Norwegian University of Science and Technology

Project report

Comparing Collision Avoidance Methods for ASVs: BC-MPC versus Velocity Obstacles

Anette Uttisrud

Trondheim, June 21, 2019

Supervisor:
Morten Breivik, NTNU

Co-supervisor:
Bjørn-Olav Holtung Eriksen, NTNU

Preface

This report is the final work of the course TTK4550 Engineering Cybernetics, Specialization Project. It is taken as a part of the final year of the master's degree programme in Cybernetics and Robotics at the Norwegian Univeristy of Science and Technology.

It has been a challenging year in many ways, and i have learned a lot working with this project. I would like to thank my supervisors for being patient, supportive and for providing good help.

Abstract

Collision avoidance (COLAV) systems is a critical and challenging part of an autonomous vehicle. This report focuses on the maneuvering part of COLAV for high-speed autonomous surface vehicles (ASVs).

COLAV algorithms can be divided into reactive and deliberate algorithms. The reactive are memoryless and consider only the current state, whereas the deliberate are more computationally heavy and capable of finding global solutions. Typically, algorithms can consider long term, mid-term and short-term COLAV. This project focuses on short-term COLAV, and the two algorithms branching-course model predictive control (BC-MPC) and velocity obstacle (VO). VO is a simple first-order method searching in the velocity space, and is popular in robotics and motion planning. BC-MPC is on the other hand a more advanced algorithm designed for marine vessels.

The methods and their perfomance for high-speed ASVs are evaluated focusing on their capabilities of COLAV, the compliance with the international regulations for preventing collisions at sea (COLREGs), and the robustness towards noise. In order to do this, a simulation environment considering the two algorithms and a model of the Telemetron ASV is implemented in MATLAB and SIMULINK.

Numerical simulations of several scenarios designed to test the capabilities of COLAV and COLREGs are performed. Further, the robustness towards measurement noise is evaluated through Monte Carlo simulations of selected scenarios. The results are evaluated qualitatively and quantitatively in terms of COLREGs compliance and a set of performance metrics. Both algoritms show COLAV capabilities. BC-MPC performs in general better, and outperforms VO in terms of COLREGs compliance and robustness towards noise.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Previous work . . . 2
- 1.3 Problem Formulation . . . 3
- 1.4 Contributions . . . 3
- 1.5 Outline . . . 3

2 Theory **5**
- 2.1 Vessel Modeling . . . 5
  - 2.1.1 Coordinate Frames . . . 5
  - 2.1.2 Modeling . . . 8
- 2.2 Guidance and Motion Control . . . 9
  - 2.2.1 Line of sight (LOS) Guidance . . . 9
  - 2.2.2 Path Following and Trajectory Tracking . . . 10
  - 2.2.3 Motion Control . . . 11
- 2.3 Configuration Spaces . . . 12
- 2.4 international regulations for preventing collisions at sea (COLREGs) 13
  - 2.4.1 The Rules . . . 13
  - 2.4.2 COLREGs Compliance for ASVs . . . 16
- 2.5 Collision Avoidance (COLAV) Methods . . . 17
  - 2.5.1 Reactive, Deliberate and Hybrid Methods . . . 18
  - 2.5.2 Branching-course MPC (BC-MPC) . . . 18
  - 2.5.3 Velocity Obstacles (VO) . . . 27

3 Simulator Implementation **35**
- 3.1 Simulator Overview . . . 35
- 3.2 Obstacles . . . 36
- 3.3 Measurement Noise . . . 36
- 3.4 Control and ASV model . . . 37
- 3.5 Implementation of BC-MPC . . . 38
- 3.6 Implementation of VO . . . 39

3.6.1 Algorithm Overview . . . 39
3.6.2 Velocity Search Space . . . 40
3.6.3 Dynamic Obstacles . . . 41
3.6.4 Static Obstacles . . . 42
3.6.5 Guidance . . . 43
3.6.6 Cost Function . . . 44
3.6.7 Controller Setpoints . . . 44

4 Simulation results **45**
4.1 Scenarios . . . 46
4.2 Performance Metrics . . . 47
4.3 Simulation Setup . . . 48
4.4 Simulations Without Noise . . . 50
4.4.1 Overtaking . . . 51
4.4.2 Head-on . . . 53
4.4.3 Crossing from Starboard . . . 54
4.4.4 Head-on and Crossing from Port . . . 56
4.4.5 Crossing from starboard and port . . . 57
4.4.6 Static obstacle scenario A . . . 59
4.4.7 Static obstacle scenario B . . . 60
4.4.8 Summary of Results Without Noise . . . 62
4.5 Simulations with Measurement Noise . . . 64
4.5.1 Head-on . . . 64
4.5.2 Crossing from Starboard . . . 66
4.5.3 Summary of Results with Noise . . . 70

5 Conclusion and Future Work **73**
5.1 Conclusion . . . 73
5.2 Future Work . . . 74

List of Tables

2.1 SNAME notation for marine vessels 7

4.1 Telemetron specifications and parameters. 48
4.2 Controller gains. 48
4.3 BC-MPC parameters. 49
4.4 VO parameters. 50
4.5 Noise parameters. 50
4.6 Simulation results, overtaking. 51
4.7 Simulation results, head-on 53
4.8 Simulation results, crossing from starboard. 55
4.9 Simulation results, head-on, and crossing from port. 57
4.10 Simulation results, crossing from starboard and crossing from port. 58
4.11 Simulation results, static scenario A. 60
4.12 Simulation results, static scenario B. 61
4.13 Simulation results, metrics. 62
4.14 Simulation results, COLREGs compliance. 63
4.15 Simulation results, head on with measurement noise. 64
4.16 Simulation results, crossing form starboard with measurement noise 67

List of Figures

2.1 Motion in 6 degree of freedom (DOF). 6
2.2 Line of sight 11
2.3 COLREGs situations 17
2.4 Predicted position trajectories. 19
2.5 Elliptical COLREGs regions and variables. 25
2.6 Velocity obstacle calculated by the Minowski sum. 29
2.7 Velocity obstacle with collapsed configuration space. 30

3.1 Simulator overview 36
3.2 Overview of VO implementation 38
3.3 Overview of VO implementation 39
3.4 Velcocity search space for VO 41
3.5 Velocity obstacle for a satic obstacle 43

4.1 Simulations of overtaking scenario 51
4.2 Overtaking scenario, speed and course 52
4.3 Overtaking scenario, course rate 53
4.4 Simulations of head-on scenario 54
4.5 Simulations of crossing from starboard scenario 55
4.6 Simulations of head-on and crossing from port scenario 56
4.7 Simulations of crossing from starboard and crossing from port scenario 58
4.8 Simulations of static scenario A 59
4.9 Simulations of static scenario B 61
4.10 Simulation results of head-on with measurement noise 65
4.11 Simulation results of crossing from starboard with measurement noise 68
4.12 Crossing from starboard results: integral of absolute course rate (IACR)(35) and integral of absolute speed rate (IASR)(35) 70

Abbreviations

ASV autonomous surface vehicle.

BC-MPC branching-course MPC.

COLAV collision avoidance.

COLREGs international regulations for preventing collisions at sea.

CPA closest point of approach.

DOF degree of freedom.

DRVO dynamic reciprocal velocity obstacle.

DW dynamic window.

FF-FB-C feed-forward-feedback-course.

IACR integral of absolute course rate.

IASR integral of absolute speed rate.

IMO International Maritime Organization.

LOS line of sight.

MDO minimum distance to obstacle.

MPC model predictive control.

NED north-east-down.

RRT rapidly exploring random tree.

RVO reciprocal velocity obstacle.

SNAME Society of Naval Architects and Marine Engineers.

SOG speed over ground.

TD travel distance.

TT travel time.

UAV unmanned autonomous vehicle.

USV unmanned surface vehicle.

VO velocity obstacle.

Chapter 1

Introduction

1.1 Motivation

A collision avoidance (COLAV) system is critical for autonomous surface vehicles (ASVs). The system should be able to avoid collisions with obstacles in its surroundings. Typically reactive characteristics are required to avoid suddenly moving or late detected obstacles. However, avoiding collisions is not enough for a satisfying behavior. Marine vessels should act in a safe manner preventing such situations to occur. The international regulations for preventing collisions at sea (COLREGs) states what is considered as 'the rules of the road' at sea, and marine vessels should act in compliance with these rules.

In order to follow the rules, the COLAV system needssome situation awareness, and should plan its maneuvers ahead of time in accordance to the surrounding environment. An important part of COLREGs is to maneuver in an observable manner and in ample time. In addition, vessels are typically heading towards a goal. In order to find global soltuions such that the reach of this goal can be guaranteed, the cabpability to compute and emphasize the available information is important. Hence, a well-functioning COLAV sytem has many considerations to make, and should be able to both plan the future and act in real time. However, such a system and algorithms relies on accurate information of its surroundings. Operating at sea, there can be many disturbances and sources to noise. It is therefore important that a COLAV systems is robust towards noise.

In other words, there are many challenges in implementing a safe and reliable COLAV system for ASVs. In this project, the focus has been to evaluate and compare short term COLAV methods that acts in compliance with COLREGs.

1.2 Previous work

COLAV algorithms range from reactive algorithms considering only the current state to deliberate algorithms utilizing more information and computation in order to give global solutions.

Examples of reactive algorithms are dynamic window (DW) [1] and velocity obstacle (VO) [2], which both search in the velocity space. DW was introduced in 1997 as a collision avoidance method for ground vehicles [1], and is modified and applied for unmanned autonomous vehicles (UAVs) [3] and ASVs [4]. VO was introduced in 1998 [2], as has been applied in different ways with several extensions. Examples are the application to unmanned surface vehicles (USVs) considering COLREGs [5] [6], and reciprocal velocity obstacle (RVO) [7] which takes active actions of other obstacles into account. Dynamic reciprocal velocity obstacle (DRVO) [8] develops the method further, and considers RVO for ASVs taking COLREGs into account.

Examples of reactive directional algorithms are potential field methods.The artificial potential field method treats the robot as a paritcle moving in a artificial force field [9]. The method is developed and improved [10] [11] [12], also for marine vessels considering COLREGs[13].

Examples of deliberate algorithms are rapidly exploring random tree (RRT) [14], $A*$ [15] and model predictive control (MPC). RRT are suited for various planning problems, and is designed to handle dynamics and high degrees of freedom. $A*$ is a heuristic informed algorithm, typically used for path finding. Model predictive control (MPC) permits the use of dynamic models of the system to optimize a finite time-horizon while satisfying a set of constraints. It is used within a wide range of motion control, also for marine vessels [16] [17]. The branching-course MPC (BC-MPC) algorithm is design for marine vessels to be robust towards noisy estimates of obstacles keeping the typical marine maneuvers and COLREGs in mind [18].

1.3 Problem Formulation

The following tasks are given for the project report

1. Get familiar with COLAV algorithms relevant for ASVs, focusing on VO and branching-course MPC (BC-MPC).
2. Implement a simulation framework in MATLAB/SIMULINK that simulates the Telemetron ASV with BC-MPC and VO as COLAV algorithms. The implementation should consider COLREGs.
3. Perform numerical simulations to evaluate and compare BC-MPC with VO in different scenarios. The evaluation should consider COLREGs compliance, quantitative performance metrics and robustness towards measurement noise.

1.4 Contributions

The contributions of the work with this project are

- Implementation of a simulator in MATLAB/SIMULINK for simulation of ASV COLAV considering COLREGs.
- Numerical simulations of several scenarios, both with and without noise. The performance of BC-MPC and VO are evaluated qualitatively and quantitatively based on COLREGs compliance and a set of performance metrics.

1.5 Outline

Chapter 2 presents necessary background theory within vessel modelling, guidance, control and COLAV. Chapter 3 explains the implementation of the simulation environment and the algorithms. Chapter 4 introduce a set of scenarios, which considers both dynamic and static obstacles in both simple and more complex scenarios. Then, the simulation setup are explained and the performance metrics defined. The simulation results of all the scenarios are presented and evaluated. Last, two of the scenarios are simulated with measurement noise as

Monte Carlo simulations. A conclusion and suggestions for further work follow in Chapter 5.

Chapter 2

Theory

In this chapter, the background theory used in the implementation and simulation is presented. Section 2.1 starts with an introduction to coordinate frames, rotations between them and the commonly used notation by Society of Naval Architects and Marine Engineers (SNAME) for marine vessels. General vectorial models of vessels in six and three degree of freedom (DOF) are then presented before introducing a model-based, two DOF model for high-speed ASVs. Section 2.2 explaines line of sight (LOS) guidance, path following and trajectory tracking before a model-based controller is presented. The configuration space of a vessel is explained in Section 2.3. Section 2.4 explaines the COLREGs rules relevant for this report. Finally, Section 2.5 discusses different types of COLAV methods before BC-MPC and VO are explained in detail.

2.1 Vessel Modeling

2.1.1 Coordinate Frames

In order to describe the pose, or the position and orientation, of a vessel, coordinate frames are used.

The north-east-down (NED) frame $\{n\} = (x_n, y_n, z_n)$ is rotated such that x_n points towards true north, y_n points towards true east and z_n points downwards

normal to the Earth's surface. A NED-frame fixed to a point on the Earth's surface is often used as the reference frame for local navigation. The NED-frame is then assumed to be inertial such that Newton's laws apply, and the pose of the vessel is given relative to this frame.

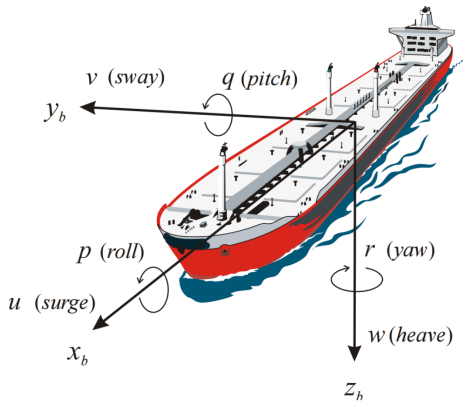


Figure 2.1: Motion in 6 degree of freedom (DOF) [19].

The body frame $\{b\} = (x_b, y_b, z_b)$ is fixed to the craft and is useful to describe velocities and forces acting on the ship. The origin o_b is usually located midships in the waterline. The x_b axis points forward in the longitudinal direction, y_b is the transversal axis pointing towards starboard and z_b is directed downwards and completes the right-handed system.

For vessels moving in six DOF, the pose is described by the and orientation along and about the three axes. The translational and rotational motion is described by the time derivatives of the pose, as illustrated in Figure 2.1. The notation of the Society of Naval Architects and Marine Engineers (SNAME) [20] listed in Table 2.1 is commonly used to denote these values for marine vessels.

Table 2.1: SNAME notation for marine vessels

Motion	**Positions and Euler angles**	**Linear and angular velocities**	**Forces and moments**
surge	x	u	X
sway	y	v	Y
heave	z	w	Z
roll	ϕ	p	K
pitch	θ	q	M
yaw	ψ	r	N

The horizontal speed, speed over ground (SOG), is defined by the surge speed u and the sway speed v as

$$U = \sqrt{u^2 + v^2}. \tag{2.1}$$

For the rest of this report, speed over the ground is denoted as speed. The course χ denotes the direction of the speed and is defined as

$$\chi = \psi + \beta, \tag{2.2}$$

where ψ is the yaw or heading of the vessel describing its orientation, and β the sideslip. This is illustrated in Figure 2.2.

The pose of the vessel is described in the inertial frame, typically NED, and the linear and angular velocities are usually described in BODY. A transformation between the two frames is necessary in order to describe the velocities in NED and can be carried out by a rotation

$$\boldsymbol{v^n} = \boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}})\boldsymbol{v^b} \tag{2.3}$$

where $\boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}})$ is a combination of three principal rotations, one about each axis,

$$\boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi}. \tag{2.4}$$

The reverse transformation is the inverse or the transposed rotation matrix

$$\boldsymbol{R}_{\boldsymbol{b}}^{\boldsymbol{n}}(\boldsymbol{\Theta}_{\boldsymbol{nb}})^{-1} = \boldsymbol{R}_{\boldsymbol{b}}^{\boldsymbol{n}\top}(\boldsymbol{\Theta}_{\boldsymbol{nb}}) = \boldsymbol{R}_{\boldsymbol{n}}^{\boldsymbol{b}}(\boldsymbol{\Theta}_{\boldsymbol{nb}}) = \boldsymbol{R}_{z,\psi}^{\top}\boldsymbol{R}_{y,\theta}^{\top}\boldsymbol{R}_{x,\phi}^{\top}. \tag{2.5}$$

2.1.2 Modeling

The equations of motion in six DOF can be written on a vectorial form

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.6}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.7}$$

where $\boldsymbol{\nu} = [u, v, w, p, q, r]^\top$ is the velocity vector, $\boldsymbol{\tau} = [X, Y, Z, K, M, N]^\top$ is the force vector [19]. The external forces from wind and waves are denoted as $\boldsymbol{\tau}_{wind}$ and $\boldsymbol{\tau}_{wave}$, and the inertia, coriolis and damping matrices as $\boldsymbol{M}, \boldsymbol{C}(\boldsymbol{\nu})$ and $\boldsymbol{D}(\boldsymbol{\nu})$, respectively.

The model is commonly reduced to a three DOF model neglecting heave, roll, and pitch,

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.8}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{2.9}$$

where $\boldsymbol{\eta} = [N, E, \psi]^\top$, $\boldsymbol{\nu} = [u, v, r]^\top$, $\boldsymbol{\tau} = [X, Y, N]^\top$ is the vessel pose in $\{n\}$, velocity in $\{b\}$ and force in $\{b\}$, respectively.

Vessels are carried by the hydrostatic (restoring forces) and hydrodynamic (added mass and damping) pressure, which dominates at different speeds. The hydrostatic forces dominate at lower speeds when the vessel is in the displacement region. For higher speeds when the hydrodynamic forces dominate, the vessel is in the planning region.

