Master's thesis

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

Andreas Steinsmo

MPC-based and COLREGs-aware Trajectory Planning and Collision Avoidance for both Low-Speed and High-Speed ASVs

Master's thesis in Cybernetics and Robotics
Supervisor: Morten Breivik
Co-supervisor: Emil Hjelseth Thyri

June 2023

NTNU
Norwegian University of
Science and Technology

Andreas Steinsmo

MPC-based and COLREGs-aware Trajectory Planning and Collision Avoidance for both Low-Speed and High-Speed ASVs

Master's thesis in Cybernetics and Robotics
Supervisor: Morten Breivik
Co-supervisor: Emil Hjelseth Thyri
June 2023

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

NTNU
Norwegian University of
Science and Technology

Abstract

This masters thesis presents two approaches for mid-level trajectory planning for maritime autonomous vessels that avoids collision with dynamic and static obstacles, in a way that complies with a subset of the Convention on the International Regulations for Preventing Collisions at Sea (COLREG). One of the approaches is intended for fully actuated autonomous passenger ferries, where the transit is performed at low speeds. The other approach is intended for underactuated high-speed vessels, where the maneuverability of the vessel is limited compared to the fully actuated passenger ferry, which needs to be considered when planning a trajectory.

Both trajectory planners are based on Model Predictive Control (MPC), and differ in the models used internally by the MPC algorithm and the objective function to be minimized. A subset of the COLREGs is considered relevant for the task of trajectory planning, and the constraints of the MPC algorithm are formulated such that the optimal trajectory complies with the chosen subset of the COLREGs.

As part of this work, two tools are introduced in order to incentivize different types of behaviour during the planning horizon of the trajectory planners. The first tool is windows of reduced cost in the planning horizon of the MPC, and the idea presented and implemented in this work generalizes to other MPC-based trajectory-planning applications, as well as other MPC-applications where different types of behaviour is deemed useful in different parts of the planning horizon. The second tool is a set of constraints that enforce the relevant subset of the COLREGs, where the tool proposed in this work is a specific linear constraint placed on the port side of target ships, in order to improve the compliance with some of the COLREGs compared to previous work. Both tools are intended to incentivize the trajectory planner to plan early and readily apparent maneuvers to avoid collision with other vessels, in compliance with Rule 8 of the COLREGs.

The trajectory planners and the proposed tools are tested through simulations, where a model of the milliAmpere1 passenger ferry is used for low-speed transit simulations, and a scaled model of the Otter USV is used for high-speed transit simulations. Through an extensive simulation study using relevant performance metrics, the trajectory planners are shown to plan trajectories that to a large degree complies with the chosen subset of the COLREGs, while also avoiding collision with all static and dynamic obstacles. Results from the simulations also shows that introducing windows of reduced cost in the planning horizon of the MPC-based trajectory planners effectively makes the trajectory planners plan earlier and more apparent action to avoid collision. The port-side constraint is also seen to incentivize the planners to plan more apparent action in single-vessel encounters and to more robustly enforce the vessel to maneuver to the correct side in compliance with the COLREGs. This constraint is however shown to need more work if the tool is to be used in the general sense including static obstacles and multiple target ships. While these tools are shown to effectively enforce the relevant set of rules, it is also shown through a set of metrics that the increased COLREGs-compliance comes at the cost of increased control error, power consumption, actuator usage and acceleration.

Sammendrag

Denne masteroppgaven presenterer to fremgangsmåter for mellom-nivå baneplanlegging for maritime autonome fartøy som unngår kollisjon med dynamiske og statiske hindringer, på en måte som delvis følger trafikkreglene til sjøs gitt av Convention on the International Regulations for Preventing Collision at Sea (COLREG). En av fremgangsmåtene er rettet mot fullt aktuerte, autonome passasjerferger, hvor fartøyene har lav fart. Den andre fremgangsmåten er rettet mot underaktuerte fartøy med høyere fart, hvor manøvrerbarheten til fartøyet er begrenset sammenlignet med fullt aktuerte passasjerferger, en faktor som må tas hensyn til når fartøyets bane planlegges.

Begge baneplanleggerne er basert på Model Predictive Control (MPC), og bruker forskjellige modeller internt i MPC-algoritmen, samt forskjellige objektivfunksjoner som skal minimeres av planleggeren. En del av COLREG-reglene er ansett som relevant når det skal planlegges bane, og begrensningene lagt til i MPC-algoritmen er formulert på en slik måte at den optimale banen følger de valgte reglene.

Som en del av dette arbeid er to verktøy introdusert for å insentivere forskjellige typer adferd underveis i planleggingshorisonten til baneplanleggerne. Det første verktøyet er vinduer med redusert kostnad i planleggingshorisonten til MPCen, og ideen presentert og implementert i dette arbeidet generaliserer til andre MPC-baserte baneplanleggingsapplikasjoner, samt andre MPC-applikasjoner hvor det kan være nyttig å ha forskjellige typer adferd i forskjellige deler av planleggingshorisonten. Det andre verktøyet er et sett med begrensninger som kan brukes til å forbedre COLREGs-kompitabiliteten til baneplanleggerene, og begrensningen som er introdusert i dette arbeidet er en spesifikk begrensning plassert på babord side av målfartøyet, som forbedrer ytelsen til baneplanleggerene sammenlignet med tidligere arbeid. Begge verktøyene er ment å insentivere baneplanleggerne til å planlegge tidlige og tydelige manøvre for å unngå kollisjon med andre fartøy på en måte som følger Regel 8 i COLREG-reglene.

De foreslåtte baneplanleggerne og verktøyene er testet gjennom simuleringer, hvor en modell av passasjerfergen milliAmpere1 er brukt for lav-hastighets simuleringer, samt en skalert modell av Otter USVen for høy-hastighets simuleringer. Gjennom en omfattende simuleringsstudie og bruk av relevante metrikker er baneplanleggerne vist å planlegge baner som i stor grad følger det valgte settet med regler fra COLREG, mens man også unngår kollisjoner med alle statiske og dynamiske hindringer. Simuleringsresultater viser også at bruk av kostreduksjonsvinduer effektivt fører til at baneplanleggerne planlegger tidligere og tydeligere manøvre for å unngå kollisjon. Begrensningen plassert på babord side av målfartøyet er også vist å føre til at tydeligere manøvre blir planlagt av baneplanleggerne i én-til-én møter, samtidig som den også fører til at riktig manøvreringsside for eget fartøy blir valgt på en mer robust måte. Denne begrensningen er derimot også vist å trenge mer jobb hvis verktøyet skal kunne brukes mer generelt med både statiske og flere dynamiske hindringer. Begge verktøyene er også gjennom bruk av relevante metrikker vist å føre til høyere reguleringsfeil i reguleringssystemet, høyere energiforbruk av fartøyet, økt bruk av aktuatorer samt høyere akselerasjon.

Preface

This work concludes my M.Sc degree in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). During this work I have been given the opportunity of working with topics regarding autonomous vessels, a field I find great interest in and look forward to working with after my studies have ended. The work has been challenging, but all in all, I look back at it as the most rewarding period of my studies.

First, I would like to express my gratitude towards my supervisor Morten Breivik and my co-supervisor Emil H. Thyri. Throughout working with this masters thesis, you have always been available for feedback and help when needed. Feeding on your knowledge through the fall of 2022 during my specialization project thesis and this spring for my masters thesis has been highly educational, motivating and inspiring.

I would also like to thank Jon Bernhard Høstmark for your input on the more practical aspects of navigating at sea. Due to my lack of knowledge within the more practical aspects of this field, the conversation we had regarding these topics has been helpful, and has worked as great inspiration when working with the topics in this work.

I would like to thank my girlfriend, Kristin, for always standing by my side and cheering me on during my years studying at NTNU. The studies have in no way been easy, and having you by my side has been essential for me to be able to finally deliver this masters thesis. I am grateful for the time you have spent listening to me ramble on about control systems, optimization, Kalman filters, the intricacies of the COLREGs and everything regarding making autonomous vessels not collide.

Last but not least, I would like to thank my family, Hans Martin, Bodil, Hannah and Eirik, as well as my in-laws Bjørn Ingar and Wenche. Your continued support and motivation during my studies is highly appreciated, and will not be forgotten.

- Throughout this semester I have had bi-weekly meetings with Morten Breivik and Emil H. Thyri, where I have gotten feedback on the work and suggestions for the way forward. In addition to the bi-weekly meetings, Emil has also helped me with questions regarding the simulator used in the work.

- This work is done in cooperation with Zeabuz, through having Emil H. Thyri as my co-supervisor. The work builds upon the work done by Emil in his PhD thesis. As part of this, I was at the beginning of the semester given access to relevant code in Zeabuz' codebase, including a simulator that included functionality for testing collision-avoidance algorithms, as well as a model of the milliAmpere1 ferry used in this work. This code also included the basis of the trajectory planner that was further developed in the work presented in this thesis.

Andreas Steinsmo
Trondheim, June 12, 2023

Table of Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

NTNU	=	Norwegian University of Science and Technology
COLREGs	=	International Regulations for Preventing Collisions at Sea
VHF	=	Very High Frequency
COLAV	=	Collision Avoidance
AIS	=	Automatic Identification System
ASV	=	Autonomous Surface Vessel
USV	=	Uncrewed Surface Vessel
MPC	=	Model Predictive Control
DNSD	=	Dynamic Navigation Safety Domain
SBG	=	Set-Based Guidance
OS	=	Ownship
TS	=	Target Ship
MPCC	=	Model Predictive Contouring Control
DP	=	Dynamic Positioning
NED	=	North-East-Down
DOF	=	Degrees Of Freedom
GNC	=	Guidance, Navigation and Control
CG	=	Center of Gravity
CO	=	Coordinate Origin
CB	=	Center of Buoyancy
CF	=	Center of Flotation
LOS	=	Line Of Sight
NLP	=	Nonlinear Program
OCP	=	Optimal Control Problem
OT	=	Overtaking
HO	=	Head-on
GW	=	Give-way
SO	=	Stand-on
SF	=	Safe
RBS	=	Relative Bearing Sector
SS	=	Situation Sector
IAEW	=	Integral of absolute value of the error multiplied by the energy consumption
IADC	=	Integral of absolute differentiated control
IAEW-WT	=	Integral of the absolute value of the error multiplied by the energy consumption, wear and tear
IAA	=	Integral of Absolute Acceleration
CPA	=	Closest Point of Approach
NMPC	=	Nonlinear Model Predictive Control
DCPA	=	Distance at Closest Point of Approach
CRW	=	Cost-reduction Window

Chapter 1

Introduction

This chapter provides the motivation for this work, a description of the considered problem, a description of the software provided by Zeabuz, the contributions from the work, and finally an outline for the rest of the thesis.

1.1 Motivation

Autonomous systems are being developed and deployed in several fields today, and the maritime industry is no exception. The motivation for making maritime vessels such as urban passenger ferries or smaller survey vessels autonomous includes reduced cost for the companies employing these services, as well as an increasing shortage of qualified crews working on maritime vessels. Autonomous systems can be designed to optimize their performance with respect to different objectives, without having to compromise on the safety of the passengers using the systems. Some objectives that can be optimized for autonomous vessels at sea are the distance travelled by vessels, the energy consumed by doing so in terms of e.g. actuator usage and the time used to reach the destination [20].

During the fall of 2022, the Norwegian University of Science and Technology (NTNU) together with the spin-off company Zeabuz performed trial operations with the autonomous passenger ferry milliAmpere2 [3]. The ferry autonomously shuttled passengers back and forth across a canal in Trondheim for three weeks [1]. Trial operations such as this shows the potential of autonomous systems within the maritime industry, and gives developers of such systems key insight needed to further develop and improve the systems, enabling autonomy technology. An image from the trial operations in Trondheim is shown in Figure 1.1.

On June 8th, 2023, the first autonomous passenger ferry for commercial use was launched in Stockholm, Sweden. The passenger ferry named MF Estelle is intended to run 15 hours per day between Kungsholmen and Söder Mälarstrand, with a capacity of

[1] Video from the opening ceremony of the milliAmpere2 trial operations in Trondheim: https://www.youtube.com/watch?v=j3v47HiJmos (Accessed 03/04/2023).

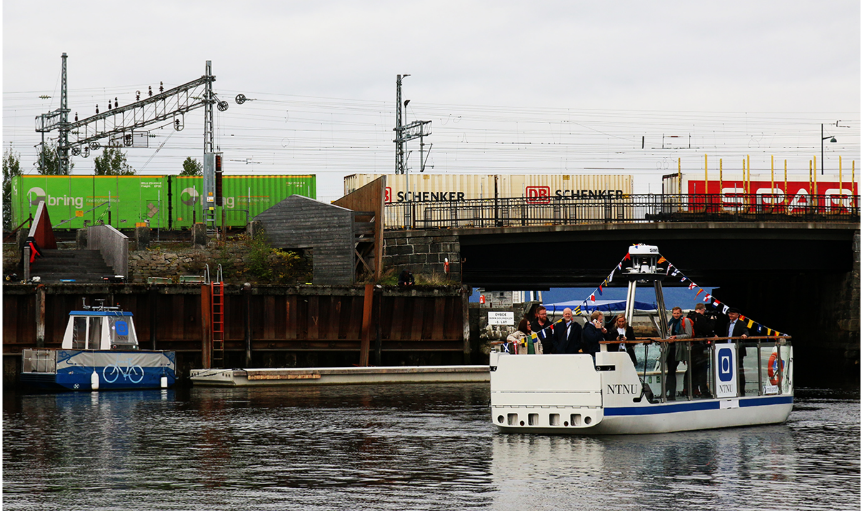


Figure 1.1: Image taken from the opening day (September 21, 2022) of the trial operations performed by NTNU and Zeabuz during the fall of 2022 in Trondheim. The autonomous ferry milliAmpere2 can be seen to the right, and its smaller cousin milliAmpere1 can be seen to the left. Courtesy of [3].

25 passengers [21]. The transport service provider Torghatten and the technology supplier Zeabuz are behind the project, where Zeabuz delivers the autonomy system onboard the passenger ferry. This commercial operation is a key milestone in the development of autonomous maritime systems. The ferry is designed by the Brødrene Aa shipyard, and is shown in Figure 1.2.

Along with new possibilities, maritime autonomy also poses new challenges that must be handled. Autonomous systems need to be trusted by the people using the systems, in order for people to be willing to let e.g. an autonomous ferry transfer them across a canal in the absence of a captain. In order to achieve this trust in autonomous systems, the systems need to be proven safe. Trial operations like the one mentioned above are an important means to achieve this, together with well thought-out solutions by the engineers and designers in charge of developing these systems.

Designing safe and efficient autonomous systems, whether it be marine vessels or cars, requires the systems to comply with the set of rules that apply in their respective operational domain. For vessels maneuvering at sea, this common rule set is the International Regulations for Preventing Collisions at Sea (COLREGs). One challenge with implementing a COLREGs-compliant system on a computer is that good seamanship also plays an important role in order to ensure safety for marine vessels. Good seamanship comes from experience, and the definition of good seamanship varies based on the context and the area of operation. What is deemed a safe action for a vessel maneuvering at open sea might not be safe in a densely populated harbour or a narrow canal in an urban environment.

Another tool frequently used by manned vessels at sea is Very High Frequency (VHF) radio communication between vessels. Many situations can be resolved early through

(a) The autonomous passenger ferry MF Estelle, designed by Brødrene Aa. Courtesy of Brødrene Aa [21].

(b) MF Estelle on water the day of the its launch, June 8th, 2023. Courtesy of Morten Breivik.

Figure 1.2: Images of the autonomous passenger ferry slated for commercial operation in Stockholm, Sweden, starting June 12th, 2023.

communication between the vessels involved in the situation, making VHF an important tool in the context of collision avoidance (COLAV). In the absence of VHF communication onboard autonomous vessels, the importance of adhering to the COLREGs is further underlined. Resolving situations through VHF communication is however not problem free, making the need for a common set of rules important. In Marine Guidance Note (MGN) 324 published by the Maritime and Coastguard Agency, some guidelines for the use of VHF and Automatic Identification System (AIS) are laid out [22]. The note states that the use of VHF radio under some circumstances might not be helpful, and may even prove to be dangerous. An example mentioned in the note is a collision between a bulk carrier and a container ship in 2014, where the navigating officers relied solely on VHF communication to avoid collision. This lead to maneuvers that violated the COLREGs, which further lead to a collision between the vessels. One of the complicating factors in this particular situation according to the note was that the VHF communication was not conducted in English, which was both of the ships' working language. Examples such as this show why the COLREGs serves as an essential foundation in order to avoid collision at sea.

At its core, a computer will do exactly what you tell it to. This makes it hard to implement good seamanship as part of an autonomous system. The motivation for part of this work is to transfer some aspects of what is good seamanship into the world of autonomous marine systems. It must however be underlined that the COLREGs are an essential foundation for autonomous systems at sea. Especially in the absence of VHF communication with other vessels and a captain with years of navigational experience, the COLREGs serves as a minimum to be followed by all autonomous vessels maneuvering at sea.

1.2 Previous work

This section presents literature and previous work reviewed in order to get a better understanding of the problem at hand, as well as the landscape of the field of maritime autonomy and COLREGs-compliant COLAV. Section 1.2.1 is mostly copied from the preceding specialization project thesis [20].

1.2.1 Autoferry project

NTNU is currently working on nine digital transformation projects, of which one of the projects is called ”Autonomous all-electric passenger ferries for urban water transport”, or just Autoferry [23]. The main goal of the Autoferry project is to develop methods for transporting people on autonomous passenger ferries in urban water channels [24]. The project centers around eight PhD positions within the following fields:

1. Automation and autonomy
2. Multi-sensor tracking via shore- and ferry-based sensors
3. All-electric power and propulsion
4. Human factors, remote monitoring and control
5. Communications and cyber security
6. Risk management

The work presented in this thesis builds upon work done within the first of the fields above, namely automation and autonomy.

Two previous PhD candidates that have done work within the automation and autonomy field in the Autoferry project are Emil H. Thyri and Glenn Bitar [4][25]. Both candidates have presented work within the field of trajectory planning/tracking and COLAV for Autonomous Surface Vessels (ASVs).

In his PhD thesis, Emil H. Thyri focuses on trajectory planning and COLAV for ASVs that adhere to the COLREGs [4]. The COLREGs are the ”rules of the road” at sea, and set out navigation rules to be followed by vessels maneuvering at sea, with the purpose of avoiding collisions [26]. Among Emil’s contributions are an improvement of an algorithm used for encounter classification, a novel domain for dynamic obstacles and a trajectory planning algorithm for ASVs operating in confined areas with traffic [4]. Thyri has also been the co-supervisor during this work, and this work is based on and builds upon his work.

Glenn Bitar also focuses on optimization-based trajectory planning in his PhD thesis, including undocking, transit and docking. Among Glenn’s contributions are an energy-optimized trajectory planner, a method for automatic docking of an ASV as well as work done on NTNU’s experimental platform for autonomous passenger ferries, milliAmpere1.

One commonality between the theses is a three-layered hybrid architecture used for maritime COLAV, inspired by an architecture proposed by Eriksen and Breivik [27] among others. An example of such a hybrid structure is shown in Figure 1.3. Such hybrid structures are common in literature regarding COLAV at sea, and this work is also based on a

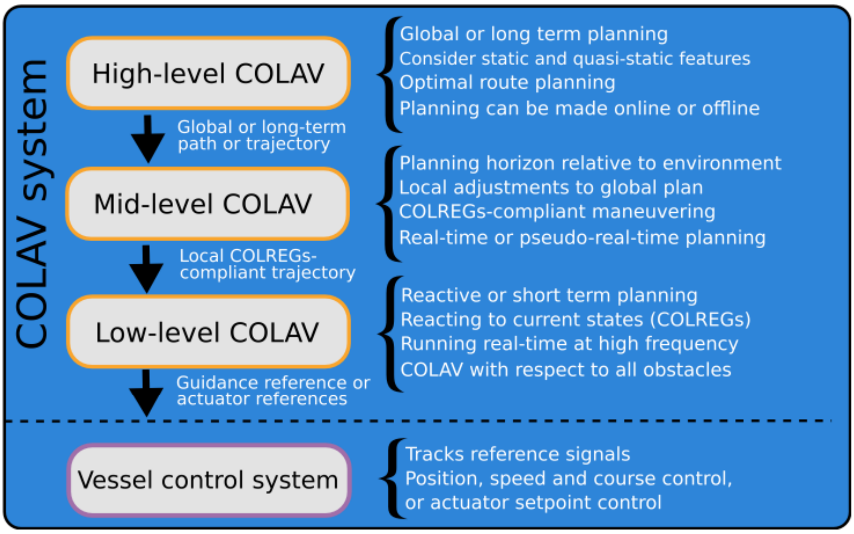

COLAV system
High-level COLAV
Global or long-term path or trajectory
Mid-level COLAV
Local COLREGs-compliant trajectory
Low-level COLAV
Guidance reference or actuator references
Vessel control system
Global or long term planning
Consider static and quasi-static features
Optimal route planning
Planning can be made online or offline
Planning horizon relative to environment
Local adjustments to global plan
COLREGs-compliant maneuvering
Real-time or pseudo-real-time planning
Reactive or short term planning
Reacting to current states (COLREGs)
Running real-time at high frequency
COLAV with respect to all obstacles
Tracks reference signals
Position, speed and course control,
or actuator setpoint control

Figure 1.3: Three-layered hybrid COLAV structure. Courtesy of Emil H. Thyri [4].

layered architecture. The reason for using a hybrid structure like this is because the task of planning how to maneuver a vessel from A to B is a very complex one. The planning often considers a large set of objectives, e.g. collision avoidance, minimizing risk, energy efficiency, passenger safety and COLREGs-compliance to name a few. It is hard to consider all these objectives in one monolithic algorithm, leading to a hybrid structure where the objectives are distributed in a layered structure. These layers are also structured with a decreasing planning horizon as one moves from the high-level to the low-level layers. There is also a difference in algorithm runtime demands across the layers. For the high-level layer a low runtime is not crucial, at least not within reasonable bounds. For the mid-level and low-level layers, the algorithms should on the other hand be quite fast to run. As an example, the low-level layer should have a runtime comparable to the dynamics of the vessel. The mid-level layer should on the other hand have a runtime that is comparable to the dynamics of the surrounding environment.

1.2.2 Maritime COLAV

This work has a focus on COLREGs-compliant, collision-free trajectory planning at sea. A challenge with the COLREGs is that it is written by humans for humans. This leads to a qualitative formulation of the rules that relies on human interpretation, and it is therefore not trivial to translate these rules to quantitative algorithms and lines of code as part of the software responsible for the trajectory planning. This can also be seen in literature regarding maritime COLAV, since much of the literature does not handle the COLREGs. As the field of maritime COLAV is a practical problem in that the algorithms are to be used in a real-world setting, the algorithms need to adhere to the COLREGs in order to be fully deployed and trusted in a real-world environment.

A survey on autonomous COLAV at sea is performed in [28]. In this survey, 48 pub-

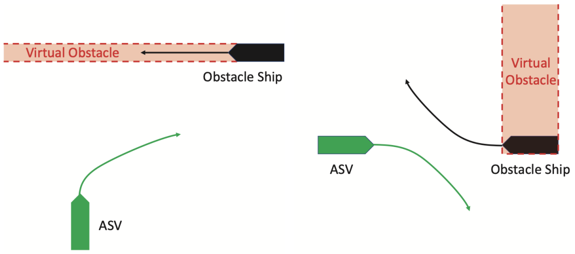


Figure 1.4: Virtual obstacles used to enforce COLREGs-compliance The ownship is shown in green, and the target ship is shown in black. Courtesy of [5].

lications were filtered out of more than 150 publications in order to evaluate the scientific landscape of autonomous COLAV at sea. Some remarks concluded by the authors are:

1. Most approaches in the literature only address a small subset of the COLREGs rules.
2. Approaches mostly handle navigation in open sea, and narrow areas like canals etc. are not often handled.
3. Most approaches are tested through simulations only, and real-life practical experiments are not often performed.

The authors of the survey also underline the difficulty in recommending appropriate algorithms for COLREGs-compliant COLAV algorithms, as there are no apparent common performance benchmarks to assess the performance of the different algorithms.

Several ways to approach the problem of planning COLREGs-compliant trajectories/paths are described in the literature, from standard path planning algorithms such as A*, Rapidly-exploring Random Trees (RRT), etc. to more advanced algorithms such as Model Predictive Control (MPC).

One algorithm based on RRT, COLREG-RRT, is described in [5]. This algorithm aims to handle COLREGs-compliance by extending target vessels in a way that forces the RRT algorithm to plan paths that are COLREGs-compliant. This way of extending the obstacles is referred to as Virtual Obstacles, and is shown in Figure 1.4. One potential problem with extending obstacles in this way is that one could end up restricting the search space of the algorithm too much. This might not be a problem in single vessel encounters, but when multiple vessels are considered the search space might get too restrictive. This is also potentially a problem in situations where there are multiple encounters with conflicting rules.

Another way of approaching the problem is presented in [6], where the main contribution is a ship domain referred to as the Dynamic Navigation Ship Domain (DNSD). In this article, domains for both the ownship and the target ship are defined, and the idea is to switch between an obstacle avoidance mode and a path following mode. The switch between the modes is performed based on whether the domains of the ownship and target ship overlap or not. The DNSD proposed in the article is shown in Figure 1.5. The domain lengths $\mathbf{R}_f$ and $\mathbf{R}_s$ are based on the length, width, and the relative speed

$$\Delta U = U_{OS} - U_{TS}, \tag{1.1}$$

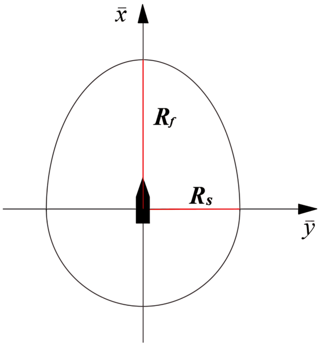


Figure 1.5: Dynamic Navigation Ship Domain proposed by [6]. Courtesy of [6].

where U_{OS} and U_{TS} are the speed of the ownship (OS) and target ship (TS), respectively. In the context of this work, the ownship is the vessel that is to follow the trajectory planned by the trajectory planners, and target ships are surrounding vessels that the ownship is observing and shall avoid colliding with. The tactical diameter is also taken into account when defining the domain, which represents the transverse distance of the vessel when turning 180° from the start of steering [6]. In this way the domains of the ownship and target ship take their respective dynamics and maneuverability into account before planning an action, which is generally preferable when planning paths in a maritime domain. Contrary to e.g. cars that often are in the same order of magnitude when it comes to dynamic responses, different kinds of vessels come in highly varying sizes and with varying capabilities when it comes to maneuverability. Planning paths based on this domain is then done based on the Set-Based-Guidance (SBG) method described in [29]. This method is based on defining a few key waypoints around the target ship domain for the ownship to follow, while ensuring the resulting path is COLREGs-compliant. One disadvantage of methods based on setting waypoints to be followed by a path-following scheme is that this does not directly involve any planning of the speed along the path. Often this will involve traversing along the path at a constant speed, which might not always be preferable. In scenarios where action is urgent, a larger acceleration might be preferable in order to stay clear of danger. In order to to plan both the geometrical path as well as the velocity profile along the path, algorithms such as MPC can be preferable.

