NTNU
Kunnskap for en bedre verden

KANDIDAT

Kjerstad Kristian (10080)

PRØVE

TTK4550 1 Teknisk kybernetikk, fordypningsprosjekt

Emnekode	TTK4550
Vurderingsform	Arbeider
Starttid	19.08.2019 07:00
Sluttid	17.12.2019 14:00
Sensurfrist	17.01.2020 22:59
PDF opprettet	01.06.2020 11:51

Seksjon 1

Oppgave	Tittel	Oppgavetype
1	Ny oppgave	Filopplasting
2	Ny oppgave	Filopplasting

1 Ny oppgave

1. Prosjektoppgave

Last opp din prosjektoppgave her.

Din fil ble lastet opp og lagret i besvarelsen din.

Last ned | Fjern | Erstatt

Filnavn:	Prosjektoppgave_Kristian Kjerstad.pdf
Filtype:	application/pdf
Filstørrelse:	3.24 MB
Opplastingstidspunkt:	17.12.2019 13:51
Status:	**Lagret**

2

Ny oppgave

2. Eventuelt vedlegg

Last opp eventuelle vedlegg til prosjektoppgaven her.

Last opp filen her. Maks én fil.

Følgende filtyper er tillatt: **.zip** Maksimal filstørrelse er **35 GB**.

Velg fil for opplasting

Collision Avoidance System for Ships Utilizing Other Vessels' Intentions

TTK4550 Preliminary project 2019

Kristian Kjerstad

Main supervisor: Edmund Førland Brekke, NTNU
Co-supervisors: Steinar Låg (DNV GL) and Tom Arne Pedersen (DNV GL)

Department of Engineering Cybernetics
Norwegian University of Science and Technology

Abstract

The collision avoidance system for an autonomous ship is of vital importance for navigational safety. This motivates the idea of utilizing other vessels' intentions to improve performance, which is the topic of this preliminary project.

A study on manually operated ships was performed to reveal that one of the main reasons for collisions at sea is fatigue among crew members that leads to not paying enough attention to traffic situations. A literature study on existing collision avoidance methods has also been performed to provide insights on existing collision avoidance methods and how other vessels' intentions can be used to reduce the risk of collision. The initial design of a collision avoidance algorithm was started, where the proposed algorithm is based on the Simulation-Based Model Predictive Control method and considered how the existing method can be extended to include intentions. Furthermore, a simulator has been developed that is able to simulate the behavior of a Platform Supply Vessel in addition to static and dynamic obstacles. The control system, line of sight guidance and ability to detect collisions has been tested to show acceptable performance.

Table of contents

Nomenclature

$\boldsymbol{\eta}$	Position and orientation vector
$\boldsymbol{\nu}$	Velocity vector
$\boldsymbol{\tau}$	Generalized force vector
ψ	Heading angle
χ	Course angle
x	Position along x-axis
y	Position along y-axis
z	Position along z-axis
$\boldsymbol{p}$	Position vector
u	Speed in x-direction
v	Speed in y-direction
r	Yaw rate
$\boldsymbol{R}(\psi)$	Rotation matrix
$\boldsymbol{M}$	Inertial matrix
m	Mass
$\boldsymbol{D}$	Damping matrix
$\boldsymbol{C}$	Coriolis-centripetal matrix
T	Time constant
k_p	Controller gain
β	Relative bearing angle
θ	Sideslip angle
p_k	Waypoint number k
$(\cdot)_d$	Desired value for a variable
$(\cdot)_{ref}$	Reference value for a variable

Abbreviations

AI	=	Artificial Intelligence
COLREG	=	The International Regulations for Preventing Collision at Sea.
VO	=	Velocity obstacles
MPC	=	Model Predictive Control
GNC	=	Guidance, navigation and control
LOS	=	Line of sight
ILOS	=	Integral line of sight
DOF	=	Degree of freedom
NED	=	North-East-Down
VFF	=	Virtual Force Field
MVFF	=	Modified Virtual Force Field
IMO	=	International Maritime Organization
RRT	=	Rapidly-Exploring Random Tree
PD (Controller)	=	Proportional Derivative (Controller)
CPA	=	Closest Point of Approach
PSV	=	Platform Supply Vessel
AIS	=	Automatic Identification System
VHF	=	Very High Frequency
GPS	=	Global Positioning System
ECDIS	=	Electronic Chart Display and Information System
FMI	=	Functional Mock-Up Interface
USV	=	Unmanned Surface Vehicle
MIMO	=	Multi-Input Multi-Output
VTS	=	Vessel traffic service

Chapter 1

Introduction

1.1 Motivation

In recent years, autonomous systems have become more and more popular, and the growth is expected to increase in the years to come (Saracco, 2017). Autonomous systems are used for numerous different applications, ranging from self-driving cars to house cleaning robots. In the maritime industry, autonomous ships have become an active area of research. One of the first large scale research projects on autonomous ships was MUNIN (Maritime Unmanned Navigation through Intelligence in Networks), a feasibility study whose aim was to show the feasibility of an autonomous vessel (MUNIN, 2016).

What are the benefits of autonomous ships? When having no crew members, there is a potential for reducing costs. The MUNIN project predicted a saving of 7 million dollars per autonomous vessel over a 25-year period in crew cost and fuel consumption (O'Brien, 2018). Another big advantage of an autonomous ship over a manually operated ship is the potential for making the voyage safer. According to Allianz Global Corporate & Specialty (2017) it is estimated that between 75% and 96% of all accidents in the shipping sector are caused by human errors. In addition to reducing the number of accidents, having autonomous ships results in reduced consequences of accidents due to fewer human lives being at stake.

One of the challenges with autonomous ships is the integration of autonomous ships into the existing fleet of manually operated ships. Furthermore, existing regulations and laws have not been adapted to apply for autonomous ships yet (Ørnulf Jan Rødseth and Burmeister, 2012). An example of this is the International Regulations for Preventing Collisions at Sea (COLREGs) from International Maritime Organization (1972). These are navigational rules that must be followed to prevent collision between vessels. How-

ever, the COLREGs are formulated in a way that makes them difficult to implement in an automatic navigation system. For instance, many of the COLREGs rules have imprecise formulations. The rules contain keywords like ”safe distance” and ”safe speed”, without defining what these values should be.

Collision avoidance is an important aspect of autonomous ships. For autonomous ships to be commercially accepted, they need to be at least as safe or safer than existing ships (Rolls-Royce, 2017). Having a well-functioning collision avoidance system, that is able to follow the COLREGs, is an important step towards making vessels more safe. Even though COLREGs are written for human-operated vessels, autonomous vessels should also be required to follow these rules. This is based on the assumption that in the near future, both human-operated and autonomous ships need to interwork in the same areas. This is the reason why COLREG compliance is important.

Collision avoidance is a popular area of research. Statheros (2008) gives an overview of different types of collision avoidance methods. There are several existing collision avoidance methods that have been developed to be able to comply with the COLREGs. This includes the COLREG-RRT method from Chiang and Tapia (2018), which is a global path planning method able to find COLREGs compliant trajectories. Other methods worth mentioning are the Modified Virtual Force Field (MVFF) method by Lee et al. (2004) and Velocity Obstacles (VO) by Kuwata et al. (2014). There has also been developed Model Predictive Control (MPC) based methods to solve the collision avoidance problem. One of the first uses of MPC for ship collision avoidance with COLREGs compliance was the Simulation-Based Model Predictive Control (SBMPC) method by Johansen et al. (2016) which, according to Chiang and Tapia (2018), is one of the current state of the art methods for collision avoidance today.

There has been some research on cooperative collision avoidance where a ship can communicate and coordinate its actions with other ships. Tam and Bucknall (2013) developed a cooperative path planning algorithm able to find COLREGs compliant paths, while Li et al. (2019) propose a coordination strategy to solve the collision avoidance problem in situations where a vessel must consider several obstacle ships at the same time. However, there has not been done any research on a collision avoidance method that utilizes other vessels' intentions and at the same time investigates COLREGs compliance and the effect of environmental constraints (such as shallow waters and shorelines) that limits the maneuverability. This motivates the task of developing a collision avoidance system that utilizes other vessels' intentions, to investigate if the performance of existing methods for collision avoidance can be improved.

The hypothesis that the inclusion of other vessels' intentions will improve the performance of the collision avoidance is based on the idea that it will be easier to compute a collision-free trajectory for a ship if the future trajectories of obstacle ships are known. In addition, many existing algorithms, including SBMPC, the future trajectory of obstacles ships are estimated. There are several ways this estimation can be performed. Xu et al. (2011) developed a neural network to predict the future positions of the ship based on ship motion data and Murray and Perera (2018) present and compare the performance

of several data-driven trajectory prediction methods that utilize historical AIS data. Successful inclusion of intentions will make such estimation methods obsolete for the case of collision avoidance.

1.2 Problem description

The goal of this project is to design and implement a navigational guidance system that utilizes information about other vessels' intentions or near-future route plans to improve the performance. The system should be able to plan and execute a safe route to avoid collisions with other vessels, but also avoid grounding with fixed obstacles such as archipelago and be able to act according to international regulations at sea. The main research question to be answered in this project: *is it possible to use other vessels' intentions to improve already existing algorithms for collision avoidance*? As of now, there has not been done an extensive amount of research to answer this question.

In order to answer the question at hand, several tasks must be completed. The work on this project will be completed in two phases. The first phase is the preliminary project, and the second phase is the master's thesis. The work to be done in the preliminary project is the following proposed tasks:

- Perform a literature survey on collision avoidance methods for surface vessels and assess the properties of different approaches for collision avoidance. Consider how other vessels' intentions could be exploited to reduce collision risk.
- Create a simulator for the vessel, the environment, other vessels and demonstrate the capabilities of the developed system. To introduce readiness for further work the following capabilities should be considered implemented:
  - Boundaries for land and "no go" zones.
  - Different future trajectories for targets.
- Do a brief and initial study of how ships are manually operated to avoid collisions and identify the most critical situations and the main reasons for collisions. (To be further elaborated in master's thesis).
- Start the design of a collision avoidance system capable of navigating safely in various types of waters, and to act on suddenly occurring obstacles, as well as utilizing other vessels intentions. (Design to be further developed and implemented in master's thesis).

The remaining work on the project, including the actual implementation and assessment of the collision avoidance system utilizing other vessels' intentions, will be completed in the master's thesis.

1.3 Contributions

The main contribution of this preliminary project is to investigate how ships are manually operated to avoid a collision and find the most common reasons for collisions for manually operated ships. This provides valuable insight into what features are needed in an automatic collision avoidance system. A literature study of existing collision avoidance methods has been performed, and four different collision avoidance methods have been studied in detail (COLREG-RRT, Modified Virtual Force Field (MVFF), Velocity Obstacles (VO) and Simulation-Based Model Predictive Control (SBMPC)). Recommendations on how to improve their performances by utilizing other vessels' intentions have been given. In addition, the initial design of a collision avoidance system based on the SBMPC method has been proposed. This design

A simulator has been developed in MATLAB & Simulink (MATLAB, 2017), and the simulations have been performed in DNV GL's CyberSea platform. A ship model for a Platform Supply Vessel has been implemented, along with a control system and guidance system. The performance of the control and guidance system has been tested. The simulator also includes functionality for adding obstacle ships and static obstacles such as boundaries for land.

1.4 Outline

The theoretical background for this project is given in Chapter 2. In this chapter, important concepts such as a mathematical 3 degree of freedom ship model, line of sight guidance and model predictive control are discussed.

In Chapter 3, details about COLREGs will be given in addition to an explanation of how ships are manually operated in order to avoid collisions. At the end of the chapter, common reasons for collisions at sea are discussed.

Chapter 4 contains a literature study on collision avoidance for surface vessels and gives a detailed explanation of four different collision avoidance methods. For these methods, the possibility to improve performance by utilizing other vessels' intentions has been investigated.

The initial design of a collision avoidance system that utilizes other vessels' intentions is given in Chapter 5. The design is based on the existing SBMPC algorithm and focuses on how intentions can be included. In addition, data about intentions and requirements for the algorithm will be discussed.

Chapter 6 describes the development of a simulator and the ship model used in the simulator. A 3 degree of freedom ship model of a platform supply vessel has been implemented, and its maneuverability performance has been tested by using one of DNV GL's

simulators. This chapter explains how the control system and LOS guidance has been implemented, and also describes how static and dynamic obstacles are included.

The developed simulator's performance has been tested, and Chapter 7 contains the simulator testing results for the control system and guidance system. The ability to detect collisions is also investigated.

Chapter 8 gives a summary of the conclusions that can be drawn from the project and Chapter 9 gives an overview of further work to be done.

Chapter 2

Background theory

This chapter will present the relevant background theory for this project. The chapter starts with an introduction to guidance, navigation and control. A mathematical 3 degrees of freedom ship model, line of sight guidance and feedback linearizing control will also be presented. Towards the end of the chapter, a discussion of model predictive control will be given in addition to a presentation of the International Regulations for Preventing Collisions at Sea (COLREGs).

2.1 Guidance, navigation and control

This section will provide the necessary theoretical background related to guidance, navigation and control. The material is based on Fossen (2011). A motion control system for a marine craft usually consists of three main components, which communicate with each other through data and signal transmission. These components are called Guidance, Navigation and Control (GNC), and are interconnected as seen in figure 2.1.

The navigation system is responsible for directing the craft by estimating states such as position and heading. This is done by using different kinds of sensors, for instance a global navigation satellite system (GNSS), accelerometers and gyroscopes. Usually, the sensor and navigation data are processed by an observer such as the Kalman filter. This observer is able to perform noise filtering, reconstruction of unmeasured states in addition to prediction of states. The predicted states are used as input to the guidance and control system.

The guidance system is responsible for computing the desired path for the craft to follow, which also includes the desired speed. The desired path computed by the guidance

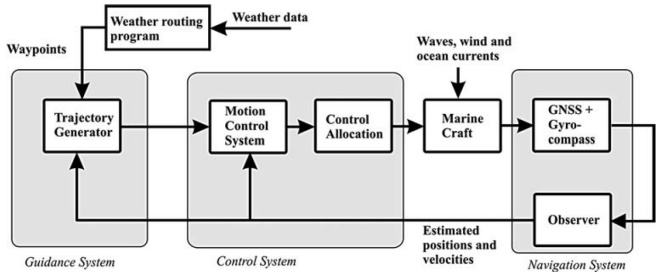


Figure 2.1: Connection between guidance, navigation and control. Figure is taken from Fossen (2011).

system often consists of a series of waypoints and can be approximated by the straight lines between the waypoints. This is illustrated in Figure 2.2. In order to compute the desired path, several types of data will be used, including weather data, estimated states, geographical data (information about shallow waters, shorelines, etc.) in addition to human operator inputs. Often, some kind of optimization technique is used in order to find the optimal path, which can take into account fuel consumption, minimum time to reach destination and collision avoidance.

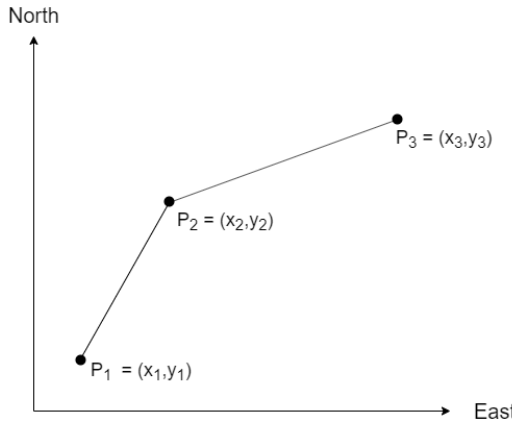


Figure 2.2: The desired path for a marine craft can be specified as the combination of straight lines between waypoints.

The control system takes the desired path (computed by the guidance system) as input and determines the necessary control forces in order to make the marine craft follow the desired path. Typically, such a control system consists of a speed controller responsible for keeping the desired speed and a heading autopilot, which steers the heading of the craft towards the desired heading angle ψ_d. This desired heading angle is the angle that makes the marine craft follow the desired path. A line of sight guidance method is often used to compute ψ_d.

2.2 A 3 degrees of freedom ship model

A model of a ship with 3 degrees of freedom (DOFs) will be used in this project. Such a model will only consider the horizontal plane motion of the ship, where the three degrees of freedom are surge, sway and yaw as seen in Figure 2.3. There also exist more advanced 6 DOF models, but in this project a 3 DOF model will be used since it will provide sufficient accuracy.

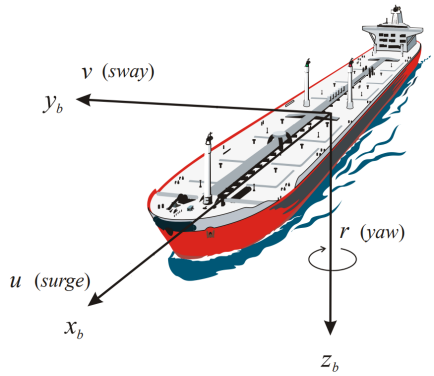


Figure 2.3: The 3 degrees of freedom of a marine craft. Figure is taken from Fossen (2011).

When analyzing the motion of marine crafts, it is convenient to use two different coordinate frames. The first one is the North-East-Down (NED) coordinate frame, denoted $\{n\}$. This coordinate frame has its origin fixed to the surface of the earth, and as the name suggests, the x-axis points north, the y-axis points East and the z-axis points downwards normal to the Earth's surface (towards the Earth's center of gravity). When modeling the marine craft, the NED frame is assumed to be inertial (the Earth's rotation is disregarded). The second coordinate frame is the body-fixed frame, denoted $\{b\}$, whose origin moves with the marine craft. The origin of $\{b\}$ is often chosen to be at the same location as the center of gravity of the craft. The x-axis points in direction from aft to fore, the y-axis points to the right and the z-axis points downwards.

The 3 DOF model can be written in vectorial form as in equation (2.1).

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r &= \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \end{aligned} \tag{2.1}$$

A brief explanation of the terms will be given. For a more in-depth discussion of the physical interpretation of the different terms, the reader is referred to Fossen (2011).

The vector $\boldsymbol{\eta} = [x, y, \psi]^T$ consists of the position (x, y) and heading angle ψ of the ship with respect to $\{\text{n}\}$. $\boldsymbol{R}(\psi)$ is the rotation matrix from $\{b\}$ to $\{n\}$, given as a principal

rotation about the z-axis. The expression for $\boldsymbol{R}(\psi)$ is given by equation (2.2).

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

The velocity vector of the vessel with respect to {b} is given by $\boldsymbol{\nu} = [u, v, r]^T$. u and v are the speed in surge and sway direction, while r is the angular velocity about the z-axis (also called yaw rate). $\boldsymbol{\nu}_r$ is the velocity vector of the craft relative to the velocity of the ocean current.

$\boldsymbol{M} = \boldsymbol{M}_A + \boldsymbol{M}_{RB}$ is the inertial matrix, where $\boldsymbol{M}_{RB}$ is the rigid-body inertia matrix and $\boldsymbol{M}_A$ is the added mass matrix (added mass is a virtual mass added because the vessel must move some volume of fluid as it accelerates or decelerates). The expressions for these matrices are given by equations (2.3) and (2.4)

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

$$\boldsymbol{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{r}} & -N_{\dot{r}} \end{bmatrix} = \begin{bmatrix} A_{11}(0) & 0 & 0 \\ 0 & A_{22}(0) & A_{26}(0) \\ 0 & A_{62}(0) & A_{66}(0) \end{bmatrix} \tag{2.4}$$

where m is the mass of the ship, x_g is the location of center of gravity (CG) along the x-axis of the body frame relative to the chosen origin of the body frame (CO). I_z is the moment of inertia about the z-axis. The elements in the $\boldsymbol{M}_A$ matrix are the hydrodynamic derivatives. For a 3 DOF model, the added mass matrix $\boldsymbol{M}_A$ can be found by using a zero-frequency model. Therefore, the hydrodynamic derivatives can be found from the 6x6 frequency-dependent added mass matrix $\boldsymbol{A}(\omega)$ for frequency $\omega = 0$. In equation (2.4), $A_{ij}(0)$ is the matrix element of $\boldsymbol{A}(\omega = 0)$ in row i and column j.

$\boldsymbol{C}_{RB}(\boldsymbol{\nu})$ is the rigid-body coriolis-centripetal matrix (due to rotation of $\{b\}$ about the inertial frame $\{n\}$), and is given by equation (2.5)

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \tag{2.5}$$

$\boldsymbol{C}_A(\boldsymbol{\nu}_r)$ is the hydrodynamic coriolis-centripetal matrix, given by equation (2.6). It will be assumed that the ocean current has no velocity, then $\boldsymbol{\nu}_r = \boldsymbol{\nu}$

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \boldsymbol{C}_A(\boldsymbol{\nu}) \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v + Y_{\dot{r}} r \\ 0 & 0 & -X_{\dot{u}} u \\ -Y_{\dot{v}} v - Y_{\dot{r}} r & X_{\dot{u}} u & 0 \end{bmatrix} \tag{2.6}$$

$\boldsymbol{D}(\boldsymbol{\nu}_r) = \boldsymbol{D}(\boldsymbol{\nu}) = \boldsymbol{D}\boldsymbol{\nu} + \boldsymbol{d}(\boldsymbol{\nu})$ is the damping matrix, which consists of a linear damping term, $\boldsymbol{D}\boldsymbol{\nu}$, and a nonlinear damping term $\boldsymbol{d}(\boldsymbol{\nu})$. The linear damping is important

for low-speed, while nonlinear damping dominates at higher speed. In this project it will be assumed that the nonlinear damping term is zero. The expression for $\boldsymbol{D}$ is given in equation (2.7).

$$\boldsymbol{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \tag{2.7}$$

$\boldsymbol{D}$ consists of moment coefficients. The diagonal terms in $\boldsymbol{D}$ can be found using equation (2.8)

$$\begin{aligned} -X_u &= B_{11v} = \frac{m + A_{11}(0)}{T_{surge}} \\ -Y_v &= B_{22v} = \frac{m + A_{22}(0)}{T_{sway}} \\ -N_r &= B_{66v} = \frac{I_z + A_{66}(0)}{T_{yaw}} \end{aligned} \tag{2.8}$$

where T_{surge}, T_{sway} and T_{yaw} are time constants in surge, sway and yaw respectively. The off-diagonal terms $-N_v$ and $-Y_r$ are assumed to be zero.

The vector $\boldsymbol{\tau}$ are the generalized forces and moments that act on the ship. $\boldsymbol{\tau}_{wave}$ and $\boldsymbol{\tau}_{wind}$ are environmental forces and moments from waves and wind that act on the ship.

2.3 Path following with line of sight guidance

Path following is the task of following a predefined, desired path independent of time (Fossen, 2011). A popular method to solve the path following problem is LOS guidance. Fossen (2011) describes a lookahead-based LOS steering law that calculates the desired course angle χ_d necessary to make the vessel follow a straight line path between two waypoints $p_k = (x_k, y_k)$ and $p_{k+1} = (x_{k+1}, y_{k+1})$, as shown in Figure 2.4. The position of the craft is given by (x, y) and the goal is to minimize the cross-track error e, which is the shortest distance from the vessel to the path.

The straight line path to follow is always defined by only two waypoints. If the desired path of the vessel consists of more than two waypoints, as shown in Figure 2.2, a switching mechanism is used. For instance, if the desired path consists of three waypoints p_1, p_2 and p_3, the vessel first tries to follow the straight line path between the active waypoints p_1 and p_2. When the vessel reach the waypoint p_2, the switching mechanism changes the active waypoints to p_2 and p_3.

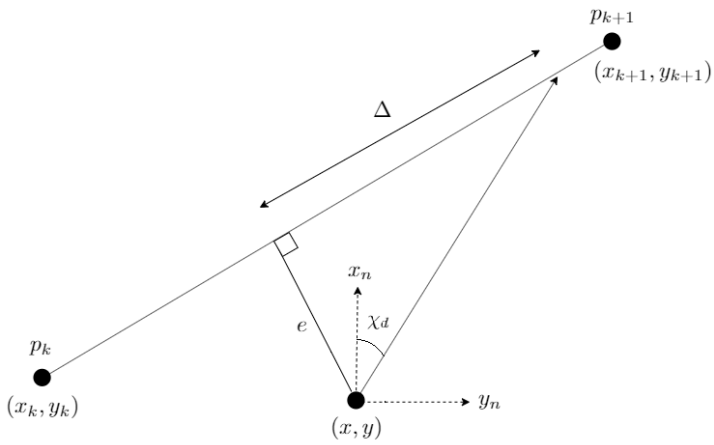


Figure 2.4: Line of sight guidance. The desired path is the straight line path between waypoints p_k and p_{k+1}.