The vessel considered in this report is the dual-use Telemetron ASV, which accelerates to higher speeds up to 18 m/s. Due to the high speed and small size of the Telemetron ASV, it operates in displacement, semi-displacement and planning regions [21]. The model (2.9) is best suited for vessels operating in displacement

[19], and is therefore not optimal for modeling the Telemetron ASV. An alternative 2DOF control-oriented model for high-speed ASVs is therefore proposed [21]. The vessel kinematics is given by

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \cos(\chi) & 0 \\ \sin(\chi) & 0 \\ 0 & 1 \end{bmatrix} \tag{2.10}$$

$$\dot{\chi} = r + \dot{\beta}, \tag{2.11}$$

where β is the sideslip and $\chi = \psi + \beta$ is the course. Definig the state vector as $\boldsymbol{x} = [U, r]^\top$, the dynamcis is given by

$$\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{\tau}. \tag{2.12}$$

where $\boldsymbol{M}(\boldsymbol{x}) = \text{diag}\left(m_U(\boldsymbol{x}), m_r(\boldsymbol{x})\right)$ is the inertia matrix and $\boldsymbol{\sigma}(\boldsymbol{x}) = \left[\sigma_U(\boldsymbol{x}), \sigma_r(\boldsymbol{x})\right]^\top$ is the damping term. $\boldsymbol{\tau} = [\tau_m, \tau_\delta]$ is the torque vector, where $\tau_m \in [0, 1]$ is the motor torque controlling the speed and $\tau_\delta \in [-1, 1]$ the rudder torque controlling the course. Note that with the controlled states U and r, the model is suited for underactuaded ASVs which can not control surge, sway and yaw independently.

The inertia and damping terms are identified from a series of experiments with the Telemetron ASV [21].

2.2 Guidance and Motion Control

A guidance system is used to calculate the desired states, usually course and speed, for a moving vessel required to follow a target or a desired path or trajectory. The motion controller controls the actuators in order for the vessel states to track the reference.

2.2.1 LOS Guidance

LOS guidance is a three-point scheme. It typically takes a stationary reference point $\boldsymbol{p}_0^n = \left[N_0, E_0\right]$, a target $\boldsymbol{p}_t^n = \left[N_t, E_t\right]$ and the interceptor with the position $\boldsymbol{p}^n(t)$. The control objective can be formulated as

$$\lim_{t\to\infty}[\boldsymbol{p}^n(t) - \boldsymbol{p}_t^n(t)] = \mathbf{0} \tag{2.13}$$

[22]. The goal for the interceptor is to intercept the path from the stationary reference point to the target. This is done along the LOS vector from the interceptor to a point on the path depending on the lookahead distance, which is illustrated in Figure 2.2.

For lookahead-based steering, the desired course is calculated based on the cross-track error e and the lookahead-distance $\Delta(t) > 0$, which decides how fast the convergence towards the LOS vector should be. The desired yaw angle χ_d is calculated as

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

$\chi_p = \alpha_k$ is the path-tangential angle and $\chi_r(e)$ is the velocity-path relative angle given by

$$\chi_r(e) := \arctan\Big(-\frac{e}{\Delta}\Big). \tag{2.15}$$

2.2.2 Path Following and Trajectory Tracking

The main difference between trajectory tracking and path following is the consideration of time. A trajectory tracking control system considers time, and forces the system output $\boldsymbol{y}(t)$ to track the trajectory $\boldsymbol{y}_d(t)$ [19].

A path is time-invariant, and a piecewise linear path is typically defined by n waypoints $\boldsymbol{w} = \big[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n\big]$ where $\boldsymbol{w}_i = \big[N_i, E_i\big]$.

For path following of a waypoint generated path, the stationary reference point can be set to the initial waypoint $\boldsymbol{p}_k = \big[N_k, E_k\big]$, and the target to its successor, $\boldsymbol{p}_{k+1} = \big[N_{k+1}, E_{k+1}\big]$. For paths with more than two waypoints, a switching criterion to decide which two points to consider is needed. An example of this is the radius of acceptance: If the interceptor is within a certain radius of the target, the target is considered reached. The stationary reference point can then be reinitialized to the previous target, with its successor initialized as the next target [19].

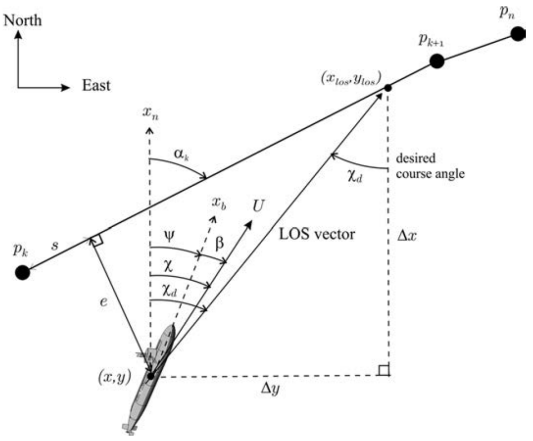


Figure 2.2: Line of sight[19]

2.2.3 Motion Control

A model-based controller based on the Telemetron model (2.10) - (2.12) is presented in this chapter. It combines proportional-integral feedback with model-based feedforward to increase the closed-loop performance [21], and is augmented by including the course as a state to the controller named feed-forward-feedback-course (FF-FB-C) [4]. This allows a state to be tracked, and the controller has shown good perfomance in experiments [4]. The controller is given by

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{x}}_d + \boldsymbol{\sigma}(\boldsymbol{x}_d) - \boldsymbol{M}(\boldsymbol{x})\boldsymbol{K}_p\tilde{\boldsymbol{\zeta}} - \boldsymbol{K}_i \int_{t_0}^{t} \tilde{\boldsymbol{\zeta}}_1(\gamma)d\gamma, \tag{2.16}$$

where $\tilde{\boldsymbol{\zeta}}$ and $\tilde{\boldsymbol{\zeta}}_1$ are the error terms

$$\begin{aligned} \tilde{\boldsymbol{\zeta}} &= \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_d \\ \Upsilon(\chi - \chi_d) \end{bmatrix} = \begin{bmatrix} \tilde{U} \\ \tilde{r} \\ \tilde{\chi} \end{bmatrix} \\ \tilde{\boldsymbol{\zeta}}_1 &= \begin{bmatrix} \tilde{U} \\ \tilde{\chi} \end{bmatrix}, \end{aligned} \tag{2.17}$$

where Υ maps to $[-\pi, \pi)$. The matrices $\boldsymbol{K}_p > 0$ and $\boldsymbol{K}_i > 0$ are the proportional and integral gains given as

$$\begin{aligned} \boldsymbol{K}_p &= \begin{bmatrix} k_{p_U} & 0 & 0 \\ 0 & k_{p_r} & k_{p_{chi}} \end{bmatrix} \\ \boldsymbol{K}_i &= \begin{bmatrix} k_{i_U} & 0 \\ 0 & k_{i_{chi}} \end{bmatrix}. \end{aligned} \tag{2.18}$$

With the course defined as $\chi = \psi + \beta$, the sideslip β has to be considered. From experiments, it is seen that the sideslip is slowly varying at moderate speed [4]. Assuming $\dot{\beta} = 0$, the following relation between the desired course and desired rate of turn can be made

$$\begin{aligned} r_d &= \dot{\chi}_d \\ \dot{r}_d &= \ddot{\chi}_d. \end{aligned} \tag{2.19}$$

2.3 Configuration Spaces

When working with motion planning and collision avoidance, it is useful to have an interface towards the environment. Let $\mathcal{W} \in \mathbb{R}^2$ be the two-dimensional workspace denoting the world. For a surface vessel with a configuration defined as $q = \boldsymbol{\eta}$, the configuration space $\mathcal{C} \subset \mathcal{W}$ is the set of all possible configurations q, or the set of all rigid-body transformations, that can be applied to the vessel [23]. Let $\mathcal{A}(q) \subset \mathcal{W}$ be the footprint, or the set of points occupied in $\mathcal{W}$ by the vessel with configuration q. Similarly, let $\mathcal{O} = \cap \mathcal{O}_i \subset \mathcal{W}$ be all points occupied by obstacles. The free space is then the set of all configurations that do not collide

$$\mathcal{C}_{free} = \{q \in \mathcal{C} | \mathcal{A}(q) \cap \mathcal{O} = \emptyset\}, \tag{2.20}$$

and the obstacle region is the compliment

$$\mathcal{C}_{obs} = \mathcal{C} \setminus \mathcal{C}_{free}. \tag{2.21}$$

2.4 international regulations for preventing collisions at sea (COLREGs)

The international regulations for preventing collisions at sea (COLREGs) were published by International Maritime Organization (IMO) in 1972 and acts as the rules of the road at sea [24]. They establish navigation rules for vessels at sea to prevent collisions and consider both general behavior and specific obligations on how to act in different situations.

The rules that are of the most relevance for this project are the ones that consider maneuvering. They are from part B, Steering and Sailing. Rule 8 considers actions to avoid collision. Rule 16 and 17 actions by keep-way and stand-on vessels. Rule 13, 14, 15 considers actions in respectively overtaking, head-on and crossing situations.

2.4.1 The Rules

Rule 8: Action to Avoid Collision

Rule 8 considers the general obligations for vessels to avoid a collision, including acting in observable manners due to good seamanship.

Rule 8 (a). *Any action to avoid collision shall be taken in accordance with the Rules of this Part and shall, if the circumstances of the case admit, be positive, made in ample time and with due regard to the observance of good seamanship.*

Rule 8 (b). *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.*

Rule 8 (c). *If there is sufficient sea room, alteration of course alone may be the most effective action to avoid a close-quarters situation provided that it is made in good time, is substantial and does not result in another close-quarters situation.*

Rule 8 (d). *Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance. The effectiveness of the action shall be carefully checked until the other vessel is finally past and clear.*

Rule 8 (e). *If necessary to avoid a collision or allow more time to assess the situation, a vessel shall slacken her speed or take all way off by stopping or reversing her means of propulsion.*

(i).*A vessel which, by any of these Rules, is required not to impede the passage or safe passage of another vessel shall, when required by the circumstances of the case, take early action to allow sufficient sea-room for the safe passage of the other vessel.*

(ii). *A vessel required not to impede the passage or safe passage of another vessel is not relieved of this obligation if approaching the other vessel so as to involve risk of collision and shall, when taking action, have full regard to the action which may be required by the Rules of this part.*

(iii). *A vessel the passage of which is not to be impeded remains fully obliged to comply with the Rules of this part when the two vessels are approaching one another so as to involve risk of collision.*

Rule 13: Overtaking

Rule 13 defines when an overtaking situation occurs and the obligations for the overtaking vessel

Rule 13 (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.*

Rule 13 (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.*

Rule 13 (c). *When a vessel is in any doubt as to whether she is overtaking another, she shall assume that this is the case and act accordingly.*

Rule 13 (d). *Any subsequent alteration of the bearing between the two vessels shall not make the overtaking vessel a crossing vessel within the meaning of these Rules or relieve her of the duty of keeping clear of the overtaken vessel until she is finally past and clear.*

Rule 14: Head-on Situations

Rule 14 defines when a head on situation occurs, and the obligations for the involved vessels

Rule 14 (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.*

Rule 14 (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 other vessel.*

Rule 14 (c). *When a vessel is in any doubt as to whether such a situation exists she shall assume that it does exist and act accordingly.*

Rule 15: Crossing Situation

Rule 15. *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.*

Rule 16: Action By Keep-way Vessel

Rule 16. *Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.*

Rule 17: Action By Stand-On Vessel

Rule 17 defines the obligations for the stand on vessel

Rule 17 (a).

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

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

Rule 17 (b). *When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid a collision.*

Rule 17 (c). *A power-driven vessel which takes action in a crossing situation in accordance with subparagraph (a)(ii) of this Rule to avoid collision with another power-driven vessel shall, if the circumstances of the case admit, not alter course to port for a vessel on her own port side.*

Rule 17 (d). *This Rule does not relieve the give-way vessel of her obligation to keep out of the way.*

2.4.2 COLREGs Compliance for ASVs

In some situations, it might be required to take maneuvers that are in conflict with some COLREGs rules in order to obtain good seamanship. This makes it complicated to measure and claim whether a COLAV algorithm is COLREGs compliant or not. There are also some rules that consider signals, lights and other aspects that are not directly related to maneuvering. Therefore, it is in most cases more relevant to look at COLREGs compliance for ASVs with respect to a set of rules.

It can be useful to identify the COLREGs situation. Consider two vessels A and B, and let the relative bearing be

$$\beta = \Upsilon\left(\arctan\left(E_A - E_B, N_A - N_B\right) - \psi_B\right), \tag{2.22}$$

where $\boldsymbol{p_A} = [N_A, E_A]$ is the position of vessel A, $\boldsymbol{p}_B = [N_B, E_B]$ is the position of vessel B and ψ_B the heading of vessel B. The rule identified by vessel A relative to vessel B is then

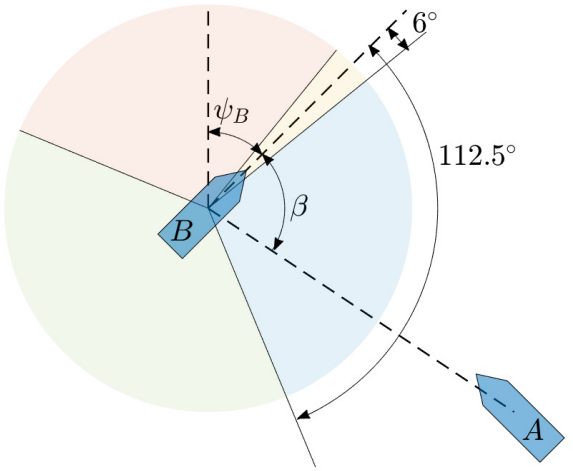


Figure 2.3: COLREGs situations.

$$\text{Rule} = \begin{cases} \text{Head-on} & \text{if} & -6 \leq \beta < 6 \\ \text{Crossing from port} & \text{else if} & 6 \leq \beta < 112.5 \\ \text{Overtake} & \text{else if} & 112.5 \leq \beta < 180 \cup -180 \leq \beta < -112.5 \\ \text{Crossing from starboard} & \text{else if} & -112.5 \leq \beta < -6. \end{cases} \tag{2.23}$$

An illustration is given in Figure 2.3. The angle of overtaking, 112.5°, is specified in Rule 13. The angle of head-on varies in different applications, but an angle of 6° is typically used. Note that this only identifies the rule based on the realtive bearing. To decide if a collision situation occurs, other factors must also be evaluated [25]. For example, if vessel A in Figure 2.3 had the opposite orientation, there would be no collision situation even with a short distance between the vessels.

2.5 Collision Avoidance (COLAV) Methods

Where guidance considers the vessel state and its desired path, and solves the path following problem, COLAV methods considers the current environment with its surrounding objects and their states. However, what information that is considered and how it is handled differs between methods. This section gives a brief

discussion of different algorithms, before the BC-MPC and the VO algorithms are explained in detail.

2.5.1 Reactive, Deliberate and Hybrid Methods

COLAV methods can be divided into reactive and deliberate methods. Reactive COLAV methods have limited information and are memoryless. They act based on the current state and environment only. This gives low computationally cost and allows fast algorithms. Such algorithms are therefore suitable for short term local planning, typically avoiding collisions with suddenly detected or maneuvering obstacles. There is however a risk that the reactive methods find the local optimal solution that is a dead end in the global environment. Global convergence towards the goal can therefore not be guaranteed with reactive methods.

Deliberate methods consider more information and are more suitable to find global solutions. However, a consequence of this is usually a more computationally heavy algorithm.

A combination of deliberate and reactive methods i a hybrid architecture are usually required for satisfying path planning and collision avoidance.

2.5.2 Branching-course MPC (BC-MPC)

The branching-course MPC (BC-MPC) algorithm is based on model predictive control (MPC), where a finite time horizon is optimized based on the model of the system while respecting constraints. The first step is then executed before the algorithm replans.

BC-MPC generates a discrete search space of trajectories based on dynamic constraints. The trajectories are calculated from a sequence of possible maneuvers and forms a tree of the same depth as the number of considered maneuvers. For each trajectory or branch, an objective function is calculated and minimized, such that the trajectory with the minimum cost is chosen. The objective function considers the distance to dynamic and static obstacles, trajectory tracking and translational cost of the speed and course maneuver.

The BC-MPC algorithm was proposed by Eriksen [18], and is rendered in this section. First, the iterative trajectory generation and its steps are explained. Then, the objective function and its terms are explained.

Trajectory Generation

The discrete search space of trajectories forms a tree, were each trajectory is defined by a predicted velocity trajectory $\bar{\boldsymbol{u}}_d(t) = \left[\bar{U}_d(t), \bar{\chi}_d(t)\right]^\top$ and the predicted pose trajectory $\bar{\boldsymbol{\eta}}(t) = \left[\bar{N}(t), \bar{E}(t), \bar{\chi}(t)\right]^\top$. The tree is generated level by level, initialized with the initial state as the root node. Subtrajectories are generated from the root node, each representing one maneuver. Leaf nodes are then initialized at the end of the subtrajectories, and the process is repeated until the desired depth is reached. An example of a tree of predicted position trajectories is shown in Figure 2.4.

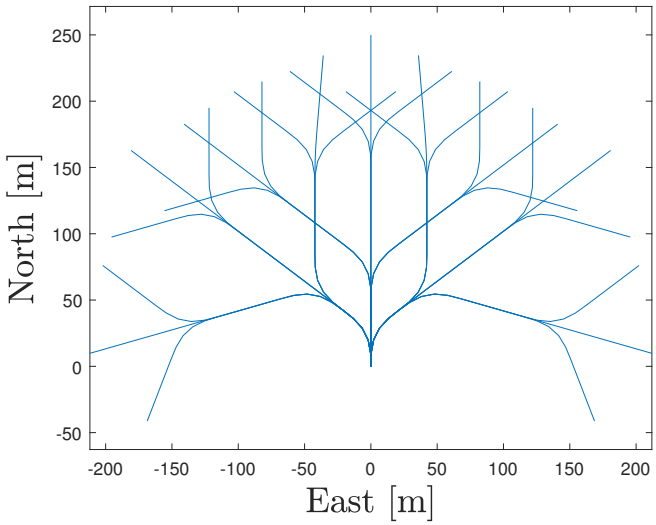


Figure 2.4: An example of a tree of predicted position trajectories. The number of speed maneuvers considered for each level is $\boldsymbol{N_U} = [1, 1, 1]$, and the number of course maneuvers $\boldsymbol{N}_\chi = [3, 3, 3]$. This results in a tree width depth 3 and 27 branches.