MPC is a frequently used algorithm within path/trajectory planning for autonomous systems. A COLREGs-aware trajectory planner is presented in [7], and is based on Model Predictive Contouring Control (MPCC), which is a specific MPC formulation. The trajectory planner works by defining a set of constraints in the plane, and as long as these constraints are satisfied the resulting trajectory should adhere to a subset of the COLREGs. The constraints proposed by [7] are shown in Figure 1.6 for a give-way situation where the ownship is to make a starboard turn and pass the target ship behind its stern in order to adhere to the COLREGs. One disadvantage of defining these constraints is the amount of parameters that go into defining one constraint, as shown in Figure 1.6. The size of the square about the target ship (with vertices pb, ps, sb, ss) has to be defined. This is done in [7] by utilizing the target ship's (estimated) length and width and a *footprint radius* ρ,

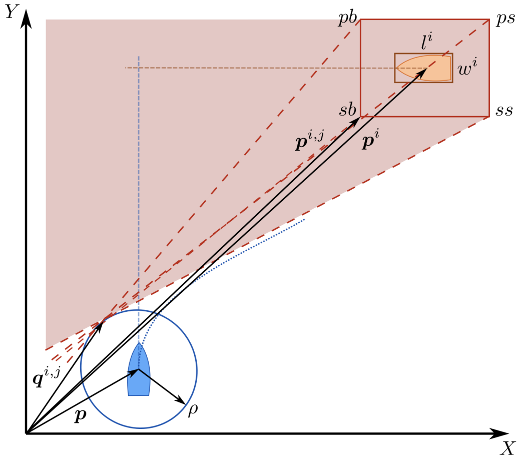


Figure 1.6: Constraints fed to the MPC controller in [7]. The blue vessel is the ownship, the orange vessel is the target ship and the red region denotes the region in which the ownship is not allowed to enter in order to adhere to the COLREGs. The situation shown is a situation in which the ownship is to make a starboard turn, and pass the target ship behind its stern according to the COLREGs. Courtesy of [7].

which is the same ρ used as radius of the circle about the ownship. The anti-clockwise tangent lines from each of these vertices to the circle about the ownship are then used as constraints. One problem with this approach is that one might again risk restricting the space in which the MPC is allowed to plan a trajectory too much, especially in situations where there are multiple TSs or encounters with conflicting rules. Another problem might occur as a result of the constraints in the plane being defined by estimated data such as the width and length, especially if AIS not used. If such data is provided by the situational awareness onboard the vessel, this data comes with uncertainties from the different sensors used. This will however be a problem in most path planning algorithms, as paths can only be calculated based on the perceived environment of operation. The MPC algorithm is described in detail in Section 2.7.3.

As seen in this section, there are several ways to tackle the problem of COLREGs-compliant trajectory planning for autonomous vessels, each with their own strengths and weaknesses. A commonality between the different works mentioned above is the lack of a common way to assess the quality of the algorithms. This makes it hard to compare the performance of the methods based on their respective results. The methods are also most often tested through simulations, making it hard to assess the applicability of the methods in a real-life environment involving disturbing external factors and uncertain data from the situational awareness modules responsible for feeding data to the path/trajectory planning algorithms. This is in line with the findings of the mentioned survey [28].

The difficulty in assessing the quality of the algorithms through a common set of performance metrics is seldom handled in literature regarding maritime COLAV, at least when it comes to assessing the COLREGs-compliance of the algorithms. Defining performance metrics for assessing COLREGs-compliance is however not trivial, as the COLREGs are written by humans, for humans, and a rule can be interpreted differently by different people. Despite the challenges with interpreting these rules, some efforts have been made to develop suitable metrics for some of the rules. In [30] a framework for testing collision avoidance algorithms is developed. The masters thesis is written in cooperation with DNV, and some of the metrics defined and implemented in the work are able to detect the safety of an algorithm during encounters, whether a maneuver is performed in ample time, and if the maneuver is readily apparent, to mention some of the metrics. These metrics are based on the work presented in the PhD thesis written by Kyle Woerner [31], which is also partly the inspiration for a part written in [32] about the evaluation of safety and COLREGs-compliance. A step towards a standardized way to evaluate COLREGs-compliant autonomous COLAV algorithms is also presented in [33], where a set of metrics and algorithms for quantifying COLREGs-compliance is defined.

1.3 Problem description

The purpose of this work is to research, develop and evaluate a mid-level trajectory planner for ASVs operating in areas with traffic. The trajectory planner must comply with a selected subset of the COLREGs, and must plan trajectories that are suitable for vessels with a varying degree of actuation and operating speed. The trajectory planner can be seen as a part of a three-layered, hybrid structure for trajectory planning frequently used in literature regarding trajectory planning and COLAV at sea. The mid-level trajectory planner shall produce COLREGs-compliant, dynamically feasible trajectories that are collision-free with static, quasi-static and dynamic obstacles. Specifically, the objectives of this work are:

- Research existing algorithms for trajectory planning in the context of autonomous vessels.
- Implement and test a trajectory planner aimed towards autonomous passenger transfer at low speed.
- Implement and test a trajectory planner aimed towards underactuated, high-speed vessels.
- Test the implemented methods based on simulations, using relevant performance metrics in order to get a clear picture of the advantages and disadvantages of the methods.
- The implemented methods should show a good trade-off between efficiency and safety, where safety is of highest priority.

1.4 Software platform from Zeabuz

A simulator was provided by Zeabuz through Emil H. Thyri at the start of the work, hereinafter referred to as the ZeabuzSimulator. At the time the ZeabuzSimulator was provided, it contained the following key functionality:

- A vessel model of the milliAmpere1 passenger ferry and a dynamic positioning (DP) controller for the ferry.
- An interpolation module responsible for calculating the reference signals for the controllers used in this work by interpolating the optimal trajectory from the proposed MPC-based trajectory planners.
- A simulator framework that facilitates testing of COLAV algorithms. This simulator makes it possible to simulate the milliAmpere1 vessel model, as well as simulating target ships by simulating first-order Nomoto models using LOS guidance and speed and heading control. This makes it possible to set up vessel-to-vessel encounter scenarios, in order to test the COLAV algorithms.
- A target tracking module responsible for estimating the trajectory of target ships. The estimation of target ship trajectories is based on the assumption that a target ship will keep a constant speed and course, based on the estimated speed and course.
- An encounter classification module that classifies encounters based on methodology described in Section 2.8.
- The basis of the trajectory planners implemented in this work. This trajectory planner included a target ship domain described in Section 3.7, the optimal control problem described in Section 3.4 and the linear model and the objective function described in Section 3.4.1.
- The ability to simulate a first-order Nomoto model in order to produce the basis of the reference trajectory described in Section 3.5.

1.5 Contributions

The contributions from this work are summarized as follows:

- A literature study on existing algorithms for COLREGs-compliant trajectory planning and COLAV.
- A study on the challenges regarding COLREGs-compliance in the context of autonomous vessels.
- Implementation of two tools that can be utilized in the context of MPC-based trajectory planning. The first tool is windows of reduced cost in the planning horizon of the MPC, and the second tool is a linear constraint placed on the port side of target ships. Both tools are intended to increase the trajectory planners' compliance with Rule 8 and Rule 16 of the COLREGs.
- Development and implementation of a trajectory planner intended for underactuated, high-speed vessels in order to assess how MPC-based trajectory planning transfers to such vessels.

- Improvements of the ZeabuzSimulator are implemented, facilitating further development of the trajectory planners presented in this work. These improvements are mainly:
  - The addition of a scaled Otter USV model together with a guidance and control scheme facilitating further development of trajectory planners for underactuated, high-speed vessels.
  - The addition of the ability to simulate scenarios involving static obstacles in the form of land using geospatial data describing the area of interest.
- Simulation-based testing and evaluation of the different methods, using relevant performance metrics in order to assess the COLREGs-compliance of the methods, as well as the energy consumption, actuator wear-and-tear, control system performance and absolute acceleration of the vessel.

The subset of the COLREGs considered in this work are mainly Rule 8 and Rules 13-17.

1.6 Outline

Chapter 2 presents the background theory for the work presented in this thesis. Chapter 3 presents the trajectory planning algorithms and the main contributions of this work. Simulation results and discussions are presented in Chapter 4, while Chapter 5 presents the conclusions drawn from this work, and suggests further work to be done.

Chapter 2

Background theory

This chapter presents background theory for the work presented in this thesis.

2.1 COLREGs

The COLREGs set out the traffic rules to be followed when maneuvering at sea, similarly to the traffic rules that are followed when driving a car. The rules are divided into six parts, together forming a total of 41 rules to be followed by vessels maneuvering at sea [26]. The six parts are as follows:

- Part A: General
- Part B: Steering and Sailing
- Part C: Lights and Shapes
- Part D: Sound and Light signals
- Part E: Exemptions
- Part F: Verification of compliance with the provisions of the Convention

The part most relevant for the task of path/trajectory planning is part B (Rules 4-19), which among other elements handles how vessels should conduct themselves in the presence of other vessels, e.g. in case of an encounter. Another part relevant to collision avoidance is Part C (Rules 20-31), which handles lights and shapes that vessels shall display. This section presents the rules most relevant for the task of trajectory planning, and is based on [34]. After the relevant rules are described, some of the challenges with implementing COLREGs-compliant autonomous systems is discussed in Section 2.1.1.

Rule 6: Safe speed

Rule 6 states that all vessels shall at all times proceed at a safe speed, such that proper and effective action can be taken at all times. The term ”safe speed” is here a relative term, and depends on the situation at hand. Rule 6 states that determining a safe speed depends on visibility, traffic density and the maneuverability of the ownship.

Rule 7: Risk of collision

Rule 7 states that all vessels shall use all available means to determine the risk of collision, and that risk should be deemed to exist if there is any doubt. For an autonomous vessel maneuvering without a crew onboard, the assessment of risk is highly dependent on a well functioning situational awareness system onboard the vessel.

Rule 8: Action to avoid collision

Rule 8 states that actions to avoid collision shall be positive, made in ample time and with due regard to the observance of good seamanship. The rule also states that any alteration of course and/or speed shall, if the circumstances admit, be large enough to be readily apparent to another vessel observing visually or by radar. A succession of small changes in course and/or speed should be avoided according to this rule. The rule also states that with sufficient sea room, alteration of course alone may be the most effective action to avoid close-quarters situations.

Rule 9: Narrow channels

Rule 9 is relevant for this work as it deals with vessels maneuvering in narrow channels, as is the case for a typical passenger ferry transferring passengers in an urban environment. The rule states that vessels maneuvering along narrow channels shall keep near to the outer limit of the channel on starboard side as possible. The rule also states that smaller vessels ($< 20m$ long) shall not impede the passage of vessels that can only navigate safely within a narrow channel.

Rule 13: Overtaking situation

Rule 13 handles situations where one vessel is overtaking another. The rule states that any vessel overtaking another vessel shall keep out of the way of the vessel being overtaken. It also classifies an overtaking vessel as a vessel that comes up on another vessels side from a direction more than 22.5 degrees behind its beam. When there is doubt whether a vessel is overtaking another, the rule states that the overtaking vessel shall assume this is the case and act accordingly. Finally, Rule 13 states that any subsequent alteration of the bearing between the vessels shall not make the overtaking vessel a crossing vessel. An overtaking situation is shown in Figure 2.1.

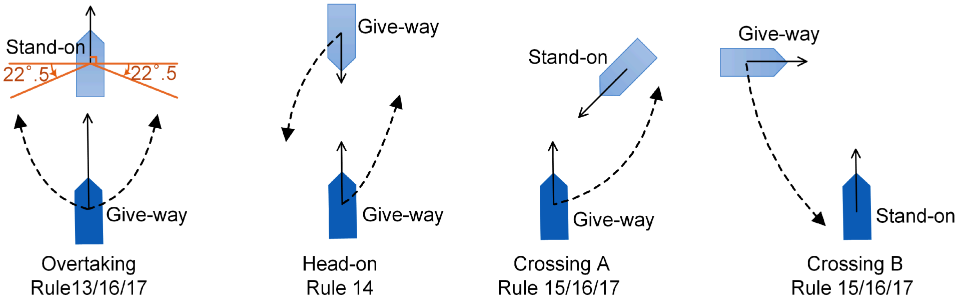


Figure 2.1: COLREGs rules 13-17 visualization. Courtesy of [8].

Rule 14: Head-on situation

Rule 14 handles head-on situations, i.e. situations where vessels are approaching each other bow-to-bow. The rule states that two vessels meeting in a head-on situation with the risk of collision shall alter their course to starboard, meaning they the vessels shall pass on the port side of each other. A head-on situation and a COLREGs-compliant maneuver is shown in Figure 2.1.

Rule 15: Crossing situation

Rule 15 handles crossing situations, i.e. situations as depicted in the two right-most illustrations in Figure 2.1. The rule states that when two power-driven vessels are crossing, the vessel that has the other vessel on its starboard side shall keep out of way and avoid crossing ahead of the other vessel.

Rule 16: Action by give-way vessel

Rule 16 handles vessels that has a give-way role, i.e. a role in which it is directed to keep out of the way of another vessel. The rule states that a give-way vessel shall to the best of its ability take early and substantial action to keep well clear of the other vessel.

Rule 17: Action by stand-on vessel

Rule 17 handles vessels that has a stand-on role, meaning a role in which the other vessel in an encounter is to keep out of the stand-on vessels way. The rule states that the stand-on vessel shall keep its course and speed as long as the give-way vessel keeps out of the stand-on vessels way. The rule does however state that the stand-on vessel might need to alter its speed or course to avoid collision if the give-way vessel does not seem to keep out of way in compliance with the rules.

Rule 21: Definitions (lights)

Rule 21 defines the different lights that shall be displayed by a vessel at sea. All lights do not apply to all kinds of vessels, and which lights shall be applied by the different

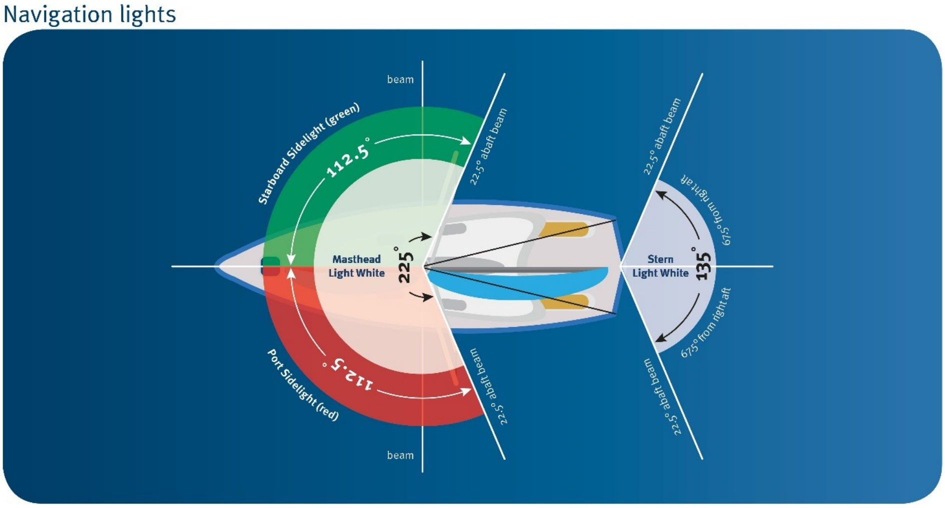


Figure 2.2: Navigation lights according to COLREGs Rule 21. Courtesy of [9].

kinds of vessels are described in the rest of the rules in part C. Specifically, Rule 23 states that a power-driven vessel underway shall exhibit a masthead light forward, sidelights and a sternlight. A second masthead light shall be displayed by vessels longer than 50 meters, which is out of the scope of this work. Rule 20 of part C also states that the lights mentioned above shall be complied with from sunset to sunrise, and from sunrise to sunset in restricted visibility and in other circumstances when deemed necessary.

The masthead light is according to Rule 21 a white light that is placed over the fore and aft centreline of the vessel. The light shall show an unbroken light over an arc of the horizon of 225 degrees, and shall be fixed such that the light is shown from right ahead to 22.5 degrees abaft the beam on either side of the vessel.

The sidelights refer to a green and red light that are mounted on each side of the vessel. The green light shall be mounted on the starboard side of the vessel and the red light shall be mounted on the left side of the vessel. Each light shall show an unbroken light over 112.5 degrees, from right ahead to 22.5 degrees abaft of the beam on their respective sides.

The sternlight is a white light mounted on the stern of the vessel. This light shall show an unbroken light of 135 degrees, such that the light covers 67.5 degrees from right aft of the vessel on both sides.

The lights defined above are visualized in Figure 2.2. Especially the sidelights are useful for a vessel to understand the intent of another vessel better, especially after sunset or when the visibility is restricted. If two vessels are e.g. in a head-on situation like shown in Figure 2.1, the vessels shall see each others red light. For unmanned vessels these lights are useful for showing the vessels intentions, which is important in encounters where unmanned vessels meets manned vessels. Without clear information about the unmanned vessels intentions, the people in control of the manned vessel might have difficulties when deciding their own actions, and unwanted situations might occur as a result.

2.1.1 COLREGs challenges

As seen in Rule 8 regarding safe speed, terms such as *ample time*, *good seamanship* and *readily apparent* are introduced. As discussed in [18], these are terms that are hard to translate to lines of code to be executed by a machine. What is ample time will depend on the speed of the vessels involved in the situation, the size of the vessels as well as the maneuverable space surrounding the vessels. For an autonomous vessel it is also hard to implement good seamanship into its autonomy system, as good seamanship arguably requires years of experience navigating at sea.

What is good seamanship might also vary based on where the vessel is maneuvering, as different ports and areas around the world might have different local rules of conduct. This is handled by Rule 1 (b) of the COLREGs, which is given in [34] as

> Rule 1 (b): *Nothing in these Rules shall interfere with the operation of special rules made by an appropriate authority for roadsteads, harbours, rivers, lakes or inland waterways connected with the high seas and navigable by seagoing vessels. Such special rules shall conform as closely as possible to these Rules.*

While the COLREGs takes such local rulings into account, this further complicates the development of autonomous vessels. As seen above, the rule states that such special rules shall conform closely to the COLREGs, but knowing the special rules of an area might still be essential when developing the autonomy system for a vessel that is to maneuver in that area.

Although not listed in the previous section, Rule 2 of the COLREGs is an interesting, and often misunderstood rule. Rule 2 is split in two parts, (a) and (b), and is given in [34] as

> Rule 2 (a): *Nothing in these Rules shall exonerate any vessel, or the owner, master or crew thereof, from the consequences of any neglect to comply with these Rules or of the neglect of any precaution which may be required by the ordinary practice of seamen, or by the special circumstances of the case.*
>
> Rule 2 (b): *In construing and complying with these Rules due regard shall be had to all dangers of navigation and collision and to any special circumstances, including the limitations of the vessels involved, which may make a departure from these Rules necessary to avoid immediate danger.*

This rule essentially opens up for the possibility of the need to make a departure from the rest of the COLREGs, and introduces an important term: *the ordinary practice of seamen*. In [10], Captain Rajeev Jassal argues that this term is an attempt to fill any gaps in the COLREGs, and can be translated to *common sense*. In the article an interesting example is described, involving an overtaking situation illustrated in Figure 2.3. In this situation both vessels are nearing a turn in a narrow canal, and the ownship has a higher speed than the target ship. If the ownship is to perform an overtaking in this situation, it will have two alternatives according to Rule 13; overtaking the target ship on its port side or overtaking it on its starboard side. The vessels are however nearing a turn of the canal, meaning common sense (or the ordinary practice of seamen) suggests the target ship shall be overtaken on its port side as the target ship is likely to make a starboard turn to complete

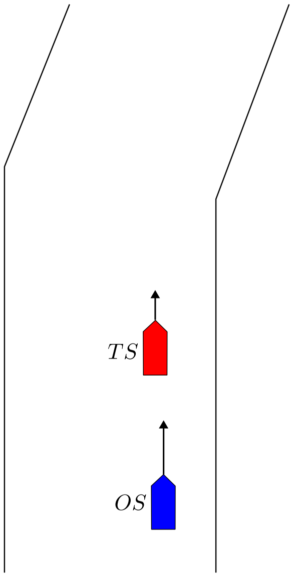


Figure 2.3: Example of an overtaking situation, where the ownship is shown in blue, and the target ship is shown in red. Inspired by [10].

the turn. Rule 13 does however state that an overtaking can be performed on both sides. If the ownship is to overtake the target ship on its starboard side in this situation it will therefore comply with Rule 13, but as [10] argues, the ownship might still be charged with a violation of the COLREGs if a collision is to happen as a result of this COLREGs-compliant maneuver. This is because the ownship has violated Rule, 2 as it has failed to comply with the ordinary practice of seamen. Situations like this shows how complicated the COLREGs are to interpret in the context of autonomous systems. Implementing the common sense needed to resolve the situation above is difficult to do on a computer, as the computer will only do what the lines of code tell it to.

Another (more complex) situation that involves conflicting rules is illustrated in Figure 2.4, inspired by a scenario described in a seminar about COLREGs-compliant COLAV [11]. In this example there are mainly two rules of the COLREGs that apply. The target ship TS_1 is a crossing vessel, that is also significantly larger than the ownship. This makes the ownship a give-way vessel according to Rule 15 regarding crossing situations, meaning the ownship shall avoid crossing ahead of the other vessel according to Rule 15, and take early and substantial action to keep well clear according to Rule 16 regarding give-way vessels. On the other hand, the target ship TS_2 is an overtaking vessel, making the ownship a stand-on vessel according to Rule 13, as the ownship shall avoid making TS_2 a crossing vessel. According to Rule 17 regarding stand-on vessels, the ownship shall then keep its speed and course. If the ownship is to comply with the rules imposed by TS_2, it will effectively breach the rules imposed by TS_1 and vice versa. Good seamanship might indicate a good solution w.r.t. TS_1 and TS_2 is for the ownship to slacken its speed in order to let TS_1 cross ahead, and TS_2 overtake the ownship on its starboard side. There

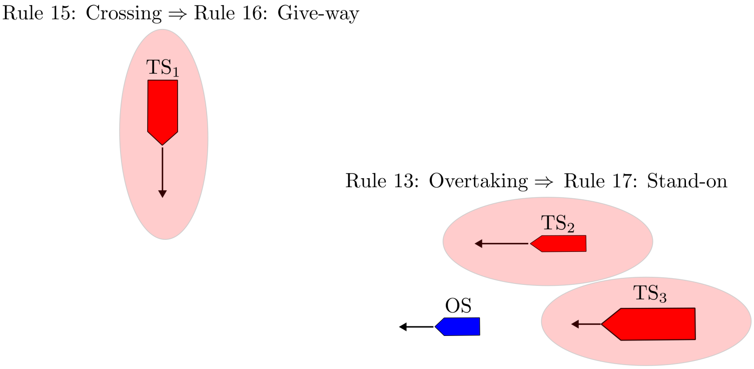

Rule 15: Crossing ⇒ Rule 16: Give-way
TS₁
Rule 13: Overtaking ⇒ Rule 17: Stand-on
TS₂
TS₃
OS

Figure 2.4: Example of conflicting COLREGs rules. Here the ownship is shown in blue, and the three target ships TS_1, TS_2 and TS_3 are shown in red. The arrows indicate the directions of the vessels, and the magnitude of the arrows are an indication of their respective speeds. Inspired by Øystein Engelhardtsen from DNV [11].

is however a significantly larger vessel, TS_3, abaft the ownship. This vessel makes it dangerous for the ownship to slacken its speed, and once again leads to the ownship having to keep its speed. The ownship could also increase its speed and turn to its port side, in order to cross ahead of TS_1, effectively avoiding the potential encounters with TS_2 and TS_3. This would however violate Rule 15, and potentially also Rule 2. The author of this work does not know any solution to this situation, but it makes for an interesting discussion, once again showing that the COLREGs doesn't handle all possible situations that one might encounter at sea. In this particular situation, the solution would however likely involve breaching some of the rules of the COLREGs, and possibly adhering to Rule 2 through going for a solution that complies with *the ordinary practice of seamen*.

Based on the situations described above, Rule 2 of the COLREGs in sorts becomes the *one rule to rule them all*, and introduces multiple challenges when it comes to dealing with violations of the COLREGs by a court. If a vessel e.g. adheres to Rule 13 in the situation illustrated in Figure 2.3 and this leads to a collision, the vessel has complied with one rule of the COLREGs, while violating another rule, namely Rule 2. The question then becomes; What rule is highest ranked in the hierarchy of the COLREGs? Is Rule 2 higher up in the hierarchy than e.g. encounter-specific rules such as Rules 13-15? It is in such situations up to the individual court to place the responsibility of the collision on the correct vessel. It will then in practice end up in a *tug of war* between Rule 2 and the other rules in situations like described above, if they were to lead to a collision involving the ownship. If the practice of good seamanship (Rule 2) is to be weighted higher than a particular rule (e.g. Rule 13), this makes it harder to develop autonomous vessels, as this is the part that is the hardest to translate into lines of code. It might seem like this is the case, as written in [34]:

> *If a departure from the Rules is necessary to avoid immediate danger, a vessel would not only be justified in departing from them, but may be expected to do so.*

It is also worth mentioning that where Rule 2 allows for a departure of the COLREGs in certain situations, Rule 2 is itself a rule of the COLREGs, meaning a departure from the COLREGs is not made in practice.

The challenges surrounding Rule 2 are further complicated by the fact that a violation of the COLREGs can be fined, according to §55 of the Norwegian Ship Safety and Security Act (Skipssikkerhetsloven) [35]. According to §55, violations of the COLREGs can be fined based on the violation alone, meaning a collision is not needed. This complicates the development of autonomous systems at sea, for example by looking at the situation in Figure 2.4. If the ownship in this figure violates a subset of the rules in order to avoid collision in accordance with Rule 2, is there a possibility of subsequently being fined despite of a collision having been avoided?

When looking through the COLREGs, another problem that seems evident is that the rules are to a large degree written with respect to single-vessel encounters. In most situations where there are more than one target ship, it is likely possible to assess the situation as a series of single-vessel encounters, as is done in this work. This assumption holds well in open sea, but less in narrow waters. The assumption also does not hold when assessing situations like described earlier in this section, as these situations leads to multiple encounters that need to be assessed and handled at once. Situations like these are not to the author of this works knowledge handled by the COLREGs. It is of course difficult to handle all kinds of situations and possible edge-cases in a set of rules like the COLREGs, but it does nevertheless pose challenges when developing autonomous systems. It is arguably fair to ask the question if the COLREGs as they stand today are even suited for development of autonomous systems. This question is asked in [36], where the vagueness of the COLREGs is pointed out as one of the main challenges for the development of COLAV algorithms for autonomous vessels.

