model* and *obstacle*. This modular structure makes the code easier to understand and also facilitate changing the type of model used in the trajectory prediction.

The *sb_mpc* class implements the SB-MPC algorithm itself in the method *getBestControlOffset()*. An example of how this can be used is shown in Appendix A. It also contains set/get-methods for the tuning parameters.

The model used to predict the ASV's trajectory, as described in section 2.1, is defined in the *ship_model* class. The integrator used is a simple first order Euler. As problems with regard to the run-time appeared in connection with the experimental testing a linear prediction method was also added. Which of these methods that is used in the simulations can be chosen via the set/get-methods of the *sb_mpc* class.

The *obstacle* class was implemented to better organize the code. Each obstacle vessel is represented by an *obstacle* object which at initialization makes a straight-line prediction of the vessel's future movements withing the prediction horizon as described in section 3.2.1.

Optimization with regards to the run-time of the algorithm has not been a prioritized task in this implementation, but it has been kept in mind during the implementation. For instance, the *Eigen* library was chosen for matrix operations and is along with *Armadillo* one of the fastest libraries for this type of operations. Dynamic allocation of variables has also been deliberately avoided. What is clear is that the prediction of the ASV's own trajectory is the most time consuming task. The runtime also increases drastically with the number of obstacles to avoid.

3.3.2 ROS

The ROS framework provides a very powerful tool for simulated testing. The simulator used in the work was crated by [18] during the work with his master thesis and already contains an implementation of the VO algorithm. The nodal structure and the extensive documentation and tutorial base available makes adding new features easy. The ease with which one can swap between using different CAS nodes makes it very suitable for comparing the behavior of several methods.

But there are of course drawbacks. First of all it is not straightforwards to speed up the simulations. This becomes a challenge if the tests are to be performed with realistic parameters where the safety margins will extend the duration of the simulations significantly. Debugging was also a challenge. ROS provides a debug feature that prints to the terminal and a log file, but a method of stepping through the code and inspect variables was not found. It is however possible to debug each node separately in the editor *eclipse* with all the usual tools. In this case the connection with the rest of the system is lost and input from other nodes must be entered manually.

3.3.3 ROS to OBS

The nodal structure of the simulator encourages a clear separation between the communication between nodes and the procedures performed within them. This made separating out the CAS from its framework easy work.

The integration of the SB-MPC into the OBS did however require several adjustments. First of all, the concerns with regard to the run-time of the algorithm lead to the implementation of linear prediction for ASV trajectories. Secondly, in the interface between the OBS and the SB-MPC the following preprocessing of the input was added:

- Conversion of the measured positions from latitude and longitude into a xy-coordinate system with the unit of meters and the ASV positioned in the origin. In the ROS simulator used in the development the unit used for position is meters, but this information is usually given in latitude and longitude and should ideally also be used in the SB-MPC.
- Estimation of the current position of obstacles using linear prediction. The obstacle vessels in the OBS are based on AIS-data transmitted by vessels in the Trondheimsfjorden. These transmissions are made at irregular intervals and position estimation between the updates is therefore necessary. This can also be included in the SB-MPC.
- Reduction of the number of obstacles that are considered in the SB-MPC by only including vessels that are within a certain distance from the ASV. The incoming AIS-data covers a very large area. To avoid unnecessary calculations vessels that are too far away to be considered a potential hazard should be left out.

The interface between the SB-MPC and OBS was implemented by Thomas Ingebretsen at Marine Robotics, much because of the limited time available, but should finally be incorporated into the SB-MPC.

Chapter 4

Simulation Study

The simulations have been run with the parameters shown in table 4.1. When using simple linear prediction the prediction horizon T and the step length dt has been increased. It should also be noted that because the seeming instant efficiency of a speed reduction when using linear prediction the cost of speed deviation K_P and speed change K_{dP} had to be adjusted to avoid excessive speed changes. In both cases the possible control offsets are:

- Course offset [degrees]: -90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90.
- Speed factor: [1.0, 0.5, 0.0, -1.0], which is equal to [nominal propulsion, slow forward, stop, full reverse]

For the simulations utilizing the VO method the algorithm that has been used is the one described in the master thesis *Guidance System for Autonomous Surface Vehicles* [18]. The parameters used in these simulations are the same as in the original, apart from the MIN_DIST_ and RADIUS that has been increased from 100 and 10, to 200 and 30 respectively. This was done to get the minimum distance between the vessels more in line with the chosen limits for this thesis. It was attempted to increase the minimum distance even further, however this lead to problems in some of the scenarios.

---

[1] Distance given in meters

Table 4.1: Cost function and simulation parameters

	Euler's Method	Linear Prediction	
Parameter	Value	Value	Unit
T	150.0	300.0	sec
dt	0.05	0.5	sec
p	1.0	1.0	
q	4.0	4.0	
d_{close}	200	200	m
d_{safe}	40	40	m
K_{coll}	0.5	0.5	
ϕ_{AH}	15	15	deg
ϕ_{OT}	68.5	68.5	deg
ϕ_{HO}	22.5	22.5	deg
ϕ_{CR}	68.5	68.5	deg
κ	3.0	3.0	
K_P	2.5	4.0	
K_χ	1.3	1.3	
$K_{\Delta P}$	2.0	3.5	
$K_{\Delta\chi,\text{starboard}}$	0.9	0.9	
$K_{\Delta\chi,\text{port}}$	1.2	1.2	

Figure explanation

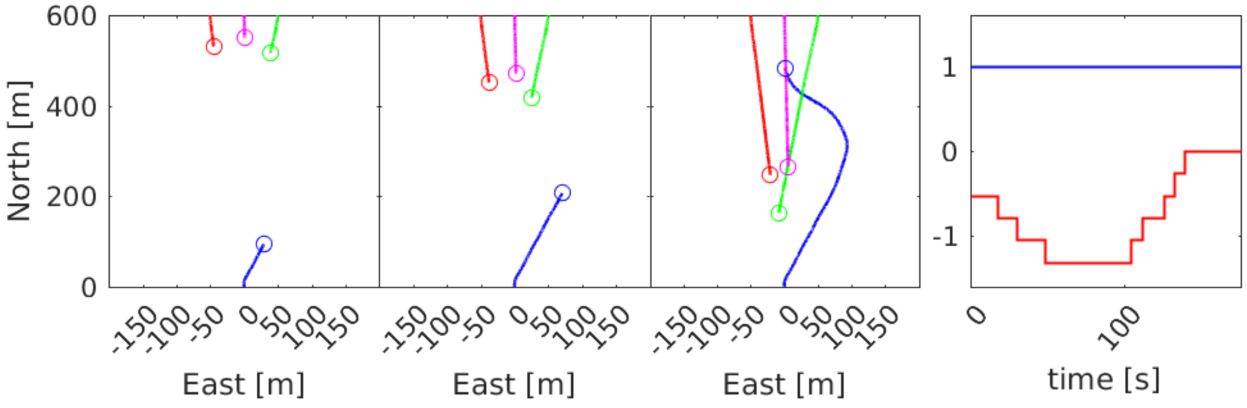

Figure 4.1: Example plot - The three plots on the left of the figure are snapshots taken at different times during the simulation. The circles mark the vessel's current position and the lines mark their progress up to this point. The ASV is marked in blue in all the plots, Ship1 is red, Ship2 is green and Ship3 is pink. The plot to the right is showing the speed-factor (blue) and the course-offset (red) in radians.

Table 4.2: Simulation Waypoints, single obstacle [1]

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	0.0	0.0	0.0	600.0
Waypoint 1	0.0	600.0	0.0	0.0

(a) Single vessel, head-on

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	0.0	0.0	-180.0	170.0
Waypoint 1	0.0	600.0	180.0	170.0

(b) Single vessel, crossing from port

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	0.0	0.0	300.0	300.0
Waypoint 1	0.0	600.0	-300.0	300.0

(c) Single vessel, crossing from starboard

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	0.0	0.0	300.0	0.0
Waypoint 1	700.0	0.0	700.0	0.0

(d) Single vessel, overtaking

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	300.0	0.0	0.0	0.0
Waypoint 1	700.0	0.0	700.0	0.0

(e) Single vessel, being overtaken

4.1 Ideal Conditions - Single Obstacle

Head-On Situation

This is maybe the simplest scenario. To comply with the COLREGS the ASV simply has to avoid the obstacle vessel by doing a clear evasive maneuver to starboard in good time. The results of using the Euler's method

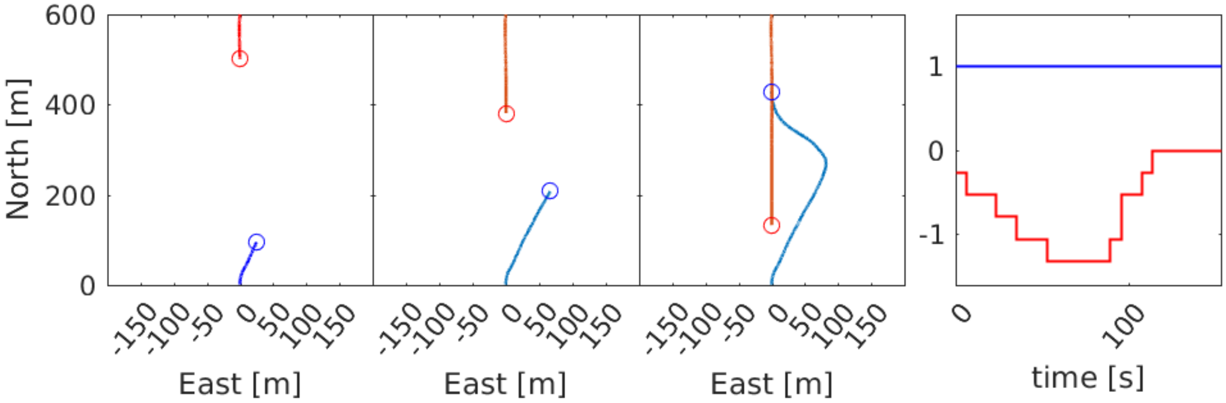

Figure 4.2: Head-on scenario. SB-MPC collision avoidance using Euler's Method

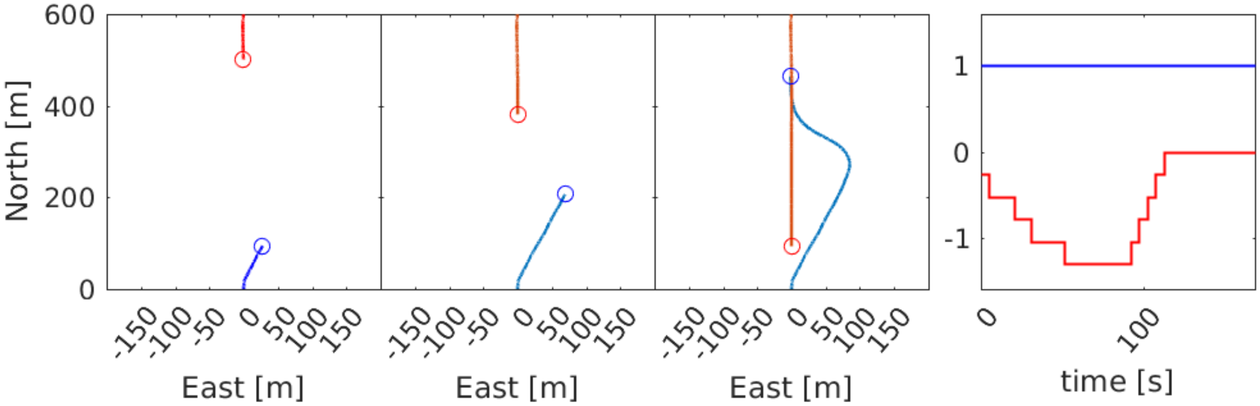

Figure 4.3: Head-on scenario. SB-MPC collision avoidance using linear prediction

and linear prediction implementation are in this case quite similar, as can be seen in figure 4.2 and 4.3. In both cases there is a step-wise increase in the heading offset. This is caused by the ASV moving further away from the planned path between each optimization. To keep an approximately constant course away from the planned path the offset must be increased. The VO reaction comes later than the SB-MPC's,this is a consequence of the difference in the minimum acceptable distance between vessels. But the qualitative behavior is quite similar, there is a gradual increase in the heading offset before it quickly returns to zero when the collision hazard disappears. The ASV's speed also decreases as the obstacle vessel approaches. This seems somewhat unnecessary, but will in the case of a collision reduce the damages.

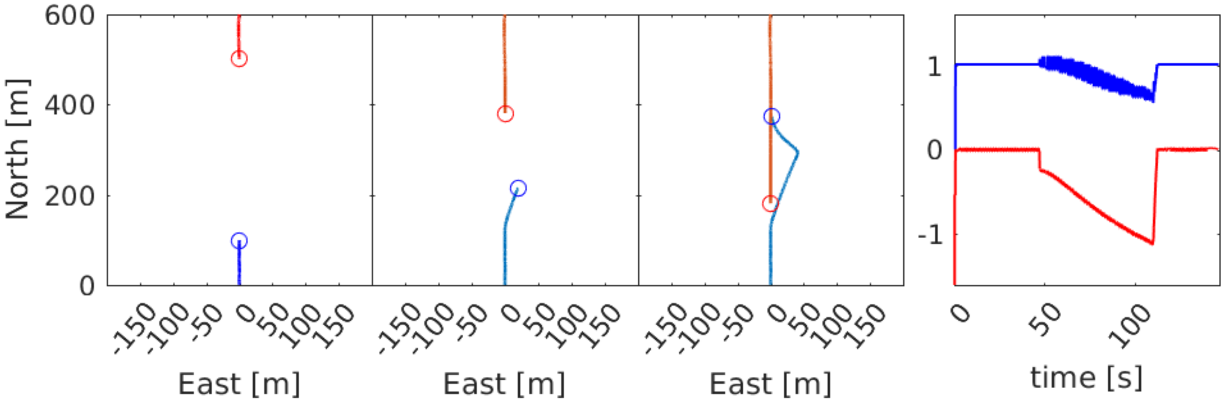

Figure 4.4: Head-on scenario. Collision avoidance using velocity obstacle

Vessel Crossing from Port

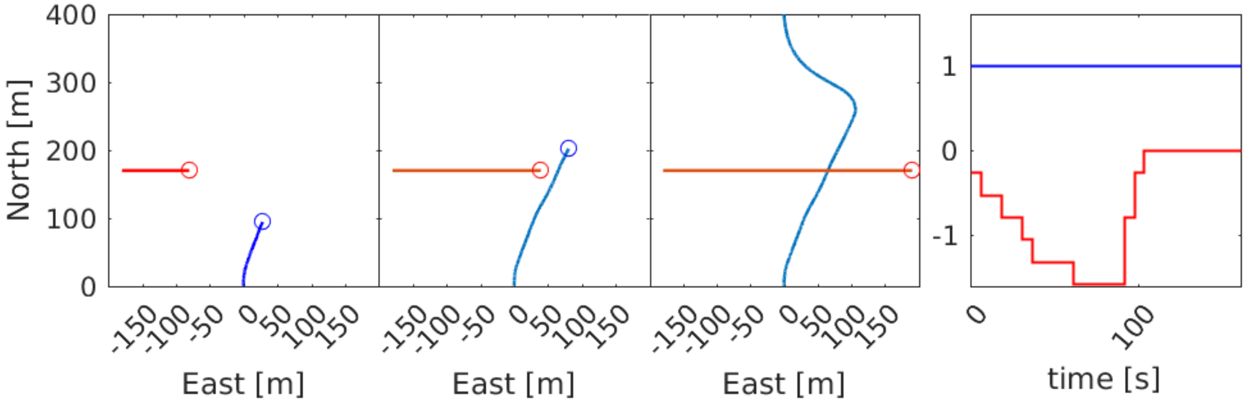

Figure 4.5: Vessel crossing from port scenario. SB-MPC collision avoidance using Euler's Method