The desired course angle χ_d can be found from equation (2.9)

$$\chi_d = \chi_p + \chi_r(e) \tag{2.9}$$

where $\chi_p = \alpha_k = atan2(y_{k+1} - y_k, x_{k+1} - x_k)$ (called the path-tangential angle) and $\chi_r(e) = arctan(\frac{-e}{\Delta})$. Δ is called the lookahead distance, which can be seen in Figure 2.4. A large Δ will give slow convergence to the desired, straight path. However, a large Δ also reduces amount of oscillations in position around the desired path.

According to Børhaug et al. (2008), the LOS guidance method above has good convergence properties. However, in the presence of environmental disturbances like ocean currents, waves and wind, the LOS guidance method can have convergence problems and experience path deviation. Børhaug et al. (2008) developed a modified version of the LOS guidance method called Integral Line-Of-Sight (ILOS) where use of integral action can counteract disturbances due to ocean currents, waves and wind. The integral action can be included by modifying the expression for $\chi_r(e)$.

2.4 Feedback linearizing controllers

In order to make the ship follow the predefined desired path, a controller for speed and heading is needed. This can be implemented as a feedback linearizing controller. The concept behind a feedback linearizing controller is to transform the nonlinear system dynamics into linear system dynamics (Fossen, 2011). The transformation can be done either in the body frame for velocity control or in the NED frame for position and attitude control.

For the case of velocity control, the linear ship dynamics can be written as $\dot{\boldsymbol{\nu}} = \boldsymbol{a}^b$, where $\boldsymbol{\nu} = [u, v, r]^T$ and $\boldsymbol{a}^b$ can be interpreted as a commanded acceleration vector. From equation (2.1) we have

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu}) = \boldsymbol{\tau} \tag{2.10}$$

where $\boldsymbol{n}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu}$. By choosing

$$\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}) \tag{2.11}$$

we can substitute the expression for $\boldsymbol{\tau}$ in equation (2.11) into equation (2.10) to yield the expression in equation (2.12)

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu}) = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}) \tag{2.12}$$

which can be simplified to $\dot{\boldsymbol{\nu}} = \boldsymbol{a}^b$. The next step is to choose the commanded acceleration $\boldsymbol{a}^b$. A simple choice is a regular proportional (P) controller, shown in equation (2.13)

$$\boldsymbol{a}^b = -\boldsymbol{K}_p\tilde{\boldsymbol{\nu}} \tag{2.13}$$

where $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \boldsymbol{\nu}_d$. $\boldsymbol{\nu}_d$ is the desired velocity vector for the ship to follow and

$$\boldsymbol{K}_p = \begin{bmatrix} k_{p,1} & 0 & 0 \\ 0 & k_{p,2} & 0 \\ 0 & 0 & k_{p,3} \end{bmatrix} \tag{2.14}$$

where $k_{p,i}$ $(i = 1, 2, 3)$ are controller gains.

2.5 Optimization and model predictive control

2.5.1 Optimization

Optimization is concerned with maximizing (or minimizing) an objective function that often is subject to constraints on its variables (Nocedal and Wright, 2006). In other words, there is a search for the best solution. Optimization is used in a number of different areas, ranging from engineering applications such as control to operations research used by the management of organizations. The standard form of an optimization problem is given in equation (2.15). In this case, $f(\boldsymbol{x})$ is the objective function to be maximized, and the problem is solved by finding values for the variables in the vector $\boldsymbol{x}$ that maximize $f(\boldsymbol{x})$ while satisfying the constraints $g(\boldsymbol{x}) \leq 0$ and $h(\boldsymbol{x}) = 0$. The constraints of the problem put limits on how variable values can be chosen.

$$\begin{aligned} &\text{maximize} f(\boldsymbol{x}) \\ &\text{subject to } g(\boldsymbol{x}) \leq 0 \\ &\qquad\qquad h(\boldsymbol{x}) = 0 \end{aligned} \tag{2.15}$$

2.5.2 Model predictive control

Model Predictive Control (MPC) is an iterative control method which solves an optimization problem at each time step. A high level algorithm for MPC from Foss and Heirung (2016) is given below.

1. Get the current plant state $x_{t'}$ (or estimate it).
2. Solve an open loop optimization problem on the prediction horizon from t' to $t'+N$ with $x_{t'}$ as the initial condition. This solution gives a set of state estimates $\hat{x}_{t'}$ to $\hat{x}_{t'+N}$ and the corresponding control inputs $u_{t'}$ to $u_{t'+N-1}$ necessary to make the plant reach these states.
3. Apply the first control input, $u_{t'}$, to the real plant. This will make the plant move from current state $x_{t'}$ to the new state $x_{t'+1}$. Then go back to step 1.

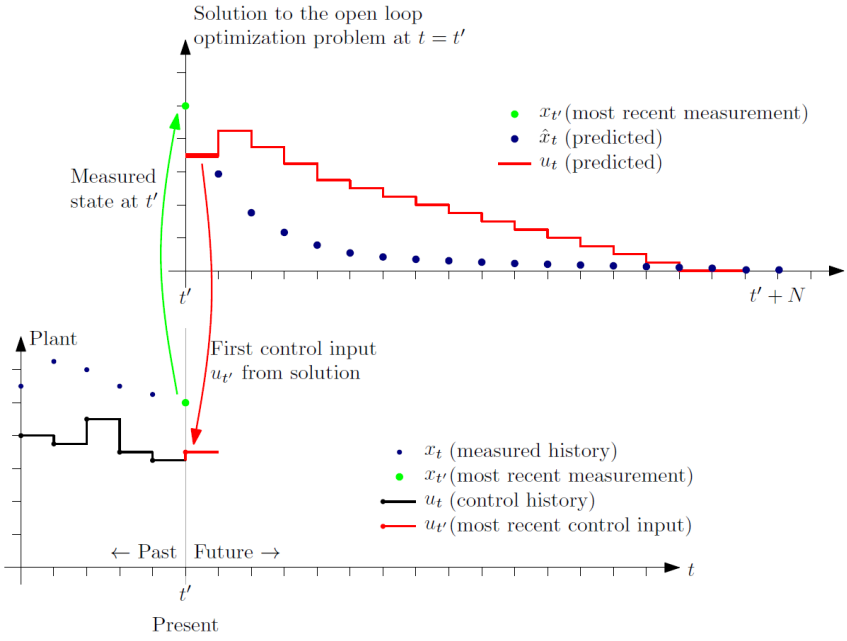


Figure 2.5: Illustration of the MPC principle. The bottom graph shows the history of states for the plant as blue dots and history of control inputs as black bars. At current time t', an optimization problem is solved with x'_t as initial condition, illustrated in the graph on top. The solution of the optimization problem gives a series of control inputs, where the first control input $u_{t'}$ is applied to the plant. Figure is taken from Foss and Heirung (2016).

The MPC principle is visualized in Figure 2.5. The bottom graph in Figure 2.5 shows the history of states and control inputs for the plant. In each time step, an open loop optimization problem is solved on the prediction horizon from t' to $t'+N$ and the solution

of this optimization problem is illustrated in the top graph in Figure 2.5. In order to obtain this solution, a mathematical model of the plant is used. The solution to this optimization problem is a prediction of the future behavior of the system, described as a set of state estimates ($\hat{x}_{t'}$ to $\hat{x}_{t'+N}$) illustrated as blue dots in the top graph in Figure 2.5. In addition, the future control inputs ($u_{t'}$ to $u_{t'+N-1}$) necessary to achieve this predicted behavior are found. They are illustrated as red bars in the top graph in Figure 2.5. The first control input found from the solution of the optimization problem, $u_{t'}$, is applied to the physical plant to steer the system in a way that minimizes an objective function. This objective function is chosen according to the designer's goal. For instance, the designer can decide that he wants the states of the system to stay close to the desired states while keeping control inputs small.

A challenge with using MPC for collision avoidance is that complex collision avoidance scenarios might lead to non-convex optimization problems to be solved, where finding the best solution is difficult. Therefore, implementing an MPC for collision avoidance purposes is challenging. The MPC based collision avoidance system by Johansen et al. (2016) avoids solving a traditional optimization problem with numerical optimization as used in conventional MPCs. Instead, concepts from robust MPC are used where the optimization is done over a finite set of control behaviors. The method will be more thoroughly discussed in chapter 4.2.4.

2.6 COLREGs

For both manually operated and autonomous ships, the COLREGs define rules that must be followed to prevent accidents and ensure a safe voyage. The current COLREGs were published by the International Maritime Organization (IMO) in 1972 and came into effect a few years later in 1977 (U.S Department of Homeland Security, U.S. Coast Guard, 2014). The COLREGs include a total of 41 rules and are divided into 6 separate sections: A - General, B - Steering and Sailing, C - Lights and Shapes, D - Sound and E - Light, F - Exemptions, G - Verification of compliance with the provisions of the Convention (International Maritime Organization, 1972). For the case of collision avoidance, part B is of most interest. However, some requirements in part C and D are also relevant. Part C contains requirements on the use of lights to make the ship more visible and part D contains requirements on the use of sounds to alert nearby vessels. The rules in part B for Steering and Sailing define different collision scenarios and specify what maneuvers should be taken in order to prevent collision between ships (International Maritime Organization, 1972).

2.6.1 Presentation of relevant COLREGs

In this project, the relevant COLREGs can be found in part B - Steering and Sailing. In the following, the most important COLREGs for collision avoidance will be presented.

The material here is taken from the regulations issued by the International Maritime Organization (1972), but does not replicate the rules in their entirety. Only a summary of the most important parts will be given. For a complete description of all COLREGs, the reader is referred to the report by the International Maritime Organization (1972).

Rule 8 - Action to avoid collision

Any action taken to avoid collision shall follow the rules of Part B - Steering and Sailing. In addition, actions shall be taken in ample time and result in passing at a safe distance. If there is sufficient sea room, alteration of course angle alone may be the most effective action. Any alteration of course and/or speed to avoid collision should be readily apparent for other vessels. A succession of small alterations of course and/or speed should be avoided. If necessary to avoid collision, a vessel shall reduce her speed, stop or reverse.

Rule 13 - Overtaking

Any vessel overtaking any other vessel shall keep out of the way of the vessel being overtaken. A vessel shall be deemed to be overtaking when coming up to another vessel from a direction more than 22.5 degrees abaft her beam. This situation is shown in Figure 2.6.

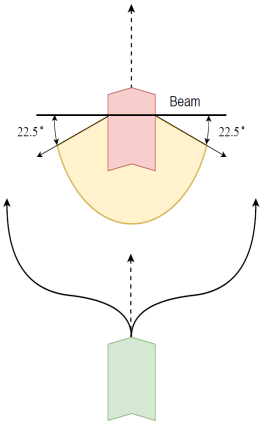


Figure 2.6: Overtaking situation. Rule 13 specifies that the overtaking vessel shall keep out of the way of the vessel being overtaken by passing on either side. Figure is taken from Henriksen (2018).

Rule 14 - Head-on situation

When two power-driven vessels are meeting on reciprocal (or nearly reciprocal) courses, each shall alter her course to starboard so that each pass on the port side of the other. This situation is shown in Figure 2.7.

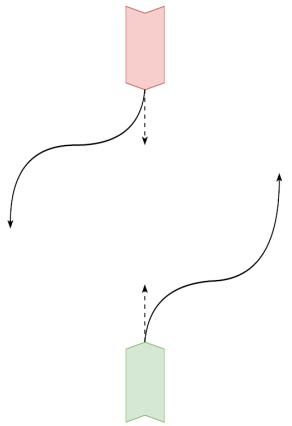


Figure 2.7: Head-on situation. The two vessels must alter their course to starboard according to Rule 14. Figure is taken from Henriksen (2018).

Rule 15 - Crossing situations

When two power-driven vessels are crossing, the vessel which has the other on her starboard side shall keep out of the way. If the circumstances allow, the vessel directed to keep out of the way should avoid crossing ahead of the other vessel. This situation is shown in Figure 2.8.

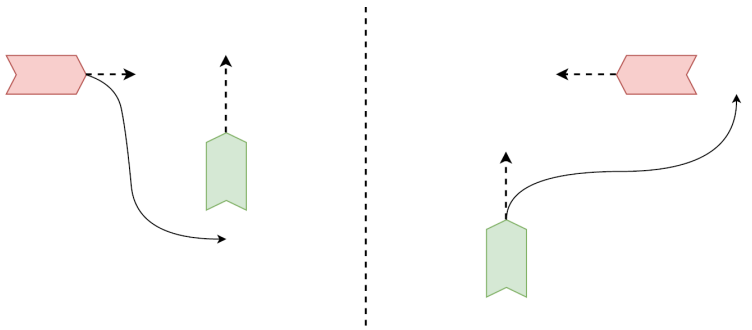


Figure 2.8: Crossing situation. The vessel which has the other on her starboard side shall keep out of the way according to Rule 15. Figure is taken from Henriksen (2018).

Rule 16 - Action by give-way vessel

The vessel which is directed to keep out of the way of other vessels is referred to as the give-way vessel. The give-way vessel shall take early and substantial actions to keep well clear.

Rule 17 - Action by stand-on vessel

The vessel which is directed to keep her course and speed is referred to as the stand-on vessel. The stand-on vessel can, however, take action to avoid collision if the give-way vessel does not take appropriate action to avoid collision. When the stand-on vessel is so close to the other vessel that collision cannot be avoided by the action of the give-way vessel alone, the stand-on vessel shall take action to try to avoid collision. In a crossing situation where the give-way vessel does not take appropriate action to avoid collision, the stand-on vessel should, if the circumstances admit, not alter her course to her own port side.

2.6.2 Challenges with COLREGs

There are several challenges with making a collision avoidance system comply with the COLREGs. The first challenge is that the COLREGs are written for human navigators and are open to different interpretations. This means that the rules can be difficult to implement in a collision avoidance system, due to imprecise formulations. As an example, rule 8 - action to avoid collision states that ”Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance”. What should be regarded as a ”safe distance” is not specified. A second difficulty is that the collision avoidance system needs a way to check which COLREGs rule applies in different situations, and take actions that comply with the relevant COLREGs. A third problem with COLREGs is that they are not designed for complex situations involving several vessels (Tam and Bucknall, 2013).

An example of a complex situation where applying COLREGs is difficult can be seen in Figure 2.9. In this situation, ship A and B are in a crossing situation. According to Rule 15, ship B is the stand-on vessel relative to ship A. Therefore, ship A should keep out of the way of B, but if ship A turns either left or right, a collision with ship C or D might occur.

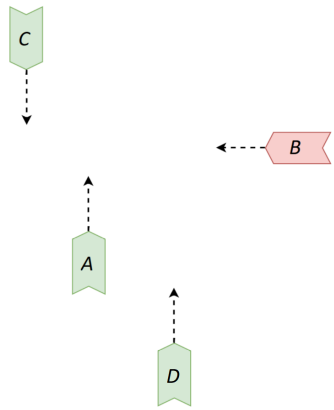


Figure 2.9: In this situation, ship B is required to be stand-on vessel relative to vessel A, and ship A should keep out of the way from B. However, the presence of ships C and D limits the maneuvering options for ship A.

Chapter 3

Manual operation of ships and reasons for collisions

This chapter will explain how ships are manually operated and what kind of technical equipment is used on board ships to help the navigators. Based on collisions investigated by the International Maritime Organization and interviews conducted with navigators from DNV GL, common reasons for collisions at sea will be presented.

3.1 Manual operation of ships

In order to investigate how a ship is manually operated, interviews have been conducted with two experienced navigators from DNV GL. Much of the material in the following sections was obtained during these interviews. The manual operation of a ship differs slightly in different parts of the voyage. When a ship travels from an origin port to a destination port, the voyage can be divided into three main phases.

The first phase begins when the ship is leaving the origin port. Crew members on deck inspect the ship, handle the lines and ropes and make sure that the ship is seaworthy. A pilot enters the ship and helps guide the ship out of the port and gives recommendations on where to steer the ship.

The second phase of the voyage is traveling on the open sea, and this phase starts when the pilot leaves the ship. Requirements for the crew composition depend on several factors such as the type of ship, traffic situation along the planned route and the operational procedures defined by the shipowner. The crew on the bridge is responsible for steering

the ship. As a minimum, there has to be one officer of the watch who is responsible for steering the ship. During daytime, when the visibility is good, there is usually only one person on the bridge. When visibility is bad due to weather conditions or night-time, there is also a lookout on the bridge in addition to the officer of the watch. The lookout is responsible for visual monitoring of other ships and obstacles. Also, the ship always has a captain on board the ship with navigational experience who is responsible for every aspect of the voyage and vessel. In difficult navigational situations, the officer of the watch can ask the captain for advice.

In the final phase, the ship is docking at the destination port. This phase is quite similar to the first phase. A pilot enters the ship to gives advice regarding where to steer the ship, and the crew on the deck prepares the ropes and lines.

3.1.1 Technical equipment on board ships

There is a wide range of technical equipment used on board the ship for navigational purposes. There are sensors that measure the ship's parameters such as speed, rudder angle, rate of turn for the rudder angle, engine RPM in addition to measurement of external factors such as wind speed. Global Position System (GPS) and radar are used in order to show the ship's location and the location of surrounding ships and obstacles. Automatic Identification System (AIS) is a tool that supplements the radar and is used to track other ships. Real-time information about position, course and speed for other ships is available through AIS, and the information is displayed on a monitor or an Electronic Chart Display and Information System (ECDIS). AIS also provides other types of information, such as destination port and ship type. Radar and AIS are important tools to gain an overview of the traffic situation.

The ship utilizes different kinds of equipment to communicate with surrounding ships. Different colored lights on different sides of the ship allow the ship to be more visible, and the use of lights is especially important at night. The ship is also equipped with a horn that can be used to alert nearby ships when visibility is poor. Another important way of communication is to perform drastic changes in heading angle to clearly show the direction the ship will be traveling. The ship is equipped with a Very High Frequency (VHF) radio which allows the captain to communicate directly with other ships. In collision situations, use of VHF allows the ships involved to cooperate on finding maneuvers that will avoid a collision. However, this should generally be avoided since improper use of VHF radio can also lead to misunderstandings. There have been collision situations where VHF radio was used to plan collision avoiding maneuvers, but communication happened with the wrong ship.

Figure 3.1: Technical equipment on the bridge of a ship. Figure is taken from https://www.captecamericas.com/marine-simulation-integration-ships-bridge-installation/.

3.1.2 Collision avoidance for manually operated ships

For manually operated ships, the COLREGs specifies what maneuvers that should be taken to avoid a collision. Some COLREGs rules that are relevant for collision avoidance are presented in chapter 2.6. Generally, the navigator uses both visual monitoring and technical equipment such as radar and AIS in order to detect obstacles. In a collision situation, the navigator should perform collision avoiding maneuvers as soon as possible and the maneuvers done should be visible to the surrounding ships. For instance, a change of heading angle of 5° is not visually noticeable by other ships. Instead, a larger change of 15°-30° should be performed.

Statheros (2008) presents several factors that influence how collision avoidance is performed for manually operated ships. The first factor is the type of the ship, which determines which evasive maneuvers that can be performed. Small speed boats have high maneuverability and can avoid obstacles with more ease than a 200-meter long oil tanker. Due to poor maneuverability of large ships, changes in speed and heading can be slow. Therefore, collision avoiding maneuvers have to be performed early. The second factor is the type of traffic, which can be categorized as either inside a confined environment (near ports or in canals) or out in the open sea. Being in a confined environment drastically limits the maneuverability of the ship. In addition, confined environments often have more dense traffic. The third factor is weather conditions, which influence the maneuverability and can make it difficult to detect obstacles in the case of fog. However, technical equipment such as radar and AIS helps the navigator to get a better overview of the traffic situation and makes it possible to detect obstacles.

3.2 Collision statistics

Even though modern navigation and communication equipment such as AIS, GPS and VHF radio is extensively used, maritime collisions still occur on a regular basis. Collisions at sea can potentially have disastrous consequences. Perhaps the worst collision incident occurred in 1987, where the overcrowded passenger ferry MS Doña Paz collided with the oil tanker MT Vector near Marinduque in the Philippines. This tragedy led to over 4000 fatalities (Mariano, 2017).

According to the European Maritime Safety Agency (2018), between 2011 and 2017 a total of 23264 ships were involved in marine accidents and a total of 683 human lives were lost. In 2017 alone, a total of 3647 ships were involved in maritime accidents (European Maritime Safety Agency, 2018). Among the accidents in 2017, there were a total of 1018 persons injured and 61 fatalities. The number of persons injured per year due to maritime accidents has been relatively constant at around 1000 persons per year since 2014, but the number of fatalities due to maritime accidents has been decreasing since 2015.

During the period 2011-2017, around half of all accidents with casualties were related to navigational problems such as contact, collisions and groundings. 23.2% of all accidents with casualties in this period were due to collisions. Figure 3.2 contains statistics on how many accidents happened in different areas of the sea in the period 2011-2017. The majority of accidents occur near ports and near the coast.

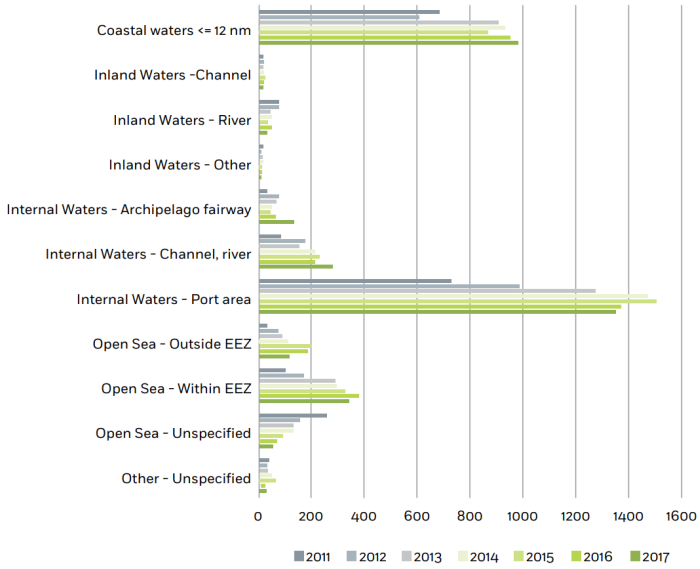


Figure 3.2: Statistics on how many accidents occured in different areas of the sea in the period 2011-2017. Figure is taken from European Maritime Safety Agency (2018).

A large part of the research within Maritime Safety is concerned with the analysis of historical data, not on measures that can be implemented to reduce the number of collisions. It is possible to compare collision statistics before and after the inclusion of a safety measure, but this is rarely done. However, using well-functioning technology has been shown to reduce the number of accidents in some cases. According to Vanem et al. (2008), the use of ECDIS caused a reduction of grounding frequency of between 19% and 38%.

3.3 Common reasons for collisions at sea

One of the most common reasons for accidents at sea is human errors. European Maritime Safety Agency (2018) performed investigations on 1645 accidents in the period 2011-2017 and found that 57.8% of these accidents were caused by human errors. In the shipping industry, Allianz Global Corporate & Specialty (2017) estimates that between 75% and 96% of all accidents are caused due to human errors.

A collision can be regarded as a specific type of maritime accident. Human errors are also a common cause for collisions at sea. An overview of which human errors that contribute to collisions can be seen in Figure 3.3 from Ziarati and Ziarati (2007). In this study, poor lookout, bad decision-making and poor manning levels were the most common human-related factors that led to collisions.

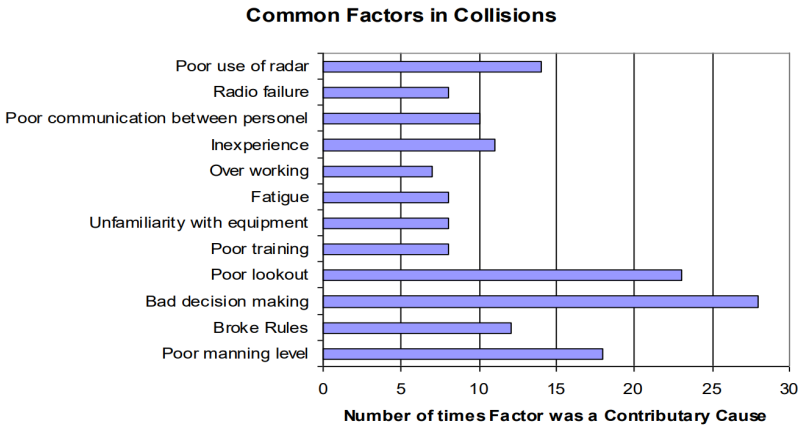


Figure 3.3: Overview of common factors that led to collisions. Figure is taken from Ziarati and Ziarati (2007).