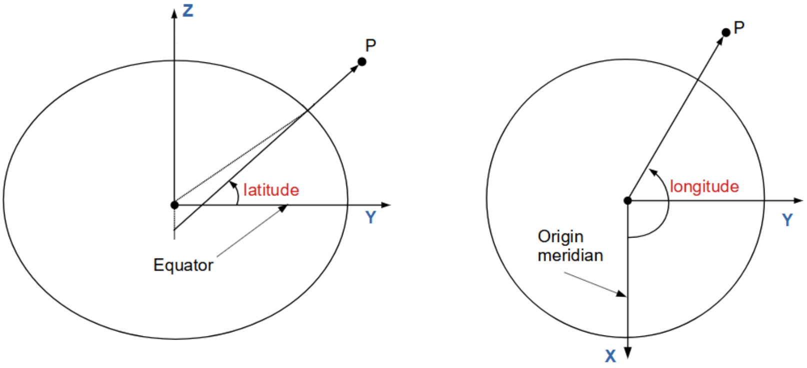


Figure 2.5: Geographic coordinate frame. Courtesy of SBG Systems [12].

2.2 Coordinate frames

In order to analyze and describe the motion of a vessel on the water surface, multiple coordinate systems needs to be utilized. In this thesis, there will be a focus on three reference frames, namely the geographic reference frame, the north-east-down (NED) reference frame and the vessels body-fixed reference frame. The theory on these coordinate frames is based on theory presented in [1], and is copied from the specialization project thesis preceding this work [20].

2.2.1 Geographic reference frame

The geographic reference frame describes a point on the surface of the Earth by longitude-latitude pairs (l, μ). The longitude is the angle of the point in the equatorial plane, and the latitude is the angle in the meridian plane. The coordinate system is visualized in Figure 2.5. This coordinate system is often used for global navigation.

2.2.2 North-East-Down (NED)

Because the geographic reference frame represents points in degrees, this coordinate frame is not convenient to work with in local navigation. The north-east-down (NED) coordinate frame is a tangent plane on the Earths surface, attached to the geographic reference frame at some fixed longitude-latitude pair (l_0, μ_0). The x-axis of the coordinate system points towards the true North, the y-axis points towards East and the z-axis points downwards to the Earths center [1]. This coordinate system is shown in Figure 2.6.

2.2.3 Body-fixed reference frame (BODY)

The body fixed reference frame is fixed to the vessels body. The x-axis of the BODY frame is the longitudinal axis of the vessel (positive towards fore), the y-axis is the transversal

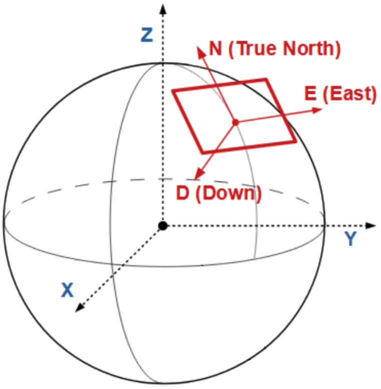


Figure 2.6: NED coordinate frame. Courtesy of SBG Systems [12].

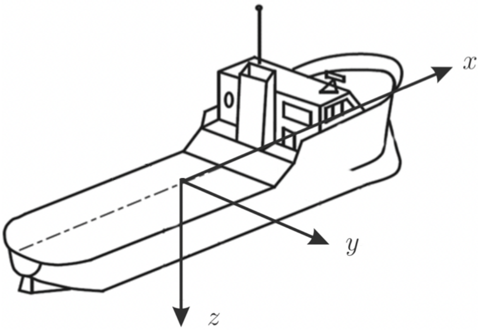


Figure 2.7: BODY coordinate frame. Courtesy of Tristan Perez [13].

axis of the vessel (positive towards starboard) and the z-axis is the normal axis of the vessel (positive downards) [1]. The BODY frame is shown in Figure 2.7.

2.3 SNAME notation

For this thesis, the notation of SNAME (1950) for marine vessels will be used. The notation is presented in [1], and is shown in Table 2.1 for 6 degrees of freedom (DOFs). The positions and Euler angles in the table refer to the vessels position and pose in the NED coordinate frame, while the forces, moments and velocities are given in the BODY frame. This section is also copied from the preceeding specialization project thesis [20].

DOF		**Forces and moments**	**Linear and angular velocities**	**Positions and Euler angles**
1	Surge motion	X	u	x
2	Sway motion	Y	v	y
3	Heave motion	Z	w	z
4	Rotation about x-axis (roll)	K	p	ϕ
5	Rotation about y-axis (pitch)	M	q	θ
6	Rotation about z-axis (yaw)	N	r	ψ

Table 2.1: SNAME (1950) notation for marine vessels. Courtesy of Thor I. Fossen [1].

For convenience sake, the general motion of a marine craft is described by the vectors for generalized position $\boldsymbol{\eta}$, velocity $\boldsymbol{\nu}$ and force $\boldsymbol{\tau}$. 6 DOF representations of the generalized vectors are defined as follows by [1] as

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} X \\ Y \\ Z \\ K \\ M \\ N \end{bmatrix}. \tag{2.1}$$

It is also useful to define some reference points in the vessel BODY frame, namely the CO, CG, CB and CF. The CO is the coordinate origin of the BODY frame and is used as the coordinate origin of the guidance, navigation and control (GNC) systems. The CG is the center of gravity relative to the CO, the CB is the center of buoyancy relative to the CO and the CF is the center of floation relative to the CO [1].

The SNAME (1950) notation for marine vessels also defines a compact way of expressing hydrodynamic derivatives. An example of this notation is given by (2.2) as

$$Y_{\dot{u}} \triangleq \frac{\partial Y}{\partial \dot{u}}. \tag{2.2}$$

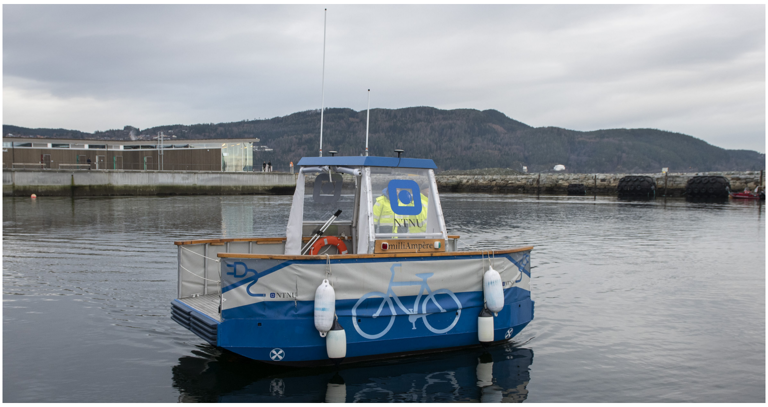


Figure 2.8: The milliAmpere1 passenger ferry. Courtesy of [14].

2.4 Vessels used for testing

The methods developed in this work are tested through simulations. The simulations are performed using models of the milliAmpere1 passenger ferry and the Otter USV. These two vessels are chosen due to their difference in maneuverability, in order to test the methods across different vessels.

2.4.1 The milliAmpere1 ferry

The milliAmpere1 passenger ferry was constructed at NTNU in 2017, and serves as a half-scale prototype of the full-scale ferry milliAmpere2. The prototype has been used in research within the field of maritime autonomy, and has been used by several M.Sc. and PhD students and professors at NTNU [37].

The milliAmpere1 is 5m long, and 2.8m wide and capable of carrying 6 passengers. Originally, the milliAmpere1 was equipped with 2 azimuth thrusters of 2kW each. One thruster was placed in the middle of the front of the ferry, and the other was placed in the middle of the back. During 2023, the milliAmpere1 will however be upgraded with the same thruster configuration as the full-scale passenger ferry milliAmpere2. This means the milliAmpere1 will have one azimuth thruster in each of the four corners of the vessel. The milliAmpere1 is also equipped with a complete sensor setup for situational awareness, consisting of radar, LIDAR and optical and infrared cameras [37]. The milliAmpere1 ferry is shown in Figure 2.8.

2.4.2 Otter USV

The Otter USV is 200 cm long and 108 cm wide, and is designed like a catamaran. It is the smallest ASV delivered by the Trondheim based company Maritime Robotics, and has a maximum speed of 6 knots. The ASV is powered by two electric motors in the back of the vessel, such that turning is done by applying different thrust to the two motors [15]. Figure 2.9 shows the Otter USV.

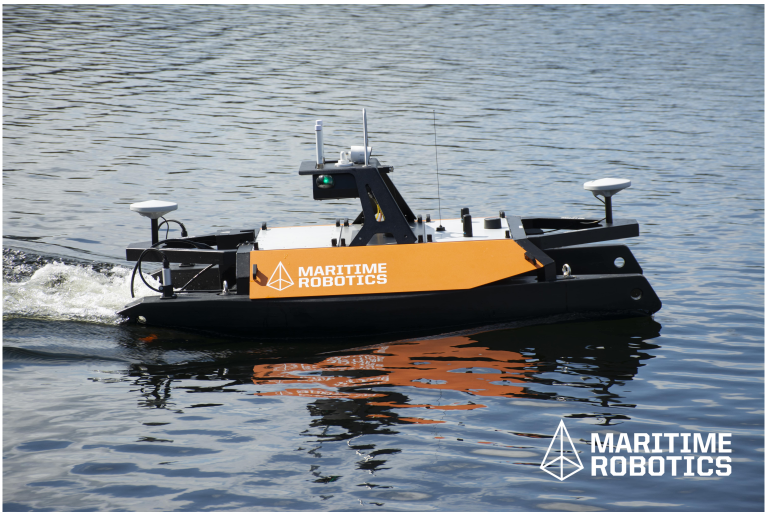


Figure 2.9: The Otter USV. Courtesy of Maritime Robotics [15].

2.5 3DOF vessel maneuvering model

A commonality between the vessels described in the previous section is that maneuvering models needs to be defined in order to describe their motion as accurately as possible. A 3DOF model frequently used in the literature is defined in [1], where only DOFs 1, 2 and 6 from (2.1) are considered. The 3DOF generalized vectors are defined by [1] as

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}, \qquad \boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ r \end{bmatrix}, \qquad \boldsymbol{\tau} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix}. \tag{2.3}$$

A 3DOF maneuvering model is then defined by [1] as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}, \tag{2.4}$$

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{M}_{A}\dot{\boldsymbol{\nu}}_r + \boldsymbol{N}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave}, \tag{2.5}$$

where

$$\boldsymbol{N}(\boldsymbol{\nu}_r) := \boldsymbol{C}_A(\boldsymbol{\nu}_r) + \boldsymbol{D} + \boldsymbol{D}_n(\boldsymbol{\nu}_r). \tag{2.6}$$

The rotational matrix $\boldsymbol{R}(\psi)$ in (2.4) is needed because the vessel's pose $\boldsymbol{\eta}$ is described in the NED reference frame, while the vessel's velocity $\boldsymbol{\nu}$ is described with respect to the vessel's body-fixed reference frame. The rotational matrix then describes a rotation from the BODY frame to the NED frame about the z-axis of the vessel, and is defined in [1] as

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.7}$$

The matrices $\boldsymbol{M}_{RB}$ and $\boldsymbol{M}_A$ together form the system inertia matrix $\boldsymbol{M}$. $\boldsymbol{M}_{RB}$ describes the rigid-body kinetics of the vessel, and $\boldsymbol{M}_A$ describes the hydrodynamic added mass by the fluid surrounding the vessel [1]. A 3DOF representation of these matrices are defined in [1] as

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \tag{2.8}$$

$$\boldsymbol{M}_{A} = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix}, \tag{2.9}$$

where m, x_g and I_z in (2.8) are the vessel mass, the location of the CG along the x-axis and the moment of inertia about the z-axis, respectively. The terms in (2.9) are hydrodynamic derivatives expressed by the SNAME (1950) notation. As mentioned, these two matrices together form the complete system inertia matrix

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A. \tag{2.10}$$

The Coriolis-centripetal matrix $\boldsymbol{C}$ is formed by the matrices $\boldsymbol{C}_{RB}(\boldsymbol{\nu})$ and $\boldsymbol{C}_A(\boldsymbol{\nu}_r)$ in (2.5) and (2.6) respectively. Similarly to the system inertia matrix, $\boldsymbol{C}_{RB}$ describes the rigid-body kinetics, and $\boldsymbol{C}_A$ describes the added hydrodynamic forces [1]. 3DOF representations of these matrices are defined in [1] as

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix}, \tag{2.11}$$

$$\boldsymbol{C}_{A}(\boldsymbol{\nu}_r) = -\begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 \end{bmatrix}, \tag{2.12}$$

where m, x_g and the hydrodynamic derivatives are the same as in (2.8) and (2.9), and u, v and r are the linear- and angular velocities from (2.3). The subscript $(\cdot)_r$ denotes that the velocity takes account of the ocean current speed, and the relative speeds are defined in [1] as

$$u_r = u - u_c, \tag{2.13}$$

$$v_r = v - v_c, \tag{2.14}$$

where u_c and v_c are the velocity components of the current.

The matrices $\boldsymbol{D}$ and $\boldsymbol{D}_n(\boldsymbol{\nu}_r)$ in (2.6) describe the linear- and nonlinear hydrodynamic damping experienced by the vessel. 3DOF representations of these matrices are defined in

[1] as

$$\boldsymbol{D} = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}, \tag{2.15}$$

$$\boldsymbol{D}_n(\boldsymbol{\nu}_r) = -\begin{bmatrix} X_{|u|u}|u_r| & 0 & 0 \\ 0 & Y_{|v|v}|v_r| + Y_{|r|v}|r| & Y_{|v|r}|v_r| + Y_{|r|r}|r| \\ 0 & N_{|v|v}|v_r| + N_{|r|v}|r| & N_{|v|r}|v_r| + N_{|r|r}|r| \end{bmatrix}, \tag{2.16}$$

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

The terms $\boldsymbol{\tau}$, $\boldsymbol{\tau}_{wind}$ and $\boldsymbol{\tau}_{wave}$ on the right-hand side of (2.5) contain the control inputs, generalized wind forces and generalized wave-induced forces, respectively [1].

2.5.1 Scaling the vessel model

A part of this work is to work with underactuated vessels that are faster than the milliAmpere1 ferry and the Otter USV, more specifically vessels that are approximately 15 meters long and capable of traveling at speeds of 15 to 20 knots. In order to model such a vessel, this work involves scaling the Otter USV model already obtained through the Marine Systems Simulator [2] to fit these requirements.

Scaling a vessel model can be done through non-dimensionalization of the existing model, i.e. making all dimensional parameters of the model dimensionless. After the non-dimensionalization of the model, the model can be scaled up to the desired size through dimensionalization. Two methods for non-dimensionalization commonly used when designing ship control systems are described in [1], namely the *Prime system* of SNAME (1950) and the *Bis system* of Norrbin (1970). These systems differ in the variables that are used for normalization, referred to as the normalization variables.

The Prime system uses the vessel's speed U, the length L, the mass unit $1/2\rho L^3$ and the time unit L/U as normalization variables [1, p. 675]. One problem with the Prime system is that normalization of the velocities u, v and w involves dividing by the speed U. This means the Prime system involves singularities at zero speed U, which makes it unsuitable for e.g. dynamic positioning applications.

The Bis system does on the other hand use the length L and the time unit $\sqrt{L/g}$ as normalization variables. This means the Bis system does not lead to singularities at zero speed, which is the reason why the Bis system is used for scaling in this work, even though there is little risk of the vessel having zero speed. The mass unit used by the Bis system is $\mu\rho\nabla$, where μ is the density ratio ($\mu = 1$ for floating ships), ρ is the density of water and ∇ is the hull contour displacement of the vessel. The normalization variables used by the Bis system are summarized in Table 2.2.

Scaling a vessel model then involves using the Bis variables from Table 2.2 to non-dimentionalize/normalize each dimensional parameter, and then dimensionalize it again to the new size. This procedure can be shown through an example, where the moment of inertia I_z is scaled.

To begin with, the original length and mass of the vessel is denoted L_1 and m_1 respectively, and the scaled length and mass L_2 and m_2 respectively. In order to non-dimensionalize I_z, the Bis-variable for inertia moment from Table 2.2 is used. Using the notation $I_{z,1}$ to describe the original moment of inertia about the z-axis and $I'_{z,1}$ to describe the non-dimensionalized variable then yields

$$I'_{z,1} = \frac{I_{z_1}}{\mu\rho\nabla_1 L_1^2} = \frac{I_{z,1}}{m_1 L_1^2}, \tag{2.17}$$

where the second equality comes from $\mu\rho\nabla_1 = m_1$. The scaled moment of inertia $I_{z,2}$ can then be obtained through dimensionalization of $I'_{z,1}$ as

$$I_{z,2} = I'_{z,1} \cdot \mu\rho\nabla_2 L_2^2 = \frac{I_{z,1}}{m_1 L_1^2} \cdot m_2 L_2^2 = \frac{m_2 L_2^2}{m_1 L_1^2} I_{z,1}. \tag{2.18}$$

Based on the above, it is useful to define two scaling coefficients as

$$c_l = \frac{L_2}{L_1}, \tag{2.19}$$

$$c_m = \frac{m_2}{m_1}, \tag{2.20}$$

which gives

$$I_{z,2} = c_m c_l^2 I_{z,1}. \tag{2.21}$$

Because scaling will always involve dividing and multiplying by the Bis variable as above, and g and ρ will cancel during these operations, the scaling of each unit can be described by the coefficients c_l and c_m only. This is shown in the right column of Table 2.2.

Unit	**Symbol**	**Bis variable**	**Scaling variable**
Length	[m]	L	c_l
Mass	[kg]	$\mu\rho\nabla$	c_m
Inertia moment	[kg m^2]	$\mu\rho\nabla L^2$	$c_m c_l^2$
Time	[s]	$\sqrt{L/g}$	$\sqrt{c_l}$
Position	[m]	L	c_l
Angle	[rad]	1	1
Linear velocity	[m/s]	$\sqrt{Lg}$	$\sqrt{c_l}$
Angular velocity	[rad/s]	$\sqrt{g/L}$	$1/\sqrt{c_l}$
Linear acceleration	[m/s^2]	g	1
Angular acceleration	[rad/s^2]	g/L	$1/c_l$
Force	[N]	$\mu\rho g\nabla$	c_m
Moment	[Nm]	$\mu\rho g\nabla L$	$c_m c_l$

Table 2.2: Bis system variables used for non-dimensionalization and scaling. The Bis variables are based on [1]. The coefficients c_l and c_m are defined by (2.19) and (2.20), respectively.

2.6 Simulating the vessels

In order to simulate the vessels for testing in this work, a few key components needs to be in place. Among these are the control allocation, which results in the control inputs $\boldsymbol{u}$, as well as guidance laws used to calculate the reference signals for the controllers used by the vessels. The theory presented in this section is based on [1]. Section 2.6.1 is mostly copied from the specialization project thesis that was written prior to this work [20], the difference being that it is generalized for this work. Section 2.6.2 is copied from [20], while Section 2.6.3 is based on [38].

2.6.1 Control allocation

Control allocation means distributing the generalized forces given by $\boldsymbol{\tau}$ in (2.5), to the actuators in terms of the actual control inputs $\boldsymbol{u}$. Based on $\boldsymbol{\tau}$ and the control matrix $\boldsymbol{B}$, the control input $\boldsymbol{u}$ is defined in [1] as

$$\boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{\tau}. \tag{2.22}$$

The control matrix $\boldsymbol{B}$ has to be calculated based on the thrust configuration matrix $\boldsymbol{T}$ and the force coefficient matrix $\boldsymbol{K}$. The numerical values in the force coefficient matrix are often found through experimental data. The thrust configuration matrix is based on the actuator configuration of the specific vessel, and is generally given by [1] as

$$\boldsymbol{T} = \begin{bmatrix}\boldsymbol{t}_1 & \cdots & \boldsymbol{t}_r\end{bmatrix} \tag{2.23}$$

where r is the number of control inputs. Using (2.22), the control input $\boldsymbol{u}$ can be calculated as

$$\boldsymbol{u} = (\boldsymbol{K}\boldsymbol{T})^{-1}\boldsymbol{\tau}. \tag{2.24}$$

2.6.2 Line-of-sight guidance

Given a set of waypoints describing the path in the NED frame as a list of (x, y)-coordinates, the line-of-sight (LOS) guidance law provides a reference χ_d for the heading autopilot in order for the vessel to follow the piecewise-linear path described by the waypoints.

When planning a path between two waypoints, it is useful to define a path-tangential coordinate system. This coordinate system is obtained by performing a clockwise rotation of π_p degrees with respect to the NED coordinate frame. The coordinate system is shown in Figure 2.10. This figure also shows the along-track and cross-track error x_e^p and y_e^p respectively. These errors are defined in [1] as

$$\begin{bmatrix} x_e^p \\ y_e^p \end{bmatrix} = \boldsymbol{R}_p^n(\pi_p)^\top \left[\left(\begin{bmatrix} x^n \\ y^n \end{bmatrix} - \begin{bmatrix} x_i^n \\ y_i^n \end{bmatrix}\right)\right], \tag{2.25}$$

where (x^n, y^n) denotes the vessels position in the NED frame, (x_i^n, y_i^n) denotes the waypoints position in the NED frame and the rotation matrix $\boldsymbol{R}_p^n$ is given by

$$\boldsymbol{R}_p^n(\pi_p) = \begin{bmatrix} \cos(\pi_p) & -\sin(\pi_p) \\ \sin(\pi_p) & \cos(\pi_p) \end{bmatrix}. \tag{2.26}$$

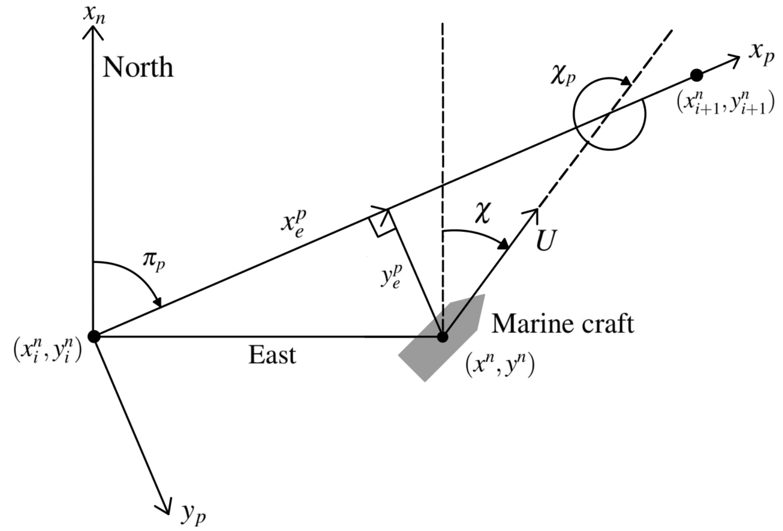


Figure 2.10: Path-tangential coordinate system. Adapted from Thor. I. Fossen [1].

When a vessel is going from waypoint $\boldsymbol{p}_i$ to waypoint $\boldsymbol{p}_{i+1}$, the desired course angle χ_d for the vessel can then be calculated using the Proportional LOS Guidance Law [1]

$$\chi_d = \pi_p - \tan^{-1}(K_p y^p_e), \tag{2.27}$$

where K_p is a proportional gain given by

$$K_p = \frac{1}{\Delta}, \tag{2.28}$$

where the lookahead distance $\Delta > 0$ is a design parameter.

When there are multiple waypoints along a path, it is important for the guidance system to know when to switch waypoints. Instead of considering waypoints $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$, the system should at some point consider waypoints $\boldsymbol{p}_{i+1}$ and $\boldsymbol{p}_{i+2}$ and so on. One suitable criterion for switching waypoints is by considering the along-track error x^p_e and the total along-track distance between the waypoints d_{i+1}. The switch can then be performed based on the criteria given in [1] as

$$d_{i+1} - |x^p_e| \leq R_{switch}, \tag{2.29}$$

where $R_{switch} > 0$ is the distance from the next waypoint along the path between from the previous waypoint in which it is acceptable to perform a switch, and the along-track distance d_{i+1} is given by

$$d_{i+1} = \|\boldsymbol{p}_{i+1} - \boldsymbol{p}_i\|_2. \tag{2.30}$$

2.6.3 Target tracking

A controller controlling the speed of a vessel is implemented as part of this work, leading to the need for a speed reference $U_d > 0$. One way of calculating this reference is to use guidance laws for target tracking, where the goal is for the ownship to track the motion of a target, making the ownship an *interceptor* [38]. Mathematically, the objective of a target-tracking scenario is formulated in [38] as

$$\lim_{t \to \infty} (\boldsymbol{p}_t^n(t) - \boldsymbol{p}^n(t)) = \boldsymbol{0}, \tag{2.31}$$

where $\boldsymbol{p}_t^n$ and $\boldsymbol{p}^n$ are the positions of the target and the ownship in the NED frame respectively.

When the ownship is pursuing the target, the idea presented in [38] is to assign the ownship reference velocity as

$$\boldsymbol{v}_d(t) = \boldsymbol{v}_t(t) + \boldsymbol{v}_a(t), \tag{2.32}$$

where $\boldsymbol{v}(t) = \dot{\boldsymbol{p}}^n(t)$, $\boldsymbol{v}_t(t) = \dot{\boldsymbol{p}}_t^n(t)$, and $\boldsymbol{v}_a(t)$ is a relative approach velocity added to the velocity reference in order to allow the ownship to increase/reduce its velocity to be able to track the target. The $\boldsymbol{v}_a$ component is in [38] modelled as a sigmoid function based on the distance between the ownship and the target. In [38] this added speed component is chosen as

$$\boldsymbol{v}_a(t) = U_{a,max}(t) \frac{\tilde{\boldsymbol{p}}(t)}{\sqrt{\tilde{\boldsymbol{p}}^\top \tilde{\boldsymbol{p}} + \Delta_p^2}}, \tag{2.33}$$

where $U_{a,max} > 0$ is the maximum added velocity, $\Delta_p > 0$ is a parameter used to influence the transient behaviour of the speed similar to the lookahead distance Δ used in (2.28) and

$$\tilde{\boldsymbol{p}}(t) = \boldsymbol{p}_t(t) - \boldsymbol{p}(t) \tag{2.34}$$

is the line-of-sight vector between the ownship and the target. A suitable choice for $U_{a,max}$ is given in [38] as

$$U_{a,max}(t) = \zeta U_t(t), \quad \zeta < 1, \tag{2.35}$$

in order to give the ownship a speed advantage over the target by ensuring

$$U_{max} > U_t(t) + U_{a,max}(t) \quad \Rightarrow \quad U_{max} > U_t(t) + \zeta U_t(t). \tag{2.36}$$

The assigned speed to the ownship then takes the form

$$\boldsymbol{v}_d(t) = \boldsymbol{v}_t(t) + U_{a,max}(t) \frac{\tilde{\boldsymbol{p}}(t)}{\sqrt{\tilde{\boldsymbol{p}}^\top \tilde{\boldsymbol{p}} + \Delta_p^2}} \tag{2.37}$$

and the speed reference used as reference by the speed controller can finally be calculated as

$$U_d(t) = \|\boldsymbol{v}_d(t)\|_2. \tag{2.38}$$

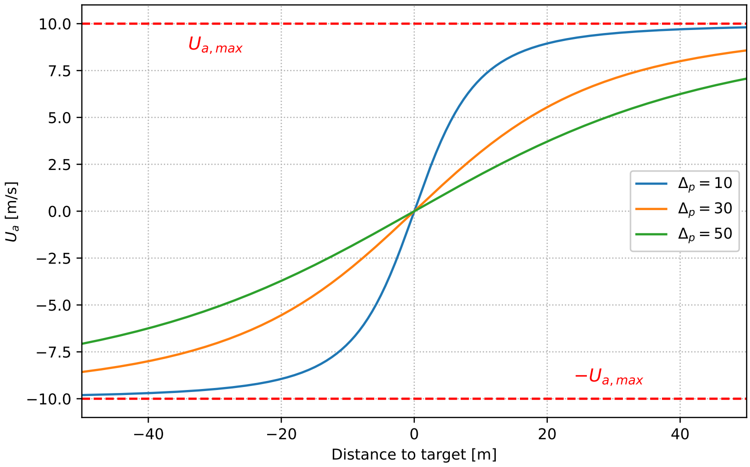


Figure 2.11: Relative approach speed $U_a(t)$ as a function of Euclidean distance between the own-ship and the target with three choices of Δ_p. The maximum added velocity $U_{a,max}$ is here set to 10 m/s. Inspired by [16].