In this scenario the obstacle is the give-way vessel and the ASV has the right to stay on. As the obstacle vessel does not take action to avoid collision, the ASV is however forced to take action. Again, the results using a linear and non-linear simulation method in the SB-MPC are very similar. In both cases the ASV passes in front of the obstacle. The ASV continues for a period with max offset such as to reach a safe distance between the vessels as soon as possible. The difference in the distance considered safe between the SB-MPC and the VO also affects scenario. In this case the two vessels seem to cross quite close. It can also be noted that the ASV actually

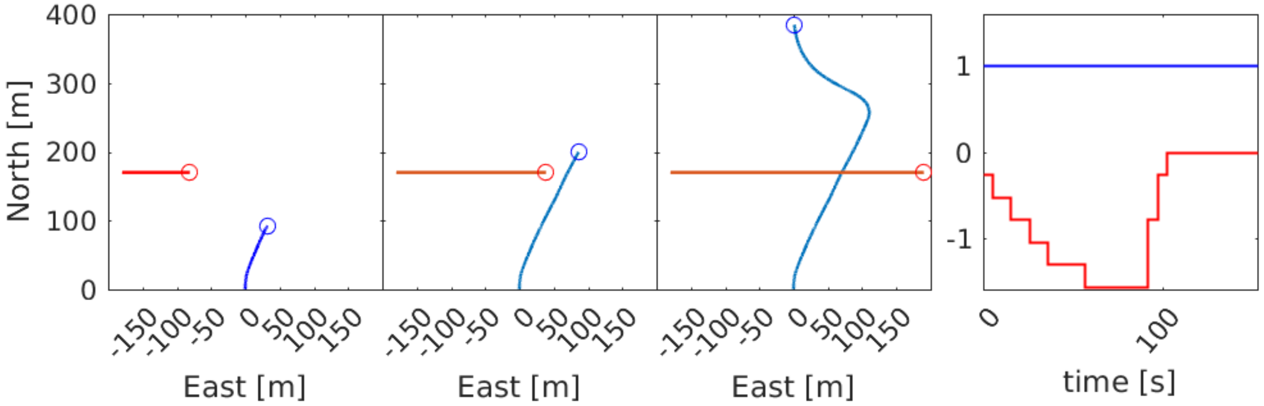

Figure 4.6: Vessel crossing from port scenario. SB-MPC collision avoidance using linear prediction

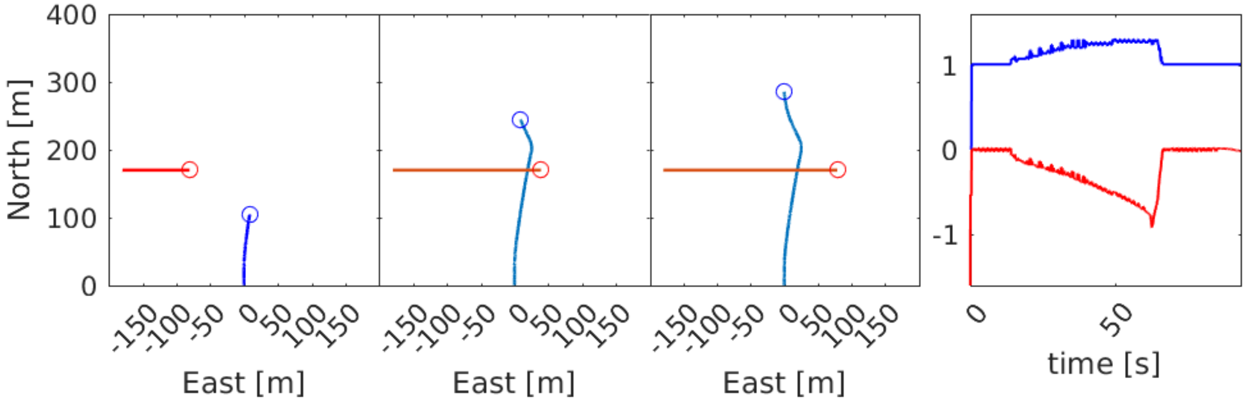

Figure 4.7: Vessel crossing from port scenario. Collision avoidance using velocity obstacle

increases it's speed as it approaches the obstacle.

Vessel Crossing from Starboard

The ASV is in this case the give-way vessel while the obstacle vessel has the right to stay on. This scenario enlightens a problem with this implementation of the SB-MPC-method. Using both linear predict and Euler's method (figure 4.5 and 4.6 respectively) the ASV makes a turn towards port before it changes course starboard

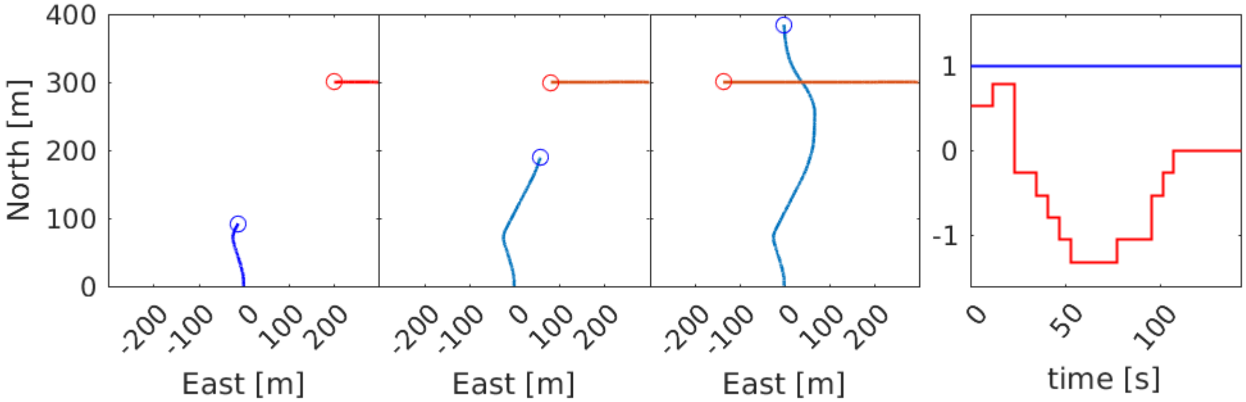

Figure 4.8: Vessel crossing from starboard scenario. SB-MPC collision avoidance using Euler's Method

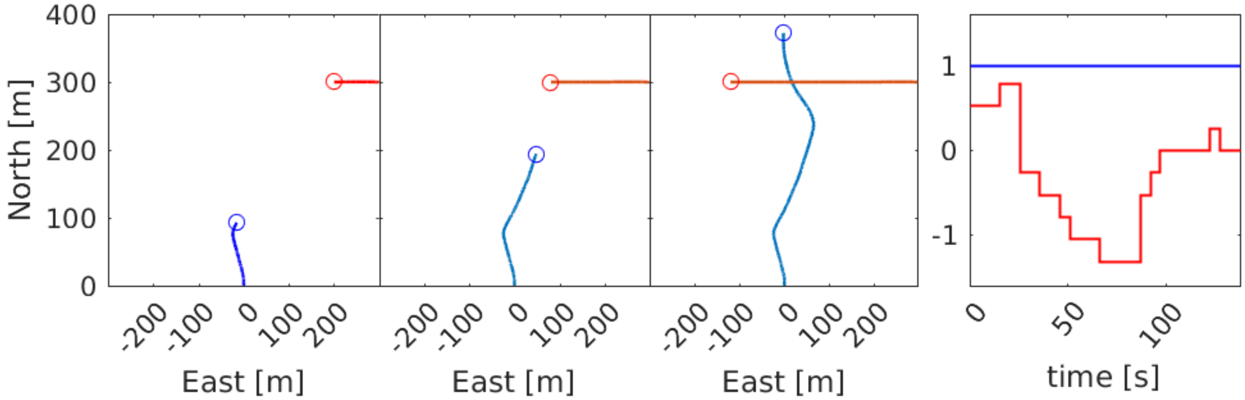

Figure 4.9: Vessel crossing from starboard scenario. SB-MPC collision avoidance using linear prediction

to pass behind the obstacle. This happens just as the obstacle enters the area where the COLREGS are said to apply. As the obstacle vessel has the right to stay on, any trajectory causing the ASV to cross in front of it is a clear violation of the COLREGS and is penalized in the cost function. Crossing too close behind the obstacle is also penalized, thus a significant course change to starboard is necessary. A large course change is however also costly and and thus making a small port turn the cheapest option. This maneuver causes the obstacle vessel to stay outside the COLREGS-region and the cost is thus only from the change itself and the deviation from the planned path. As the two vessels get closer the predicted obstacle trajectory will stay within the COLREGS-region independent of the course offset. When this happens paths passing port of the obstacle will be penalized

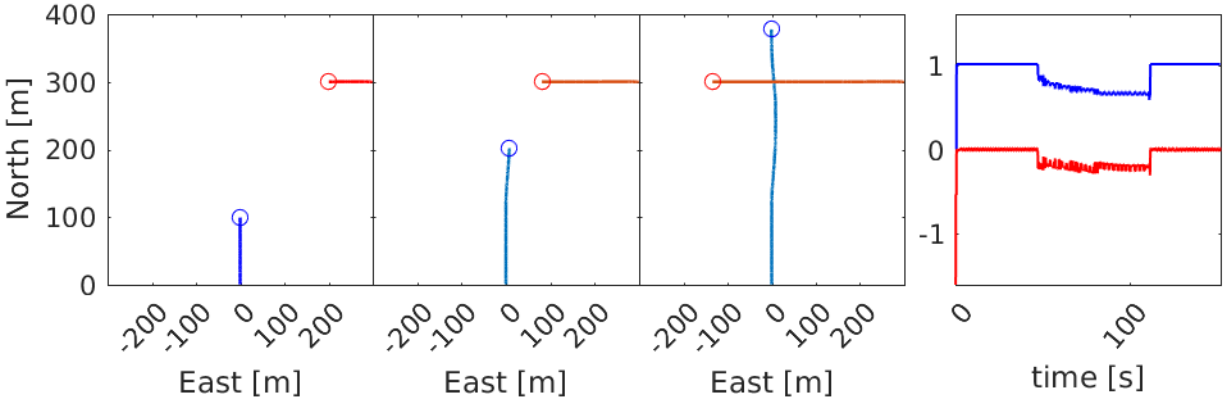

Figure 4.10: Vessel crossing from starboard scenario. Collision avoidance using velocity obstacle

and an offset leading to a path passing behind the obstacle will be cheaper. As the COLREGS states that clear action must be taken this behavior is an obvious problem, but with more careful tuning it should be possible to avoid this. Another option is to make the cost related to violating the COLREGS dependent on the distance between vessels.

Same as the SB-MPC the VO produces a path that crosses behind the obstacle vessel. This is done by reducing both the speed and heading angle. As in the previous scenarios the ASV is allowed to pass closer to the obstacle because the minimal distance acceptable is smaller. Provided that would be possible to increase the minimum distance between the vessels, the VO must be said to be the better choice in this situation. Behaving in a predictable manner is one of the most effective means to avoid collision.

Overtaking a vessel

In an overtaking situation the vessel being overtaken has the right to stay on. The vessel doing the overtaking can pass on either side of the stay-on vessel. Again there seems to be some indecisiveness in the SB-MPC. This arises from the fact that the change in heading required to regain the planned path increases. In this case the ASV starts making an overtaking maneuver by turning starboard. As it removes itself further south of the planned path , the desired heading to reach the planned path will be towards north-east. When the heading-offset reaches -30 degrees the cost of deviating further from the planned path becomes larger than the cost of

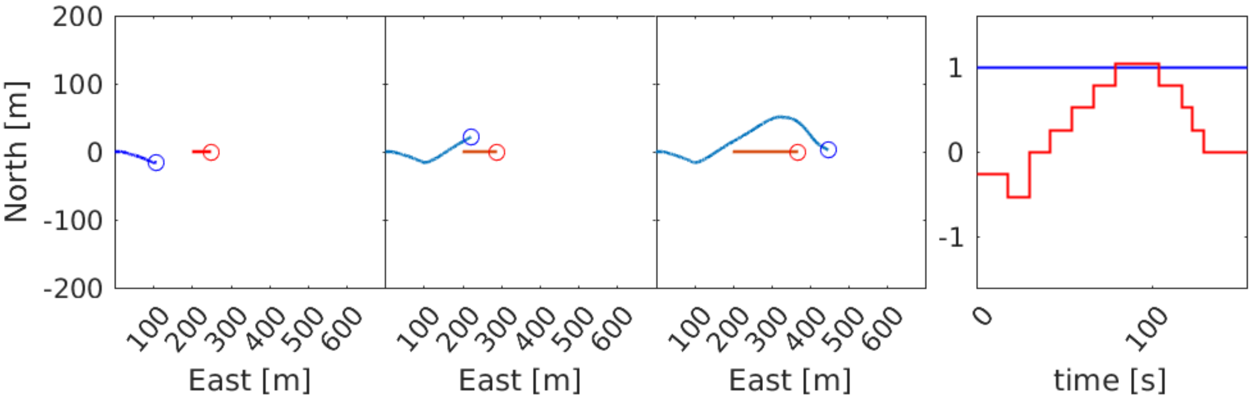

Figure 4.11: Overtaking a vessel scenario. SB-MPC collision avoidance using Euler's Method