To limit the growth of the tree, a limited number $\boldsymbol{N}_U = [N_{U_1}, N_{U_2}, \ldots, N_{U_B}]$ of speed maneuvers and $\boldsymbol{N}_\chi = [N_{\chi_1}, N_{\chi_2}, \ldots, N_{\chi_B}]$ course maneuvers are considered for each of the B levels. The prediction time at each level is given by $\boldsymbol{T} = [T_1, T_2, \ldots, T_B]$, and the prediction time for the full trajecoty is then $T_{full} = \sum_{i=1}^{B} T_i$.

For the generation of each sub-trajectory, a set of acceleration samples A_d and

motion primitives $\dot{U}_{d,i}(t)$ and $\dot{r}_{d,i}(t)$ representing the possible maneuvers are calculated based on the control input constraints. Desired trajectories for speed, course and course rate are then generated by integration and used to form a set of desired velocity trajectories $\mathcal{U}$. A first-order error model is then used for feedback-correction of the trajectories and forms a set of predicted velocity trajectories $\bar{\mathcal{U}}$. Finally, the predicted velocity trajectories are integrated to form the set of feedback-corrected pose trajectories $\bar{\mathcal{H}}$.

Motion Primitives

The minimum and maximum possible control inputs $\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}$ are calculated as

$$\begin{aligned} \boldsymbol{\tau}_{min} &= \text{sat}(\boldsymbol{\tau}_0 + T_{ramp}\dot{\boldsymbol{\tau}}_{min}, \boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}) \\ \boldsymbol{\tau}_{max} &= \text{sat}(\boldsymbol{\tau}_0 + T_{ramp}\dot{\boldsymbol{\tau}}_{max}, \boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}), \end{aligned} \tag{2.24}$$

where T_{ramp} is the ramp time, $\boldsymbol{\tau}_0$ the current control input and $\dot{\boldsymbol{\tau}}_{min}, \dot{\boldsymbol{\tau}}_{max}$ the minimum and maximum control input rate of change. The saturation function $\text{sat}(\boldsymbol{a}, \boldsymbol{a}_{min}, \boldsymbol{a}_{max})$ saturates $\boldsymbol{a}$ such that all elements a_i satisfies $a_{min_i} \leq a_i \leq a_{max_i}$.

The minimum and maximum possible accelarations $\dot{\boldsymbol{X}}_{min} = \left[\dot{U}_{min}, \dot{r}_{min}\right], \dot{\boldsymbol{X}}_{max} = \left[\dot{U}_{max}, \dot{r}_{max}\right]$ are then calculated from the vessel model considering the current velocity $\boldsymbol{X}_0$:

$$\begin{aligned} \dot{\boldsymbol{X}}_{min} &= \boldsymbol{M}^{-1}(\boldsymbol{\tau}_{min} - \boldsymbol{\sigma}(\boldsymbol{X}_0)) \\ \dot{\boldsymbol{X}}_{max} &= \boldsymbol{M}^{-1}(\boldsymbol{\tau}_{max} - \boldsymbol{\sigma}(\boldsymbol{X}_0)). \end{aligned} \tag{2.25}$$

The set of possible accelerations can then be created as

$$A_d = \left\{ \left(\dot{U}, \dot{r}\right) \in \mathbb{R} \times \mathbb{R} \middle| \dot{U} \in \left[\dot{U}_{min}, \dot{U}_{max}\right], \dot{r} \in [\dot{r}_{min}, \dot{r}_{max}] \right\}, \tag{2.26}$$

and sampled uniformly to create the discrete set of N_U speed accelerations and

N_χ course accelerations

$$\begin{aligned} \dot{\boldsymbol{U}}_{samples} &= \left\{ \dot{U}_{max,1}, \dot{U}_{max,2}, \ldots, \dot{U}_{max,N_U} \right\} \\ \dot{\boldsymbol{r}}_{samples} &= \left\{ \dot{r}_{max,1}, \dot{r}_{max,2}, \ldots, \dot{r}_{max,N_\chi} \right\}. \end{aligned} \tag{2.27}$$

A set of motion primitives, which represents the maneuvers, are then calculated as the piecewise-linear speed and course acceleration trajectories

$$\dot{U}_{d,i}(t) = \begin{cases} k_{U,i} t & , 0 \leq t < T_{ramp} \\ \dot{U}_{i_{max}} & , T_{ramp} \leq t < T_U - T_{ramp} \\ \dot{U}_{i_{max}} - k_{U,i}(t - (T_U - T_{ramp})) & , T_U - T_{ramp} \leq t < T_U \\ 0 & , T_U \leq t \leq T \end{cases} \tag{2.28}$$

$$\dot{r}_{d,i}(t) = \begin{cases} k_{r,i} t & , 0 \leq t < T_{ramp} \\ 2\dot{r}_{i_{max}} - k_{r,i} t & , T_{ramp} \leq t < 2T_{ramp} \\ 0 & , 2T_{ramp} \leq t < T_\chi - 2T_{ramp} \\ -k_{r,i}(t - (T_\chi - 2T_{ramp})) & , T_\chi - 2T_{ramp} \leq t < T_\chi - T_{ramp} \\ -2\dot{r}_{i_{max}} + k_{r,i}(t - (T_\chi - 2T_{ramp})) & , T_\chi - T_{ramp} \leq t < T_\chi \\ 0 & , 2T_\chi \leq t \leq T, \end{cases} \tag{2.29}$$

T_U and T_χ are the speed and course maneuver times, and $k_{U,i} = \frac{\dot{U}_{max,i}}{T_{ramp}}$ and $k_{\chi,i} = \frac{\dot{\chi}_{max,i}}{T_{ramp}}$. $\dot{U}_{max,i}$ is the sampled acceleration for the speed motion primitive $i \in [1, N_U]$, and $\dot{r}_{max,i}$ is the sampled acceleration for the speed motion primitive $i \in [1, N_\chi]$.

Desired Course, Course Rate, and Speed Trajectories

The trajectories of the desired speed, course rate and course are found by integrating the motion primitives (2.28)- (2.29):

$$U_{d,i}(t) = U_{d,0} + \int_{t_0}^{t} \dot{U}_{d,i}(\gamma) d\gamma, i \in \left[1, N_U\right] \tag{2.30}$$

$$r_{d,i}(t) = r_{d,0} + \int_{t_0}^{t} \dot{r}_{d,i}(\gamma)d\gamma, i \in \left[1, N_\chi\right] \tag{2.31}$$

$$\chi_{d,i}(t) = \chi_{d,0} + \int_{t_0}^{t} \dot{\chi}_{d,i}(\gamma)d\gamma, i \in \left[1, N_\chi\right] \tag{2.32}$$

Note that sideslip is not included, and is therefore assumed zero. For the first level, the initial values correspond to the desired values of the previous BC-MPC iteration. For the other levels, the initial values correspond to the previous sub trajectory. This ensures continuity of the reference passed to the controller.

By setting $r_{d,0} = 0$, each maneuver starts and ends with a constant course motion. From (2.29), the integral of $\dot{r}_{d,i}(t)$ is zero, and hence a maneuver will start and end with the same course rate.

The vessel model and actuator constraints are used to remove infeasible velocities before the desired speed and course trajectories form a set of desired velocity trajectories

$$\mathcal{U}_d = \{U_{d,1}(t), U_{d,2}(t), ..., U_{d,N_U}(t)\} \times \left\{\chi_{d,1}(t), \chi_{d,2}(t), ..., \chi_{d,N_\chi}(t)\right\}. \tag{2.33}$$

Feedback-corrected Predicted Trajectories

In order to obtain continuity, there is no feedback included in the calculation of the desired velocity trajectories. Feedback-corrected pose trajectories are therefore calculated by simulating the closed-loop error dynamics of the vessel and vessel controllers. The error dynamics are approximated by first-order linear models:

$$\dot{\tilde{U}} = \frac{1}{T_{\tilde{U}}}\tilde{U} \tag{2.34}$$

$$\dot{\tilde{\chi}} = \frac{1}{T_{\tilde{\chi}}}\bar{\chi}, \tag{2.35}$$

where $\tilde{U} = \bar{U} - U_d$ and $\tilde{\chi} = \bar{\chi} - \chi_d$ is the errors, and $T_{\tilde{U}}$ and $T_{\tilde{\chi}}$ are the time constants. The predicted speed and course trajectories are then found as

$$\bar{U}_i(t) = \tilde{U}_0 e^{\frac{1}{T_{\bar{U}}}(t-t_0)} + U_{d,i}(t), i \in [1, N_U] \tag{2.36}$$

$$\bar{\chi}_i(t) = \tilde{\chi}_0 e^{\frac{1}{T_{\bar{\chi}}}(t-t_0)} + \chi_{d,i}(t), i \in [1, N_\chi]\,. \tag{2.37}$$

The feedback is now introduced through $\tilde{U}_0 = U_0 - U_{d,0}$ and $\tilde{\chi}_0 = \chi_0 - chi_{d,0}$. The set of predicted velocity trajectories is then given by

$$\bar{\mathcal{U}} = \left\{\bar{U}_1(t), \bar{U}_2(t), \ldots, \bar{U}_{N_U}\right\} \times \left\{\bar{\chi}_1(t), \bar{\chi}_2(t), \ldots, \bar{\chi}_{N_\chi}\right\} \tag{2.38}$$

The vessel position trajectories $\bar{\boldsymbol{p}} = \left[\bar{N}(t), \bar{E}(t)\right]^\top$ are calculated by integrating the time derivative

$$\dot{\bar{\boldsymbol{p}}} = \begin{bmatrix}\cos(\bar{\chi}) \\ \sin(\bar{\chi})\end{bmatrix} \bar{U} \tag{2.39}$$

with the current vessel position as the initial condition. The pose trajectories $\bar{\boldsymbol{\eta}} = \left[\bar{N}(t), \bar{E}(t), \bar{\chi}(t)\right]^\top$ are then given as

$$\bar{\mathcal{H}} = \bar{\boldsymbol{\eta}}(t; \bar{U}(t), \bar{\chi}(t)) | (\bar{U}(t), \bar{\chi}(t)) \in \bar{\mathcal{U}} \tag{2.40}$$

Desired Acceleration

To ensure that there exists a trajectory that converges towards the desired trajectory, the set of acceleration samples can be adjusted to include the desired acceleration. The desired acceleration is calculated for each step with a modified path tracking algorithm. A path particle following the desired path at the desired speed over ground (SOG) is defined,

$$\chi_{d,LOS} = \chi_{path} + \arctan\Big(-\frac{e}{\Delta}\Big) \tag{2.41}$$

$$U_{PP} = U cos(\chi - \chi_{path}) + \gamma_s s \tag{2.42}$$

the desired SOG and course acceleration is then calculated from the previous desired

$$\dot{U}'_d = \frac{U_{d,LOS} - U_{d,0}}{T_U - T_{ramp}} \tag{2.43}$$

$$\dot{r}'_d = \frac{\chi_{d,LOS} - \chi_{d,0}}{T_{ramp}\left(T_{chi} - 2T_{ramp}\right)} \tag{2.44}$$

The set of acceleration samples can then be aligned with de desired acceleration. If $\dot{U}'_d \in A_d$, the nearest sample in $\dot{\boldsymbol{U}}_{samples}$ is replaced by $\dot{U}'_d$. Similarly, the nearest sample in $\dot{\boldsymbol{r}}_{samples}$ is replaced by $\dot{r}'_d$ if $\dot{r}'_d \in A_d$, .

Optimization

The objective function considers trajectory alignment, avoidance of dynamic and static obstacles and transitional cost. It is given by

$$\begin{aligned} G(\bar{\boldsymbol{\eta}}(t), \boldsymbol{u}_d(t); \boldsymbol{p}_d(t)) = w_{al}\text{align}(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_d(t)) + w_{av,m}\text{avoid}_m(\bar{\boldsymbol{\eta}}(t)) \\ + w_{av,s}\text{avoid}_s(\bar{\boldsymbol{\eta}}(t)) + w_t\text{tran}(\boldsymbol{u}_d(t)), \end{aligned} \tag{2.45}$$

where $w_{al}, w_{av_m}, w_{av_s}, w_t \leq 0$ are tuning parameters weigthing the objective terms.

Trajectory Alignment

Trajectory alignment considers Euclidian and angular error between the predicted pose trajectory and the desired path. The desired course is calculated as

$$\chi_d(t) = \arctan(\dot{E}_d(t), \dot{N}_d(t)), \tag{2.46}$$

and the alignment cost is then

$$\text{align}(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_d(t)) = \int_{t_0}^{t_0+T_{full}} \left(w_p \left|\left| \begin{bmatrix} \bar{N}(\gamma) \\ \bar{E}(\gamma) \end{bmatrix} - \boldsymbol{p}_d(\gamma) \right|\right|_2 + w_\chi |\Upsilon(\bar{\chi}(\gamma) - \chi_d(\gamma))| \right) d\gamma. \tag{2.47}$$

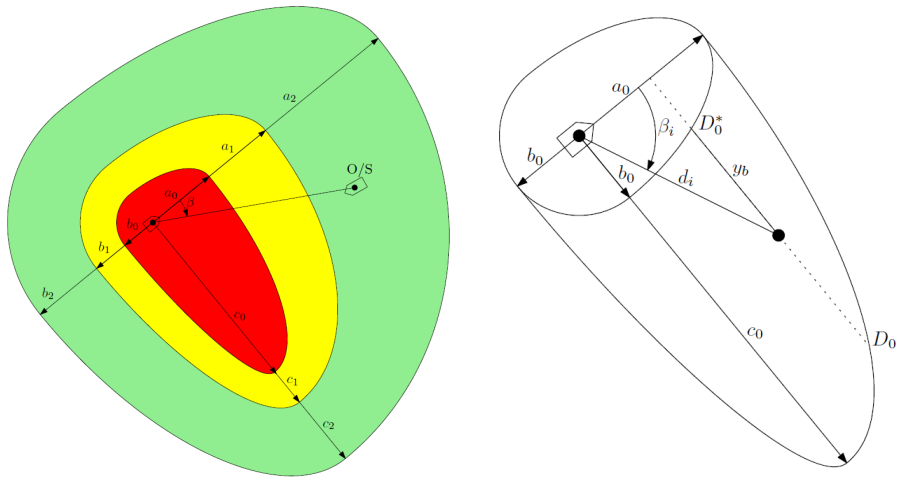
$w_p > 0$ and $w_\chi > 0$ controls the relative weighting of the Euclidean and course error.

Obstacle Avoidance

For obstacle avoidance, a collision, safety and margin region is calculated for each obstacle. The regions consist of three elliptical and one circular region and have a greater radius on the starboard and bow side of the obstacle. The radius D_k of the region k varies with the relative bearing between ownship and the obstacle, β_i, and is given by

$$D_k(\beta_i) = \begin{cases} b_k & \text{if} \quad \beta_i < -\frac{\pi}{2} \\ \frac{a_k b_k}{\sqrt{(b_k \cos\beta_i)^2 + (a_k \sin\beta_i)^2}} & \text{if} \quad -\frac{\pi}{2} \leq |\beta_i| < 0 \\ \frac{a_k c_k}{\sqrt{(c_k \cos\beta_i)^2 + (a_k \sin\beta_i)^2}} & \text{if} \quad 0 \leq |\beta_i| < \frac{\pi}{2} \\ \frac{b_k c_k}{\sqrt{(c_k \cos\beta_i)^2 + (b_k \sin\beta_i)^2}} & \text{if} \quad \frac{\pi}{2} \leq |\beta_i| \end{cases} \tag{2.48}$$

where a_k, b_k and $c_k = b_k + d_{\text{COLREGs}}$ are the minor, major and COLREGs axes of the collision, margin and safety regoin.



(a) Elliptical COLREGs function with minor and major axis.

(b) Varables for the inner elliptical penalty function.

Figure 2.5: Elliptical COLREGs regions and variables [18].

The elliptical regions motivate for passing the obstacle on her port or aft side and hence consideration of COLREGs. They also motivate for keeping a greater

distance to the obstacle if it is passed abaft or on its starboard side. An option to the elliptical regions is circular regions that does not consider COLREGs [18].

For the elliptical cost, an additional inner penalty is added to avoid constant cost inside the collision region. The inner penalty is given by

$$\text{inner\_penalty}_i(\bar{\boldsymbol{\eta}}(t)) = \begin{cases} 1 & \text{if} \quad d_i < D_0^* \\ 1 - \frac{y_b(d_i,\beta_i)}{d_{\text{COLREGs}}} & \text{if} \quad D_0^* \leq d_i < D_0 \\ 0 & \text{else,} \end{cases} \tag{2.49}$$

where $y_b(d_i, \beta_i)$ is the distance in the y-direction of the obstacle body frame from the D_0^* region to the point (d_i, β_i), and D_0^* is given by

$$D_0^*(\beta_1) = \begin{cases} \frac{a_0 b_0}{\sqrt{(b_0 \cos\beta_i)^2 + (a_0 \sin\beta_i)^2}} & \text{if } |\beta_i| < \frac{\pi}{2} \\ b_0 & \text{else.} \end{cases} \tag{2.50}$$

The totalt penalty is then

$$\begin{aligned} \text{penalty}_{i,\text{COLREGs}(\bar{\boldsymbol{\eta}}(t))} = \text{inner\_penalty}_i(\bar{\boldsymbol{\eta}}(t)) \\ + \begin{cases} 1 & \text{if} \quad d_i < D_0 \\ 1 + \frac{\gamma_1 - 1}{D_1 - D_0} & \text{if} \quad D_0 \leq d_i < D_1 \\ \gamma_1 - \frac{\gamma_1 (d_i - D_1)}{D_2 - D_1} & \text{if} \quad D_1 \leq d_i < D_2 \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{2.51}$$

Static Obstacles

An approach for handling static obstacles with BC-MPC is to represent the static obstacles as occupancy grids [26]. Each cell takes an occupancy value $O(\boldsymbol{p}) \in [0, 100]$, where $O(\boldsymbol{p}) = 0$ and $O(\boldsymbol{p}) = 100$ represents and empty and an occupied cell, respectively. Each obstacle is padded with a decaing gradient.