Using that $\boldsymbol{v}_d(t) = \begin{bmatrix}\dot{x}^n & \dot{y}^n\end{bmatrix}^\top$, a reference course angle can also be calculated as

$$\chi_d(t) = \text{atan2}(\dot{y}^n, \dot{x}^n). \tag{2.39}$$

An example of the relative approach speed $U_a(t) = \|\boldsymbol{v}_a(t)\|_2$ is shown in Figure 2.11 as a function of the Euclidean distance to the target with different choices of Δ_p.

2.7 Optimization and control

In this section, the theory behind optimization and optimal control is described. This covers the basics behind optimization, nonlinear programming, and how an optimal control problem (OCP) is defined. Finally, a frequently used method within optimal control, MPC, is described. The theory and notation presented in this section is based on [19].

2.7.1 Optimization

Optimization problems refer to a class of mathematical problems where the goal is to minimize or maximize some objective function subject to a number of constraints. For simplicity this section covers problems where the goal is to minimize some objective, but a minimization problem is translated to a maximization problem simply by changing the sign of the objective function. Mathematically an optimization problem is defined in [19] as

$$\begin{aligned} &\min_{\boldsymbol{z}\in\mathbb{R}^n} && f(\boldsymbol{z}), && (2.40)\\ &\text{subject to} && c_i(\boldsymbol{z}) = 0, \quad i \in \mathcal{E}, && (2.41)\\ & && c_i(\boldsymbol{z}) \geq 0, \quad i \in \mathcal{I}, && (2.42)\end{aligned}$$

where $f(\cdot)$ is the objective function to be minimized, $\boldsymbol{z}$ is the n-dimensional decision variable and $c_i(\cdot)$ are the constraints. The constraints are defined in terms of two disjoint index sets $\mathcal{E}$ and $\mathcal{I}$, which holds the indices of the equality-constraints and inequality-constraints, respectively. These constraints together form a feasible region in $\mathbb{R}^n$ in which the solution lies. Such optimization problems have a wide range of applications, from maximizing profits in economical applications to minimizing the energy consumed by a vessel going from A to B. One class of optimization problems are Nonlinear Programs (NLP), where the objective $f(\cdot)$ and/or some of the constraints $c_i(\cdot)$ are nonlinear functions.

2.7.2 Dynamic optimization

Dynamic optimization refers to the optimization of dynamic systems. In dynamic optimization, all decision variables are functions of time, as the goal is to optimize with respect to some dynamic model where the modelled process changes over time. Most often the dynamic model of the system at hand is discretized in [19] as

$$\begin{aligned} \boldsymbol{x}_{t+1} &= g(\boldsymbol{x}_t, \boldsymbol{u}_t), && (2.43)\\ \boldsymbol{u}_t &\in \mathbb{R}^{n_u}, && (2.44)\\ \boldsymbol{x}_t &\in \mathbb{R}^{n_x}, && (2.45)\end{aligned}$$

where $\boldsymbol{u}_t$ is the control input at time t, $\boldsymbol{x}_t$ is the system state at time t and $g(\boldsymbol{x}_t, \boldsymbol{u}_t)$ calculates the state $\boldsymbol{x}_{t+1}$ at time $t+1$ based on the current state $\boldsymbol{x}_t$ and control input $\boldsymbol{u}_t$. The control input $\boldsymbol{u}_t$ is assumed to be constant between t and $t+1$, and the state $\boldsymbol{x}_t$ is only defined at time t.

An example of a standard dynamic optimization problem is given in [19] as

$$\min_{\boldsymbol{z}\in\mathbb{R}^n} \quad f(\boldsymbol{z}) = \sum_{t=0}^{N-1} \frac{1}{2}\boldsymbol{x}_{t+1}^\top \boldsymbol{Q}_t \boldsymbol{x}_{t+1} + d_{x_{t+1}}^\top \boldsymbol{x}_{t+1} + \frac{1}{2}\boldsymbol{u}_t^\top R_t \boldsymbol{u}_t + d_{u_t}^\top \boldsymbol{u}_t, \tag{2.46}$$

$$\text{subject to} \quad \boldsymbol{x}_{t+1} = g(\boldsymbol{x}_t, \boldsymbol{u}_t), \qquad t = 0, \dots, N-1, \tag{2.47}$$

$$\boldsymbol{x}^{low} \leq \boldsymbol{x}_t \leq \boldsymbol{x}^{high}, \qquad t = 1, \dots, N, \tag{2.48}$$

$$\boldsymbol{u}^{low} \leq \boldsymbol{u}_t \leq \boldsymbol{u}^{high}, \qquad t = 0, \dots, N-1, \tag{2.49}$$

$$-\Delta\boldsymbol{u}^{high} \leq \Delta\boldsymbol{u}_t \leq \Delta\boldsymbol{u}^{high}, \qquad t = 0, \dots, N-1, \tag{2.50}$$

$$\boldsymbol{Q}_t \geq 0, \qquad t = 1, \dots, N, \tag{2.51}$$

$$\boldsymbol{R}_t \geq 0, \qquad t = 0, \dots, N-1, \tag{2.52}$$

$$\boldsymbol{x}_0, \boldsymbol{u}_{-1} \quad \text{given}, \tag{2.53}$$

where (2.47) represents the discrete time model of the system, (2.48) and (2.49) represent lower and upper bounds on the states and control input of the system and (2.51) represents lower and upper bounds for the change in control input. The parameters $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ are parameters used for weighting in the objective function. For a minimization problem as above, a high $\boldsymbol{R}_t$ relative to $\boldsymbol{Q}_t$ will e.g. penalize high control inputs. It is also worth noting that $\boldsymbol{z}$ will have the form

$$\boldsymbol{z} = \left[\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_N^\top, \boldsymbol{u}_0^\top, \dots \boldsymbol{u}_{N-1}^\top\right]^\top \tag{2.54}$$

where N is the planning horizon of the optimization problem, leading to $\boldsymbol{z}$ having dimension $n = N(n_x + n_u)$. The problem (2.46) - (2.53) is referred to as a *finite horizon open loop optimization problem* due to the lack of feedback control, meaning the problem is solved once over a finite prediction horizon of N steps [19].

2.7.3 Model predictive control

Model Predictive Control (MPC) is a popular algorithm that merges the gap between dynamic optimization and control by adding feedback control, effectively closing the loop leading to a *closed-loop optimization problem.* This is obtained by solving a finite horizon optimization problem like described in Section 2.7.2 at each timestep, using an estimate of the current state of the dynamic system as the initial conditions $\boldsymbol{x}_0$ and the previous control input as $\boldsymbol{u}_{-1}$ in (2.53) as described in [19]. The finite horizon optimization problem (2.46)-(2.53) is then solved over a horizon of N prediction steps using these initial conditions. From the solution of this problem, the first m control inputs are applied to the dynamic model. The algorithm is summarized by Algorithm 1.

Algorithm 1: Output feedback MPC procedure. Courtesy of [19].

```
for $t = 0, 1, 2, \ldots$ do
    Compute an estimate of the current state $\hat{\boldsymbol{x}}_t$ based on measured data up until
      time $t$;
    Solve a dynamic optimization problem on the prediction horizon from $t$ to
      $t + N$ with $\hat{\boldsymbol{x}}_t$ as the initial condition;
    Apply the first control input $\boldsymbol{u}_t$ from the solution of the dynamic optimization
      problem;
end
```

2.8 Encounter classifications

In order to make a trajectory planner adhere to the COLREGs described in Section 2.1, a method for classifying the different encounters described in Rules 13-15 and Rule 17 needs to be utilized. A vessel-to-vessel encounter classification algorithm is described in [17], which is able to classify overtaking (OT), head-on (HO), give-way crossing (GW), stand-on (SO) and safe (SF) encounters. The SF encounter type is defined as a situation where the ownship is not required to follow any of the encounter-specific COLREGs rules described in Section 2.1.

The classification method described in [17] is a two-step process where the relative bearing sector (RBS) is determined in the first step, and the situation sector (SS) is determined in the second step. The RBS is determined based on the relative bearing of the target ship from the ownship, given in [17] as

$$\varphi = \text{atan2}((E_{TS} - E), (N_{TS} - N)) - \chi \tag{2.55}$$

where χ is the course of the ownship, E_{TS} and E are the east positions of the target ship and ownship respectively, and N_{TS} and N are the north positions of the target ship and ownship, respectively. The RBS is divided into four sectors of a circle surrounding the ownship: R1, R2, R3 and R4. The target ship will then be put into one of these four sectors based on the relative bearing of the target ship φ and the sector angles $[-\theta_2, -\theta_1, \theta_1, \theta_2]$.

In the second step of the encounter classification, the encounter is classified based on the SS. The SS is determined based on the course of the target ship relative to the ownship, given in [17] as

$$\chi_{rel} = \chi_{TS} - \chi, \tag{2.56}$$

where χ_{TS} and χ are the course of the target ship and ownship, respectively. This angle determines the encounter, and is based on the rotated sector angles $[-\theta_2', -\theta_1', \theta_1', \theta_2']$, where $\theta_1' = \theta_1 - \varphi_{TS}$ and $\theta_2' = \theta_2 - \varphi_{TS}$, where

$$\varphi_{TS} = \text{atan2}((E - E_{TS}), (N - N_{TS})) \tag{2.57}$$

is the bearing of the ownship from the target ship. The encounter classification algorithm proposed by [17] is shown in Figure 2.12. In this figure, the OT-encounter is split in two sub-encounters: OT$_p$ and OT$_s$, which are defined as an overtaking on the port side

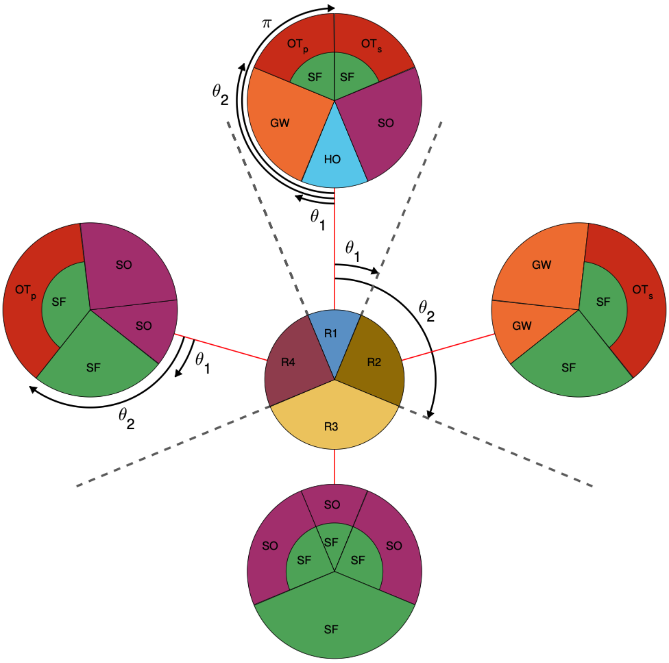


Figure 2.12: Visualization of how the encounter classification is done. The ownship is located at the center of the middle circle, and each of the other four circles represent a target ship located at four different positions relative to the ownship. The RBS sectors can be seen in the middle circle as R1, R2, R3 and R4, with sector angles $[-\theta_2, -\theta_1, \theta_1, \theta_2]$. The situation sectors are seen in the outer circles, and provide the final answer of the encounter classification. These sectors are determined by the rotated sector angles $[-\theta_2', -\theta_1', \theta_1', \theta_2']$. Courtesy of [17].

and starboard side of the target ship, respectively. Some of the SS in the figure have two encounters in one sector, and [17] proposes that the outer encounter classification is chosen when the involved vessels have a closing range, while the inner one is chosen for an increasing range in these cases.

2.9 Performance metrics

In order to evaluate the methods proposed in this thesis, some performance metrics need to be defined. The first subsection of this section presents performance metrics with respect to the performance of the motion control system onboard the vessel, the energy consumed by the vessel, the actuator wear-and-tear as well as the acceleration performed by the vessel. This part is mostly copied from the specialization project thesis that was written prior to this work [20], the difference being that not only the control error of the yaw angle of the vessel is considered in this work. The second subsection handles performance metrics used to assess the COLREGs-compliance of the vessel, and is based on [30].

2.9.1 Motion control metrics

It is always possible to look at plots of the tracking error, control error and similar visual metrics. A problem with this approach is that purely visual performance assessment won't always tell the whole story of what is going on. Defining some well suited performance metrics to the specific application is therefore always a good idea.

Some metrics used to assess the performance of different control systems are described in [16]. A subset of these metrics is considered useful also for this work, as they can help assess how the control system handles the different generated paths. The most useful metrics are the integral of the absolute value of the error multiplied by the energy consumption (IAEW), the integral of absolute differentiated control (IADC) and the combination of these two (IAEW-WT).

The integral of the absolute value of the error multiplied by the energy consumption (IAEW) combines the control error and the energy consumption in one metric, instead of only looking at the control error. This is useful to assess not only how well the control system is able to follow its reference, but also how much energy is consumed by the vessel by doing so. The IAEW is defined in [16] as

$$IAEW \triangleq \int_0^t |e(\sigma)|d\sigma \int_0^t P(\sigma)d\sigma, \tag{2.58}$$

where $e(t)$ is the control error of the system. For this work, the control error is calculated based on references and outputs with different unit, contrary to what was done in the specialization project thesis [20]. In [39] it is proposed to normalize the references and outputs such that all variables are in the interval $[0, 1]$ within the expected operation space of the vessel. This means that for e.g. a controller controlling the speed $U = \sqrt{u^2 + v^2}$ and yaw ψ of the vessel, the combined control error is defined in [39] as

$$e(t) = \sqrt{(\bar{U}(t) - \bar{U}_d(t)^2 + (\bar{\psi}(t) - \bar{\psi}_d(t))^2} \tag{2.59}$$

where $\bar{(\cdot)}$ denotes a normalized variable that is in the interval $[0, 1]$. Because the speed has units m/s and the yaw angle has unit rad, this effectively avoids potential problems with the IAEW metric penalizing the numerically larger control error in speed more than control error in yaw angle.

The power consumption $P(t)$ of the system is defined in [16] as

$$P(t) = \|\boldsymbol{\nu}(t)^\top \boldsymbol{\tau}(t)\|_2. \tag{2.60}$$

The integral of absolute differentiated control (IADC) gives an indication of actuator usage, and can therefore be used as an indicator of the wear-and-tear on the vessel actuators as a result of the specific path. The IADC is defined in [16] as

$$IADC \triangleq \int_0^t \|\dot{\boldsymbol{\tau}}(\sigma)\|_2 d\sigma, \tag{2.61}$$

where $\dot{\boldsymbol{\tau}}(t)$ is calculated numerically as

$$\dot{\boldsymbol{\tau}}(t) = \frac{\boldsymbol{\tau}(t) - \boldsymbol{\tau}(t-h)}{h}, \tag{2.62}$$

where $h > 0$ is the sampling time.

Combining the IAEW and IADC gives an indication of the tracking error, energy consumption and the actuator wear-and-tear in one performance metric, and is given by the IAEW-WT defined in [16] as

$$IAEW - WT \triangleq \int_0^t |e(\sigma)| d\sigma \int_0^t P(\sigma) d\sigma \int_0^t \|\dot{\boldsymbol{\tau}}(\sigma)\|_2 d\sigma. \tag{2.63}$$

Another important metric for all vessels that are to maneuver autonomously is the comfort experienced by the passengers onboard the vessel. This will to some degree be assessed by the metrics above, but a better way to evaluate it more clearly is to calculate the integral of absolute acceleration (IAA) during the transit. This will indicate the accelerations experienced by the passengers during transit, and thus the comfort/discomfort experienced by them. The IAA is defined as

$$IAA \triangleq \int_0^t \|\dot{\boldsymbol{\nu}}(\sigma)\|_2 d\sigma, \tag{2.64}$$

where $\dot{\boldsymbol{\nu}}$ is calculated numerically as

$$\dot{\boldsymbol{\nu}}(t) = \frac{\boldsymbol{\nu}(t) - \boldsymbol{\nu}(t-h)}{h}. \tag{2.65}$$

The performance metrics presented in this section will during simulation integrate throughout the simulation, and a commonality between the metrics is that a low final value is preferable to a higher one. Figure 2.13 shows an example of how the different metrics evolve over time, and are obtained from simulations performed as part of this work.

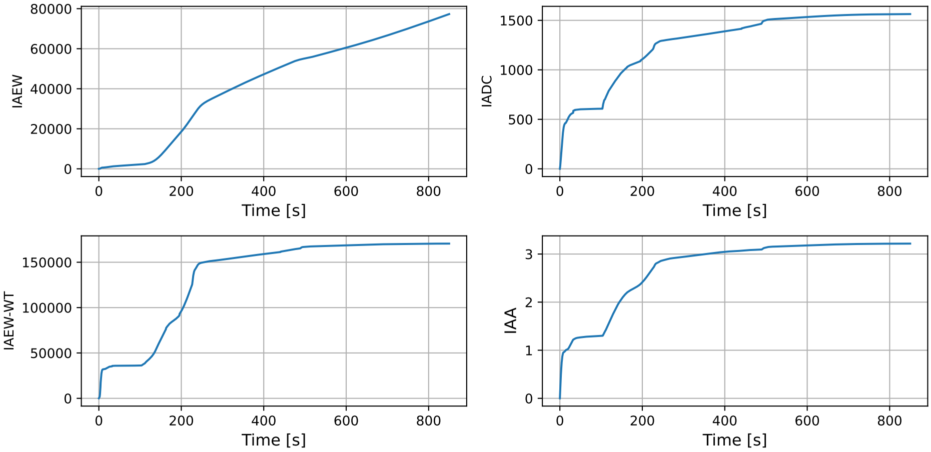


Figure 2.13: Example showing how the IAEW, IADC, IAEW-WT and IAA metrics evolve over time during a simulation. The metrics shown are obtained through early simulations in this work.

2.9.2 COLREGs-specific metrics

As discussed in Section 1.2, developing performance metrics to assess the COLREGs-compliance of a trajectory-planning algorithms is not trivial. Some metrics are however defined in [30] and [32], which are based on [31]. A subset of these metrics are also used in this work, and the notation and descriptions of the metrics in the following are based on [30]. Similarly to [30], the metrics defined in this subsection will either be defined as penalty scores $\mathcal{P}$ where a score of 1 is deemed the worst behaviour and 0 is deemed the best behaviour, or as scores $\mathcal{S}$ where 1 is deemed the best behaviour and 0 is deemed the worst.

Rule 8: Ample time and readily apparent action

Rule 8 of the COLREGs introduces the expressions *ample time* and *readily apparent action*. An action to avoid collision shall be made in ample time, and be readily apparent for other vessel(s) to observe. These are metrics that are hard to enforce in a trajectory planner, as they are situationally dependent and to some degree dependent on experience at sea.

One metric for assessing whether action is made in ample time is given in [30] as

$$\mathcal{P}_{delay} = \min\left(1, \frac{r_{detect} - r_{maneuver}}{r_{detect} - r_{cpa}}\right), \tag{2.66}$$

where r_{detect} is the distance between the ownship and target ship at the time the target ship is detected by the ownship, $r_{maneuver}$ is the distance between the vessels at the time the ownship initiates a maneuver and r_{cpa} is the distance between the vessels at the closest point of approach (CPA). These parameters can be calculated based on data from simulations or physical experiments, as the only data needed is the pose and velocities of the ownship and target ship. The r_{cpa} parameter is then simply found as the minimum distance between the vessels throughout the encounter.

The $r_{maneuver}$ parameter is harder to calculate, as it involves tracking where the ownship starts to perform a maneuver. In [30] this parameter is calculated based on observing a change in course and/or speed after the time of detection. One can then define two thresholds $\epsilon_\chi > 0$ and $\epsilon_U > 0$ as the minimum change in course and speed respectively that can be seen as the start of a maneuver. The parameter $r_{maneuver}$ can then be defined as the distance between the ownship and target ship at the time t when one or both of the criteria

$$|\chi(t_{detect}) - \chi(t)| \geq \epsilon_\chi, \tag{2.67}$$

$$|U(t_{detect}) - U(t)| \geq \epsilon_U, \tag{2.68}$$

holds [30]. As described in Section 2.1, Rule 8 of the COLREGs also states that a change in course is preferable to a change in speed. This means that the first of these criteria is arguably the most relevant in order to assess not only how early action is initiated, but also that the action is more easily observable.

One usage of this metric can be assessing the robustness of the trajectory planner based on varying the r_{detect} parameter. In this case one can vary the range within which a vessel is considered by the planner, and see how robust the planner is to late detections of other vessels.

In order to assess how readily apparent a maneuver is, [31] defines the metric

$$\mathcal{P}_{\Delta\chi_{app}}(\Delta\chi) = \min\left(0, \frac{\Delta\chi_{app} - |\Delta\chi|}{\Delta\chi_{app} - \Delta\chi_{md}}\right), \tag{2.69}$$

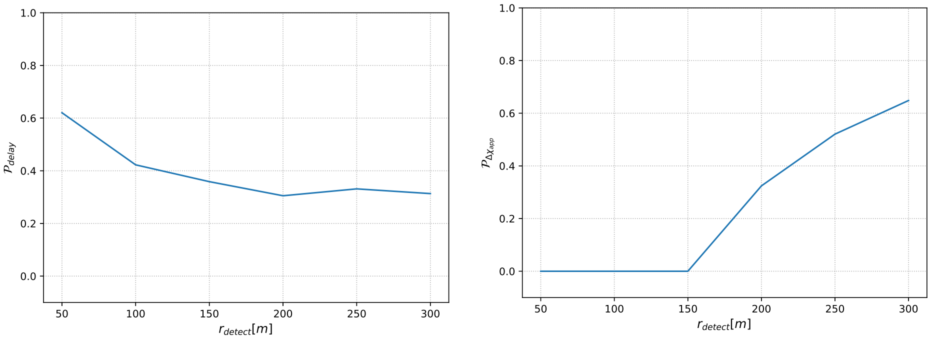
where $\Delta\chi_{app}$ is the lower course threshold that can be deemed a readily apparent maneuver, $|\Delta\chi|$ is the absolute course change of a maneuver and $\Delta\chi_{md}$ is the minimum detectable change in course. In [30] however, a more strict penalizing score is defined in order to penalize small changes in course more than (2.69). This penalization score is given in [30] as

$$\mathcal{P}'_{\Delta\chi_{app}}(\Delta\chi) = \min\left(0, 1 - \frac{|\Delta\chi|^2}{\Delta\chi_{app}^2}\right). \tag{2.70}$$

In (2.70), the only parameters that needs to be defined are $|\Delta\chi|^2$ and $\Delta\chi_{app}^2$. In [34, p. 50], it is argued that an alteration of course should be at least 30°, and should preferably be between 60° and 90° in order for the maneuver to be readily apparent to another vessel observing by radar. This means that $\Delta\chi_{app}$ should at least be 30°, and preferably between 60° and 90°. It can be noted that using e.g. 30° as the threshold ϵ_χ in (2.67) will then to some extent assess both the delay of the action as well as how readily apparent the action is in the penalization score (2.66). In [30] it is proposed to calculate the absolute course alteration of a maneuver $|\Delta\chi|$ in (2.70) as

$$|\Delta\chi| = \max\{|\chi(t_{detect}) - \chi_0|, |\chi(t_{detect} + 1) - \chi_0|, \ldots |\chi(t_{cpa}) - \chi_0|\}, \tag{2.71}$$

where χ_0 is the course of the ownship at the time of detection, i.e. $\chi_0 = \chi(t_{detect})$. This effectively means that $|\Delta\chi|$ will be the largest change in course of the ownship between the time of detection t_{detect} and the time of CPA, t_{cpa}.



(a) $\mathcal{P}_{delay}$.

(b) $\mathcal{P}'_{\Delta\chi_{app}}$.

Figure 2.14: Examples showing the COLREGs-specific performance metrics $\mathcal{P}_{delay}$ and $\mathcal{P}'_{\Delta\chi_{app}}$ as functions of the range of detection r_{detect} from simulations.

Examples of $\mathcal{P}_{delay}$ and $\mathcal{P}'_{\Delta\chi_{app}}$ are shown in Figure 2.14. In these plots, simulations are ran with different ranges of detection enforced through the ZeabuzSimulator used in this work, using an apparent course alteration threshold of $\chi_{app} = 30°$. As shown in Figure 2.14b, the $\mathcal{P}'_{\Delta\chi_{app}}$-metric gets a score of 0 for very short ranges of detection. This is likely due to the vessel having to do more aggressive maneuvers when the range of detection is shorter. This shows the need for the performance metrics described in the previous section, in order to also see e.g. the the actuator usage of the vessel in these scenarios, as a scenario leading to a good $\mathcal{P}'_{\Delta\chi_{app}}$ score might lead to a bad (i.e. high) IAEW-WT-score for example. This goes to show the need for an extensive use of relevant metrics when working with topics such as trajectory planning.