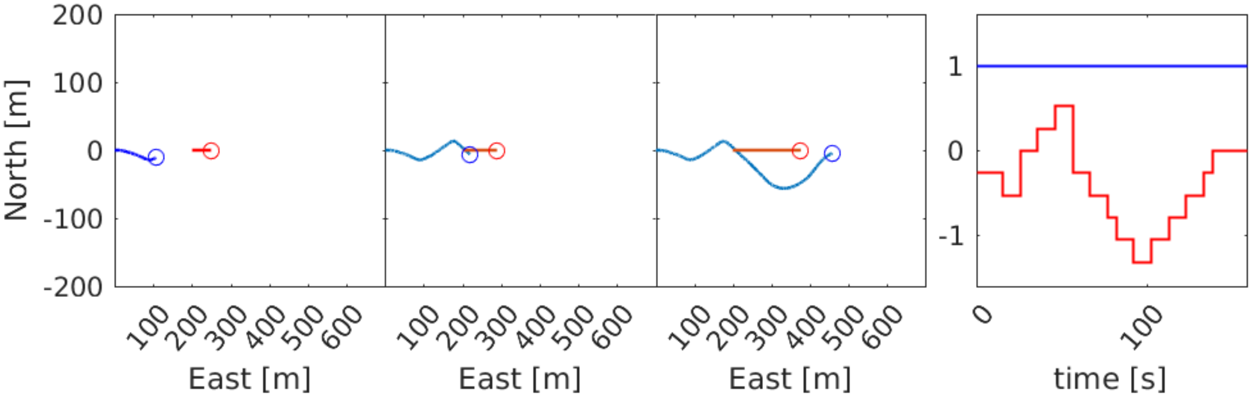

Figure 4.12: Overtaking a vessel scenario. SB-MPC collision avoidance using linear prediction

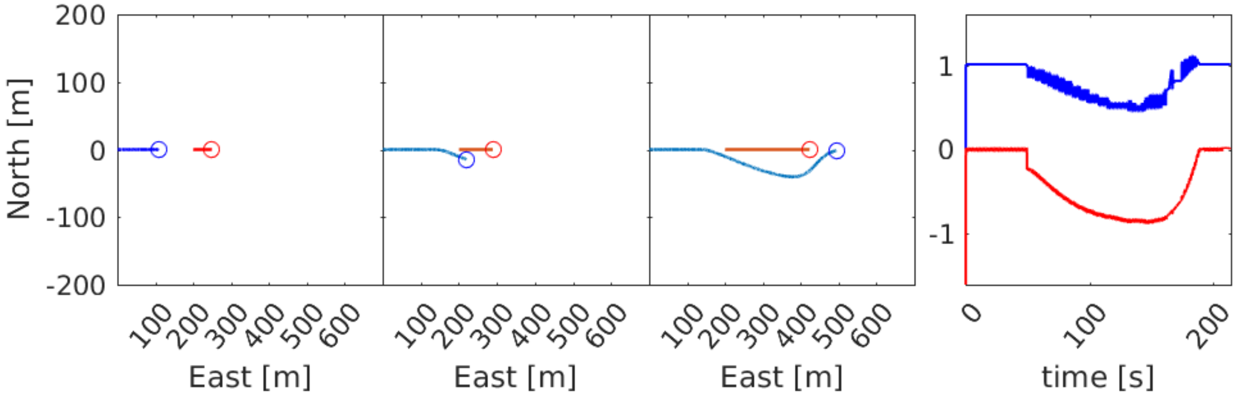

Figure 4.13: Overtaking a vessel scenario. Collision avoidance using velocity obstacle

changing the offset to zero and pass port of the obstacle vessel. In the simulation using Euler's method the cost of getting closer to the obstacle vessel keeps the ASV on a north east course. In the case of linear prediction, the change in heading appears to be instantaneous and the simulated path will then pass further away from the obstacle leading to another course change and the ASV finally passes starboard of the obstacle.

From the position plot in figure 4.13 it can be seen that VO has a more desirable path, with no unnecessary course changes. But it proceeds with a reduced speed which is counterproductive in an overtaking situation. This is presumably a consequence of increasing the radius and minimum distance parameters which forces the ASV to deviate further from the planned path than it is designed to do in an overtaking situation.

Overtaken by Vessel

In this scenario the ASV has the right to stay on, but as the obstacle is not showing any sign to give way the ASV is forced to take action. In all the simulations the ASV changes it course and avoids collision. In the cases using SB-MPC it can to human eyes seem that the change back to the planned path takes place a bit early as this starts before the obstacle has passed. A good distance is keep throughout the simulation, but as other vessels might be controlled by a human operator this is nevertheless something that should be kept in mind.

With VO the avoidance maneuver starts later, again a consequence of the difference in the minimum allowed distance. It can also be seen from the offset plot in figure 4.16 that the speed is increased, this is however also

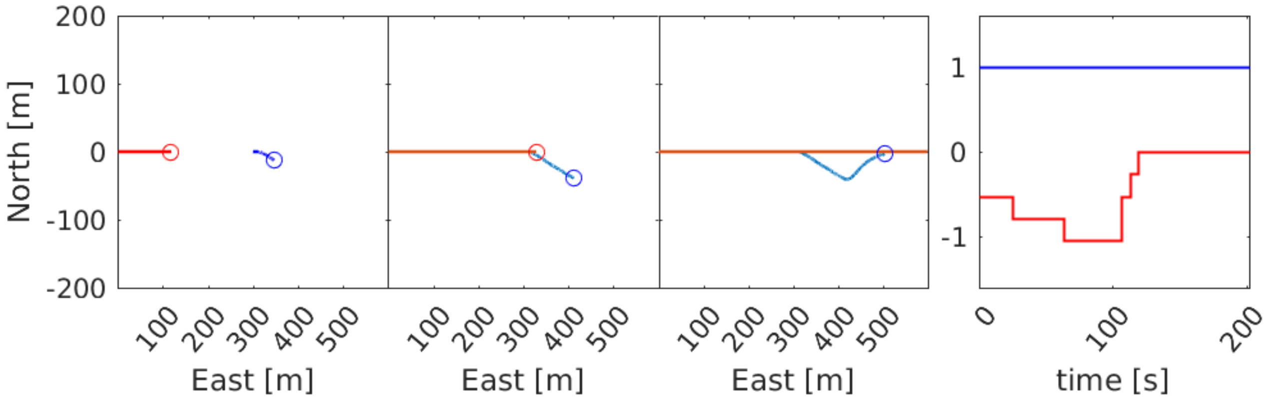

Figure 4.14: Overtaken by a vessel scenario. SB-MPC collision avoidance using Euler's method

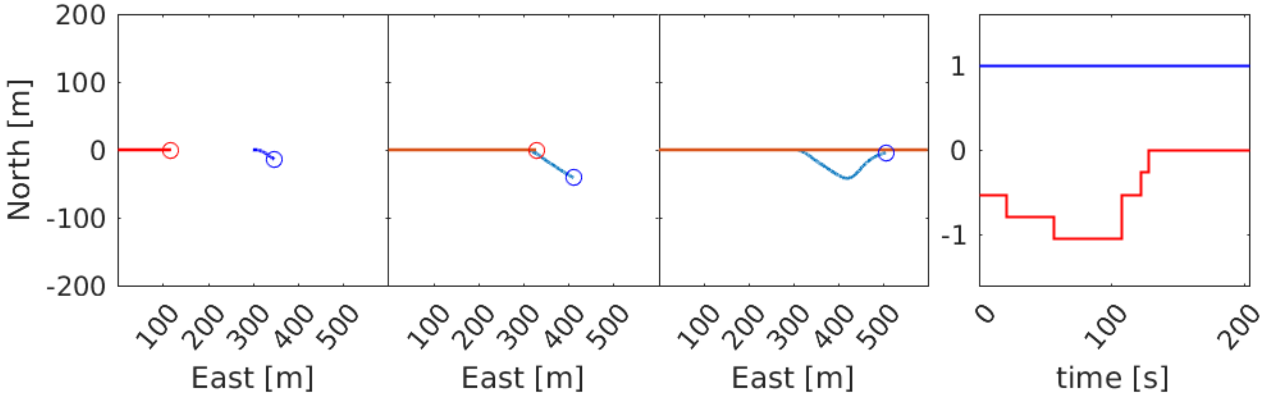

Figure 4.15: Overtaken by a vessel scenario. SB-MPC collision avoidance using linear prediction

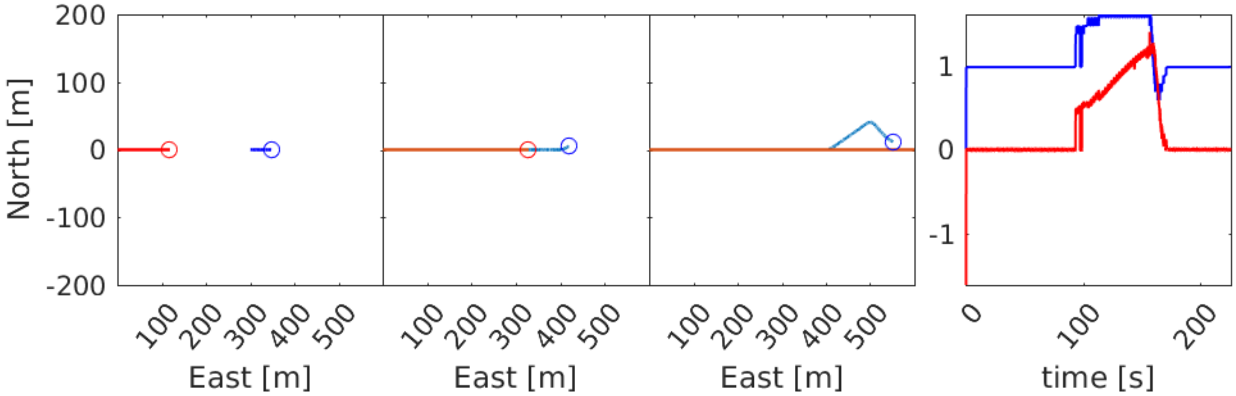

Figure 4.16: Overtaken by a vessel scenario. Collision avoidance using velocity obstacle

thought to follow from the above mentioned parameter as the ASV makes a larger course deviation than it is designed for and the situation is thus registered as a crossing situation.

4.2 Ideal Conditions - Multiple Obstacles

Table 4.3: Simulation Waypoints, multiple obstacles[2]

	ASV		Ship1		Ship2		Ship3	
	East	North	East	North	East	North	East	North
Initial position	0.0	0.0	-50.0	600.0	50.0	600.0	0.0	
Waypoint 1	0.0	600.0	0.0	0.0	-30.0	0.0	10.0	0.0

(a) Multiple vessels, head-on

	ASV		Ship1		Ship2		Ship3	
	East	North	East	North	East	North	East	North
Initial position	0.0	0.0	-190.0	190.0	-190.0	175.0	-195.0	140
Waypoint 1	0.0	600.0	150.0	165.0	150.0	140.0	-195.0	140

(b) Multiple vessels, crossing from port

	ASV		Ship1		Ship2		Ship3	
	East	North	East	North	East	North	East	North
Initial position	0.0	-250.0	585.0	440.0	600.0	480.0	600.0	420.0
Waypoint 1	0.0	600.0	-200.0	400.0	-200.0	310.0	-200.0	380.0

(c) Multiple vessels, crossing from starboard

	ASV		Ship1		Ship2		Ship3	
	East	North	East	North	East	North	East	North
Initial position	0.0	0.0	500.0	450.0	-250.0	450.0	10.0	600.0
Waypoint 1	0.0	600.0	150.0	400.0	150.0	0.0	-30.0	200.0
Waypoint 2			-400.0	150.0			-150.0	20.0

(d) Multiple vessels making course changes - I

	ASV		Ship1		Ship2		Ship3	
	East	North	East	North	East	North	East	North
Initial position	0.0	0.0	100.0	10.0	0.0	600.0	400.0	400.0
Waypoint 1	0.0	600.0	100.0	130.0	-60.0	500.0	-300.0	300.0
Waypoint 2			-500.0	600.0	-150.0	200.0		

(e) Multiple vessels making course changes - II

---

[2] Distance given in meters

Head-On Situation

With multiple vessels the situation becomes more complicated and it there can exist several viable options, but in this case it is obvious that the best action is to make a starboard maneuver as to avoid all the obstacles.

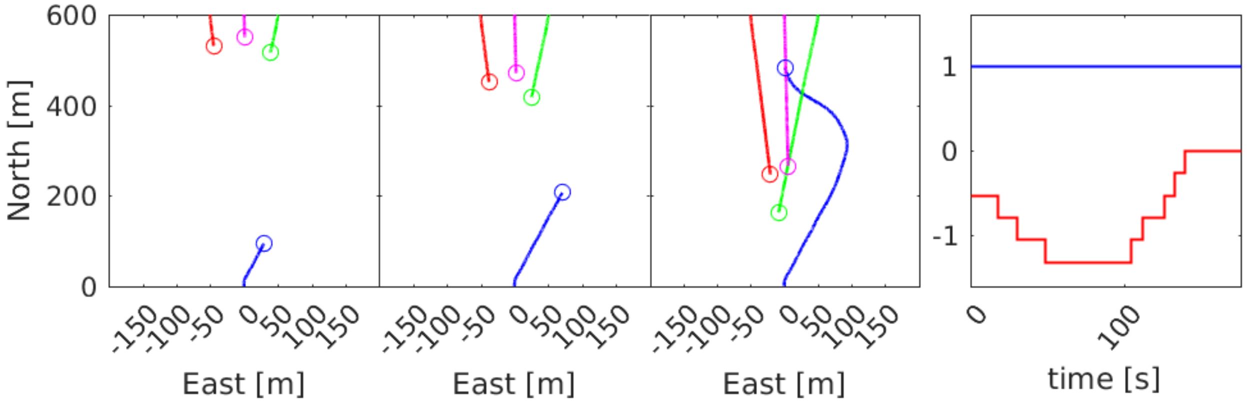

Figure 4.17: Head-on scenario. SB-MPC collision avoidance using Euler's method

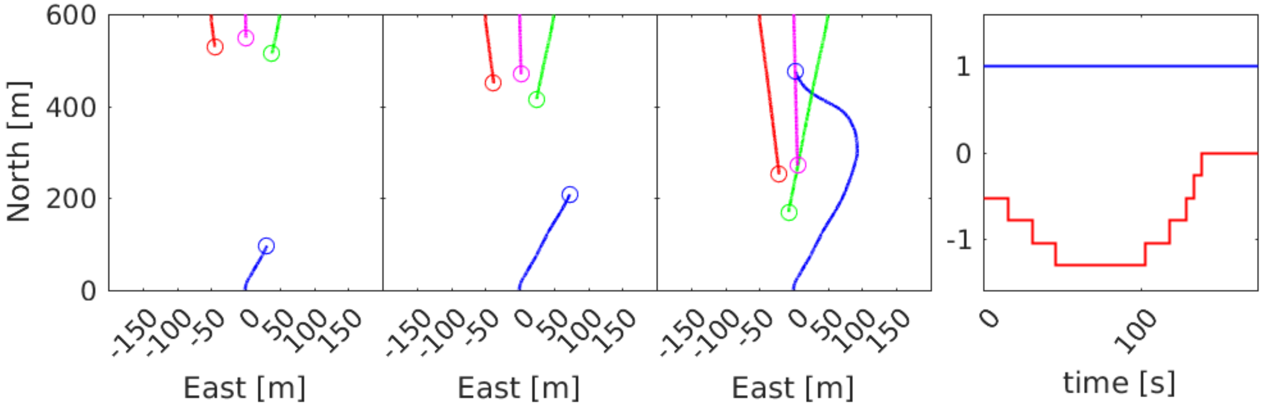

Figure 4.18: Head-on scenario. SB-MPC collision avoidance using linear prediction

With the SB-MPC both Euler's method and linear prediction produces a path starboard of the obstacles. The avoidance maneuver starts immediately and the ASV keeps a good distance to the obstacle vessels.