IMO has published investigations of different maritime accidents on their website (International Maritime Organization website, 2017), including collisions and groundings. For

each accident investigated, IMO has given a description of what happened, why it happened and what can be learned to avoid that type of accident in the future. The material from these investigations, in addition to interviews conducted with navigators from DNV GL, has been used to find the causes for collisions at sea as described in the following.

Many collisions investigated by IMO happened due to undermanned ships. According to Lloyd (2006), the shipowners want to save as much money as possible and therefore prefer to have a small crew. This results in a big workload for the crew members, thus less attention is paid to the traffic and weather conditions. When not keeping a proper lookout, the crew might not be able to detect obstacles and therefore not be able to take actions early enough to avoid collisions. Due to undermanning, crew members might be preoccupied with other tasks than keeping lookout. There are examples of collisions happening due to the crew being too busy doing paperwork and thus not paying enough attention to surrounding ships.

The undermanning on ships can cause crew members to become fatigued due to big workloads and long working hours. According to Wöhrn (2007), the crew members often experience long work hours (nine hours or more). In some cases, collision happened due to crew members falling asleep on duty. Fatigue among crew members leads to less attention being paid to surrounding ships, and it can take longer before necessary collision avoiding actions are taken. Crew members are also more likely to misread warning signals and make poor decisions when they are fatigued (Lloyd, 2006). Lack of experience could also lead to bad decision making. For instance, navigators with less experience might be willing to take larger risks and keep a too small distance away from obstacles.

Some collisions have happened due to misunderstandings, where the officer of the watch falsely assumes that the other ship will take collision avoiding actions. These types of collisions could have been avoided if the ships were able to communicate effectively over VHF radio. However, language barriers can make it difficult for ships to agree on a solution. After investigations of the collision between the bulk carrier Huayang Endeavour and the oil tanker Seafrontier in the English Chanel in 2017 (Figure 3.4), it was shown that bad communication over radio and fatigue among the crew members were main reasons for the collision (Wingrove, 2018).

Weather conditions can play an important role. Some of the collisions studied by IMO happened due to poor weather conditions, making it hard to perform visual monitoring of surrounding ships. In addition to restricted visibility, poor weather conditions can make the ship difficult to steer. The effect of weather conditions is often underestimated, especially for ships with heavy cargo (Wöhrn, 2007).

Another common reason for collision is violation of COLREGs. Even though the navigator is required to follow the COLREGs, several of the collisions investigated by IMO happened due to ships making maneuvers that were in violation with the COLREGs. In some cases, collision happened due to turning the wrong direction compared to what the COLREGs required. In other cases, collision happened due to collision avoiding maneuvers were made too late.

Figure 3.4: The aftermath of the collision between Huayang Endeavour and Seafrontier in 2017. Figure is taken from https://www.energyvoice.com/oilandgas/169893/report-into-collision-between-oil-tanker-and-bulk-carrier-on-dover-strait/.

Failure of equipment is another factor that can lead to collisions. A well-functioning navigation system and propulsion system are necessary in order to discover potential obstacles and to be able to perform collision avoiding maneuvers. Several collisions have happened due to malfunctioning propulsion and steering systems, which led to loss of control of the ship. Knowing how to properly use and be familiar with the onboard equipment is also important.

Over-reliance on the electronic equipment can sometimes be a problem (Wöhrn, 2007). For example, not all ships are equipped with AIS (especially true for smaller ships, such as fishing vessels). Accidents have occurred in the past due to trusting the AIS too much. If the ship does not appear on the AIS, the navigator might believe that there are no other ships in close proximity, which can give the navigator a false sense of security. Equipment such as AIS and radar must be used in addition to visual monitoring of the traffic situation.

To summarize the findings, one of the major reasons for collision is that crew members do not pay enough attention to traffic conditions. Not having enough crew members on duty can lead to a big workload and less attention being paid. Fatigue among crew members is dangerous and can lead to collision avoiding maneuvers being taken too late. Other common reasons for collisions are violation of COLREGs and failure of equipment. The collisions investigated by IMO happened both in confined spaces, such as fairways, and out in the open sea. However, navigation in confined spaces is more difficult since it reduces the maneuvering possibilities, partly due to more dense traffic. Many collisions don't happen solely because of the ships directly involved in the collision situation. There might be static obstacles or other ships nearby not directly involved in the collision situation that limits the possibility for maneuvering.

Chapter 4

Collision avoidance

The previous chapter discussed manual operation of ships, how collision avoidance is performed for manually operated ships and highlighted some common reasons for collisions. This chapter will be concerned with algorithms for collision avoidance, in the setting of autonomous ships. In this chapter, the terms own-ship and obstacle ship will be used. Own-ship refers to the ship that has implemented some algorithm for collision avoidance, while obstacle ship is a ship to be avoided by the own-ship.

Section 4.1 provides a literature survey on collision avoidance and gives an explanation of the difference between long-term and short-term collision avoidance methods. Examples of several types of collision avoidance methods that achieve COLREGs compliance are presented, before exploring cooperative collision avoidance methods. Section 4.2 will go more in-depth into the workings of four different collision avoidance algorithms and investigates the possibility of using other vessels' intentions to improve their performance.

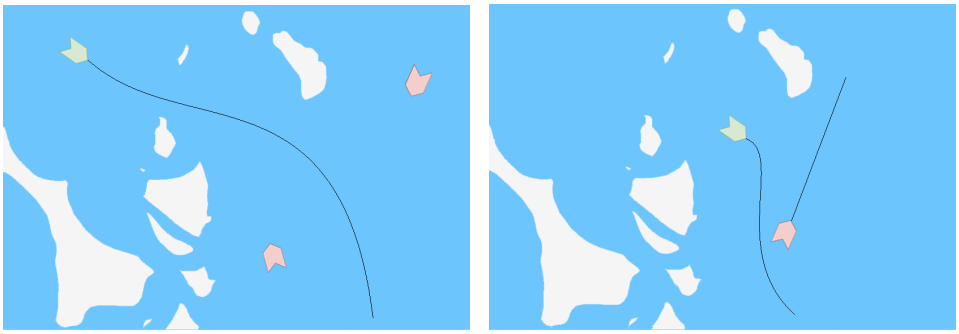
4.1 Literature survey on collision avoidance

A historical overview of several collision avoidance algorithms is given in the review article by Statheros (2008). In this paper, different approaches for collision avoidance are divided into two main categories: classical and soft computing based techniques. The classical methods are based on mathematical models of the ships' dynamics (as well as the environment), while soft computing techniques use artificial intelligence to solve the collision avoidance problem. Statheros (2008) gives an overview of several collision avoidance methods, belonging to both categories, and also mentions hybrid methods that use a combination of different techniques. An example of this is the collision avoidance system proposed by Hwang et al. (2001) which combines two AI technologies, fuzzy logic and an

expert system, to carry out collision avoidance.

The paper by Tam and Greig (2009) tries to expand the work done by Statheros (2008). This review article discusses collision avoidance methods with and without COLREGs compliance, in addition to path planning methods. In the algorithms studied by Tam and Greig (2009), there were three main limitations: no consideration of environmental factors, most dynamic ship models have unrealistic dynamics and the obstacles are not truly dynamic (which can be assumed to mean that the objects move with constant heading and speed).

Collision avoidance methods can be distinguished as either long-term or short-term algorithms, as in Bitar et al. (2019). The long-term algorithms perform long-term path planning and decide on a path from start to finish. Such algorithms typically rely on map data and will find a path that avoids static obstacles. Typically, long-term algorithms can run offline, prior to a mission. On the other hand, short-term algorithms allow a vessel to deviate from the long-term planned path when the vessel is on collision course with an obstacle. When no longer in danger of collision, the vessel can return to the planned path. These algorithms typically rely on real-time sensor data and make the vessel able to avoid dynamic obstacles. The difference between long-term and short-term collision avoidance is illustrated in Figure 4.1. According to Loe (2008), long-term path planning requires a lot of computations compared to short-term methods, which suggests that long-term algorithms are not well suited for rapidly changing environments.



(a) Long-term collision avoidance. **(b)** Short-term collision avoidance.

Figure 4.1: Difference between long-term and short-term collision avoidance. Long-term path planning determines a path from start to finish, while short-term collision avoidance makes the own-ship deviate from desired path to avoid obstacles.

It is also possible to combine both long-term and short-term collision avoidance methods into a hybrid approach, as done in Bitar et al. (2019) and Loe (2008). The hybrid approach suggested in Bitar et al. (2019) uses a three layer architecture consisting of high-level, mid-level and short-term collision avoidance. Bitar et al. (2019) implement the two top levels, where the high-level is a long-term planner based that solves an optimization problem to find an energy-optimized trajectory and the mid-level is an MPC-based long-term collision

avoidance algorithm. The hybrid approach in Loe (2008) combines a short-term Dynamic Window algorithm with a long-term, modified version of the Rapidly-Exploring Random Tree (RRT) method.

When it comes to specific methods that solve the collision avoidance problem and at the same time achieve COLREGs compliance, there are several papers worth mentioning. Chiang and Tapia (2018) developed the COLREG-RRT method, a long-term algorithm based on the RRT method to find long-term trajectories in compliance with COLREGs. Based on simulation studies with both single ship and multi-ship encounters, they achieve a high success rate. Lee et al. (2004) proposed the Modified Virtual Force Field (MVFF) method based on fuzzy logic to solve the collision avoidance problem. The method can perform both track-keeping (follow a desired path) and collision avoiding maneuvers. Kuwata et al. (2014) introduced Velocity Obstacles (VO), a short-term algorithm that calculates a set of velocities that must be avoided in order to prevent a collision. They used on-water tests with up to four different vessels to verify COLREGs compliance. The paper by Johansen et al. (2016) presents an MPC-based approach to collision avoidance called Simulation-Based Model Predictive Control (SBMPC). Simulation studies are used to show that the method is able to handle complex collision situations involving multiple dynamic obstacles. The methods presented by Chiang and Tapia (2018), Lee et al. (2004), Kuwata et al. (2014) and Johansen et al. (2016) will be further discussed in the next section.

An important aspect of any collision avoidance system is COLREGs compliance. If vessels can follow the guidelines set by the CORLEGs, many accidents due to human error could have been avoided (Campbell et al., 2012). There has not been an extensive amount of research regarding applying COLREGs evaluation to existing collision avoidance methods. According to Woerner (2016), many authors who have studied collision avoidance claims to achieve compliance with the COLREGs, without having any metric of verification or validation. Woerner (2016) created different metrics in order to evaluate collision avoidance algorithms. The master's thesis by Minne (2017) developed a framework for automatic testing of collision avoidance algorithms, using the metrics developed by Woerner (2016).

Potential collision situations where multiple ships are involved are common, especially near ports where the traffic is dense. It would be advantageous for a ship to communicate and coordinate her actions with other ships, which motivates the research regarding cooperative control of ships. However, according to Chen et al. (2018) vessels in operation in 2018 does not coordinate their actions. The advantage of such cooperative methods for collision avoidance is that ships are able to share their intentions. Cooperation between ships is not only advantageous for the purpose of collision avoidance. It can also help coordinate voyage plans to avoid congestion at ports and also make tasks like rescue missions more effective (Chen et al., 2018). In non-cooperative methods such as VO, the obstacle vessels' actions have to be predicted which is prone to errors.

In recent years, there has been some research regarding cooperative collision avoidance for vessels. Chen et al. (2018) propose an MPC-based method for cooperative control of multiple vessels. The main focus of the paper is to solve the Vessel Train Formation

problem, where several vessels travel along the same path and at the same time must avoid collisions. The ships broadcast their trajectory and a shared global variable is used to store trajectories that the vessels broadcast. Simulation results reveal that the proposed method is able to solve the Vessel Train Formation problem for up to 30 vessels. However, Chen et al. (2018) does not consider COLREGs compliance of the method.

Tam and Bucknall (2013) describe a cooperative path planning algorithm that determines a COLREGs compliant path to avoid collision for vessels on collision course. The only data that is shared among the vessels is position and heading, which means future intentions are not used (real-time data such as current position and heading is available through AIS). Simulation studies reveal that the proposed method can find collision avoidance maneuvers which complies with the COLREGs. However, avoidance of static obstacles, such as shorelines, was not considered.

Li et al. (2019) propose a coordination strategy for solving the collision avoidance problem in many-to-many collision situations, where a single ship must must consider several obstacle ships at the same time. The information that is exchanged between ships is intended course, in addition to position, speed and current course. However, compliance with COLREGs was not thoroughly investigated and the effect of reduced maneuverability due to static obstacles was not considered.

To summarize, there has been done some research on cooperative methods, but not all of them use data about future intentions. Research regarding methods that utilize future intentions of obstacle ships is very limited. The author has not been able to find a collision avoidance algorithm that utilize other vessels' intentions where COLREGs compliance is investigated and at the same time investigates the effect of static obstacles such as shorelines that reduce maneuverability for the ships involved. This motivates the work of this project to find out if other vessels' intentions can be used to improve already existing algorithms for collision avoidance.

4.2 Methods for collision avoidance

One of the tasks in this project is to assess the properties of different collision avoidance approaches and consider how other vessels' intentions could be used to improve performance. The previous section gave a general overview of previous work done regarding collision avoidance. This section will dive deeper into the workings of four different algorithms and also go beyond what is described in the papers and consider how other vessels' intentions could be utilized to improve performance of these collision avoidance methods. It is possible to use data about planned trajectory, deviations from planned trajectory, whether or not the obstacle ship will be a stand-on vessel or a give-way vessel and information about equipment malfunctions. It is assumed that such data is available. A more in-depth discussion of which intentions that that can be utilized is given in the next chapter. Even though ideas to improve performance will be given in the sections that follow, these ideas have not been tested in simulations.

There exist a vast amount of different methods for collision avoidance. Therefore, it is necessary to limit the scope of the study. Much of the work done on collision avoidance has been done for mobile robots and road vehicles. However, the methods that have been studied in this project are methods for collision avoidance for marine surface vessels that satisfies COLREGs. Both long-term and short-term methods will be considered. The objective with this study is to find out how these algorithms could be extended to include other vessels' intentions in order to hopefully improve the performance.

4.2.1 COLREG-RRT

The Rapidly-Exploring Random Tree (RRT) method was originally developed to make robots navigate among obstacles. It can be characterized as a long-term path planning method that plans a path while taking vehicle dynamics into account (Loe, 2008). Chiang and Tapia (2018) developed a method called COLREG-RRT, an RRT-based motion planner able to find long-term COLREGs compliant trajectories. A brief explanation of the RRT method is given in Chiang and Tapia (2018) and is summarized here.

In RRT, a tree consisting of nodes is sequentially built, where each node stores a state. A root node is created and stores the vessel's start state $\boldsymbol{x}_{start} = [\boldsymbol{\eta}_{start}, \boldsymbol{\nu}_{start}]^T$ where $\boldsymbol{\eta} = [x, y, \psi]^T$ and $\boldsymbol{\nu} = [u, v, r]^T$. As in chapter 2.2, x and y are the vessels' coordinates, ψ is the heading angle, u, v are velocities in x- and y-direction and r is the yaw rate. A loop with the following three steps is performed for a fixed number of iterations in order to build the tree:

1. A random state $\boldsymbol{x}_{rand}$ is selected. If $\boldsymbol{x}_{rand}$ is not in collision with any obstacles, the node in the tree with the state closest to $\boldsymbol{x}_{rand}$ is selected. This node is called $n_{nearest}$ and the state stored in $n_{nearest}$ is called $\boldsymbol{x}_{nearest}$.
2. A control action u is computed that will make the vessel move from $\boldsymbol{x}_{nearest}$ to $\boldsymbol{x}_{rand}$.
3. A forward simulation is performed. It is assumed that the vessel starts in $\boldsymbol{x}_{nearest}$ and applies the control action u for Δ_{tree} seconds. This will make the vessel move from $\boldsymbol{x}_{near}$ to a new state $\boldsymbol{x}_{new}$. If the resulting new state $\boldsymbol{x}_{new}$ is not in collision with any obstacles, a new node is created. This new node contains $\boldsymbol{x}_{new}$, control input u and a pointer to the parent node $n_{nearest}$ that contains the previous state $\boldsymbol{x}_{nearest}$.

If a node reaches the goal destination, a path can be found by traversing through the nodes in the tree since each node will contain a current state, the previous state and a control action u that makes the vessel move from the previous state to the current state. In order to speed up the tree growth, the probability of choosing the goal state as $\boldsymbol{x}_{rand}$ can be increased.

In Chiang and Tapia (2018), a joint forward simulation inspired by the method used by Xue et al. (2009) is performed during the tree growth to provide COLREGs compliant trajectories. The joint forward simulation simulates trajectories for the own-ship and the

obstacles. Consequently, both the own-ship's state and the obstacle's state need to be stored in each tree node in COLREG-RRT. The trajectories found from the joint forward simulation make it possible to anticipate collisions.

In order to find trajectories that avoid collision with obstacle ships, the first step is to determine if a collision is likely to occur when assuming that obstacle ships and the own-ship maintain their current velocity. If a collision is likely to occur, the type of collision situation is identified. If the collision situation requires the own-ship to take an action in order to avoid collision, a new trajectory that avoids collision is computed. By assuming that all obstacle ships travel with a constant velocity, the own-ship can compute the desired course angle necessary to make the ship follow the new trajectory.

Chiang and Tapia (2018) use the principle of virtual obstacles in order to find the COLREGs compliant trajectories that avoid collision. When an obstacle ship is within a distance d_{colreg} of the own-ship, a virtual obstacle is created according to the collision encounter type. This is illustrated in Figure 4.2. The virtual obstacle is an "extension" of the ship in some direction and the desired trajectory for the own-ship is constructed such that the own-ship will not collide with the virtual obstacle. The inclusion of virtual objects will make the own-ship comply with COLREGs.

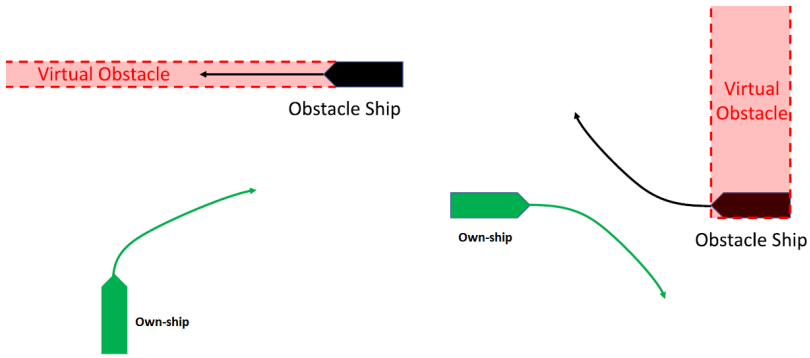


Figure 4.2: Principle of virtual objects for crossing and head-on situations. The virtual objects will force the own-ship to follow a trajectory that satisfy COLREGs. Figure is taken from Chiang and Tapia (2018).

The COLREG-RTT algorithm by Chiang and Tapia (2018) is divided into three phases. In the first phase, a tree is grown to check if there exists a direct, collision-free trajectory from the current own-ship position to the destination. If such a trajectory cannot be found, the second phase starts and tries to grow an RRT tree by using joint forward simulation and use of virtual obstacles in order to maintain COLREGs compliance during the tree growth. If no trajectory is found, the third phase performs a tree growth without considering virtual obstacles. A pseudocode of the complete algorithm is given in Chiang and Tapia (2018). These three phases make sure that a COLREGs compliant trajectory is found when possible. Even when a COLREGs compliant trajectory is not found the ship is still able to move

forward towards the destination. A trajectory is returned from one of the three phases and a heading autopilot can be used to follow this trajectory.

Utilizing other vessels' intentions to improve performance of COLREG-RRT

In the paper by Chiang and Tapia (2018), they do not consider utilizing other vessels' intentions to improve performance. This section will contain some ideas on how intentions could be included in the COLREG-RRT algorithm, assuming that information about an obstacle's planned trajectory is available. In COLREG-RRT, a joint forward simulation is performed in order to compute a trajectory for the own-ship that achieves COLREGs compliance. When computing this trajectory, it is assumed that the obstacle will continue to travel along a straight path. If it turns out that the obstacle does not travel along a straight path, using the obstacle vessel's intended planned trajectory can probably help improve the generation of a new desired trajectory for the own-ship that avoids collision. This could be implemented in the algorithm by modifying how the virtual objects are generated.

In the COLREG-RRT method, a virtual obstacle is created according to the collision encounter type, as shown in Figure 4.2. A possible modification is to update this virtual obstacle to be along the planned trajectory of the obstacle ship. This is illustrated in Figure 4.3. The red region illustrates the original virtual obstacle and the blue region illustrates the new virtual obstacle that follows the obstacle ship's planned trajectory.

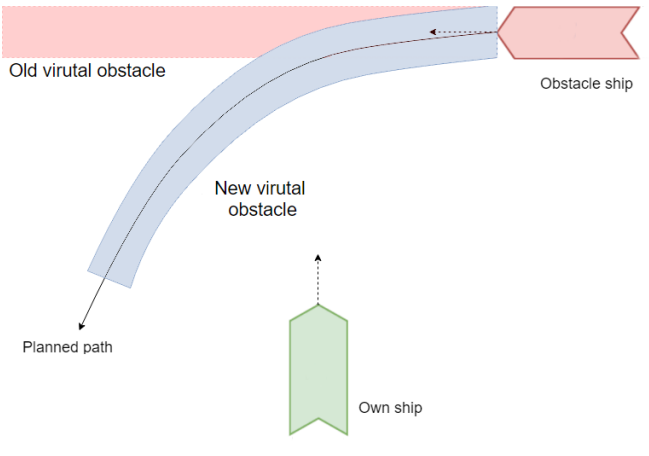


Figure 4.3: Illustration of the new virtual obstacle.

An example can be used to illustrate how intentions might help improve the performance of the COLREG-RRT algorithm. Consider the situation shown in Figure 4.4. This is a crossing situation between the own-ship A and the obstacle ship B. According to the current known position and velocity of the obstacle ship, the own-ship believes that the obstacle will continue to travel in the same direction along a straight path. According to COLREGs rule 15 for crossing situations, the own-ship must keep out of the way of

the obstacle ship by turning starboard to avoid crossing ahead of the obstacle ship. This leads to the path for the own-ship as shown in Figure 4.4. However, the assumption that the obstacle ship will travel along a straight path is wrong. The own-ship is unaware of the static obstacle that the obstacle ship B is facing. Due to this static obstacle, the obstacle ship decides to turn to her port side to avoid collision. As seen in Figure 4.4 this will lead to a potential collision between the own-ship and the obstacle ship. If the own-ship had information about the future trajectory of the obstacle ship such that the virtual obstacle was updated to follow the planned obstacle trajectory, then the own-ship could have followed a safer trajectory.

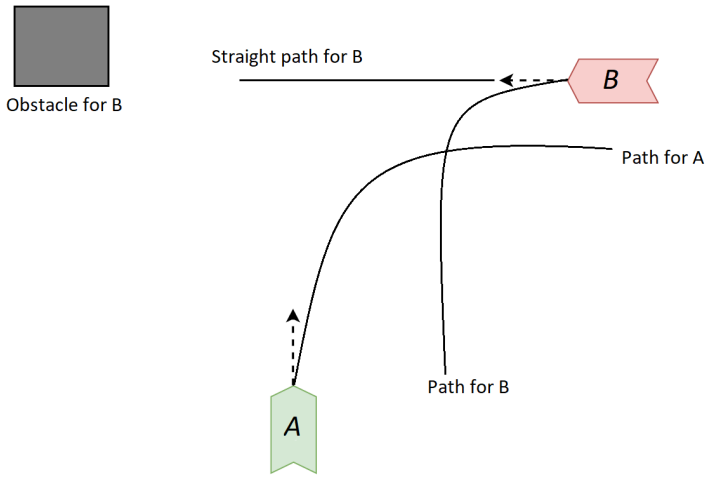


Figure 4.4: Collision situation that illustrates how other vessels' intentions can potentially improve the performance of COLREG-RRT.

4.2.2 Modified Virtual Force Field

The Virtual Force Field (VFF) method has been successfully used in mobile robotics. The main idea of VFF is to let the vehicle be influenced by two forces: one force that pulls the vehicle towards the next waypoint and another force that pushes the vehicle away from the obstacle. This idea is illustrated in Figure 4.5.

This method does not work very well for moving obstacles. Lee et al. (2004) have proposed a modified version of VFF, called Modified Virtual Force Field (MVFF), that is able to handle moving obstacles. The MVFF method is designed for surface vessels and it provides both a track-keeping mode and a collision avoidance mode that satisfies COLREGs. The desired heading angle for the vessel is given by $\psi_d = \psi_{tk} + \psi_{ca}$ where ψ_{tk} is the desired heading angle for track-keeping mode and ψ_{ca} is the desired heading angle for collision avoidance mode.