A cost is then calculated for each cell as

$$\text{avoid}_s\left(\bar{\boldsymbol{\eta}}\left(t\right)\right) = \int_{t_0}^{t_0+T} O\left(\bar{\boldsymbol{p}}\left(\gamma\right)\right) d\gamma. \tag{2.52}$$

Transitional cost

Transitional cost is included to increase robustness, and considers changes in the desired velocity trajectory $\boldsymbol{u}_d = \left[U_d, \chi_d\right]^\top$ compared to the previous desired velocity trajectory $\boldsymbol{u}_d^- = \left[U_d^-, \chi_d^-\right]^\top$ for the first maneuver. The minimum speed and course difference $e_{U,min},\ e_{\chi,min}$ for all candidates $\boldsymbol{u}_d \in \mathcal{U}_d$ are defined as

$$e_{U,\min} = \min_{\boldsymbol{u}_d(t)\in\mathcal{U}_d} \int_{t_0}^{t_0+T_1} \left|U_d(\gamma) - U_d^-(\gamma)\right| \mathrm{d}\gamma \tag{2.53}$$

$$e_{\chi,\min} = \min_{\boldsymbol{u}_d(t)\in\mathcal{U}_d} \int_{t_0}^{t_0+T_1} \left|\chi_d(\gamma) - \chi_d^-(\gamma)\right| \mathrm{d}\gamma. \tag{2.54}$$

where T_1 is the step time of the first maneuver. A transitional error is added if the speed or course error is greater than the minimum value,

$$\text{tran}\left(\boldsymbol{u}_d(t)\right) = \begin{cases} 1 & \text{if} \quad \int_{t_0}^{t_0+T_1} \left|U_d(\gamma) - U_d^-(\gamma)\right| \mathrm{d}\gamma > e_{U,\min} \\ & \text{or} \quad \int_{t_0}^{t_0+T_1} \left|\chi_d(\gamma) - \chi_d^-(\gamma)\right| \mathrm{d}\gamma > e_{\chi,\min} \\ 0 & \text{else.} \end{cases} \tag{2.55}$$

The transitional cost can also be considered individually for speed and course [26].

2.5.3 Velocity Obstacles (VO)

Velocity obstacles (VOs) is a reactive algorithm in the velocity space and was introduces by Fiorini and Schiller in 1998 [2]. Based on the current state of the vessel and the obstacles, a velocity obstacle forms a set of inadmissible velocities in the velocity space. An objective function is then calculated to find the optimal velocity of the remaining admissible velocities in the search space.

Different extensions can be added to the algorithm by introducing additional constraints to the velocity set or terms in the objective function. Examples of

such extensions are COLREGs consideration [5] or vessel dynamic consideration [2].

The Velocity Obstacle

Let $\boldsymbol{p} \in \mathbb{R}^2$ and $\boldsymbol{v} \in \mathbb{R}^2$ be the horizontal position and velocity of a vessel, respectively. A ray starting in position $\boldsymbol{p}$ going in the direction $\boldsymbol{v}$ is then defined as

$$\lambda(\boldsymbol{p}, \boldsymbol{v}) = \{\boldsymbol{p} + t\boldsymbol{v} | t \leq 0\}. \tag{2.56}$$

Consider a craft A with shape $\mathcal{A}$ and an obstacle B with shape $\mathcal{B}$, with positions and velocities $\boldsymbol{p}_A, \boldsymbol{v}_A$ and $\boldsymbol{p}_B, \boldsymbol{v}_B$, respectively. The velocity obstacle VO_B^A is a cone in the velocity space representing the velocities $\boldsymbol{v}_A$ of the craft A that will result in a collision with the obstacle B assuming that A and B keeps the current velocities. VO_B^A can be defined as

$$VO_B^A(\boldsymbol{v_B}) = \{\boldsymbol{v_A} | \lambda(\boldsymbol{p}_A, \boldsymbol{v}_A - \boldsymbol{v}_B) \cap (\mathcal{B} \oplus -\mathcal{A}) \neq \emptyset\}, \tag{2.57}$$

where $\mathcal{A} \oplus \mathcal{B} = \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}$ is the Minowski sum and $-\mathcal{A} = \{-a | a \in \mathcal{A}\}$ is the reflection of $\mathcal{A}$.

Search Space and Cost

Consider the velocity space $\mathcal{V}$. The admissible velocity space is then given by

$$\mathcal{V}_{\text{adm}} = \mathcal{V} \setminus VO_B^A \tag{2.58}$$

However, it is in most situations useful to consider a discrete search space $\mathcal{U} = \{U_1, U_2, \ldots, U_n\} \times \{\chi_1, \chi_2, \ldots, \chi_m\}$ of velocity candidates $(U_i, \chi_j), i \in [1, n], j \in [1, m]$.

The desired velocity of $\boldsymbol{v}_A$ is then found by solving the optimization problem

$$\boldsymbol{v}_A = \underset{\boldsymbol{v}_A \in \mathcal{U}_{adm}}{\text{argmin}} \; J(\boldsymbol{v}_A) \tag{2.59}$$

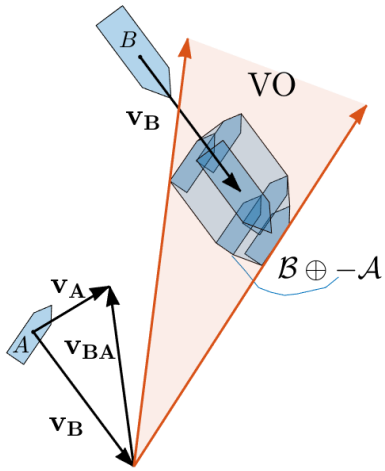


Figure 2.6: Velocity obstacle calculated by the Minowski sum.

where $J(\boldsymbol{v}_A)$ is a cost function and $\mathcal{U}_{adm}$ is the set of admissible velocity candidates.

Collapsing the Configuration Space

In order to reduce the problem, the configuration space of the vessels can be collpased such that the footprint is discs with radius r_A and r_B. VO_B^A is then given by the intersection of the disc with center $\boldsymbol{p}_B + \boldsymbol{v}_B$ and radius $r_{AB} = r_A + r_B$, and the cone with apex in $\boldsymbol{p}_A + \boldsymbol{v}_B$ bounded by tangents δ_r and δ_f [2], as illustrated in Figure 2.7.

Denoting the vector between A and B as $\boldsymbol{p}_{AB} = \boldsymbol{p}_B - \boldsymbol{p}_A$ and the realtive velocity $\boldsymbol{v}_{BA} = \boldsymbol{v}_A - \boldsymbol{v}_B$, the angle α from the centerline of the cone to the tangents are found as:

$$\alpha = \Upsilon\left(\arcsin\left(\frac{r_{AB}}{\|\boldsymbol{p}_{AB}\|}\right)\right). \tag{2.60}$$

The angle β is defined as

$$\beta = \arctan(\boldsymbol{p}_{AB_E}, \boldsymbol{p}_{AB_N}) \tag{2.61}$$

where $\boldsymbol{p}_{AB_E}$ and $\boldsymbol{p}_{AB_N}$ are the east and north coordinates of $\boldsymbol{p}_{AB}$, respectively.

The velocity cone with the collapsed configuration space can then be defined as

$$VO_B^A = \{\boldsymbol{v}_A | \beta - \alpha \leq \arctan(\boldsymbol{v}_{AB_E}, \boldsymbol{v}_{AB_N}) \leq \beta + \alpha\}. \tag{2.62}$$

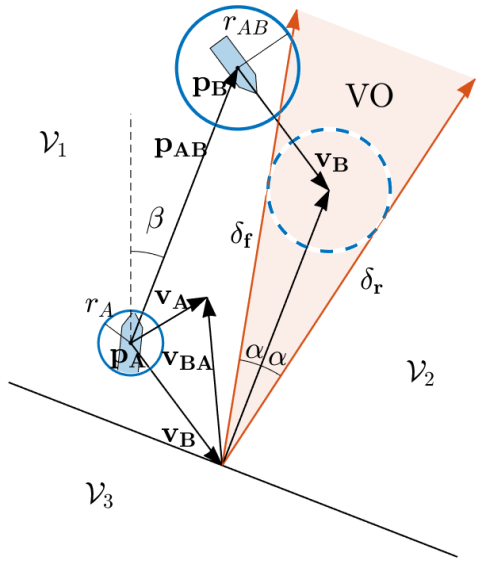


Figure 2.7: Velocity obstacle with collapsed configuration space.

Closest Point of Approach (CPA)

The Velocity Obstacle only considers which velocities that will result in a collision in the future and not when this collision will happen. Therefore, the time and distance to the closest point of approach (CPA) can be calculated and used to decide which obstacles to consider or prioritize. The time to CPA, t_{CPA} is given by

$$t_{\text{CPA}} = \begin{cases} 0 & \text{if } \|\boldsymbol{v}_A - \boldsymbol{v}_B\| \leq \epsilon \\ \frac{(\boldsymbol{p}_A - \boldsymbol{p}_B)\cdot(\boldsymbol{v}_A - \boldsymbol{v}_B)}{\|\boldsymbol{v}_A - \boldsymbol{v}_B\|^2} & \text{otherwise} \end{cases}, \tag{2.63}$$

and the distance to CPA, d_{CPA}, by

$$d_{CPA} = \|(\boldsymbol{p}_A + \boldsymbol{v}_A t_{CPA}) - (\boldsymbol{p}_B + \boldsymbol{v}_B t_{CPA})\|. \tag{2.64}$$

A collision situation is then likely to happen in the near future if

$$0 \leq t_{CPA} \leq t_{CPA_{max}} \ \wedge \ d_{CPA} \leq d_{CPA_{min}}. \tag{2.65}$$

Velocity Sets

To handle the velocities in the search space, the remaining velocity space $\mathcal{V} \backslash VO_B^A$ is divided in to three velocity sets, $\mathcal{V}_1, \mathcal{V}_2$ and $\mathcal{V}_3$, as illustrated in Figure 2.7 [6]. $\mathcal{V}_1$ is where A passes B on its left-hand side, and is defined as

$$\mathcal{V}_1 = \{\boldsymbol{v} | \boldsymbol{v} \notin VO_B^A(\boldsymbol{v}_B) \cup \mathcal{V}_3, [(\boldsymbol{p}_B - \boldsymbol{p}_A) \times (\boldsymbol{v}_A - \boldsymbol{v}_B)]_z < 0\}, \tag{2.66}$$

where $[\,\cdot\,]$ extracts the z-component of the vector. $\mathcal{V}_2$ is the velocities where A passes B on its right-hand side and is given by

$$\mathcal{V}_2 = \{\boldsymbol{v} | \boldsymbol{v} \notin VO_B^A(\boldsymbol{v}_B) \cup \mathcal{V}_3 \cup \mathcal{V}_1\}. \tag{2.67}$$

V_3 is the set of velocities where the relative velocity of A points away from the obstacle B,

$$\mathcal{V}_3 = \{\boldsymbol{v} | \boldsymbol{v} \notin VO_B^A(\boldsymbol{v}_B), (\boldsymbol{p}_B - \boldsymbol{p}_A)\cdot(\boldsymbol{v}_A - \boldsymbol{v}_B) < 0\}. \tag{2.68}$$

COLREGs Consideration

An approach to consider COLREGs with VO is to identify the COLREGs rule and apply COLREGs constraints to the search space [5]. The COLREGs situation

is identified based on the relative state between A and B, and one or more of the four COLREGs rules may be selected: overtaking, head-in, crossing from starboard and crossing from the port.

If the head-on or crossing from starboard rules are selected, $\mathcal{V}_1$ are made inadmissible, such that the search space is constrained to

$$\mathcal{V}_{adm} = \{\boldsymbol{v} | \boldsymbol{v} \notin VO_B^A \cup \mathcal{V}_1\}. \tag{2.69}$$

To avoid chattering behavior, hysteresis can be added to the COLREGs rule selection [5], demanding a rule to be selected several times in a row before it is applied. If the criteria for the COLREGs situation corresponding to the currently applied rule is satisfied within the last n_h timesteps, the current rule still applies. If not, the rule corresponding to the currently identified COLREGs situation is applied. This approach introduces a memory and deliberate characteristics to the reactive algorithm.

Static Obstacles

For static obstacles, the velocity cone can be calculated as for dynamic obstacles, with an obstacle velocity $\boldsymbol{v}_B = \boldsymbol{0}$. However, considering all velocities towards static objects is not necessarily efficient. Consider a discrete search space $\mathcal{U} = (U_i, \chi_j)$. Transversing a ray $\boldsymbol{\lambda}(\boldsymbol{p}_A, U_i, \chi_i)$

$$\boldsymbol{\lambda}(\boldsymbol{p}_A, U_i, \chi_i) = \boldsymbol{p}_A + U_i t \begin{bmatrix} \cos(\chi_i) \\ \sin(\chi_i) \end{bmatrix} \tag{2.70}$$

for each velocity candidate (U_i, χ_i) that lies inside the velocity cone. The time to collision with the first point on $\boldsymbol{\lambda}$ that intersects with the obstacle, t_{CPA}, can then be calculated for each velocity candidate (U_i, χ_i). The velocity obstacle can be limited to include only the velocity candidates where $0 < t_{CPA} < t_{CPA_{min}}$. Note that this allows the vessel to travel closer to the obstacle at the cost of a low speed [25].

Multiple Obstacles

For multiple obstacles, the inadmissible velocity set is simply obtained by the superposition of the velocity cones. For n obstacles $B_1, B_2, \dots B_n$, the admissible search space is given by

$$\mathcal{V}_{adm} = \{\boldsymbol{v} | \boldsymbol{v} \notin VO^A_{B_1}(\boldsymbol{v}_{B_1}) \cup VO^A_{B_2}(\boldsymbol{v}_{B_2}) \cup \dots VO^A_{B_n}(\boldsymbol{v}_{B_n})\}. \tag{2.71}$$

Chapter 3

Simulator Implementation

The simulator is implemented in MATLAB and SIMULINK, Mathworks 2018b. The simulation environment is in two dimensions, suitable for simulating surface vessels. It is assumed that all vessels have zero sway motion, $v = 0$, and that conditions are ideal without ocean currents and other disturbances, with the exception of measurement noise in some simulations. In the absence of sway motions and currents, the sideslip β is zero, meaning that the course is equal to the heading angle, $\chi = \psi$ [19].

For simplicity, vessel footprints or size are not considered in this implementation. Hence, the vessel positions are configured as points $\boldsymbol{p} = \begin{bmatrix} N, E \end{bmatrix}^\top$ which describes the north and east position. This point is also considered the origin of the vessel's BODY frame.

3.1 Simulator Overview

Figure 3.1 shows an overview of the simulator. An explanation of the implementation for each part is described further in this chapter. Guidance is implemented differentely for the respective algorithms, and is therefore described together with the colav implementation of BC-MPC (Section 3.5) and VO (Section 3.6).

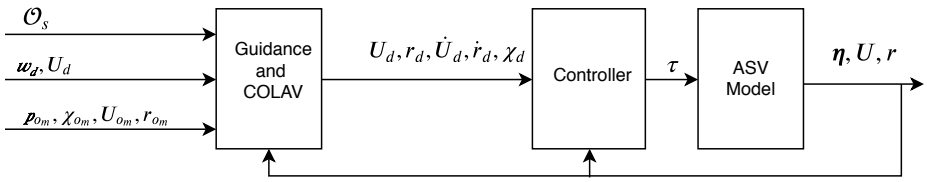


Figure 3.1: Simulator overview. $\mathcal{O}_s$ is the set of statis obstacles, $\boldsymbol{w_d}$ the desired waypoints, U_d the desired speed and $\boldsymbol{p}_{o_m}$, χ_{o_m}, U_{o_m}, r_{o_m} are the postions, course, speed and course rate of the dynamic obstacles.

3.2 Obstacles

Dynamic obstacles are modeled by the 3DOF model (2.9). The configuration is given as

$$o_{m_i} = (\boldsymbol{p}_{o_m,i}, \chi_{o_m,i}, U_{o_m,i}, r_{o_m,i}), \tag{3.1}$$

and the set of all dynamic obstacles are denoted as $\mathcal{O}_m$.

Static obstacles are defined by polygons

$$\mathrm{o}_{s_i} = (\boldsymbol{N}_{o_s,i}, \boldsymbol{E}_{o_s,i}) \tag{3.2}$$

where $\boldsymbol{N}_i = \left[N_{o_{s,1}}, N_{o_{s,2}}, ..., N_{o_{s,n}}\right]^\top$ and $\boldsymbol{E}_i = \left[E_{o_{s,1}}, E_{o_{s,2}}, ..., E_{o_{s,n}}\right]^\top$ are the north- and east- cooridinates of the n vertices of obstacle i. The set of all static obstacles are denoted as $\mathcal{O}_s$. For BC-MPC, the polygons are represented by an occupancy grid as described in Section 2.5.2. The occupancy grid is implemented using the Robotics Systems Toolbox in MATLAB.

3.3 Measurement Noise

The measurement noise is modeled by the continuous-time stochastic Wiener process, which is used to represent the integral of a white noise Gaussian process [27]. The measurement noise is generated by simulating the system (A, B) over the same timespan as the simulation with a random input generated by the

MATLAB function *randn*. The code for noise generation is provided by Co-Supervisor Bjørn-Olav Holtung Eriksen. The system matrices are defined as

$$\boldsymbol{A} = \text{diag}\left(-\frac{1}{T_{\boldsymbol{p}}}, -\frac{1}{T_{\boldsymbol{p}}}, -\frac{1}{T_{\psi}}, -\frac{1}{T_u}\right) \tag{3.3}$$

and

$$\boldsymbol{B} = \text{diag}\left(\frac{k_{\boldsymbol{p}}}{T_{\boldsymbol{p}}}, \frac{k_{\boldsymbol{p}}}{T_{\boldsymbol{p}}}, \frac{k_{\psi}}{T_{\psi}}, \frac{k_u}{T_u}\right), \tag{3.4}$$

where k and T are positive gains and time constants.

3.4 Control and ASV model

The ASV is modeled by the Telemetron model (2.10) - (2.12), and the implemented controller is the model-based FF-FB-C controller (2.16), which has shown good performance [4]. Physical limitations are included by saturation of control inputs, SOG and yaw rate.