The VO also passes starboard of the obstacles, but has a less smooth trajectory, it also reduces it's speed as the

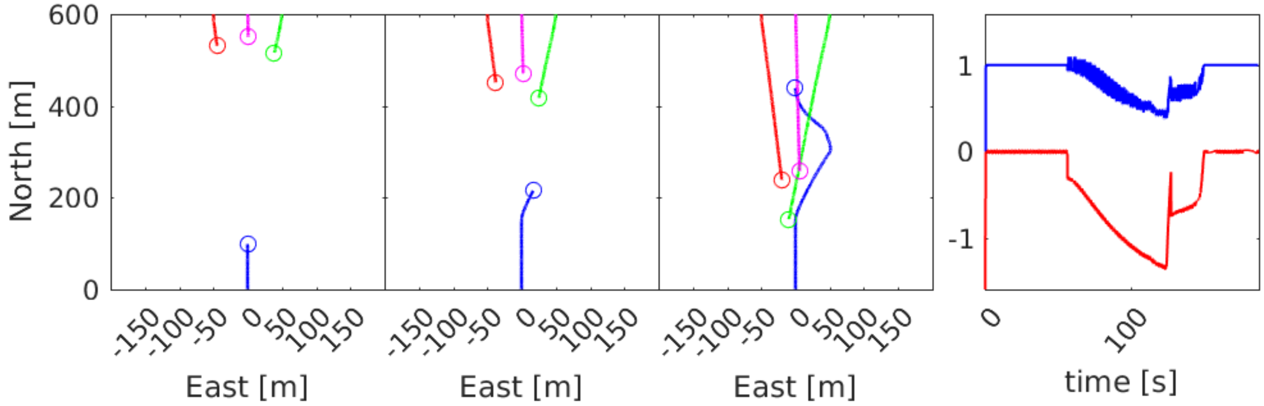

Figure 4.19: Head-on scenario scenario. Collision avoidance using velocity obstacle

distance to the obstacles decreases.

Vessels Crossing from Port

As in the single obstacle scenario the ASV has the right to stay on. But as no action to avoid collision is taken by the other vessels, it is forced to take action.

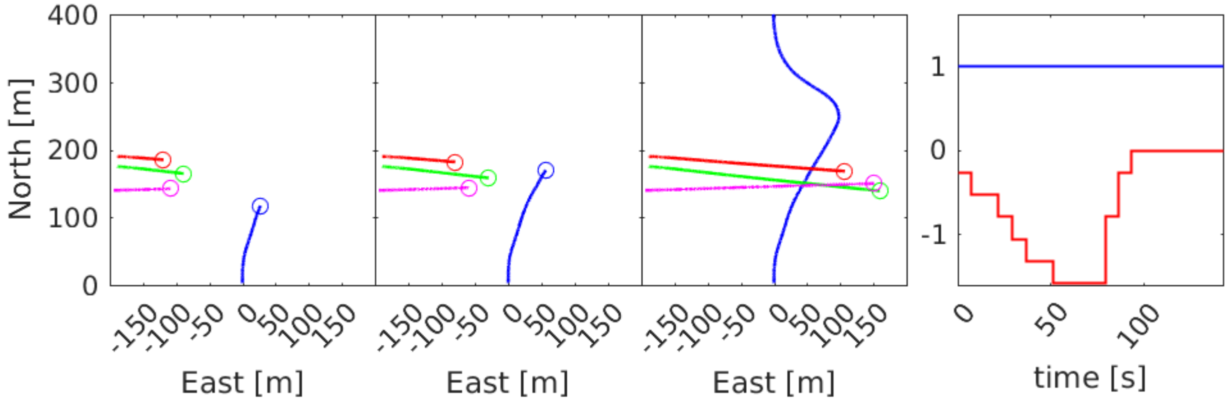

Figure 4.20: Vessel crossing from port scenario. SB-MPC collision avoidance using Euler's method

Again, the two models in the SB-MPC method show a similar behavior immediately making a starboard turn to

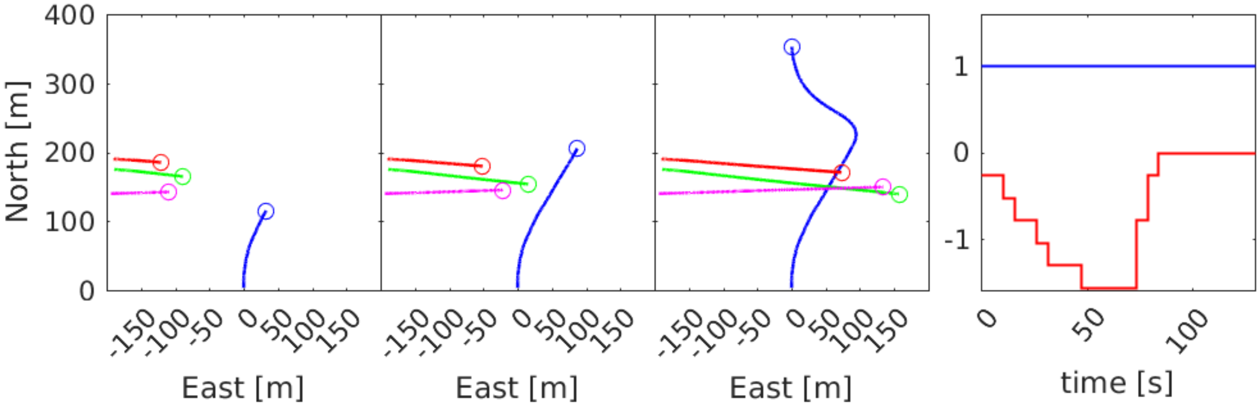

Figure 4.21: Vessel crossing from port scenario. SB-MPC collision avoidance using linear prediction

keep a safe distance to the obstacle vessels. The offset is kept at the maximum of - 90 degrees even after passing the obstacles to attain a larger distance between them and the ASV.

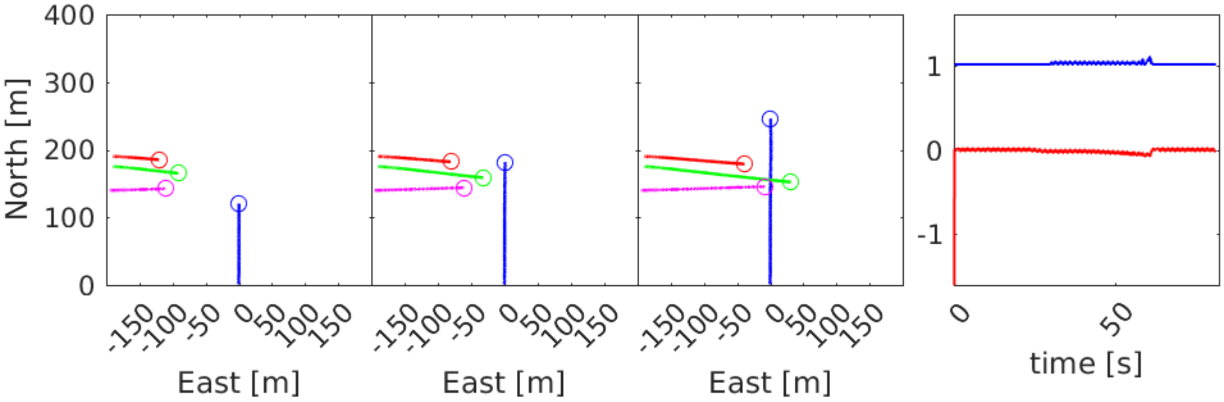

Figure 4.22: Vessel crossing from port scenario. Collision avoidance using velocity obstacle

The difference in the acceptable minimum distance between the SB-MPC and the VO has become very clear in this scenario. This is explained by the fact that the ASV has the right to stay on, meaning that violation of the safety zone is the only factor that affects the ASV's behavior. There are some minor adjustments to both the speed and the course, but it does not take any significant action to avoid the obstacles.

Vessels crossing from Starboard

The ASV must in this scenario give way to the obstacle vessels, either by reducing its speed or making a turn to pass behind the other vessels.

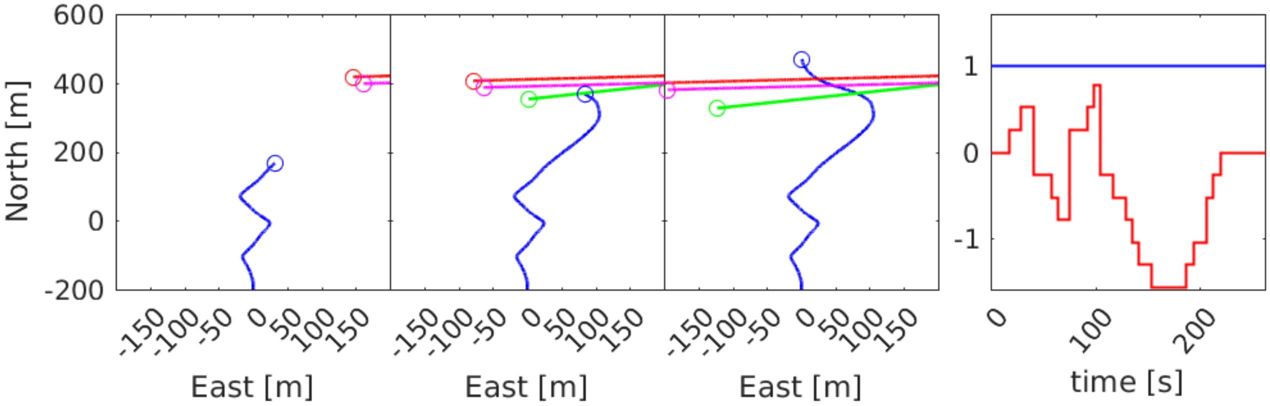

Figure 4.23: Crossing from starboard scenario. SB-MPC collision avoidance using Euler's method

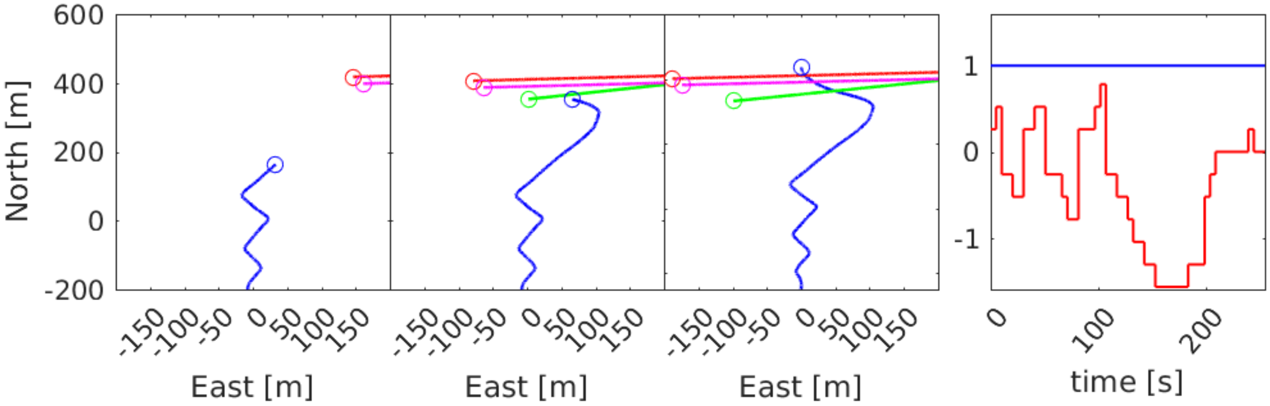

Figure 4.24: Crossing from starboard scenario. SB-MPC collision avoidance using linear prediction

From the plots it is clear that there are still improvements to make with regards to tuning of the SB-MPC method. Both the linear prediction and Euler's method implementations causes several unnecessary course changes before passing the obstacles. This is not optimal with regards to fuel efficiency nor for the COLREGS compliance.

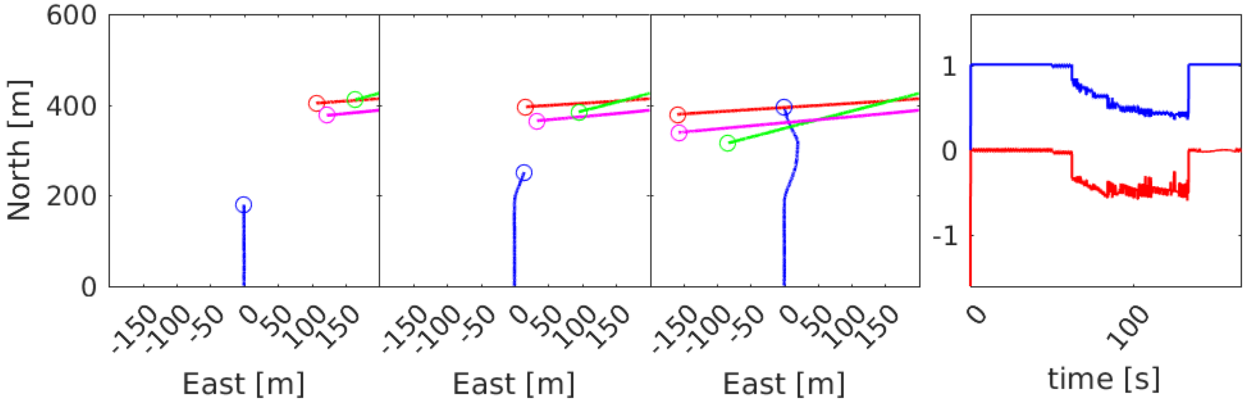

Figure 4.25: Crossing from starboard scenario. Collision avoidance using velocity obstacle

The VO behave very nicely in this scenario. It makes an early course change and also gradually reduces the speed for so to pass behind the obstacles in a safe distance.

Vessels making course changes - I

In scenarios involving multiple obstacles that change their course during the simulation it is more difficult to decide on what would be the optimal behavior. It is however important to see how the CAS reacts to this kind of behavior.

In this scenario there is a notable difference between linear prediction and Euler's method. Both methods produce a path that starts with a starboard offset. As the ASV reaches a distance where its path is no longer influenced by Ship1 (green) and Ship2 (pink), the course offset decreases. Then the course of Ship3 (red) changes so that it will pass very close to the ASV. At such a close distance the cheapest solution to avoid getting too close is to reduce speed while still keeping an offset to the course.

As the implementation using linear prediction has a higher cost on deviation from planned speed and speed changes there is no speed offset. Instead it applies a maximum course offset for a longer period than in the simulation using Euler's method and thus avoids making any further adjustments when the closest obstacle, marked in red, changes its course.

