NTNU
Norwegian University of
Science and Technology

Collision Avoidance for Autonomous Surface Vehicles Using Velocity Obstacle and Set-Based Guidance.

Helene Myre

Master of Science in Cybernetics and Robotics
Submission date: June 2016
Supervisor: Edmund Førland Brekke, ITK
Co-supervisor: Signe Moe, ITK
Bjørn Olav Holtung Eriksen, ITK
Øystein Engelhardtsen, DNV GL

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

A good and reliable collision avoidance method is essential when operating Autonomous Surface Vehicles (ASVs) at sea. But even though avoiding collisions is the most important part, making sure the ASV's evasive maneuvers makes sense to surrounding vessels, is also a vital part of a well preforming system. In order to both avoid collisions and ensure reasonable behavior, the collision avoidance method used should be implemented according to the international regulations for preventing collisions at sea, also known as COLREGs.

Set-Based Guidance is used as a way of achieving a desired behavior in a robot and can, amongst other things, also be used as a collision avoidance method. In this thesis, a collision avoidance method for dynamic obstacles based on Set-Based Guidance has been developed. The method is also compliant with the main rules of COLREGs.

In order to test the collision avoidance method, a simulator with a 3 DOF mathematical model of a vessel has been implemented together with LOS guidance and a low-level controller. Simulations were preformed with the different situations that can occur between the ASV and an obstacle: head on, overlapping and crossing, as well as with a composed scenario including more than one obstacle.

In order to benchmark the resulting method, another collision avoidance method, the Velocity Obstacle, was also implemented and simulated with the same system and situations. Both Velocity Obstacle and Set-Based Guidance have their strengths and weaknesses, but Set-Based Guidance showed a lot of potential when it came to taking substantial action during an evasive maneuver, as well as moving in straight lines, something that are both desirable behavior according the main COLREGs rules.

Sammendrag

En god og pålitelig kollisjonsunngåelsesmetode er essensielt når man betjener autonome overflatefartøy (ASVer) på åpent hav. Men selv om kollisonsunngåelse er den viktigste biten, så er det å opptre forutsigbart ovenfor omkringliggende fartøy også en avgjørende del for å oppnå et velfungerende system. For å både unngå kollisjoner og samtidig opptre fornuftig bør kollisjonsunngåelsesmetoden som brukes implementeres med hensyn til de internasjonale reglene for kollisjonsunngåelse på havet (COLREGs).

Set-Based Guidance (SBG) er brukt for å oppnå ønsket oppførsel hos en robot og kan blant annet også brukes som en kollisjonsunngåelsesmetode. I denne masteroppgaven vil det bli utviklet en kollisjonsunngåelsesmetode for dynamiske hindre, basert på SBG, og som også er i tråd med hovedreglene til COLREGs.

For å teste kollisjonsunngåelsesmetoden så har en simulator med tre frihetsgrader blitt implementert sammen med LOS styring og en lavnivåkontroller. Simuleringene har vært gjennomført med de ulike situasjonene som kan oppstå mellom en ASV og et hinder: front mot front, forbikjøring og kryssing, i tillegg til et sammensatt tilfelle med mer enn ett hinder.

For å kunne drøfte kvaliteten på den ferdige metoden ble en annen kollisjonsunngåelsesmetode, kalt Velocity Obstacle (VO), implementert og simulert med det samme systemet og de samme situasjonene. Både VO og SBG hadde sterke og svake sider, men SBG viste stort potensiale når det kom til å utføre klare og tydelige handlinger i løpet av unnamanøvre, i tillegg til å sørge for at ASVen bevegde seg i rette linjer, noe som begge er ønskelig oppførsel når det kommer til hovedreglene til COLREGs.

Preface

This thesis is written as a compulsory part of the Master's degree in Engineering Cybernetics at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU).

I would like to thank my supervisors Edmund Førland Brekke, Signe Moe and Bjørn-Olav Holtung Eriksen for all their great help and guidance throughout this semester. I would also like thank all my friends and fellow students for these five challenging, but amazing years at NTNU.

Helene Myre
Trondheim, June 2016

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In today's society, automated systems are becoming a more and more important part of our every day life. For several decades, robots have been seen replacing manual labor in the industry, and in more recent history even at home with automated lawn mowers and vacuum cleaners. Automated vehicles such as driverless cars and self driven drones have been under extensive research the last decade and still are. In this thesis, a Autonomous Surface Vehicles (ASVs) is going to be the basis.

According to research presented at MARTECH 2004 [4] approximately 50 % of accidents at sea are results of human errors and an additional 30% of the accidents should have been picked up on and prevented by humans, but were not. These numbers gives a good indication of the advantage it may be to remove humans from the equation. The use of ASVs would make the sea a much safer place. Deaths, injures and damages could be prevented as well as making transportation by sea cheaper due to less need of staff on board the ship. For this to become a reality, trustworthy and effective algorithms must be proven to steer the vessel safely towards the desired location, as well as move in a manner that makes ones intentions clear to surrounding vessels. The key building block for accomplishing this is a good and reliable collision avoidance method which performs according to the COLREGs rules .

1.2 Contribution

Set-Based Guidance is used to generate references, desired trajectories or desired velocities for a robotic system that, if satisfied, will result in a specified, desired behavior [5]. It prioritizes tasks and allows all of them to be performed at the same time by projecting task velocities from lower prioritized tasks through the null space of higher ones. With the use of set-based tasks, Set-Based Guidance can be used as a collision avoidance method for static obstacles [1]. Obstacle avoidance is the prioritized task expressed as the distance between the ASV and the obstacle and the solutions is the entire set that is not occupied by the obstacle or any safety buffer surrounding it. But when using an ASV, most of the obstacles are not going to be static, but dynamic ones in the shape of other vessels. Therefor, being able to avoid dynamic obstacles is crucial in other to have a useful collision avoidance method.

In this thesis, a collision avoidance method, able to handle dynamic obstacles and based on Set-Based Guidance, has been developed. Previously, it has been suggested that Set-Based Guidance could be used as a collision avoidance method [1], but no solutions as to how to handle dynamic obstacles has been presented. Because of this, the main contribution in this thesis is a detailed suggestion on how to adapt Set-Based Guidance in order to handle dynamic obstacles, as well as making the method compliant with the main rules of COLREGs. In order to evaluate the resulting method, another collision avoidance method, known as Velocity Obstacle, has been compared to Set-Based Guidance

1.3 Outline

Chapter 2 presents a brief literature review on the collision avoidance and path planning methods that have dominated the field in the later years, while Chapter 3 covers the needed background theory. In Chapter 4, a detailed explanation on the approach used to develop and implement the collision avoidance method for dynamic obstacles based on Set-based Guidance is given. Chapter 5 offers the details concerning the implementation of the system. In Chapter 6, all the simulation results are presented, as well as some in dept analyses of some of the results. The results are presented by figures in the thesis, but can also be seen as videos on YouTube[1]. A discussion of the results is given in Chapter 7 and a conclusion on the behavior can be found in Chapter 8. In the end, some pointers on how to perhaps improve the resulting method has been given in Chapter 9.

[1] https://www.youtube.com/playlist?list=PL8iPY5FBK9hH2edjxISr7Jg5EibDhHbuQ

Chapter 2

Review on Collision Avoidance and Path Planning Methods

Guidance and collision avoidance have undergone extensive research in the later years. Many different methods exists together with a number of different version and combinations of the methods. This chapter is going to be a review of some of the methods that have dominated the field in the later years.

Local and global methods
As mentioned above, guidance and collision avoidance can be achieved with the use of several different methods. These methods can be divided into two main groups; local and global. Local methods only looks at a certain, confined space around the ASV and the collision avoidance algorithms can only react to the obstacles within this space. Velocity Obstacle, Set-Based Guidance, Vector Field Histogram and Dynamic Window Approach are all local methods. A global method acts more like a path planner than a collision avoidance method. One looks at the entire configuration space (see section 3.3.2), where all obstacles are known, and find an obstacle free path from current to desired position. Examples of global methods are A* and Rapidly-Exploring Random Trees (RRT).

Actions made by a local method is only based on the data available at that exact moment. This makes local methods both computationally inexpensive and fast to adapt to sudden changes in the surrounding environment [6]. A drawback with local methods is that they only look at a limited area around the ASV and therefore carry the risk of getting stuck in a local minima. An example of this can be seen in Figure 2.1, where the red square illustrates the area seen by the local method. The ASV passes between what are believed to be two separate obstacles but is really just one obstacle with a wide gap that traps the ASV. The global methods do not risk getting trapped like this, but they are more computationally expensive and can not react to sudden changes at well as local methods.

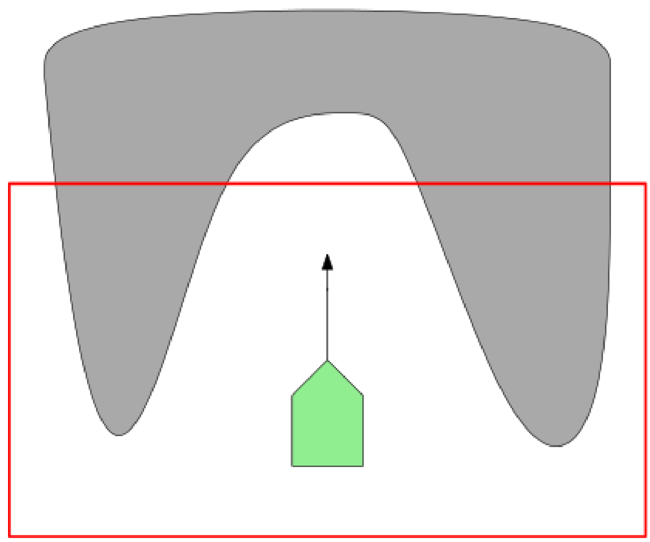

Figure 2.1: The difference between global and local methods, red box is illustrating the area of the local method.

2.1 Vector Field Histogram (VFH)

Borenstein and Koren suggested in 1991 a method of collision avoidance for mobile robots called Vector Field Histogram (VFH) [7], [8]. It was presented as a solution on the fundamental issues of the Potential Field methods, presented by Khatib in 1985 [9]. The methods will not be discussed any further in this thesis, but the issues can be studied in the article by Koren and Borenstein from 1991 [10]. VFH consists of three main steps. The first is to produce a two-dimensional Cartesian histogram grid that represents the static, real-time environment surrounding the robot. The second step is to transform the first histogram into a one-dimensional polar histogram, by dividing the surrounding area into pie shaped sectors representing the different directions and calculating the obstacle density in each of them. In the last step, the direction with an obstacle density lower that a certain threshold and closest to the desired goal is the one chosen.

After the initial presentation of VFH, modified versions has been published under different names. The first updated version was published by Borenstein and Ulrich in 1998 and was named VFH+ [11]. A second was named VFH* and presented in 2000 by the same authors [12].

2.2 Dynamic Window Approach (DWA)

In 1997, Fox, Burgard and Thrun presented the Dynamic Window Approach (DWA); a method for collision avoidance for mobile robots [13]. The Dynamic Window is a reduced velocity space containing only velocities that are reachable within a short amount of time and that ensures a path where the robot is able to stop safely. An objective function is determined by combining the velocity of the robot, the progress towards the goal and the distance between the robot and the obstacle. It results in a balance between keeping a high speed towards the goal and performing evasive maneuvers to avoid obstacles. The translational and rotational velocities are chosen by maximizing the objective function. Since most robots have some kind of constraints on velocity and acceleration, DWA is especially nice since it handles this naturally.

2.3 A*

A* was published by Hart, Nilsson and Raphael in 1968 [14], [15]. It is a path finding algorithm that finds all available paths between present and desired position. The different paths are assigned different costs based on things like distance or travel time and the one with the least cost is chosen. For this algorithm to work, the operating space needs to be represented by a graph, where the next step from each node can only be its neighboring nodes. A* uses a heuristic technique by first finding an estimate of the cheapest path, then it works its way out from the estimate to find the path closest free of obstacles. In a path finding situation, the estimate is typically the straight line between the position and the goal.

2.4 Rapidly-Exploring Random Trees (RRT)

Rapidly-Exploring Random Trees (RRT) was introduced by LaValle and Kuffner Jr. in 2001 [16], inspired by a paper by LaValle from 1998 [17]. RRT is based on the concept of random out-branching. The root of the tree is the starting point. For every step, random points are placed in the search space and lines are drawn between the points and the nearest part of the tree. If the lines lies entirely inside the search space, the line, or at least part of the line, will become the new branch. The length of new branches added are often limited by a growth factor. If the length of a line surpasses this limit, the new branch tip lies on the line, with a length equal to the growth factor. If not, the random point is the new branch tip. The tree is going to grow rapidly and in the end hit the goal, which is going to produce several possible paths between start and goal. According to Wikipedia, RRTs is in most cases modified to find a solution faster by increasing the likelihood of the random points hitting a certain area, for example the area around the goal, to make sure of rapid growth towards it [18].

2.5 Velocity Obstacle (VO)

Velocity Obstacle (VO) was first published by Fiorini and Shiller in 1998 [19]. VO is a motion planning method for robots in dynamic environments. It is based on finding all velocities for which the robot is going to collide with an obstacle some time in the future and therefore should be avoided. These velocities will make restricted areas in the velocity space shaped like cones, one for each obstacle. By selecting velocities outside of these cones, a collision is certain to be avoided, as long as the assumptions of the ASV reaching the new velocity and heading at once and that the obstacles keep their current velocity and heading are uphold. In 2014, Kuwata, Wolf, Zarhitsky and Huntsberger implemented an autonomous motion planning algorithm for ASVs by adapting VO in order to comply with COLREGs [3]. Their solution was based on dividing the velocity space into four different regions, and by determining which maneuver was expected by the ASV, according to COLREGs, it could calculate which regions should be avoided.

2.6 Set-Based Guidance (SBG)

Set-Based Guidance (SBG) is used to generate references, desired trajectories or desired velocities for a robotic system that, if satisfied, will result in a specified desired behavior. It uses inverse kinematics to ensure that several tasks can be preformed at the same time. This is done by assigning the tasks different prioritizing and projecting task velocities through the null space of higher prioritized tasks. Two different types of tasks can be used in SBG; equality tasks and set-based tasks. Equality tasks are tasks with just one valid value, whereas set-based tasks has a range of valid values. The ability to handle set-based tasks was first presented in 2015 by Moe, Teel, Antonelli and Pettersen [1], [5]. They used a general, fully actuated robotic system with n DOF and states described by its joint values $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T$. Tasks and task velocities can then be defined as

$$\boldsymbol{\sigma}(\boldsymbol{q}) = \boldsymbol{f}(\boldsymbol{q}) \tag{2.1}$$

$$\dot{\boldsymbol{\sigma}}(\boldsymbol{q}) = \boldsymbol{\delta f}(\boldsymbol{q})\boldsymbol{\delta q}\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.2}$$

where $\boldsymbol{\sigma}$ represents the tasks and $\dot{\boldsymbol{\sigma}}(\boldsymbol{q})$ the task velocities. $\boldsymbol{J}$ is the configuration-dependent task Jacobian matrix and $\dot{\boldsymbol{q}}$ is the joint velocities. C is denoted as the set of valid values for a set-based task.

In order to determine which tasks that are active, a tangent cone is defined for a closed set $C = [a, b]$ as [1]

$$\boldsymbol{T}_C(\boldsymbol{\sigma}) = \begin{cases} [0, \infty) & \boldsymbol{\sigma} = a \\ \mathbb{R} & \boldsymbol{\sigma} \in P \\ (-\infty, 0] & \boldsymbol{\sigma} = b \end{cases}, \tag{2.3}$$

where P is the interior of C.

A task $\boldsymbol{\sigma}$, with a valid set $C = [\boldsymbol{\sigma}_{\mathbf{min}}, \boldsymbol{\sigma}_{\mathbf{max}}]$, is active if its task velocity $\dot{\boldsymbol{\sigma}}$ lies within the tangent cone. A function is used in order to determine this, and it is illustrated in Figure 2.2. By placing $\boldsymbol{\sigma}$ according to $\boldsymbol{\sigma}_{\mathbf{min}}$ and $\boldsymbol{\sigma}_{\mathbf{min}}$, and calculating $\dot{\boldsymbol{\sigma}}$, the position on the illustration can be determined. If the position lies in a green area then $\dot{\boldsymbol{\sigma}}$ lies inside the tangent cone and the task $\boldsymbol{\sigma}$ is active. If the area is red then $\dot{\boldsymbol{\sigma}}$ does not lie with in the tangent cone and the task $\boldsymbol{\sigma}$ is not active.

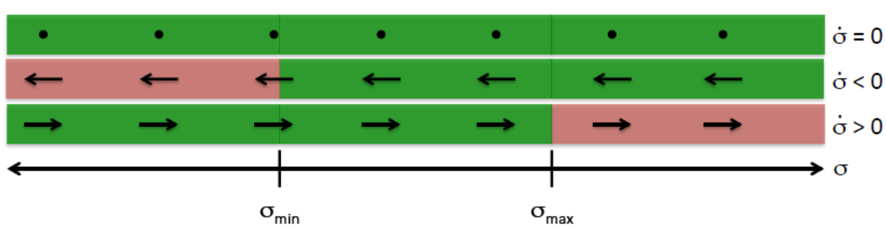

Figure 2.2: Illustration of the function used to check if a task velocity lies within the tangent cone. Figure from [1].

The authors in [1] also suggests that SBG can be used in obstacle avoidance for static obstacles. In the case with the ASV, obstacle avoidance is a task denoted as the distance between the ASV and the obstacle. The set of valid values is then the area free of obstacles, making obstacle avoidance a set-based task that could be solved with the use of SBG.

Chapter 3

Theoretical Background

The theoretical background, which is the foundation of the thesis, is covered in this chapter. The information given is based on a general case with an ASV and other vessels, viewed as obstacles, at sea. Assumptions and adaptations specially made for this thesis is covered in later chapters.

3.1 Reference Frames

Reference frames are needed to model the kinematics and kinetics of a vessel. Two seemingly similar positions could in fact be two completely different positions expressed in each their own reference frame. In a collision avoidance situation, it is essential to know the extract position and velocity of the ASV as well as any obstacles, making the use of reference frames important. In this thesis, NED and BODY reference frames have been used and they will be presented in this section, which is based on Fossen's Handbook of Marine Craft Hydrodynamics and Motion Control [2, p.16-17].

NED

The NED (North East Down) reference frame (x_n, y_n, z_n) is defined by a moving tangent plane on the surface of the earth. The x-axis is pointing North, y-axis pointing East and the z-axis is pointing downwards, normal to the plane. See Figure 3.1.