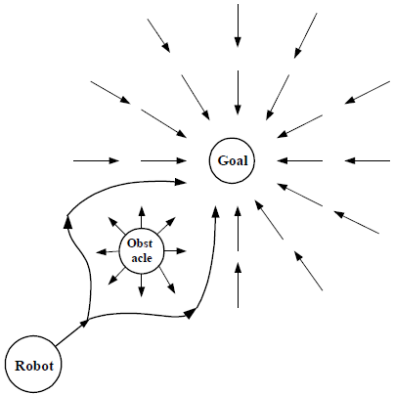


Figure 4.5: The main idea of VFF. The obstacle provides a pushing force and the goal destination provides a pulling force. These forces are used to find the path for the robot. Figure is taken from Lee et al. (2004).

Track-keeping mode

The track-keeping mode calculates the desired track-keeping heading angle, ψ_{tk}, in order to make the surface vessel follow the desired trajectory. To calculate ψ_{tk}, two force vectors $\boldsymbol{F}_a$ and $\boldsymbol{F}_p$ are used. These forces are defined in equations (4.1a) and (4.1b)

$$\boldsymbol{F}_a = \alpha \boldsymbol{e}_a \tag{4.1a}$$

$$\boldsymbol{F}_p = \beta \boldsymbol{e}_p \tag{4.1b}$$

where $\boldsymbol{e}_a$ is the unit vector from the own-ship to the next waypoint and $\boldsymbol{e}_p$ is the unit vector that points towards the desired path. These vectors can be seen in Figure 4.6.

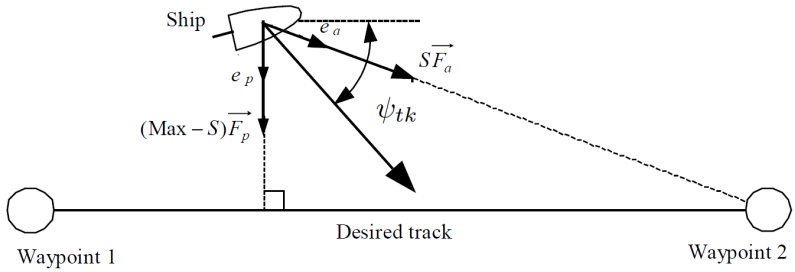


Figure 4.6: Track-keeping for MVFF. $\boldsymbol{F}_a$ points towards the next waypoint and $\boldsymbol{F}_p$ points towards the desired track. Figure is taken from Lee et al. (2004).

β and α are parameters determined by the fuzzy rules R_1 - R_3 in (4.2)

$$\begin{aligned} R_1 &: \text{IF } d \text{ is } Near \text{ THEN } \alpha \text{ is } B \text{ and } \beta \text{ is } S, \\ R_2 &: \text{IF } d \text{ is } Middle \text{ THEN } \alpha \text{ is } M \text{ and } \beta \text{ is } M, \\ R_3 &: \text{IF } d \text{ is } Far \text{ THEN } \alpha \text{ is } S \text{ and } \beta \text{ is } B \end{aligned} \tag{4.2}$$

d is a linguistic variable whose set of possible values are letters or words. The variable d describes the shortest distance between the vessel and the desired path and has the possible values {*Near, Middle, Far*}. The possible values for α and β are $\{S, M, B\}$ where S, M and B are abbreviations for Small, Medium and Big respectively. The values of α and β determine how the vessel will behave. A large α and a small β make the craft move quickly towards the next waypoint. On the other hand, a large β and a small α make the craft move quickly towards the desired path.

The final track-keeping heading angle is found by equation (4.3)

$$\psi_{tk} = angle(S\boldsymbol{F}_a + (Max - S)\boldsymbol{F}_p) \tag{4.3}$$

where Max is a constant that depends on the specific vessel and the mode number S is a number in the range $[0, Max]$.

Collision avoidance mode

In order to find the heading angle for the collision avoidance mode, ψ_{ca}, several fuzzy rules are used. An example of a single fuzzy rule is given below.

$$\begin{gathered} \text{IF } d_{obs} \text{ is } Near \text{ and } v_{rel} \text{ is } S \\ \text{and } \theta \text{ is } LF \text{ and } \phi \text{ is } LS \\ \text{THEN } \psi_{ca} \text{is} 22.5^\circ \end{gathered}$$

d_{obs} is the distance between the own-ship and the obstacle, with possible values {*Near, Far* }. v_{rel} is the absolute value of the relative velocity vector between the own-ship and the obstacle, with possible values $\{S, M, B\}$. θ is the location of the obstacle relative to the ship, found by the angle between the own-ship's heading and the obstacle ship (see Figure 4.7). The possible values for θ are $\{LB, LS, LF, F, RF, RS, RB\}$ where L and R are abbreviations for left and right. B, S and F are abbreviations for Back, Side and Front respectively. ϕ is the moving direction for the obstacle with respect to a local coordinate frame fixed to the obstacle. The possible values of ϕ are $\{LB, LS, F, RS, RB\}$. An illustration of the parameters θ and ϕ is shown in Figure 4.7. The variable ψ_{ca} can have values $\{0, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ\}$.

In the example fuzzy rule given above, the obstacle ship is $Near$ the own-ship and the obstacle ship is located "Left Front" of the own-ship. The obstacle ship is approaching the own-ship from the "Left Side". In this situation, the own-ship is commanded to turn right at an angle of 22.5°. The example fuzzy rule above can be generalized to the fuzzy rule

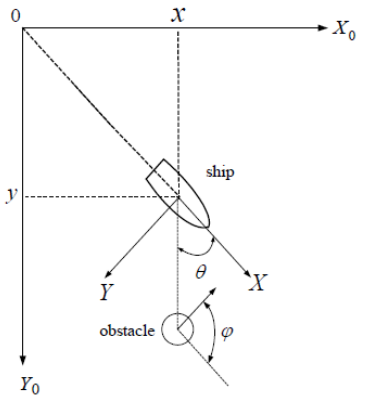


Figure 4.7: Illustration of the parameters θ and ϕ used in the fuzzy rule 4.4. Figure is taken from Lee et al. (2004).

formulation given in (4.4).

$$\begin{aligned} &\text{IF } d_{obs} \text{ is } (LVd_{obs})_i \text{ and } v_{rel} \text{ is } (LVv_{rel})_i \\ &\quad \text{and } \theta \text{ is } (LV\theta)_i \text{ and } \phi \text{ is } (LV\phi)_i \\ &\quad \text{THEN } \psi_{ca} \text{ is } angle \end{aligned} \tag{4.4}$$

i is the index to denote the fuzzy rule number and $LV*$ represents a linguistic value for the linguistic variable $*$.

This MVFF method is not too difficult to implement, due to few tunable parameters. A shortcoming of the work done by Lee et al. (2004) is to only consider the collision avoiding maneuver of turning right at an angle of $22.5°, 45°, 67.5°$ or $90°$. With these limited options for movement, a total of 210 fuzzy rules were created. Implementing more maneuver options (for instance the ability to turn left and change the speed) will require more fuzzy rules and therefore more computational power. Another shortcoming is that the simulations performed to test the performance of the algorithm only considered a single obstacle. According to Kuwata et al. (2014), the MVFF method does not scale well to multiple obstacle ships.

Utilizing other vessels' intentions to improve performance of MVFF

Possible ways to include other vessels' intentions to the algorithm developed by Lee et al. (2004) will now be presented. The desired angle for the collision avoidance mode of the MVFF method is found by the fuzzy rule formulation in (4.4). It would be possible to extend this rule to include the obstacle vessel's intended trajectory. For instance, a new variable P could be included in the fuzzy rule to denote where the intended trajectory of the obstacle ship will be relative to the own-ship. Unlike in the paper, it is assumed that the own-ship is able to turn left.

The behaviour of the own-ship with the new fuzzy rule can be illustrated with an example. Assume that the planned trajectory for the obstacle ship is known to the own-ship. Imagine that we define a half circle around the own-ship A with radius r as shown in Figure 4.8. This half circle is divided into four regions: $Left$, $FrontLeft$, $FrontRight$ and $Right$ (abbreviated as L, FL, FR and R in the figure). Assume an obstacle ship B's planned trajectory lies within two of these four regions, as shown in Figure 4.8. The red straight line represents the obstacle ship's planned trajectory. The idea is that the own-ship should not make a turn such that it's bow points into a region that contains the obstacle's planned trajectory. In the situation in Figure 4.8, the own-ship A cannot turn such that it's velocity vector $\boldsymbol{\nu}_A$ points into the $Left$ or $FrontLeft$ region. The only allowed direction should be to move in the $FrontRight$ or $Right$ direction.

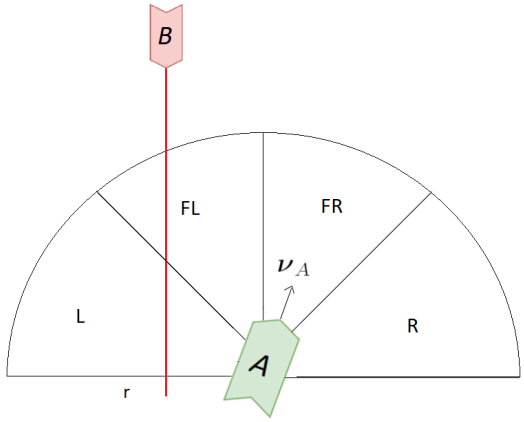


Figure 4.8: Determination of possible movements utilizing other vessels' intentions. The obstacle's planned trajectory lies within the regions FL and L, thus the own-ship A should only be allowed to move into the regions FR and R. In this case, the velocity vector $\boldsymbol{\nu}_A$ indicates that the own-ship is currently moving into the region FR.

As mentioned above, this behavior can be implemented by modifying the fuzzy rule in (4.4) by adding another variable P. This variable's value depends on which of the four regions L, FL, FR and R the obstacle's planned trajectory lies within. For instance, if the obstacle's planned trajectory lies within the regions L, FL and FR, then $P =$ "$L/FL/FR$". If the obstacle's planned trajectory lies within region R, then $P =$ "L". The own-ship can only move into a region that is not stored in P. The new fuzzy rule is given in (4.5).

$$\begin{gathered}\text{IF } d_{obs} \text{ is } (LVd_{obs})_i \text{ and } v_{rel} \text{ is } (LVv_{rel})_i \\ \text{and } \theta \text{ is } (LV\theta)_i \text{ and } \phi \text{ is } (LV\phi)_i \\ \text{and } P \text{ is } (LVP)_i \\ \text{THEN } \psi_{ca} \text{ is } angle\end{gathered} \tag{4.5}$$

The variable P is used to restrict the values the variable $angle$ can take. For instance, if $P =$ "$L/FL/FR$" then $angle$ can only be chosen such that the own-ship's velocity vector points into the region R.

4.2.3 Velocity Obstacles

The Velocity Obstacles (VO) method was first introduced by Fiorini and Shiller (1998) for robot motion planning. Since then, several modifications of the algorithm have been published. The version presented here is the VO algorithm proposed by Kuwata et al. (2014), which extends the main idea of VO to the context of maritime navigation and also considers COLREGs compliance.

The idea behind the algorithm is to generate a cone-shaped obstacle in the velocity space called VO. As long as the own-ship's velocity vector is outside the set of velocities defined by the VO, there will not be any future collisions (assuming that the velocity vectors does not change). The VO algorithm is fast, therefore it is well suited for high-speed operations.

Let $\boldsymbol{p} \in \mathbb{R}^2$ be the ship's position and let $\boldsymbol{\nu} \in \mathbb{R}^2$ be the ship's velocity vector. A ray starting from $\boldsymbol{p}$ going in direction of $\boldsymbol{\nu}$ is defined as in equation (4.6).

$$\lambda(\boldsymbol{p}, \boldsymbol{\nu}) = \{\boldsymbol{p} + t\boldsymbol{\nu} \mid t \geq 0\} \tag{4.6}$$

Let the geometrical shape of the own-ship be denoted by the set $\mathcal{A}$ and let the geometric shape of the obstacle ship be denoted by $\mathcal{B}$. The VO of obstacle ship B relative to the own-ship A is defined as in equation (4.7)

$$VO_A^B(\boldsymbol{\nu}_B) = \{\boldsymbol{\nu}_A \mid \lambda(\boldsymbol{p}_A, \boldsymbol{\nu}_A - \boldsymbol{\nu}_B) \cap (\mathcal{B} \oplus -\mathcal{A}) \neq \emptyset\} \tag{4.7}$$

where $\boldsymbol{\nu}_A$ and $\boldsymbol{\nu}_B$ are the velocity vectors of the own-ship and the obstacle ship respectively. $\boldsymbol{p}_A$ and $\boldsymbol{p}_B$ are the position vectors of the own-ship and the obstacle ship respectively. The set operation $\mathcal{A} \oplus -\mathcal{B} = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B}\}$ is called the Minkowski sum and $-\mathcal{A} = \{-\boldsymbol{a} \mid \boldsymbol{a} \in \mathcal{A}\}$ is the reflection of the set $\mathcal{A}$.

An interpretation of equation (4.7) is that the cone-shaped VO is the region where the rays originating from ship A going in the direction of $\boldsymbol{\nu}_A - \boldsymbol{\nu}_B$ intersects the geometric region $\mathcal{B} \oplus -\mathcal{A}$, as illustrated in the right part of Figure 4.9. The region $\mathcal{B} \oplus -\mathcal{A}$ is the geometric shape of the obstacle B expanded by the geometric region of the own-ship A. This "expanded B-obstacle" can be seen in the right part of Figure 4.9. As stated earlier, as long as the own-ship's velocity vector lies outside the VO, it will not collide with the obstacle ship.

The collision situation can be divided into three regions, as illustrated in Figure 4.10. If the own-ship has a velocity vector in region $\mathcal{V}_1$, the own-ship avoids the obstacle while seeing it on the right. In region $\mathcal{V}_2$, the own-ship avoids the obstacle while seeing it on the left. In region $\mathcal{V}_3$, the own-ship moves away from the obstacle. The method presented by Kuwata et al. (2014) uses information about these regions to make the algorithm find maneuvers that satisfy the requirements specified by the COLREGs in head-on, crossing and overtaking situations.

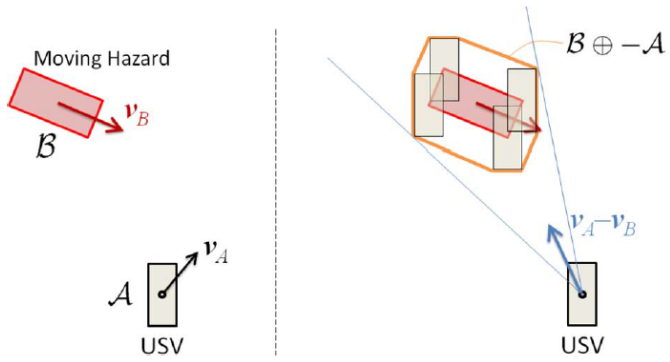


Figure 4.9: Graphical view of the cone-shaped VO. $\mathcal{A}$ is the geometric shape of the own-ship and $\mathcal{B}$ is the geometric shape of the obstacle ship. In this case, the Unmanned Surface Vehicle (USV) is denoted as the own-ship. Figure is taken from Kuwata et al. (2014).

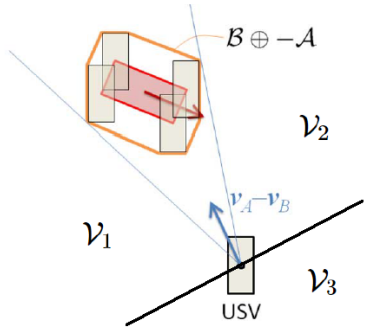


Figure 4.10: The situation can be divided into three different regions: $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$. These regions can be used to determine on which side the own-ship will pass the obstacle ship. Figure is taken from Kuwata et al. (2014).

The VO algorithm

The first step in the algorithm presented by Kuwata et al. (2014) is to use a pre-collision check to determine if a situation is likely to lead to a collision in the near future. Closest Point of Approach (CPA) can be defined as the position at which two moving objects reach their closest possible distance. Both time to CPA, t_{CPA}, and distance at CPA, d_{CPA}, is computed. For each moving ship, COLREGs apply only if the situation is likely to lead to a collision situation in the near future. In other words, COLREGs will apply only if the pre-collision check in equation (4.8) is true.

$$0 \leq t_{CPA} \leq t_{max} \wedge d_{CPA} \leq d_{min} \tag{4.8}$$

The next step is to find out if the own-ship is facing a head-on, crossing or overtaking situation as defined by the COLREGs. Kuwata et al. (2014) do not go into many details regarding how to identify each individual collision situation. However, Stenersen (2015) presents an approach to VO where the relative bearing angle is used to identify which collision situation applies. Relative bearing β is the clockwise angle from the heading of the own-ship to the line-of-sight vector from the own-ship to the obstacle. Figure 4.11 illustrates how to find the bearing angle β for own-ship A. In order to determine which collision situation that applies, the following rules can be used:

1. Head-on if $\beta \in [-15°, 15°]$
2. Crossing from right if $\beta \in [15°, 112.5°]$
3. Crossing from left if $\beta \in [-112.5°, -15°]$
4. Overtaking if $\beta \in [112.5°, 180°) \cup [-180°, -112.5°)$

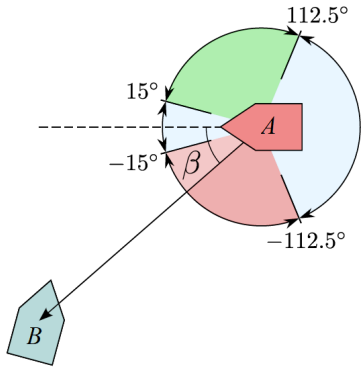


Figure 4.11: Bearing angles for own-ship A can be used to determine the type of COLREGs situation that applies. Figure is taken from Stenersen (2015).

The last step in the algorithm is to determine the best velocity vector for the own-ship. This is done by searching the velocity space grid. Each cell (v_i, ψ_j) in this grid represents a speed v and a heading angle ψ. The VO and COLREGs put constraints on which speeds and headings that can be chosen. The set of speeds and headings that cannot be chosen, due to VO and COLREGs violation, is called the constraint set. A cost function is used to calculate a cost for each cell outside the constraint set. The cost function proposed by Kuwata et al. (2014) is given by equation (4.9)

$$J_{ij} = \frac{w_\tau}{\tau_{ij}} + w_v \left\| \boldsymbol{\nu}_{ref} - \begin{bmatrix} v_i \cos \psi_j \\ v_i \sin \psi_j \end{bmatrix} \right\|_Q \tag{4.9}$$

where w_τ is the weight for time to collision, τ_{ij} is time to collision when own-ship has speed v_i and heading angle ψ_j, w_v is a weight on deviation from the desired velocity, $\boldsymbol{\nu}_{ref}$ is the desired velocity and $\|\cdot\|_Q$ is a weighted two-norm with a 2-by-2 weight matrix Q. When all costs are computed, the (v_i, ψ_j) pair with the lowest cost is chosen for the own-ship to follow.

Utilizing other vessels' intentions to improve performance of VO

The algorithm proposed by Kuwata et al. (2014) could be extended to include other vessels' intentions. In the VO algorithm, when trying to find a (v, ψ) pair, the constraint set contains the set of velocities and headings that cannot be chosen due to the VO and COLREGs violation. Utilizing other vessels' future planned trajectory could be used to extend the constraint set. A possible way to implement this is to create a virtual obstacle that overlaps the planned trajectory of an obstacle ship. Any (v, ψ) pair that will make the own-ship collide with the virtual obstacle will be added to the constraint set.

An example illustrates how this can be done. Figure 4.12 shows an example of an overtaking situation where the own-ship A wants to overtake the obstacle ship B. Due to an island on the left-hand side, the obstacle ship has planned to turn right as indicated by the planned path for B in Figure 4.12. The blue region defined by two curves at a distance r from the planned path for B defines a virtual obstacle. The (v, ψ) pairs that will make the own-ship collide with this virtual obstacle will be added to the constraint set. The region defined by angle ψ_r in Figure 4.12 represents the all heading angles that put the own-ship on collision course with the obstacle ship.

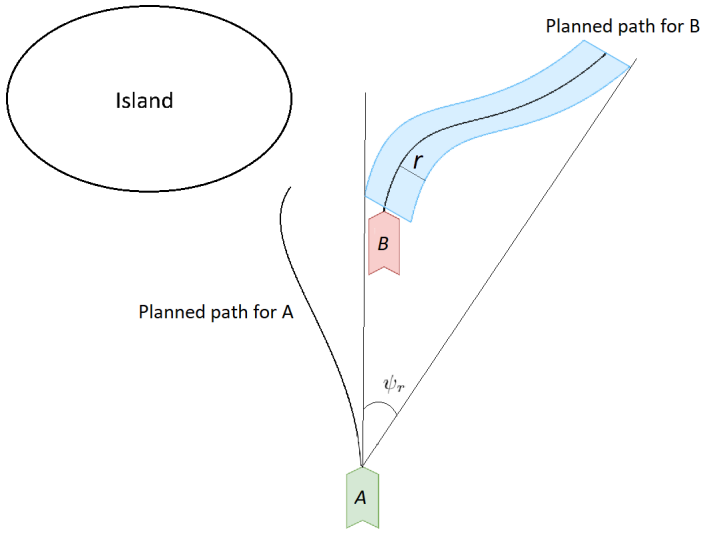


Figure 4.12: Overtaking situation where the planned path for obstacle ship B is used to create a virtual obstacle, defined by the blue region. a (v, ψ) pair that makes the own-ship A travel on collision course with the virtual obstacle is added to the constraint set.

A drawback with this implementation is that the calculation to check if a collision will happen between the virtual obstacle and the own-ship can be difficult to perform. An alternative implementation method would be to modify the cost function given in equation

(4.9) such that heading angles in the set ψ_r, that makes the own-ship be directed towards the planned path for the obstacle ship, will have increased cost.

There are other ways intentions could be used to improve the performance of the algorithm. Assume that the obstacle ship tells the own-ship which role it will take in a potential collision situation. The obstacle ship will either continue along her path and take the role of a stand-on vessel or will alter her course and take the role of a give-way vessel. This information can be used to make changes to the cost function in equation (4.9). If the obstacle ship takes the role of a give-way vessel, the own-ship will take the role as the stand-on vessel. Therefore, (v, ψ) pairs that will make the own-ship continue to travel along her planned trajectory can have an increased cost. On the other hand, if the obstacle ship takes the role as a stand-on vessel, (v, ψ) pairs that will make the own-ship alter her course will have a decreased cost.

4.2.4 Simulation-Based Model Predictive Control

A different approach to solve the collision avoidance problem is to use Model Predictive Control (MPC). There has been quite a lot of research regarding the use of MPC for collision avoidance purposes in a number of different areas. Liu et al. (2015) created an MPC-based obstacle avoidance method for ground vehicles, Keller et al. (2015) published an MPC-based approach for collision avoidance for cars in critical traffic situations and Bousson (2008) presents an MPC-based algorithm to prevent collision between aircrafts.

In the case of marine crafts, one of the first uses of MPC for collision avoidance with COLREGs compliance was presented by Johansen et al. (2016). According to Chiang and Tapia (2018), the method by Johansen et al. (2016) is one of the current state-of-the-art methods for collision avoidance. The use of MPC for collision avoidance is motivated by the fact that many existing methods for collision avoidance, as presented in Statheros (2008) and Tam and Greig (2009), do not scale well to handle several dynamic obstacles and at the same time can take into account ship dynamics and environmental disturbances (Johansen et al., 2016). According to Chen et al. (2018), there are many advantages of using MPC for maritime collision avoidance. Because maneuverability in a collision situation is restricted, it is an advantage that the MPC is able to predict trajectories into the future, which makes it possible to detect possible collisions early.

Johansen et al. (2016) present a collision avoidance method based on MPC called Simulation-Based Model Predictive Control (SBMPC) where the collision avoidance functionality is separated from the mission planning. The collision avoidance method determines a control behavior, which consists of a course offset and a speed command, that is sent to the autopilot. The best control behavior is found by evaluating different scenarios and choosing the control behavior associated with the lowest hazard. In the following, this method will be presented in more detail.

System overview

The collision avoidance method presented by Johansen et al. (2016) uses a modular approach, where the collision avoidance system is separated from the mission planning. An overview of the system is given in Figure 4.13. The mission planning provides the nominal path for the ship to follow as a set of waypoints, in addition to a nominal speed command. The collision avoidance module uses sensor data, predicted obstacle trajectories and the own-ship's nominal path to compute a collision-free trajectory close to the nominal path. The output of the collision avoidance system is a propulsion command and a course angle offset.