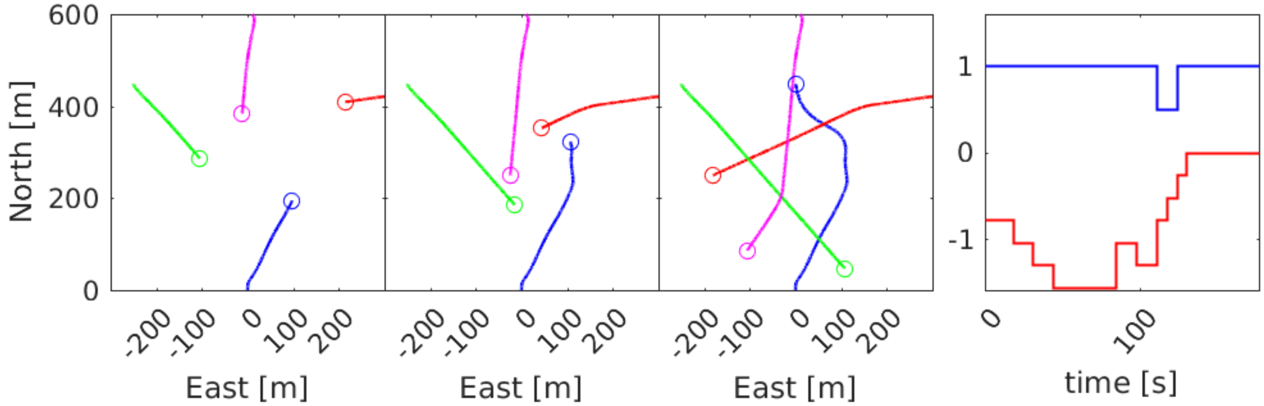

Figure 4.26: Multiple obstacles changing course scenario I. SB-MPC collision avoidance using Euler's method

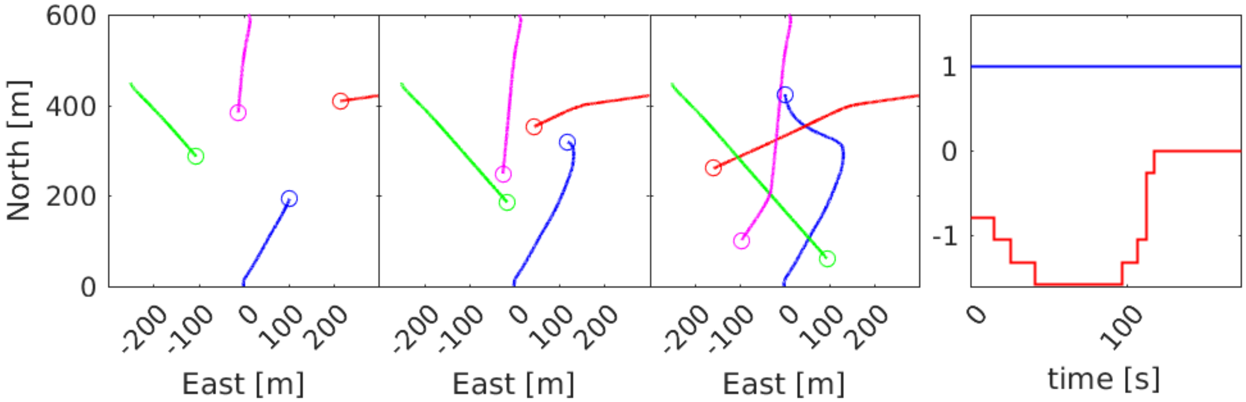

Figure 4.27: Multiple obstacles changing course scenario I. SB-MPC collision avoidance using linear prediction

This scenario shows a problem related to the limited planning capabilities in the VO. The method assumes linear behavior in both the ASV and the other vessels and choses a velocity that will stay valid. But because the algorithm lacks the risk assessment of the SB-MPC it is not likely to find an optimal solution. In this situation it ends up in a position where it is locked in by the vessels around it and the ASV comes to a complete halt while the course offset changes rapidly.

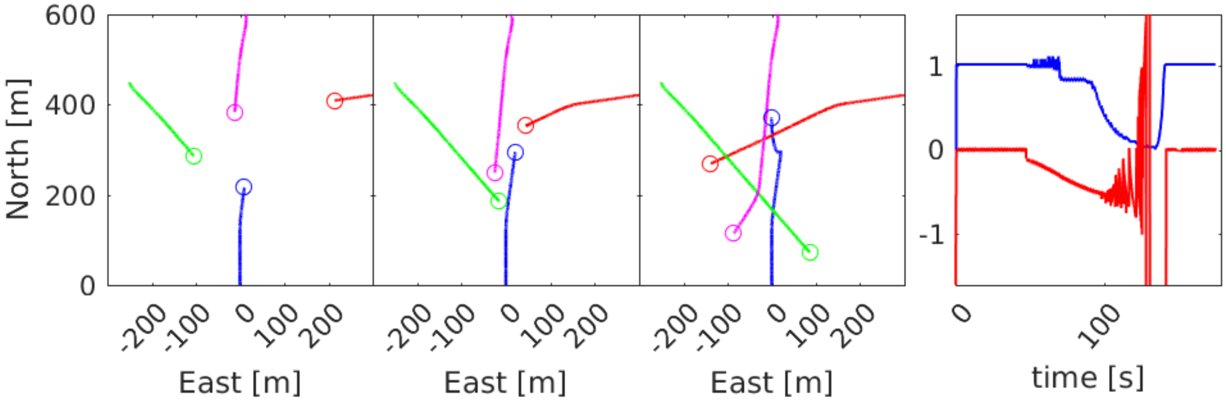

Figure 4.28: Multiple obstacles changing course scenario II. Collision avoidance using velocity obstacle

Vessels making course changes - II

This simulation represents an interesting problem when an overtaking situation changes into a crossing, see [2].

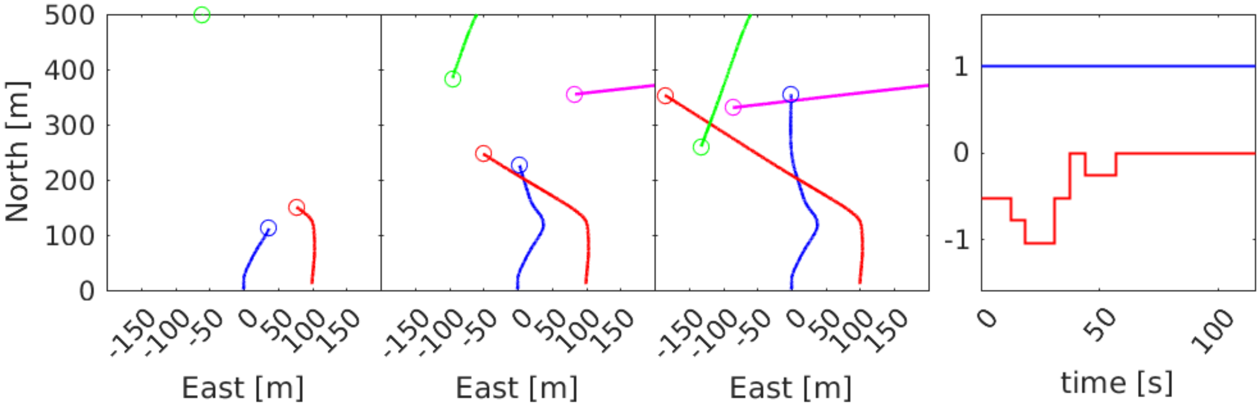

Figure 4.29: Multiple obstacles changing course scenario II. SB-MPC collision avoidance using Euler's method

As the simulation starts the ASV is moving parallel to Ship1 (red), but with a lower speed. The risk of collision is therefore minimal and it is the Ship3 (pink) that influences the course, leading to the ASV turning starboard to pass behind it.

When the Ship1 makes a port turn it is on a direct collision course with the ASV. The SB-MPC then reduces the course offset to zero and they run approximately parallel until the obstacle again is a safe distance ahead and the ASV can again follow the planned path. By this time the Ship3 (pink) has advanced sufficiently to be out of the ASV's safety zone and no further maneuvers are required.

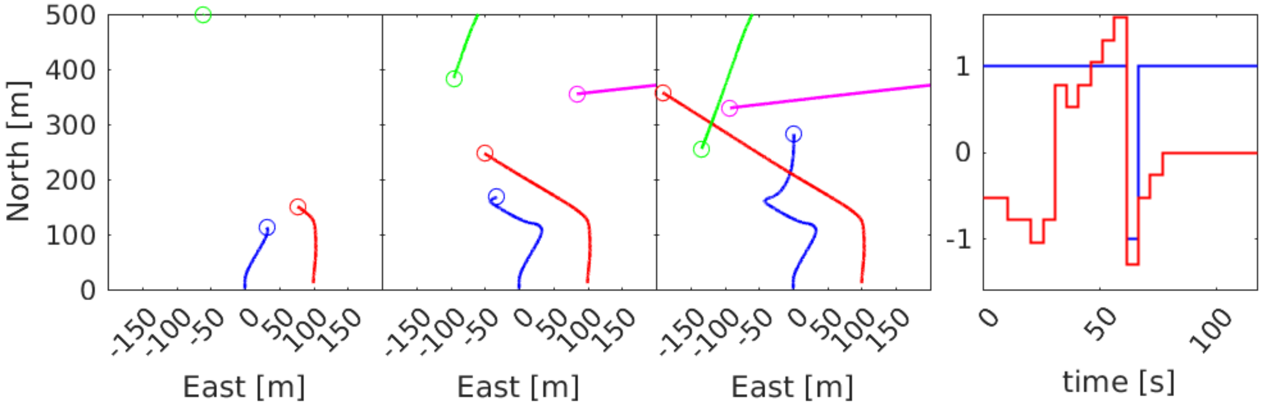

Figure 4.30: Multiple obstacles changing course scenario II. SB-MPC collision avoidance using linear prediction

The linear prediction implementation shows a slightly different behavior in this situation and takes stronger action to avoid collision when Ship1 (red) changes its course. The SB-MPC sets the offset to the maximum 90 degrees, i.e. port of the planned course. Then the speed offset is set to -1, i.e. reverse while the course offset again is set starboard of the planned course. This rapid and drastic change in speed and course puts a strain on the ASV's actuators and should be avoided.

In this case the VO performs quite well. The decreasing speed along with the course change lets Ship3 (red) pass in front without any excess of maneuvering.

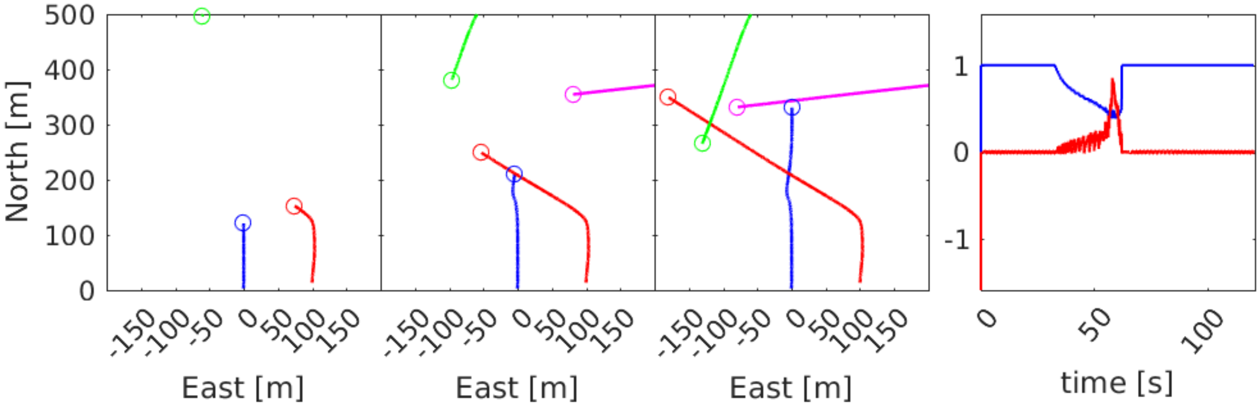

Figure 4.31: Multiple obstacles changing course scenario II. Collision avoidance using velocity obstacle

4.3 Problem Scenarios

Table 4.4: Simulation Waypoints, single obstacle [3]

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	0.0	0.0	0.0	600.0
Waypoint 1	0.0	300.0	0.0	0.0
Waypoint 2	200.0	600.0		

(a) ASV missing waypoint

	ASV		Ship1	
	East [m]	North [m]	East [m]	North [m]
Initial position	0.0	0.0	0.0	600.0
Waypoint 1	50.0	200.0	0.0	0.0
Waypoint 2	-200.0	250.0		

(b) Course change with course offset

Missing waypoint

As mentioned in chapter 3, the SB-MPC is completely disconnected from the guidance module. An unwanted effect of this appears when an obstacle vessel forces the ASV off path and causes it to miss the waypoint. As there is no internal correction, the ASV continues on the same course and with the same speed, going completely off the preplanned path.

The solution to this problem depends on the objective of the ASV. If the waypoints are placed to constitute an approximate path for the vessel an alternative switching criterion can be implemented so that the switch to the next waypoint happen when the vessel passes the current. This is the "progress along path" [13, 18] or "along-track" method [4]. This solution does however defy the point of removing the guidance from the CAS as it too requires knowledge about the guidance system.

It is possible to envision some sort of detection of missed waypoints, but it can seem like the implementation described in [8] where the guidance system is integrated in the CAS is more appropriate choice.