Another important metric described in [30] is the safety-metric $\mathcal{S}_{safety}$, which attempts to quantify the overall safety of encounters. The safety metric is not a penalty-score, meaning a value of 1 is deemed a safe encounter and a value of 0 is deemed unsafe. The safety-metric is defined in [30] as

$$\mathcal{S}_{safety}(r_{cpa}) = \begin{cases} 1, & \text{if} \quad R_{min} \leq r_{cpa}, \\ 1 - \gamma_{nm}\left(\dfrac{R_{min} - r_{cpa}}{R_{min} - R_{nm}}\right), & \text{if} \quad R_{nm} \leq r_{cpa} < R_{min}, \\ 1 - \gamma_{nm} - \gamma_{col}\left(\dfrac{R_{nm} - r_{cpa}}{R_{nm} - R_{col}}\right), & \text{if} \quad R_{col} \leq r_{cpa} < R_{nm}, \\ 0, & \text{else}, \end{cases} \tag{2.72}$$

where R_{min} is the minimum acceptable passing distance at CPA, R_{nm} is the near-miss passing distance and R_{col} is the distance at CPA where a collision most likely would occur, meaning $R_{col} < R_{nm} < R_{min}$. A visualization of these distances is shown in Figure 2.15. The parameters γ_{nm} and γ_{col} are penalty parameters that decide the severity of the ownship passing a target ship at R_{nm} and R_{col}, respectively. It is stated in [30] that

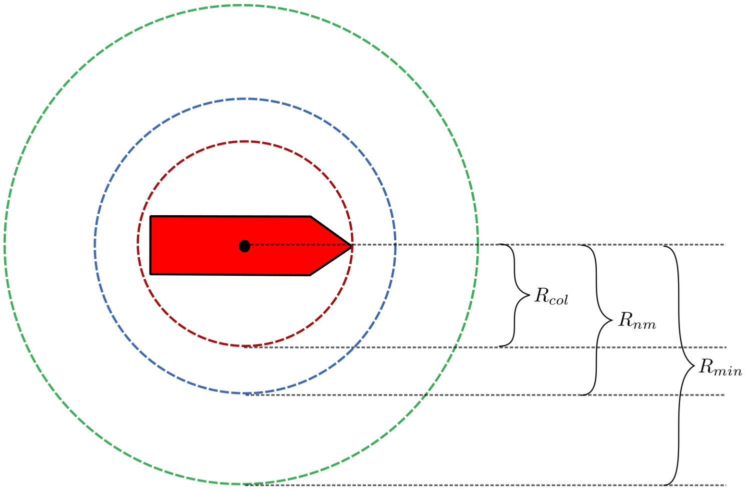


Figure 2.15: Distances used when calculating the safety-score $\mathcal{S}_{safety}$.

these parameters should be constrained by

$$\gamma_{nm} + \gamma_{col} \leq 1. \tag{2.73}$$

When calculating the safety-score, the distances shown in Figure 2.15 must be defined, and the choices must be chosen with care and has to take the context of the situation into account. Calculating the safety score of an encounter in a narrow canal where vessels are going at slow speeds requires different choices than in a situation at open sea with vessels going at higher speeds. Figure 2.16 shows an example of $\mathcal{S}_{safety}$ as a function of the range at CPA r_{cpa}, where $R_{col} = 20$m, $R_{nm} = 30$m and $R_{min} = 50$m. The different choices of γ_{nm} and γ_{col} are indicated in the figure, and shows how a higher γ_{col} penalizes the ownship passing the target ship at closer ranges more than a lower γ_{col}.

For this work, all performance metrics except the $\mathcal{S}_{safety}$-metric are characterized by a low score being preferable to a higher score. In order to avoid confusion, a new safety score is defined as

$$\mathcal{P}_{safety} = 1 - \mathcal{S}_{safety}, \tag{2.74}$$

keeping the notation of $\mathcal{P}$ denoting a penalizing metric where a score of 0 is the best outcome, and a score of 1 is the worst.

2.9.3 Metric summary

A summary of the performance metrics used in this work, along with a short description and their respective equations is given in Table 2.3.

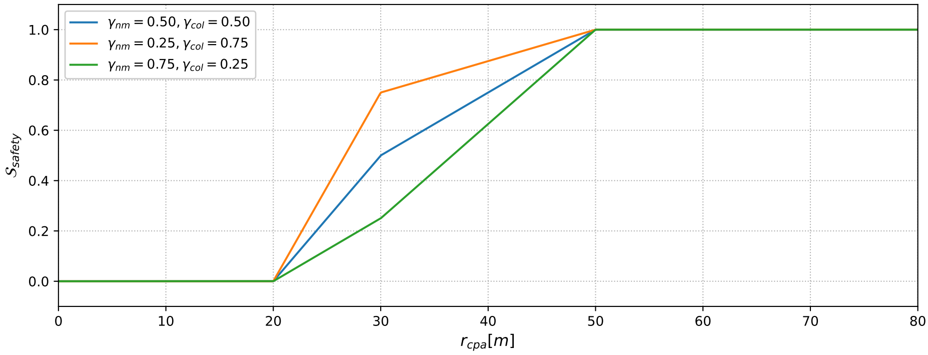


Figure 2.16: Examples of the safety-score $\mathcal{S}_{safety}$ as a function of the range at CPA r_{cpa}. The different choices of the weighting parameters γ_{nm} and γ_{col} are indicated by the colors.

Metric	Description	Equation
IAEW	Penalize control error and power consumption	(2.58)
IADC	Penalize actuator usage	(2.61)
IAEW-WT	Penalize control error, power consumption and actuator usage	(2.63)
IAA	Penalize accelerations	(2.64)
$\mathcal{P}_{delay}$	Penalize late maneuvers	(2.66)
$\mathcal{P}'_{\Delta\chi_{app}}$	Penalize non-apparent maneuvers	(2.70)
$\mathcal{P}_{safety}$	Penalize non-safe maneuvers	(2.72)

Table 2.3: Summary of performance metrics used in this work.

Chapter 3

Trajectory planning method

This chapter describes the trajectory planning methods in this work. One method is intended for low-speed transit with a fully actuated passenger ferry model, and one method is intended for high-speed transit with an underactuated vessel model. A target tracking module is responsible for estimating the trajectories of the simulated target ships, which is done based on the assumption that the target ships will keep a constant course and speed. The information about the target ships are used as input to the MPC-based trajectory planners described in this chapter, in order to calculate a trajectory that is collision-free with dynamic obstacles, and that is in compliance with a subset of the COLREGs. A map data module is responsible for downloading geospatial data describing the area of operation in the form of polygons, and is also used as input by the trajectory planners in order to ensure the resulting trajectory is collision-free with static obstacles.

The last input used by the MPC-based trajectory planner is a set of waypoints describing an initial guess of a piecewise-linear path for the ownship to follow. These waypoints can e.g. be calculated by a high-level trajectory planner, and for this work the high-level trajectory planner proposed in the preceding specialization project [20] is used. This high-level trajectory planner produces a global path from start to goal, that is collision-free with static obstacles.

Based on these inputs, the trajectory planners described in this chapter calculates optimal trajectories that are collision-free with static and dynamic obstacles, and that complies with a subset of the COLREGs. This trajectory is then in simulations used as input to an interpolation module, responsible for feeding reference signals to the vessel controllers. The context of the MPC-based trajectory planner in a larger system consisting of the interpolation module, a vessel controller and the control allocation module is shown in Figure 3.1. The internals of the MPC-based trajectory planner block in this figure is described in detail in this chapter, before it is summarized at the end of the chapter.

What is implemented as part of this work, and what was implemented prior as part of the ZeabuzSimulator, is stated clearly in the following sections, as well as in Section 1.4.

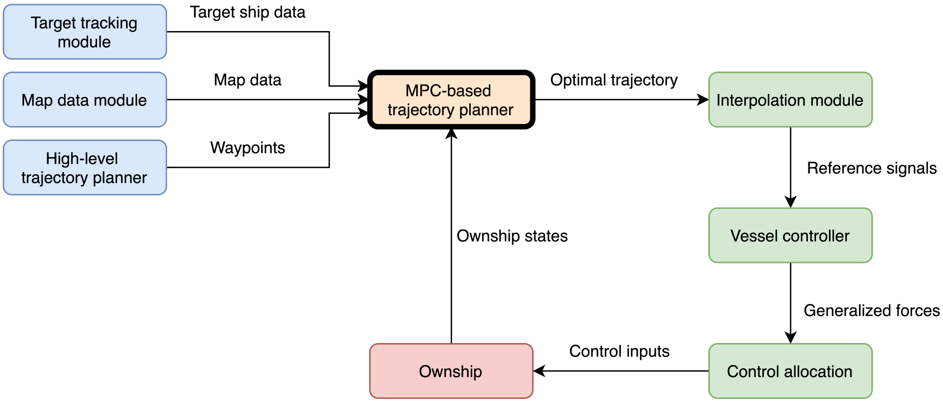


Figure 3.1: The context of the proposed MPC-based trajectory planner in a larger system, going from the inputs fed to the trajectory planner to the control inputs used by the ownship. Inspired by [18].

3.1 Software setup

The trajectory planners presented in this work are implemented using the Python programming language. The trajectory planners are based on Nonlinear MPC (NMPC), where the optimization problem is solved using the CasADi Python API, which is an open source tool for nonlinear numerical optimization and algorithmic differentiation [40][1]. It is a popular tool used for optimal control purposes, and more specifically for NMPC-applications. For this work, the IPOPT NLP-solver is used to solve the NLPs in the NMPC. CasADi contains a symbolic framework, allowing the programmer to express numerical expressions. The NMPC implementation is described in Section 3.4.

Another package used in this work is the OSMnx Python package[2], which allows for downloading map data in the form of e.g. polygons from OpenStreetMap.com. These polygons can be downloaded based on the bounding-box describing the area of interest, which are in this work used for handling static obstacles in form of land for the trajectory planners. The process of downloading geospatial data is described in Section 3.6

3.2 Otter USV model

The Otter USV model used in this work is based on a 6DOF Otter USV model implemented in the Marine Systems Simulator (MSS) Toolbox [2]. For the purpose of simplicity, the model is however reduced to 3 DOFs in this work, meaning the Otter USV is modelled according to Section 2.5. This section defining the 3DOF model of the Otter USV is based on [41], where a 6DOF model of the vessel is described in detail.

The 3DOF parameters of the Otter USV are summarized in Table 3.1, where most parameters are related to the matrices defined in Section 2.5. The exceptions are the pa-

[1] CasADi documentation: https://web.casadi.org/docs/ (Accessed 04/02/2023)

[2] OSMnx library documentation: https://osmnx.readthedocs.io/en/stable/ (Accessed (04/02/2023)

rameters related to the control allocation, namely l_1, l_2 and k_{prop}. The lever arm lengths l_1 and l_2 are based on a flat-bottomed vessel (pontoon), and the force coefficient k_{prop} is based on experimental data [2]. The lever arm lengths are used to set up the thrust configuration matrix $\boldsymbol{T}$ in (2.23), which is defined in [41] as

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1 & \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ l_1 & -l_2 \end{bmatrix}. \tag{3.1}$$

The force coefficient k_{prop} is defined for one propeller, and is used as a scalar in [2], replacing the force coefficient matrix $\boldsymbol{K}$ in (2.24).

It should also be noted that nonlinear damping is only applied to the yaw motion of the vessel, meaning all nonlinear damping terms in (2.16) are set to zero except from $N_{|r|r}$.

3.2.1 Scaled Otter USV model

In order to scale the Otter USV according to Section 2.5.1, the scaling coefficients

$$c_l = \frac{L_2}{L_1}, \tag{3.2}$$

$$c_m = \frac{m_2}{m_1}, \tag{3.3}$$

Parameter	Description	Value	Unit
L	Vessel length	2	[m]
B	Vessel beam	1.08	[m]
m	Vessel mass	55	[kg]
x_g	CG location relative to CO along x-axis	0.20	[m]
I_z	Moment of inertia about z-axis	13.75	[kg m^2]
U_{max}	Maximal speed	3.09	[m/s]
$X_{\dot{u}}$	Hydrodynamic derivative	-5.50	[kg]
$Y_{\dot{v}}$	Hydrodynamic derivative	-82.50	[kg]
$Y_{\dot{r}}$	Hydrodynamic derivative	0	[kg m]
$N_{\dot{v}}$	Hydrodynamic derivative	0	[kg m]
$N_{\dot{r}}$	Hydrodynamic derivative	-23.38	[kg m^2]
X_u	Linear damping term	-77.55	[kg/s]
Y_v	Linear damping term	0	[kg/s]
Y_r	Linear damping term	0	[kg m/s]
N_v	Linear damping term	0	[kg m/s]
N_r	Linear damping term	-37.13	[kg m^2/s]
$N_{\|r\|r}$	Nonlinear damping term	-371.25	[kg m^2/s]
l_1	Left propeller lever arm	0.39	[m]
l_2	Right propeller lever arm	0.39	[m]

Table 3.1: Otter USV model parameters. Numerical values from [2].

has to be defined. The length and mass of the original vessel, L_1 and m_1, are known through Table 3.1. The length and mass of the new vessel is based on the high-speed patrol boat described in [42], which is 15m long and has a displacement of no more than 15 tons. Using these numbers yields

$$c_l = \frac{15}{2} = 7.5, \tag{3.4}$$

$$c_m = \frac{15000}{55} \approx 272.73 \tag{3.5}$$

The parameters of the scaled Otter USV model are given in Table 3.2, where the scaling parameters are based on the theory from Section 2.5.1 and Table 2.2. Finally, it must be noted that scaling a vessel in this manner might not give a realistic scaled model, as some of the hydrodynamic forces, damping forces etc. might not be accurately scaled using the Bis system. The Bis system is however deemed the most useful scaling method for this work, as it is a commonly used method for designing ship control systems [1].

3.2.2 Otter USV motion control

A controller controlling the speed U and the heading ψ of the Otter USV is implemented in this work. As the control of the vessel is not a central topic of this work, a standard Proportional-Integral-Derivative (PID) controller is used for the yaw control, and

Parameter	Original value	Unit	Scaling	Scaled value
L	2	[m]	c_l	15
B	1.08	[m]	c_l	8.10
m	55	[kg]	c_m	15 000
x_g	0.20	[m]	c_l	1.50
I_z	13.75	[kg m^2]	$c_m c_l^2$	210 937.50
U_{max}	3.09	[m/s]	$\sqrt{c_l}$	8.46
$X_{\dot{u}}$	-5.50	[kg]	c_m	-1 500
$Y_{\dot{v}}$	-82.50	[kg]	c_m	-22 500
$Y_{\dot{r}}$	0	[kg m]	$c_m c_l$	0
$N_{\dot{v}}$	0	[kg m]	$c_m c_l$	0
$N_{\dot{r}}$	-23.38	[kg m^2]	$c_m c_l^2$	-358 593.75
X_u	-77.55	[kg/s]	$c_m/\sqrt{c_l}$	-7 722.69
Y_v	0	[kg/s]	$c_m/\sqrt{c_l}$	0
Y_r	0	[kg m/s]	$c_l\sqrt{c_l}$	0
N_v	0	[kg m/s]	$c_l\sqrt{c_l}$	0
N_r	-37.13	[kg m^2/s]	$c_m c_l^{3/2}$	-207 963.41
$N_{\lvert r \rvert r}$	-371.25	[kg m^2/s]	$c_m c_l^{3/2}$	-2 079 634.09
l_1	0.39	[m]	c_m	2.96
l_2	0.39	[m]	c_m	2.96

Table 3.2: Scaled Otter USV model parameters. c_l and c_m are given by (3.4) and (3.5), respectively.

a Proportional-Integral (PI) controller is used for the speed control. The general PI and PID control laws are defined in [1] as

$$\boldsymbol{\tau}_{PI} = -\boldsymbol{K}_p\tilde{\boldsymbol{\eta}} - \boldsymbol{K}_i \int_0^t \tilde{\boldsymbol{\eta}}(\tau)d\tau, \tag{3.6}$$

$$\boldsymbol{\tau}_{PID} = -\boldsymbol{K}_p\tilde{\boldsymbol{\eta}} - \boldsymbol{K}_d\dot{\tilde{\boldsymbol{\eta}}} - \boldsymbol{K}_i \int_0^t \tilde{\boldsymbol{\eta}}(\tau)d\tau, \tag{3.7}$$

where $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_d$ and $\boldsymbol{\eta}_d$ is the desired state. The terms $\boldsymbol{K}_p$, $\boldsymbol{K}_d$ and $\boldsymbol{K}_i$ are the proportional, derivative and integral gain, respectively. The controller used for controlling the speed U and yaw ψ of the Otter is then defined as

$$\tau_X = -k_p\tilde{U} - k_i \int_0^t \tilde{U}(\tau)d\tau, \tag{3.8}$$

$$\tau_N = -k_p\tilde{\psi} - k_d\tilde{r} - k_i \int_0^t \tilde{\psi}(\tau)d\tau, \tag{3.9}$$

respectively. The integral terms in (3.8) and (3.9) are calculated through numerical integration in the actual controller used in this work, and the references U_d and ψ_d are obtained through the target tracking guidance laws (2.38) and (2.39) from Section 2.6.3, respectively.

3.3 milliAmpere1 model

The milliAmpere1 model used in this work is based on the model identification of the milliAmpere1 ferry presented in the work by Anders A. Pedersen [43]. This model, along with a DP controller for the ferry was already implemented in the ZeabuzSimulator provided by Zeabuz at the start of this work.

3.4 Optimal control problem defintion

This section presents the OCP which is formulated and solved to find collision-free and dynamically feasible trajectories. The OCP definition and notation presented in this section is based on [18].

The OCP in this work is discretized with N_p discretization steps, and is defined in [18] as

$$\begin{aligned} \min \quad & \theta(\boldsymbol{\omega}), & (3.10)\\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{\omega}) = \boldsymbol{0}, & (3.11)\\ & \boldsymbol{h}(\boldsymbol{\omega}) \geq \boldsymbol{0}, & (3.12) \end{aligned}$$

where $\theta(\boldsymbol{\omega})$ is the objective function to be minimized, $\boldsymbol{\omega}$ is a vector containing the decision variables of the problem, defined as

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{x}_0^\top & \boldsymbol{u}_0^\top & \dots & \boldsymbol{x}_{N_p-1}^\top & \boldsymbol{u}_{N_p-1}^\top & \boldsymbol{x}_{N_p}^\top \end{bmatrix}^\top, \tag{3.13}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ are the states of the system, and $\boldsymbol{u} \in \mathbb{R}^m$ is the control input of the system, meaning $\boldsymbol{\omega} \in \mathbb{R}^{(n+m)N_p+n}$. The terms $\boldsymbol{g}(\boldsymbol{\omega})$ and $\boldsymbol{h}(\boldsymbol{\omega})$ in (3.11) and (3.12) are the sets of equality constraints and inequality constraints, respectively. The inequality constraints added to $\boldsymbol{h}(\boldsymbol{\omega})$ are presented in Section 3.6, Section 3.7 and Section 3.8.

The equality constraints $\boldsymbol{g}(\boldsymbol{\omega})$ in (3.11) are used to enforce the vessel model used internally by the NMPC-based trajectory planner. The vessel model is assumed to be on the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{3.14}$$

and two different vessel models are used in this work for two separate trajectory planners. One trajectory planner uses a linear model based on [18] described in Section 3.4.1, and this planner is intended for fully actuated vessels maneuvering at lower speeds in e.g. narrow canal environments. The other trajectory planner using a nonlinear model based on [27] is described in Section 3.4.2, and takes the heading of the vessel into account when planning trajectories. This trajectory planner is intended for underactuated vessels maneuvering at higher speeds in more open environments, where the turning rate of the vessel should be taken into account to ensure the dynamic feasibility of the trajectory. The trajectory planner intended for fully actuated, low-speed vessels is hereinafter referred to as the *low-speed trajectory planner*, and the trajectory planner intended for underactuated, high-speed vessels is referred to as the *high-speed trajectory planner*.

In order to solve the OCP given by (3.10) - (3.12), the states $\boldsymbol{x}_k$, with $k = 0, ..., N_p$ in (3.13) needs to be discretized. There are several methods for discretizing the trajectory, and for this work the direct multiple shooting method is used, similarly to [18]. The direct multiple shooting method works by splitting up the trajectory in N_p pieces at time t [44]. The trajectory will then be defined on the time horizon $t + N_p \cdot h$, where $h > 0$ is the sampling time between the discretized steps, referred to as the discretization step. At each step k of the discretized trajectory, the states $\boldsymbol{x}_{k+1}$ at step $k+1$ are calculated by integrating the vessel model based on the control input $\boldsymbol{u}_k$ at step k using some integrating function. For this work, the first order Euler integration method is used. Based on this method, the states at step $k+1$ are calculated as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h \cdot \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{3.15}$$

where $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is the system model (3.14). The right-hand side of (3.15) is then defined as the integrating function

$$\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_k + h \cdot \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{3.16}$$

meaning the model is enforced for a discretized trajectory starting at $t = t_0$ by ensuring

$$\begin{align} \boldsymbol{x}_{t_0} - \boldsymbol{x}_0 &= \boldsymbol{0}, \tag{3.17} \\ \boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0) - \boldsymbol{x}_1 &= \boldsymbol{0}, \tag{3.18} \\ &\vdots \tag{3.19} \\ \boldsymbol{F}(\boldsymbol{x}_{N_p-1}, \boldsymbol{u}_{N_p-1}) - \boldsymbol{x}_{N_p} &= \boldsymbol{0}, \tag{3.20} \end{align}$$

leading to the equality/shooting constraint being defined in [18] as

$$\boldsymbol{g}(\boldsymbol{\omega}) = \begin{bmatrix} \boldsymbol{x}_{t_0} - \boldsymbol{x}_0 \\ \boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0) - \boldsymbol{x}_1 \\ \boldsymbol{F}(\boldsymbol{x}_1, \boldsymbol{u}_1) - \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{F}(\boldsymbol{x}_{N_p-1}, \boldsymbol{u}_{N_p-1}) - \boldsymbol{x}_{N_p} \end{bmatrix}. \tag{3.21}$$

These equality-constraints were already implemented in the ZeabuzSimulator provided by Zeabuz.

In the two coming subsections, the objective functions (3.10) and vessel models (3.14) used by the two trajectory planners in this work are defined.

3.4.1 Low-speed trajectory planner using linear vessel model

The linear vessel model used in this work is the same vessel model used in [18]. The choice of vessel model used by an MPC-based trajectory planner must be done with care. It is possible to use an advanced model that is close to the actual dynamics of the vessel that is to follow the trajectory, e.g. a 3DOF model like described in Section 2.5. This would however lead to a long runtime, and the trajectory planner would likely be too slow to be used online during transit. This is why a simpler, linear model is defined in [18] as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{a} \end{bmatrix}, \tag{3.22}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} N \\ E \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} \dot{N} \\ \dot{E} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{a} = \begin{bmatrix} \ddot{N} \\ \ddot{E} \end{bmatrix}, \tag{3.23}$$

where N and E are the north and east position of the vessel respectively.

The objective function $\boldsymbol{\theta}(\boldsymbol{\omega})$ used in this trajectory planner is defined in [18] as

$$\theta_{lm}(\boldsymbol{p}, \boldsymbol{a}) = \sum_{k=0}^{N_p-1} \tilde{\boldsymbol{p}}_{k+1}^\top K_{k+1}^p \tilde{\boldsymbol{p}}_{k+1} + \boldsymbol{a}_k^\top K_k^a \boldsymbol{a}_k, \tag{3.24}$$

where the subscript lm denotes the objective function is used in the planner utilizing the *linear model* and

$$\tilde{\boldsymbol{p}}_k = \boldsymbol{p}_k - \boldsymbol{p}_{d,k} \tag{3.25}$$

is the position error to the desired position $\boldsymbol{p}_{d,k}$ at timestep k. The constants $K_k^p > 0$ and $K_k^a > 0$ are tuning parameters used for weighting the terms in the objective. Choosing a high positive value for K_k^a relative to K_k^p will then penalize accelerations more than deviations from the reference trajectory. The reference trajectory

$$\boldsymbol{x_d} = \begin{bmatrix} \boldsymbol{p}_{d,1}^\top & \boldsymbol{v}_{d,1}^\top & \boldsymbol{p}_{d,2}^\top & \boldsymbol{v}_{d,2}^\top & \cdots & \boldsymbol{p}_{d,N_p}^\top & \boldsymbol{v}_{d,N_p}^\top \end{bmatrix}^\top \tag{3.26}$$

is described in detail in Section 3.5, where $\boldsymbol{p}_{d,k}$ and $\boldsymbol{v}_{d,k}$ are the reference position and speed at timestep k, respectively.

The linear model and objective function defined in this section was already implemented in the ZeabuzSimulator when provided at the beginning of this work, as part of the basis of the low-speed trajectory planner. This trajectory planner outputs a discretized trajectory that is optimal w.r.t. the objective function (3.24) and the constraints (3.11) and (3.12), which is on the form given in [18] as

$$\boldsymbol{x}^{opt} = \begin{bmatrix} \boldsymbol{a}_0^{opt} & \boldsymbol{p}_1^{opt} & \boldsymbol{v}_1^{opt} & \boldsymbol{a}_1^{opt} & \dots & \boldsymbol{a}_{N_p-1}^{opt} & \boldsymbol{p}_{N_p}^{opt} & \boldsymbol{v}_{N_p}^{opt} \end{bmatrix}. \tag{3.27}$$

3.4.2 High-speed trajectory planner using nonlinear vessel model

As part of this work, a trajectory planner aimed towards underactuated, high-speed vessels is implemented. High-speed is in this work in the magnitude of $\approx$ 20 knots and above. An underactuated vessel has its limitations that needs to be considered when planning a trajectory, as e.g. the scaled Otter USV model used for simulations in this work will require a trajectory with more restricted turning rates in order for it to be able follow the trajectories tightly. It can also be argued that a trajectory planner for high-speed vessels should overall involve fewer large turns in order to not expend unnecessary amounts of energy. A certain level of turning is of course required to go from A to B, but the turning radius should preferably be as large as possible.

The nonlinear, kinematic model used internally by this trajectory planner takes the heading and turning rate of the vessel into account, and is defined in [27] as

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u} \tag{3.28}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} N \\ E \\ \psi \end{bmatrix} \quad \text{and} \quad \boldsymbol{u} = \begin{bmatrix} U \\ r \end{bmatrix}. \tag{3.29}$$

Here, N, E and ψ are the north and east positions, and the yaw angle of the vessel, respectively. The yaw angle is used instead of the course, meaning the vessel is assumed to have zero side-slip, leading to the yaw angle and course angle of the vessel being aligned [27]. The control input of this model is the speed derivatives defined as $\boldsymbol{a} = \begin{bmatrix} \dot{U} & \dot{r} \end{bmatrix}^\top$. This model is implemented through this work, and an objective function penalizing deviations from a reference trajectory, acceleration and turning rate is proposed. This objective function has the form

$$\theta_{nlm}(\boldsymbol{\eta}, \boldsymbol{v}, \boldsymbol{a}) = \sum_{k=0}^{N_p-1} \tilde{\boldsymbol{p}}_{k+1}^\top K_{k+1}^p \tilde{\boldsymbol{p}}_{k+1} + K_{k+1}^{tr} r^2 + K_k^a \dot{U}^2, \tag{3.30}$$

where nlm is short for nonlinear model. In this objective function, deviations from the positions of the reference trajectory are still penalized, as well as the turning rate r of the vessel and the acceleration $\dot{U}$. For this trajectory planner, the desired trajectory is on the form

$$\boldsymbol{x_d} = \begin{bmatrix}\boldsymbol{\eta}_{d,1}^\top & \boldsymbol{v}_{d,1}^\top & \boldsymbol{\eta}_{d,2}^\top & \boldsymbol{v}_{d,2}^\top & \cdots & \boldsymbol{\eta}_{d,N_p}^\top & \boldsymbol{v}_{d,N_p}^\top\end{bmatrix}^\top \tag{3.31}$$

and the output is a discretized optimal solution of the OCP given by (3.11), (3.12) and (3.30) on the form

$$\boldsymbol{x}^{opt} = \begin{bmatrix}\boldsymbol{a}_0^{opt} & \boldsymbol{\eta}_1^{opt} & \boldsymbol{v}_1^{opt} & \boldsymbol{a}_1^{opt} & \cdots & \boldsymbol{a}_{N_p-1}^{opt} & \boldsymbol{\eta}_{N_p}^{opt} & \boldsymbol{v}_{N_p}^{opt}\end{bmatrix}. \tag{3.32}$$

3.5 Reference trajectory

This section describes how the reference trajectory $\boldsymbol{x}_d$ in (3.26) and (3.31) is calculated. Both trajectory planners uses the same reference trajectory, as the information about the heading angle used in the high-speed trajectory planner can be calculated as

$$\psi_k = \text{atan2}(\dot{E}_k, \dot{N}_k), \quad k \in [1, N_p], \tag{3.33}$$

at timestep k, and the yaw rate r at timestep k is calculated as

$$r_k = \frac{\psi_k - \psi_{k-1}}{h} \quad k \in [2, N_p], \tag{3.34}$$

where h is the discretization step. The calculation of the reference trajectory is based on methodology presented in [18], and is implemented in the trajectory planner as part of this work.