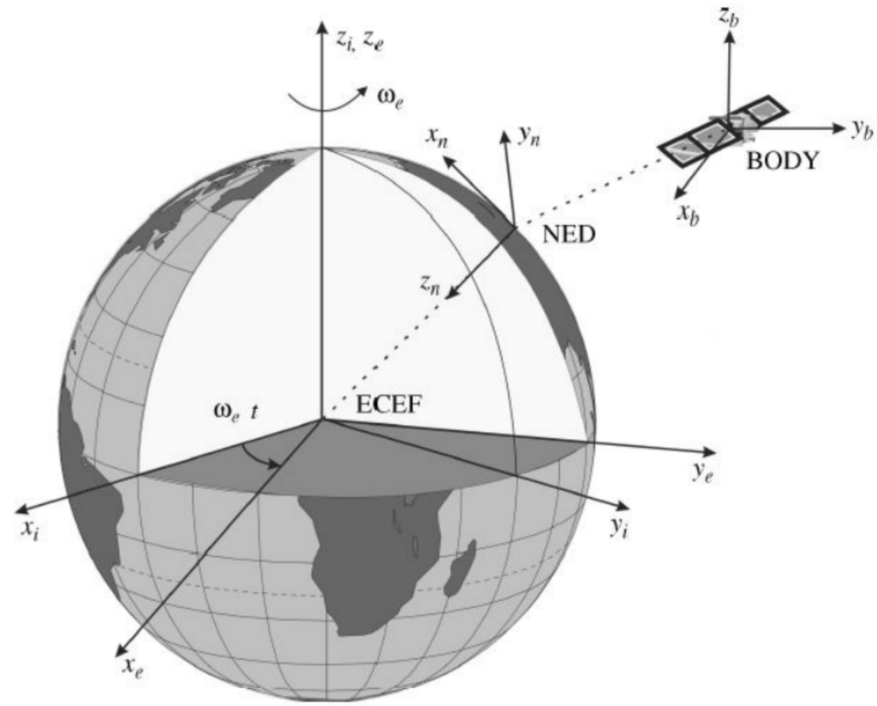

Figure 3.1: Coordinate frames. Figure from [2, p.17]

BODY

BODY (body-fixed) reference frame (x_b, y_b, z_b) is a frame fixed to the body of the object of interest, and therefore move with the body. The origin and the axes are chosen due to the objects natural behavior. Where roll is rotation around the x-axis, pitch is rotation around the y-axis and yaw is rotation around the z-axis. In the case with the ASV, the position and orientation in NED are actually the origin of the BODY frame located on the ASV relative the NED frame.

3.2 Mathematical Model

This section is presenting the mathematical model of a vessel, together with a common simplification of the model. The section is based on Fossen's Handbook of Marine Craft Hydrodynamics and Motion Control [2].

Table 3.1: Motions and rotations for all the six DOF a vessel can operate, with SNAME (1950) notation. Table collected from [2, p.16]

DOF		forces and moments	linear and angular velocities	positions and Euler angles
1	motions in the x-direction (surge)	X	u	N
2	motions in the y-direction (sway)	Y	v	E
3	motions in the z-direction (heave)	Z	w	D
4	rotation about the x-axis (roll, heel)	K	p	ϕ
5	rotation about the y-axis (pitch, trim)	M	q	θ
6	rotation about the z-axis (yaw)	N	r	ψ

6 DOF model

The mathematical model of a vessel can be represented with a complete 6 Degrees of Freedom (DOF) vectorial representation called the marine craft equations of motion:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J_\theta}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{3.1}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g_0} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}}, \tag{3.2}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{P}_{bn}^{n} \\ \boldsymbol{\theta}_{nb} \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{v}_{bn}^{b} \\ \boldsymbol{\omega}_{bn}^{b} \end{bmatrix}, \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{f}_{b}^{b} \\ \boldsymbol{m}_{b}^{b} \end{bmatrix}, \tag{3.3}$$

where $\boldsymbol{\eta}$ denotes the position and orientation vector containing $\boldsymbol{P}_{bn}^{n}$ and $\boldsymbol{\theta}_{nb}$, which are the distance between NED and BODY expressed in NED coordinates, and a vector of Euler angles, respectively. The vector $\boldsymbol{\nu}$ consist of the linear velocity vector $\boldsymbol{v}_{bn}^{b}$ and the angular velocity vector $\boldsymbol{w}_{bn}^{b}$, both are decomposed in the BODY frame. The vector $\boldsymbol{\tau}$ includes the forces $\boldsymbol{f}_{b}^{b}$ and moments $\boldsymbol{m}_{b}^{b}$ acting on the craft in the BODY frame. Furthermore, $\boldsymbol{J}_\theta(\boldsymbol{\eta})$ is transformation matrix, while the matrices $\boldsymbol{M}$, $\boldsymbol{C}(\boldsymbol{\nu})$ and $\boldsymbol{D}(\boldsymbol{\nu})$ is the inertia, Coriolis and damping matrix, respectively. The vector $\boldsymbol{g}(\boldsymbol{\eta})$ contains generalized gravitational and buoyancy forces, while $\boldsymbol{g_0}$ is a collected term for all static restoring forces and moments due to ballast systems and water tanks. In the end, $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{wave}}$ are vectors of environmental forces.

Table 3.1 show all possible motions and rotations for the six DOF a vessel can operate, expressed with the SNAME (1950) notation. The six DOF are surge, sway, heave, roll, pitch and yaw, and they are illustrated in Figure 3.2.

By using the SNAME (1950) notation from Table 3.1 in Equation 3.3, one gets:

$$\boldsymbol{\eta} = [N, E, D, \phi, \theta, \psi]^T, \tag{3.4}$$

$$\boldsymbol{\nu} = [u, v, w, p, q, r]^T, \tag{3.5}$$

$$\boldsymbol{\tau} = [X, Y, Z, K, M, N]^T. \tag{3.6}$$

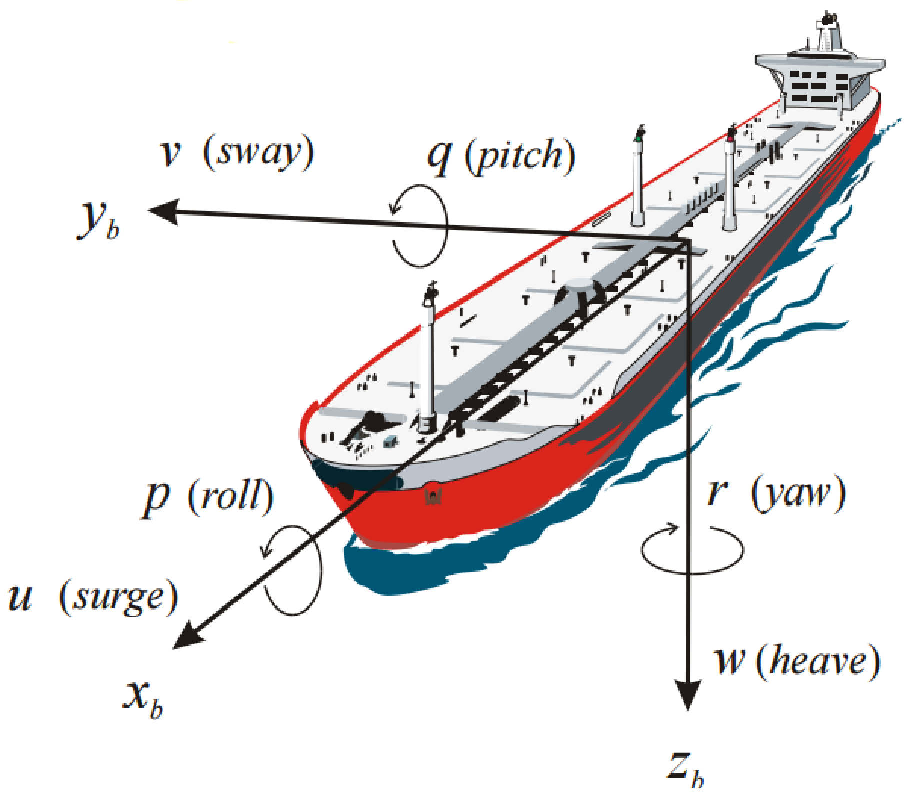

Figure 3.2: Illustation of all the 6 DOF operable by a vessel. Figure collected from [2, p.16].

3 DOF model

A common simplification of the marine craft equations (Eq.(3.1)-(3.2)) are to assume that there is no heave w and that roll ϕ and pitch θ are small, which are reasonable assumptions for most conventional ships. When applying these simplifications, the elements corresponding to heave, roll and pitch are neglected, leaving only surge, sway and yaw. As a natural result, the simplified marine craft equation consist of three DOF rather than six:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\nu}, \tag{3.7}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{\tau_{wind}} + \boldsymbol{\tau_{wave}}, \tag{3.8}$$

where

$$\boldsymbol{\eta} = [N, E, \psi]^T, \tag{3.9}$$

$$\boldsymbol{\nu} = [u, v, r]^T, \tag{3.10}$$

$$\boldsymbol{\tau} = [X, Y, N]^T. \tag{3.11}$$

This simplified version of the marine craft equations are going to be used for the remainder of this thesis. The origin of the BODY reference system is located at the ship's center of gravity.

The transformation matrix $\boldsymbol{J}(\psi)$ from Equation 3.7 is the same as the rotation matrix $\boldsymbol{R}_{z,\psi}$, which gives

$$\boldsymbol{J}(\psi) = \boldsymbol{R}_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.12}$$

Both the system inertia matrix and the Coriolis-centripetal matrix is found by adding the rigid-body matrix and the added mass matrix together.

The system inertia matrix is

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_{A}, \tag{3.13}$$

with

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \ \boldsymbol{M}_{A} = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix}, \tag{3.14}$$

where m is the mass of the ASV, I_z is the moment of inertia about the z-axis and $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$ and $N_{\dot{r}}$ are the constant, added mass terms.

The Coriolis-centripetal matrix is

$$\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}), \tag{3.15}$$

with

$$\begin{gathered}\boldsymbol{C}_{RB} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix}, \\ \boldsymbol{C}_A = \begin{bmatrix} 0 & 0 & -Y_{\dot{v}}v + \frac{Y_{\dot{r}}+N_{\dot{v}}}{2}r \\ 0 & 0 & X_{\dot{u}}u \\ Y_{\dot{v}}v + \frac{Y_{\dot{r}}+N_{\dot{v}}}{2}r & -X_{\dot{u}}u & 0 \end{bmatrix},\end{gathered} \tag{3.16}$$

where u and v are the linear velocities in x- and y-direction and r is the angular velocity about the z-axis.

The total hydrodynamic damping matrix $\boldsymbol{D}(\boldsymbol{\nu})$ can be determined by adding the linear part $\boldsymbol{D}_L$ and the nonlinear part $\boldsymbol{D}_{NL}(\boldsymbol{\nu})$ together:

$$\boldsymbol{D}(\boldsymbol{\nu}) = \boldsymbol{D}_L + \boldsymbol{D}_{NL}(\boldsymbol{\nu}), \tag{3.17}$$

with

$$\begin{gathered}\boldsymbol{D}_L = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}, \\ \boldsymbol{D}_{NL}(\boldsymbol{\nu}) = \\ -\begin{bmatrix} X_{|u|u}|u| + X_{uuu}u^2 & 0 & 0 \\ 0 & Y_{|v|v}|v| + Y_{vvv}v^2 & 0 \\ 0 & 0 & N_{|r|r}|r| + N_{rrr}r^2 \end{bmatrix},\end{gathered} \tag{3.18}$$

where X_u, Y_v, Y_r, N_v and N_r are constant, linear damping terms and $X_{|u|u}$, $Y_{|v|v}$, $N_{|r|r}$, X_{uuu}, Y_{vvv} and N_{rrr} are constant, nonlinear damping terms.

The vessel model consists of two control inputs; the rudder angle and the propeller force, which are presented in the generalized force vector $\boldsymbol{\tau}$. Since the amount of control inputs is lower than the degrees of freedom, the vessel is *underactuated*. The generalized force vector is

$$\boldsymbol{\tau} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ l_r F_y \end{bmatrix}, \tag{3.19}$$

where l_r is the rudder length. F_x is proportional to $n \cos \delta$ and F_y is proportional to $n \sin \delta$, where n is the propeller shaft speed and δ is the rudder deflection angle.

3.3 Operating spaces

Operating spaces are used in order to give an exact description of a space considered in a certain situation. Two different kind of spaces are given in this section.

3.3.1 Workspace

The workspace $\mathcal{W}$ is defined by Li, Cao and Yang as the set of points that can be reached by the end-effector of a robot [20]. For the ASV, this will correspond to the workspace being the set of all possible positions that can be reached, and with no limitations on the movement of the ASV, the workspace will be represented as $\mathcal{W} = \mathbb{R}^2$. Static and dynamic obstacles such as main land or other vessels are denoted as $\mathcal{O} \subset \mathcal{W}$. The workspace is a global space not limited to any specific vessel.

3.3.2 Configuration space

According to Spong, Hutchinson and Vidyasagar [21], a *configuration* is a description of the location of every point of a robot, and a *configuration space* $\mathcal{C}$ is the set of all possible configurations. In the case of an ASV at sea, this will translate to the configuration being the position and orientation of the ASV and the configuration space being all possible positions and orientations. The configuration space can therefore be represented as $\mathcal{C} = \mathbb{R}^2 \times SO(2)$, where $\mathbb{R}^2$ is the two dimensional plane containing the positions and $SO(2)$ is the unit circle containing all possible rotations. The set within the configuration space that is occupied by obstacles is called the *configuration space obstacle* and is expressed as

$$\mathcal{C}_{obs} = \{\boldsymbol{\eta} \in \mathcal{C} | \boldsymbol{\mathcal{A}}(\boldsymbol{\eta}) \cap \mathcal{O} \neq \emptyset\}, \tag{3.20}$$

where $\boldsymbol{\mathcal{A}}(\boldsymbol{\eta})$ is the subset of the workspace $\mathcal{W}$ occupied by the ASV with the configuration $\boldsymbol{\eta}$, and $\mathcal{O}$ is the set of positions in the workspace occupied by obstacles. The set of configurations that are safe from collision is called the *free configuration space* and can be expressed as

$$\mathcal{C}_{free} = \mathcal{C} \backslash \mathcal{C}_{obs}. \tag{3.21}$$

The ASV should always stay within $\mathcal{C}_{free}$ and out of $\mathcal{C}_{obs}$.

3.4 Line of Sight Guidance

Path following is a motion control scenario of following a predefined path with no constraints regarding time [22]. Line of Sight (LOS) guidance can be used to obtain path following.

In LOS guidance the vessel is following a vector called LOS-vector pointing from the vessel towards a point (x_{los}, y_{los}) on the straight line between the previous waypoint WP_k and next one WP_{k+1}, see Figure 3.3. A desired course angle $\chi_d(e)$ is calculated in order to steer the vessel along the LOS-vector:

$$\chi_d(e) = \chi_p + \chi_r(e), \tag{3.22}$$

where

$$\chi_p = \alpha_k = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \tag{3.23}$$

is the *path-tangential angle* and

$$\chi_r(e) = \arctan\left(\frac{-e}{\Delta}\right) \tag{3.24}$$

is the *velocity-path relative angle.*

The variable e is the cross-track error normal to the path:

$$e = -[x - x_k]\sin(\alpha_k) + [y - y_k]\cos(\alpha_k), \tag{3.25}$$

and Δ is the predefined lookahead distance. All variables can be seen in Figure 3.3.

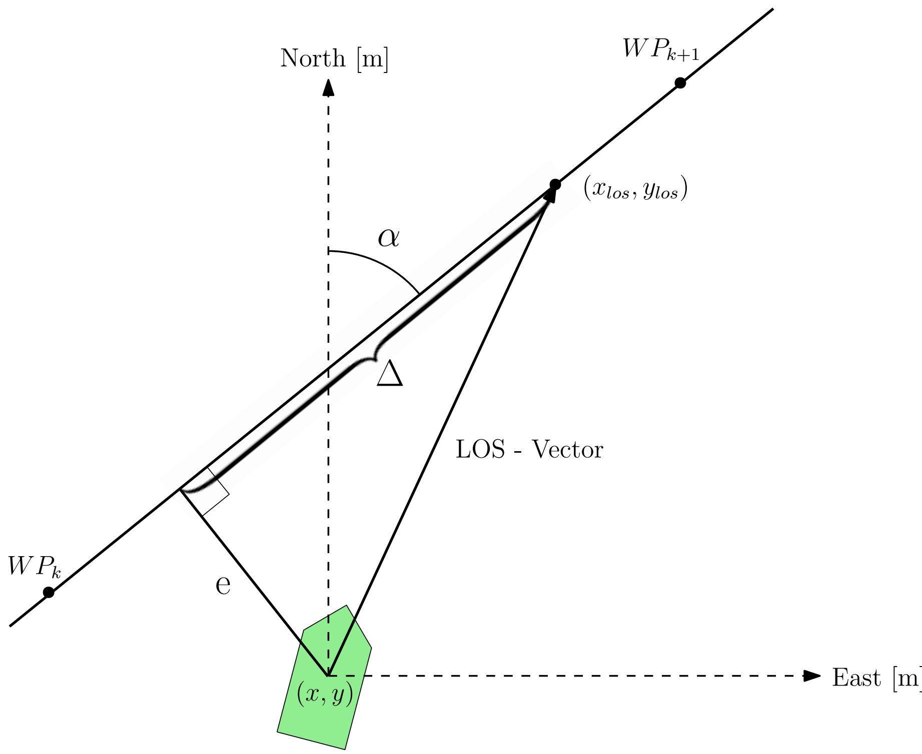

Figure 3.3: Variables in LOS Guidance illustrated.

3.5 COLREGs

COLREGs is the international regulations for preventing collisions at sea formalized by International Maritime Organization (IMO) in 1972 and became effective in 1977 [23]. The COLREGs rules consist of five parts A-E covering different areas. For this thesis part B is the most relevant consisting of steering and sailing rules. Within part B rule number 8 (b) and (d) and number 13-17 are the most important for this thesis in order for the AVS to act satisfyingly in collision avoidance situations. The rules are collected from [24].

3.5.1 COLREGs Rules

Rule 8 - Action to avoid collision

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

Rule 13 - Overtaking

(a) *Notwithstanding anything contained in the Rules of part B, sections I and II, any vessel overtaking any other shall keep out of the way of the vessel being overtaken*
(b) *A vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam, that is, in such a position with reference to the vessel she is overtaking, that at night she would be able to see only the sternlight of that vessel but neither of her sidelights.*
(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.*
(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 situation

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

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 give-way vessel

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

Rule 17 - Action by stand-on vessel

(a)
(i) *Where one of two vessels is to keep out of the way the other shall keep her course and speed.*
(ii) *The latter vessel may however take action to avoid collision by her manoeuvre alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules.*
(b) *When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid collision.*
(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.*
(d) *This Rule does not relieve the give-way vessel of her obligation to keep out of the way.*

3.5.2 COLREGs in collision avoidance

At sea, it is important that one's maneuvers makes sense to the surrounding vessels. Odd behavior can make surrounding vessels uncertain about one's initiations which could lead to propagation of the odd behavior. Therefore, it is essential that all maneuvers related to interactions between vessels are clear and precise. Following the international guideline in COLREGs is very important for this purpose.