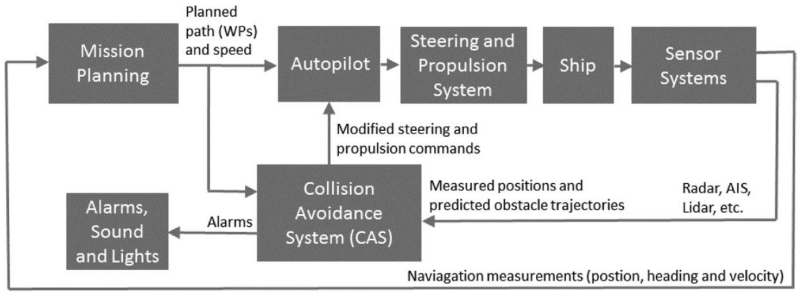


Figure 4.13: Block diagram of the system. The collision avoidance system is implemented as a separate module. Figure is taken from Johansen et al. (2016).

Control behavior and scenarios

The collision avoidance system must decide on a control behavior for the own-ship, which consists of a course angle offset and a propulsion command for the autopilot. The choice of control behavior is made by simulating different scenarios. Each scenario consists of a single control behavior for the own-ship, the own-ship's simulated trajectory when following this control behavior and a predicted trajectory for each obstacle ship. The simulation time in the scenario is called the prediction horizon. In order to choose which control behavior to apply to the own-ship, the concept of hazard minimization is used. The total hazard is computed for each scenario and is a function of COLREGs compliance, collision hazard and grounding hazard. The control behavior in the scenario that minimizes the total hazard is chosen and applied to the own-ship. Johansen et al. (2016) recommend a minimum set of possible control behaviors as

- Course offset: -90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75 or 90 degrees.
- Propulsion command: keep nominal speed, slow forward, stop or full reverse.

The propulsion command for the ship is $P \in [-1, 1]$ where $P = -1$ is reverse, $P = 0$ is stop and $P = 1$ is nominal forward speed. This minimum set of control behaviors gives a total of $13 \cdot 4 = 52$ different control behaviors. It would be advantageous to have

even more control behaviors to choose from and perhaps even be able to change control behaviors during the prediction horizon. However, there is a trade-off between the number of possible control behaviors and computational complexity.

Prediction of obstacle and own-ship trajectories

Johansen et al. (2016) use a simple method for predicting the future trajectory for each obstacle. It is assumed that the future obstacle trajectories are straight lines, which means that the obstacle ship's future position can be calculated from the simple relation in equation (4.10)

$$\boldsymbol{p}(t_0 + T) = \hat{\boldsymbol{p}}(t_0) + \hat{\boldsymbol{v}}(t_0) \cdot (T - t_0) \tag{4.10}$$

where t_0 is the current point in time, $\boldsymbol{p}(t_0 + T)$ is the position at time $t_0 + T$, $\hat{\boldsymbol{p}}(t_0)$ is an estimate of the position at time t_0 and $\hat{\boldsymbol{v}}(t_0)$ is an estimate of the obstacle's velocity at time t_0.

In order to estimate the future trajectory of the own-ship, a 3 DOF model similar to the one described in chapter 2.2 is used to simulate the behavior of the ship. An autopilot based on LOS guidance is used to steer the ship. This guidance method finds the necessary course angle χ_{LOS} to make the ship follow the nominal path. In order to avoid collisions along the nominal path, the collision avoidance system gives a course angle offset χ_{ca} to the autopilot. Therefore, the reference course command for the autopilot is $\chi_c = \chi_{LOS} + \chi_{ca}$. This causes the own-ship to deviate from the nominal path when there is a risk of collision.

COLREGs compliance

To be able to choose a control behavior that satisfies COLREGs, the type of collision situation has to be determined. In Figure 4.14, a collision situation is shown. The blue and red dots represent the predicted position of the own-ship and the obstacle ship respectively in scenario k. The blue and red curves in front of blue and red dots are the predicted trajectories of the own-ship and the obstacle ship respectively. The black vector $\boldsymbol{L}_i^k(t)$ is a unit vector pointing in the LOS direction from the own ship to the obstacle ship i in scenario k. The blue vector $\boldsymbol{v}_0^k(t)$ and the red vector $\boldsymbol{v}_i(t)$ represent the predicted velocities of the own-ship and the obstacle ship i respectively in scenario k.

Several binary variables are used to determine if the own-ship is in a situation where COLREGs applies. The vectors shown in Figure 4.14 are used to find the value of these variables. The own-ship is said to be $CLOSE$ to obstacle ship i at time t in scenario k if condition in equation (4.11) is true.

$$d_{0,i}^k(t) \leq d_i^{cl} \tag{4.11}$$

$d_{0,i}^k(t)$ is the predicted distance between the own-ship and the obstacle ship i in scenario k at time t and d_i^{cl} is the distance two ships must have before COLREGs start to apply.

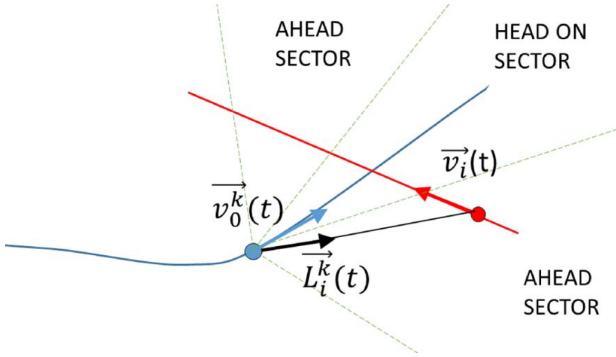


Figure 4.14: Collision situation between the own-ship and an obstacle ship. Figure is taken from Johansen et al. (2016).

If the distance between the two ships is larger than d_i^{cl}, COLREGs does not have to be considered.

The own-ship is $OVERTAKEN$ by the obstacle ship i if the obstacle ship has a higher speed than the own-ship and condition in equation (4.12) is true.

$$\boldsymbol{v}_0^k(t) \cdot \boldsymbol{v}_i(t) > \cos(68.5)^\circ |\boldsymbol{v}_0^k(t)||\boldsymbol{v}_i(t)| \tag{4.12}$$

The obstacle ship i and the own-ship are *HEAD-ON* at time t in scenario k if the obstacle ship i is $CLOSE$ to the own-ship, the obstacle ship's speed $|\boldsymbol{v}_i(t)|$ is not close to zero and conditions in equations (4.13a) and (4.13b) are true.

$$\boldsymbol{v}_0^k(t) \cdot \boldsymbol{v_i}(t) < -\cos(22.5)^\circ |\boldsymbol{v}_0^k(t)||\boldsymbol{v}_i(t)| \tag{4.13a}$$

$$\boldsymbol{v}_0^k(t) \cdot \boldsymbol{L}_i^k(t) > \cos(\phi_{ahead})||\boldsymbol{v}_0^k(t)| \tag{4.13b}$$

ϕ_{ahead} is a tunable parameter.

Obstacle ship i is $CROSSED$ by the own-ship at time t in scenario k if it is $CLOSE$ to the own-ship and condition in equation (4.14) is true.

$$\boldsymbol{v}_0^k(t) \cdot \boldsymbol{v}_i(t) < \cos(68.5^\circ) |\boldsymbol{v}_0^k(t)||\boldsymbol{v}_i(t)| \tag{4.14}$$

The obstacle ship i is $STARBOARD$ relative to the own-ship at time t in scenario k if the bearing angle the vector $\boldsymbol{L}_i^k(t)$ is larger than the own-ship's heading angle.

Hazard computation and scenario selection

A hazard is associated with each scenario. The control behavior in the scenario with the lowest hazard is chosen for execution. A new control behavior is chosen every 5 seconds,

to be able to utilize new sensor data. The hazard is a function of several parameters such as risk collision factor, cost of collision, grounding penalty and a function f that represents the cost of deviation from traveling in a straight path. These parameters will be discussed next.

The risk collision factor is defined as in equation (4.15)

$$\mathcal{R}_i^k(t) = \begin{cases} \frac{1}{|t-t_0|^p}\left(\frac{d_i^{safe}}{d_{0,i}^k(t)}\right)^q, & \text{if } d_{0,i}^k(t) \leq d_i^{safe} \\ 0, & \text{otherwise} \end{cases} \tag{4.15}$$

The parameters d_i^{safe} and the exponent $q \geq 1$ are chosen such that the own-ship keeps enough distance from the obstacle, as required in COLREGs rule 16. t_0 is current time and $t > t_0$ is a future point in time. The exponent $p \geq 0.5$ is a factor that determines how important the time until the event occurs is. Avoiding collisions in the near future is more important than avoiding collisions in the far future.

The cost of collision with obstacle ship i at time t in scenario k is given by equation (4.16).

$$\mathcal{C}_i^k(t) = K_i^{coll}|\boldsymbol{v}_0^k(t) - \boldsymbol{v}_i^k(t)|^2 \tag{4.16}$$

This cost is proportional to the difference in velocity with proportionality factor K_i^{coll}. This proportionality factor depends on several parameters such as the size of the obstacle.

A binary variable $\mu_i^k \in \{0, 1\}$ is used to indicate a violation of COLREGs rules 14 or 15 (head-on and crossing situations) between the own-ship and the obstacle ship i in scenario k at time t. A value of 1 means that COLREGs are violated. In order to determine if COLREGs rule 14 or 15 are violated, equation (4.17) is used. *RULE 14* and *RULE 15* are binary variables that evaluate to $True$ if the respective COLREG rule is violated. Equation (4.17) also implicitly incorporates rule 13 for the overtaking situation.

$$\begin{aligned} \text{RULE 14} &= \text{CLOSE \& STARBOARD \& HEAD-ON} \\ \text{RULE 15} &= \text{CLOSE \& STARBOARD \& CROSSED } \textit{ NOT OVERTAKEN} \\ \mu_i^k(t) &= \text{RULE 14 or RULE 15} \end{aligned} \tag{4.17}$$

Compliance with COLREGs rules 8 and 16 (which requires that collision maneuvers should be taken early and that the give-way vessel should keep out of the way) is achieved by choosing the prediction horizon length to be a lot longer than the time it takes to make a significant change in heading angle and speed.

The total hazard for scenario k based on the information available at time t_0 is given by equation (4.18).

$$\begin{aligned} \mathcal{H}^k(t_0) = \max_i \max_{t \in D(t_0)} &(\mathcal{C}_i^k(t)\mathcal{R}_i^k(t) + \kappa_i \mu_i^k(t)) \\ &+ f(P^k, \chi_{ca}^k) + g(P^k, \chi_{ca}^k) \end{aligned} \tag{4.18}$$

t_0 is the current time and the set of discrete time samples is denoted by

$$D(t_0) = t_0, t_0 + T_s, ..., t_0 + T$$

where T_s is the discretization interval and T is the length of the prediction horizon. κ_i is a tunable parameter. The total hazard $\mathcal{H}^k(t_0)$ depends on the previously mentioned risk collision factor in equation (4.15) and the cost of collision $\mathcal{C}_i^k(t)$ given in equation (4.16). Inclusion of the binary variable μ_i^k in the total hazard makes sure that violation of COLREGs will increase the hazard. The total hazard also depends on a function $g(\cdot)$, which represents a penalty for grounding. The function $f(\cdot)$ given in (4.19) is included to give higher cost for control behaviors that make the own-ship deviate from the nominal path and nominal speed.

$$f(P^k, \chi_{ca}^k) = k_P(1 - P^k) + k_\chi(\chi_{ca}^k)^2 + \Delta_P(P^k - P_{last}^k) + \Delta_\chi(\chi_{ca}^k - \chi_{ca,last}^k) \quad (4.19)$$

Δ_P and Δ_χ are penalty functions. Parameters k_P and k_χ are positive tuning parameters that determine how important it is to keep nominal speed and course. As before, P is the propulsion command and χ_{ca} is the course angle offset.

Simulation results in the paper by Johansen et al. (2016) reveal that this method is able to handle different collision scenarios while complying with the COLREGs. The majority of the simulation results discussed in Johansen et al. (2016) are cases where the obstacles travel along a straight path. Also, functionality to make the obstacle ships comply with COLREGs is not included. Tuning of parameters is important to achieve a good result. SBMPC can be difficult to implement due to a large number of tunable parameters.

As with the other collision avoidance methods in the previous subchapters, it is possible to include intentions to the SBMPC algorithm. This will be further discussed in the next chapter.

Chapter 5

Initial design of a collision avoidance system utilizing other vessels' intentions

The previous chapter discussed how some existing methods for collision avoidance can be modified to utilize other vessels' intentions. Now, the initial design of a collision avoidance system utilizing other vessels intentions will be given. This chapter begins by specifying some assumptions made and the choice of the design before continuing with a discussion on which data about a ship's intentions that can be used by a collision avoidance system. Towards the end of the chapter, the initial design of a collision avoidance system utilizing other vessels' intentions will be given. The initial design in this preliminary project will be concerned with how to include other vessels' intentions to the original SBMPC method.

5.1 Assumptions, scope and choice of the design

For this preliminary project, it is assumed that all vessels are equipped with a communication system that allows ships to share data about their intentions. It is also assumed that all vessels are equipped with AIS, hence real-time data about position, velocity and heading angle is available. All measured states, both for the own-ship and obstacle ships, are assumed to be accurate and without measurement noise. It is also assumed that a long-term path planning method is used to provide the long-term planned route, as a set of waypoints.

There are several possible approaches to design a collision avoidance algorithm. One way is to use a negotiation system where ships involved in the collision communicate with each other and together agree on which maneuvers each ship should take to satisfy COLREGs. The principle of such a negotiation system is presented by Szlapczynska (2015). However, this preliminary project will take a different approach, because in future work the performance of an existing collision avoidance method will be compared with another method that utilizes other vessels' intentions. Therefore, the design will be based on an existing collision avoidance method. An advantage of using an existing collision avoidance method as a basis for the design is that in case of communication failure of intentions, the algorithm can fall back on the existing collision avoidance method. This will improve the robustness.

An initial design of a short-term collision avoidance system utilizing other vessels' intentions will be presented. In addition to achieving COLREGs compliance, the algorithm is required to be able to avoid both static and dynamic obstacles. The long-term path planning is outside the scope of the proposed algorithm. The design will be based on the existing SBMPC method from Johansen et al. (2016) and the method will be extended to include other vessels' intentions. The proposed modified SBMPC algorithm is meant to be used by an autonomous ship, but it might be possible to also use it as a decision support system that tells the navigator on a manned ship which heading angle and propulsion command should be chosen.

In chapter 4.2, four different collision avoidance algorithms were presented. There are several reasons why the design of the new collision avoidance algorithm is based on SBMPC rather than on the other collision avoidance methods in chapter 4.2. The scope of this project is restricted to a short-term algorithm for collision avoidance, which excludes the COLREG-RRT method. The MVFF method from Lee et al. (2004) was based on the use of fuzzy rules. This method will not be chosen for further development because the paper only considers a very restricted set of possible actions. In addition, only encounters involving one obstacle ship are tested and the algorithm might not perform as well in the presence of several obstacles. The VO algorithm by Kuwata et al. (2014) does not consider vehicle dynamics and motion of obstacle ships, which suggests that the method could benefit from including intentions. However, according to Moe and Pettersen (2017) it is challenging to integrate VO with existing path-following methods. Therefore, this method will not be chosen either.

5.2 Data about a ship's intentions

A collision avoidance system utilizing other vessels' intentions needs data from obstacle ships about their future intentions. As of today, real-time information about obstacle ships such as speed, heading and position is available through AIS. A problem is that most of the future intentions for a manually operated ship are only known to the officer of the watch, and therefore not available for automatic transmission. To be able to transmit data about

intentions, the navigator needs to manually input his intentions into a computer system. For autonomous ships, data about intentions would already be available.

There has not been a lot of research regarding sharing of intentions. However, early in 2019 the STM BALT SAFE project started, where the objective is to improve the safety of navigation (STM Balt Safe, 2019). Among other things, the STM BALT SAFE project investigates the possibility of exchanging information between ships to increase situational awareness, which includes testing and development of a concept for exchanging route plans. The project is estimated to be completed in June 2021. A forerunner project for STM BALT SAFE was the MONALISA project, which lasted from 2010-2013. MONALISA considered a concept for route exchange between ships where a shore-based ship traffic coordination center receives the route suggestion from several ships, checks for collisions and gives feedback to the ships. The route is accepted if there is no danger of collision. On the other hand, the coordination center suggests changes to the route if a collision is probable. As part of MONALISA, Porathe et al. (2014) tested how cadets and bridge officers reacted to a prototype of this route exchange concept. The study showed a high degree of acceptance for the concept.

Possible data about future intentions for ships that could be used to improve the performance of a collision avoidance system is important to consider. Perhaps the most obvious choice would be data about the planned route, which is already stored in the navigation system. The future route plan could consist of a set of waypoints that define the future positions of the ship, and linear interpolation could then be used to approximate the future route as a set of straight lines between these waypoints. Knowing the future route of obstacle ships would be useful in order to plan collision avoidance maneuvers for the own-ship. A natural question that occurs is how many future waypoints of the planned route are necessary to share? This would depend on the distance between the own-ship and the obstacle ship.

Before the voyage starts, the route for a ship is planned. A problem with exchanging future route plans is that the route plan for the obstacle ships can change during the voyage, as illustrated in Figure 5.1. Therefore, it would be useful to regularly update the planned route to include planned deviations from the original planned route. The data exchanged between ships should contain up to date information about the planned route. In addition to the updated planned route, information about speed changes for the obstacle ships could be useful. When both speed and future route is available, the future position at different points in time can be computed.

It might be useful to know which role the obstacle ship is planning to take, either as a give-way vessel or a stand-on vessel. If the COLREGs require that the obstacle ship should be a give-way vessel, while the obstacle ship has planned to be a stand-on vessel, the own-ship needs to perform collision avoiding maneuvers.

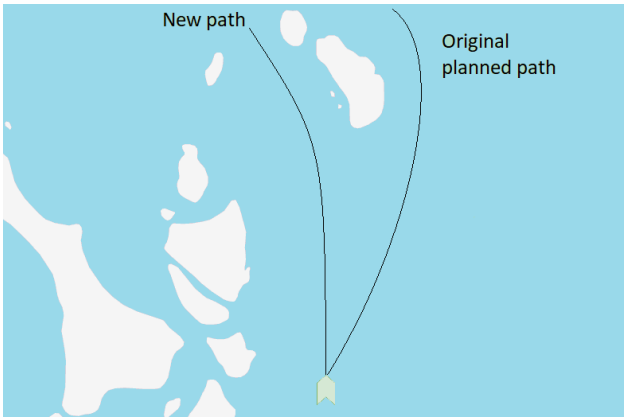


Figure 5.1: During the voyage the originally planned path can change.

5.3 Description of the modified SBMPC algorithm

As previously mentioned, the algorithm to be designed is based on the SBMPC algorithm from Johansen et al. (2016) and the algorithm will be extended to include other vessels' intentions. SBMPC is a short-term collision avoidance algorithm. A modular approach is used, where the collision avoidance part is separated from the mission planning. The collision avoidance module suggests a control behavior (consisting of a course offset and a propulsion command) that allows the own-ship to deviate from the planned path to avoid obstacles. To find the best control behavior, several scenarios are simulated. Each scenario consists of a control behavior for the own-ship, a computed own-ship trajectory when following this control behavior and a trajectory for each obstacle ship. The simulation length of a scenario is called the prediction horizon. A hazard score for each scenario is computed and the control behavior associated with the scenario with the lowest total hazard is chosen. The SBMPC method is discussed in more detail in chapter 4.2.4.

5.3.1 Requirements and important aspects to consider

The modified SBMPC algorithm is required to achieve COLREGs compliance since many collisions happen due to the violation of COLREGs. To achieve COLREGs compliance it is necessary to keep a safe distance from obstacles and perform collision avoiding maneuvers as early as possible. As mentioned in chapter 3.1.2, maneuvers made need to be clearly visible to surrounding ships. Consequently, instead of turning an angle of $10°$, the ship should instead turn with a more aggressive angle, perhaps $20° - 30°$. A course alteration is often preferred over a speed alteration since a course alteration is more visible to surrounding ships. Another reason for this preference is that due to the slow dynamics of ships, a significant course alteration is often faster than a significant speed alteration (Li et al., 2019). Rule 8 in the COLREGs also specifies that alteration of course angle often is

the most effective action. However, reducing the speed is often beneficial due to increasing time to collision which allows for more time to perform collision avoiding maneuvers.

The algorithm should be able to avoid both static and dynamic obstacles at the same time. Environmental obstacles such as shallow waters and shorelines will be included as static obstacles, which implies that the ship should be able to avoid grounding as well as collisions with other ships. From the study of common reasons for collisions in chapter 3.3, it was found that a collision often occurs due to restricted maneuverability due to static obstacles and surrounding ships not directly involved in the collision situation.

A problem with using other vessels' intentions in a collision avoidance system is that the intentions will change over time, depending on how the situation develops. Therefore, data about intentions needs to be updated regularly. When new intentions are available, the total hazard should be computed again and the system chooses a new control behavior.

5.3.2 Modifications to utilize other vessels' intentions

This section will discuss three different ways to include other vessels' intentions to the SBMPC algorithm proposed by Johansen et al. (2016). In order to predict the future trajectory of each obstacle ship, the original SBMPC method assumes that the obstacles will continue along a straight line. The future positions for the obstacles can then be calculated as shown in equation (4.10). If the obstacle ship's future planned trajectory is utilized, it can be used to replace the estimated future obstacle positions. To find obstacle positions from their planned route, the velocity needs to be known. One option is to assume that the obstacle ship will keep her current velocity. Another option is to use data about future planned velocity in the calculation. If data about planned deviations from the planned trajectory is available, this data can be used to update the planned trajectory.

Adding a variable to indicate a future collision

The obstacle vessel's intentions could be used to improve the hazard computation. The expression for total hazard can be modified by including the term given by equation (5.1).

$$FC_i^k(t) \cdot (2 + P^k) \cdot w_{FC} \tag{5.1}$$

$FC_i^k(t)$ is a boolean variable to indicate a Future Collision. In a given scenario, the future trajectory of the own-ship is computed by applying the control behavior for that scenario. Assuming that the speed and future trajectory for obstacle ship i is known, the future position of the obstacle ship at time t can be found. If the own-ship is closer than l meters away from the obstacle ship i at time t in the prediction horizon in scenario k, then $FC_i^k(t) = True$. An example is shown in Figure 5.2. In this situation each of the scenarios A, B and C have a different control behavior and therefore a different own-ship trajectory. The obstacle ship's trajectory is the same in all scenarios. The planned path for

the obstacle ship intersects the own-ship trajectory in scenario A, indicating that there is a potential collision. In this example, $FC_i^k(t)$ can only become $True$ in scenario A. If the own-ship and the obstacle ship are closer than a distance l at time t during the prediction horizon in scenario A, then $FC_i^{k=A}(t) = True$.

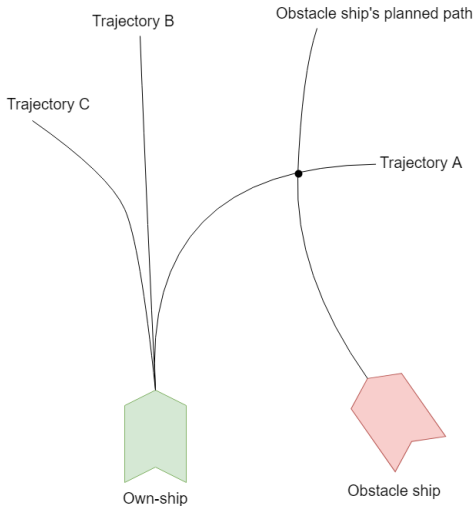


Figure 5.2: Illustration of how to find the value of $FC_i^k(t)$. The variable $FC_i^k(t)$ can only become $True$ in scenario A if the two ships are sufficiently close at time t in the prediction horizon.

The binary variable $FC_i^k(t)$ is multiplied with the term $(2 + P^k)$ where $P^k \in [-1, 1]$ is the propulsion command in scenario k. If $P^k = 1$ (keep nominal speed), the hazard will be higher than if $P^k = -1$ (full reverse). $FC_i^k(t)$ is also multiplied with w_{FC}, a tunable parameter to determine how important $FC_i^k(t)$ is for the total hazard. The final expression for the total hazard is shown in equation (5.2).

$$\begin{aligned}\mathcal{H}^k(t_0) = \max_i \max_{t \in D(t_0)} (\mathcal{C}_i^k(t)\mathcal{R}_i^k(t) + \kappa_i \mu_i^k(t) + FC_i^k(t)(2 + P^k)w_{FC}) \\ + f(P^k, \chi_{ca}^k) + g(P^k, \chi_{ca}^k)\end{aligned} \tag{5.2}$$

An explanation of the terms in equation (5.2) is given in chapter 4.2.4.