3.5 Implementation of BC-MPC

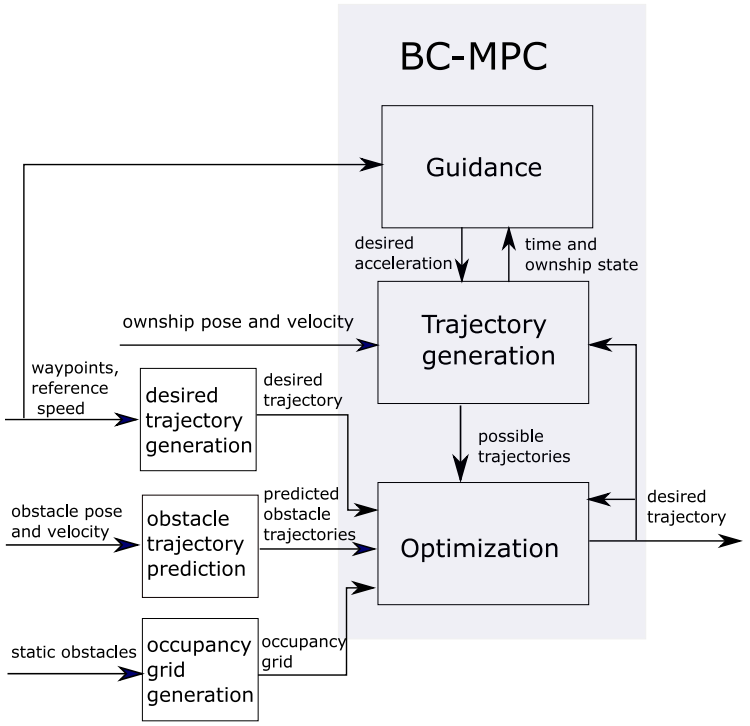


Figure 3.2: Overview of BC-MPC implementation

Figure 3.2 gives an overview of the implementation. The BC-MPC algorithm is implemented as described in Section 2.5.2. The trajectory alignment is considered in the cost function, and the guidance is taken care of in the trajectory generation. Note that the guidance is not considering the ownship state directly, but ensures that there is a trajectory converging towards the goal if applicable.

The reference trajectory is calculated from the desired path given by waypoints and the desired speed, with the trajectory speed set to $U_t = U_d > 0$. The predicted obstacle trajectories are calculated assumed that the obstacle keeps its current measured speed and course.

The integration method used is the first order forward Euler integration

$$y_{n+1} = y_n + hf\left(x_n, y_n\right), \tag{3.5}$$

where $h > 0$ is the timestep.

For the optimal trajectory, the next step of the predicted velocity trajectory, the desired course rate trajectory and the desired acceleration trajectory are sent to the controller as the reference.

3.6 Implementation of VO

3.6.1 Algorithm Overview

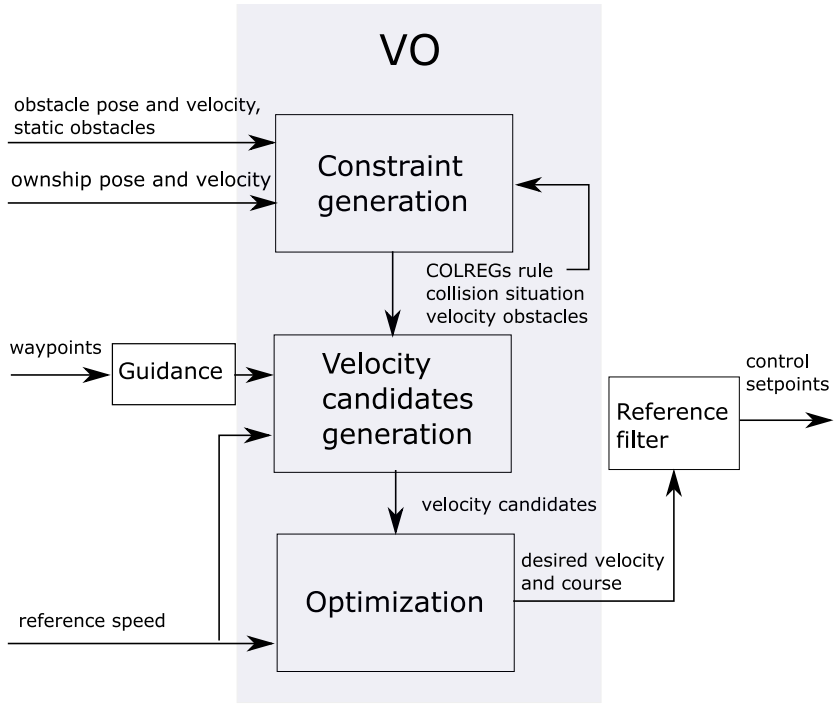


Figure 3.3: Overview of VO implementation

The implementation of VO is described Section 2.5.3 and this Section, and is based on previous apllications of VO for surface vessels considering COLREGs [6] [25]. Figure 3.3 provides an implmentation overview. The algorithm is implemented with hard constraints, such that velocity candidates in conflict with VO or COLREGs constraints are considered inadmissible and removed from the search space. If there are no admissible velocities left, the algorithm fails.

The algorithm flow can be summarized as follows:

- For each dynamic obstacle
  - Calculate t_{CPA} and d_{CPA} with the current velocity and identify if a collision situation occurs
  - Identify and select COLREGs rule
  - If there is a collision situation or a COLREGs rule is selected, calculate and apply VO and COLREGs constraints
- Calculate and apply VO constraints for static obstacles
- Generate a discrete velocity search space, the velocity set $\mathcal{U}$ of velocity candidates (U_i, χ_i)
- For the remaining velocity candidates, solve the optimization problem and set the velocity candidate with the minimum cost as setpoints
- Pass setpoints U_{sp} and χ_{sp} to the reference filter. From reference filter, pass $U_d, \dot{U}_d, r_d, \dot{r}_d, \chi_d$ to the controller

3.6.2 Velocity Search Space

A discrete velocity grid of velocity candidates (U_i, χ_i) where $U_{min} \leq U \leq U_{max}$ and $-\pi \leq \chi \leq \pi$ is generated and forms the search space $\mathcal{U} = \{U_1, U_2, \ldots, U_n\} \times \{\chi_1, \chi_2, \ldots, \chi_m\}$. The number of samples is chosen such that each sample represents a suffient change in course or speed. If the reference U_d or χ_d are within the search space, the nearest sample is replaced with the reference, as demonstrated in Figure 3.4.

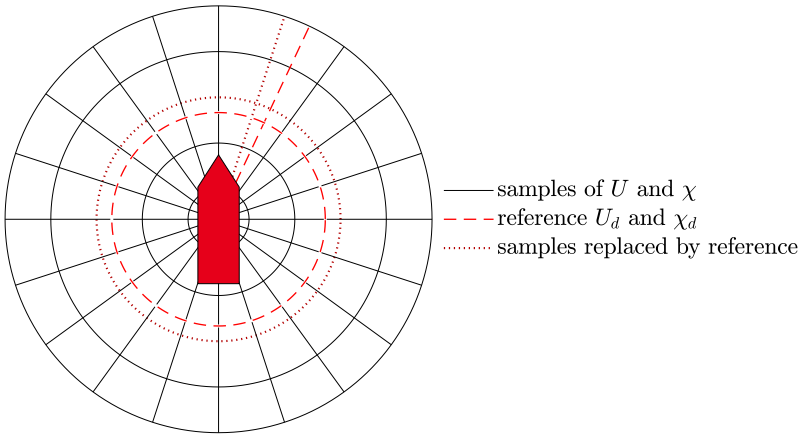


Figure 3.4: Example of a discrete velocity grid, where the nearest sample is aligned with the desired velocity candidate.

3.6.3 Dynamic Obstacles

For all dynamic obstacles, it is first determined if they are in a collision situation using the CPA approach (2.65), considering the current velocity of the ownship. If the obstacle is not considered to be in a collision situation, no COLREGs situation is identified. If the obstacle is in a collision situation, the COLREGs situation is identified based on the relative bearing calculated as

$$\beta = \Upsilon\left(\arctan(E - E_{o_s}, N - N_{o_s}) - \psi_{o_s}\right), \tag{3.6}$$

where $\boldsymbol{p}$ is the position of the ownship, and $\boldsymbol{p}_{o_s}$ and ψ_{o_s} is the position and heading of the obstacle. The rule is then identified as described in Section 2.4.2.

The applied COLREGs rule is then selected using the approach described in Section 2.5.3. Note that with the hysteresis, another rule than the identified can be selected. Also, a rule can be selected even if there is no collision situation if it is identified within the last n_h timesteps.

For all obstacles which are either in a collision situation or has an applied COLREGs rule, the velocity obstacle is calculated and added to the constraint set. All dynamic obstacles are represented as discs with the center in the measured obstacle position and radius r_o. Hence, VO_B^A is calculated as (2.62). If the head-on or

crossing from starboard COLREGs rule is applied, $\mathcal{V}_1$ is added to the constraint set as described in Section 2.5.3.

3.6.4 Static Obstacles

Static obstacles are represented as polygons, and they are expanded by a border $B = \begin{bmatrix} \boldsymbol{N}_b & \boldsymbol{E}_b \end{bmatrix}$ of radius r_{o_s} with rounded edges where $\boldsymbol{N}_b = \begin{bmatrix} N_{b_1}, \ldots, N_{b_n} \end{bmatrix}$ and $\boldsymbol{E}_b = \begin{bmatrix} E_{b_1}, \ldots, E_{b_n} \end{bmatrix}$. The shortest vector from the ownship position $\boldsymbol{p}$ to the polygon is denoted as $\boldsymbol{p}_s$. The minimum and maximum angles with vectors from $\boldsymbol{p}$ to an edge of B are found as

$$\begin{aligned} \alpha_{min} &= \min_{N_{b_i}, E_{b_i} \in B} \Upsilon \left(\arctan\left(\boldsymbol{p}_{s_N}, \boldsymbol{p}_{s_E} \right) - \arctan\left(N_{b_i} - N, E_{b_i} - E \right) \right) \\ \alpha_{max} &= \max_{N_{b_i}, E_{b_i} \in B} \Upsilon \left(\arctan\left(\boldsymbol{p}_{s_N}, \boldsymbol{p}_{s_E} \right) - \arctan\left(N_{b_i} - N, E_{b_i} - E \right) \right) \end{aligned} \tag{3.7}$$

The angles α_f and α_r is then chosen such that the velocity obstacle occupies a sector equal to or less than π:

$$\begin{aligned} \alpha_f &= \begin{cases} \alpha_{min} & \text{if} \quad \alpha_{max} - \alpha_{min} \leq \pi \\ \alpha_{max} & \text{else} \end{cases} \\ \alpha_r &= \begin{cases} \alpha_{max} & \text{if} \quad \alpha_{max} - \alpha_{min} \leq \pi \\ \alpha_{min} & \text{else.} \end{cases} \end{aligned} \tag{3.8}$$

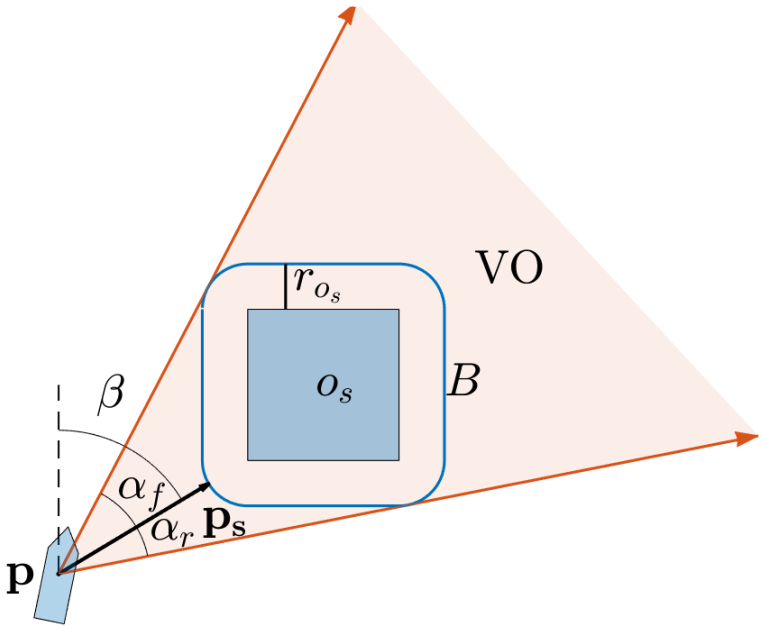


Figure 3.5: The velocity obstacle (VO) for a static obstacle o_s with a border B of radius r_{o_s}. $\boldsymbol{p}_s$ is the shortest vector from ownship $\boldsymbol{p}$ to B, and α_f, α_r are the angles from $\boldsymbol{p}_s$ to the vectors bounding the velocity obstacle.

Considering the approach for static obstacles in Section 2.5.3, the velocity obstacle is calculated as

$$VO = \{\boldsymbol{v} | \beta - \alpha_f \leq \arctan(\boldsymbol{v}_E, \boldsymbol{v}_N) \leq \beta + \alpha_r \cap 0 \leq t_{CPA}(\boldsymbol{v}, \boldsymbol{p}, o) \leq t_{CPA_{min}}\} \tag{3.9}$$

where $t_{CPA}(\boldsymbol{v}, \boldsymbol{p}, o)$ is the time to collision with the obstacle o following the ray from $\boldsymbol{p}$ in the direction $\boldsymbol{v}$.

3.6.5 Guidance

The desired course reference is calculated by LOS lookahead-based guidance as described in Section 2.2.1. The desired speed is not considered in the guidance,

and hence the VO implementation considers only path following and not trajectory tracking, in opposition to BC-MPC.

3.6.6 Cost Function

The choosen cost function is

$$J_{i,j} = \boldsymbol{v}_e^\top W \boldsymbol{v}_e, \tag{3.10}$$

where $\boldsymbol{v}_e = \boldsymbol{v}_d - \boldsymbol{v}_{ij}$ is the error velocity vector representing the difference of the desired velocity $\boldsymbol{v}_d = \left[U_d \cos(\chi_d), U_d \sin(\chi_d)\right]^\top$ and the velocity candidate $\boldsymbol{v}_{ij} = \left[U_j \cos(\chi_i), U_j \sin(\chi_i)\right]^\top$. The error velocity is the only parameter considered in the objective function since other considered parameters are implemented as constraints.

3.6.7 Controller Setpoints

The velocity pair with the minimum cost, (U_d, χ_d) is passed to the controller togheter with $\dot{U}_d$, r_d and $\dot{r}_d$, which are found using a reference filter [19]. In order to acheive continuous references to the controller, a second order model is used for speed, and a third order model for course

$$\ddot{U}_d + 2\zeta\omega\dot{U}_d + \omega^2 U_d = \omega^2 U_s \tag{3.11}$$

$$\dddot{\chi}_d + (2\zeta + 1)\omega\ddot{\chi}_d + (2\zeta + 1)\omega^2\dot{\chi}_d + \omega^3\chi_d = \omega^3\chi_s. \tag{3.12}$$

U_s, χ_s are the set points from VO and $U_d, \dot{U}_d, \chi_d, \dot{\chi}_d = r_d, \ddot{\chi}_d = \dot{r}_d$ are the reference passed to the controller. The damping constant $\zeta = 1$ to achieve a critically damped system, and ω is tuned based on guidelines by Fossen [19].

Chapter 4

Simulation results

In order to evaluate and compare the BC-MPC and VO algorithms, different scenarios are simulated in the MATLAB/SIMULINK simulation environment as described in Chapter 3.

Both single-obstacle scenarios evaluating one COLREGs situation and more complex multi-obstacle scenarios as simulated by Eriksen [18] are performed. Two different scenarios with static obstacles are also simulated. In addition, the two single-obstacle scenarios head-on and crossing-from-starboard are simulated as Monte Carlo simulations with measurement noise. Section 4.1 provides an explaination of each scenario. To limit the scope of the report, the obstacles keep constant speed and course, and are therefore non-maneuvering in all of the scenarios. Further, only two senarios are simulated with noise.

The simulation setup, including the parameters, is presented in Section 4.3. The results from the simulations without noise are presented and discussed in Section 4.4, and from the simulations with noise in Section 4.5.

The simulations without noise are evaluated quantitatively based on the performance metrics defined in Section 4.2 and the compliance with COLREGs rule 8, 13-15 and 17. Rule 16 is not evaluated as it is assumed covered by rule 8 and 13-15. For the simulations with noise, COLREGs compliance is not evaluated quantitatively but included in the qualitative evaluation. Furthermore, when VO and its performance are discussed, it refers to the implementation explained in Section 3.6.

4.1 Scenarios

The scenarios simulated and evaluated in this report include both single- and multi-obstacle scenarios where the ownship faces one or more targets, and static obstacle scenarios where the ownship must maneuver to avoid land. For all scenarios, the reference path is heading north starting at the initial position of the ownship $\begin{bmatrix} N_0, E_0 \end{bmatrix}^\top = \begin{bmatrix} 0, 0 \end{bmatrix}^\top$. The constant reference speed is set to $U_d = 10m/s$. The scenarios are created such that the considered COLREGs rules are evaluated.

The simulated scenarios are:

Overtaking (O) In the overtaking scenario, the ownship is following the same path as the target. However, the ownship keeps a higher speed and is abaft the target. The ownship should, therefore, overtake the target according to Rule 13. The Rule does not specify on which side ownship should pass the target.

Head-on (HO) In the head-on scenario, the ownship faces a target on a reciprocal course. The vessels are heading directly towards each other, and according to Rule 14, they both have a keep-out-of-the-way obligation. Ownship should make a starboard maneuver to cross the target on its port side.

Crossing from starboard (CS) In the crossing from starboard scenario, the target is crossing the path of the ownship from starboard. According to Rule 15, the ownship should then keep out of the way, and avoid passing ahead of the target if possible. In other words, the ownship should make a starboard maneuver and pass abaft the target.

Head on and crossing from port (HO+CP) In the head-on and crossing from port scenario, the ownship faces two targets. The first is on reciprocal course heading directly towards the ownship. The ownship has a keep out of the way obligation to this target and should make a starboard maneuver to pass on the target's port side. At the same time, there is a second target crossing from port. The ownship has a stand-on obligation to this target and should keep its obligations according to Rule 17.

Crossing from starboard and crossing from port (CS+CP) In the crossing from starboard and crossing from port scenario, the ownship faces two crossing targets simultaneously. However, they are crossing from opposite sides, and the ownship has both a keep out of the way and a stand-on obligation to the respective targets. According to Rule 15, the ownship should

make a starboard maneuver and pass abaft the target crossing from starboard. At the same time, the ownship should keep its stand-on obligation to the target crossing from port according to Rule 17.