The first step of calculating the reference trajectory $\boldsymbol{x}_d$ is to simulate a first-order Nomoto model through key waypoints making up the path from start to goal. These key waypoints can be provided by a high-level path planner, and for this work the high-level path planner developed during the preceding specialization project [20] is used. This path planner provides a path that is collision-free with static obstacles, based on AIS data and geospatial data.

The first-order Nomoto model used for simulating the reference trajectory is simulated with yaw-rate constraints, and is simulated through the waypoints using the LOS guidance law described in Section 2.6.2. This simulated trajectory is calculated before each NMPC-planning, and uses the initial conditions $\boldsymbol{x}_0$ of the OCP of each NMPC-run as a starting point. This simulation results in a discretized trajectory $\boldsymbol{x}_{ref}$ on the form

$$\boldsymbol{x}_{ref} = \begin{bmatrix}\boldsymbol{p}_{ref,1}^\top & \boldsymbol{v}_{ref,1}^\top & \boldsymbol{p}_{ref,2}^\top & \boldsymbol{v}_{ref,2}^\top & \cdots & \boldsymbol{p}_{ref,N_p}^\top & \boldsymbol{v}_{ref,N_p}^\top\end{bmatrix}^\top, \tag{3.35}$$

with the discretization step h between discretized steps, meaning $\boldsymbol{x}_{ref} \in \mathbb{R}^{(n+m)N_p}$.

The first NMPC-run uses $\boldsymbol{x}_{ref}$ directly as the reference trajectory $\boldsymbol{x}_d$ in (3.26) and (3.31). After this first run, $\boldsymbol{x}_d$ is however calculated slightly different. As discussed in [18], the OCP solved by the NLP-solver can result in trajectories that pass on either side of a target ship, because the OCP is non-convex for all target ships, giving an optimal solution (local minima) on either side of the target ship. To avoid paths being planned on wrong sides of the target ship w.r.t. the COLREGs, [18] proposes calculating the reference

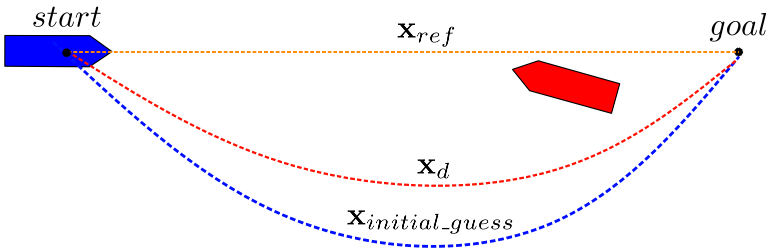


Figure 3.2: Example of how the reference trajectory $\boldsymbol{x}_d$ for the OCP is calculated based on the time-shifted and padded previous optimal trajectory $\boldsymbol{x}_{initial\_guess}$ and the reference trajectory obtained through simulating a simple kinematic model from the ownship position to goal using LOS guidance. The ownship is shown in blue, the target ship in red and the three different trajectories are indicated in the figure as well as the start and goal of the transit. These trajectories are not calculated using the planner, and are only drawn to visualize how $\boldsymbol{x}_d$ is calculated.

trajectory in a manner that increases the chances of the trajectory at least being planned on the same side in consecutive runs of the planner. This reference trajectory is planned in two steps:

1. First, a new initial guess $\boldsymbol{x}_{initial\_guess}$ of an optimal path is calculated based on the previous optimal trajectory $\boldsymbol{x}_{opt}$ calculated by the NLP-solver. The previous optimal trajectory is time-shifted by the time between NMPC-runs, denoted dt_{nmpc}. After this time-shifting, $\boldsymbol{x}_{initial\_guess}$ will be dt_{nmpc} seconds too short. This is fixed by once again simulating the first-order Nomoto model, but now starting at the end of the time-shifted previous optimal trajectory $\boldsymbol{x}_{opt}$. The end of $\boldsymbol{x}_{initial\_guess}$ is then padded by the first dt_{nmpc} seconds of this simulated trajectory. The simulated trajectory used as padding at the end of $\boldsymbol{x}_{initial\_guess}$ is also simulated with a yaw rate restriction enforced on the Nomoto model. Restricting the yaw rate of the model is especially important when generating the reference trajectory for the trajectory planner aimed towards underactuated, high-speed vessels.

2. When $\boldsymbol{x}_{initial\_guess}$ is calculated, the desired trajectory $\boldsymbol{x}_d$ is calculated as a weighted average between $\boldsymbol{x}_{ref}$ and $\boldsymbol{x}_{initial\_guess}$ as

$$\boldsymbol{x}_d = \kappa \boldsymbol{x}_{ref} + (1-\kappa)\boldsymbol{x}_{initial\_guess}, \tag{3.36}$$

   where $\kappa \in [0, 1]$ is a tuning parameter that is used to adjust the weighting between the reference trajectory $\boldsymbol{x}_{ref}$ and the time-shifted and padded previous optimal trajectory $\boldsymbol{x}_{initial\_guess}$. During an encounter, the deviation from $\boldsymbol{x}_{ref}$ should be large at some point, as the ownship shall deviate from the reference trajectory in order to avoid collision. Because of this, the κ parameter should be chosen such that $\boldsymbol{x}_d$ is calculated based on $\boldsymbol{x}_{initial\_guess}$ rather than $\boldsymbol{x}_{ref}$ in order to reduce the overall cost of the OCP solution.

The three trajectories $\boldsymbol{x}_{ref}$, $\boldsymbol{x}_{initial\_guess}$ and $\boldsymbol{x}_d$ are visualized in Figure 3.2. This figure shows an example of how the reference trajectory $\boldsymbol{x}_d$ might be calculated based on the simulated trajectory $\boldsymbol{x}_{ref}$ and the time-shifted and padded previous optimal trajectory

$\boldsymbol{x}_{initial\_guess}$. Although this figure is drawn based on a fictional scenario and not actually calculated during simulations, this example is meant to show how $\boldsymbol{x}_d$ would be calculated using a $\kappa > 0.5$, meaning the $\boldsymbol{x}_{initial\_guess}$ trajectory is weighted more than $\boldsymbol{x}_{ref}$ in the weighted average (3.36).

As mentioned, this way of calculating the reference trajectory is proposed in [18] as a way of reducing the risk of the solution trajectory fluctuating between local minimas on either side of the target ship. The method is however dependent on the correct side of the target ship being chosen during the initial planning, as it effectively strengthens the previous guess of an optimal trajectory. A way of further enforcing the trajectory planner to plan trajectories on the correct side of the target ship is proposed in Section 3.8.3.

3.6 Static obstacle constraints

This section describes how the constraints for the static obstacles are formulated, and is based on [45] and [18]. The inclusion of static obstacles in the trajectory planner is implemented as part of this work, which also makes it possible to simulate complex scenarios with static obstacles using the ZeabuzSimulator.

When planning a trajectory for a vessel maneuvering on water there are multiple obstacles that need to be considered by the trajectory planner in order to avoid collision. One class of obstacles are static obstacles. Static obstacles are obstacles that are completely still, e.g. land surrounding the water or islands. Quasi-static obstacles can also be added to this group of obstacles, as these are obstacles that are moving so slowly that they can be considered static in a practical sense. Such quasi-static obstacles can be moored ships or floating docks/piers.

Static obstacles are in this work represented by polygons. For moored ships, these polygons can be generated based on information from AIS data, and other obstacles such as land, islands and piers can be represented based on map data. One way of downloading map data when working with Python is by using the OSMnx library. Using this library, polygons can be downloaded based on the bounding-box describing the area of interest. This bounding-box is defined as

$$bbox = \begin{bmatrix} lat_{max} & lat_{min} & lon_{max} & lon_{min} \end{bmatrix}, \tag{3.37}$$

where lat_{max} and lat_{min} are the northern and southern latitude of the area respectively, and lon_{max} and lon_{min} are the eastern and western longitude of the area respectively. An example of of how polygons can be downloaded is shown in Listing 1. In this example, the different types of area of interest are listed in the `tags`-dictionary. It is also possible to download all polygons in the area, but for the sake of reducing the amount of data the trajectory planner has to consider, filtering out polygons not of interest is reasonable. The different types of areas used by OpenStreetMap (`tags` in Listing 1) are available through the OpenStreetMap wiki-pages [3].

When all polygons representing static- and quasi-static obstacles are available, the constraints used by the trajectory planner can be defined. For this work this is done based

[3] Different types of areas used by OpenStreetMap: https://wiki.openstreetmap.org/wiki/Map_features (Accessed 06/03/2023)

```
import osmnx as ox
polys = ox.geometries_from_bbox(
    lat_max, lat_min,
    lon_max, lon_min,
    tags={
    'place'      : ['island', 'islet'],
    'natural'    : ['water', 'bay'],
    'man_made'   : ['pier', 'breakwater'],
    'landuse'    : ['retail', 'industrial']
    })
```

Listing 1: Example of Python code for extracting geometries from OpenStreetMap using the Python library OSMnx.

on methods described in [45] and [18], where a safe operating region for the vessel is defined in real time based on the ownship position $\boldsymbol{p}_{os}$. First, the area surrounding the ownship is divided into N_{sect} equally sized sectors of a circle with radius r. The radius of the circle works as a limitation on the distance within which points are considered. A set $\mathcal{S}_i, i \in [1, ..., N_{sect}]$ is then defined for each sector, and contains the exterior points of all polygons in the respective sector. The goal is then to find the point $\boldsymbol{p}_{closest,i}$ within each set $\mathcal{S}_i$ that is closest to the ownship position $\boldsymbol{p}_{os}$. This is done based on

$$\boldsymbol{p}_{closest,i} = \text{argmin}_{\boldsymbol{p} \in \mathcal{S}_i} \|\boldsymbol{p} - \boldsymbol{p}_{os}\|_2. \tag{3.38}$$

When the closest point in each sector is defined, the constraints can be defined. In this work, these constraints are calculated as the lines that are tangent to each of the closest points on circles with radius d_i, where d_i is the distance from $\boldsymbol{p}_{os}$ to $\boldsymbol{p}_{closest,i}$. The tangent line of a circle with radius d_i passing through $\boldsymbol{p}_{closest,i}$ is then satisfied by all points $\boldsymbol{p}$ that satisfy the equation given in [45] as

$$\underbrace{\frac{(\boldsymbol{p}_{closest,i} - \boldsymbol{p}_{os})^\top}{\|\boldsymbol{p}_{closest,i} - \boldsymbol{p}_{os}\|_2}}_{\boldsymbol{A}_i} \boldsymbol{p} = \underbrace{\frac{(\boldsymbol{p}_{closest,i} - \boldsymbol{p}_{os})^\top \boldsymbol{p}_{closest,i}}{\|\boldsymbol{p}_{closest,i} - \boldsymbol{p}_{os}\|_2}}_{b_i}, \tag{3.39}$$

and in a more compact form as

$$\boldsymbol{A}_i \boldsymbol{p} = b_i. \tag{3.40}$$

An inner convex, obstacle free space for the ownship to maneuver safely within is then formed by all points $\boldsymbol{p}$ that satisfy

$$\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_{N_{sect}} \end{bmatrix} \boldsymbol{p} \leq \begin{bmatrix} b_1 - d_{stat} \\ b_2 - d_{stat} \\ \vdots \\ b_{N_{sect}} - d_{stat} \end{bmatrix}, \tag{3.41}$$

where $d_{stat} \geq 0$ is an optional safety distance added such that the ownship keeps a safety-distance to static obstacles. This will effectively shrink the convex, obstacle free space, and has to be done carefully in order to avoid restricting the space too much. The static constraints are added to the inequality constraints (3.12) at timestep k as

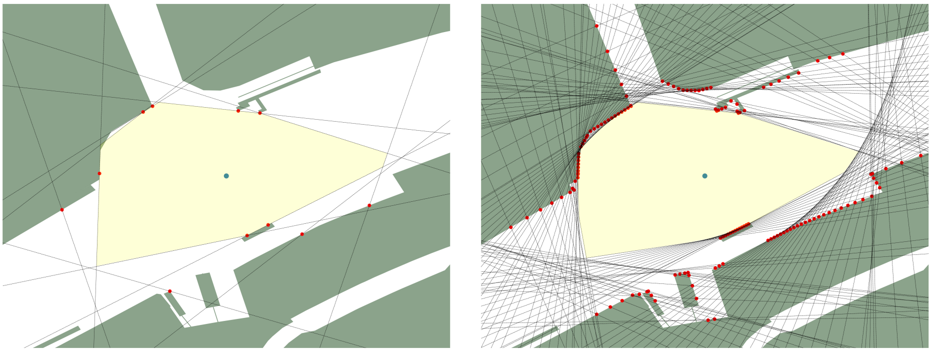
$$\boldsymbol{h}_{stat}(\boldsymbol{p}_k) = \begin{bmatrix} b_1 - d_{stat} - \boldsymbol{A}_1\boldsymbol{p}_k \\ b_2 - d_{stat} - \boldsymbol{A}_2\boldsymbol{p}_k \\ \vdots \\ b_{N_{sect}} - d_{stat} - \boldsymbol{A}_{N_{sect}}\boldsymbol{p}_k \end{bmatrix}, \tag{3.42}$$

where $\boldsymbol{p}_k$ is calculated directly from the desired position $\boldsymbol{p}_{d,k}$ when using the low-speed trajectory planner, and $\boldsymbol{p}_k$ can be found as the N and E entries in the desired pose $\boldsymbol{\eta}_{d,k}$ when using the high-speed trajectory planner.

During transit, the constraints are added by calculating the constraints based on the positions $\boldsymbol{p}_{os}$ in the reference trajectory $\boldsymbol{x}_d$ described in Section 3.5. Adding the constraints for static obstacles will increase the runtime of the overall trajectory planning. One can add the constraints at each timestep $k \in [0, N_p]$ of the reference trajectory, but this will lead to increased runtime. To mitigate this, the constraints are only formulated for each dt_{stat}'th discretization step in the planning horizon. E.g. with $dt_{stat} = 5$, which with a discretization step of $h = 1$s leads to the static obstacle constraints being defined each 5'th second during planning. Choosing a too large dt_{stat} and h can lead to collisions with static obstacles, as the static obstacles will then be defined too seldom during the planning horizon of N_p steps. Choosing a too small value for dt_{stat} will on the other hand lead to an unnecessarily high number of constraints, and will give increased runtime. It must however be underlined that safety is of highest priority. The implementation used in this work is based on the Python programming language, and if the trajectory planner is written in e.g. C++, the runtime will likely be significantly decreased, meaning a lower value for dt_{stat} is preferred for all practical purposes in order to make sure the static constraints are defined *often enough*.

Two examples using $N_{sect} = 12$ and $N_{sect} = 180$ are shown in Figure 3.3, where the ownship position $\boldsymbol{p}_{os}$ is shown as a blue dot and the closest points $\boldsymbol{p}_{closest,i}$ are shown as red dots. The tangent lines calculated using (3.39) are shown as black dashed lines. This figure illustrates that $N_{sect} = 12$ is a sufficient amount of sectors for the purpose of this work.

In [18], the lines are chosen to be tangent to an ellipse with its major axis aligned with the desired course at position $\boldsymbol{p}_{os}$. This effectively opens up the convex, obstacle free space in the desired direction along the path. Using a circle instead of an ellipse poses the possibility of restricting the inner convex, obstacle free space too much, but for this work it is deemed sufficient if the constraints are generated based on sufficiently close positions along a reference trajectory for the MPC planner. Also, choosing an elliptic shape instead of a circular might make the planning space too narrow in e.g. narrow canal environments. There are in other words pros and cons with both approaches, and the operational environment should be taken into account when deciding what is best.



(a) $N_{sect} = 12$.

(b) $N_{sect} = 180$.

Figure 3.3: Static obstacle constraints for $N_{sect} = 12$ and $N_{sect} = 180$ to calculate the closest points to the ownship. The ownship is shown as the blue dot, the red dots are the closest points to the ownship in each sector and the lines forming the constraints are shown as black dashed lines passing through the closest points. The yellow area surrounding the ownship is the inner convex, obstacle free space formed by the constraints.

3.7 Dynamic obstacle constraints

This section describes how constraints are defined based on dynamic obstacles in order to avoid collision with other vessels in a way that complies with the COLREGs Rules 13-15 and Rule 17. The key idea of this methodology is assigning a domain to each target ship, a TS domain, that is designed such that as long as the planned trajectory does not enter the TS domain, the trajectory complies with Rules 13-15 and partially with Rule 17. This TS domain is described in [18] and was already implemented in the trajectory planner handed out as part of the ZeabuzSimulator at the beginning of this work.

The TS domain described in [18] is a straight line that divides the plane in half, and is at its core based on two parameters; l and α. The l-parameter is the shortest distance from the target ship to the TS domain, and $\alpha \in (-\pi, \pi]$ is the angle of a normal vector from the target ship to the closest point on the domain. The shortest distance from the target ship to the TS domain works as a minimum distance for the ownship to keep away from the target ship, and should be chosen with care in order to both keep a reasonable distance to the target ship, while still ensuring there is enough space for the trajectory planner to plan a feasible trajectory.

In order to define α, the side of the target ship that the ownship shall pass on needs to be determined. This is done by defining a port-starboard split angle, defined in [18] as

$$\alpha_s = \alpha_{v_{rel}} + \alpha_{\delta_s}, \tag{3.43}$$

where $\alpha_{v_{rel}}$ is the angle of the relative velocity vector

$$\boldsymbol{v}_{rel} = \boldsymbol{v}_{TS} - \boldsymbol{v}_{OS}, \tag{3.44}$$

where $\boldsymbol{v}_{OS} = \begin{bmatrix} \dot{N}_{OS} & \dot{E}_{OS} \end{bmatrix}^\top$ and $\boldsymbol{v}_{TS} = \begin{bmatrix} \dot{N}_{TS} & \dot{E}_{TS} \end{bmatrix}^\top$ are the north-east velocities of the ownship and target ship, respectively. The angle $\alpha_{v_{rel}}$ is then defined as

$$\alpha_{v_{rel}} = \text{atan2}(\dot{E}_{rel}, \dot{N}_{rel}). \tag{3.45}$$

The α_{δ_s} parameter is an encounter-specific tuning parameter used to enforce maneuvers at the correct side of the target ship w.r.t. the COLREGs. The angle α is then defined in [18] as

$$\alpha = \begin{cases} \varphi_{TS} + \alpha_d, & \text{if} \quad \varphi_{TS} > \alpha_s, \\ \varphi_{TS} - \alpha_d, & \text{else,} \end{cases} \tag{3.46}$$

where α_d is another encounter-specific parameter used to make the ownship pass the target ship in a COLREGs-compliant manner. The choice of adding or subtracting α_d in (3.46) is based on whether the ownship shall pass on the starboard or port side of the target ship, where

$$\varphi_{TS} > \alpha_s \quad \Rightarrow \quad \text{pass on port side of the target ship,} \tag{3.47}$$

$$\varphi_{TS} \leq \alpha_s \quad \Rightarrow \quad \text{pass on starboard side of the target ship.} \tag{3.48}$$

An example of how the TS domain is set up for a head-on encounter is shown in Figure 3.4. The points $\boldsymbol{p}_B$ and $\boldsymbol{p}_D$ are the points that define the constraint added to the inequality constraints (3.12). The point $\boldsymbol{p}_D$ is the point on the TS domain boundary that is closest to the target ship, and is calculated in [18] using the normal vector on the domain boundary pointing out of the domain

$$\boldsymbol{n}_D = \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}, \tag{3.49}$$

and the point $\boldsymbol{p}_D$ is then defined in [18] as

$$\boldsymbol{p}_D = \boldsymbol{p}_{TS} + \boldsymbol{n}_D \cdot l. \tag{3.50}$$

Based on this point, the constraint $h_{TS}(\boldsymbol{p}, \boldsymbol{p}_{TS})$ is then defined such that the distance between the ownship and the point $\boldsymbol{p}_B$ is always > 0. The point $\boldsymbol{p}_B$ is calculated in [18] as

$$\boldsymbol{p}_B = \boldsymbol{p}_D + (\boldsymbol{p} - \boldsymbol{p}_D)^\top \boldsymbol{n}_{p_d} \boldsymbol{n}_{p_d}, \tag{3.51}$$

where $\boldsymbol{n}_{p_d}$ is the tangent vector to the domain boundary

$$\boldsymbol{n}_{p_d} = \begin{bmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{bmatrix}. \tag{3.52}$$

The constraint $h_{TS}(\boldsymbol{p}, \boldsymbol{p}_{TS})$ added to (3.12) is then defined in [18] as

$$h_{TS}(\boldsymbol{p}, \boldsymbol{p}_{TS}) = \boldsymbol{n}_D^\top (\boldsymbol{p} - \boldsymbol{p}_B). \tag{3.53}$$

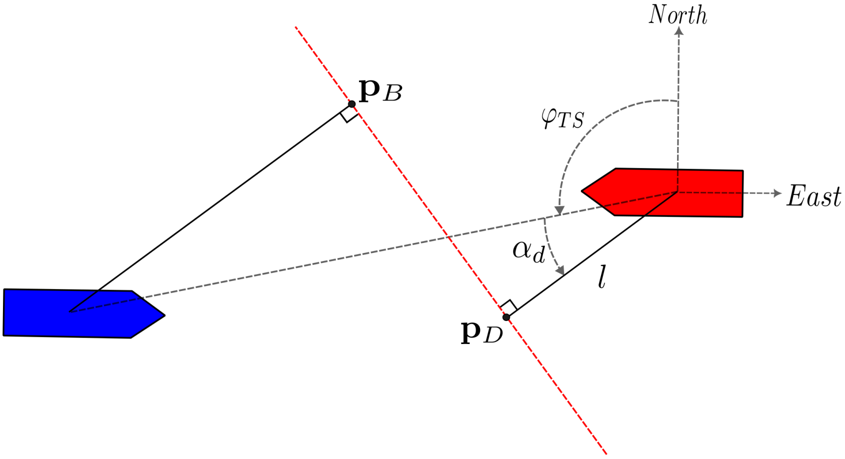


Figure 3.4: Example of how the TS domain is set up for a head-on encounter. The ownship is shown in blue, the target ship is shown in red and the boundary of the TS domain is shown as a red dashed line. Inspired by [18].

3.8 Enforcing compliance with Rule 8 and Rule 16

This section describes two tools that can be used to enforce compliance with Rule 8 and Rule 16 of the COLREGs. One of the tools is windows of reduced cost in the planning horizon of the MPC, and the tool proposed in this work extends methodology described in [18]. The other tool proposed in this work is a simple linear constraint placed on the port side of the target ship, that can be used to enforce readily apparent maneuvers made in ample time. Both these tools are implemented as part of this work. Before introducing the tools, the way in which the target ships are prioritized during the planning horizon is defined based on [18].

3.8.1 Adding and prioritizing target ships

When planning a trajectory over a horizon of N_p discretized steps in the presence of one or more target ships, there needs to be some way of prioritizing the target ships. For this work, this is done based on the estimated time it takes for the ownship to enter and exit a critical distance at closest point of approach (DCPA) of the target ship, similarly to what is done in [18]. This critical DCPA is denoted $d_{crit} > 0$, and the time it takes for the ownship to enter the d_{crit} of the target ship is hereinafter denoted t_{crit}^{enter}, and the time it takes to exit d_{crit} is denoted t_{crit}^{exit}. Similarly to what is done in [18], target ships are in this work considered by the trajectory planner if they satisfy the entry criteria

$$t_{crit}^{exit} < N_p h - t_{after\_passing}, \tag{3.54}$$

where $N_p h$ is the duration of the planning horizon in seconds, and $t_{after\_passing}$ is a time parameter that ensures target ships are not considered if there is not enough time for planning left of the planning horizon after the encounter. This is first and foremost a safety measure, such that the trajectory planner does not end up planning a trajectory blindly without sufficient information about the situation at hand.

The only exemption from the entry criteria (3.54) are encounters that are classified as stand-on encounters. In these cases, the ownship shall stand on by keeping its speed and course, and the target ship is obliged to give way for the ownship. In practice, this means that the trajectory planner won't have to plan any maneuver in order to give way for the target ship. There is however the possibility that the target ship does not give way, meaning the trajectory planner shall at some point start considering the target ship and give way in order to avoid collision. The entry criteria for target ships in situations where the ownship has a stand-on role is inspired by [18], and is given as

$$t^{enter}_{crit,stand\_on} < t_{stand\_on\_reaction\_time}, \tag{3.55}$$

where $t^{enter}_{crit,stand\_on} \leq t^{enter}_{crit}$ is the time it takes for the ownship to enter a critical DCPA of the target ship in stand-on encounters, $d_{crit\_stand\_on} \leq d_{crit}$. The $t_{stand\_on\_reaction\_time}$ parameter is another time parameter that decides the minimum amount of time the ownship will need to make a maneuver in order to avoid collision. The safest is to set $d_{crit\_stand\_on} = d_{crit}$, meaning $t^{enter}_{crit,stand\_on} = t^{enter}_{crit}$. This might however cause problems if $t_{stand\_on\_reaction\_time}$ is too short. Too large values of $d_{crit\_stand\_on}$ and $t_{stand\_on\_reaction\_time}$ can cause the ownship to maneuver too early, making it breach its stand-on obligations. Too small values can on the other hand lead to unwanted close quarter situations, which is unsafe for the passengers onboard. Setting these parameters is in other words not trivial, and has a close connection to Rule 2 of the COLREGs discussed in Section 2.1.1. Too large values can lead to breaching Rule 17 regarding stand-on vessels, while still adhering to Rule 2 that allows for a departure from the rest of the COLREGs in order to avoid collision. Too small values can on the other hand lead to behaviour that is more in line with Rule 17, while leading to unnecessarily close encounters that might be dangerous or at least uncomfortable for the passengers onboard.

If there are multiple target ships in the planning horizon, the trajectory planner will in this work prioritize the vessel with the smallest t^{enter}_{crit}, which is likely to be the encounter that requires action first. The set of all target ships considered by the trajectory planner based on (3.54) and (3.55) is hereinafter denoted $\mathcal{N}_{pri}$.