The different circumstances where an evasive maneuver could be needed to avoid a collision can be divided into four situations; head on, overtaking, crossing from left and crossing from right [3]. COLREGs rules are based on human judgment of situations and therefore have few specific regulations in place to decide which COLREGs situation applies at a certain time. The rules contain one boundary (Rule 13(b)), which states that a vessel is in an overtaking situation if it comes up with another vessel from a direction more than 22.5 degrees from the beam of the second vessel, see Figure 3.4. Aside form this set boundary, the rest is decided by looking at what has been used successfully by others. According to [25], [26], [27] and [28], choosing head on angle to be in total of 30 degrees wide, centered around the heading of the vessel is a good choice, leaving crossing from right and crossing from left 97.5 degrees on each side, see Figure 3.5. The angle β is the relative bearing between the ASV and an obstacle, it is calculated as:

$$\beta = atan2(y_{\text{asv}} - y_{\text{obs}}, x_{\text{asv}} - x_{\text{obs}}) - \psi_{\text{obs}} \tag{3.26}$$

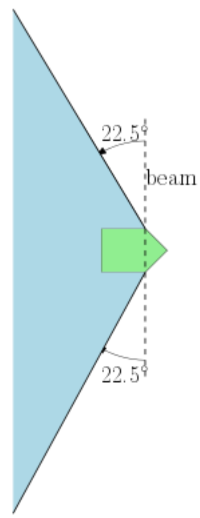

Figure 3.4: The COLREGs rules defines an overtaking as a situation where a vessel comes up with another vessel from a direction more than 22.5 degrees from its beam, illustrated by the blue area.

As a result of the COLREGs rules the behavior in the different situations should be as follows:

Head on
In the situation of head on, both vessels should avoid collision by changing the course to the starboard as seen in Figure 3.6.

Overtaking
In case of an overtaking, the vessel being overtaken should keep steady speed and course. COLREGs rule 13 allows the vessel doing the overtaking to pass the other vessel on both sides, but starboard side tends to be favored [3] as long as there are no apparent advantage by choosing on port side, see Figure 3.7.

Crossing
When crossing from either left(port) or right (starboard) the vessel having the other vessel on its starboard side is the give-way vessel and should change its course so it passes behind the other vessel. The other vessel should keep a steady speed and course, see Figure 3.8.

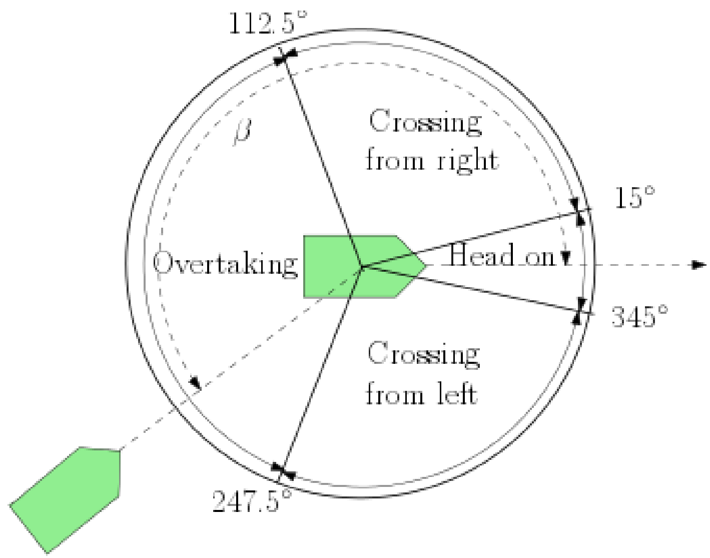

Figure 3.5: The boundaries between the different COLREGs situations. In this case the vessel in the bottom left corner is overtaking the other vessel. The angle β is the relative bearing between the two vessels and decides the current situation.

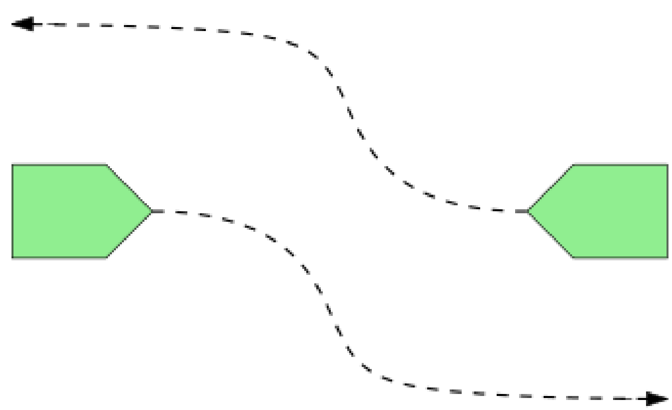

Figure 3.6: The correct way of steering in case of a head on situation, according to COLREGs.

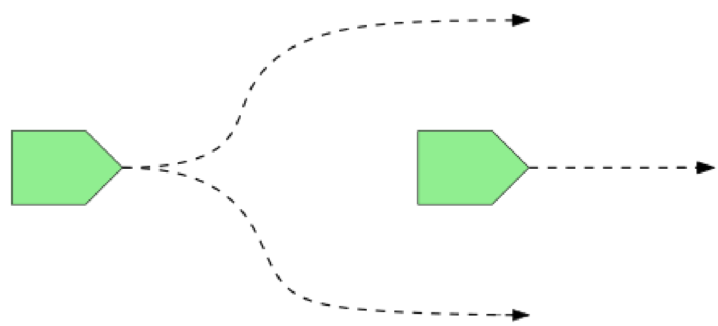

Figure 3.7: The correct ways of steering in case of an overtaking, according to COLREGs.

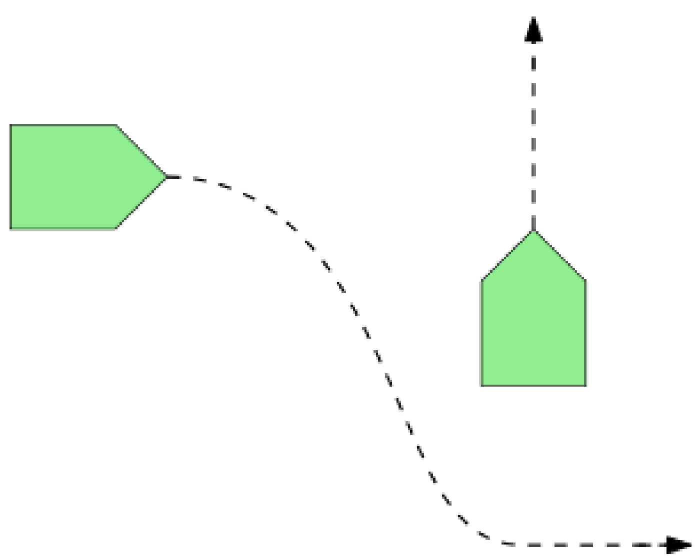

Figure 3.8: The correct way of steering in case of a crossing situation, according to COLREGs.

3.6 Closest Point of Approach

Closest point of approach, also known as CPA, is the point where the ASV is closest to an obstacle at any time. Current velocity and heading for both the ASV and the obstacle are used in order to calculate CPA. The time until CPA and distance between the ASV and the obstacle at CPA are calculated as follows [3]:

$$t_{CPA} = \begin{cases} 0 & \text{if } \|\boldsymbol{v}_A - \boldsymbol{v}_B\| \leqslant \epsilon \\ \frac{(\boldsymbol{p}_A - \boldsymbol{p}_B)\cdot(\boldsymbol{v}_A - \boldsymbol{v}_B)}{\|\boldsymbol{v}_A - \boldsymbol{v}_B\|^2} & \text{otherwise} \end{cases}, \tag{3.27}$$

$$d_{CPA} = \|(\boldsymbol{p}_A + \boldsymbol{v}_A t_{CPA}) - (\boldsymbol{p}_B + \boldsymbol{v}_B t_{CPA})\|, \tag{3.28}$$

where t_{CPA} is the time to closest point of approach and d_{CPA} is the distance between the ASV and an obstacle at that time. An illustration of CPA can be seen in Figure 3.9, where the red and blue dots represents positions at a certain time. It can be seen that CPA occurs right after the third dot. If the ASV is represented by object A and the obstacle by object B, t_{CPA} is the time it takes A to reach CPA, and d_{CPA} is the length of the green bar.

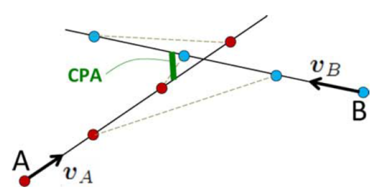

Figure 3.9: Illustration of the closest point of approach (CPA), where the red and blue dots are positions of the ASV and an obstacle at certain time steps. CPA occurs right after the third dot. Figure from [3].

3.7 Velocity Obstacle

Velocity Obstacle, also known as VO, for a vessel is the set of velocities for which the vessel is going to collide with an obstacle sometime in the future as long as both vessels keeps a constant velocity. The principle of the VO collision avoidance method is to determined the velocities that should be avoided in order to prevent any collisions in the future. The equations used in this section are based on [3].

The VO for vessel A with shape $\mathcal{A}$ with respect to an obstacle B with shape $\mathcal{B}$ is expressed like

$$VO_{A|B} = \{\boldsymbol{v}_A \,|\, \lambda(\boldsymbol{p}_A, \boldsymbol{v}_A - \boldsymbol{v}_B) \cap (\mathcal{B} \oplus -\mathcal{A}) \neq \emptyset\}\,, \tag{3.29}$$

where $\boldsymbol{p} \in \mathbb{R}^2$ denotes a position vector, $\boldsymbol{v} \in \mathbb{R}^2$ a velocity vector of a vessel and

$$\lambda(\boldsymbol{p}, \boldsymbol{v}) = \{\boldsymbol{p} + t\boldsymbol{v} \,|\, t \geqslant 0\} \tag{3.30}$$

is the expression of a ray starting in $\boldsymbol{p}$ and going in $\boldsymbol{v}$-direction.

The following set operations has been used in Equation 3.29:
Minkowski sum:

$$\mathcal{A} \oplus \mathcal{B} = \{\boldsymbol{a} + \boldsymbol{b} \,|\, \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B}\} \tag{3.31}$$

Reflection:

$$-\mathcal{A} = \{-\boldsymbol{a} \,|\, \boldsymbol{a} \in \mathcal{A}\} \tag{3.32}$$

By assuming that that both vessel A and B are disc shaped, Equation 3.29 can be simplified as following:

$$VO_{A|B} = \{\boldsymbol{v}_A \,|\, \lambda(\boldsymbol{p}_A, \boldsymbol{v}_{BA}) \in D(\boldsymbol{p}_{AB}, r_{AB})\}\,, \tag{3.33}$$

where $\boldsymbol{v}_{BA} = \boldsymbol{v}_A - \boldsymbol{v}_B$ and $D(\boldsymbol{p}_{AB}, r_{AB})$ is a disc with center in $\boldsymbol{p}_{AB} = \boldsymbol{p}_B - \boldsymbol{p}_A$ and radius $r_{AB} = r_A + r_B$. The vectors $\boldsymbol{p}_{AB}$ and $\boldsymbol{v}_{BA}$ are the relative position and velocity of vessel B with respect to vessel A, while r_A and r_B are the radii of the disc representation of vessel A and vessel B respectively.

The VO for a vessel with an obstacle present is going to be represented with a cone shaped ”danger zone” in the velocity space, also known as the VO cone. As long as the vessel maintains a velocity outside the VO cone it is not going to collide with the obstacle, assuming the velocity vectors are constant over time. CPA is used in order to rule out obstacles that is fare away or that is going to pass a safe distance away from the ASV. An obstacle is deemed as a real treat if the following criteria are fulfilled [3]:

$$0 \leqslant t_{CPA} \leqslant t_{max} \text{ and } d_{CPA} \leqslant d_{min} \tag{3.34}$$

Which means that an obstacle needs to be within a certain distance of the ASV within a certain time limit to be considered as an obstacle that possibly needs to be avoided. All obstacles which do not fulfill these requirements are not taken into account when desired velocity and heading are chosen. This also rules out obstacles that has already passed the ASV, since these would get a negative t_{CPA}.

Because of the VO being cone shaped, Equation 3.33 can be simplified further, making the task of determining whether a velocity vector lies within the VO cone a lot easier:

$$VO_{A|B} = \left\{ \boldsymbol{v}_A \,|\, \boldsymbol{v}_{BA} \cdot \boldsymbol{p}^{\perp}_{AB,left} \geqslant 0 \wedge \boldsymbol{v}_{AB} \cdot \boldsymbol{p}^{\perp}_{AB,right} \geqslant 0 \right\}, \tag{3.35}$$

where $\boldsymbol{p}^{\perp}_{AB,left}$ and $\boldsymbol{p}^{\perp}_{AB,right}$ are vectors perpendicular to the boundaries of the VO cone pointing inwards [29]. They can be expressed as:

$$\boldsymbol{p}^{\perp}_{AB,left} = \boldsymbol{R}(-\alpha + \frac{\pi}{2})\frac{\boldsymbol{p}_{AB}}{\|\boldsymbol{p}_{AB}\|}, \tag{3.36}$$

$$\boldsymbol{p}^{\perp}_{AB,right} = \boldsymbol{R}(\alpha - \frac{\pi}{2})\frac{\boldsymbol{p}_{AB}}{\|\boldsymbol{p}_{AB}\|}, \tag{3.37}$$

where

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.38}$$

is the rotation matrix and

$$\alpha = \arcsin(\frac{r_A + r_B}{\|\boldsymbol{p}_{AB}\|}) \tag{3.39}$$

is the angle between the cone boundary and $\boldsymbol{p}_{AB}$, which can be seen in Figure 3.10. Obstacles close to the ASV will produce a wide cone while obstacle further away will result in a more narrow cone.

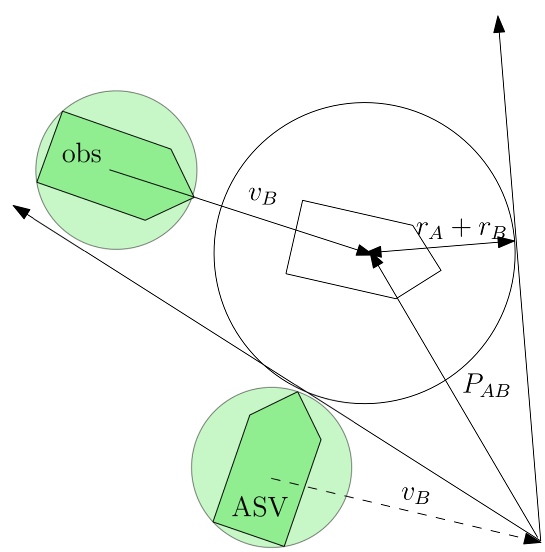

Figure 3.10: Illustration of the different parameters that makes the Velocity Obstacle cone.

According to [3], the velocity space can be divided into four different regions: $VO_{A|B}$, $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$, where

$$\mathcal{V}_1 = \left\{ \boldsymbol{v}_A \,|\, \boldsymbol{v}_A \notin VO_B^A \cup \mathcal{V}_3 \wedge [\boldsymbol{p}_{AB} \times v_{BA}]_z < 0 \right\}, \tag{3.40}$$

$$\mathcal{V}_2 = \left\{ \boldsymbol{v}_A \,|\, \boldsymbol{v}_A \notin \mathcal{V}_1 \cup \mathcal{V}_3 \cup VO_B^A \right\} \tag{3.41}$$

and

$$\mathcal{V}_3 = \{ \boldsymbol{v}_A \,|\, \boldsymbol{p}_{AB} \cdot \boldsymbol{v}_{BA} < 0 \}. \tag{3.42}$$

The regions are illustrated in Figure 3.11. If a velocity within $VO_{A|B}$ is chosen, the ASV is going to collide with the obstacle sometime in the future. If a velocity in $\mathcal{V}_1$, $\mathcal{V}_2$ or $\mathcal{V}_3$ is chosen the ASV will pass with the obstacle on the starboard side, on port side or it will move away from the obstacle, receptively.

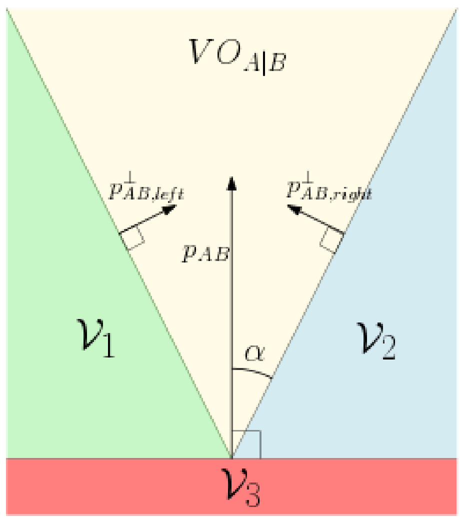

Figure 3.11: Illustration of the four regions in VO.

3.8 Set-Based Guidance

As mentioned in the literature review in Section 2.6, obstacle avoidance can be considered to be a set-based task. With a fully actuated system, the collision avoidance method would have two different tasks; path following and obstacle avoidance. The two tasks would provide two references to the system; one that only takes path following into account and one that has obstacle avoidance as first priority, but also represents path following by projecting its task velocity through the null space of obstacle avoidance.

SBG is developed with a fully actuated system in mind, and as mentioned in Section 3.2, this is typically not the case for ASVs. With the underactuated system used in this thesis, the ASV can not fulfill both tasks at once, instead, the collision avoidance method consists of two separate guidance laws; one for path following ψ_{pf}, and one for obstacle avoidance ψ_{oa}. Obstacle avoidance is the task with the highest priority and the system should always switch from path following mode to obstacle avoidance mode if the ASV get too close to an obstacle. The task of obstacle avoidance is defined as the distance between the ASV and the obstacle, denoted as $\boldsymbol{\sigma}$ and can be calculated as

$$\boldsymbol{\sigma} = \sqrt{(\boldsymbol{p(t)} - \boldsymbol{p_o(t)})^{\boldsymbol{T}}(\boldsymbol{p(t)} - \boldsymbol{p_o(t)})}, \tag{3.43}$$

where $\boldsymbol{p(t)}$ and $\boldsymbol{p_o(t)}$ are the positions of the ASV and obstacle, respectively.

A valid interval where a collision is avoided and therefor the task is satisfied can be defined and is typically $C = [\sigma_{min}, \infty)$, where $\sigma_{min} > 0$ is the smallest distance allowed between the ASV and the obstacle without needing to do any evasive maneuvers.