Penalize deviation from nominal path for stand-on vessel

The total hazard expression in equation (5.2) can be further modified. Assume the obstacle ship tells which role she plans to take in a collision situation, either as a stand-on vessel or a give-way vessel. The role of obstacle ship i in scenario k can be represented by a binary variable r_i^k whose value is $True$ if the obstacle ship i intends to take the role as a give-way vessel at any point in time during the prediction horizon. If $r_i^k = True$ this implies that the own-ship takes the role of a stand-on vessel (should keep her current course), which means that the control behavior that gives large course offset should have

an increased cost. The reason behind this comes from rule 17 in the COLREGs, which states that the stand-on vessel should, if possible, keep her current course. To implement this, the binary variable r_i^k multiplied with a scaling parameter k_r will be added to the function $f(P^k, \chi_{ca}^k)$ in the total hazard expression. The new function becomes

$$f(P^k, \chi_{ca}^k, r_i^k) = k_P(1 - P^k) + k_\chi(\chi_{ca}^k)^2 + \Delta_P(P^k - P_{last}^k) + \Delta_\chi(\chi_{ca}^k - \chi_{ca,last}^k) + r_i^k k_r \tag{5.3}$$

The function $f(P^k, \chi_{ca}^k, r_i^k)$ in equation (5.3) penalize deviations from the nominal path. If $r_i^k = True$ (the obstacle ship i is a give-way vessel and therefore the own-ship is a stand-on vessel), then the value of the function $f(P^k, \chi_{ca}^k, r_i^k)$ will increase (deviation from nominal path will increase) which makes the total hazard increase. The function $f(P^k, \chi_{ca}^k, r_i^k)$ should be maximized over i.

Improve the accuracy of the COLREGs violation variable

The total hazard $\mathcal{H}^k(t_0)$ is computed by maximizing over each time step in the prediction horizon. Since the original SBMPC implementation assumes that the obstacle ships follow a straight line, the velocity vector $v_i(t)$ for the obstacle ship i will have the same angle throughout the prediction horizon. When the future route and future speed for the obstacle ship are known, the position and the velocity vector $v_i(t)$ can be found at future points in time. Since the angle of the velocity vector is used to compute the binary variable $\mu_i^k(t)$, which represents violation of COLREGs rules 14, 15 and 13, utilizing intentions to obtain a better approximation of $v_i(t)$ could potentially make the binary variable $\mu_i^k(t)$ be more accurate.

Chapter 6

Simulator development

In this chapter, the simulator development will be presented. A 3 DOF model of a Platform Supply Vessel is implemented using MATLAB and simulations are performed with DNV GL's CyberSea as simulator core. This chapter will explain how the reference models, control system and LOS guidance are implemented. The inclusion of obstacle ships and static obstacles in the simulator will also be discussed.

6.1 Simulator overview

In this project, the simulator consists of three different modules. The simulator core used in this project is CyberSea, developed by DNV GL. MATLAB & Simulink have been used to create Functional Mockup Units (FMUs). An overview of the simulator modules can be found in Figure 6.1. The PSV module contains the implementation of a Platform Supply Vessel (PSV), including guidance and control systems for the vessel. A module for dynamic obstacles (obstacle ships) is also included. These obstacle ships are also PSVs and use the same parameters, control system and guidance system as the own-ship in the PSV module. Finally, the last module is the collision detection module. It will take the state of the obstacle ships and the own-ship as input, in addition to the position and parameters for the static obstacles. The static obstacles are used to represent environmental obstacles such as archipelago and shores.

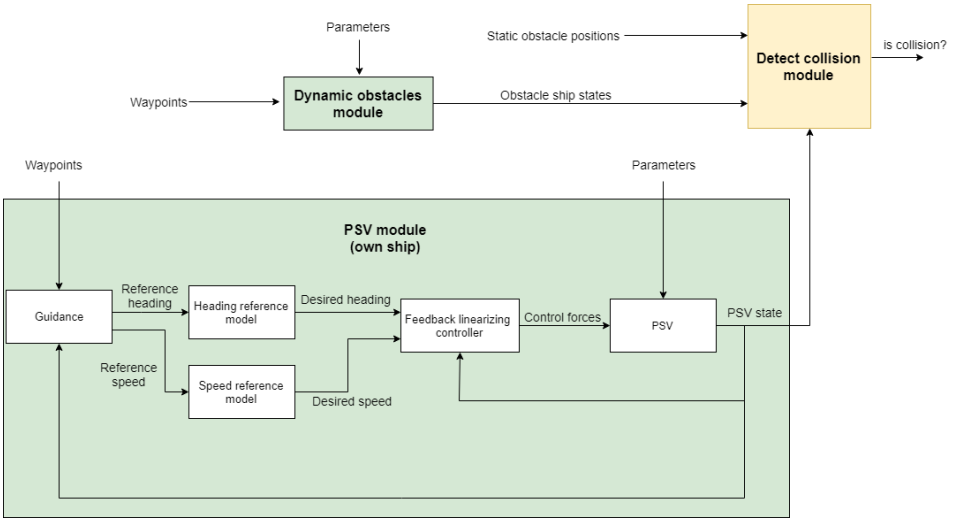


Figure 6.1: Block diagram of modules for the simulator.

6.2 3 DOF Ship model

The ship model implemented in the PSV module in Figure 6.1 is a 116-meter long Platform Supply Vessel and the model parameters are provided by DNV GL. This is the same ship model used by Minne (2017). The ship parameters can be seen in Table 6.1. $|Y_v|$ and $|N_r|$ have been increased compared to the parameters used in Minne (2017) to improve the stability of the ship.

In order to simulate the dynamics of the ship, the 3 DOF ship model from chapter 2.2 is used. For simplicity, the effects of environmental forces are not considered. It is also assumed that the ship's state vector $[x, y, y, \psi, u, v, r]^T$ is directly available, without measurement noise.

6.3 Testing maneuverability of a 6 DOF model

Maneuverability tests for a 6 DOF PSV model have been performed using DNV GL's simulator. The goal of these maneuverability tests was to find realistic behavior regarding maneuverability for a ship and also determine realistic upper limits for forces and moments that can be provided by the actuators. Since the 6 DOF model is quite similar to the 3 DOF PSV model in chapter 6.2, the behavior of the two models should be similar. The 6 DOF ship model used for testing has a total of six actuators, as shown in Figure 6.2. During transit, only the rear thrusters 4 and 6 are used. Therefore, only thrusters 4 and 6 as seen in Figure 6.2 have been used in the simulations. DNV GL's simulator is able to provide log

Parameter	Value	Unit
m	15524000	kg
ship length	116	m
ship width	25	m
I_z	$1.0437 \cdot 10^{10}$	$kg \cdot m^2$
x_g	-3.7	m
$X_{\dot{u}}$	-979290	kg
$Y_{\dot{v}}$	-10727527	kg
$Y_{\dot{r}}$	-11357800	$kg \cdot m$
$N_{\dot{r}}$	$-6.2422 \cdot 10^9$	kg/m^2
X_u	-1650	kg/s^2
Y_v	-1050060	kg/m^2
Y_r	0	$kg \cdot m/s$
N_v	0	$kg \cdot m/s$
N_r	-2452793600	$kg \cdot m^2/s$
T_{surge}	10000	s
T_{sway}	2500	s
T_{yaw}	1700	s

Table 6.1: Model parameters for the PSV.

files with relevant signal values, and these log files have been used to generate the plots in this section. For simplicity, the effect of external forces has been turned off during these tests.

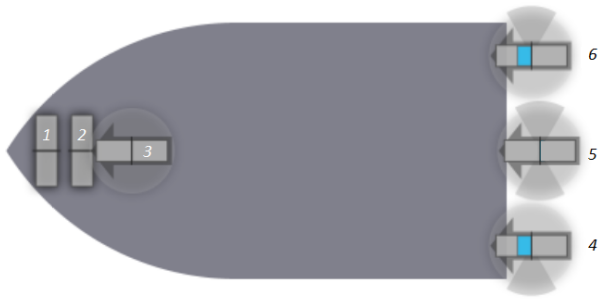


Figure 6.2: Actuators on the 6 DOF ship model used to test maneuverability. This ship is equipped with three main thrusters (4,5 and 6), two tunnel thrusters (1 and 2) and one rotating azimuth thruster (3).

6.3.1 Surge maneuverability

During this test, the goal was to determine how fast this ship is able to accelerate in surge direction and also find limits for thruster forces in surge direction. The ship starts with an initial speed of 0 m/s and both thruster 4 and 6 are at 100% capacity. The resulting speed and surge force responses can be seen in Figure 6.3. It seems like this model converges towards a speed of 5 m/s after approximately 300 seconds and a speed of 3 m/s is achieved after approximately 80 seconds. In this case, both thrusters delivered an equal amount of force. Each thruster was able to provide a maximum of 450 000 N. The maximum force is achieved after approximately 15 seconds. The reason why the thrust force starts to decline in Figure 6.3 is partly due to thruster losses.

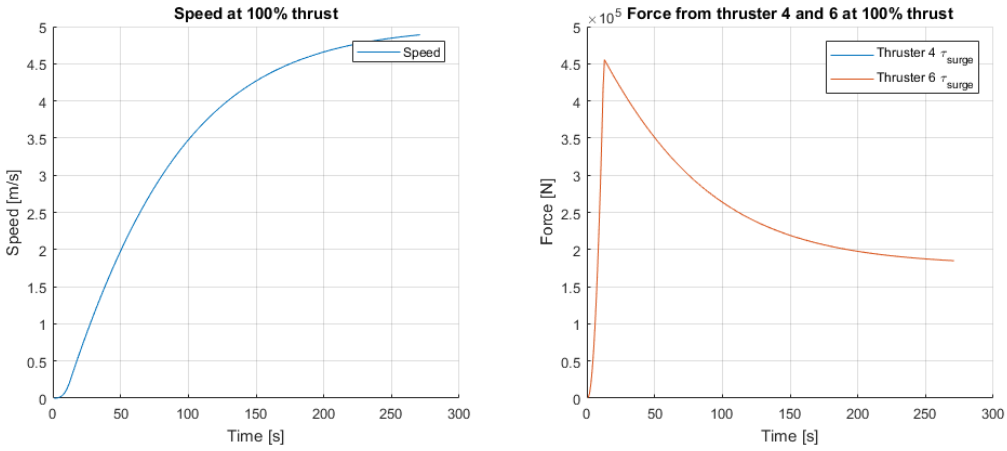


Figure 6.3: Speed and force response for surge maneuverability test, when both thruster 4 and 6 are at 100% capacity.

6.3.2 Yaw maneuverability

The goal of the yaw maneuverability tests was to find how fast the ship is able to change its heading angle, and also find limits for maximum yaw torque the actuators can deliver. In the first test, the ship starts with zero initial speed and heading, while thruster 4 was set to 100% capacity and a rudder angle of -15° (thruster 6 was set to 0%). The moment and yaw responses can be seen in Figure 6.4. Thruster 4 is able to deliver a maximum yaw moment of approximately 10 000 000 Nm. It takes almost 140 seconds before a heading angle of 180° is achieved. The peak yaw moment of 10 000 000 Nm is achieved after approximately 10 seconds, but the time it takes for the rudder to physically change its angle is not considered.

In the second yaw maneuverability test, the ship starts with zero speed and heading, while thrusters 4 and 6 are set to 85% of full capacity with a rudder angle of 0°. When the

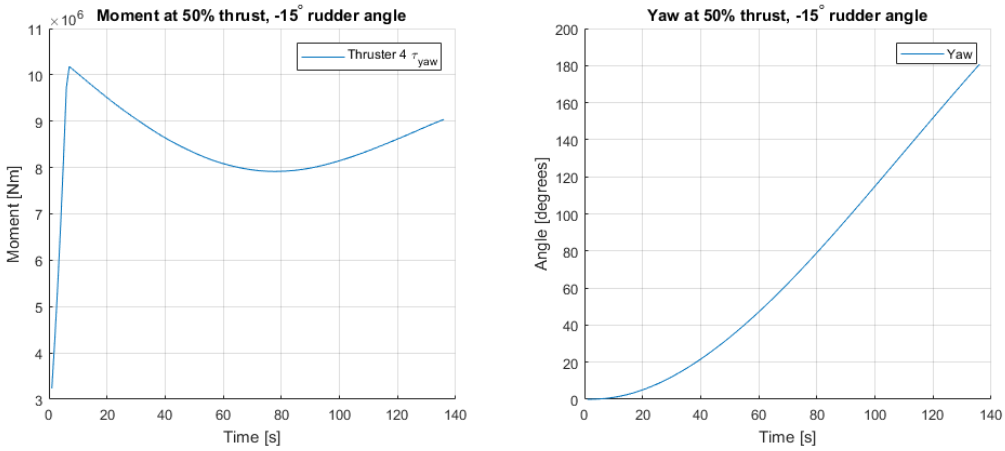


Figure 6.4: Yaw moment and yaw (heading angle) response for the first yaw maneuverability test, where thruster 4 is set to 100% capacity and -15°.

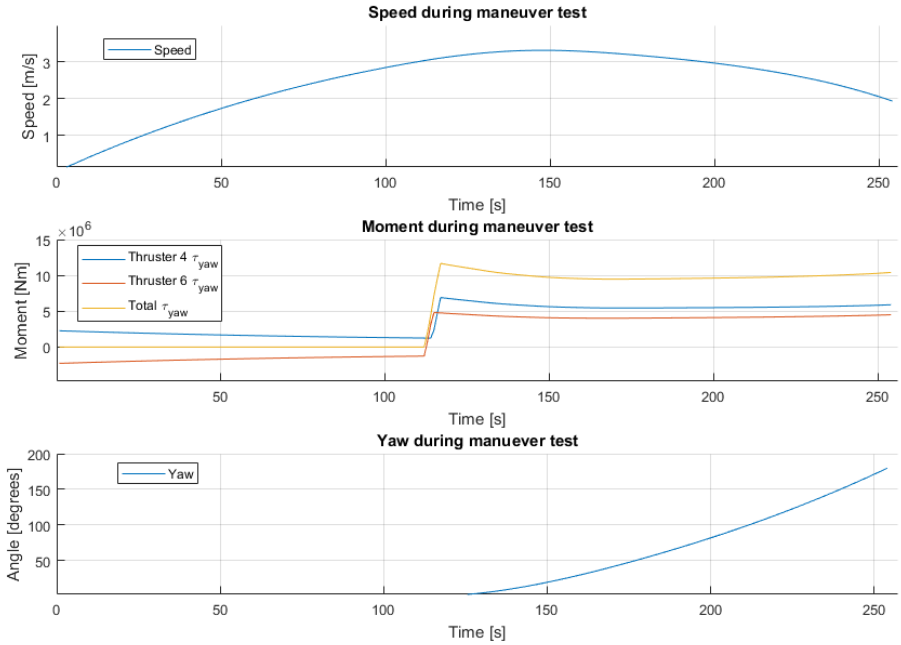


Figure 6.5: Speed, yaw moment and yaw (heading) angle response in the second maneuverability test, where thrusters 4 and 6 are set to 85% capacity and rudder angles are changed from 0° to -30° after 115 seconds.

ship has reached a speed of approximately 3.1 m/s (after 115 seconds), both rudder angles are changed to -30°. The response in speed, yaw moment and heading angle can be seen in Figure 6.5. Turning the ship from 0° to 180° takes 120 seconds. After approximately 60 seconds the ship has turned 45°. The actuators can deliver a combined maximum moment of just over 11 000 000 Nm.

6.3.3 Summary of maneuverability test results

The most important results from the maneuverability tests are summarized below. Since these results have been obtained for a 6 DOF model, it is expected that the 3 DOF model implemented in this project will have similar behavior, but not exactly the same.

- When starting at a speed of 0 m/s, it takes approximately 80 seconds to reach 3 m/s and approximately 300 seconds to reach 5 m/s.
- Each of the two thrusters is able to provide a maximum force in surge direction of 450 000 N.
- It will take approximately 15 seconds before maximum force in surge direction is achieved.
- With an initial speed of 0 m/s, a 180° turn takes approximately 140 seconds (using one thruster).
- With an initial speed of 3.1 m/s, a 180° turn takes approximately 120 seconds (using both thrusters). After approximately 60 seconds, the ship has turned 45°.
- Actuators are able to deliver approximately 11 000 000 Nm combined yaw torque.
- It will take approximately 10 seconds before the maximum moment in yaw is achieved (not including the time it takes for rudders to physically change their orientations).

6.4 Implementation of the control system and LOS guidance

To achieve good performance in a collision avoidance system, the underlying control system for speed and heading needs to work well. In this section, the implementation of the control system and LOS guidance will be presented. In the following discussion, the reader should be aware of the distinction between desired values and reference values. The desired values are the values that the control system tries to follow. These desired values are computed by the reference models (to be discussed in the next section). On the other hand, the reference values are the values provided by the guidance system.

6.4.1 Reference models for controllers

Reference models are used to generate feasible values for speed and heading that the ship is able to follow. The principle is explained in Figure 6.6. If the reference heading angle from the guidance system is a step signal, it can cause problems since the ship is not able to follow such abrupt changes in heading. Therefore, a reference model can be used to smooth out the step signal and provide a feasible, desired heading angle that the ship is able to follow. The control system tries to make the ship follow the desired heading angle computed by the reference model. In the simulator developed in this project, two different reference models are used. One to generate the desired heading and another reference model is used to generate the desired velocity.

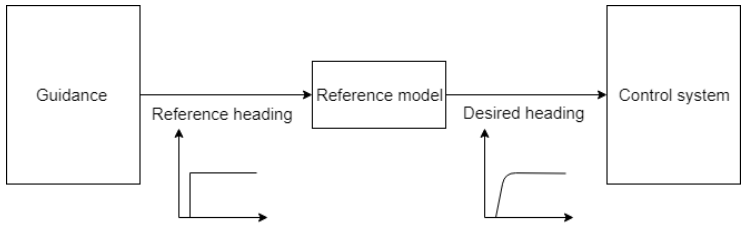


Figure 6.6: Principle for a reference model for heading.

Velocity reference model

The reference model for velocity receives reference velocity as input from the guidance system and calculates the desired velocity vector $\boldsymbol{\nu}_d = [u_d, v_d, r_d]^T$ as output. The same notation as in chapter 2.2 will be used, where u and v are speed in surge and sway and r is the yaw rate. The subscript d will be used to denote a desired value and subscript ref will be used for reference values. The velocity controller's aim is to make the ship follow $\boldsymbol{\nu}_d$ as closely as possible. It is desirable to avoid step signal in both $\boldsymbol{\nu}_d$ and $\dot{\boldsymbol{\nu}}_d$. Therefore, a second-order lowpass filter is used as a reference model. The resulting velocity reference model is a second-order lowpass filter given by equation (6.1) from Fossen (2011).

$$\ddot{\boldsymbol{\nu}}_d = 2\boldsymbol{\Delta}\boldsymbol{\Omega}\dot{\boldsymbol{\nu}}_d + \boldsymbol{\Omega}^2\boldsymbol{\nu}_d = \boldsymbol{\Omega}^2\boldsymbol{\nu}_{ref} \tag{6.1}$$

$\boldsymbol{\nu}_d$ is the desired velocity vector and $\boldsymbol{\nu}_{ref}$ is the reference velocity vector that is computed by the guidance system. $\boldsymbol{\Delta}$ and $\boldsymbol{\Omega}$ are symmetric, positive definite design matrices. Their diagonals contain relative damping ratios and natural frequencies as shown in equations (6.2a) and (6.2b).

$$\boldsymbol{\Delta} = diag\{\zeta_1, \zeta_2, \zeta_3\} \tag{6.2a}$$

$$\boldsymbol{\Omega} = diag\{\omega_{n1}, \omega_{n2}, \omega_{n3}\} \tag{6.2b}$$

The chosen parameters for relative damping ratios and natural frequencies can be seen in Table 6.2.

Parameter	Value
ζ_1	1
ζ_2	1
ζ_3	1
ω_{n1}	0.035
ω_{n2}	0.14
ω_{n3}	0.19

Table 6.2: Relative damping ratios and natural frequencies used in the velocity reference model.

The velocity reference model is implemented by using the state-space version of equation (6.1), given by equation (6.3).

$$\dot{\boldsymbol{x}}_d = \boldsymbol{A}_d \boldsymbol{x}_d + \boldsymbol{B}_d \boldsymbol{r} \tag{6.3}$$

where $\boldsymbol{x}_d = [\boldsymbol{\nu}, \dot{\boldsymbol{\nu}}]$ and $\boldsymbol{r} = [u_{ref}, v_{ref}, r_{ref}]^T$. The reference velocity in y-direction, v_{ref} is set to 0, because the ship will be used for path following. r_{ref} is also set to zero. The matrices $\boldsymbol{A}_d$ and $\boldsymbol{B}_d$ are given by equations (6.4a) and (6.4b)

$$\boldsymbol{A}_d = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{\Omega}^2 & -2\boldsymbol{\Delta}\boldsymbol{\Omega} \end{bmatrix} \tag{6.4a}$$

$$\boldsymbol{B}_d = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Omega}^2 \end{bmatrix} \tag{6.4b}$$

where $\boldsymbol{0}$ is a 3x3 zero matrix and $\boldsymbol{I}$ is a 3x3 identity matrix.

Heading reference model

The reference model for the heading angle ψ is chosen to be a third-order lowpass filter, to avoid step signals in ψ, $\dot{\psi}$ and $\ddot{\psi}$. The third-order lowpass filter is given by equation (6.5) from Fossen (2011).

$$\boldsymbol{\eta}_d^{(3)} + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\eta}}_d + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d + \boldsymbol{\Omega}^3\boldsymbol{\eta}_d = \boldsymbol{\Omega}^3\boldsymbol{\eta}_{ref} \tag{6.5}$$

$\boldsymbol{I}$ is the 3x3 identity matrix, $\boldsymbol{\eta}_{ref} = [x_{ref}, y_{ref}, \psi_{ref}]^T$ are the reference positions and heading provided by the guidance system.

The heading reference model is implemented by using the state-space version of equation (6.5), given by equation (6.6)

$$\dot{\boldsymbol{x}_d} = \boldsymbol{A}_d \boldsymbol{x}_d + \boldsymbol{B}_d \boldsymbol{r} \tag{6.6}$$

where $\boldsymbol{x}_d = [\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \ddot{\boldsymbol{\eta}}]^T$, $\boldsymbol{r} = [0, 0, \psi_{ref}]^T$ and the matrices $\boldsymbol{A}_d$ and $\boldsymbol{B}_d$ are given in

equations (6.7a) and (6.7b).

$$\boldsymbol{A}_d = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{\Omega}^3 & -(2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2 & -(2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega} \end{bmatrix} \tag{6.7a}$$

$$\boldsymbol{B}_d = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{\Omega}^2 \end{bmatrix} \tag{6.7b}$$

The matrices $\boldsymbol{\Delta}$ and $\boldsymbol{\Omega}$ are the same matrices as the ones used in the velocity reference model.

6.4.2 Velocity and heading controllers

Two different controllers are used to control the ship. A Multi-Input Multi-Output (MIMO) feedback linearizing controller for speed control and a Proportional (P) controller for heading control. A general description of feedback linearizing controllers are given in chapter 2.4.

The speed controller is implemented as a feedback linearizing controller given in equation (6.8)

$$\boldsymbol{\tau} = \boldsymbol{M}(-\boldsymbol{K}_p\tilde{\boldsymbol{\nu}}) + \boldsymbol{n}(\boldsymbol{\nu}) \tag{6.8}$$

where $\boldsymbol{\tau}$ is the force vector, $\boldsymbol{M}$ is the mass matrix and $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \boldsymbol{\nu}_d$ and $\boldsymbol{\nu}_d = [u_d, v_d, r_d]^T$ is the output from the speed reference model. The nonlinear term $\boldsymbol{n}(\boldsymbol{\nu})$ is given by equation (6.9)

$$\boldsymbol{n}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}\boldsymbol{\nu} + \boldsymbol{C}_A\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu} \tag{6.9}$$

where $\boldsymbol{C}_A$ and $\boldsymbol{C}_{RB}$ are coriolis-centripetal matrices and $\boldsymbol{D}$ is a damping matrix. The controller gain matrix $\boldsymbol{K}_p$ is given by equation (6.10).

$$\boldsymbol{K}_p = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{6.10}$$

Inspired by Minne (2017), the heading controller is implemented as a P controller as shown in equation (6.11), to provide the desired yaw rate r_d in the feedback linearizing controller.

$$r_d = k_{p,\psi}(\psi_d - \psi) \tag{6.11}$$

where ψ_d is the desired heading provided by the reference model. The controller gain is chosen to be $k_{p,\psi} = 0.04$.