---

[3] Distance given in meters

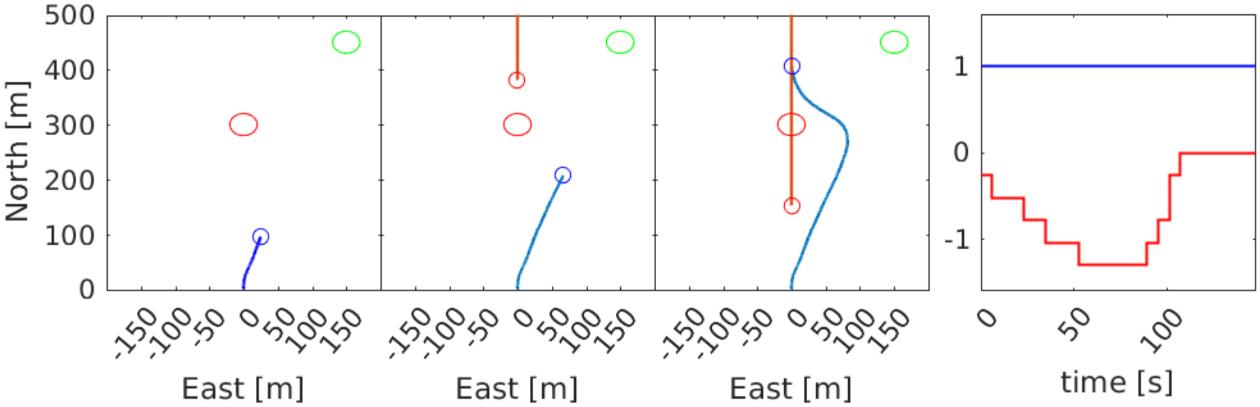

Figure 4.32: ASV missing waypoint scenario, SB-MPC collision avoidance using Euler's method

Course change with course offset

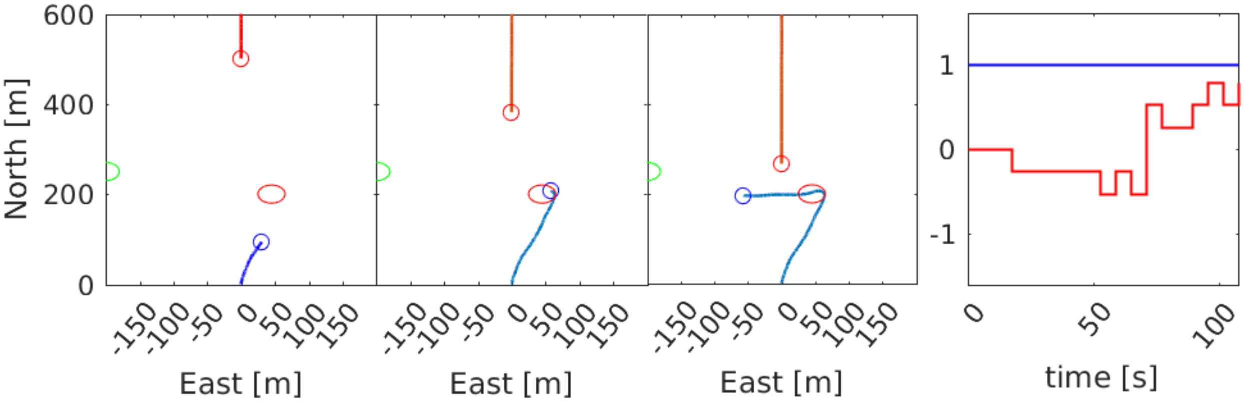

Figure 4.33: Course change with course offset scenario, SB-MPC collision avoidance using Euler's method

Another scenario that might cause problems is when a switch between waypoints are preformed while there is an offset to the desired course. As a large offset would lead to missing the waypoint, this can not be tested with maximum offset under the current implementation. For small offsets, 45 degrees in this scenario, it does not have a negative impact on the performance of the ASV. A possible solution is nevertheless to increase the update frequency of the offsets to minimize the time where an outdated offset is applied.

4.4 Changing the Scenario Grid

The scenario grid is made up by all the possible combinations of speed and course offsets. [8] sates that the grid should be as fine as computation time allows as this will increase the performance of the system. The SB-MPC preforms well in most of the simple COLREGS situations. It is therefore more interesting to see what consequences a finer grid will have on the ASV's behavior in more complicated situations. The integrator used in the following simulations is Euler's first order method and the simulation and tuning parameters are kept as in the previous sections, see table 4.1.

4.4.1 Modified Control Scenario Grid I

As the tuning has been set to favor changing course over speed in a collision avoidance situation the speed factors are kept as before to investigate the effect of a very fine-grained set of possible course offsets.The control scenarios that are used in the following simulations are as follows:

- Course offset [degrees]: -90, -85, -80, -75, -70, -65, -60, -55, -50, -45, -40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90.
- Speed factor: [1.0, 0.5, 0.0, -1.0], which is equal to [nominal propulsion, slow forward, stop, full reverse]

The total number of control behaviors is in this case is 37 × 4 = 148 which is a large number compared to the original set which includes 4 × 13 = 52 different scenarios.

Vessels changing course scenario I

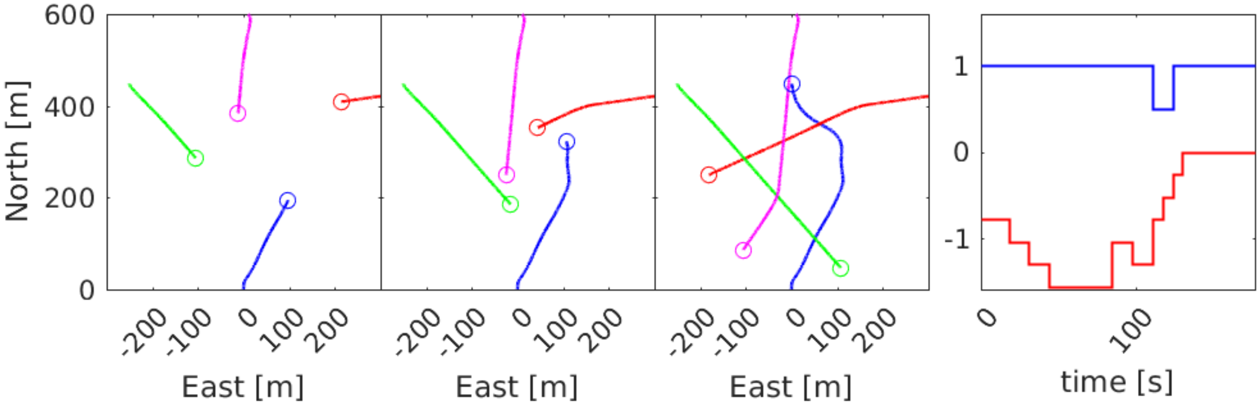

Figure 4.34: Multiple obstacles changing course scenario I. SB-MPC collision avoidance using Euler's method and original scenario grid

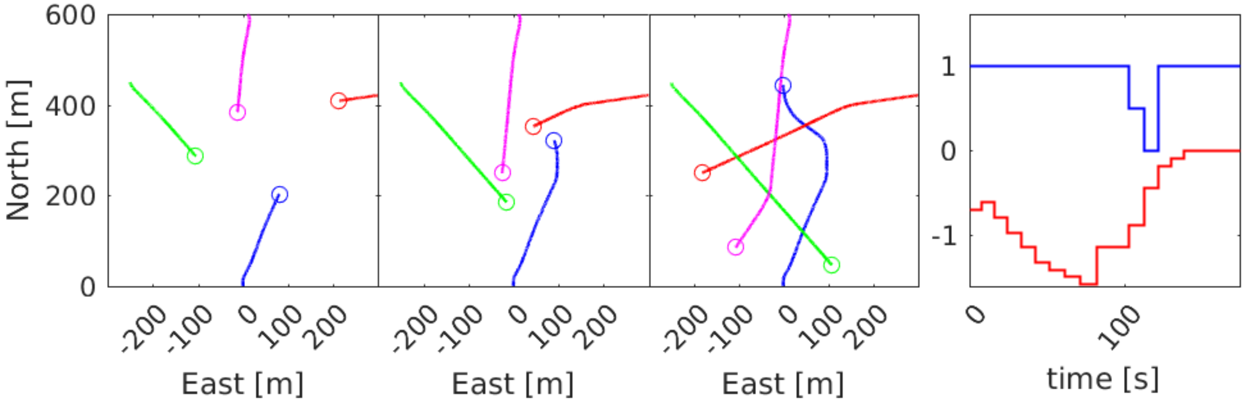

Figure 4.35: Multiple obstacles changing course scenario I. SB-MPC collision avoidance using Euler's method and modified scenario grid I.

Vessels changing course scenario II

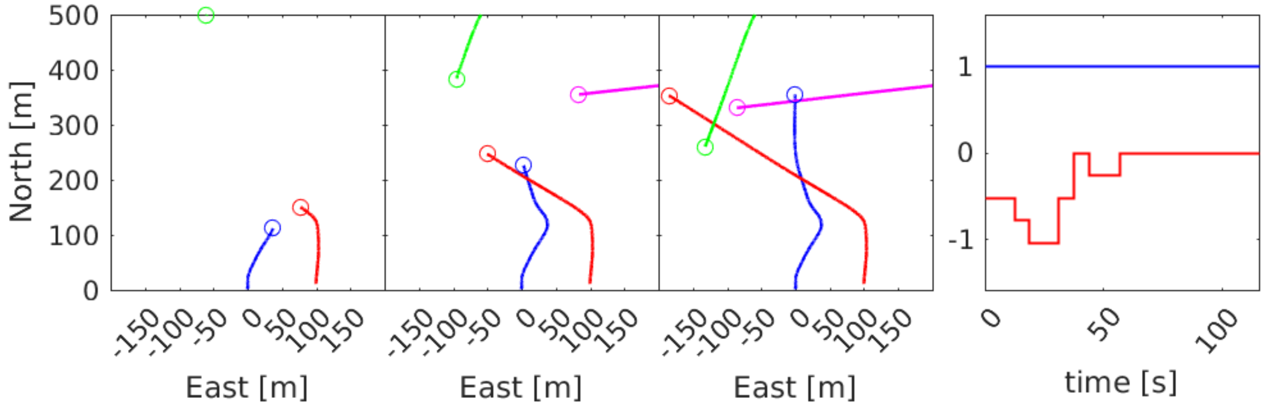

Figure 4.36: Multiple obstacles changing course scenario II. SB-MPC collision avoidance using Euler's method and original scenario grid

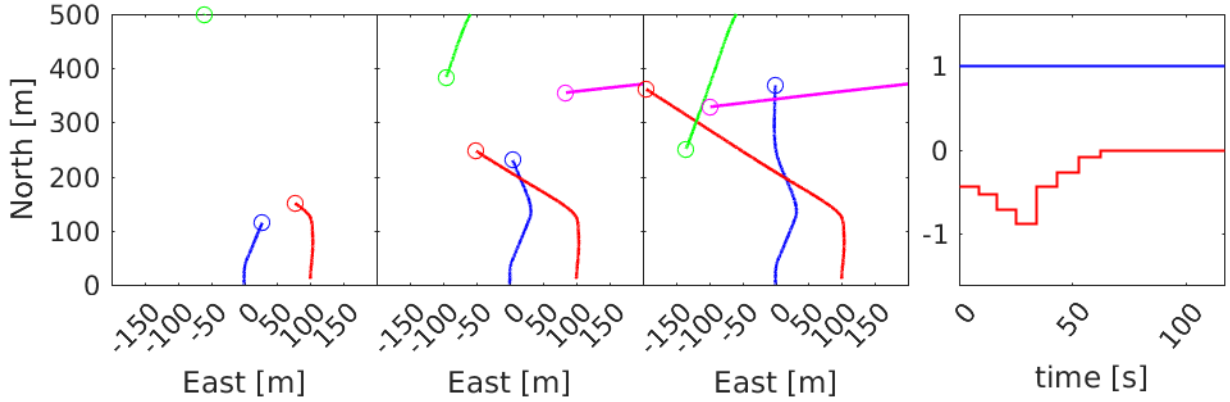

Figure 4.37: Multiple obstacles changing course scenario II. SB-MPC collision avoidance using Euler's method and modified scenario grid I.

Modified Control Scenario Grid II

As seen in the previous section changing the grid of one of the offsets also affects the use of the other. In this section the number of control behaviors have been increased from the original set by using a finer grain for both speed and course and now totals 6 × 19 = 114 scenarios. This is less than the 148 possibilities using scenario grid I, but still a big increase from the original 52.

- Course offset [degrees]: -90, -80, -70, -60, -50, -40, -30, -20, -10, 0, 10, 20, 30, 40, 50, 60, 70, 80, 90.
- Speed factor: [1.0, 0.75, 0.5, 0.0, -0.5, -1.0]

Vessels changing course scenario I

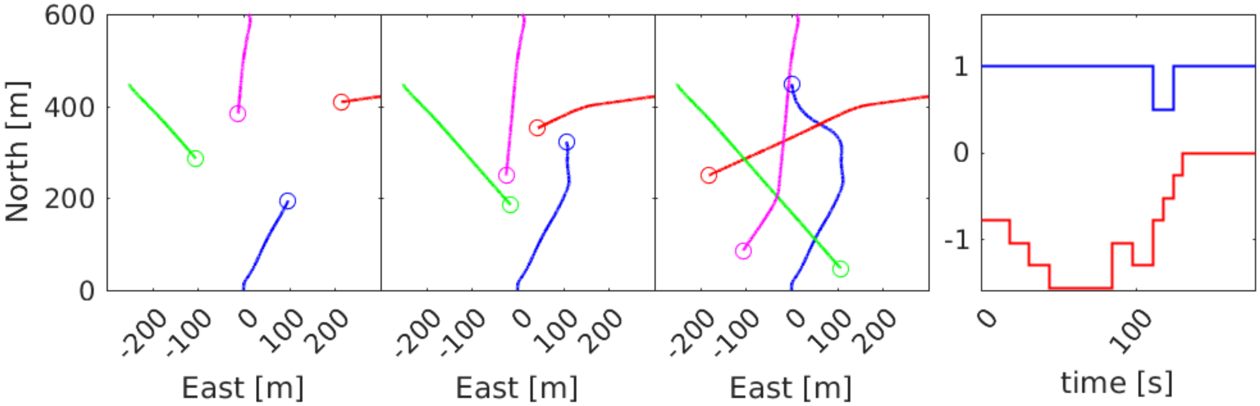

Figure 4.38: Multiple obstacles changing course scenario I. SB-MPC collision avoidance using Euler's method and original scenario grid

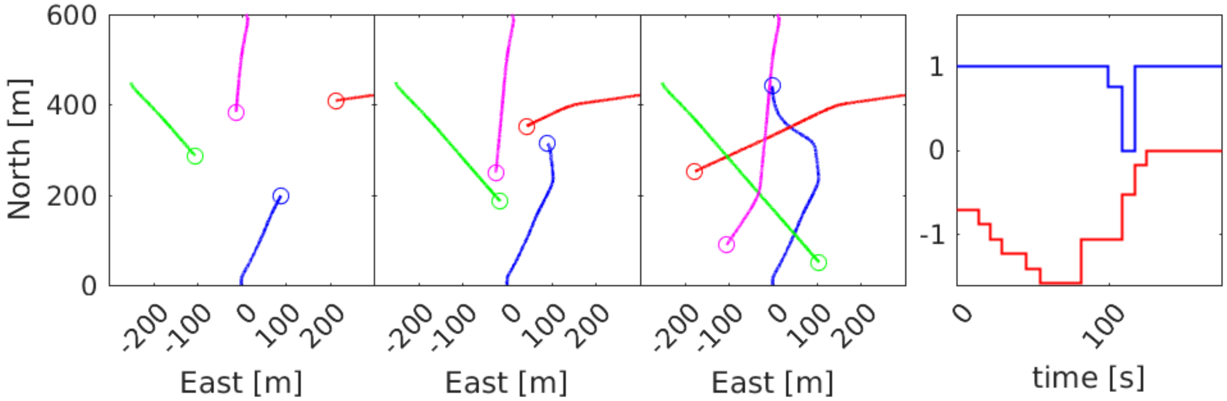

Figure 4.39: Multiple obstacles changing course scenario I. SB-MPC collision avoidance using Euler's method and modified scenario grid II.