The system should only switch to obstacle avoidance mode if σ is less than a certain safety radius R_s and it should stay there until switching back to path following mode can be done without σ getting smaller. The rate of change of σ, denoted as $\dot{\sigma}$, is calculated by sending desired path following velocity and heading into a LOS controller (from section 3.4) and thereby comparing position before and after the hypothetical switch back to path following. If $\dot{\sigma} < 0$ it means that the AVS and the obstacle are going to move closer together if the mode is switched back to path following, while $\dot{\sigma} \geqslant 0$ means that the distance is going to increase, or at least stay the same as before. Naturally, switching from obstacle avoidance to path following is only done when $\dot{\sigma} \geqslant 0$.

Chapter 4

Collision Avoidance Method Based on Set-Based Guidance

SBG is presented in the litterateur review in Section 2.6 and in the background theory in Section 3.8 as a method that can be used in obstacle avoidance with static obstacles. This chapter focuses on the main contribution of this thesis: how SBG has been used in order to develop a collision avoidance method that can handle both static and dynamic obstacles.

SBG with static obstacles has a simple and elegant switch mechanism between path following and obstacle avoidance which can be seen in Algorithm 1.

Algorithm 1 Set-Based Guidance with Static Obstacles - Mode Switch

```
procedure MODE SWITCH
    if σ > R_s then
        mode = path following
        ψ_des = ψ_pf
    else
        if σ̇ ⩾ 0 then
            mode = path following
            ψ_des = ψ_pf
        else
            mode = obstacle avoidance
            ψ_des = ψ_oa
        end if
    end if
end procedure
```

As mentioned earlier, σ is the distance between the ASV and the obstacle, and R_s is the safety radius surrounding the obstacle. $\dot{\sigma}$ is a measurement of the change in σ if the mode was path following. The purpose of $\dot{\sigma}$ is to see how the distance between the ASV and the obstacle would change if the mode was switched back to path following or stay in path following if that is already the current mode. If the distance increases or stay the same, giving $\dot{\sigma} \geqslant 0$, it means that it is safe to either switch to path following from obstacle avoidance, or stay in path following. On the other hand, if the distance where to decrease, giving $\dot{\sigma} < 0$, the mode should be obstacle avoidance.

The distance σ defined in Equation 3.43 can be written as

$$\sigma = \sqrt{(\boldsymbol{p(t)} - \boldsymbol{p_o(t)})^{\boldsymbol{T}}(\boldsymbol{p(t)} - \boldsymbol{p_o(t)})} = \sqrt{(x - x_o)^2 + (y - y_o)^2}, \tag{4.1}$$

where

$$\boldsymbol{p(t)} = \begin{bmatrix} x \\ y \end{bmatrix} \text{ and } \boldsymbol{p_o(t)} = \begin{bmatrix} x_o \\ y_o \end{bmatrix} \tag{4.2}$$

are the positions of the ASV and the obstacle, respectively. This gives

$$\dot{\sigma} = \frac{(x - x_o)(\dot{x} - \dot{x_o}) + (y - y_o)(\dot{y} - \dot{y_o})}{\sigma}, \tag{4.3}$$

with

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \boldsymbol{R}(\psi_d) \begin{bmatrix} u \\ v \end{bmatrix} \text{ and } \begin{bmatrix} \dot{x_o} \\ \dot{y_o} \end{bmatrix} = \boldsymbol{R}(\psi_o) \begin{bmatrix} u_o \\ v_o \end{bmatrix}, \tag{4.4}$$

where

$$\boldsymbol{R}(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \tag{4.5}$$

is the rotation matrix, and u, v, u_o and v_o are the velocities for the ASV and the obstacle in x- and y-direction. ψ_d and ψ_o are the desired heading for the ASV and the current heading to the obstacle. The desired heading ψ_d is found with the use of LOS guidance (see Section 3.4).

This way of mode switching works fine with static obstacles and with the assumption of no delay when changing heading. The ASV could switch from path following to obstacle avoidance when it first touches the safety radius R_s, and pass the obstacle on either side by traveling along, but not inside, the border of the radius until $\dot{\sigma} \geqslant 0$ and it is safe to switch back to path following.

When dynamic obstacles are introduced the previous mode switcher contains several issues. Dynamic, or moving, obstacles would in most cases mean other vessels and ones behavior around other vessels should be in line with COLREGs. This entails, amongst other things, that a succession of small alterations of heading should be avoided according to Rule 8(b). Another consequence of COLREGs is that, according to Rule 13-15, specific behavior is expected at the different situations, something that is not in line with the previous mode switcher, where the ASV could pass on either side of the obstacle. In the end, the assumption of no delay in heading alterations is not viable, and turning rate should be taken into account.

Set-Based Guidance with Dynamic Obstacles

The initial switch between path following and obstacle avoidance is triggered by the ASV reaching the safety radius, but the ASV should never pass on the inside of the radius. Also, with the ASV needing time to change heading during evasive maneuvers, this means that two safety radii are needed; one containing the forbidden area around the obstacle and one to trigger obstacle avoidance. The two radii are called $R_{s,inner}$ and $R_{s,outer}$ and can be seen in Figure 4.1. The inner safety radius was chosen by common sense, while the outer radius was found by testing different scenarios and checking that the ASV had enough time to do evasive maneuvers. In the simulations the inner radius $R_{s,inner}$ was chosen to be $20\,m$ while the outer radius $R_{s,outer}$ to be $70\,m$.

When the ASV enters $R_{s,outer}$ and triggers the switches between path following mode and obstacle avoidance mode a three step procedure is run:

Step 1 - Determine COLREGs situation
The COLREGs situation between the ASV and the obstacle is determined by calculating the relative bearing β with the use of Equation 3.26 from Section 3.5.2. The angle β can be seen in Figure 4.1. The COLREGs situation can be either head on, overtaking, crossing from right or crossing from left.

Step 2 - Decide which side to pass on
Which side of the obstacle the ASV should pass on is decided by the COLREGs situations. If the situation is crossing from left, the obstacle is the give-way vessel and the ASV should just keep its course. A suggestion on a safety feature in case the obstacle does not complying with COLREGs can be seen in Chapter 9. If the situation is either head on or crossing from right, the starboard side of the obstacle is always chosen. In the case of overtaking, the angle between the position of the ASV and the next waypoint, called θ, is calculated. If the angle is within some threshold, the ASV passes on the starboard side, and if not, it passes on the port side. In the simulations, this threshold value is $\frac{5\pi}{4}$.

Step 3 - Find a new waypoint
Tangent points on the the $R_{s,inner}$ circle from the position of the ASV and the next waypoint are calculated and can be seen illustrated in Figure 4.2 as Q_{A1}, Q_{A2}, Q_{W1} and Q_{W2}.

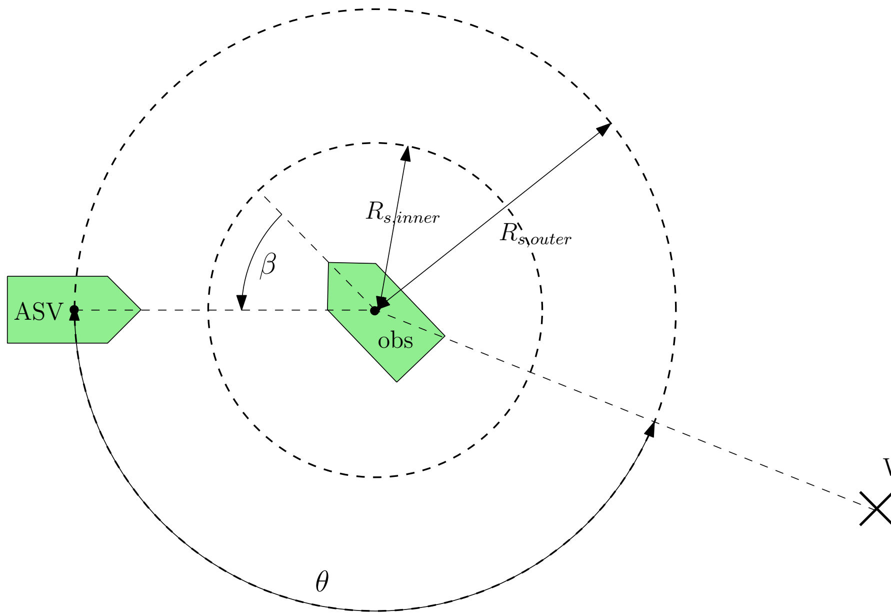

Figure 4.1: Illustration of the two different radii surrounding the obstacle and the angles calculated in order to determined the COLREGs situation and whether the ASV should pass on starboard or port side in case of an overtaking.

A tangent point on a circle with radius r from an external point P1 can be found by first drawing a line between the two tangent point Q1 and Q2 on the circle and then fining the vector $\boldsymbol{P}$ from the center P2 of the circle to the external point P1, see Figure 4.3. The point Q0 is the intersection between $\boldsymbol{P}$ and the line between Q1 and Q2.

By drawing lines between the point Q0, Q1, P1 and P2, three similar and right triangles can be seen. The drawn lines are shown in Figure 4.4 and the triangles can be seen in Figure 4.5.

By using the Pythagorean theorem as well as the fact that the ratio between sides in similar triangles are the same, the sides in all the triangles can be found, see Figure 4.6

The point Q0 can be found by starting with P2 and adding the unit vector of P multiplied with the length between P2 and Q0, giving

$$Q0 = P2 + \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|}\frac{r^2}{\|\boldsymbol{P}\|}. \tag{4.6}$$

Finally, the tangent points Q1 and Q2 can be calculated by starting with Q0 and adding the unit vector of $\boldsymbol{P}$ multiplied with the distance between Q0 and

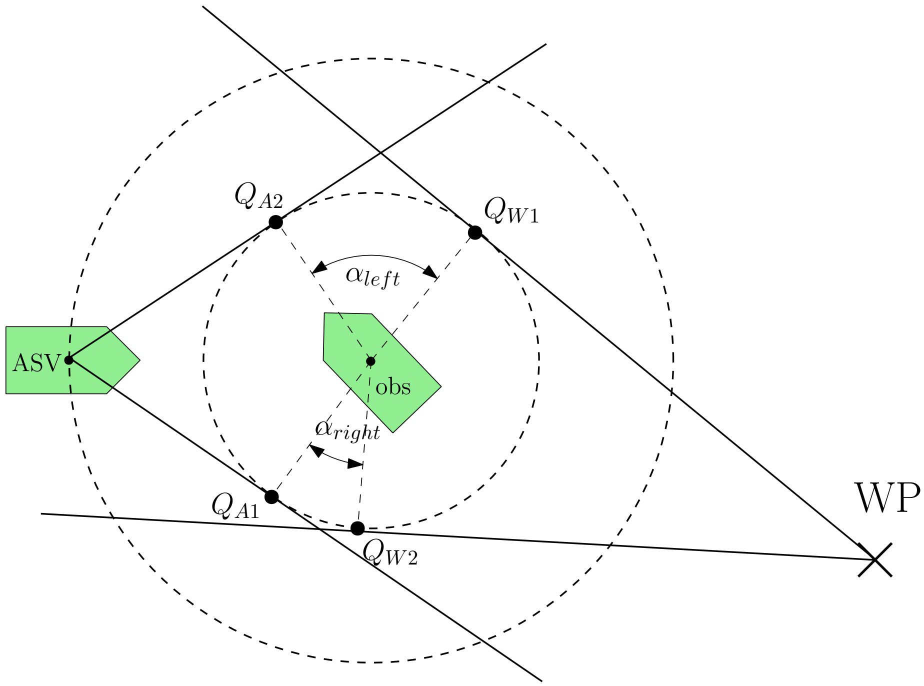

Figure 4.2: Tangent lines drawn in order to find a new waypoint.

Q1 and the rotation matrix with an angle of $\frac{\pi}{2}$ radians for Q1 and $\frac{5\pi}{4}$ radians for Q2, giving

$$Q1 = Q0 + \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|}\frac{r}{\|\boldsymbol{P}\|}\sqrt{\|\boldsymbol{P}\|^2 - r^2}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{4.7}$$

and

$$Q2 = Q0 + \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|}\frac{r}{\|\boldsymbol{P}\|}\sqrt{\|\boldsymbol{P}\|^2 - r^2}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{4.8}$$

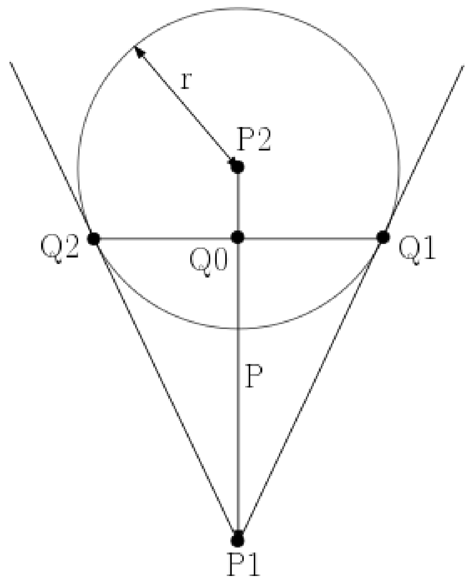

Figure 4.3: Finding tangent points - Tangents on a circle from an external point.

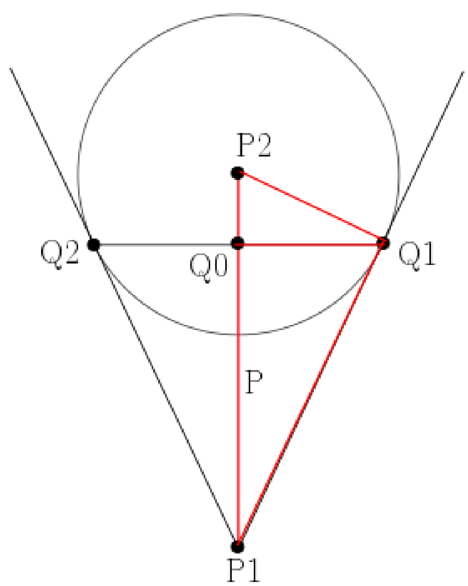

Figure 4.4: Finding tangent points - Lines drawn between Q0, Q1, P1 and P2.

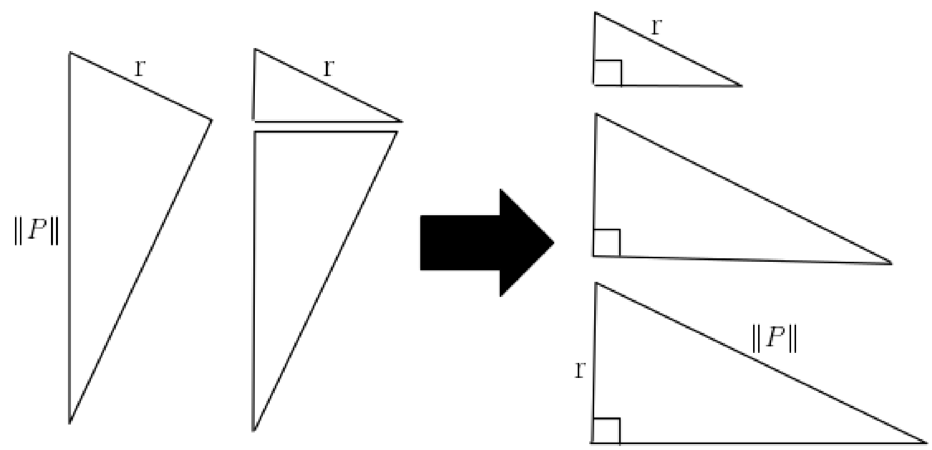

Figure 4.5: Finding tangent points - Three similar, right triangles found by the drawn lines.

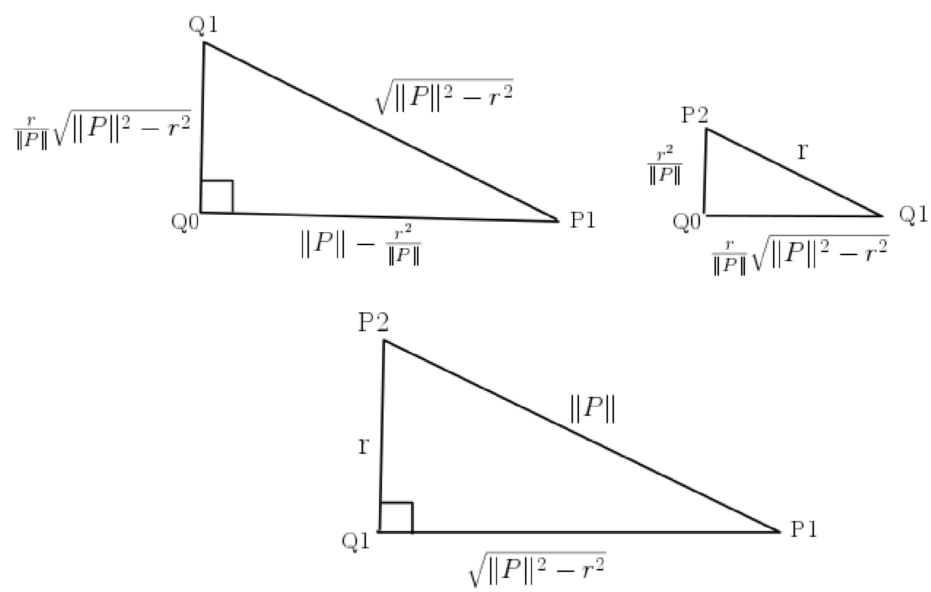

Figure 4.6: Finding tangent points - All sides of the triangles calculated in order to find the tangent points.

Afterwards, the intersection points between the tangent lines are found by using a MATLAB function called polyxpoly [30]. The function takes two line segments as input and returns the intersection point. One of the intersection points are the new waypoint, which one are determined by the side to pass the obstacle on decided in the previous step. The angles between the tangent point pairs,

α_{left} and α_{right}, are calculated and can be seen in Figure 4.2. If the angle corresponding to the side of which the ASV will pass is greater than $\frac{3}{4}\pi$, two new way point is added instead of just one. In case of a head on situation the position of the obstacle used to find the new way point is moved to CPA (see Section 3.7) in order to make up for the obstacle's movement during the passing. And if the situation is overtaking, both the original position of the obstacle and CPA is used, making sure that the ASV keeps its distance both in the front and back of the obstacle.