6.4.3 Implementation of LOS guidance

The LOS guidance method from chapter 2.3 is implemented to find the reference heading angle necessary to make the ship follow a straight line defined by two different waypoints. Reference course angle χ_r can be found from equation (2.9). Fossen (2011) defines the course angle as $\chi_r = \psi_r + \theta$. By assuming a sideslip angle $\theta = 0$ the reference course angle is equal to the reference heading angle.

The lookahead distance in the LOS guidance is chosen to be $\Delta = 1000$. A smaller value for Δ will give more aggressive steering and faster convergence to the path defined by the waypoints, but can make the vessel's position oscillate around the desired path. The switching mechanism in chapter 2.3 is implemented as shown in equation (6.12)

$$(x_{k+1} - x(t))^2 + (y_{k+1} - y(t))^2 \leq R^2 \tag{6.12}$$

where (x_{k+1}, y_{k+1}) is the position of the target waypoint and $(x(t), y(t))$ is the position of the ship. When the ship is a distance less than $R = 300$ from the target waypoint, the waypoints will switch. Fossen (2011) recommends choosing R as twice the ship length.

6.5 Implementation of static and dynamic obstacles

Functionality for adding static and dynamic obstacles is included in the developed simulator. There are two types of static obstacles called "circle obstacle" and "boundary obstacle". These obstacles can represent different environmental constraints such as archipelago, shallow water zones and shores. Figure 6.7 shows an example of two such static obstacles. The boundary obstacle is represented by a straight line, where one side is defined as illegal and the other side is defined as legal. If a ship's bow crosses this line and moves from the legal side to the illegal side, a collision is detected. In the simulations, it is assumed that the own-ship and obstacle ships always start on the legal side. For the circle obstacle, a collision is detected if a ship's bow crosses the circle's boundary.

The dynamic obstacle module introduced in chapter 6.1 has the functionality to add one or more obstacle ships to the simulation scenario. All obstacle ships have the same parameters, same control system and same LOS guidance as the own-ship PSV in chapter 6.2. Therefore, the implementation of obstacle ships is very similar to the implementation of the own-ship in the PSV module discussed previously.

To detect a collision between ships, a circle with a radius equal to half of the ship length (a radius of 58 meters) is defined. This circle is illustrated in Figure 6.8. If the bow of another ship is inside this circle, a collision is detected between the two ships involved. Consequently, a collision can be detected even if there is no physical impact because the circle covers a region larger than the ship itself. However, this is acceptable because the COLREGs require that two ships have a safe distance from each other anyway.

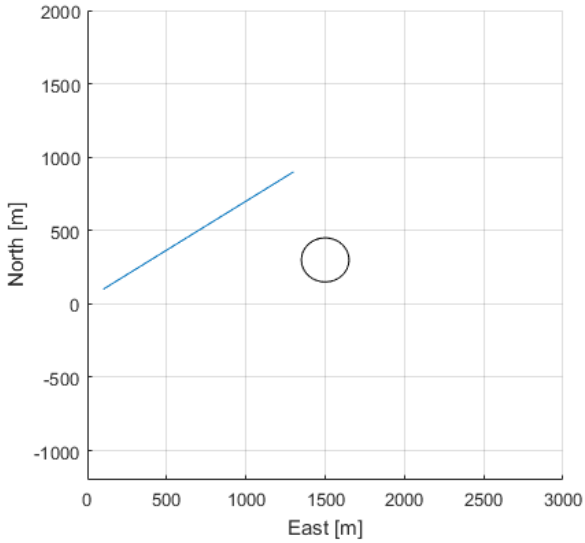


Figure 6.7: Boundary and circle obstacles plotted in the North-East coordinate system.

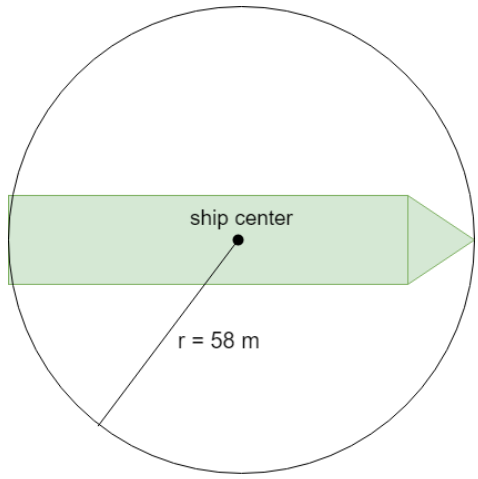


Figure 6.8: A circle with radius of 58 meters is used to detect a collision between ships. A collision is detected if the bow of another ship is inside this circle.

6.6 Simulation scenarios

A simulation scenario is defined by several parameters: number of obstacle ships, number of static obstacles, initial position for all ships and static obstacles, initial speed and heading for all ships and a planned path for all ships. In order to ensure that a collision is

bound to happen, the initial positions, headings and planned route for the ships need to be chosen with care. In order to choose initial heading angles that will lead to a collision (assuming ships will follow a straight line), it is possible to define a set of heading angles that the initial heading angle can be chosen from. An example of a set of acceptable heading angles is $\{0°, 45°, 90°, 125°, 180°, 225°, 270°, 315°\}$. Such a set is visualized in Figure 6.9. If two or more ships have initial heading angle chosen from this set (and none of the ships have the same initial heading), then a collision will eventually happen assuming that the ships follow a straight line. The point of collision is the center of the circle shown in the figure.

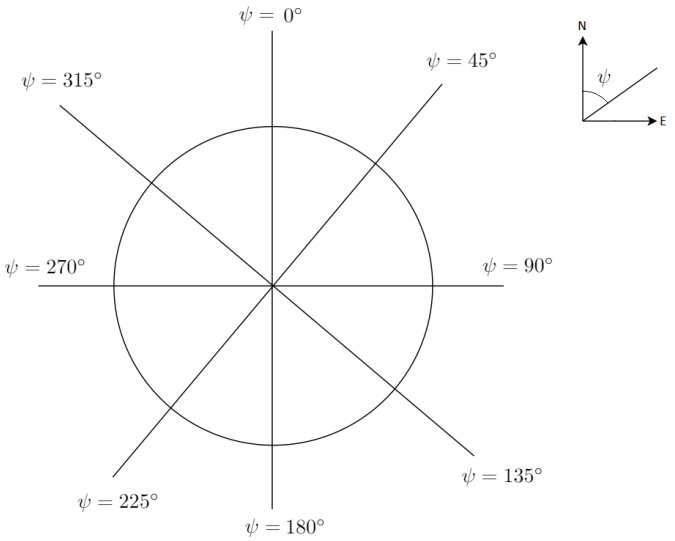


Figure 6.9: Possible choice of initial heading angles for a simulation scenario to ensure that a collision will happen (assuming ships travel on a straight line).

When initial heading angles are chosen, it is possible to calculate initial positions for the ships. First, choose a point $P_{\text{col}} = (x_{\text{col}}, y_{\text{col}})$ where two or more ships will collide. Next, choose a time until collision as t_{ttc}. Then we can calculate the initial position from equations (6.13a) and (6.13b).

$$x_{\text{init}} = x_{\text{col}} - t_{\text{ttc}} u_{\text{init}} \cos(\psi_{\text{init}}) \tag{6.13a}$$

$$y_{\text{init}} = y_{\text{col}} - t_{\text{ttc}} u_{\text{init}} \sin(\psi_{\text{init}}) \tag{6.13b}$$

where u_{init} is the initial speed and ψ_{init} is the initial heading angle.

If ships are allowed to travel in a curved path, finding scenario parameters that ensure a collision is a bit complex. Assume that in a given scenario there will be a collision between an obstacle ship following a straight path and the own-ship ship following a curved path. This scenario is illustrated in Figure 6.10. In such cases, the first step is to decide an

initial starting position and a curved path for the own-ship (defined by a set of waypoints $\{P_i\}$), and a straight path for the obstacle ship, defined by an initial heading angle. The point of collision is chosen as $P_{\text{col}} = (x_{\text{col}}, y_{\text{col}})$. Time until collision can be approximated by the relation $t_{\text{ttc}} = \frac{d}{u}$ where t_{ttc} is time to collision, d is the length of the curved path approximated by straight line segments between waypoints, u is the speed of the obstacle ship. When t_{ttc} is known, the initial position for the obstacle ship can be calculated from equations (6.13a) and (6.13b).

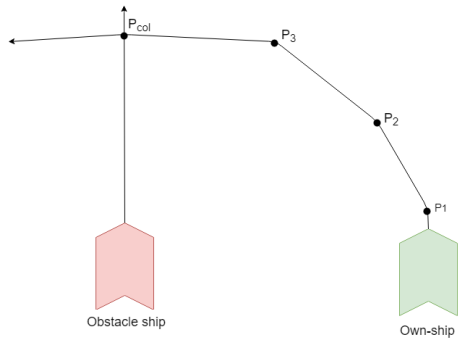


Figure 6.10: Scenario where the own-ship follows a curved path.

Chapter 7

Simulation results

In the previous chapter, the implementation of control system, guidance system as well as inclusion of static and dynamic obstacles in the simulator was discussed. In this chapter, the performance of the simulator will be tested. This chapter starts with an analysis of the performance of the control system (both speed control and heading control) for the own-ship in the PSV module, and continues with analysis of the performance of LOS guidance for the own-ship and for the dynamic obstacles. At the end of chapter, all the modules in the simulator will be connected together to test collision detection abilities of the simulator.

7.1 Performance of the control system

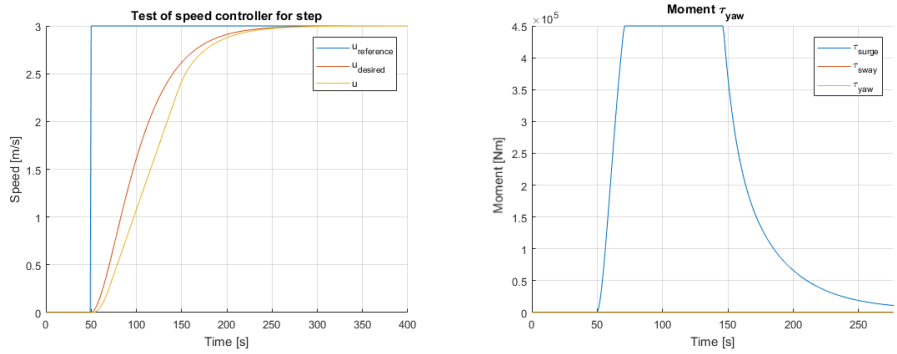
The control system for the PSV consists of two parts. A feedback linearizing controller is used to control the speed and a P controller is used to control the heading angle of the ship. The control system implementation is discussed in more detail in chapter 6.4.2. The purpose of the two control systems is to make the ship follow the desired heading angle and desired speed calculated by the reference models.

7.1.1 Performance of the speed controller

The performance of the speed controller is tested by providing a step in reference speed. The speed controller tries to follow the desired speed calculated from the speed reference model for this step reference. Based on the surge maneuvering test in chapter 6.3.1, it was found that a speed of 3 m/s is reached after approximately 80 seconds. Based on this result, the parameters for the speed reference model is chosen such that a change in speed of 3

m/s for the desired speed signal (output from the reference model) takes approximately 200 seconds (slightly over 3 minutes). The reason why the speed reference signal is chosen to reach 3 m/s quite a bit slower than what the results from the maneuver test indicate, is that the ship needs to decelerate in order to avoid overshoot. In the maneuver test, the thrusters were able to operate at 100% capacity the whole time. The saturation on the force in surge direction is chosen to be 450 000 N, which was found to be a realistic value based on the maneuverability tests in 6.3. From experimentation, it was found that the response in speed will be too slow if the saturation on surge is too low.

Initial velocities of the ship are set to zero, while the heading angle is kept at $0°$. The response of the ship from a step in the reference speed from 0 m/s to 3 m/s after 50 seconds can be seen in Figure 7.1a, and the corresponding force generated is displayed in Figure 7.1b. The force in surge direction does saturate after a while, which causes a deviation between the desired speed $u_{desired}$ and the actual speed u. However, the speed u converges to the desired speed after approximately 200 seconds without any overshoot. In the maneuvering tests in chapter 6.3, the maximum force in surge was achieved after approximately 15 seconds. The speed controller test shows a similar response. From the response in Figure 7.1b it can be seen that it takes approximately 20 seconds to reach maximum force in surge.



(a) Step response in speed

(b) Yaw moment

Figure 7.1: Response of speed controller with a step reference in speed from 0 m/s to 3 m/s after 50 seconds.

7.1.2 Performance of the heading controller

The performance of the heading controller is tested in a similar manner, by providing a step in the reference heading angle. The heading reference model calculates the desired heading for the ship to follow based on this step reference. The parameters of the heading reference model are chosen such that a heading change of 45° will take approximately 50 seconds. This is a realistic choice, based on the result from the yaw maneuvering test in chapter 6.3 where a heading change of 45° took approximately 60 seconds with a speed of

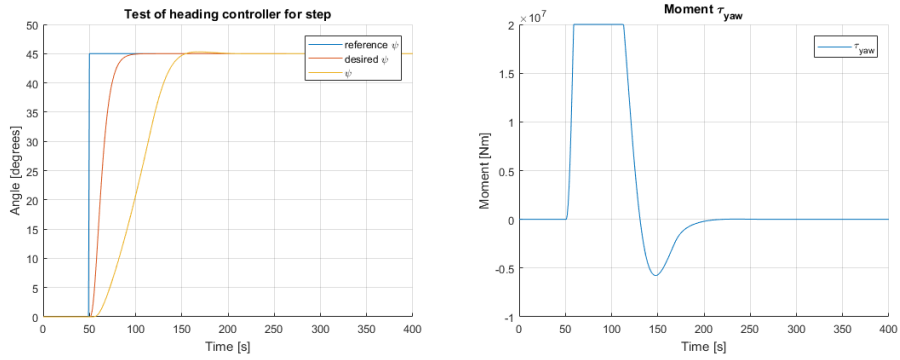
3.1 m/s. The controller is tuned such that minimal oscillations in heading angle ψ appear from a step in reference of 45°, while at the same time achieving a relatively fast response.

Initial velocity of the ship is set to $\boldsymbol{\nu} = [u = 3, v = 0, r = 0]^T$, and initial heading angle is set to zero. The response of the ship from a step in reference heading from 0° to 45° after 50 seconds can be seen in Figure 7.2a. The corresponding yaw moment can be seen in Figure 7.2b. It takes approximately 100 seconds before a heading change of 45° is achieved. Due to the saturation of yaw moment, there is a significant deviation between the desired heading and actual heading angle ψ. The yaw moment saturates after approximately 10 seconds. From maneuvering tests in chapter 6.3, the yaw moment also saturated after 10 seconds. However, the time it takes for the rudders to physically change their angles is not taken into account in the maneuvering tests. Therefore, for a real ship it is expected that it will take a longer time than 10 seconds before the maximum yaw moment is achieved.

Another test of the heading controller is performed. With the same initial velocity and heading, a step in reference heading from 0° to 180° is given. The heading response can be seen in Figure 7.3a. With the chosen heading controller parameters, a large step in heading reference can lead to a small overshoot in heading angle for the ship, but this was not the case for this particular example. In the yaw maneuver test in chapter 6.3, a heading change of 180° took approximately 120 seconds with a speed of 3.3 m/s. From Figure 7.3a we observe that a change in heading angle of 180° takes approximately 250 seconds, which is twice as long as in the maneuver test in the previous chapter. The reason for the slow response of the implemented model is that the terms in the damping matrix D had to be increased in order to achieve stability. Without this scaling of the terms in the damping matrix, the ship will have a faster heading response, but becomes unstable for larger velocities. To cope with the problem of slow heading response, the gain in the heading controller can be increased. However, increasing it too much will cause oscillations in the heading angle. Also, the saturation limit had to be increased compared to the limit found in the maneuver test. From the yaw maneuvering test, it was found that the yaw moment peaked at 11 000 000 Nm. However, the saturation on yaw moment for the 3 DOF PSV is chosen to be almost twice as high, with a value of $2 \cdot 10^7$ Nm. With a saturation on moment in yaw significantly smaller than this, the response in yaw is too slow.

7.2 Performance of LOS guidance

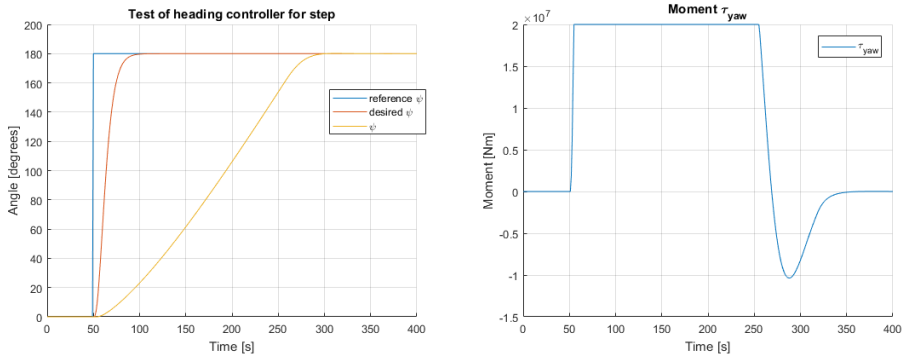
The LOS guidance method with waypoint switching is used to make the PSV in the PSV module follow a path consisting of straight-line segments. To check the performance of the LOS guidance, two waypoints at $(0, 0)$ and $(4000, 1000)$ are provided to the guidance system that defines the desired path to follow (x-axis points north and y-axis points east). The ship is moving with a constant velocity of 5 m/s, the initial position is set to (0, 500) and the initial heading angle is set to 0° (pointing north). The resulting trajectory can be



(a) Step response in heading

(b) Yaw moment

Figure 7.2: Step response of heading controller with a step reference from 0° to 45°.



(a) Step response in heading

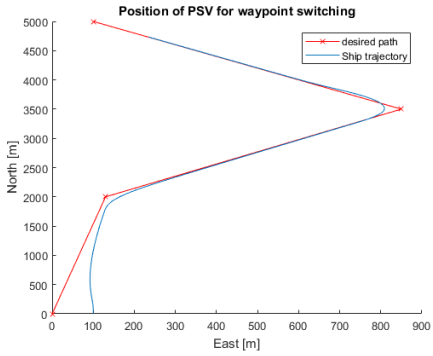
(b) yaw moment

Figure 7.3: Step response of heading controller with a step reference from 0° to 180°.

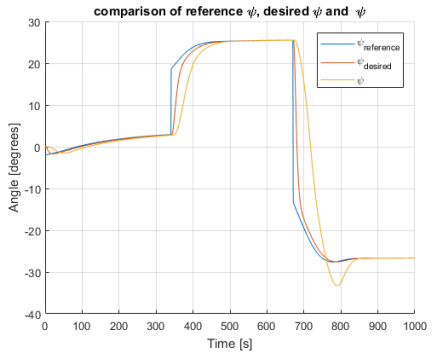
seen in Figure 7.5a and the resulting heading angle can be seen in Figure 7.5b. The ship is able to follow the straight path defined by the two waypoints, although it takes some time to converge to the path. The heading angle of the ship is able to follow the heading angle computed by the guidance system, $\psi_{reference}$, quite well. After the ship has passed the final waypoint, the ship will continue to move in the same direction.

Figure 7.4a shows the trajectory of the ship when trying to follow another path defined by the four waypoints $p_1 = (0, 0)$, $p_2 = (2000, 130)$, $p_3 = (3500, 850)$ and $p_4 = (5000, 100)$. The corresponding heading angle for the ship is shown in Figure 7.4b. The initial position of the ship is $(100, 0)$ with an initial heading angle of 0°. When switching to the last waypoint, it can be observed from Figure 7.4b that the heading angle has a slight overshoot, but is able to converge to the desired heading quite fast. During testing, it was found that the ship is able to follow paths with quite sharp turns as long as

the speed is low enough. If the speed is high, the ship has trouble to keep up with the sharp turns.

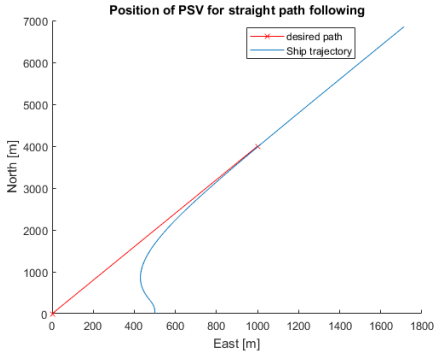


(a) Trajectory of ship

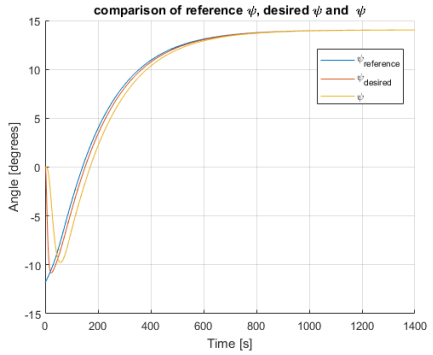


(b) Heading angle

Figure 7.4: LOS guidance performance for a path consisting of several waypoints.



(a) Trajectory of ship



(b) Heading angle

Figure 7.5: LOS guidance performance for straight path following.

The obstacle ships in the dynamic obstacles module are implemented the same way as the own-ship in the PSV module. Therefore, all the parameters for control systems, reference models and LOS guidance are the same. Figure 7.6 shows the LOS guidance performance of three obstacle ships. Initial values for position, heading and speed for the obstacle ship are presented in table 7.1.

As can be seen in Figure 7.6, all obstacle ships are able to follow their desired trajectory with acceptable accuracy. Even though the trajectories of the obstacle ships intersect each other, collisions are not considered in this case.

Parameter	Obstacle ship 1	Obstacle ship 2	Obstacle ship 3
x	700	0	100
y	0	500	2000
ψ	π	0	$-\frac{\pi}{2}$
u	3	3	3

Table 7.1: Parameter values for obstacle ships for LOS guidance testing.

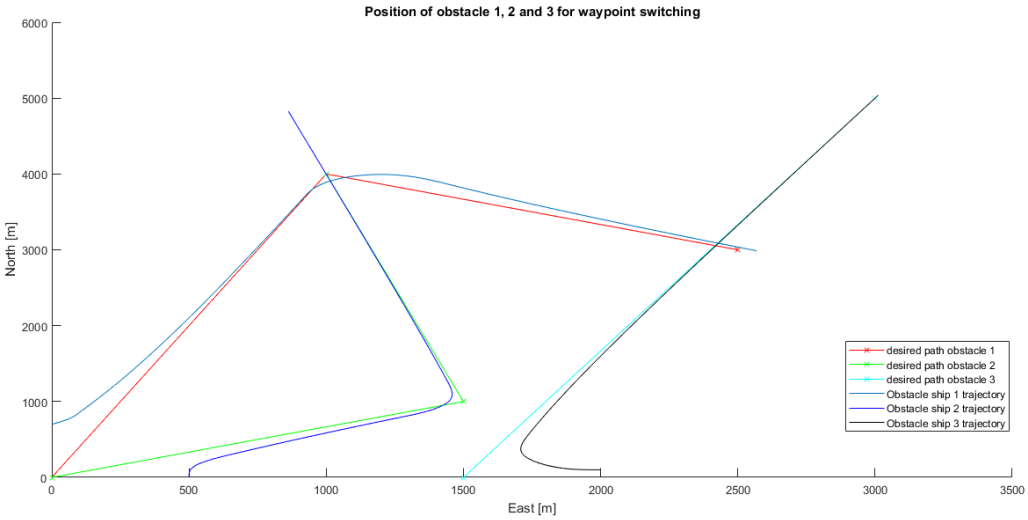


Figure 7.6: LOS guidance performance for three obstacle ships.

7.3 Ability to detect collision

The ability to detect collision between two ships and collision between a ship and a static obstacle is tested in three different scenarios. These scenarios are generated by using the two procedures discussed in chapter 6.6.

7.3.1 Scenario A

The first scenario is a head-on situation between the own-ship and an obstacle ship, as illustrated in Figure 7.7. The point of collision is $P_{\text{col}} = (500, 500)$, time to collision is $t_{\text{ttc}} = 400$ seconds and initial velocities for the own-ship and the obstacle ship are 3 m/s and 5 m/s respectively. Initial heading angles are chosen as $\psi_{\text{own-ship}} = 180°$ and $\psi_{\text{obstacle ship}} = 0°$. By using equation (6.13a) and (6.13b), the initial positions are calculated to be $P_{\text{own-ship}} = (1700, 500)$ and $P_{\text{obstacle ship}} = (-1500, 500)$. Both ships are steered

towards the point P_{col} by using LOS waypoint guidance.