Vessels changing course scenario II

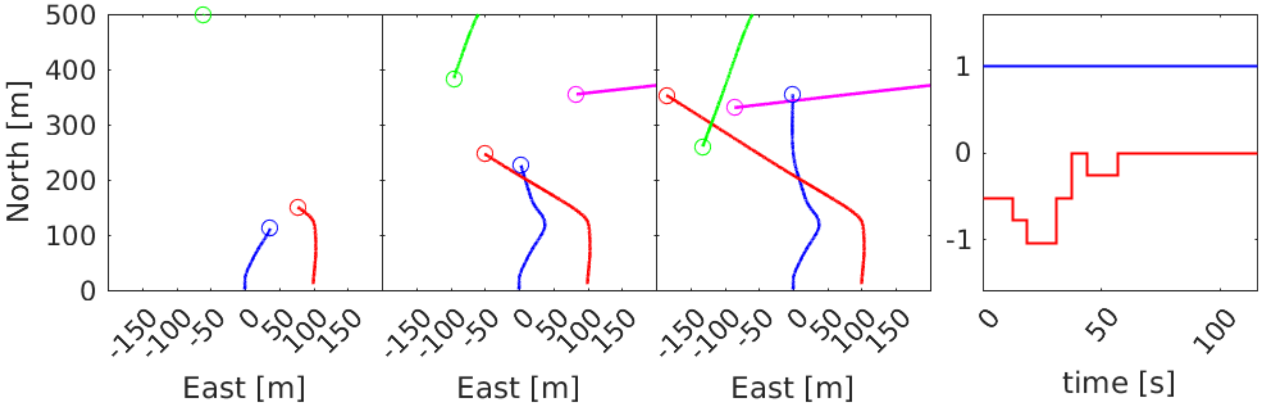

Figure 4.40: Multiple obstacles changing course scenario II. SB-MPC collision avoidance using Euler's method and original scenario grid

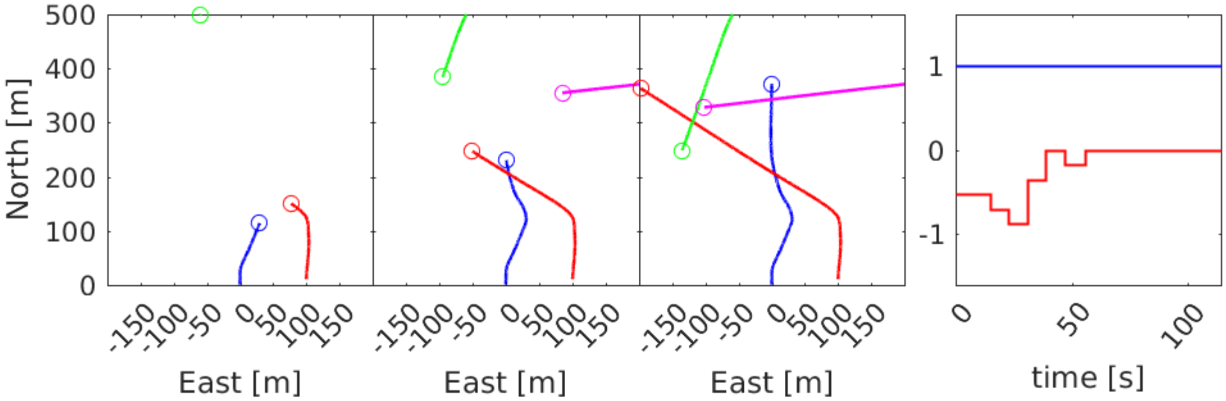

Figure 4.41: Multiple obstacles changing course scenario II. SB-MPC collision avoidance using Euler's method and modified scenario grid II.

4.4.2 Discussion

The overall behavior is quite similar for the different scenario grids. It does however seem like a finer grid removes some of the unpredictability that is evident when using the original grid. This can be seen by comparing for instance figure 4.36 and 4.37. It should also be noted that in the situations where a speed change is applied by the SB-MPC using the original scenario grid, the speed change applied is larger when a finer scenario grid is used. This indicates that the scenario grid directly affects the behavior of the system and different tuning might be necessary for each set of control behaviors.

Chapter 5

Experimental Testing

Simulations are a good tool to test the performance and correctness of an algorithm, but they can not replace full-scale testing. In a simulator, simplifications must be made and this will cause inaccuracies in the simulations. The simulation results must therefor be verified by full-scale trials.

The SB-MPC was separated from its ROS framework and was with some minor structural changes implemented as a stand-alone library.

5.1 On-Board Simulator

As a preparation for the full-scale testing the SB-MPC scenarios was also run on Maritime Robotics' OBS. The obstacle vessels in the simulation are real vessels in the Trondheimsfjorden and their position is based on received AIS-data. As there were no possibility of constructing scenarios only an easily realizable subset of the previous simulations have been tested.

The limitations of the on-board computer made the version using a linear model the most adapted. This method allows the increase of the step-length used in the SB-MPC and thereby permits a longer prediction horizon than would be possible with the more computation heavy non-linear model.

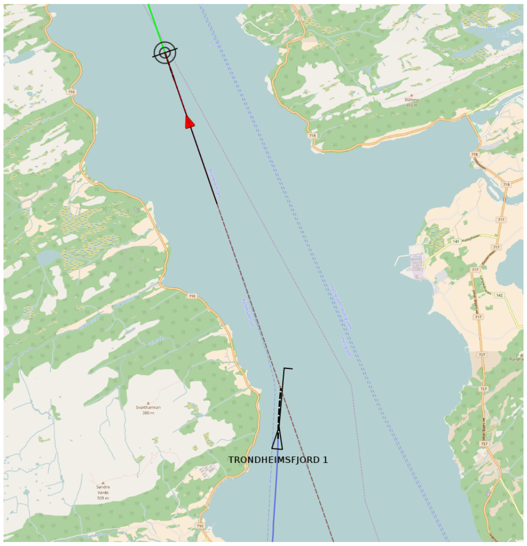

Figure 5.1: Screen-shot from OBS-simulator

Because of the limited time and resources available, path planning using waypoints was not implemented. The ASV is therefore running on a fixed heading and speed. The consequence of this is a slightly different behavior than in the previous simulations. In this section the ASV will not attempt to get back on track after an avoidance maneuver, it only changes back to the set heading. This also effects the course offsets. Instead of the step-wise increase seen in the plots in chapter 4 the course offsets now stay constant for the duration of the maneuver.

Table 5.1: Cost-function and simulation parameters

Parameter	Value	Unit
T	400.0	sec
dt	0.1	sec
p	1.0	
q	4.0	
d_{close}	300.0	m
d_{safe}	100.0	m
K_{coll}	3.0	
ϕ_{AH}	15.0	deg
ϕ_{OT}	68.5	deg
ϕ_{HO}	22.5	deg
ϕ_{CR}	68.5	deg
κ	3.0	
K_P	1.4	
K_χ	1.3	
$K_{\Delta P}$	2.3	
$K_{\Delta\chi,\text{starboard}}$	0.9	
$K_{\Delta\chi,\text{port}}$	1.2	

Head-on Scenario

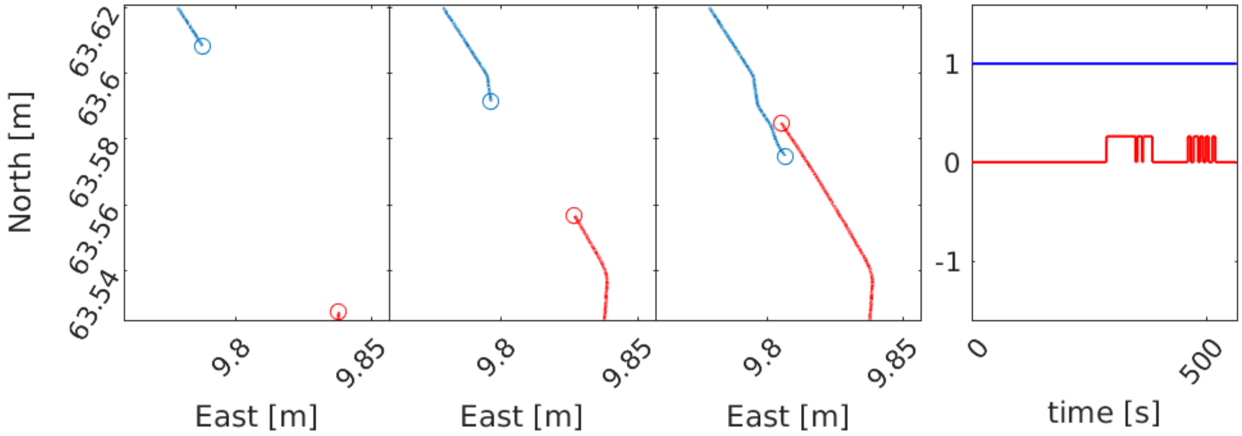

Figure 5.2: Vessel head-on

As can be seen in figure 5.3 the ASV turns starboard in good time and passes port of other vessel. The minimum distance between the two vessels is approximately 300 meters.

Vessel Crossing from Port Scenario

As in the ROS simulations the other vessel has no awareness of the ASV and even though the ASV has the right to stay on, it is forced to give-way when the other vessel fails to change its course.

Vessel Crossing from Starboard Scenario

The ASV, now in the position of the give-way vessel, again adheres to the COLREGS by turning starboard and passing behind the stay-on vessel. The minimum distance between the ships is in this case roughly 290 meters.

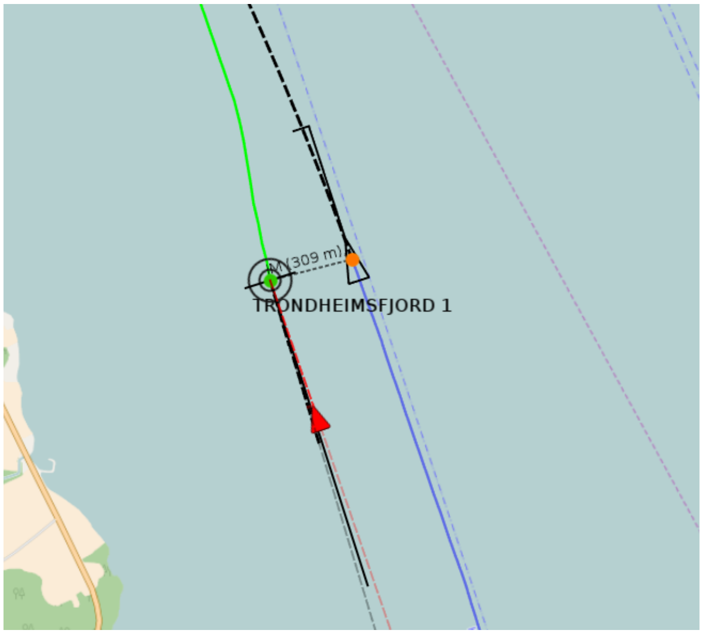

Figure 5.3: Screen-shot from OBS, head-on scenario. SB-MPC collision avoidance using linear prediction

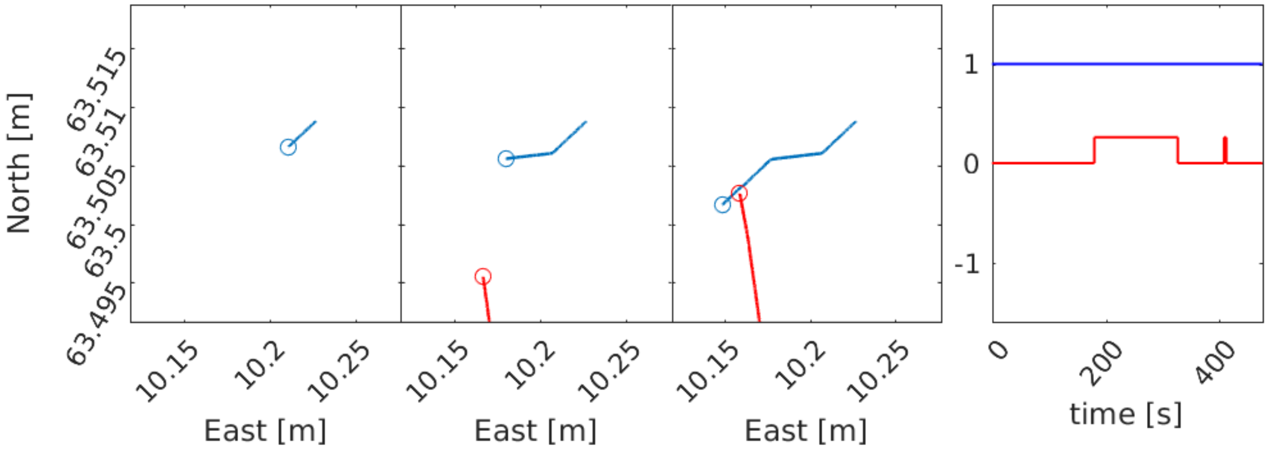

Figure 5.4: Vessel crossing from port scenario. SB-MPC collision avoidance using linear prediction

Overtaking a Vessel Scenario

In an overtaking scenario, the vessel doing the overtaking can pass on either side of the vessel being overtaken. In this cas the ASV passes port of the other vessel at a distance of approximately 180 meters. The relatively short

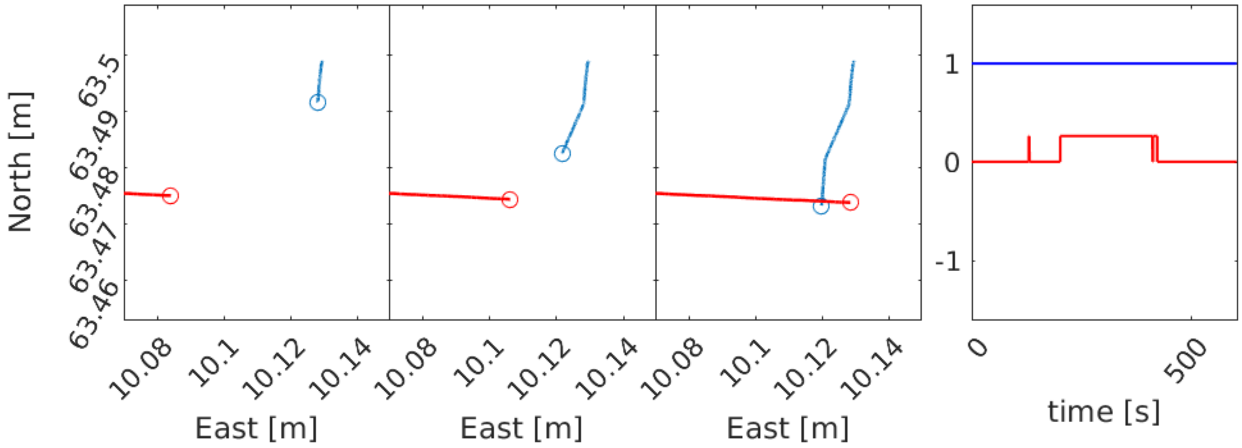

Figure 5.5: Vessel crossing from starboard scenario. SB-MPC collision avoidance using linear prediction

distance between the vessels is allowed because of the low relative speed between them.