3.8.2 Windows of reduced cost

Windows of reduced cost is an idea presented in [18], and is further developed in this work. The idea is to assign windows of reduced cost in the planning horizon of the trajectory planners. These windows are defined such that the weighting terms in (3.24) and (3.30) are reduced during a portion of the planning horizon, giving the trajectory planner more freedom to e.g. accelerate during the window if the cost associated to acceleration is reduced. In [18], these windows are defined such that the cost associated to acceleration is reduced for a short period in ample time before an encounter between the target ship and ownship. The reduction of the cost associated to deviations from the desired trajectory $\boldsymbol{x}_d$ starts at the same time as the cost associated to the acceleration, but lasts longer. This effectively incentivizes the trajectory planner to perform much of the acceleration early in the encounter, while giving it more freedom to deviate from the reference trajectory in order to avoid collision.

For this work, the windows of reduced cost are split up into two parts, CRW1 and CRW2, where CRW is short for cost-reduction window. The idea behind splitting up the windows into two parts is that these two parts are defined based on the same set of parameters, regardless of context. It is then possible to reduce an arbitrary cost parameter differently in CRW1 and CRW2, for an arbitrary number of cost parameters in the cost function. This makes the windows of reduced cost a tool that can be used in any MPC-application where such functionality is useful, as the parameters used to define the windows can be defined based on the domain of the MPC-application. In this work, this functionality is useful in order to incentivize the trajectory planner to plan readily apparent maneuvers in ample time. This is achieved through e.g. reducing the cost associated to acceleration more during CRW1 relative to CRW2. This incentivizes the trajectory planner to plan most of the maneuver during CRW1, as accelerating during CRW1 costs less than accelerating during CRW2. The way the windows are set up in this work has a similar effect as the method described in [18] when the low-speed trajectory planner using the linear model (3.22) is used. The effect is however better seen in objective functions with more cost parameters, as is the case for the high-speed trajectory planner using the nonlinear model (3.28). Before defining these windows, they can be summarized as a tool that makes it possible to incentivize different types of behaviour in different parts of the planning horizon in MPC applications.

In order to define the cost-reduction windows, some reduction-coefficients are defined. These are defined as K^p_{CRW1}, K^p_{CRW2}, K^a_{CRW1}, K^a_{CRW2}, K^{tr}_{CRW1} and K^{tr}_{CRW2}, and all reduction-coefficients can take values in the interval $[0, 1]$. The coefficients for CRW2 will also typically be assigned such that e.g. $K^a_{CRW2} > K^a_{CRW1}$. The two windows of reduced cost are then defined by the start and end of the windows, denoted $CRW1_{start}$, $CRW1_{end}$, $CRW2_{start}$ and $CRW2_{end}$. Using the reduction-coefficients and the intervals defined above, the cost gains in (3.24) and (3.30) when using windows of reduced cost are defined as

$$K^p_k = \begin{cases} K^p_k \cdot K^p_{CRW1}, & \text{if} \quad kh \in [CRW1_{start}, CRW1_{end}], \\ K^p_k \cdot K^p_{CRW2}, & \text{if} \quad kh \in [CRW2_{start}, CRW2_{end}], \\ K^p_k, & \text{else}, \end{cases} \tag{3.56}$$

for the cost associated to the deviation from the reference trajectory,

$$K^a_k = \begin{cases} K^a_k \cdot K^a_{CRW1}, & \text{if} \quad kh \in [CRW1_{start}, CRW1_{end}], \\ K^a_k \cdot K^a_{CRW2}, & \text{if} \quad kh \in [CRW2_{start}, CRW2_{end}], \\ K^a_k, & \text{else}, \end{cases} \tag{3.57}$$

for the cost associated to the absolute acceleration, and finally

$$K^{tr}_k = \begin{cases} K^{tr}_k \cdot K^{tr}_{CRW1}, & \text{if} \quad kh \in [CRW1_{start}, CRW1_{end}], \\ K^{tr}_k \cdot K^{tr}_{CRW2}, & \text{if} \quad kh \in [CRW2_{start}, CRW2_{end}], \\ K^{tr}_k, & \text{else}, \end{cases} \tag{3.58}$$

for the cost associated to the turning rate when using the high-speed trajectory planner. The k and h parameters in (3.56) - (3.58) denotes the discretized step of the planning horizon, and the discretization step, respectively.

In order to define the starting point and duration of the cost-reduction windows, two parameters needs to be defined. The first window should in compliance with Rule 8 start in *ample time* before an encounter. As argued earlier, what is ample time is a hard parameter to set, as it is highly dependent on the context of the situation. For this work, it is however proposed to set this as

$$t_{ample\_time} = 0.8 \cdot t_{crit}^{enter,min}, \tag{3.59}$$

where $t_{crit}^{enter,min}$ is the minimum t_{crit}^{enter} across all target ships considered by the trajectory planner,

$$t_{crit}^{enter,min} = \min_{t_{crit}^{enter}} (\mathcal{N}_{pri}). \tag{3.60}$$

The first window of reduced cost, CRW1, is then chosen to start at

$$CRW1_{start} = t_{crit}^{enter,min} - t_{ample\_time} = 0.2 \cdot t_{crit}^{enter,min}. \tag{3.61}$$

The rationale behind choosing t_{ample_time} to be 80% of t_{crit}^{enter} is to give the trajectory planner some time to plan after these cost windows are suddenly introduced in the planning horizon, and is based on results from simulations performed early during this work. It is also chosen such that a maneuver is started fairly soon after a situation involving a possible encounter is discovered. The end of the first cost-reduction window is chosen to be at a time

$$CRW1_{end} = CRW1_{start} + t_{maneuver}, \tag{3.62}$$

where $t_{maneuver}$ is a short period of time meant to give the vessel some time to perform the maneuver. This is a parameter that has to be set w.r.t. the context, and shall e.g. be shorter in narrow canals than in open waters. A shorter $t_{maneuver}$ will also potentially lead to a more aggressive maneuver relative to a longer one.

The second window of reduced cost is defined to start after the first window ends, meaning

$$CRW2_{start} = CRW1_{end} + h. \tag{3.63}$$

This window then lasts until the planned trajectory has exited the critical DCPA to the target ship with the largest t_{crit}^{exit}, meaning

$$CRW2_{end} = t_{crit}^{exit,max}, \tag{3.64}$$

where

$$t_{crit}^{exit,max} = \max_{t_{crit}^{exit}} (\mathcal{N}_{pri}). \tag{3.65}$$

It is possible to add a time-period after t_{crit}^{exit} the second window of reduced cost as well, but for this work it is deemed reasonable to return to the original costs in (3.24) and (3.30)

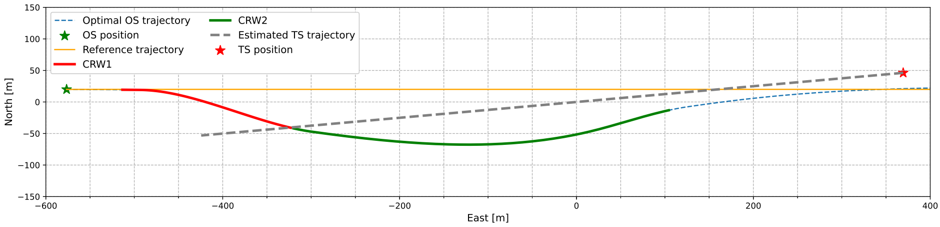


Figure 3.5: Illustration of windows of reduced cost in the planning horizon in a head-on encounter. The ownship is the left-most vessel denoted by a green star, and the target ship is shown to the right as a red star. The grey line stretching from the target ship is the predicted trajectory of the target ship, and the orange line stretching from the ownship is the reference trajectory $\boldsymbol{x}_{ref}$ produced by simulation of a kinematic model using LOS guidance. The optimal trajectory calculated by the trajectory planner is the line that is partly red, green and blue. The red portion of this line is the first window of reduced cost, CRW1, and the green portion is the second window of reduced cost, CRW2.

straight after the encounter is finished in order to converge back onto the reference trajectory once the encounter is over. To summarize, the cost reduction windows are assigned as

$$\text{CRW1} = \{t \quad : \quad t \in [0.2 \cdot t_{crit}^{enter,min}, \quad 0.2 \cdot t_{crit}^{enter,min} + t_{maneuver}]\}, \tag{3.66}$$

$$\text{CRW2} = \{t \quad : \quad t \in (0.2 \cdot t_{crit}^{enter,min} + t_{maneuver}, \quad t_{crit}^{exit,max}]\}. \tag{3.67}$$

An example of how the windows of reduced cost looks during the planning horizon is shown in Figure 3.5.

3.8.3 Port-side constraint

Another tool for enforcing readily apparent maneuvers made in ample time in head-on encounters is proposed in this work. This tool is a constraint that only applies during the second window of reduced cost, i.e. when $kh \in [CRW2_{start}, CRW2_{end}]$. This constraint is a straight line that is parallel to the course of the target ship in head-on encounters. The line is placed a distance d_{port} from the port side of the target ship, and is calculated using the estimated heading of the target ship ψ_{TS} and the distance d_{port}. First, the normal vector pointing out from the port side of the target ship is calculated as

$$\boldsymbol{n}_{port} = \begin{bmatrix} -\cos(\psi_{TS}) \\ \sin(\psi_{TS}). \end{bmatrix}, \tag{3.68}$$

A point that is d_{port} to the port side of the target ship is then calculated as

$$\boldsymbol{p}_{port} = \boldsymbol{p}_{TS} + d_{port}\boldsymbol{n}_{port}. \tag{3.69}$$

The line forming the constraint is then given by all points $\boldsymbol{p}$ that satisfies

$$\frac{\left(\boldsymbol{p}_{port} - \boldsymbol{p}_{TS}\right)^{\top}}{\|\boldsymbol{p}_{port} - \boldsymbol{p}_{TS}\|^2}\boldsymbol{p} = \frac{\left(\boldsymbol{p}_{port} - \boldsymbol{p}_{TS}\right)^{\top}\boldsymbol{p}_{port}}{\|\boldsymbol{p}_{port} - \boldsymbol{p}_{TS}\|^2}. \tag{3.70}$$

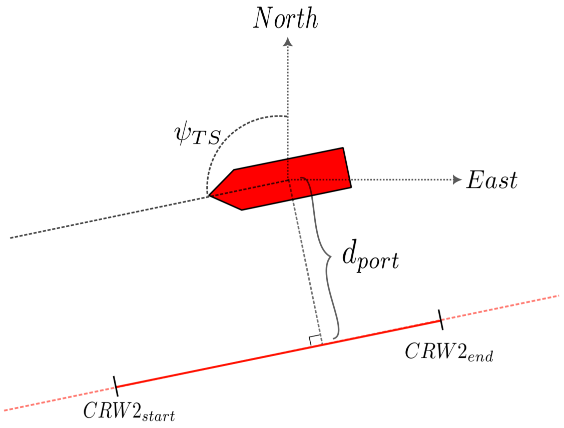


Figure 3.6: Visualization of the constraint used in head-on encounters to enforce readily apparent maneuvers made in ample time. The target ship is shown in red, and the constraint is shown as the red line south of the target ship. $CRW2_{start}$ and $CRW2_{end}$ are points in time that denote the start and end of the second window of reduced cost, respectively.

This calculation is similar to the calculation of the tangent lines to the closest points when defining the constraints for the static obstacles. The difference is that in this case, the ownship shall be outside of this line relative to the target ship. This means the constraint are all points $\boldsymbol{p}$ that satisfies

$$\underbrace{\frac{(\boldsymbol{p}_{port}-\boldsymbol{p}_{TS})^{\top}}{\|\boldsymbol{p}_{port}-\boldsymbol{p}_{TS}\|^2}}_{\boldsymbol{A}_{port}}\boldsymbol{p} \geq \underbrace{\frac{(\boldsymbol{p}_{port}-\boldsymbol{p}_{TS})^{\top}\boldsymbol{p}_{port}}{\|\boldsymbol{p}_{port}-\boldsymbol{p}_{TS}\|^2}}_{b_{port}}, \tag{3.71}$$

or in more compact form

$$\boldsymbol{A}_{port}\boldsymbol{p} \geq b_{port} \tag{3.72}$$

and the constraint added to (3.12) is defined as

$$h_{port}(\boldsymbol{p},\boldsymbol{p}_{TS}) = \boldsymbol{A}_{port}\boldsymbol{p} - b_{port}. \tag{3.73}$$

The constraint is visualized in Figure 3.6 for a head-on encounter.

The rationale behind this constraint is that it directly dictates what should be done during the first window of reduced cost. As the line is pushed further to the port side of the target ship by increasing d_{port}, the trajectory planner will be forced to accelerate more and earlier during the first window of reduced cost, in order to satisfy this constraint during the second window of reduced cost. This is however a rather strict constraint, as the constraint can be compared to a rubber band. As d_{port} is increased, the rubber band is stretched further. The rubber band will however snap at some point. In this context, this snap will occur if $t_{maneuver}$ is too short and d_{port} is too large. This will lead to the trajectory planner being unable to plan a trajectory that satisfies the constraint at the beginning of the second

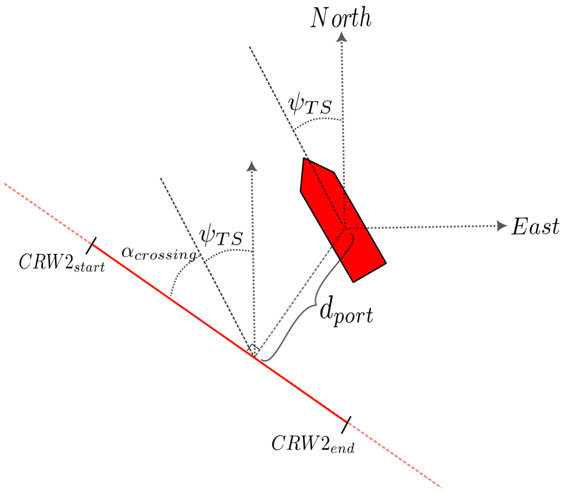


Figure 3.7: Visualization of the constraint used in give-way crossing encounters. The target ship is shown in red, and the constraint is shown as the red line south of the target ship. $CRW2_{start}$ and $CRW2_{end}$ are points in time that denote the start and end of the second window of reduced cost, respectively.

window of reduced cost, without getting an unfeasible acceleration during the first window of reduced cost. The distance d_{port} must therefore be chosen quite conservatively, and in this work it is set to be a fraction of the minimum distance to keep from the target ship, meaning the l-parameter used when defining the TS domain. Choosing e.g. $d_{port} = 0.5 \cdot l$ leads to a less strict constraint, while still possibly enforcing an early and readily apparent maneuver.

The constraint is also added in crossing encounters where the ownship has a give-way role, but for another reason than enforcing early and readily apparent maneuvers. As mentioned in Section 3.5, the trajectory planner has the potential of choosing the wrong side for passing the target ship. The constraint can therefore be seen as an additional insurance that the correct side is chosen in head-on encounters by adding it like described above. This is also the case for give-way crossing encounters, but for these encounters the normal vector pointing out from the port side is calculated as

$$\boldsymbol{n}_{port} = \begin{bmatrix} -\cos(\psi_{TS} - \alpha_{crossing}) \\ \sin(\psi_{TS} - \alpha_{crossing}), \end{bmatrix}, \tag{3.74}$$

where $\alpha_{crossing}$ is a small angle that makes sure the line is not parallel to the target ship, but instead angled such that the trajectory planner plans a trajectory abaft the target ship in crossing encounters where the ownship has the target ship on its starboard side. This can be seen as a way of further enforcing compliance with Rule 15 of the COLREGs regarding crossing situations. The way the port-side constraint is added in give-way crossing encounters is shown in Figure 3.7.

The port-side constraint as described in this section is only added to the low-speed trajectory planner in this work. During simulations, the constraint was however seen to cause problems in higher speeds when using the high-speed trajectory planner. This led to the constraint only being added to the high-speed trajectory planner halfway into the second window of reduced cost, i.e. when

$$CRW2_{start} + 0.5(CRW2_{end} - CRW2_{start}) \leq k \cdot h \leq CRW2_{end}. \tag{3.75}$$

This means that for the high-speed trajectory planner, the port-side constraint is mostly added in order to enforce the trajectory planner to plan trajectories on the correct side of target ships.

3.9 Trajectory planning pipeline summary

The input to the NMPC-based trajectory planner described in this chapter is map data describing the static obstacles in the form of land, data describing dynamic obstacles from a module responsible for target tracking and an initial guess of a path in the form of waypoints. The map data consists of polygons, and are downloaded using the OSMnx Python package based on a bounding box describing the area of interest, as described in Section 3.6. The data describing target ships comes from a simple target tracking module in this work, that estimates the trajectory of the target ships based on their current estimated speed and heading. The waypoints making up an initial guess for a trajectory for the ownship is provided by a high-level path planner implemented in the specialization project preceding this work, and is described in [20].

During the first run of the trajectory planner, the desired trajectory $\boldsymbol{x}_d$ is calculated based on a first-order Nomoto model simulated along the initial waypoints from the high-level path planner. After this, $\boldsymbol{x}_d$ is calculated based on a weighted average between the simulated reference trajectory $\boldsymbol{x}_{ref}$ and the time shifted and padded previous optimal trajectory $\boldsymbol{x}_{initial\_guess}$, as described in Section 3.5. When $\boldsymbol{x}_d$ is defined, it is time to define the relevant constraints for the NLP-solver used by the trajectory planner, before solving the NLP. The estimated trajectory from the target tracking module is used to define the constraints for dynamic obstacles, as described in Section 3.7 and Section 3.8.3. The constraints for the static obstacles are calculated as described in Section 3.6, where the constraints are defined for the desired position $\boldsymbol{p}_d$ at each dt_{stat}'th discretization step in the planning horizon.

In order to incentivize the trajectory planner to plan trajectories that comply with Rule 8 and Rule 16 of the COLREGs, two windows of reduced cost are calculated based on the estimated time to entering and exiting the critical DCPA, t_{crit}^{enter} and t_{crit}^{exit}, respectively. These windows are only calculated for target ships that are in a set of prioritized target ships, $\mathcal{N}_{pri}$, meaning they satisfy the entry criteria (3.54) in give-way encounters, or (3.55) in stand-on encounters as described in Section 3.8.1.

After the constraints for both static and dynamic obstacles are defined, and the windows of reduced cost are defined under the right conditions, the NLP can be solved by the IPOPT NLP-solver that is installed with the CasADi Python API. The solution to the NLP is an optimal trajectory that is used as input when calculating the reference trajectory $\boldsymbol{x}_d$ in the next iteration of the trajectory planner. The optimal trajectory is also interpolated by a interpolation-module that was already implemented in the ZeabuzSimulator provided. The interpolation module is described in [18], and is responsible for feeding reference signals from the discretized optimal trajectory to the DP-controller when using the milliAmpere1 vessel, or to the speed and heading controller when using the scaled Otter USV.

The complete trajectory planning pipeline described in this chapter is shown in Figure 3.8, going from the inputs fed to the trajectory planner, to the optimal trajectory outputted by it.

The trajectory planner also involves some static parameters that needs to be set. These parameters are summarized in Table 3.3, with short descriptions and references to the equations where the parameters are used.

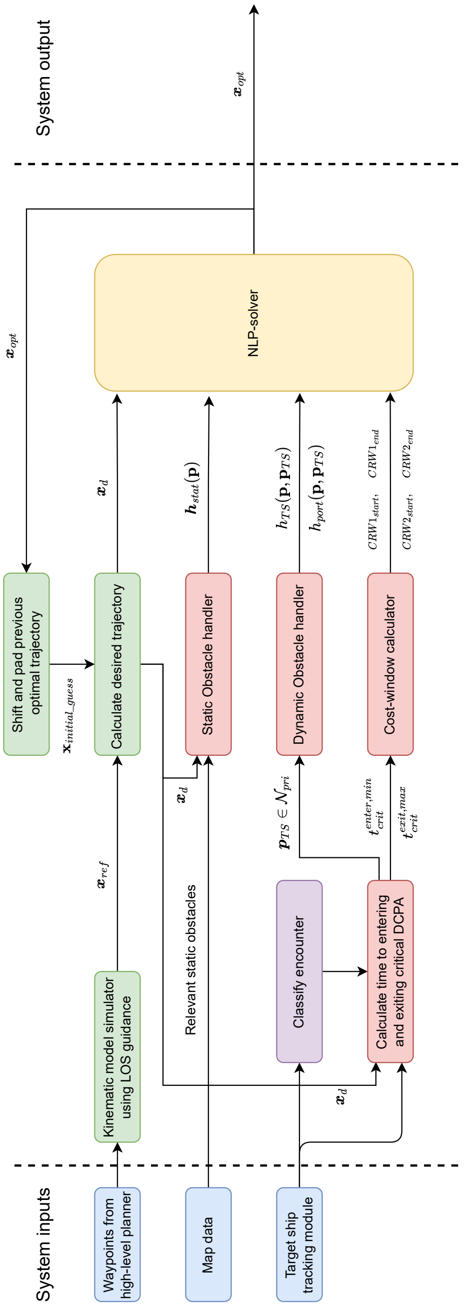

System inputs
Waypoints from high-level planner
Map data
Target ship tracking module
Kinematic model simulator using LOS guidance
$\boldsymbol{x}_{ref}$
Relevant static obstacles
Classify encounter
Calculate time to entering and exiting critical DCPA
$\boldsymbol{x}_d$
Shift and pad previous optimal trajectory
$\mathbf{x}_{initial\_guess}$
Calculate desired trajectory
$\boldsymbol{x}_d$
Static Obstacle handler
Dynamic Obstacle handler
Cost-window calculator
$\boldsymbol{p}_{TS} \in \mathcal{N}_{pri}$
$t_{crit}^{enter,min}$
$t_{crit}^{exit,max}$
$\boldsymbol{x}_d$
$\boldsymbol{h}_{stat}(\mathbf{p})$
$h_{TS}(\mathbf{p}, \mathbf{p}_{TS})$
$h_{port}(\mathbf{p}, \mathbf{p}_{TS})$
$CRW1_{start}$, $CRW1_{end}$
$CRW2_{start}$, $CRW2_{end}$
NLP-solver
$\boldsymbol{x}_{opt}$
$\boldsymbol{x}_{opt}$
System output

Figure 3.8: Overview of the NMPC-based COLAV-algorithm described in this work. Inspired by [18].

Parameter	Description	Equation
N_p	Number of discretized steps in planning horizon	(-)
h	Discretization step	(-)
K_k^p	Cost associated to deviations from a reference trajectory	(3.24)
K_k^a	Cost associated to the absolute acceleration	(3.24)
K_k^{tr}	Cost associated to the turning rate	(3.30)
dt_{nmpc}	Time interval between NMPC-runs	(-)
κ	Weighted average tuning parameter	(3.36)
dt_{stat}	Time interval between static obstacle constraints	(-)
d_{stat}	Safety distance to keep from static obstacles	(3.41)
α_{δ_s}	Encounter-specific tuning parameter	(3.43)
α_d	Encounter-specific tuning parameter	(3.46)
l	Shortest distance from the TS to the TS domain	(3.50)
$t_{after\_passing}$	Time parameter related to TS entry criteria	(3.54)
$t_{stand\_on\_reaction\_time}$	Time parameter related to TS entry criteria in stand-on encounters	(3.55)
$t_{maneuver}$	Time parameter related to the first window of redued cost	(3.62)
d_{crit}	Critical DCPA used in GW encounters	(-)
$d_{crit\_stand\_on}$	Critical DCPA used in SO encounters	(-)
K_{CRW1}^p	Reduction factor for K_k^p	(3.56)
K_{CRW2}^p	Reduction factor for K_k^p	(3.56)
K_{CRW1}^a	Reduction factor for K_k^a	(3.57)
K_{CRW2}^a	Reduction factor for K_k^a	(3.57)
K_{CRW1}^{tr}	Reduction factor for K_k^{tr}	(3.58)
K_{CRW2}^{tr}	Reduction factor for K_k^{tr}	(3.58)
d_{port}	Distance from TS to port-side constraint	(3.69)
$\alpha_{crossing}$	Small angle used in GW crossing encounters	(3.74)

Table 3.3: Overview of the 25 static parameters for the proposed trajectory planners, along with references to their respective equations.

Chapter 4

Simulation results

This chapter presents and discusses the results from simulations performed as part of this work in order to test the proposed trajectory planning methods. The chapter is divided into two sections, one section where low-speed simulations are performed using the milliAmpere1 ferry, and one section where high-speed simulations are performed using the scaled Otter USV described in Section 3.2. Each section is divided into three parts:

1. Batch simulations: In this subsection, 60 simulations are performed per scenario. Each scenario involves using the same target ship trajectory for all 60 simulations, while varying the starting position of the ownship. Batch simulations like these are inspired by the batch simulations performed in [18], and is a good way to visually show the overall behaviour of the trajectories produced by the trajectory planners for a given encounter-type. The batch simulations are performed for single vessel encounters, and includes head-on-, give-way crossing-, stand-on crossing- and overtaking encounters.

2. Simulations that show the effect of introducing windows of reduced cost and the port-side constraint described in Section 3.8 in head-on encounters. In this subsection, simulations are run with and without using windows of reduced cost and the port-side constraint described in Section 3.8. These simulations are performed in scenarios that are meant to show the effect of introducing these tools to the planner, while also challenging the tools. The results are compared, and the effect of the constraint is assessed visually through plots and based on the performance metrics described in Section 2.9.

3. Complex scenario simulations: In this subsection, simulation results from scenarios that are designed to challenge the trajectory planning methods are presented. These scenarios also includes static obstacles in the form of land, contrary to the simulations described above. This also includes scenarios that are challenging w.r.t. to the COLREGs, like the ones described in Section 2.1.1. Also in this section, the results are assessed based on plots and relevant performance metrics from Section 2.9.

The scenarios run in each of the subsections listed above are described in each subsection. For the low-speed simulations using the milliAmpere1 ferry model, the low-speed trajectory planner using the nonlinear model and objective function described in Section 3.4.1 is used. The high-speed simulations performed using the scaled Otter USV model are performed using the high-speed trajectory planer utilizing the nonlinear model and objective function described in Section 3.4.2.

4.1 Low-speed simulations: milliAmpere1

This section presents the results from the low speed simulations performed using the milliAmpere1 vessel model. Low speed is in this work considered to be $\approx$ 1-2 m/s, or $\approx$ 2-4 knots. First, the batch simulations are presented. Thereafter, the effect of the windows of reduced cost and the port-side constraint presented in Section 3.8 is assessed. Finally, some complex scenarios in narrow canal environments are presented.

4.1.1 Batch simulations

For the simulations performed using the milliAmpere1 vessel, the speed of the ownship is set to 1.5 m/s, and the speed of the target ship is set to 1 m/s. A total number of 60 simulations are performed in each scenario, where the ownship moves from west to east in each simulation. The ownship starts 200m north of the horizontal center line in the chosen NED-frame in the first simulation. Each simulation thereafter, the ownship starts $\approx$ 7m south of the previous starting position.