Static scenario A In this scenario, the ownship follows a straight reference path with a long obstacle on its starboard side. In addition, there is one obstacle blocking the reference path ahead of the ownship, making the shortest path between the obstacles. As there are no other vessels in this scenario, the COLREGs rules are not explicitly considered in this scenario.

Static scenario B In this scenario, the ownship follows a straight reference path, blocked by a wide obstacle ahead of the ownship. The ownship is then required to get far of the reference path to pass the obstacle. The COLREGs rules are not considered explicitly in this scenario.

Please refer to the simulation results in sections 4.4.1 - 4.4.7 for illustrations of the scenarios.

4.2 Performance Metrics

Performance metrics are used to evaluate and compare BC-MPC and VO quantitatively. Travel time (TT) and travel distance (TD) denotes the total travel values and are defined as

$$TT(t) = t - t_0, \tag{4.1}$$

and

$$TD(t) = \int_{t_0}^{t} |U(\gamma)| d\gamma. \tag{4.2}$$

The minimum distance to obstacle (MDO) between the ownship and the obstacles is calculated as

$$MDO(t) = \min_{(o_i,\gamma)\in(\mathcal{O},[t_0,t])} \|\boldsymbol{p}(\gamma) - \boldsymbol{p}_{o^i}(\gamma)\| \tag{4.3}$$

where $\mathcal{O} = \mathcal{O}_m \cup \mathcal{O}_s$ is the set of all dynamic and static obstacles.

The integral of absolute course rate (IACR) and integral of absolute speed rate (IASR) are used to measure how much the ship changes its course and speed, respectively, and are given as

$$IACR(t) = \frac{1}{t - t_0} \int_{t_0}^{t} |r(\gamma)| d\gamma, \tag{4.4}$$

and

$$IASP(t) = \frac{1}{t - t_0} \int_{t_0}^{t} |\dot{U}(\gamma)| d\gamma. \tag{4.5}$$

The values are normalized in order to be comparable for different travel time TT.

4.3 Simulation Setup

The Telemetron parameters [21] is shown in Table 4.1.

Table 4.1: Telemetron specifications and parameters.

Component	Description
Vessel hull	Polarcirkel Sport 845
Length	8.45 m
Height	2.71 m
Weight	1675 kg

Parameter	Value
τ_{min}	$[0, -1]^\top$
τ_{max}	$[1, 1]^\top$
$\dot{\tau}_{min}$	$[.5, .7]^\top$
$\dot{\tau}_{max}$	$[-.5, -.7]^\top$

Table 4.2: Controller gains.

Parameter	Value
k_{p_U}	.6
k_{p_r}	.35
k_{p_χ}	.15
k_{i_U}	.01
k_{i_χ}	.015

The parameters used for BC-MPC and VO are shown in Table 4.3 and 4.4, respectively. The controller parameters are taken from [4]. The BC-MPC parameters

are similar to those which are used for experiments and simulations in [18]. For VO, the parameter set is choosen such that the algorithm works well in general for all simulated scenarios.

The borders and regions around the obstacles are smaller for dynamic than for static obstacles for both algorithms. This is because the dynamic obstacles are configured as points, whereas the static obstacles are defined as polygons. The VO border radius r_{o_s} is set to half of the BC-MPC gradient radius. The dynamic obstacle radius r_o is set equal to the safety region minor axis, b_1.

Table 4.3: BC-MPC parameters.

Parameter	Value	Unit	Description
$\boldsymbol{N}_U$	$[5, 1, 1]$		number of speed motion primitives
$\boldsymbol{N}_\chi$	$[5, 3, 3]$		number of course motion primitives
$\boldsymbol{T}$	$[5, 20, 30]$	s	prediction time
a_0	50	m	collision region major axis
a_1	150	m	safety region major axis
a_2	250	m	margin region major axis
b_0	25	m	collision region minor axis
b_1	75	m	safety region minor axis
b_2	125	m	margin region minor axis
$d_{COLREGs}$	100	m	COLREGs distance
γ	0.1		obstacle cost parameter
d_p	100	m	gradient obstacle radius
w_χ	100		angular error weight
w_{al}	1		trajectory alignment weight
w_{av_s}	6000		static avoidance weight
w_{av_d}	6000		dynamic avoidance weight
w_{tc}	4800		translational cost weight
Δ	500	m	lookahead distance
γ_s	0.005		LOS along-track distance gain

Table 4.4: VO parameters.

Parameter	Value	Unit	Description
U_{min}	2	m/s	minimum SOG
U_{max}	15	m/s	maxium SOG
$t_{CPA_{min}}$	200	s	minimum time to CPA
$d_{CPA_{min}}$	30	m	minimum distance at CPA
r_o	75	m	dynamic obstacle radius
r_{o_s}	50	m	static obstacle boundary radius
n_h	50		hysteresis
ζ	1		reference filter damping constant
ω_n	1		reference filter frequency
$\boldsymbol{W}$	$diag(2, 1)$		velocity error weighting matrix

The noise parameters are provided by Bjørn-Olav Holtung Eriksen and are based on parameters from the experiments [4].

Table 4.5: Noise parameters.

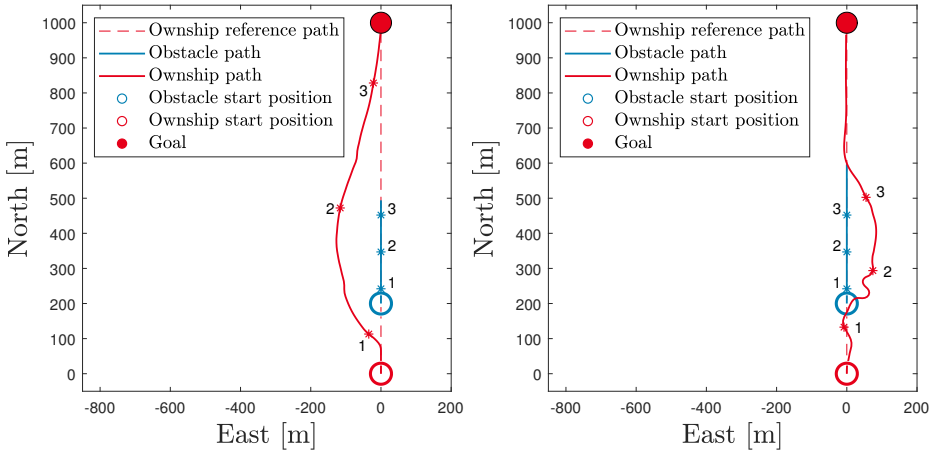
Parameter	Value	Parameter	Value
$T_{\boldsymbol{p}}$	5	$k_{\boldsymbol{p}}$	10
T_ψ	5	k_ψ	0.6
T_u	5	k_u	1

4.4 Simulations Without Noise

This section presents the simulation results from the simulations without the noise of the scenarios described in Section 4.1. For each scenario, plots of the simulation, the metrics defined in Section 4.2 and an evaluation of the behavior and the COLREGs compliance of the considered rules are presented. Last, a summary of the results is given.

4.4.1 Overtaking

In this overtaking scenario, the ownship and the target follows the same path with the same course. The ownship is abaft the target keeping a higher speed, and is overtaking the target.



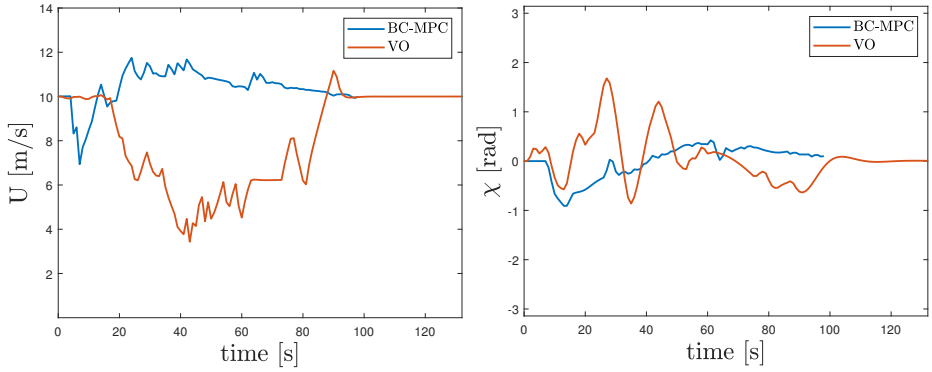
(a) BC-MPC **(b)** VO

Figure 4.1: Overtaking scenario simulations. The initial positions are marked with circles, and the three time instances $15s$, $50s$ and $85s$ are marked with star number 1, 2 and 3, respectively.

Table 4.6: Results of overtaking scenario simulation. Shows the metrics and compliance to COLREGs Rule 8 (actions to avoid collision) and Rule 13 (actions in overtaking situations).

Algorithm	TD	TT	MDO	IACR	IASR	Rule 8	Rule 13
BC-MPC	1021 m	99 s	109 m	0.0733	0.2133	Yes	Yes
VO	1057 m	133 s	64 m	0.1062	0.2767	No	Yes

As seen in Figure 4.1, BC-MPC takes a course maneuver to port and passes the target keeping a safe distance before getting back on the reference path. Figure 4.2 shows that the speed is lower while making the first course maneuver. The ownship then speeds up to compensate before settling at the reference speed. The speed is naturally decreasing while making such a maneuver, and the speed



(a) Speed. **(b)** Course.

Figure 4.2: Speed and course from overtaking scenario simulation.

maneuvers are not considered as unneseccary. BC-MPC solves this situation well in accordance with COLREGs Rule 8 and 13.

The scenario is more challenging for VO. The ownship oscillates around the reference line, slows down and makes some turns before it speeds up and passes the obstacle. This behavior is confusing, and can not be claimed as *positive, made in ample time and with due regard to the observance of good seamanship*. Hence, VO is not compliant with COLREGs Rule 8 in this scenario. Since ownship overtakes the vehicle and avoids collision, VO is considered compliant with Rule 13.

Table 4.6 shows the performance metrics from the simulations. The minimum distance to obstacle (MDO) is significantly higher with BC-MPC than with VO. Note that ownship enters the reference path ahead of the target with a better clearance with BC-MPC than with VO. This is because VO collapses the configuration space and treats the obstacle as a circular object, where the elliptical regions of BC-MPC has the major axis ahead of the obstacle.

Further, the course oscillations and slow speed of VO is not only in conflict with Rule 8, but gives higher values for TD, TT, IACR and IASR. IACR could be expected to be even higher considering the many course maneuvers. Figure 4.3 shows that the aboslute value of r is significantly higher for BC-MPC than for VO for about the first 60 seconds, which is when the overtaking takes part (figures 4.1-4.2). For the rest of the 133 seconds of the simulation with VO, the absolute value of r drops and hence the normalised value IACR is higher than the many course maneuvers indicate.

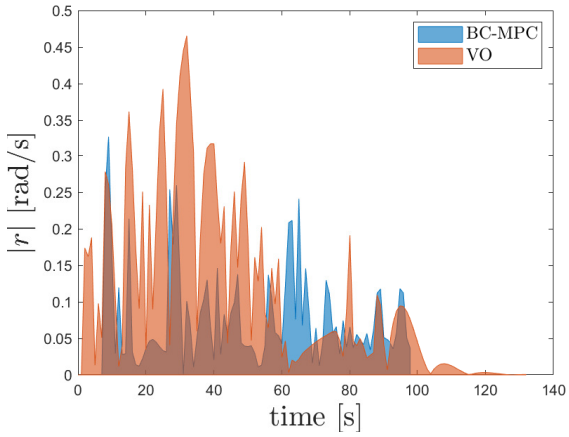


Figure 4.3: Absolute course rate from overtaking scenario simulation.

4.4.2 Head-on

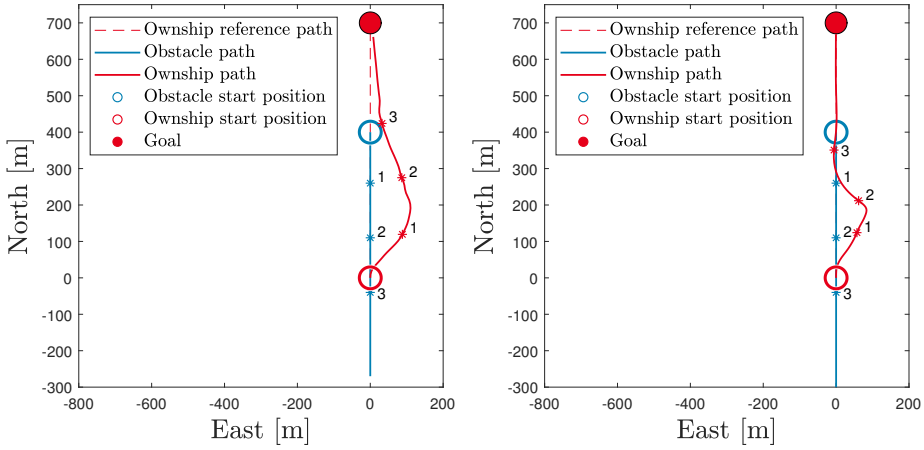
In this scenario, the ownship and the target have reciprocal courses and is on collision course. According to COLREGs rule 14, both vessels should alter their course to starboard and pass each other on the port side of each other. The target is not maneuvering in the simulations, but the results in Figure 4.4 shows that ownship keeps its obligations and alters to starboard in both cases in compliance with Rule 14.

Table 4.7: Results of head-on simulation. Shows the metrics and compliance to COLREGs Rule 8 (actions to avoid collision) and Rule 14 (actions in head-on situations).

Algorithm	TD	TT	MDO	IACR	IASR	Rule 8	Rule 14
BC-MPC	713 m	68 s	110 m	0.1071	0.1715	Yes	Yes
VO	723 m	77 s	83 m	0.0501	0.2674	Yes	Yes

There are still some differences worth noticing. The ownship makes observable maneuvers with BC-MPC and passes the target at a safe distance while keeping its speed in compliance with Rule 8.

With VO, the ownship also makes observable maneuvers. However, it slows down



(a) BC-MPC **(b)** VO

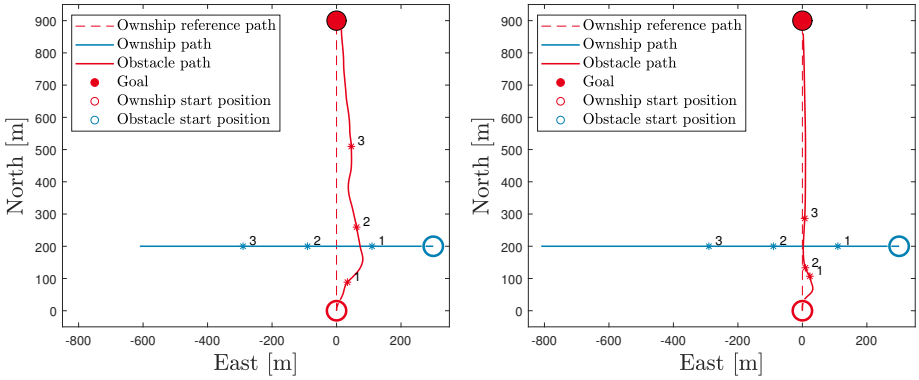
Figure 4.4: Head-on scenario simulations. The initial positions and the goal are marked with circles, and the three time instances $15s$, $30s$ and $45s$ are marked with star number 1, 2 and 3, respectively.

and passes the target at a very low speed, before it speeds up and heads back to the reference path. This behavior is unecessary, but since the ownship keeps its course it is still considered as compliant with Rule 8.

Table 4.7 shows the metrics from the simulations. TD and TT are higher for VO than for BC-MPC. IACR has the highest value for BC-MPC, and IASR has the highest value for VO. The reason for this is that BC-MPC avoids the collision mainly with course maneuvers, while VO perform speed maneuvers and slows down. Also, the trajectory alignment requires higher effort by BC-MPC, and increases IACR. MDO is higher with BC-MPC than for VO.

4.4.3 Crossing from Starboard

In this scenario, the target is crossing from starboard. According to Rule 15, the ownship should keep out of the way and avoid to pass ahead of the target.



(a) BC-MPC **(b)** VO

Figure 4.5: Crossing from starboard scenario simulations. The initial positions are marked with circles, and the three time instances $20s$, $40s$ and $60s$ are marked with star number 1, 2 and 3, respectively.

Table 4.8: Results of crossing from starboard simulations. Shows the metrics and compliance to COLREGs Rule 8 (actions to avoid collision) and Rule 15 (actions in crossing situations).

Algorithm	TD	TT	MDO	IACR	IASR	Rule 8	Rule 15
BC-MPC	905 m	92 s	78 m	0.0654	0.3238	Yes	Yes
VO	877 m	112 s	81 m	0.0356	0.3842	No	Yes

The simulation results are shown in Figure 4.5. BC-MPC makes a starboard maneuver of good observance, and passes abaft the target in compliance with Rule 8.

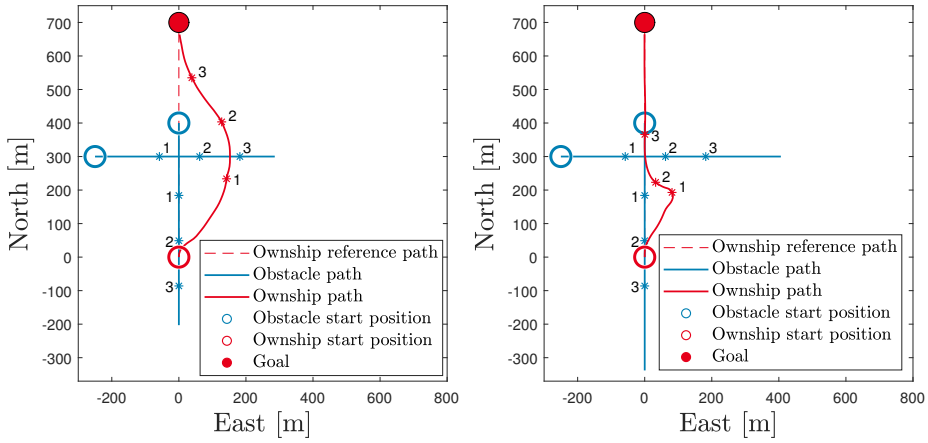
VO identifies the correct scenario and makes a starboard maneuver. However, VO slows down and makes a port maneuver towards the reference path waiting for the target to pass. This behavior can be confusing and potentially dangerous and is in conflict with Rule 8c which suggests course maneuvers to avoid a collision. VO is therefore considered compliant with Rule 15, but not with Rule 8.