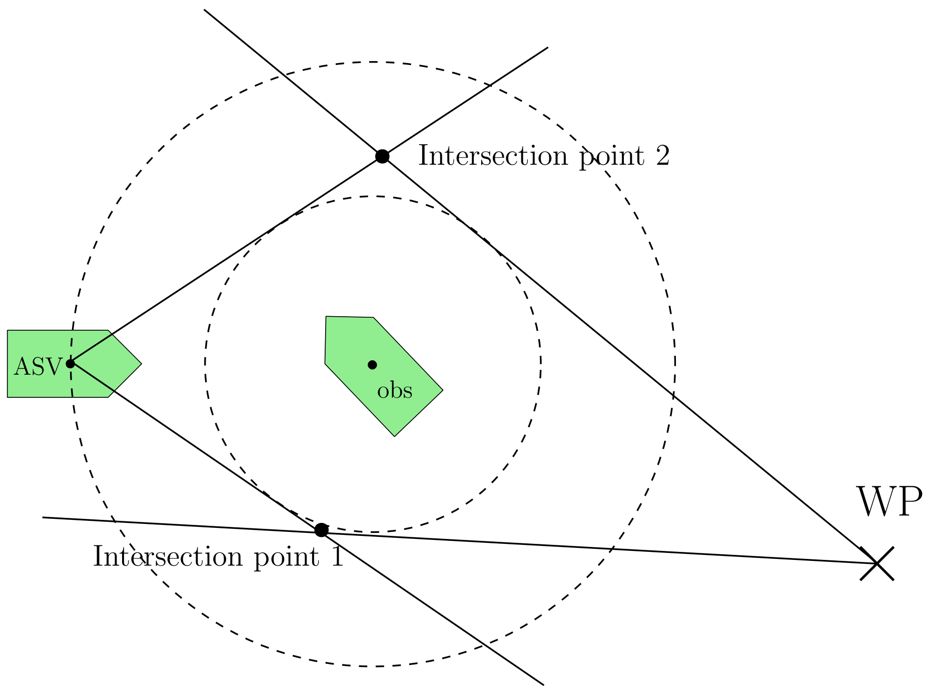

Figure 4.7: Intersection points located.

When the ASV has entered $R_{s,outer}$, the mode is switch to obstacle avoidance and the new waypoints are found and added to a list separate from the normal waypoint list. As long as there are waypoints left in the new list, the next waypoint is chosen from that list. If the mode is switched back to path following before the waypoints on the new list has been reached, the list is emptied and the next waypoint is yet again chosen from the original list.

The algorithm used in SBG with dynamic obstacles can be seen in Algorithm 2. The main differences from the one used with static obstacles is the use of a separate waypoint list instead of returning a specific heading and the checking of $\dot{\sigma} \geqslant 0$ in the beginning of the algorithm. The use of the separate waypoint list is actually just another way of assigning heading as either ψ_{pf} or ψ_{oa}. If wp_list_oa is empty, the mode is path following and ψ_d is chosen with the use of an original waypoint, which in this case is the same as ψ_{pf}. If the wp_list_oa contains one or more waypoints, the mode is obstacle avoidance and the waypoint used to calculate ψ_d in LOS guidance is collected from wp_list_oa, which is the same as assigning a ψ_{oa} to ψ_d. The reason behind the initial check of $\dot{\sigma}$ is that if $\dot{\sigma} \geqslant 0$ then being in path following mode is always safe regardless of ones position.

Algorithm 2 Set-Based Guidance with Dynamic Obstacles - Mode Switch

```
procedure MODE SWITCH
    if σ̇ ⩾ 0 then
        if previous mode == obstacle avoidance then
            mode = path following
            wp_list_oa = []
        else if previous mode == path following then
            mode = path following
        end if
    else if σ ⩽ R_{s,outer} then
        if previous mode == path following then
            mode = obstacle avoidance
            find new wp_oa
        else if previous mode == obstacle avoidance then
            mode = obstacle avoidance
        end if
    else
        if previous mode == obstacle avoidance then
            mode = path following
            wp_list_oa = []
        else if previous mode == path following then
            mode = path following
        end if
    end if
    previous mode = mode
end procedure
```

Chapter 5

Implementation

The chapter will cover the implementation of the different parts of the system as well as list the assumptions made in this thesis.

5.1 Assumptions

- Other vessels holds a constant speed and heading.
- The ASV collision avoidance controller know about every obstacles and has information about their velocity and heading available.
- No environmental forces, resulting in a negligible sideslip.
- At open sea, no mainland close by.

5.2 The system

In Figure 5.1 the block diagram of the system can be seen. In the sections below, the implementation of each of the different parts of the system are explained. The collision avoidance block is produced as two completely separate and different versions, one for each of the collision avoidance methods implemented in the thesis.

Each of the blocks in Figure 5.1 has been implemented separately in MATLAB [31] and connected in Simulink [32] with the use of MATLAB Function blocks.

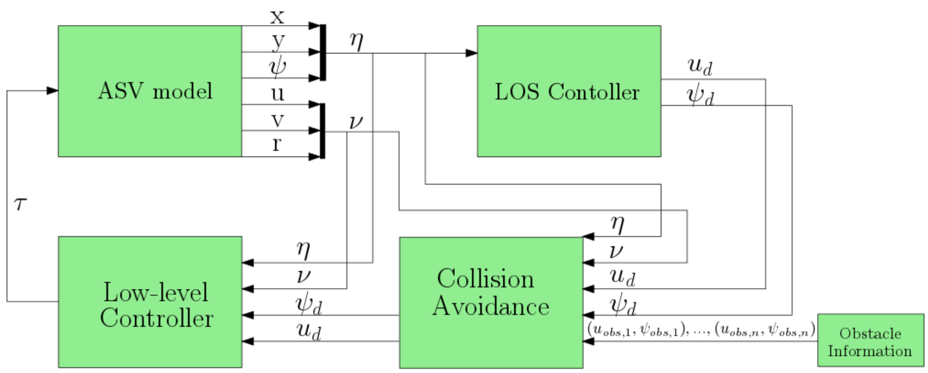

Figure 5.1: Block diagram of the system.

5.2.1 ASV model

The equations needed in this part are explained in details in the theoretical background in Section 3.2. The simplified vessel model with 3 DOF given by Equation 3.7 - 3.19 is used together with the assumption of no environmental forces. The constant terms used in the model can be seen in Table 5.1 and have been used in previous master thesis [28], [27].

In the implementation, both the ASV and the obstacles are the same kind of small vessel with properties as listed in Table 5.2. The properties are based on Viknes 830 and have, like the model parameters, also been used in other master thesis before mine [28], [27].

5.2.2 LOS Controller

The Line of Sight (LOS) controller is based on the equations and explanations in Section 3.4. Since we assume no environmental forces, the sideslip β can be assumed small enough to be disregarded. As can be seen in Figure 5.2, the sideslip β is the difference between course χ and heading ψ, and since it is assumed no sideslip, course and heading become the same thing, and Equation 3.22 can be changed to the following equation:

$$\psi_d(e) = \chi_p + \chi_r(e) \tag{5.1}$$

The predefined lookahead distance Δ is set to $40m$ in the simulations.

Table 5.1: Vessel Model Parameters

Parameter	**Value**
Added mass terms	
$X_{\dot{u}}$	0
$Y_{\dot{v}}$	0
$Y_{\dot{r}}$	0
$N_{\dot{v}}$	0
$N_{\dot{r}}$	0
Linear damping terms	
X_u	-50
Y_v	-200
Y_r	0
N_v	0
N_r	-3224
Nonlinear damping terms	
$X_{\|u\|u}$	-135
$Y_{\|v\|v}$	-2000
$N_{\|r\|r}$	0
X_{uuu}	0
Y_{vvv}	0
N_{rrr}	-3224

Table 5.2: Vessel Properties

Parameter	**Value**	**Unit**
Length	8.5	m
Width	3	m
Mass	3980	kg
Inertia	19703	kg/m^2

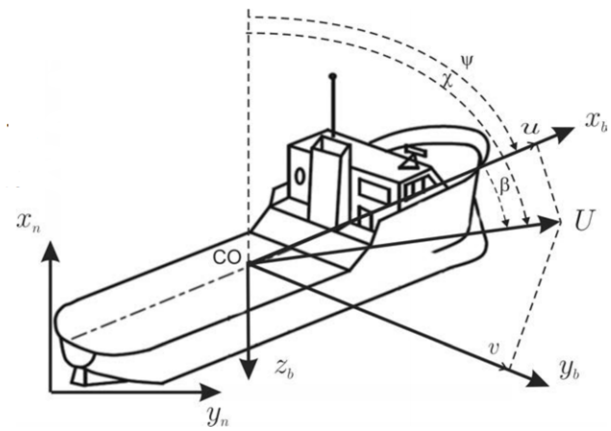

Figure 5.2: Sideslip β is equal to the difference between course χ and heading ψ. Figure from [2, p.40]

Table 5.3: Control Parameters

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

5.2.3 Low-level Controller

The low-level controller turns desired velocity and heading (u_d, ψ_d) into the control force submitted to the vessel model. It consists of two controllers; one for velocity and one for heading:

Velocity controller:

$$F_x = -X_u u - X_{|u|u}|u|u - X_{uuu}u^3 - mrv + K_{p,u}m(u_d - u) \qquad (5.2)$$

Heading controller:

$$l_r F_y = K_{p,\psi} I_z(\Gamma(\psi_d - \psi) + T_{d,\psi}(r_d - r)) \qquad (5.3)$$

Where $K_{p,u}$ and $K_{p,\psi}$ are proportional gains and $T_{d,\psi}$ is derivative time. The values chosen as parameters in the low-level controller can be seen in both Table 5.1 and Table 5.3. The values have, as with the previous parameters, also been used in other master thesis [28], [27].

5.2.4 Collision Avoidance - Velocity Obstacles

In the collision avoidance controller with VO, N_u velocities between u_{min} and u_{max} and N_{psi} headings between $-\psi_{max} + \psi$ and $\psi_{max} + \psi$ are evaluated, resulting in $N_u \cdot N_{psi}$ velocity pairs (u_i, ψ_j) as candidates to be chosen as desired velocity and heading (u_d, ψ_d). The velocity pairs are fixed in BODY frame of the ASV and can be illustrated as a grid surrounding the ASV, see Figure 5.3. The different samples of heading are distributed around the ASV, like rays emerging from the ASV, and the velocity samples are distributed in circles of different size centered around the ASV.

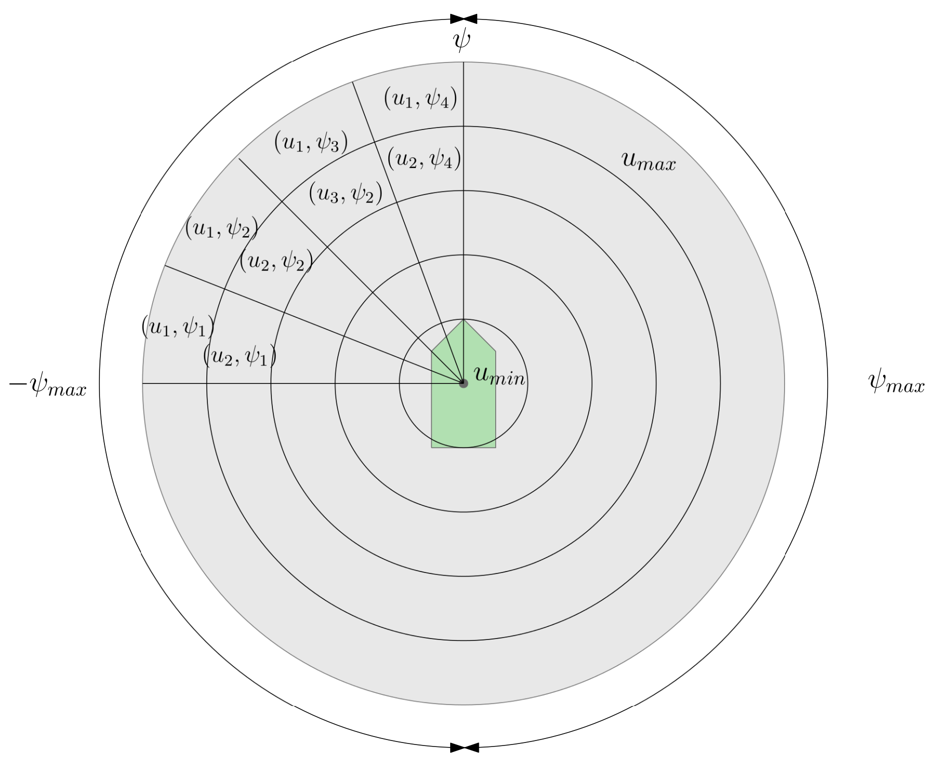

Figure 5.3: Illustration of the grid surrounding the ASV and which constitutes as the velocity pairs. The different headings are distributed around the ASV, while the different velocities are distributed from the inner circle and outwards.

The error velocity vector $\tilde{\boldsymbol{v}}_{i,j}$, COLREGs function $f_{COLREGs}(u_i, \psi_j)$, and collision function $f_{collision}(u_i, \psi_j)$ specified below are calculated and used to find the cost function C for all velocity and heading pairs, where the pair resulting in the lowest cost function value are chosen:

$$C = \alpha \tilde{\boldsymbol{v}}_{i,j}^T \boldsymbol{Q} \tilde{\boldsymbol{v}}_{i,j} + \beta f_{\text{COLREGs}}(u_i, \psi_j) + \gamma f_{\text{collision}}(u_i, \psi_j), \tag{5.4}$$

where α, β and γ are constants. The error velocity is

$$\tilde{\boldsymbol{v}}_{i,j} = \begin{bmatrix} u_d \cos(\psi_d) - u_i \cos(\psi_j) \\ u_d \sin(\psi_d) - u_i \sin(\psi_j) \end{bmatrix}, \tag{5.5}$$

and $f_{\text{COLREGs}}(u_i, \psi_j)$ and $f_{\text{collision}}(u_i, \psi_j)$ are boolean function, meaning functions who returns either 0 or 1. $f_{\text{COLREGs}}(u_i, \psi_j)$ returns 1 if the velocity pair being checked would lead the vessel to a situation where COLREGs would be violated, for example crossing in front of a vessel coming from starboard. To determined whether COLREGs are being violated, the region where the velocity pair is located are calculated using Equation 3.35 and Equation 3.40-3.40, and then compared with the known COLREGs situation and the regions that need to be avoided according to COLREGs. The function $f_{\text{collision}}(u_i, \psi_j)$ returns 1 if the velocity pair being checked are located inside the VO cone, determined by Equation 3.35, in other words: if chosen, the pair of velocity and heading are going to steer the vessel on a collision course.

Implementation of COLREGs

The use of COLREGs in VO has already been done before [3], and the implementation in this thesis is therefore heavily influenced by their approach.

The velocity space surrounding an ASV with a known obstacle can be divided into four regions as mentioned in the background theory in Section 3.7 and which can also be seen in Figure 3.11. The relation between the ASV and the obstacle can be either head on, overtaking, crossing from right or crossing from left as explained in Section 3.5.2. In every iteration the location of all velocity pairs are checked together with the relation between the ASV and the obstacle. If the pair lies in $VO_{A|B}$ both $f_{COLREGs}(u_i, \psi_j)$ and $f_{collision}(u_i, \psi_j)$ are returned true. According to COLREGs the ASV should pass on the starboard side of an obstacle if the relation is either head on or crossing from right. As a simplification, passing on the starboard side is also done when overtaking. When the relation is crossing from left, the ASV should be able to just keep on going with a constant velocity and heading. In other words, $f_{COLREGs}(u_i, \psi_j)$ is returned true if the relation is either head on, overtaking or crossing right and the velocity pair lies in $\mathcal{V}_1$.

VO is a method easily implemented together with COLREGs since every velocity pair are already being checked whether they lie inside the cone or not. By looking the relative bearing between the ASV and the obstacle, see Figure 3.5, the current COLREGs situation can be determined. By knowing the COLREGs situation together with expected behavior in the different situations (Section 3.5.2) the velocity pairs laying within the velocity cone and within regions that violate COLREGs rules are heavily penalized in the cost function and thereby preventing them from being chosen as the desired velocity pair. The reason behind heavily penalizing the velocity pairs inside the velocity cone and in regions contradicting COLREGs instead of just removing them is in case of a situation where all velocity pairs lies within either one of them. In that case, because both f_{COLREGs} and $f_{\text{collision}}$ are returned as true if the velocity pair lies within the VO cone and just f_{COLREGs} being true if the pair violate just COLREGs, the ASV will pass in front of an obstacle and violate COLREGs instead of colliding.

5.2.5 Collision Avoidance - Set-based Guidance

Since SBG is the main contribution in this thesis, a detailed description of the implementation of SBG was given in a separate chapter, that being Chapter 4.

Chapter 6

Results

As mentioned in section 3.5.2, we consider thee different kinds of interactions that can take place between the ASV and obstacles; head on, crossing and overtaking. In this chapter, all the different situations are presented for both VO and SBG. A scenario with a combination of situations has also been simulated and is going to be the basis for a more in depth representation of the data. In this thesis, the obstacles have constant velocity and heading throughout the entire simulation and they are not giving way in the cases where they normally should. Still, the collision avoidance methods should be able to avoid the obstacles even if they do not uphold COLREGs, but the situations are going to look slightly different then described in Section 3.5.2.

All the situations resulted in videos which can be seen on YouTube[1], but figures made by still frames from the videos are presented in this chapter.

6.1 COLREGs situations with Velocity Obstacle

In Figure 6.1 - 6.3, results for VO in head on, overtaking and crossing situation, respectively, are displayed. The blue boat represents the ASV and the red a moving obstacle. The blue, dotted circles are waypoints for the ASV and the red lines emerging from the ASV are the VO cone belonging to the obstacle. One can see that the cone changes size with respect to the position of the obstacle relative the ASV, as explained in Section 3.7.

In the head on situation, the ASV immediately changes its heading to the starboard side. The heading is kept quite constant until the vessels have passed each

[1] `https://www.youtube.com/playlist?list=PL8iPY5FBK9hH2edjxISr7Jg5EibDhHbuQ`

other. Afterwards, the ASV turns back towards the desired path and reaches the waypoint.

When it comes to the overtaking situation, the ASV changes course in order to pass the obstacle on starboard side. It can be seen that the heading wiggles slightly. When the ASV is aligned with the obstacle, it starts to move back towards the desired path, being careful not to cut too close in front of the obstacle.

In the crossing situation, the ASV moves a bit slower in the beginning, keeping a straight path forward, apart from the wiggling. This behavior made by the ASV is to prevent cutting in front of the obstacle. When the obstacle has passed, the ASV changes its course immediately back up towards the desired path. It cuts fairly close behind the obstacle. This is because passing behind the obstacle is deemed as safe as they are heading in completely different directions.

6.2 COLREGs situations with Set-Based Guidance

Figure 6.4 - 6.6 show collision avoidance with SBG in head on, overtaking and crossing situation, respectively. As in the previous section, the red boat represents the obstacle and the blue boat and dotted circles represent the ASV and its way points. The two red, dotted circles are the inner and outer safety radius explained in Chapter 4.

In the head on situation, the ASV turns to starboard when it enters the outer safety radius of the obstacle. It cuts close to the inner safety radius, but is still on the outside. The heading is kept steady until a clear path to the next way point is available. The clear path lies outside the inner safety radius and the ASV changes course back towards desired path.

For the overtaking situation, the ASV's starting position is already inside the outer safety radius, but it still manages to turn to starboard and steer clear of the inner safety radius of the obstacle. When safely past the obstacle, it starts turning back to the desired path without cutting too close in front of the obstacle, or in other words still avoiding the inner safety radius.