5.2 Experimental Tests

The target test platform of the implementation was the *Telemetron*, a *Polar Circle 845 Sport*. This is a relatively small self-bailing Rigid Bouyancy Boat (RBB) with a V-shaped hull, making it both stable and highly maneuverable.

Parameter	Value	Units
Length	8.45	m
Width	2.50	m
Weight	~2000	kg
Power	225	hp
Max. Speed	~34	kn

The original plan involved using another of Maritime Robotics' vessels as the obstacle, but at the time of testing this vessel was not available. It was however decided to go through with the testing and seek out suitable situations with vessels already present in the fjord. This turned out to be very difficult.

The main challenges involved finding suitable obstacles and getting into a position that would allow testing the

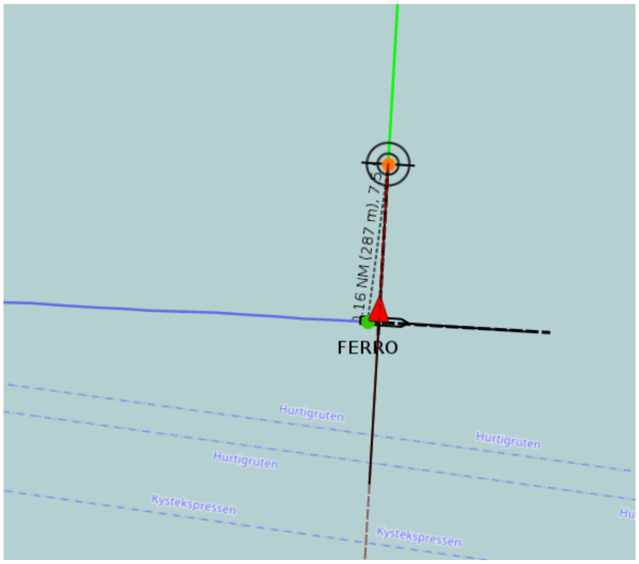

Figure 5.6: Screen-shot from OBS, vessel crossing from starboard scenario. SB-MPC collision avoidance using linear prediction

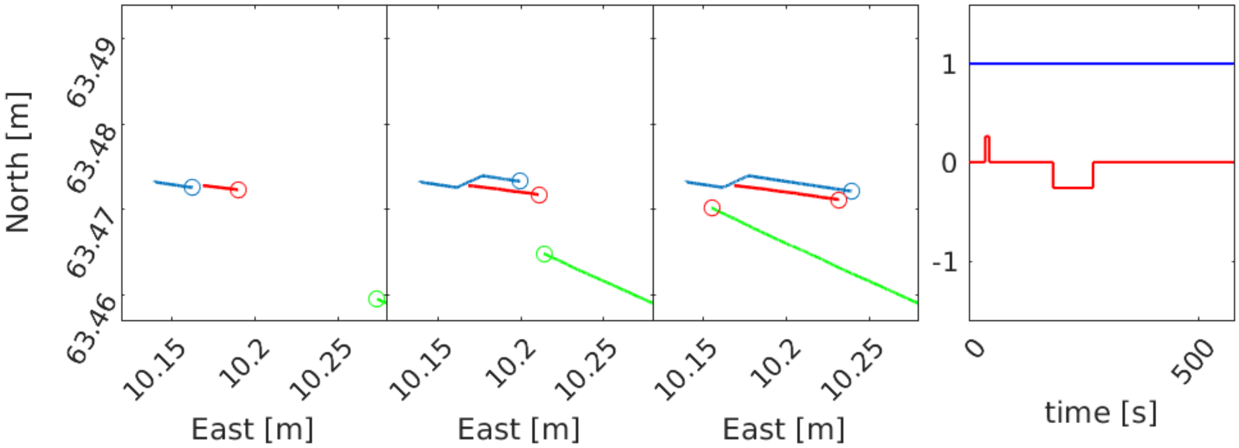

Figure 5.7: Overtaking a vessel scenario. SB-MPC collision avoidance using linear prediction

CAS in specific situations. Because of a sonar mounted on the Telemetron the maximum speed of the vessel was limited to 10 knots which made getting into position a challenge. The lack of suitable obstacle vessels also posed a big problem.

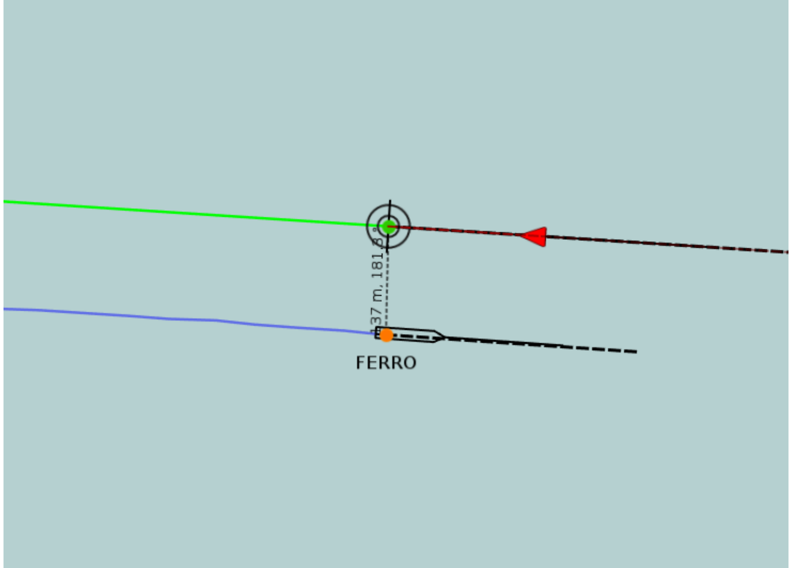

Figure 5.8: Screen-shot from OBS, overtaking a vessel scenario. SB-MPC collision avoidance using linear prediction

In the end none of the attempts of constructing the desired scenarios succeeded and time ran out. I would however say that the time spent was well used. Despite some technical problems the CAS is now a part of the Maritime Robotics' OBS and can easily be tested whenever the vessel is in use. As an alternative to test the system against other ships it was suggested to use a virtual vessel. This idea was originally discarded, but after trying to (not) collide with several large ships I would recommend this as a method that puts less stress on the operators. It should also be taken into consideration that it is not good conduct to purposely set out on a collision course with other vessels.

Figure 5.9: The vessel *Telemetron* was used as a test platform for the SB-MPC collision avoidance algorithm.

Chapter 6

Discussion

The simulation study (Chapter 4) shows that the SB-MPC performs well in a range of different situations, and avoids collision in all of the test cases. An issue that became evident was a unpredictability displayed in situations involving vessels crossing from starboard. As discussed in section 4.1, when a crossing vessel is approaching from starboard the ASV starts a collision avoidance maneuver to port before settling on a starboard course. This behavior can be limited by better tuning, but some consideration is needed to remove it completely.

It is in the more complex situations that the SB-MPC method shows it's advantages. Its ability to evaluate the future risk involved with a certain control behavior enables it to find a way out of very difficult situations. The simpler VO algorithm on the other hand, performs very well in the less complicated situations, but struggles in scenarios where multiple obstacles behave in a more random manner.

A change in course is easier for other vessels to perceive than a speed change. The tuning therefore prioritizes keeping nominal speed over keeping nominal course. The result is that in most of the scenarios simulated, collision is avoided by making a course change. It all comes down to the tuning. The method is both easy to understand and to implement, but tuning the system to achieve the wanted behavior is not straightforward and a systematic method for tuning has not been found.

The effect of different scenario grids was also tested and shows that the performance of the system is improved, but only slightly. Even though the results with the current tuning parameters were positive for all the tested grids, caution must be shown and separate tuning for different grids might be necessary.

To be able to do full-scale tests of specific scenarios it is necessary to have other vessels, real or virtual, available to pose as obstacle ships. This was not possible at the time of testing and even though an effort was made to test the SB-MPC using random boats on the fjord, it was not possible to complete the full-scale testing within the time available. However, the simulation results from the OBS, presented in chapter 5, are promising. In addition to this, the SB-MPC is now implemented on the Telemetron and full-scale tests can be preformed whenever the vessel is in use.

Chapter 7

Future Work

This implementation of a SB-MPC algorithm in this thesis is quite simple, but shows the basic workings of the method and some of the possibilities that lies within it. There are however many issues to be solved before it can be considered to be ready for use outside test-environments. Following are points that are considered important in the future development of the algorithm.

Points regarding the implementation itself:

- Make the speed of the ASV and obstacle influence the distance parameter that triggers COLREGS compliance, and the same for the parameter that decides the minimum safe distance between the vessels.
- Add weather scenarios that consider environmental forces that can influence the ASV's trajectory.
- Implement grounding avoidance.
- Optimize the code with regards to run-time.
- Improve integration technique for trajectory prediction.

Other issues for further study:

- Development of systematic methods for parameter tuning.
- The effect of different guidance methods on the behavior of the system.
- Investigation of the scale and effect of uncertainties in the measurements used.

Chapter 8

Conclusion

In this thesis a COLREGS compliant Collision Avoidance System using Simulation-Based Model Predictive Control for an ASV has been implemented. The system performed well during the simulated scenarios, avoiding the dynamic obstacles while adhering to the COLREGS.

The system was also compared with a VO based CAS. In simple scenarios involving only one obstacle their behaviors are very much alike, but in more complicated scenarios with multiple obstacles the SB-MPC shows a definite improvement. This result highlights one of the strengths of the method, namely its planning capability.

Disadvantages of the method are the need for an accurate model of the target vessel and the many tuning parameters. This necessitates modifying and testing the system for each target vessel. Finding methods for systematic parameter tuning should therefore be a priority.

Appendix A

Example code

The code in figure 4.1 shows a simple example of how to use the *sb_mpc* class. The state of the ASV is given by the vector $[x, y, u, v, r]$, and the five obstacles as $[x, y, u, v, A, B, C, D]$ where the last four elements are the dimensions of the obstacle as given by the AIS-data.

Figure A.1: Example use of the *sb_mpc* class

```cpp
#include "obstacle.h"
#include "sb_mpc.h"
#include "ship_model.h"

#include "Eigen/Dense"
#include "iostream"

int main(){

    // Guidance parameters
    double u_d = 9.34019;
    double psi_d = 1.71925;

    // Offsets
    double u_os;
    double psi_os;

    simulationBasedMpc *sb_mpc = new simulationBasedMpc();

    Eigen::Matrix<double, 6, 1> asv_state;
    asv_state << 0, 0, -0.0959975, 6.88277, 0, 0;

    Eigen::Matrix<double, 5, 9> obst_states;
    obst_states << 110.359,  146.154,  4.71239,        3,        0,       10,       10,       10,       10,
                  -826.865,  714.124,        0,        0,        0,       10,       10,       10,       10,
                  3444.27, -1950.35,  5.72293,  13.2727,        0,       10,       10,       10,       10,
                  -614.694,  735.825,  1.24093,  3.29244,        0,       10,       10,       10,       10,
                  -522.324,  6325.05,  2.27242,        0,        0,       10,       10,       10,       10;

    sb_mpc->getBestControlOffset(u_os, psi_os, u_d, psi_d, asv_state, obst_states);

    std::cout << "u_ os : " << u_os << std::endl;
    std::cout << "psi_os : " << psi_os << std::endl;

    return 0;
};
```

Appendix B

Implementation details

Stand-Alone Library

The implementation of the SB-MPC as a separate library can be found at GitHub:https://github.com/ingerbha/sb_mpc.git. The library contains three classes:

- **`sb_mpc`**: Contains the implementation of the SB-MPC algorithm.
- **`ship_model`**: Defines the vessel model used along with the integration methods available.
- **`obstacle`**: Defines the obstacle objects and the straight-line prediction of their trajectories.

This library can be used on its own and only requires that the `Eigen` library is available.

ROS Simulator with SB-MPC

The implementation of the system within the ROS framework can be found at GitHub:https://github.com/ingerbha/ros_asv_system.git. The contribution from this thesis is the `asv_ctrl_sb_mpc` package which contains the implementation of the SB-MPC algorithm. For a detailed installation guide and details on other packages see [18].

Bibliography

[1] S. Campbell, W. Naeem, and G.W. Irwin. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2):267–283, 2012.

[2] Samuel R. JR Clawson. Overtaking or crossing? Don't assume what other ship will do - Professional Mariner - August 2013, 2013.

[3] I. Eirkmen, A.M. Erkmen, F. Matsuno, R. Chatterjee, and T. Kamegawa. Snake robots to the rescue! *IEEE Robotics & Automation Magazine*, 9(3):17–25, sep 2002.

[4] Thor I. Fossen and Wiley InterScience (Online service). *Handbook of marine craft hydrodynamics and motion control.* Wiley, 2011.

[5] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, mar 1997.

[6] Joel M. Haight and Vladislav Kecojevic. Automation vs. Human intervention: What is the best fit for the best performance? *Process Safety Progress*, 24(1):45–51, mar 2005.

[7] IMO. COLREGS - International Regulations for Preventing Collisions at Sea. *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, pages 1–74, 1972.

[8] Tor Arne Johansen, Tristan Perez, and Andrea Cristofaro. Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3407–3422, dec 2016.

[9] Oussama Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *Autonomous Robot Vehicles*, pages 396–404. Springer New York, New York, NY, 1986.

[10] Steven M. Lavalle and Steven M. Lavalle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. 1998.

[11] Sang-Min Lee, Kyung-Yub Kwon, and Joongseon Joh. A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying COLREG Guidelines. *International Journal of Control, Automation, and Systems*, 2(2), 2004.

[12] Chris Llana and George Wisneskey. Handbook of the Nautical Rules of the Road, 2006.

[13] Ø. a. G. Loe. *Collision Avoidance for Unmanned Surface Vehicles*. PhD thesis, 2008.

[14] MUNIN. MUNIN – Maritime Unmanned Navigation through Intelligence in Networks, 2016.

[15] Ørnulf Jan Rødseth. Skip uten kaptein bak roret - SINTEF, 2014.

[16] Masoud Samadi and Mohd Fauzi Othman. Global Path Planning for Autonomous Mobile Robot Using Genetic Algorithm. In *2013 International Conference on Signal-Image Technology & Internet-Based Systems*, pages 726–730. IEEE, dec 2013.

[17] Thomas Statheros, Gareth Howells, and Klaus McDonald Maier. Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques. *Journal of Navigation*, 61(01):129–142, jan 2008.

[18] Thomas Stenersen. Guidance System for Autonomous Surface Vehicles. (June):1–101, 2015.

[19] Tore Stensvold. Verdens første førerløse passasjerferge kan gå over en kanal i Trondheim - Tu.no, 2016.

[20] Cheekuang Tam, Richard Bucknall, and Alistair Greig. Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. 2009.

[21] Wikipedia. International Regulations for Preventing Collisions at Sea - Wikipedia, the free encyclopedia, 2016.

[22] Wikipedia. Model Predictive Control - Wikipedia, The Free Encyclopedia, 2017.