Table 4.8 shows the metrics from the simulations. The speed-maneuver dominant behavior of VO gives lower values for TD and IACR and higher values for TT and

IASR, than with BC-MPC. MDO is a few meters shorter for BC-MPC than for VO. However, MDO appears abaft the target, and both algorithms stays outside the safety minor axis and ostacle radius, which is 75 m.

4.4.4 Head-on and Crossing from Port

In this scenario, the ownship faces two targets in head-on and crossing from port situations, respectively. According to COLREGs, ownship should make a starboard maneuver and keep out of the way for the head-on target. At the same time, it has a stand-on obligation to the crossing target. However, the targets are not maneuvering in this scenario and ownship might be required to make maneuvers as a part of the stand-on obligation in order to avoid collision and maintain safe maneuvering.



(a) BC-MPC **(b)** VO

Figure 4.6: Head-on and crossing from port scenario simulations. The initial positions are marked with circles, and the three time instances $25s$, $40s$ and $55s$ are marked with star number 1, 2 and 3, respectively.

Table 4.9: Results of head-on and crossing from port simulations. Shows the metrics and compliance to COLREGs Rule 8 (actions to avoid collisions), Rule 14 (actions in head-on situations) and Rule 17 (action by stand-on vessel).

Algorithm	TD	TT	MDO	IACR	IASR	Rule 8	Rule 14	Rule 17
BC-MPC	750 m	68 s	122 m	0.0917	0.2652	Yes	Yes	Yes
VO	724 m	83 s	82 m	0.0582	0.4372	No	Yes	No

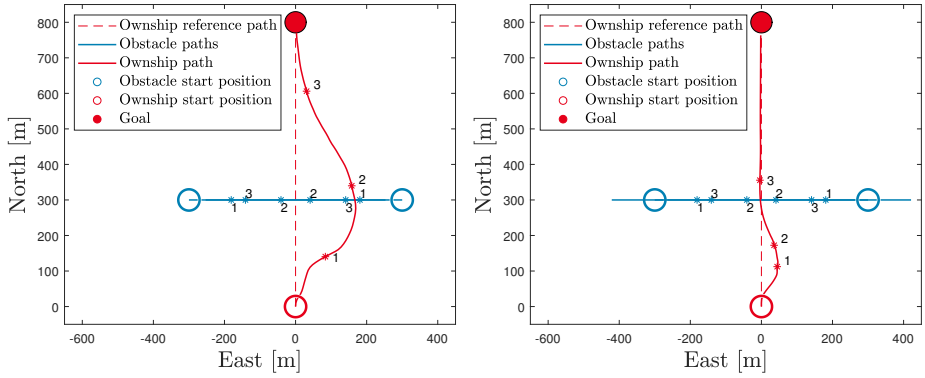
As seen in Figure 4.6, BC-MPC alters to starboard and pass the head-on vehicle at a safe distance. It avoids sudden changes in course and passes ahead of the crossing vehicle. Therefore, it keeps the obligations to both targets while acting in observable manners. The behavior is considerd to be in compliance with Rule 8, Rule 14 and Rule 17.

VO makes a starboard maneuver to pass the head-on vehicle on its port side, according to Rule 14. Then it slows down and makes a port maneuver to pass abaft the crossing vehicle before it speeds up and gets back on the reference path. This is in opposition to COLREGs Rule 17 c, which states that a stand-on vessel should not alter to port for a vessel on her port side if case admits. VO is therefore considered to be compliant with Rule 14, but not with Rule 8 and Rule 17.

Similarly to the previous scenarios, VO has a speed-maneuvering dominant behavior which results in lower values for TD and IACR and higher values for TT and IASR, as seen in Table 4.9. MDO is higher for BC-MPC as for the head-on scenario.

4.4.5 Crossing from starboard and port

In this scenario, the ownship faces two targets crossing from each side simultaneously. According to Rule 17 and Rule 15, it has a stand-on obligation to the vessel crossing from port, and a keep out of the way obligation to the vessel crossing from starboard.



(a) BC-MPC **(b)** BC-MPC

Figure 4.7: Crossing from starboard and crossing from port scenario simulations. The initial positions are marked with circles, and the three time instances 18s, 38s and 64s are marked with star number 1, 2 and 3, respectively.

Table 4.10: Results of crossing from starboard and crossing from port simulations. Shows the metrics and compliance to COLREGs Rule 8 (actions to avoid collisions), Rule 15 (actions in crossing situations) and Rule 17 (actions by stand-on vessel).

Algorithm	TD	TT	MDO	IACR	IASR	Rule 8	Rule 15	Rule 17
BC-MPC	870 m	80 s	90 m	0.0881	0.2120	Yes	Yes	Yes
VO	784 m	104 s	83 m	0.0384	0.3716	No	Yes	No

The simulation results are shown in Figure 4.7. BC-MPC solves this situation similarly as for the head-on and crossing from port scenario. The ownship makes starboard maneuvers to pass abaft the target crossing from starboard, and ahead of the target passing from port. The ownship keeps its speed and course and avoids uneseccary maneuvers while passing at a safe distance. The behavior is in compliance with Rule 8, Rule 15 and Rule 17.

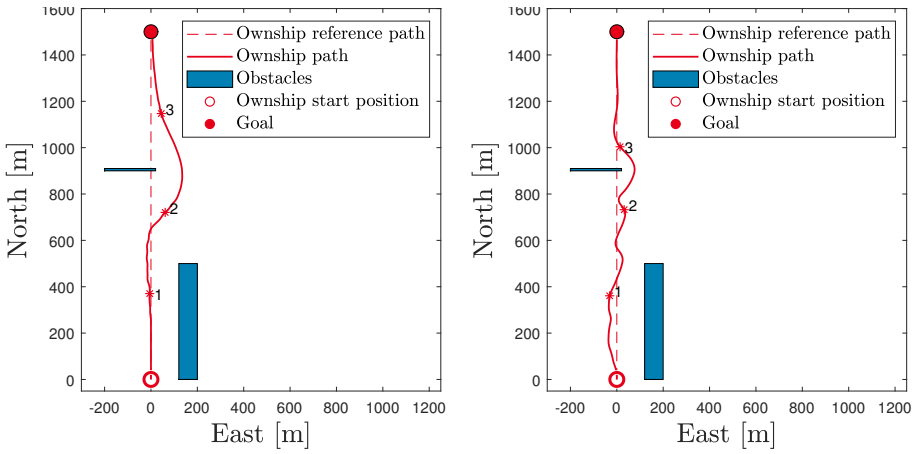
VO makes a starboard maneuver to pass abaft the target crossing from starboard. However, before the target is passed, the ownship slows down and makes a port maneuver. The ownship faces the targets while they pass before speeding up and heading back at the reference path. This is confusing and can be dangerous. The behavior is not compliant to neither Rule 8 nor Rule 17. Since the target crossing

from starboard is passed abaft, it is considered to be compliant with Rule 15.

The metrics in Table 4.10 are similar to the other crossing scenarios, with lower TD and IACR and higher TT and IASR values for VO. MDO is a few meters higher for BC-MPC than for VO.

4.4.6 Static obstacle scenario A

In this scenario, the ownship follows a reference path with one static obstacle ahead and one on its starboard side, making the shortest path going between the obstacles. There are no other vessels in this scenario, and COLREGs compliance is not considered.



(a) BC-MPC **(b)** VO

Figure 4.8: Static scenario A simulations. The initial positions are marked with circles, and the three time instances 40s, 80s and 120s are marked with star number 1, 2 and 3, respectively.

Table 4.11: Simulation results, static scenario A.

COLAV	TD	TT	MDO	IACR	IASR
BC-MPC	1540 m	150 s	111 m	0.0839	0.2868
VO	1556 m	167 s	55 m	0.0631	0.1432

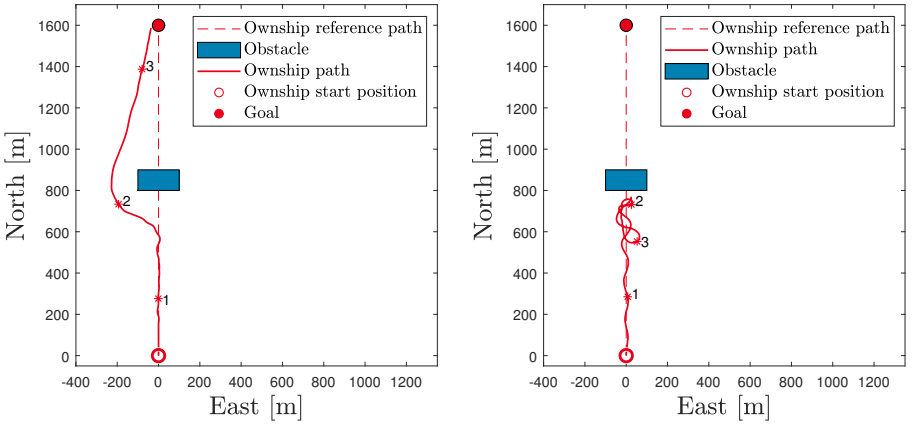
With BC-MPC, the ownship follows the reference path before making a starboard maneuver to avoid the second obstacle. The ownship keeps a safe distance to both obstacles and avoids uneseccary maneuvers.

With VO, the owships makes several turns before slowing down and passing the second obstacle. COLREGs compliance is not considered in this scenario, but the behavior of VO can still be confusing for other vessels and is not desirable.

As seen in Table 4.11, BC-MPC keeps a higher distance to the obstacles than VO. This is explained by the BC-MPC gradient border of 100 m around the obstacles for BC-MPC, and the 50 m border of VO. IACR and IASR are higher for BC-MPC than for VO. However, BC-MPC makes a bigger turn and maneuvers the speed to get back on track, resulting in lower values than VO for TD and TT.

4.4.7 Static obstacle scenario B

In this scenario, ownship is following a straight reference path blocked by a static obstacle.



(a) BC-MPC **(b)** VO

Figure 4.9: Static scenario B simulations. The initial positions are marked with circles, and the three time instances $30s$, $100s$ and $150s$ are marked with star number 1, 2 and 3, respectively.

Table 4.12: Simulation results, static scenario B.

COLAV	TD	TT	MDO	IACR	IASR
BC-MPC	1731 m	166 s	115 m	0.0787	0.3241
VO	1468 m	184 s	40 m	0.1627	0.3585

BC-MPC solves this scenario well. Ownship follows the reference path before making a port maneuver and passing the obstacle with a safe distance.

VO has more problems, and the ownship makes several turns and circles before entering the boundary around the obstacle and failing to find admissible velocities. This illustrates the limitations of VO due to the lack of consideration of vessel dynamics.

The matrics in Table 4.12 shows better values for BC-MPC than for VO for all the metrics except for TD. However, the ownship never reached the goal with VO, which explains the low TD. It could be discussed whether this scenario is realistic fro short term COLAV or if it chould be solved at the path planning level. That aside, the scenario illustrates the important limitation of VO not

considering dynamics.

4.4.8 Summary of Results Without Noise

For the scenarios described in sections 4.4.1 - 4.4.7, BC-MPC and VO solves the scenarios with variable success. The only scenario where one of the algorithms fails is the static B scenario for VO, which demonstrates the limitations of VO not considering vessel dynamics. The quantitative metrics and COLREGs compliance is summarized and discussed in this subsection.

Table 4.13: Simulation results: metrics. The table shows the metrics from the simulations described in Section 4.4.1 - 4.4.7. The bold values are the values considered best for each scenario.

Scenario	Algorithm	TD	TT	MDO	IACR	IASR
O	BC-MPC	**1021 m**	**99 s**	**109 m**	**0.0733**	**0.2133**
	VO	1057 m	133 s	64 m	0.1062	0.2767
HO	BC-MPC	**713 m**	**68 s**	**110 m**	0.1071	**0.1715**
	VO	723 m	77 s	83 m	**0.0501**	0.2674
CS	BC-MPC	905 m	**92 s**	78 m	0.0654	**0.3238**
	VO	**877 m**	112 s	**81 m**	**0.0356**	0.3842
HO+CP	BC-MPC	750 m	**68 s**	**122 m**	0.0917	**0.2652**
	VO	**724 m**	83 s	82 m	**0.0582**	0.4372
CS+CP	BC-MPC	870 m	**80 s**	**90 m**	0.0881	**0.2120**
	VO	**784 m**	104 s	83 m	**0.0384**	0.3716
Static A	BC-MPC	**1540 m**	**150 s**	**111 m**	0.0839	0.2868
	VO	1556 m	167 s	55 m	**0.0631**	**0.1432**
Static B	BC-MPC	1731 m	**166 s**	**115 m**	**0.0787**	**0.3241**
	VO	**1468 m**	184 s	40 m	0.1627	0.3585

The metrics from the simulations are summarized in Table 4.13. BC-MPC has a lower travel time (TT) than VO in all scenarios. The minimum distance to obstacle (MDO) is significantly lower for BC-MPC than VO in all scenarios except for crossing from starboard, where the ownship passes abaft the target. This shows the strenght of the elliptical regions and the major axis ahead of the obstacle. IACR is in general lower for VO, and IASR lower for BC-MPC. The values

for TD, TD, IACR and IASR are affected by the course-manuevering dominant behavior of BC-MPC and the speed-maneuvering dominant behavior of VO. VO does not have the same sense of time as BC-MPC, and this explains the high values of TT and IASR. Furthermore, IACR are affected by the different trajectory alignment behavior of the respective algorithms.

Table 4.14: Simulation results: COLREGs compliance. Summary of the COLREGs results of the simulations in sections 4.4.1 -4.4.7 with respect to Rule 8 (actions to avoid collision), rule 13-15 (actions in overtake, head-on and crossing situations) and Rule 17 (actions by stand-on vessel).

Scenario	COLAV	Rule 8	Rule 13-15	Rule 17
O	BC-MPC	Yes	**Yes**	-
	VO	No	**Yes**	-
HO	BC-MPC	**Yes**	**Yes**	-
	VO	**Yes**	**Yes**	-
CS	BC-MPC	**Yes**	**Yes**	-
	VO	No	**Yes**	-
HO+CP	BC-MPC	**Yes**	**Yes**	**Yes**
	VO	No	**Yes**	No
CS+CP	BC-MPC	**Yes**	**Yes**	**Yes**
	VO	No	**Yes**	No

From the results listed in Table 4.14, both algorithms are compliant with COLREGs Rule 13-15 for the simulated scenarios. For Rule 8 and Rule 17, which includes more qualitative and situation dependent guidelines, VO is only considered COLREGs compliant in the head-on scenario. However, it can be discussed whether the speed maneuvers with VO in the head-on scenario are in conflict with Rule 8c which suggests to avoid collision with course maneuvers.

The conflicts with Rule 8 are mainly caused by the oscillating and speed-maneuvering behavior. The conflicts with Rule 17 is affected by the hard constraints and the simple cost function of VO. Of the velocities in the admissible search spacet, VO will pick the one going towards the reference, without any other evaluation of the situation. This shows the strength of BC-MPCs consideration of time and translational cost.

The only scenario where VO is considered to be COLREGs compliant with respect

to all of the considered Rules, is the head-on scenario. In this scenario, BC-MPC is also COLREGs compliant and has better values for most of the metrics. Hence, BC-MPC performs better than VO in all the simulated scenarios with respect to the considered metrics and COLREGs Rules.

4.5 Simulations with Measurement Noise

The head-on and crossing from starboard scenarios are now simulated as Monte Carlo simulations with measurement noise, using the same randomly generated seed for both COLAV algorithms.

COLREGs compliance is not considered as a quantitative metric for these scenarios, but the metrics presented in Section 4.2 are calculated and used in the evaluation of the results together with the passing side of the target and if the simulation fails.

4.5.1 Head-on

The head-on scenario in Section 4.4.2 is now simulated 300 times with measurement noise. In accordance with COLREGs rule 14, the ownship has a keep out of the way obligation and should make a starboard maneuver to pass on the target's port side. The results in Table 4.15 show that both algorithms pass on the port side in compliance with COLREGs most of the time. VO fails to find admissible velocities 7 of 300 times.

Table 4.15: Simulation results head-on. Note that passing starboard refers to passing on the starboard side of the target, i. e. on the target's port side.