In the crossing situation, the ASV needs to do a pretty sharp turn when entering the outer safety radius of the obstacle. It passes the obstacle outside the inner radius and turns back towards the desired path when it is clear of the inner safety radius.

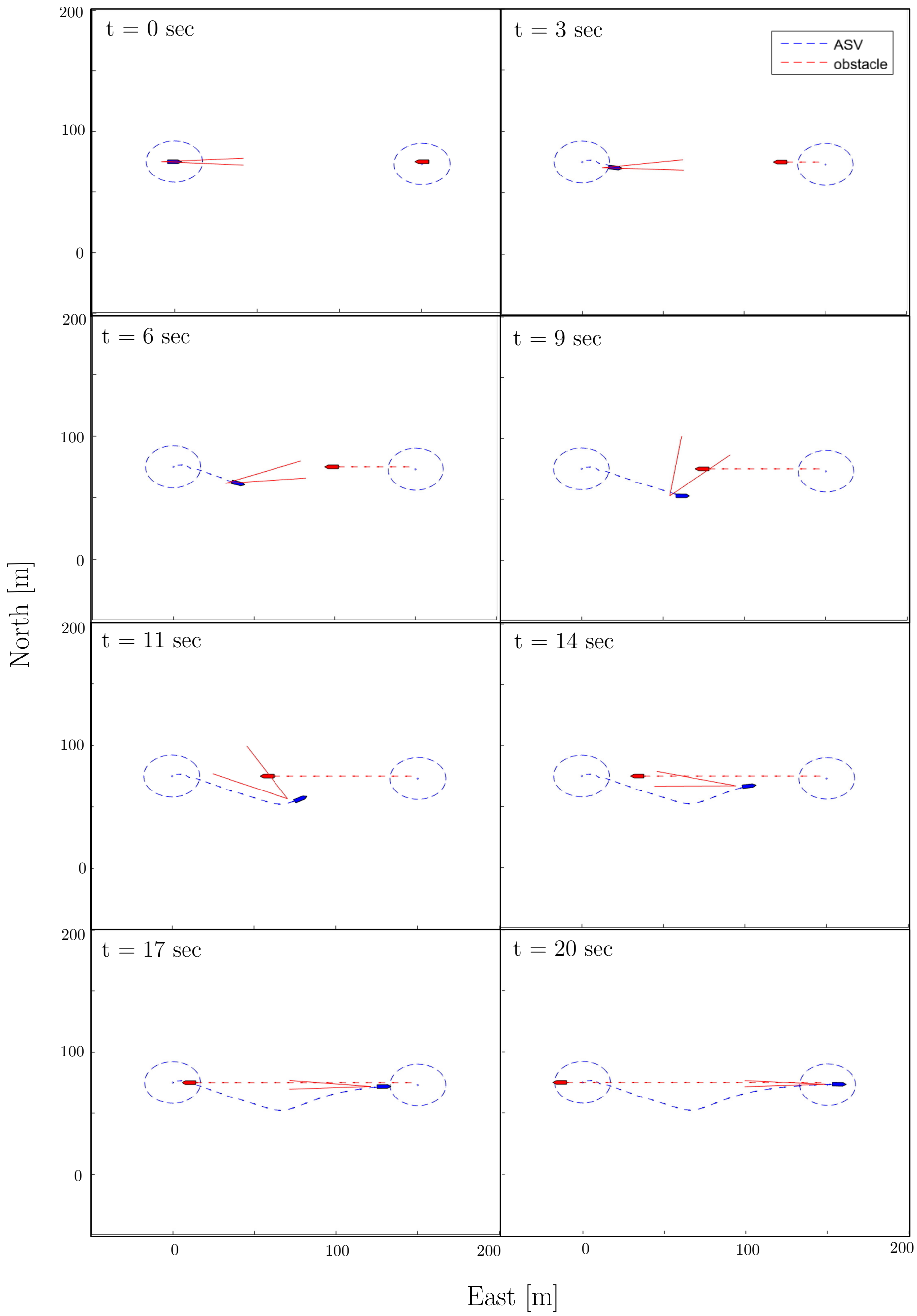

Figure 6.1: Head on situation with Velocity Obstacle.

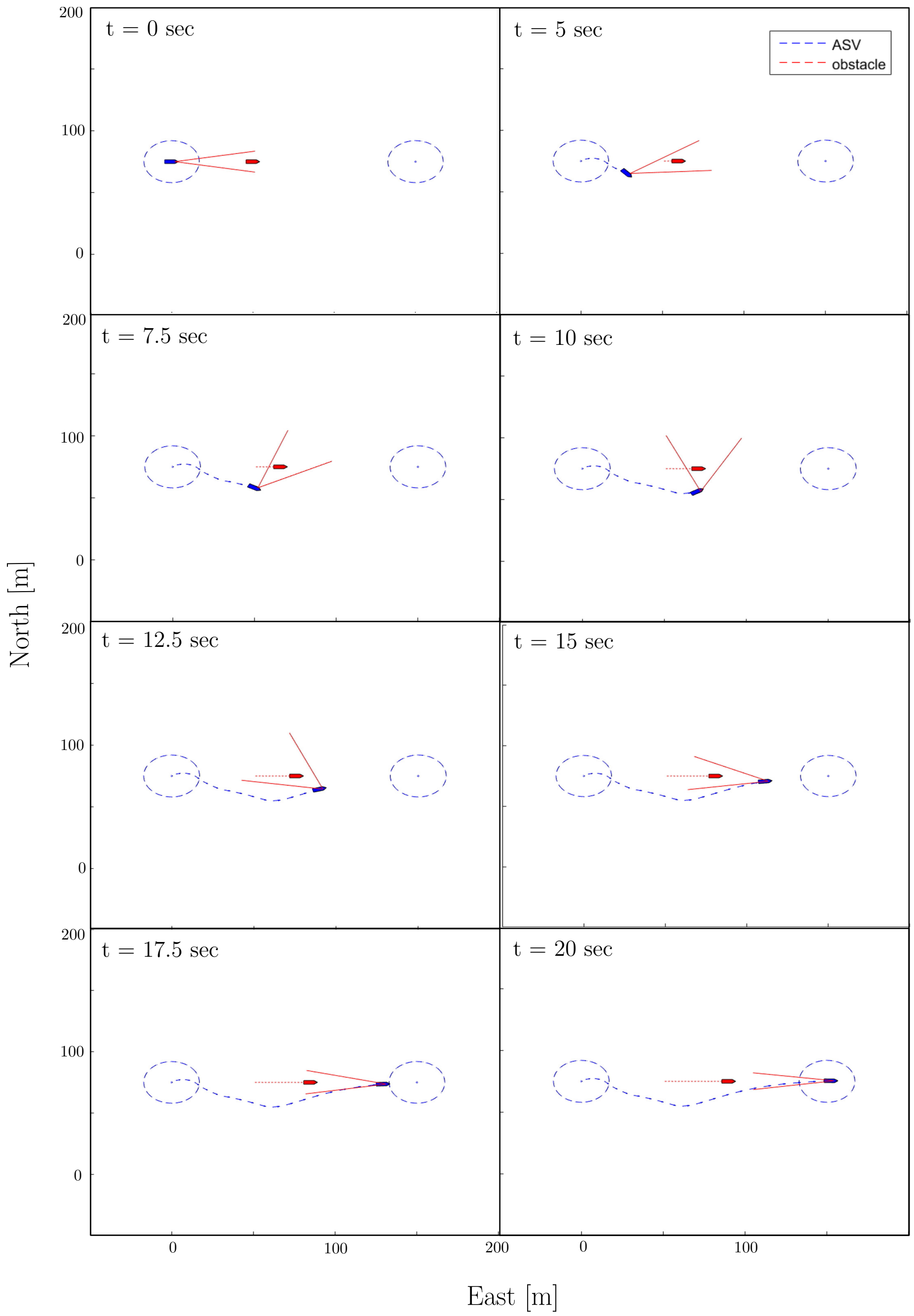

Figure 6.2: Overtaking situation with Velocity Obstacle.

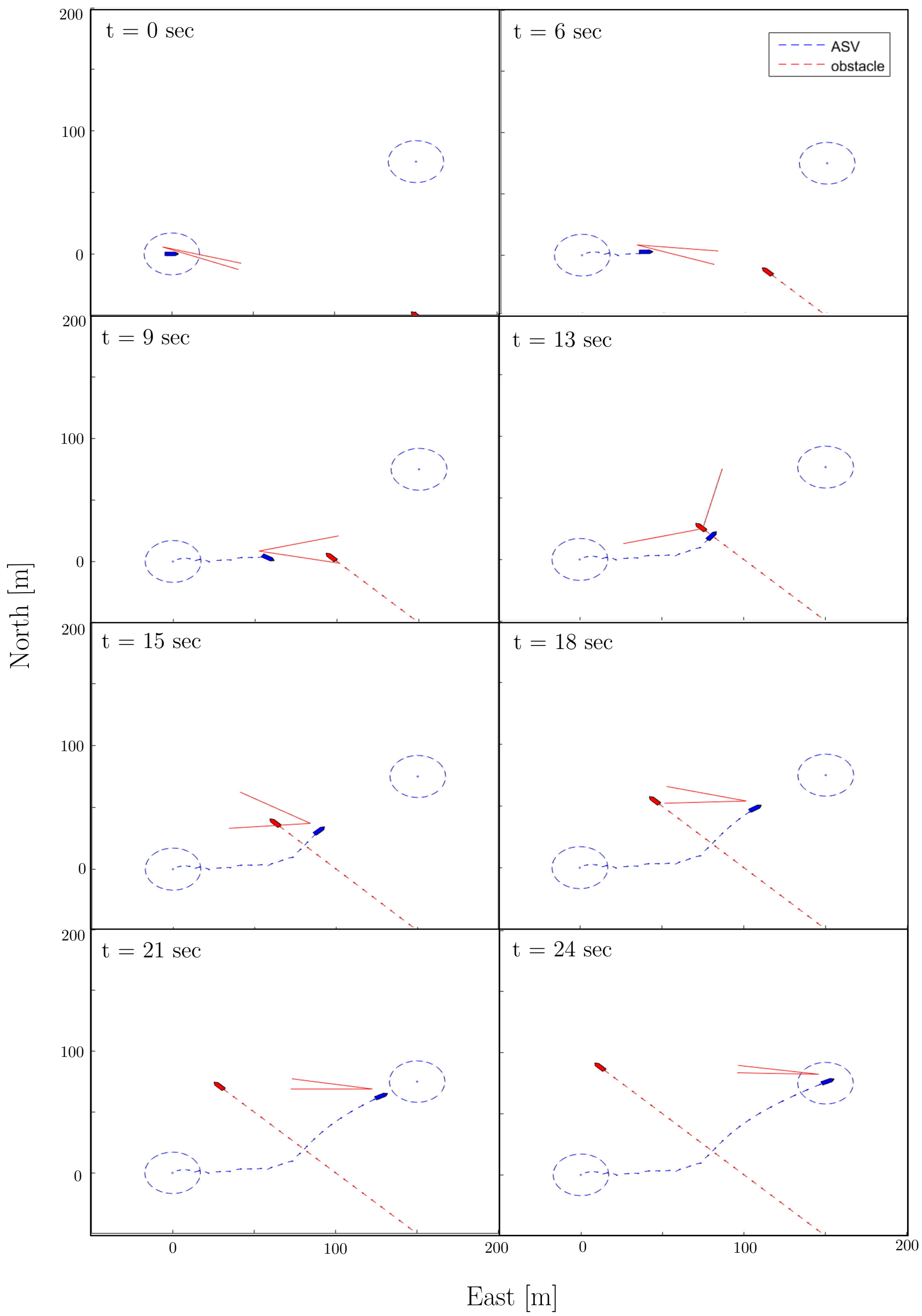

Figure 6.3: Crossing situation with Velocity Obstacle.

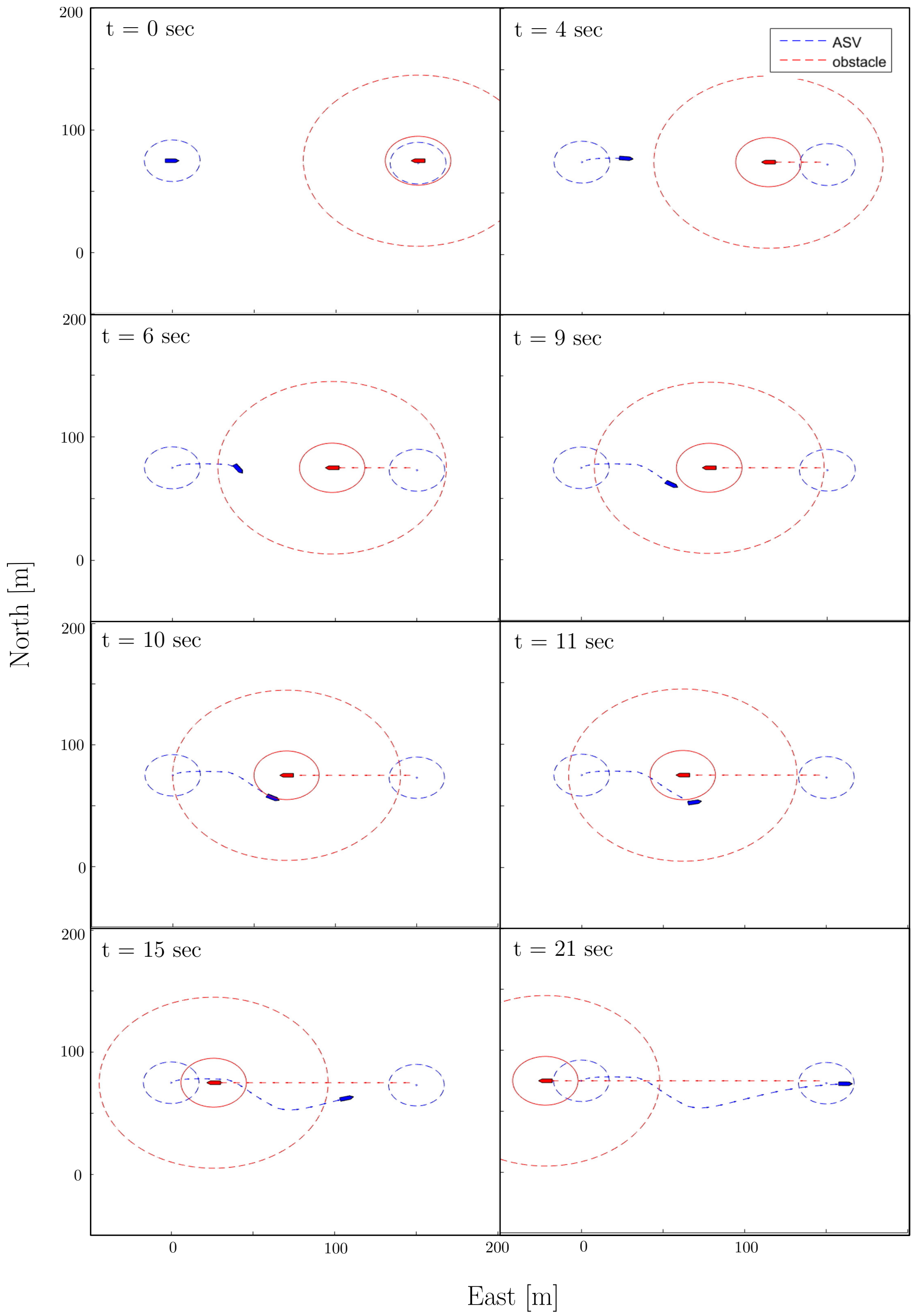

Figure 6.4: Head on situation with Set-Based Guidance.

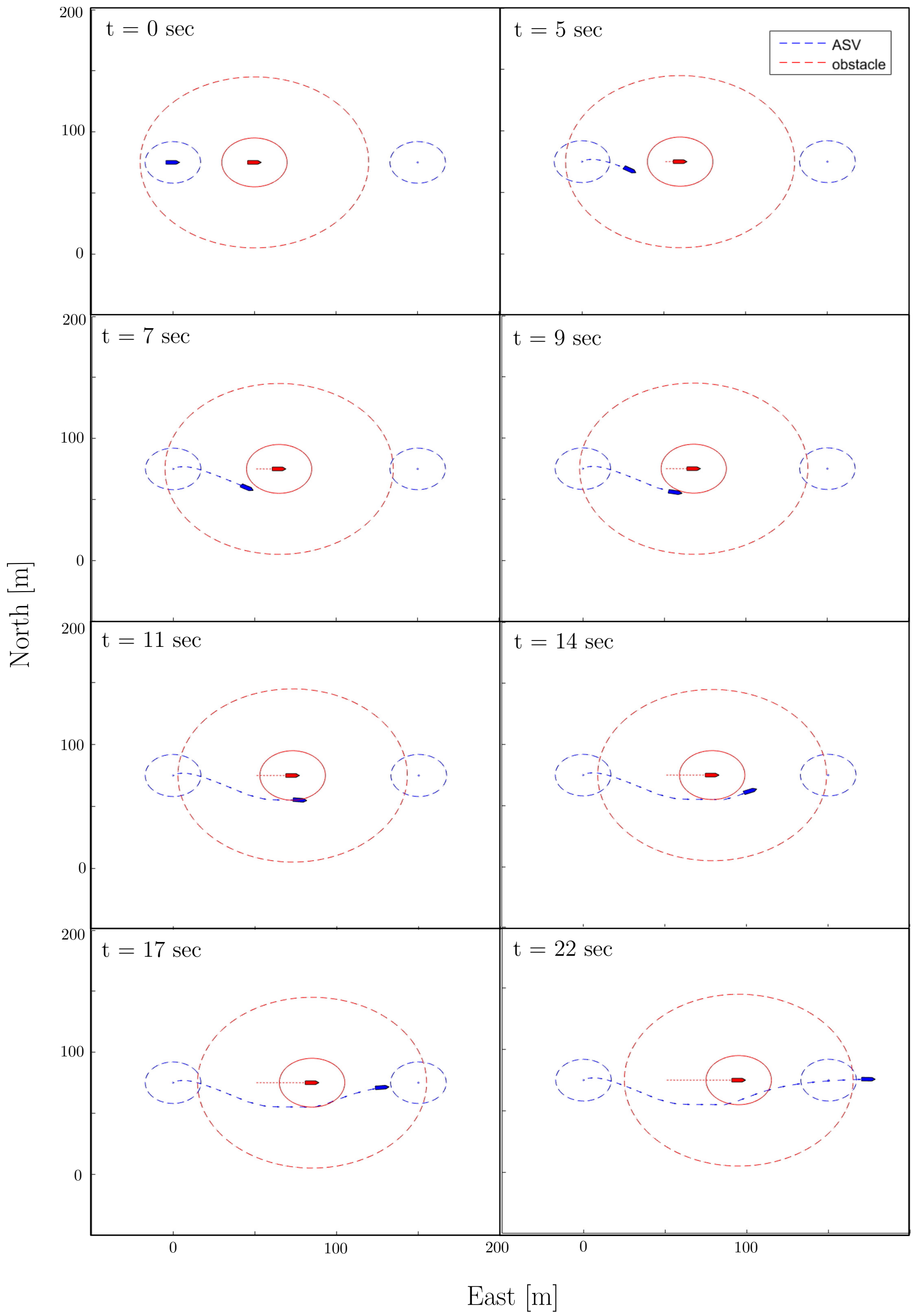

Figure 6.5: Overtaking situation with Set-Based Guidance.