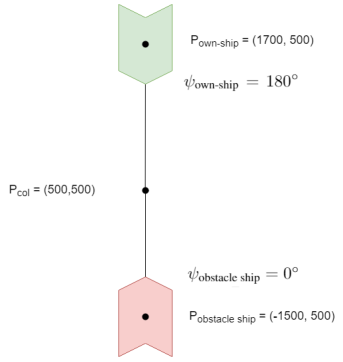


Figure 7.7: Scenario A: head-on situation between two ships.

The simulation results are shown in Figure 7.8. The collision between the ships is detected in the time interval $[387, 401]$ (the simulation continues for some time after the collision is detected).

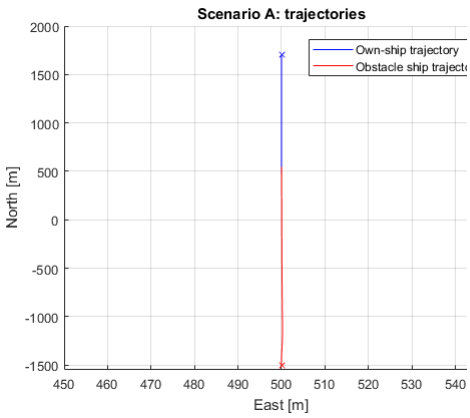
After $t = t_{\text{ttc}} = 400$ seconds, the positions of the own-ship and the obstacle ship are $(502, 500)$ and $(496, 500)$. Both these positions are close to the selected point of collision $P_{\text{col}} = (500, 500)$. The reason why the collision detection variable goes to zero at $t = 401$ seconds, when the ships' center positions are fairly close, is that the ship bows points in opposite directions. All in all, the simulation results show that the collision is detected correctly.

7.3.2 Scenario B

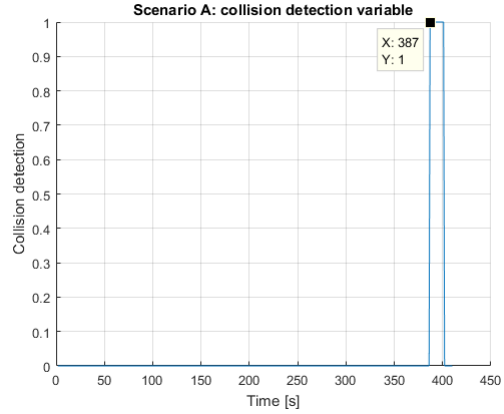
The second scenario is a bit more complex and aims to test scenario generation procedure for curved paths. The initial position of the own-ship is chosen to be $P_{\text{own-ship}} = (0, 500)$ and the curved path is defined by a set of waypoints as illustrated in Figure 7.9. The initial heading angles for the two ships are chosen to be $\psi_{\text{obstacle ship}} = -22.5°$ and $\psi_{\text{own-ship}} = 30°$.

The point of collision is $P_{\text{col}} = (1000, 2000)$. Time until collision can be calculated as $t_{\text{ttc}} = \frac{d}{u}$ where $d = 1997.2$ m is the total length of all straight line segments between the waypoints. Both ships have a speed of 3 m/s, which gives $t_{\text{ttc}} = 666$ s. Finally, the initial position for the obstacle ship is found from equations (6.13a) and (6.13b) to be $P_{\text{obstacle ship}} = (-845, 2764)$.

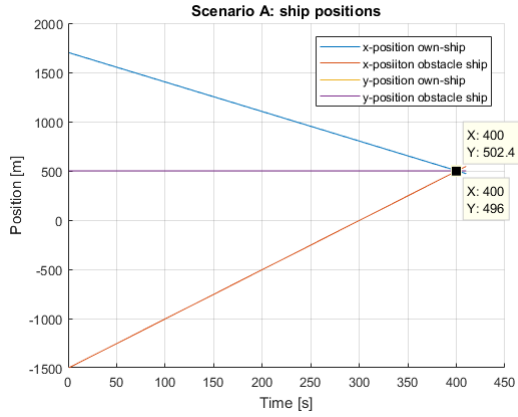
From the simulation results for scenario B in Figure 7.10 it can be seen that the collision detection variable has the value of one in the time interval $[641, 664]$. By calculations, the collision was estimated to happen after $t_{\text{ttc}} = 666$ seconds. At this point in time, the collision detection variable has a value of zero. According to calculations, the positions should



(a) Own-ship and obstacle ship trajectories.



(b) Collision detection variable (value is one when collision is detected).



(c) Position of ships.

Figure 7.8: Simulation results for scenario A.

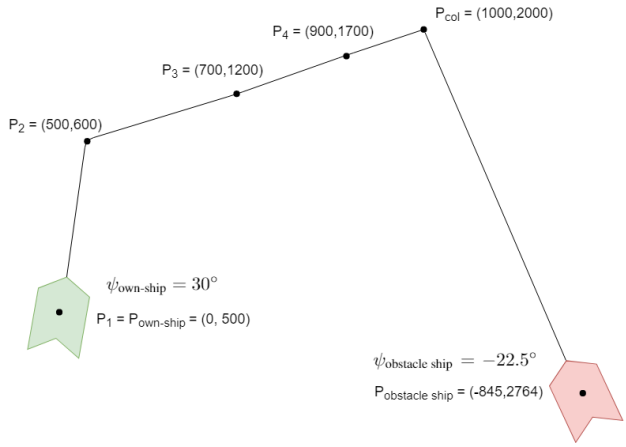


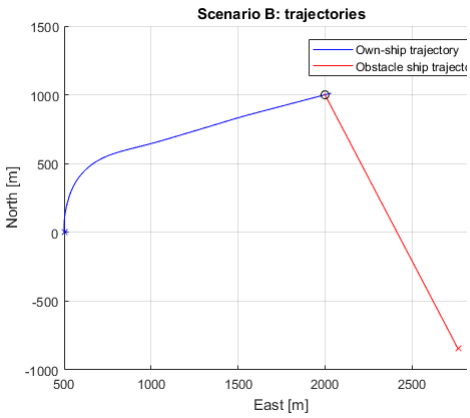
Figure 7.9: Scenario B: collision between an obstacle ship and the own-ship when own-ship follows a curved path.

be $(1000, 2000)$ for both ships after $t_{\text{ttc}} = 666$ seconds. At $t = 666$ seconds the position of the own-ship is $(1009, 2024)$ and the position of the obstacle ship is $(999, 2001)$. There is a slight error in the calculation for the position of the own-ship, due to the approximation made that the own-ship follows the straight line segments between the waypoints. Even though no collision is detected at $t = t_{\text{ttc}} = 666$ seconds, a collision is still detected relatively close to chosen collision point $P_{\text{col}} = (500, 500)$. The conclusion is that the collision detection ability in scenario B is satisfactory.

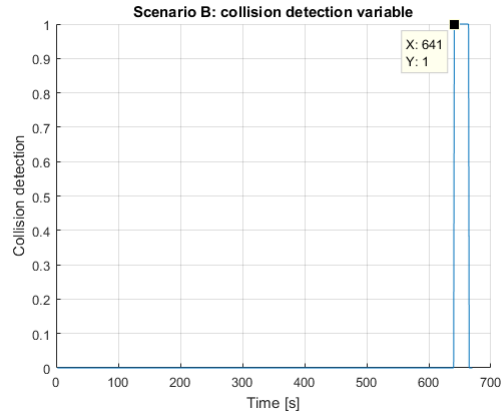
7.3.3 Scenario C

The third scenario includes two static obstacles, to check if collisions are detected between ships and static obstacles. In addition, this scenario has two obstacle ships and the own-ship. The situation is illustrated in Figure 7.11. The own-ship starts at $P_{\text{own-ship}} = (50, 500)$ with initial heading $\psi_{\text{own-ship}} = \frac{\pi}{6}$ and travels towards the circle obstacle located at $P_{\text{circle}} = (500, 700)$. Meanwhile, the first obstacle ship with initial position $P_{\text{obstacle ship 1}} = (400, 300)$ and initial heading $\psi_{\text{obstacle ship 1}} = \frac{-10\pi}{12}$ travels towards the boundary obstacle defined by the straight line between the points $P_{\text{start}} = (100, 50)$ and $P_{\text{end}} = (-300, 500)$. The second obstacle ship starting at $P_{\text{obstacle ship 2}} = (800, 0)$ with initial heading angle $\psi_{\text{obstacle ship 2}} = \frac{2\pi}{3}$ also travels towards the circle obstacle.

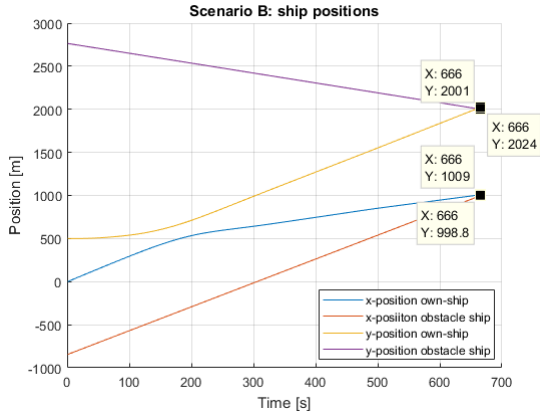
The simulation results for scenario C are shown in Figure 7.12. Collision between the own-ship and the circle obstacle, collision between obstacle ship 1 and the boundary obstacle and collision between obstacle ship 2 and the circle obstacle are detected after 39, 70 and 93 seconds respectively. The black points in the trajectory plot in Figure 7.12a show the position of each ship when a collision between the bow of the ship and the obstacle is detected. From this plot it is verified that collisions are detected correctly.



(a) Own-ship and obstacle ship trajectories.



(b) Collision detection variable (value is one when collision is detected).



(c) Position of ships.

Figure 7.10: Simulation results from scenario B.

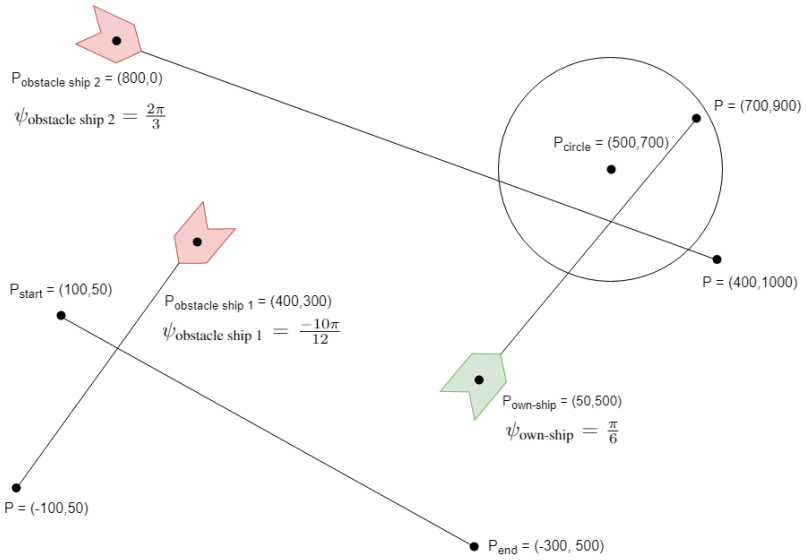
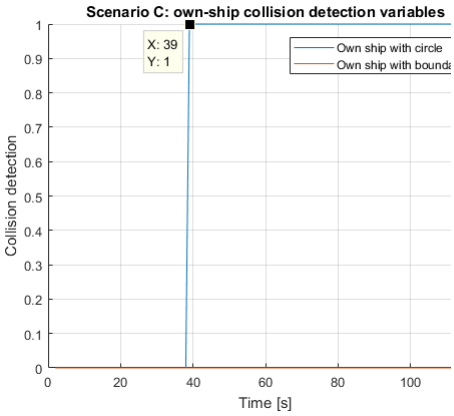
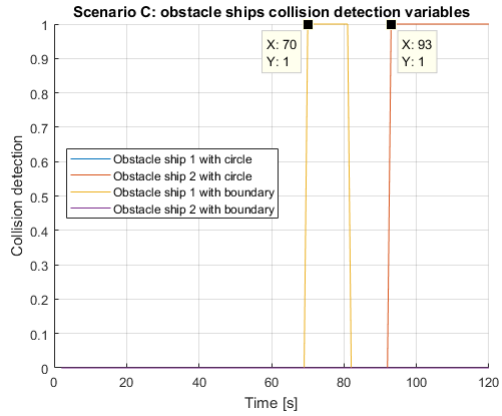


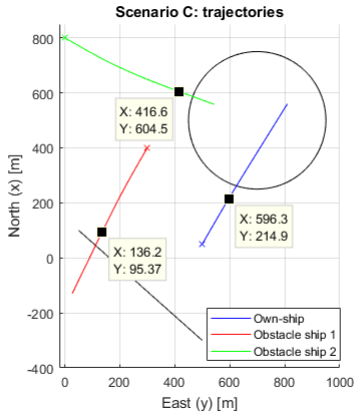
Figure 7.11: Scenario C: collision between ships and static obstacles.



(a) Collision detection variables for the own-ship.



(b) Collision detection variables for obstacle ships .



(c) Position of static obstacles and trajectories of ships.

Figure 7.12: Simulation results from scenario C.

Chapter 8

Conclusions

Collision statistics show that the majority of accidents in the shipping industry are caused by human errors, which motivates the need for a well-functioning collision avoidance system. The most common reason for collision at sea is that crew members do not pay enough attention to potential obstacles. Violation of COLREGs and failure of equipment are also factors that can lead to a collision.

An automatic collision avoidance system can help avoid collisions due to human errors. There is a vast amount of research regarding collision avoidance methods for ships that are able to achieve COLREGs compliance, including COLREG-RRT, Modified Virtual Force Field, Velocity Obstacles and Simulation-Based Model Predictive Control. Assuming that data about intentions is available, it is possible to extend these four methods to utilize other vessels' intentions, which has the potential to improve collision avoidance performance. However, in this project it has not been tested how the inclusion of intentions will actually affect the performance.

This project proposed an initial design of a collision avoidance system utilizing other vessels' intentions based on the SBMPC method from Johansen et al. (2016). Assumptions and requirements for such an algorithm were defined. The initial design focused on how to include intentions into the existing SBMPC algorithm. Intention data such as planned future route, deviation from future route, the role of the obstacle ship in a collision situation (give-way or stand-on) in addition to real-time data from AIS were used. Inclusion of intention data was used to modify the hazard computation in the SBMPC algorithm.

The developed simulator in this project is able to simulate the behavior of a PSV, by using a feedback linearizing controller to steer the heading angle and a proportional (P) controller to steer the speed. LOS guidance was also implemented to make the ship able to follow a path defined by waypoints. The developed simulator also includes functional-

ity for adding obstacle ships and static obstacles. For simplicity, the obstacle ships have the same parameters and control system as the own-ship and the static obstacles represent environmental restrictions such as shores and islands. Simulation results showed that the simulator gives acceptable performance for LOS guidance, detection of collision and control of heading and speed. The main drawback of the developed simulator was the slow response in change of heading angle, which caused a big deviation between the desired heading angle and the actual heading angle. In addition, some unrealistic, simplifying assumptions such as no environmental disturbances and perfectly measured states were made.

Chapter 9

Future work

The development of the simulator in this preliminary project lays the foundation for further work in the master's thesis, where the simulator will be extended to include an implementation of a collision avoidance algorithm that utilizes other vessels' intentions. The initial design of such an algorithm was discussed in chapter 5, but will be further developed in the master's thesis. The work that remains on the design of the modified SBMPC algorithm includes deciding how data about intentions should be transmitted, decide suitable parameter values and finding more ways that intentions could be utilized.

Future work will focus on how the inclusion of other vessel's intentions affects the performance of collision avoidance. Different scenarios must be designed and the performance of the original SBMPC method will be compared to the modified SBMPC method where intentions are utilized. Therefore, a suitable performance measure must be found.

The current implementation of the simulator does not consider environmental disturbances. In the future, testing the developed simulator with the presence of environmental disturbances will give insights on how well the control system and LOS guidance perform under more realistic circumstances. In addition, it was assumed that sensor measurements are perfect without any noise. Inclusion of measurement noise can be added to see how it will affect performance. The main problem encountered with the simulator was slow response in changing of heading angle which led to bad tracking of the desired heading angle. Since the implementation of a collision avoidance algorithm depends on a well-functioning control system, some effort can be put into solving this problem.

With the current implementation, each scenario has to be set up manually by selecting values for several parameters such as the position of static obstacles and trajectory for ships. Creating new scenarios is, therefore, a time consuming process and it would be beneficial to find a mechanism to automatically create scenarios.

Bibliography

Allianz Global Corporate & Specialty, 2017. Global claims review (technical report) .

Bitar, G., Eriksen, B.H., Lekkas, A.M., Breivik, M., 2019. Energy-Optimized Hybrid Collision Avoidance for ASVs, in: 2019 18th European Control Conference (ECC), pp. 2522–2529. doi:10.23919/ECC.2019.8795645.

Bousson, K., 2008. Model predictive control approach to global air collision avoidance. Aircraft Engineering and Aerospace Technology 80, 605–612. doi:10.1108/00022660810911545.

Børhaug, E., Pavlov, A., Pettersen, K.Y., 2008. Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents, in: 2008 47th IEEE Conference on Decision and Control, pp. 4984–4991. doi:10.1109/CDC.2008.4739352.

Campbell, S., Naeem, W., Irwin, G., 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. Annual Reviews in Control 36, 267 – 283. doi:https://doi.org/10.1016/j.arcontrol.2012.09.008.

Chen, L., Hopman, H., Negenborn, R.R., 2018. Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. Transportation Research Part C: Emerging Technologies 92, 101 – 118. doi:https://doi.org/10.1016/j.trc.2018.04.013.

Chiang, H.L., Tapia, L., 2018. COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation. IEEE Robotics and Automation Letters 3, 2024–2031. doi:10.1109/LRA.2018.2801881.

European Maritime Safety Agency, 2018. Annual overview of marine casualties and incidents 2018 (technical report) .

Fiorini, P., Shiller, Z., 1998. Motion Planning in Dynamic Environments Using Velocity Obstacles. The International Journal of Robotics Research 17, 760–772. doi:10.1177/027836499801700706.

Foss, B., Heirung, T.A.N., 2016. Merging Optimization and Control. (A note used in the course TTK4135 at the Norwegian University of Science and Technology).

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. Wiley & sons, Ltd.

Henriksen, E.S., 2018. Automatic testing of maritime collision avoidance methods with sensor fusion. Master's thesis, Norwegian University of Science and Technology .

Hwang, C.N., Yang, J., Chiang, C.Y., 2001. The design of fuzzy collision-avoidance expert system implemented by H infinity - autopilot. Journal of Marine Science and Technology 9, 25–37.

International Maritime Organization, 1972. 1972 Convention on the International Regulations for Preventing Collisions at Sea .

International Maritime Organization website, 2017. Lessons learned english. URL: http://www.imo.org/en/OurWork/MSAS/Casualties/Pages/Lessons-learned.aspx.

Ørnulf Jan Rødseth, Burmeister, H.C., 2012. Developments toward the unmanned ship (published as part of the MUNIN project) .

Johansen, T.A., Perez, T., Cristofaro, A., 2016. Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment. IEEE Transactions on Intelligent Transportation Systems 17, 3407–3422. doi:10.1109/TITS.2016.2551780.

Keller, M., Haß, C., Seewald, A., Bertram, T., 2015. A Model Predictive Approach to Emergency Maneuvers in Critical Traffic Situations, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 369–374. doi:10.1109/ITSC.2015.69.

Kuwata, Y., Wolf, M.T., Zarzhitsky, D., Huntsberger, T.L., 2014. Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles. IEEE Journal of Oceanic Engineering 39, 110–119. doi:10.1109/JOE.2013.2254214.

Lee, S.M., Kwon, K.Y., Joh, J., 2004. A fuzzy logic for autonomous navigation of marine vehicle satisfying COLREG guidelines. International Journal of Control, Automation, and Systems 2.

Li, S., Liu, J., Negenborn, R.R., 2019. Distributed coordination for collision avoidance of multiple ships considering ship maneuverability. Ocean Engineering 181, 212 – 226. doi:https://doi.org/10.1016/j.oceaneng.2019.03.054.

Liu, J., Jayakumar, P., Stein, J.L., Ersal, T., 2015. An MPC Algorithm with Combined Speed and Steering Control for Obstacle Avoidance in Autonomous Ground Vehicles.

Lloyd, M., 2006. Why ships really collide. SEAWAYS October 2006 , 10–11.

Loe, A., 2008. Collision Avoidance for Unmanned Surface Vehicles. Master's thesis, Norwegian University of Science and Technology .

Mariano, K., 2017. Remembering Doña Paz, The Deadliest Shipwreck In History Worse Than The Titanic. URL: https://www.elitereaders.com/remembering-dona-paz-deadliest-shipwreck-history-worse-titanic/.

MATLAB, 2017. version 9.2.0.538062 (R2017a). The MathWorks Inc., Natick, Massachusetts.

Minne, P.K.E., 2017. Automatic testing of maritime collision avoidance algorithms. Master's thesis, Norwegian University of Science and Technology .

Moe, S., Pettersen, K.Y., 2017. Set-Based line-of-sight (LOS) path following with collision avoidance for underactuated unmanned surface vessels under the influence of ocean currents, in: 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 241–248. doi:10.1109/CCTA.2017.8062470.

MUNIN, 2016. Autonomous systems on the rise. URL: http://www.unmanned-ship.org/munin/about/.

Murray, B., Perera, L.P., 2018. A Data-Driven Approach to Vessel Trajectory Prediction for Safe Autonomous Ship Operations, in: 2018 Thirteenth International Conference on Digital Information Management (ICDIM), pp. 240–247. doi:10.1109/ICDIM.2018.8847003.

Nocedal, J., Wright, S.J., 2006. Numerical optimization. Springer-Verlag New York.

O'Brien, C., 2018. Autonomous ships. Standard P&I Club's Technology Bulletin, September edition , 2.

Porathe, T., Borup, O., Jeong, J., Park, J., Camre, D., Brodje, A., 2014. Ship traffic management route exchange: acceptance in Korea and Sweden, a cross cultural study.

Rolls-Royce, 2017. Autonomous ships The next step (technical report) URL: https://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf.

Saracco, R., 2017. Autonomous systems on the rise. URL: https://cmte.ieee.org/futuredirections/2017/09/09/autonomous-systems-on-the-rise/.

Statheros, T., H.G..M.K., 2008. Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques. Journal of Navigation 61, 129–142.

Stenersen, T.F., 2015. Guidance System for Autonomous Surface Vehicles. Master's thesis, Norwegian University of Science and Technology .

STM Balt Safe, 2019. About STM BALT SAFE Project. URL: https://www.stmvalidation.eu/projects/stm-balt-safe/about-stm-balt-safe-project/.

Szlapczynska, J., 2015. Data Acquisition in a Manoeuver Auto-negotiation System. TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation 9, 343–348. doi:10.12716/1001.09.03.06.

Tam, C. K., B.R., Greig, A., 2009. Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. Journal of Navigation 62, 455–476.

Tam, C., Bucknall, R., 2013. Cooperative path planning algorithm for marine surface vessels. Ocean Engineering 57, 25 – 33. doi:https://doi.org/10.1016/j.oceaneng.2012.09.003.

U.S Department of Homeland Security, U.S. Coast Guard, 2014. Navigation Rules and Regulations Handbook .

Vanem, E., Eide, M.S., Gravir, G., Lepsøe, A., 2008. ECDIS AND ENC COVERAGE – FOLLOW UP STUDY, in: Technical report by DNV research Innovation.

Wingrove, M., 2018. ACCIDENT REPORT: Communications breakdown causes ship collision. URL: https://www.rivieramm.com/news-content-hub/accident-report-communications-breakdown-causes-ship-collision-24959.

Woerner, K., 2016. Multi-contact protocol-constrained collision avoidance for autonomous marine vehicles. Ph.D thesis, Massachusetts Institute of Technology .

Wöhrn, R., 2007. Collisions at sea - Unavoidable? URL: http://www.gard.no/web/updates/content/51705/collisions-at-sea-unavoidable.

Xu, T., Liu, X., Yang, X., 2011. Ship Trajectory Online Prediction Based on BP Neural Network Algorithm, in: 2011 International Conference of Information Technology, Computer Engineering and Management Sciences, pp. 103–106. doi:10.1109/ICM.2011.288.

Xue, Y., Lee, B., Han, D., 2009. Automatic collision avoidance of ships. Proceedings of The Institution of Mechanical Engineers Part M-journal of Engineering for The Maritime Environment 223, 33–46. doi:10.1243/14750902JEME123.

Ziarati, R., Ziarati, M., 2007. Review of Accidents with Special References to Vessels with Automated Systems – A Way Forward .