Algortihm	Algorithm Fail	Passing starboard	Passing port
BC-MPC	0	299	1
VO	7	293	0

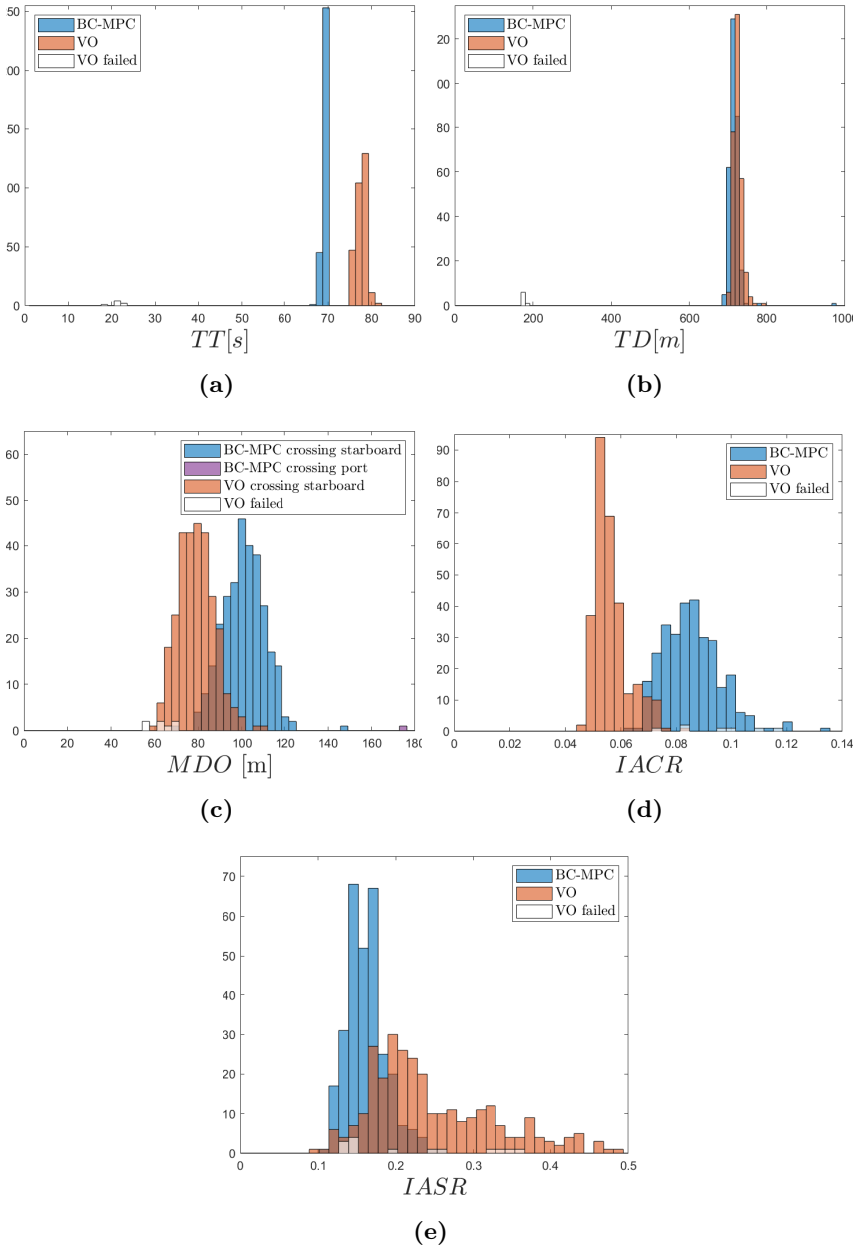


Figure 4.10: Simulation results of head-on with measurement noise

Figure 4.10 shows the performance metrics from the simulations. Both BC-MPC and VO shows similar behavior to the simulations without measurement noise in the successfull simualtions. However, an important result from this simulation is that VO fails in some scenarios. When the velocity cone is entered, VO struggels to find a way out and hence often fails of find admissible velocities. The reason for VO entering the velocity cone in the first place is mainly because of the lack of consideration of dynamics. The ownship tries to reach an infeasible state, but ends up in the velocity obstacle. Second, as VO considers only the next timestep, and not where to go further, it risks to end up in a state where there is no way out. Last, the problems with these weaknesses increase further with the noisy obstacle measurements. The ownship typically travels close to the velocity obstacle, and with noisy measurements affecting the velocity obstacle, the ownship is vulnerable when travelling close to a noisy, hard constraint.

For the successful scenarios, VO has about 10% longer travel time (TT) than BC-MPC, but travels only marginally longer distances. The average speed is, therefore, lower with VO. The one case where BC-MPC has a travel distance of almost 1000 m is when it passes the target on its starboard side.

BC-MPC keeps a longer distance to the target than VO, and for the simulations where BC-MPC passes on the target's starboard side, MDO is significantly higher.

BC-MPC has a higher value for IACR, and VO for IASR. This is similar to the results from the simulations without noise. The IACR values are higher with VO in the simulations that fails than in the succeeding simulations, and this is discussed further in Section 4.5.2.

4.5.2 Crossing from Starboard

The crossing from starboard scenario from Section 4.4.3 is now simulated 300 times with measurement noise. The results in Table 4.16 shows that VO fails 44 times. According to COLREGs Rule 15, the ownship has a keep out of way obligation and should make a starboard maneuver to pass abaft the target. VO passes abaft the target in all the successful simulations. BC-MPC passes abaft 279 times, and ahead of the target 21 times.

Table 4.16: Simulation results crossing from starboard

Algorithm	**Algorithm fail**	**Passing abaft**	**Passing ahead**
BC-MPC	0	279	21
VO	44	266	0

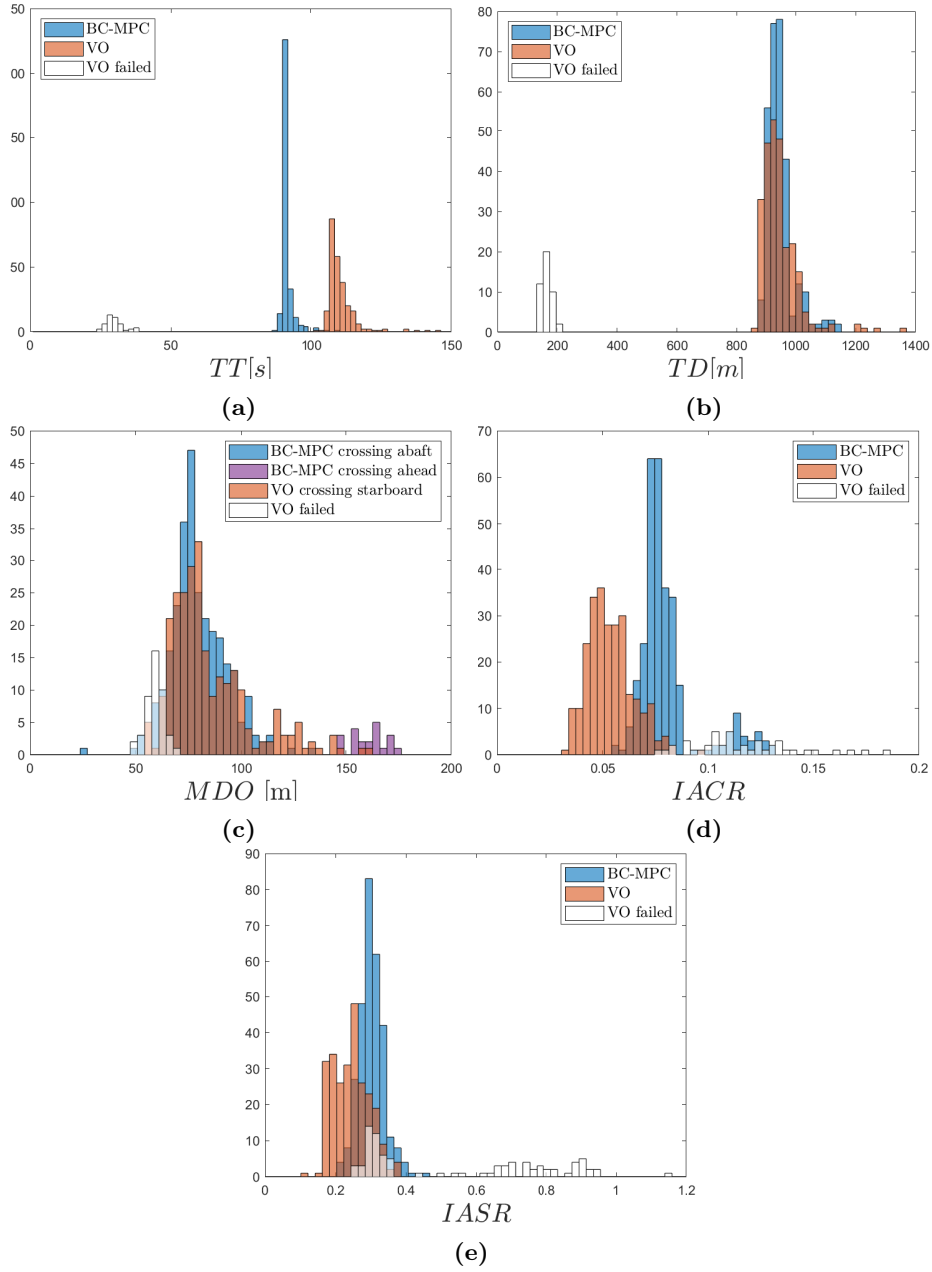


Figure 4.11: Simulation results of crossing from starboard with measurement noise. The simulations with VO where the algorithm fails are shown in white.

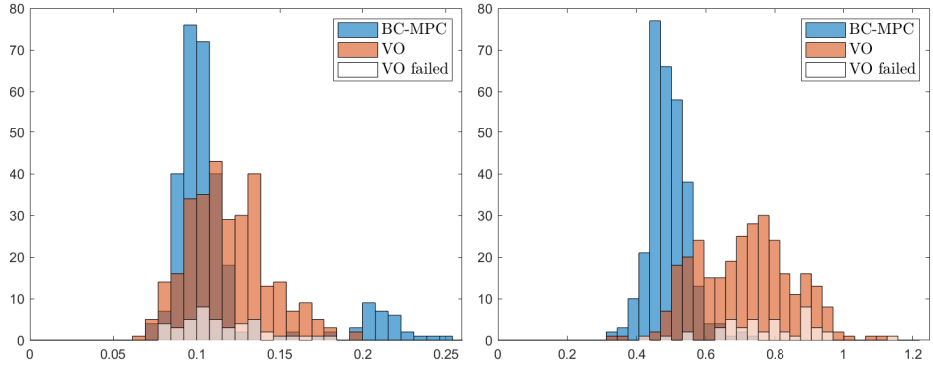
The weaknesses of VO discussed in realtion to the simulations of the head-on scenario with noise in Section 4.5.1 are even more demanding in this scenario, especially when the ownship tries to pass ahead of the target. As seen in Table 4.16, there are no such succesfull passes.

Figure 4.11 shows the performance metrics from the simulations. Both BC-MPC and VO shows similar behavior to the simulations without measurement noise in the successfull simualtions. Ownship travels about the same distance for both algorithms, and the travel time (TT) is about 10% higher with VO than with BC-MPC.

MDO does in general coincides with the safety region minor axis for BC-MPC and the obstacle radius for VO. When the owship passes ahead of the target, MDO is bigger, and coincides with the safety major axis. There is one case where MDO is 27 m for BC-MPC. Ownship is here entering the safety region but stays outside of the collision region which has a minor axis of 25 m. The higher values for MDO when BC-MPC acts in conflict with rule 15 emphasize the capability of BC-MPC to keep a safe behavior.

BC-MPC has higher values for IACR an IASR than VO, except for the scenarios were VO fails. To investigate these results further, IACR and IASR are calculated considering only the first 35 seconds of the simulation, which is about the time most scenarios fails. The results in Figure 4.12 shows that the scenarios that fails now has average values for VO.

As seen in the overtaking (section 4.4.1) and other scenarios, VO requires less effort for trajectory alignment, which happens last in the simulation. The high values of IACR and IASR for the failed scenarios are therefore explained by the lack of this part which lowers the values for VO significantly. Also, note that BC-MPC performs better than VO in terms of IACR and IASR considering the first 35 seconds, but has higher values for the full simulation because of the higher effort required for trajectory alignment.



(a) **(b)**

Figure 4.12: IACR and IASR for the first 35 seconds of crossing from starboard with measurement noise.

4.5.3 Summary of Results with Noise

Both VO and BC-MPC show to some level similar behavior to the noiseless simulations in the presence of measurement noise. The most important result is that the VO algorithm fails in a certain number of the simulations. However, there are also some other results worth to discuss.

BC-MPC acts in a few cases in conflict with COLREGs Rule 13-15, and passes the target on port side. However, in all cases where this happens, ownship keeps a higher distance to the target and avoids confusing maneuvers. Also, keep in mind that BC-MPC is designed to obey these rules if required for safe behavior. There is only one simulation where BC-MPC makes a dangerous maneuver, and that is the case with the low minimum distance to the target described in Section 4.5.2. Hence, BC-MPC succeeds in both collision avoidance and safe behaviour in accordance to good seamanship and COLREGs also in the simulations with noisy obstacle measurements.

VO has more problems keeping a safe behaviour with the noisy obstacle measurments. The algorithm fails in about 2% of the head-on simulations and about 16% of the crossing from starboard simulations. This is a significant number of fails and can cause many dangerous situations. VO is less robust towards the noise, and this is affected by the hard constraints, the short considered time horizon and the lack of vessel dynamics consideration.

BC-MPC has in general higher values than VO for IACR and IASR. However, further investigations showed that this is affected by the trajectory alignment part and that BC-MPC has the lowest values when considering only the first part.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this report, the two collision avoidance (COLAV) algorithms branching-course MPC (BC-MPC) and velocity obstacle (VO) are evaluated and compared for ASVs. This is done in terms of COLAV, safe behaviour in accordance to COLREGs and a set of performance metrics. To achieve this, a simulator is implemented in MATLAB/SIMULINK considering the Telemetron high-speed ASV model. Several scenarios inculding single-obstacle, more complex multi-obstacle and static obstacle scanerios are simulated. In addition, simulations of two single-obstacle scenarios are performed as Monte Carlo simulations with measurement noise.

The results shows that BC-MPC in general performs better than the implemented version of VO, especially with respect to COLREGs compliance and with noisy obstacle estimates. Both algorithms acts in conflict with COLREGs in some scenarios. However, where VO acts in conflict with Rule 8 and shows dangerous behavior, BC-MPC is only in conflict with rule 13-15 and maintain safe behaviour. Hence, it can be concluded that BC-MPC has success in acting safe and in compliance with COLREGs in the performed simualtions, whereas VO does not.

Where BC-MPC has a course-maneuvering dominant behaviour, VO has in opposition a speed-dominiant behaviour. The main reason for this is that VO has

no sence of time. Hence, the ownship slows down often without taking in when faling behind. Therefore, VO has a lower travel time (TT) than BC-MPC in all scenarios. This speed maneuvering dominant behavior of VO gives not only higher values for the integral of absolute speed rate (IASR), but is also in conflict with Rule 8 which suggests avoiding collisions with course maneuvers.

The minimum distance to obstacle (MDO) is in general higher for BC-MPC than for VO, except for the crossing scenarios where the ownship passes abaft the target. The capability of BC-MPC to keep a longer distance when passing ahead of a target shows together with the good COLREGs compliance the strenghts of the elliptical regions.

Considering the integral of absolute course rate (IACR), BC-MPC has in general higher values than VO. However, this is affected by the trajectory alignement effort, and BC-MPC shows better performance in terms of IACR when considering only the part of the simulations where the collision avoidance takes part. This is important, as this is the where it is critical to keep the course and avoid confusing behaviour.

The performance with noisy obstacle estimates gives important results. Where BC-MPC handles noise well, VO has problems with algorithm failure. When the velocity obstacle is entered, the owship has often problems with getting out. These problems are related to the reactive characteristics of VO, if VO does not see a way out within the next timestep, it does not see any at all. The main reasons of why the ownship enters the velocity obstacle is the missing consideration of vessel dynamics, which introduces a severe limitation of VO.

To sum up, it can be concluded that BC-MPC shows excellent performance in terms of the scope of this report. Especially the robustness towards noise and the good COLREGs compliant behaviour is worth mentioning. This is in contrast to VO, which shows some limitations in the performance of these terms.

5.2 Future Work

This report considers only a limited selection of scenarios and simulations. To evaluate the algorithms further, the following are suggested:

- Further evaluate the robustness towards noise through extensive simulations of complex scenarios with measurement and/or wave noise.

- Evaluate the performance in situations with maneuvering obstacles. This could be both passively maneuvering obstacles, or obstacles performing active COLAV maneuvering with VO, BC-MPC or other COLAV algorithms
- Evaluate the performance for vessels with a different dynamic than the Telemetron ASV
- Evaluate and compare the algorithms with a different tuning or implementation of VO to get more similar characteristics of BC-MPC and VO. This could be
  - an uncertain boundary for noise robustness
  - dynamic vessels constraints added to the velocity obstacle, or simply with a smaller search space
  - hysteresis on the desired speed

Bibliography

[1] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[2] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998.

[3] B.-O. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig, "A modified dynamic window algorithm for horizontal collision avoidance for AUVs," in *2016 IEEE Conference on Control Applications (CCA)*, IEEE, Sep. 2016.

[4] B.-O. H. Eriksen, E. F. Wilthil, A. L. Flåten, E. F. Brekke, and M. Breivik, "Radar-based maritime collision avoidance using dynamic window," in *Proc. of the 2018 IEEE Aerospace Conference*, (Big Sky, Montana, USA), Big Sky, Montana, USA, 2018, pp. 1–9.

[5] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime navigation with COLREGS using velocity obstacles," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Sep. 2011.

[6] ——, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, Jan. 2014.

[7] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, IEEE, May 2008.

[8] D. K. Kufoalor, E. F. Brekke, and T. A. Johansen, "Proactive collision avoidance for ASVs using a dynamic reciprocal velocity obstacles method," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2018.

[9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*, Springer New York, 1986, pp. 396–404.

[10] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sep. 1989.

[11] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, IEEE Comput. Soc. Press.

[12] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments," in *Proceedings., IEEE International Conference on Robotics and Automation*, IEEE Comput. Soc. Press.

[13] S.-M. Lee, K.-Y. Kwon, and J. Joh, "A fuzzy logic for autonomous navigation of marine vehicle satisfying colreg guidelines," *International Journal of Control, Automation, and Systems*, vol. 2, Jun. 2004.

[14] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[15] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[16] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[17] B.-O. H. Eriksen and M. Breivik, "MPC-based mid-level collision avoidance for asvs using nonlinear programming," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, Aug. 2017.

[18] B.-O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "The branching-course MPC algorithm for maritime collision avoidance," 2019, Submitted to Journal of Field Robotics.

[19] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, Apr. 2011.

[20] SNAME, "Nomenclature for treating the motion of a submerged body through a fluid," 1950.

[21] B.-O. H. Eriksen and M. Breivik, "Modeling, identification and control of high-speed asvs: Theory and experiments," in *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, T. I. Fossen, K. Y. Pettersen, and H. Nijmeijer, Eds. Cham: Springer International Publishing, 2017, pp. 407–431.

[22] M. Breivik and T. I. Fossen, "Guidance laws for autonomous underwater vehicles," in *Underwater Vehicles*. InTech, Jan. 2009.

[23] S. M. LaValle, "Motion planning," *IEEE Robotics & Automation Magazine*, vol. 18, no. 1, pp. 79–89, Mar. 2011.

[24] I. M. Organization, Ed. (). Convention on the international regulations for preventing collisions at sea, 1972 (colregs), [Online]. Available: `http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx` (visited on 06/18/2019).

[25] T. Stenersen, "Guidance systems for autonomous surface vehicles," Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2015.

[26] B.-O. H. Eriksen and M. Breivik, "Short-term ASV collision avoidance with static and moving obstacles," Sumbitted to IFAC CAMS, 2019.

[27] Wikipedia. (Jun. 17, 2019). Wiener process, [Online]. Available: `https://en.wikipedia.org/wiki/Wiener_process`.