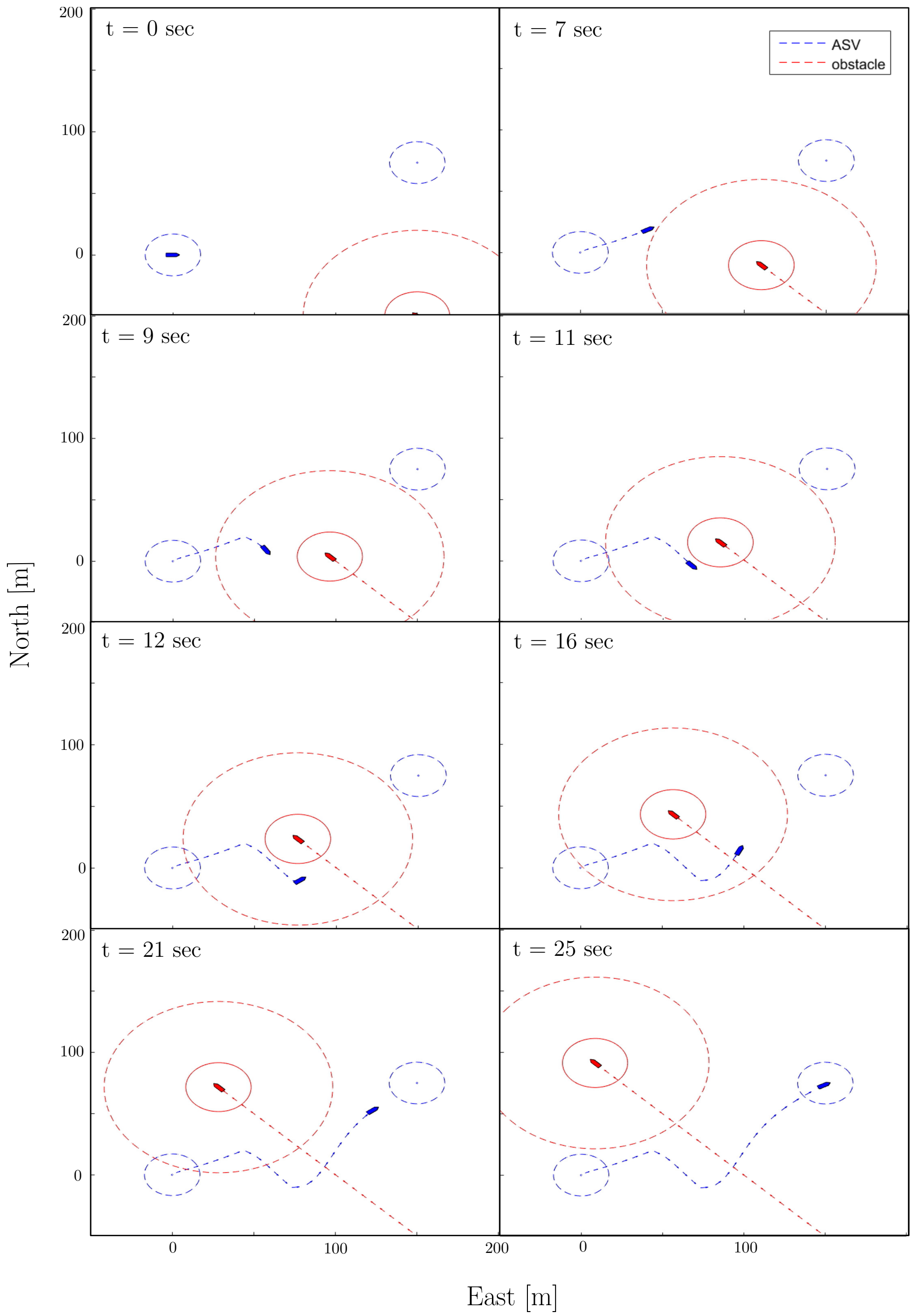

Figure 6.6: Crossing situation with Set-Based Guidance.

6.3 In depth representation of Velocity Obstacle and Set-Based Guidance

A scenario with a combination of COLREGs situations and with two obstacles has been simulated and the result is shown in this section. The two obstacles are represented with the red and green boats. In the case of VO, the VO cones are the same color as the respective obstacles they belong to, making it easy to distinguish between them. In this combined situation, the ASV first encounters a head on situation with the first obstacle, and afterwards, a crossing from right with the second one. The results form the simulations with SBG as collision avoidance method can be seen in Figure 6.7 and with VO in Figure 6.8.

In Figure 6.9, the difference in the paths chosen by the ASV when only doing path following, with no regards to any obstacles, and when using a collision avoidance system can be seen. Both SBG and VO have been used and can be seen in separate parts of the figure. The graphs show that SBG differs more from path following than VO, SBG also have sharper turns and moves in straight lines.

Figure 6.10 shows the difference in position between pure path following and the use of the collision avoidance methods. The difference is shown with respect to time, which means that a constant value in the plot indicates that the ASV is following the desired path, but some time after the path following case. Since SBG have a larger difference than VO in the end, it means that it needs more time before settling on the desired path. The reason for this can be that it either is moves slower, makes lager evasive maneuvers or a combination of both.

The velocity of the ASV with SBG and VO can be seen together with velocity during path following in Figure 6.11. The desired velocity in both cases are $10\,m/s$. Both SBG and VO have velocities close to the path following velocity the entire simulation. But while SBG just have some small deviations when turning, VO has a larger time span with small oscillations in the velocity.

The heading angle of the ASV with both collision avoidance methods can be found in Figure 6.12. With SBG, it can be seen that the method is very clear and precise when changing heading, and that it, for the most part, keeps a constant heading the rest of the time. VO, on the other hand, has a lot of oscillations in heading the first 30 seconds, but it does come to rest at the desired heading some time before SBG.

Figure 6.13 shows the control power, also known as $\boldsymbol{\tau}$, that is issued from the low-level controller and used as input in the vessel model. VO has, overall, both more and lager peaks than SBG.

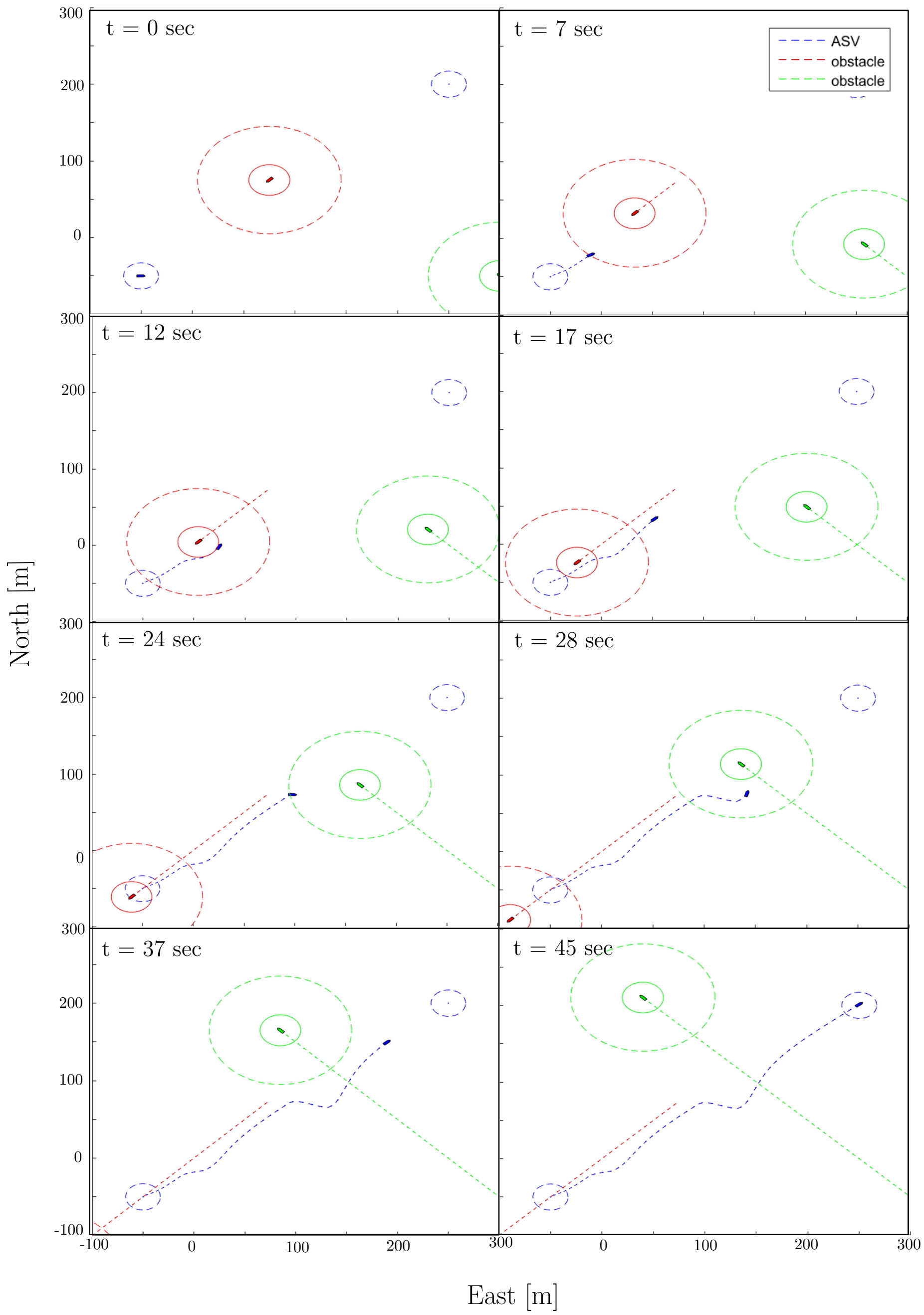

Figure 6.7: Combined situation with SBG.

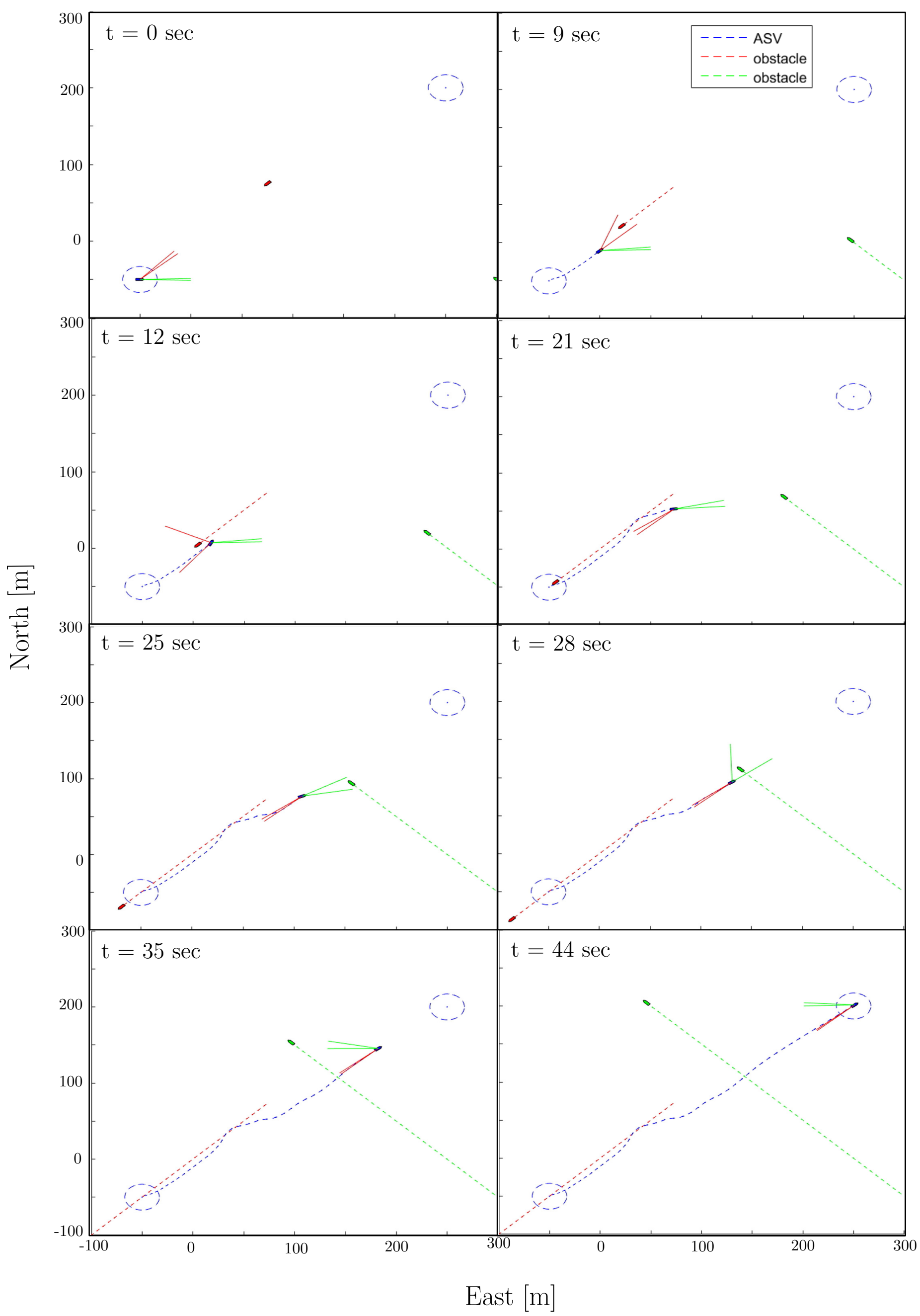

Figure 6.8: Combined situation with VO.

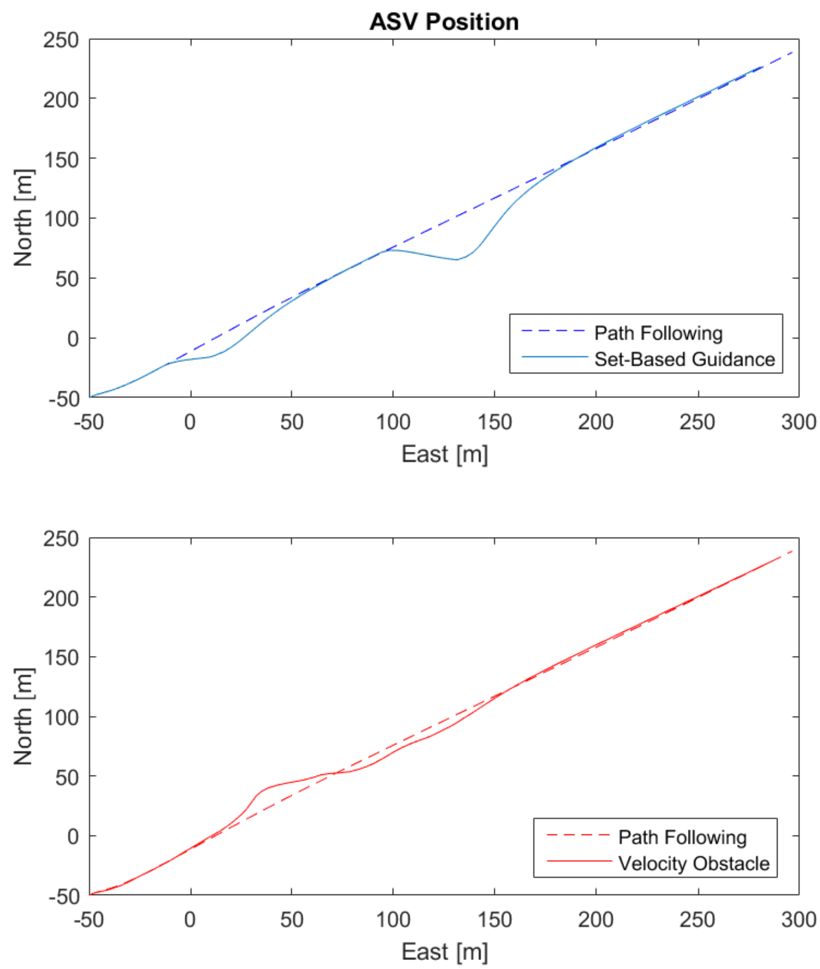

Figure 6.9: Position of the ASV in the combined situation.

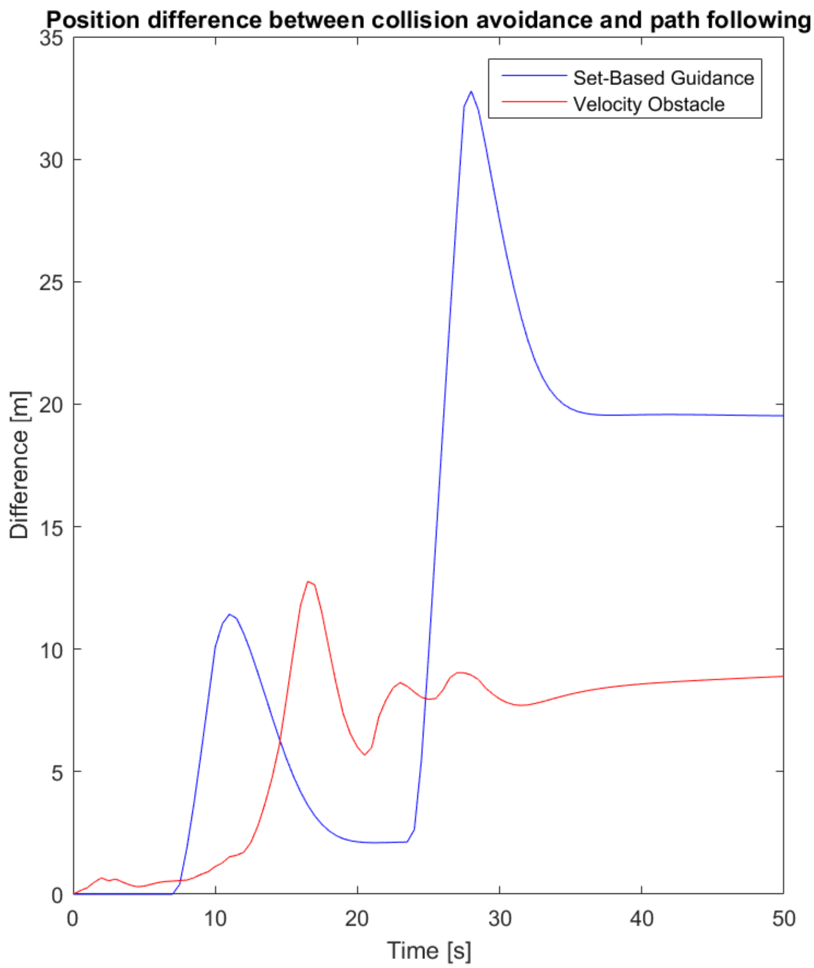

Figure 6.10: Difference in position of the ASV between pure path following and the collision avoidance methods in the combined situation.

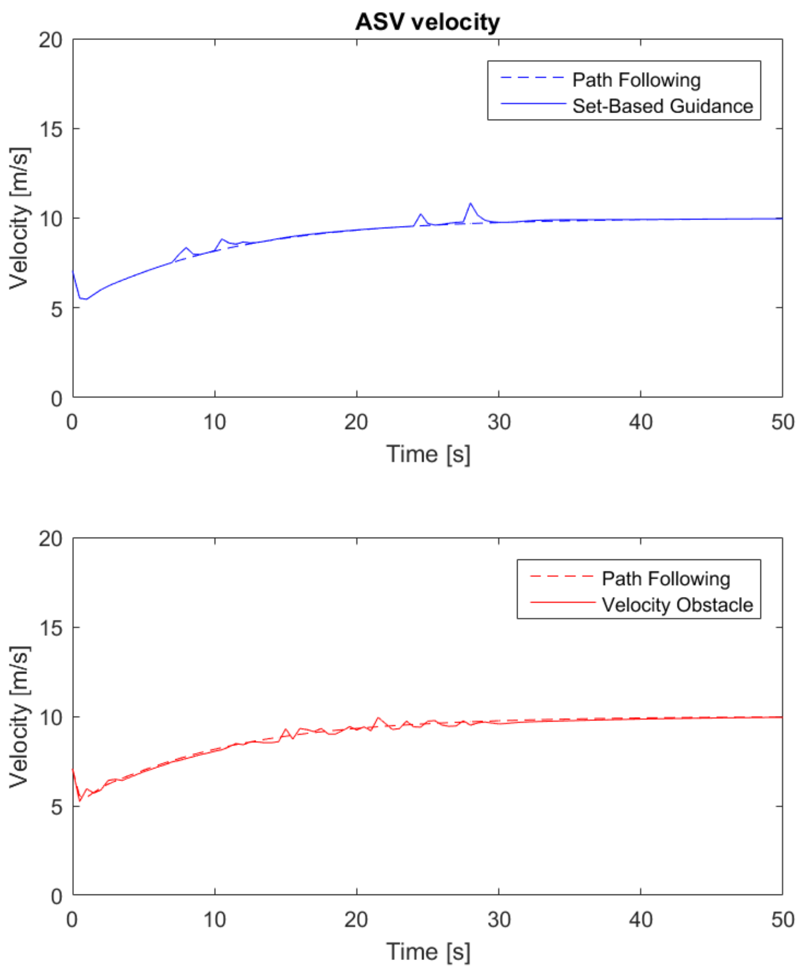

Figure 6.11: Velocity of the ASV in the combined situation.

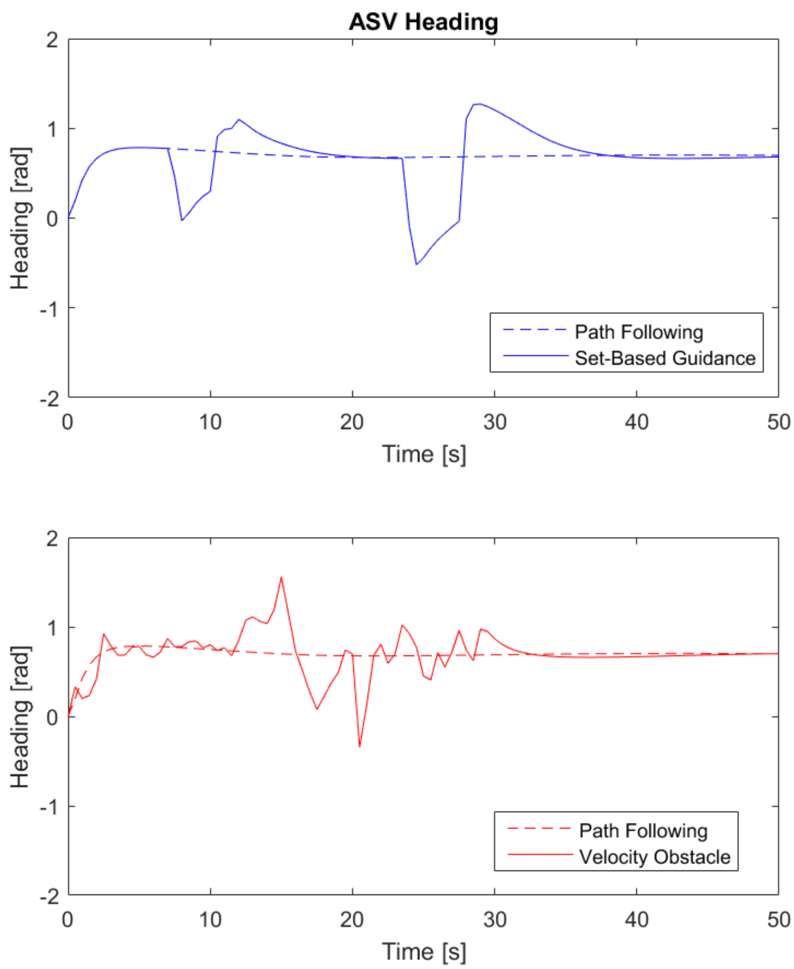

Figure 6.12: Heading of the ASV in the combined situation.

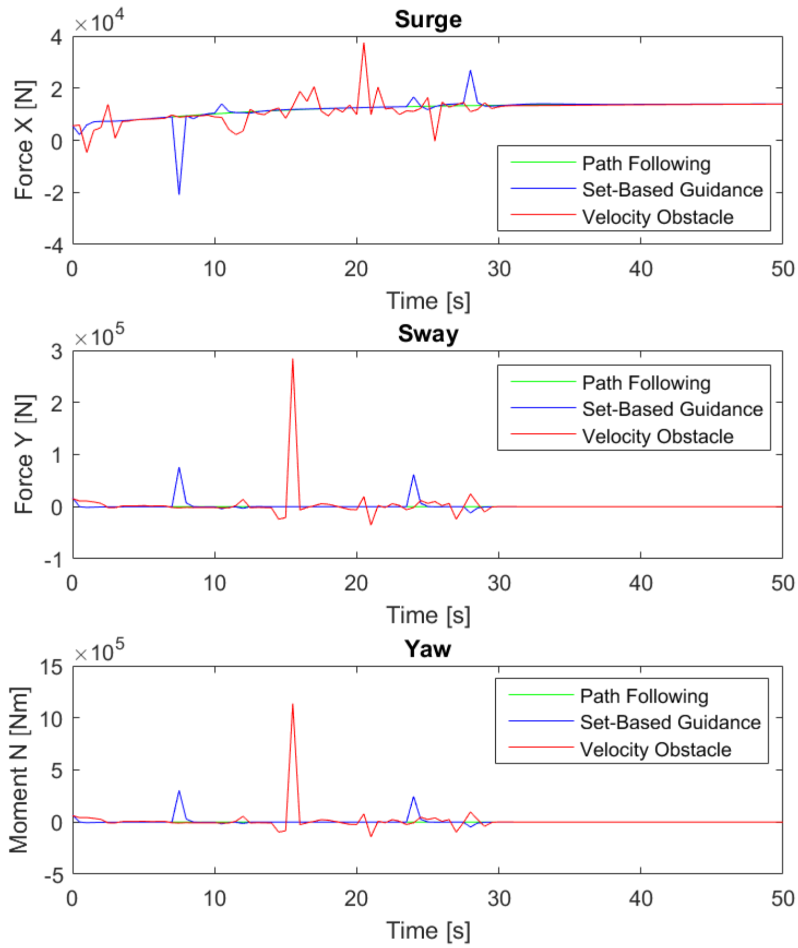

Figure 6.13: Control power in the combined situation.

Chapter 7

Discussion

In Chapter 6, it could be seen that both SBG and VO were able to avoid the obstacles and pass on the right side according to COLREGs rules 13-15. In this chapter, the quality of the avoidances, according to remainder of the main COLREGs rules, are discussed, as well as some general review of the methods.

Since the collision avoidance method based on SBG was the main contribution in this thesis, not a lot of time was spent on tuning VO. It is important to be aware of that possible weakness in the implementation of VO, making it appear worse than it is.

According to rule 8(b) in COLREGs, any changes in course and speed during an evasive maneuver should be large enough to be perceived by other vessels, also, a succession of small changes should be avoided as far as possible. This rule gives SBG a clear advantage by having such precise movements with a constant velocity and that it moves in straight lines for the most part. VO on the other hand falls short when it comes to this rule. The rapid changes in heading seen in Figure 6.12, and also, to some degree, in velocity seen in Figure 6.11, are exactly the kind of behavior that is undesired from both the COLREGs rule, but also when it comes to increasingly wear and tear on the ASV during real life operations. The reason behind this behavior is that, in standard implementations of VO, the heading and velocity are picked from a discrete selection of samples distributed equally around the ASV, as seen in Figure 5.3. If the best option for heading or velocity lies between two samples it can result this wiggling behavior. It can perhaps be solved by adding more samples, but that is still not a very good solution as the rule states that changes in heading and velocity are supposed to be noticeable, something that is not the case if the distance between the samples are to small.

COLREGs rule 16 states that the give-way vessel should take early and substantial action in order to avoid any collisions. VO uphold this rule better than

SBG. While VO starts avoiding obstacles right away, as long as the CPA requirements in Equation 3.34 are fulfilled, SBG does not take any action before the ASV crosses the outer safety radius of the obstacle. This is a drawback with SBG, not only because early action can shorten the time used in passing the obstacle, but also because the ASV can cause confusing situations by not reacting as soon as it perhaps should. An example of such a problem can be seen in Figure 6.6, where the ASV has an obstacle crossing from right, but it does not reach the outer radius before it is almost in front of the obstacle. Even though the ASV changes course right away when entering the radius and avoids the obstacle, its behavior in the time running up to the course change must seem very confusing to the other vessel. It might think that the ASV is not going to give way and start doing evasive maneuvers it self.

The way a stand-on vessel should act is explained in COLREGs rule 17. The rule says that it should keep a steady velocity and heading whilst being avoided by the give-way vessel, but also that it must take action if it becomes apparent that the give-way vessel is not upholding COLREGs. VO has the advantage of preventing collisions even though the obstacle don't uphold COLREGs. By "forbidding" every velocity pair that lies within the cone, VO makes sure that a velocity pair leading to a collision can't be chosen. This works as a safety net if a give-way obstacle do not follow COLREGs rules. But this safety feature is also a problem for VO, because it will make the ASV move out of the way in situations where rule 17 tells it to keep a steady velocity and heading whilst the other vessel performs an evasive maneuver. SBG on the other hand, will keep steady in case of being the stand-on vessel, but do not move out of the way if the give-way vessel do not uphold COLREGs. A suggestion that could give SBG the advantage that VO has, whilst not adopting the problem, is listed in Chapter 9.

A drawback with SBG is that it can not handle situations where two or more obstacles have overlapping outer safety radii. This is due to the risk of getting stuck between them. VO however, can avoid several obstacles at once, also making it possible to avoid them all with one maneuver, instead of handling them in turn like SBG. Because of the large area covered by every obstacle's safety radius in SBG, and by not being able to handle overlapping obstacles, SBG does not work well in situations with many obstacles close together.

Chapter 8

Conclusion

From a COLREGs perspective, both SBG and VO have their strengths and weaknesses. Whilst SBG is more in tune with rule 8(b) by moving in straight lines and keeping a constant speed, VO uphold rule 16 by taking actions to avoid obstacles earlier than SBG. Rule 17 consists of two parts, where SGB uphold the first one by keeping a constant course and speed when being the stand-on vessel, but not the second one since it does not have a safety net in case the other vessel does not uphold COLREGs. VO is the other way around.

VO is faster because it needs less space to pass an obstacle, but in most cases at open sea, lack of space is not going to be an issue, and there is no reason to spend as little time and space as possible when avoiding the obstacle. In those cases SBG would be at its best. If the situation is a more crowded and narrow space where fast and efficient evasive maneuvers are required, VO would to the better job. If one pictures using the collision avoidance method in, for example, a cargo transportation across a sea, the vast majority of time would be spent in situations close to the SBG favored case, while a small part would resemble the VO favored one. But still, both part is equally important. A solution could possible be to have more than one collision avoidance method and switch between them according to the situation.

If a choice between the two collision avoidance methods had to be done with only the features implemented in this thesis, the choice would be VO, for the simple reason of having a better way at handling unpredicted behavior by other vessels, as well as being more time efficient and better at handling several, close obstacles. But when that is said, SBG has a lot of potential and already has a foundation better customized for many of the COLREGs rule. With some small additions and some further development, SBG has a real chance of becoming a better collision avoidance method than VO.

Chapter 9

Further Work

Since Set-Based Guidance is the main collision avoidance method in this thesis, this chapter is going to focus on suggestions on further work that could improve this method.

- SBG should be further developed in order to manage more than one obstacle at once. This can be done by turning two overlapping safety radii into one bigger radius. The bigger radius should be based on the worst case scenario, which means that the new radius should be constructed as if the two overlapping obstacles is barely touching, even if that is not the case. This is so that the shared radius don't change size all the time, but stay one size until the two obstacles are not overlapping any more.
- The SBG method implemented in this thesis is dependent on the obstacle following COLREGs in crossing from left situations, where the obstacle should give way to the ASV. This is only a problem in the crossing from left situations because the ASV is already doing sufficient evasive maneuvers in the other situations, even if the obstacle should not uphold COLREGs. One way of ensuring no collisions in crossing from left situations, is to add a new radius inside $R_{s,outer}$, but outside $R_{s,inner}$. If the ASV crosses the new radius during a crossing from left situation, it can be assumed that the obstacle is not upholding COLREGs and an evasive maneuver is triggered. In this case the ASV should pass on the port side of the obstacle.
- More testing and tuning of the system could improve the system.

Bibliography

[1] S. Moe, A. R. Teel, G. Antonelli, and K. Y. Pettersen, "Experimental results for set-based control within the singularity-robust multiple task-priority invers kinematics," in *2015 IEEE Conference on Robotics and Biomimetics, Zhuhai, China, December 6-9, 2015*, pp. 1233–1239, 2015.

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

[3] Y. Kuwata, M. T.Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with colregs, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2014.

[4] C. C. Baker and A. K. Seah, "Maritime accidents and human performance: the statistical trail," in *Presented at MARTECH 2004, Singapore, September 22-24, 2004*, pp. 225–239, 2004.

[5] S. Moe, A. R. Teel, G. Antonelli, and K. Y. Pettersen, "Stability analysis for set-based control within the singularity-robust multiple task-priority inverse kinematics framework," in *2015 IEEE 54th Annual Conference on Decision and Control (CDC), Osaka, Japan, December 15-18, 2015*, pp. 171–178, 2015.

[6] B. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig, "A modified dynamic window algorithm for horizontal collision avoidance for auvs," *IEEE Multi-Conference on Systems and Control 2016*. forthcoming.

[7] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[8] Wikipedia, "Vector field histogram." `https://en.wikipedia.org/wiki/Vector_Field_Histogram`, 2015. [Online; accessed 04-May-2016].

[9] O. Khatib, "Real-time obstacles avoidance for manipulators and mobile robots," in *International Conference on Robotics and Automation, March, 1985*, pp. 500–505, 1985.

[10] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *International Conference on Robotics and Automation, Sacramento, California, April, 1991*, pp. 1398–1404, 1991.

[11] I. Ulrich and J. Borenstein, "Vfh+: Reliable obstacle avoidance for fast mobile robots," *Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[12] I. Ulrich and J. Borenstein, "Vfh*: Local obstacle avoidance with look-ahead verification," in *International Conference on Robotics and Automation, San Francisco, CA, April, 2000*, pp. 2505–2511, 2000.

[13] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4.1, pp. 23–33, 1997.

[14] Wikipedia, "A* search algorithm." `https://en.wikipedia.org/wiki/A*_search_algorithm`, 2016. [Online; accessed 04-May-2016].

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

[16] S. M. Lavalle and J. J. K. Jr., "Randomized kinodynamic planning," *The International Journal of Robotics Research (IJRR)*, vol. 20, no. 5, 2001.

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

[18] Wikipedia, "Rapidly exploring random tree." `https://en.wikipedia.org/wiki/Rapidly_exploring_random_tree`, 2016. [Online; accessed 06-May-2016].

[19] 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, 1998.

[20] X. L. Yi Cao, Ke Lu and Y. Zang, "Accurate numerical methods for computing 2d and 3d robot workspace," *International Journal of Advanced Robotic Systems*, vol. 8, no. 6, pp. 1–13, 2011.

[21] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley & Sons, Inc, 2006.

[22] W. Caharija, K. Y. Pettersen, A. J. Sørensen, M. Candeloro, and J. T. Gravdahl, "Relative velocity control and integral line of sight for path following of autonomous surface vessels: Merging intuition with theory," *Journal of Engineering for the Maritime Environment*, vol. 228, no. 2, pp. 180–191, 2014.

[23] I. M. Organization, "Convention on the international regulations for preventing collisions at sea, 1972 (colregs)." `http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx`. [Online; accessed 06-June-2016].

[24] *COLREGS - International Regulations for Preventing Collisions at Sea.*

[25] J. Colito, "Autonomous mission planning and execution for unmanned surface vehicles in compliance with the marine rules of the road," Master's thesis, University of Washington.

[26] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman, "A method for protocol-based collision avoidance between autonomous marine surface craft," *Journal of Field Robotics*, vol. 23, no. 5, pp. 333–346, 2006.

[27] T. Stenersen, "Guidance system for autonomous surface vehicles," Master's thesis, Norwegian University of Science and Technology.

[28] Øivind Aleksander G. Loe, "Collision avoidance for unmanned surface vehicles," Master's thesis, Norwegian University of Science and Technology.

[29] S. J. Guy, J. Chhugani, C. Kim, M. Lin, D. Manocha, and P. Dubey, "Clearpath: Highly parallel collision avoidance for multi-agent simulation," *Eurographics/ ACM SIGGRAPH Symposium on Cumputer Animation*, 2009.

[30] MathWorks, "polyxpoly." `http://se.mathworks.com/help/map/ref/polyxpoly.html`. [Online; accessed 04-June-2016].

[31] MathWorks, "Matlab." `http://se.mathworks.com/products/matlab/`. [Online; accessed 20-May-2016].

[32] MathWorks, "Simulink." `http://se.mathworks.com/products/simulink/`. [Online; accessed 20-May-2016].