The target ship trajectory is set up based on the encounter type of interest, the trajectories are defined by two waypoints and a reference speed for the batch simulations, where the two waypoints are the initial- and goal position of the target ship. All target ship trajectories are set up such that they pass (0, 0) in the north-east plane halfway through the simulation.

Two batch simulations are performed for each of the encounter types, meaning two head-on encounters are simulated, two overtaking encounters etc. The batch simulated scenarios are summarized in Table 4.1. In order to run the simulations using the low-speed trajectory planner, the static parameters given in Table 3.3 need to be defined. The parameters for these simulations are given in Table 4.2 and Table 4.3.

Scenario name	**TS starting position**	**TS goal position**
HO_1	(50, 400)	(-50, -400)
HO_2	(-50, 400)	(50, -400)
OT_1	(50, -400)	(-50, 400)
OT_2	(-50, -400)	(50, 400)
GW_1	(-370, 150)	(370, -150)
GW_2	(-370, -150)	(370, 150)
SO_1	(370, 150)	(-370, -150)
SO_2	(370, -150)	(-370, 150)

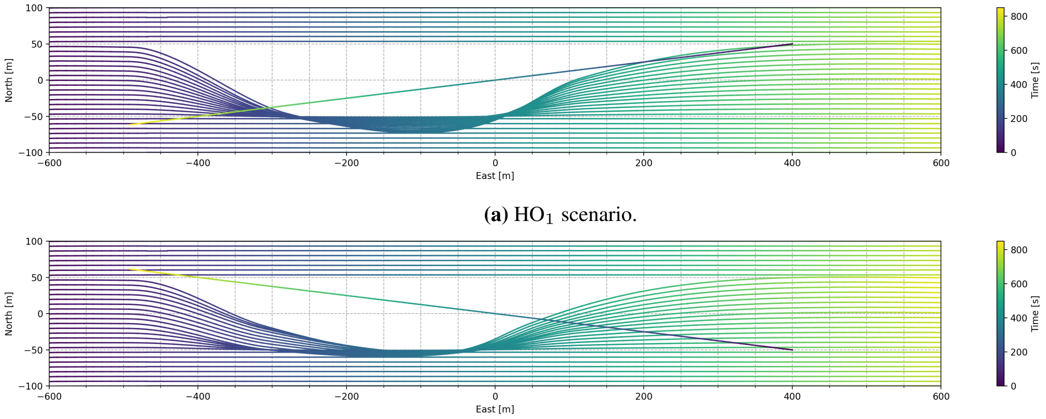
Table 4.1: Initial conditions for the batch simulations in this work performed using the milliAmpere1 vessel. HO stands for head-on, OT for overtaking, GW for give-way crossing and SO for stand-on crossing. The target ship keeps a constant speed of 1 m/s in all scenarios. All position coordinates are given in the NED frame as (N, E), where N and E are the north and east position, respectively.

Parameter	**Value**	**Unit**
N_p	400	[-]
h	2	[s]
dt_{nmpc}	4	[s]
K_k^p	2.5×10^{-6}	[m^{-2}]
K_k^a	30	[m^{-4}s^{4}]
κ	0.2	[-]
l	50	[m]
$t_{after\_passing}$	60	[s]
$t_{stand\_on\_reaction\_time}$	60	[s]
$t_{maneuver}$	120	[s]
d_{crit}	50	[m]
$d_{crit\_stand\_on}$	25	[m]
K_{CRW1}^p	10^{-5}	[-]
K_{CRW2}^p	10^{-5}	[-]
K_{CRW1}^a	0.008	[-]
K_{CRW2}^a	0.1	[-]
d_{port}	10	[m]
$\alpha_{crossing}$	$\pi/4$	[rad]

Table 4.2: Numerical values of static parameters used in the low-speed batch simulations.

Parameter	**HO**	**GW**	**OT$_p$**	**OT$_s$**	**SO**	**SF**	**Unit**
α_{δ_s}	$\pi/12$	$\pi/6$	$-3\pi/4$	$3\pi/4$	$\pi/12$	0	rad
α_d	$\pi/10$	$\pi/4$	$\pi/8$	$\pi/8$	$\pi/16$	0	rad

Table 4.3: Encounter-specific parameters used in the low-speed batch simulations. OT$_p$ and OT$_s$ stands for overtaking port side and overtaking starboard side, respectively.



(a) HO_1 scenario.

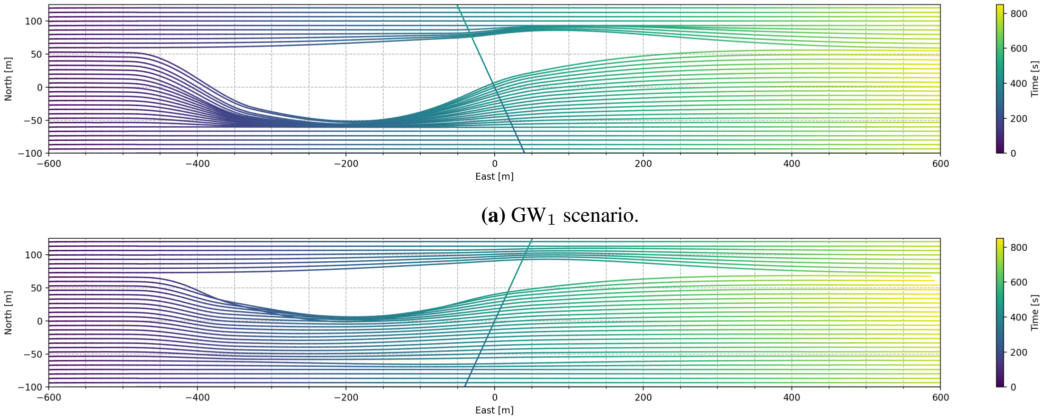
(b) HO_2 scenario.

Figure 4.1: A subset of the head-on batch simulations. The color of the trajectories indicates the time in seconds, meaning the color of the trajectory goes from purple to yellow as the time passes for both the ownship trajectories and the target ship trajectory.

Head-on encounters

A subset of the head-on batch simulations are shown in Figure 4.1, where the first and last of the 60 simulations are not shown, as they only involve the ownship going in a straight line from east to west. As seen in the plots, the ownship keeps its course throughout the simulation in all simulations above 50m north and below 50m south. This is because d_{crit} and l are set to 50m for these simulations, meaning the trajectory planner does not deem it necessary to perform any action above 50m north and below 50m south. As seen when the vessel starts between 50m north and 50m south, the trajectory planner plans maneuvers such that the ownship gives way by turning to its starboard. This is in compliance with Rule 14 of the COLREGs. The action is also performed relatively early in the horizon, especially when considering the ownship and target ship have speeds of 1.5 m/s and 1 m/s, respectively. This can be argued to be in compliance with Rule 8 and Rule 16 of the COLREGs. What is deemed ample time is however heavily context-dependent, and what exactly is ample time in this particular situation is hard to determine without much experience navigating at sea. In both situations, the ownship will however show its red sidelight mounted on its port side. This will give the target ship an early indication of the ownship's intention of turning towards starboard and give way for the target ship, which is important to avoid confusion between the vessels.

It can also be seen from Figure 4.1a where the HO_1 scenario is simulated that the starboard maneuver is slightly more aggressive. This is likely due to the port-side constraint discussed in Section 3.8, as this is a significantly stricter constraint in the HO_1 scenario than for the HO_2 scenario shown in Figure 4.1b. For the HO_2 scenario, the ownship is already on the right side of this constraint when starting the maneuver, so the constraint is never really active.



(a) GW$_1$ scenario.

(b) GW$_2$ scenario.

Figure 4.2: A subset of the 60 give-way crossing batch simulations.

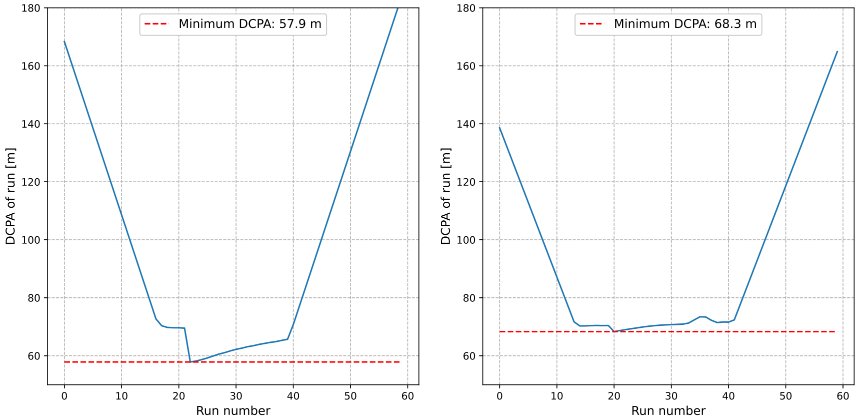
Give-way crossing encounters

A subset of the give-way crossing scenarios GW$_1$ and GW$_2$ are shown in Figure 4.2. From these plots it is slightly harder to assess the COLREGs compliance strictly based on the plots, since the ownship performs a smaller course change maneuver. The DCPA between the ownship and target ship in each of the 60 simulations are shown in Figure 4.3, and from this figure it is at least clear that the ownship keeps a distance of $>$ 55m from the target ship in all simulations. There is however a big difference between keeping a distance of 55m when crossing abaft the target ship and crossing in front of it in a give-way crossing situation. If the ownship crosses in front of the target ship at a distance of 55m, this can be argued to breach the give-way obligations of the ownship. The figures also show how the MPC-based trajectory planner plans paths that hug the contour 50m outside of the target ship. This can be seen based on the DCPA decreasing linearly for the first simulations, before it bounces above the 50m distance in the middle simulations. As the ownship is simulated further south again, the DCPA increases linearly for the last simulations.

A more clear image of the behaviour of the ownship is shown in Figure 4.4, where the speed of the ownship

$$U = \sqrt{u^2 + v^2} \tag{4.1}$$

is shown for all simulations in the GW$_2$ scenario. The color of the plot indicates what simulation is run, and the lighter the color gets, the further south the ownship starting position is. As seen from this figure, the ownship increases its speed in the plots with darker colors. This means the ownship increases its speed in order to cross in front of the target ship in the first simulations, i.e. the simulations started from the north-most positions. It is however seen that at some point, possibly between simulations 20 and 30, the ownship starts slowing down in order to let the target ship cross in front of it. These are also likely the simulations where the ownship starts turning towards starboard in Figure 4.2b. Changing the course and slowing down are both give-way actions in compliance with



(a) GW$_1$ scenario.

(b) GW$_2$ scenario.

Figure 4.3: DCPA between the ownship and target ship during give-way crossing batch simulations.

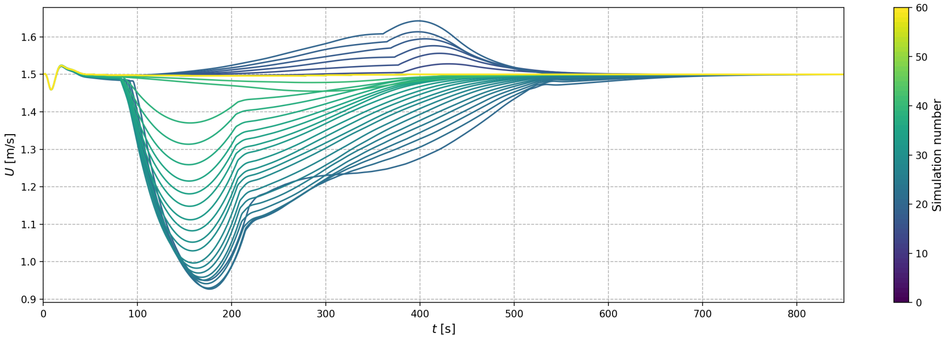
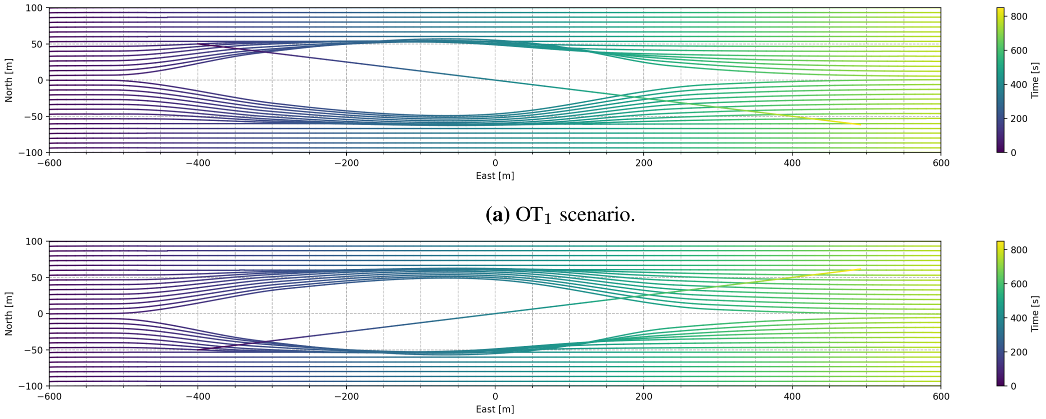


Figure 4.4: ownship speed for all 60 simulations in the GW$_2$ scenario. The first simulation starts at 200m north, and the last simulation starts -200m north.

Rule 15 and Rule 16 of the COLREGs. A case can however be made that the simulations where the ownship crosses in front of the target ship is a violation of the same rules. The threshold of what distance it is allowed for the ownship to cross in front of the target ship in crossing situations is however hard to decide, and heavily context-dependent. In these batch simulations the ownship never crosses in front of it at distances of less than 55m, which might be acceptable when the target ship has a speed of 1 m/s. The distance d_{crit} used to activate the windows of reduced cost and therefore also the port-side constraint is also a circular distance around the target ship. If this distance was set such that it forms an ellipse around the target ship, with the major axis aligned with the heading of the target ship, the port-side constraint could be activate in simulations where the ownship starts further north than what is shown in Figure 4.2. This could make the ownship give way in earlier simulations, but the threshold in which this should happen is as mentioned dependent on context.



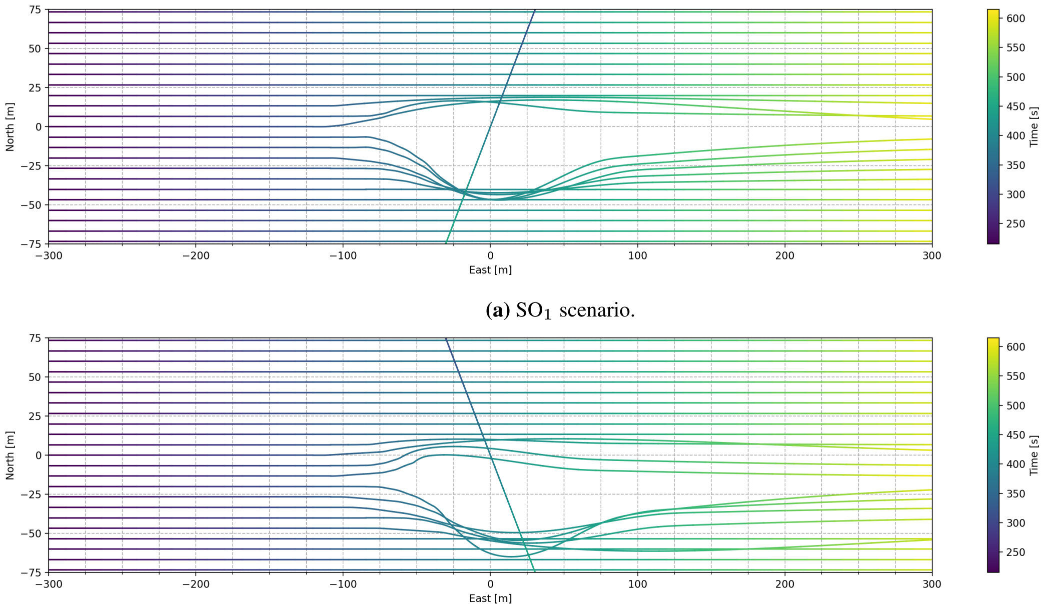
(a) OT$_1$ scenario.

(b) OT$_2$ scenario.

Figure 4.5: Batch simulations for overtaking encounters.

Overtaking encounters

A subset of the overtaking scenarios OT$_1$ and OT$_2$ are shown in Figure 4.5. The amount of simulations where the trajectory planner plans to overtake the target ship on its port side and its starboard side is approximately symmetric in both figures. In all simulations, the ownship gives way for the target ship when overtaking it, as it should w.r.t. Rule 13 of the COLREGs. It should however be noted that the ownship should pass the target ship on its port side in most of the simulations in Figure 4.5a, and on the starboard side of the target ship in Figure 4.5b. This could possibly be achieved by tuning the α_{δ_s}-parameter used to enforce the correct side for crossing. It can however be noted that all crossings are in compliance with Rule 13, as the rule allows for overtaking on both sides. The ownship should however never become a crossing vessel according to Rule 13, which can be argued to be the case for some of the simulations in both scenarios. Choosing the correct side to overtake the target ship on is hard, as discussed in Section 2.1.1.



(a) SO_1 scenario.

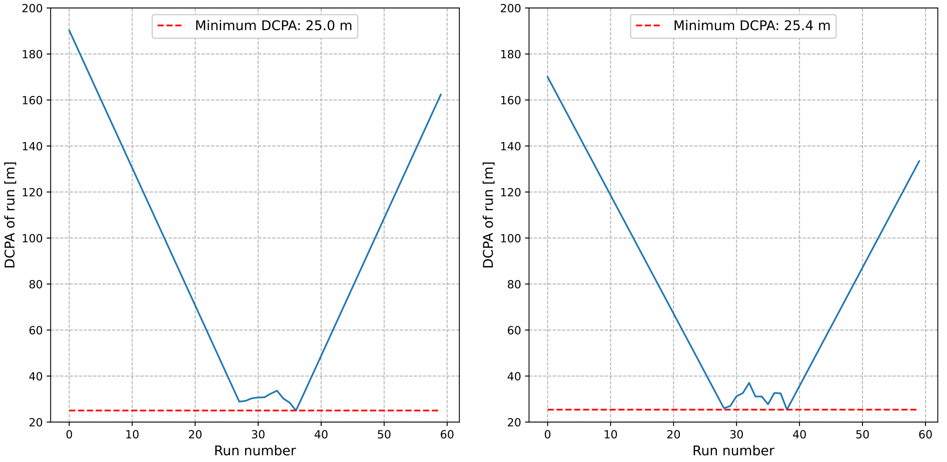
(b) SO_2 scenario.

Figure 4.6: Stand-on crossing batch simulations.

Stand-on crossing encounters

A subset of the simulations performed in the stand-on crossing scenarios SO_1 and SO_2 are shown in Figure 4.6. In the simulations performed, the target ship has a constant heading and speed, meaning it does not make any action to comply with the COLREGs itself. Therefore, these simulations test not only the compliance with Rule 17 of the COLREGs regarding stand-on vessels, but also to some extent Rule 2 of the COLREGs discussed in Section 2.1.1. This is because the target ship will not comply with the COLREGs in these simulations, meaning the ownship will not be able to keep its stand-on obligations in all simulations, as this would lead to collision at some point. The ownship will therefore need to make a departure from its stand-on obligations given by Rule 17 in order to avoid collision. Contrary to the previous figures in this section, Figure 4.6 does not include the whole time horizon in order to get a clearer picture of what is going on around the time where it is clear the target ship will not keep its give-way obligation.

As seen in both figures, the ownship takes action to avoid collision when it is clear the target ship will not. This behaviour is decided by the $t_{stand\_on\_reaction\_time}$ and $d_{crit\_stand\_on}$ parameters in Table 4.2. Here, $t_{stand\_on\_reaction\_time} = 60\,\mathrm{s}$ means the ownship takes action when there is less than 60s until it enters the stand-on critical DCPA of the target ship, which is set to $d_{crit\_stand\_on} = 25\,\mathrm{m}$ in these simulations. As seen in Figure 4.7, the distance kept from the target ship by the ownship is 25m or more for all encounters. This might be uncomfortably close for the passengers onboard depending on the situation, but these simulations are in practice simulations where emergency action is



(a) SO$_1$ scenario.

(b) SO$_2$ scenario.

Figure 4.7: DCPA between the ownship and target ship for all 60 stand-on crossing batch simulations.

necessary, meaning avoiding collision is the only priority.

4.1.2 Effect of windows of reduced cost and port-side constraint

This section aims to show the effects of introducing tools such as windows of reduced cost and the port-side constraint described in Section 3.8. The quality and necessity of these tools is shown through simulations where the ownship has a give-way obligation in a head-on situation. The head-on situation is the HO_1 scenario from Table 4.1. First the effect of introducing windows of reduced cost is assessed, and this is based on plots showing the trajectories, as well as the performance metrics from Section 2.9. Finally, the effect of also introducing the port-side constraint is assessed based on the same trajectories and performance metrics.

In order to calculate the COLREGs-specific performance metrics, some parameters has to be defined. For the metric that assess if a maneuver is made in ample time, the threshold ϵ_χ must be defined in order to calculate (2.66). This parameter shall be set to what is the minimum change in course that can be accepted as the start of a maneuver. In [34], it is argued that a course change of less than 10° is not likely to be accepted as a maneuver that is apparent for another vessel by radar or visually.

For the metric given by (2.70) that assesses if a maneuver is readily apparent, the minimum course change that can be deemed as readily apparent $\Delta_{\chi_{app}}$ needs to be defined. In [34], it is stated that a maneuver should be at least 30 degrees to be deemed readily apparent for another vessel.

The r_{det} parameter used by both (2.66) and (2.70) is in the following simulations set to the distance between the vessels at the start of the simulation, as it is assumed the ownship gets accurate information about the target ship from the very beginning of the simulation.

For the safety score (2.72) some parameters needs to be defined. These are the R_{min}, R_{nm}, R_{col}, γ_{nm} and γ_{col} parameters. For these simulations R_{min} is set to 50m, R_{nm} to 30m and R_{col} to 15m. The tuning parameters γ_{nm} and γ_{col} are set to 0.25 and 0.75, respectively. The rationale behind this is to penalize passing the target ship at R_{col} more than passing it at R_{nm} as $R_{nm} > R_{col}$.

The parameters are summarized in Table 4.4, and all other parameters used by the trajectory planner are the same as in Table 4.2 and Table 4.3.

Parameter	ϵ_{chi}	$\Delta_{\chi_{app}}$	R_{min}	R_{nm}	R_{col}	γ_{nm}	γ_{col}
Value	10	30	50	30	15	0.25	0.75
Unit	[deg]	[deg]	[m]	[m]	[m]	[-]	[-]

Table 4.4: Parameters for COLREGs-specific performance metrics for the low-speed simulations using the milliAmpere1 ferry model.

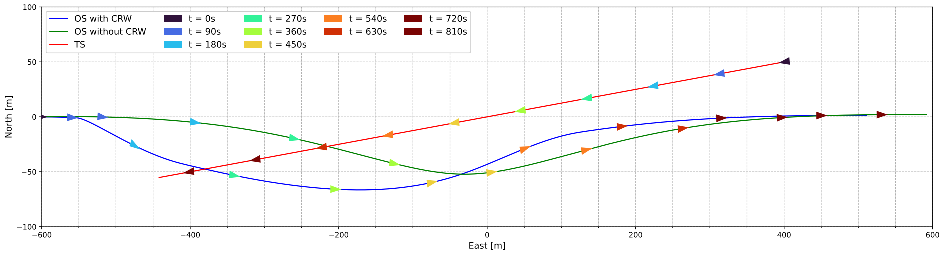


Figure 4.8: Trajectories of the ownship with and without using windows of reduced cost, as well as the target ship trajectory. The arrows shows the direction of the vessels, and the color of the arrow indicates the time passed.

Introducing windows of reduced cost

The trajectories of the ownship with and without using cost-reduction windows (CRW) as well as the target ship trajectory is shown in Figure 4.8. From these plots, the effect of introducing windows of reduced cost is quite clear, as the maneuver starts significantly earlier when using these windows.

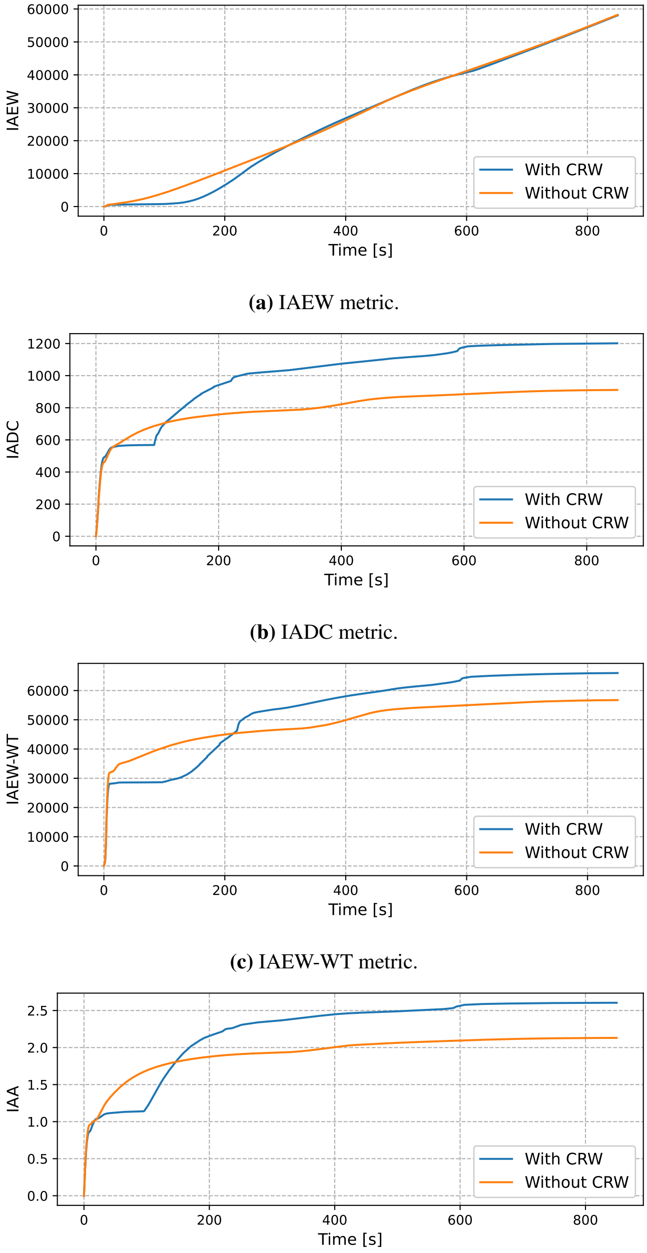
The COLREGs-specific metrics from the two simulations in Figure 4.8 are shown in Table 4.5. These metrics indicate lower scores for $\mathcal{P}_{delay}$ and $\mathcal{P}'_{\Delta_{\chi app}}$, meaning the maneuver is initiated earlier and that the maneuver is more readily apparent to observe for the target ship when using windows of reduced cost. This is as expected based on Figure 4.8. The two simulations both score very low one the $\mathcal{P}_{safety}$ score. The scores indicate a next to perfect score when it comes to the distance kept from the target ship. This is however a result of using the TS domain described in Section 3.7, as the TS domain allows for setting the minimum distance to keep from the target ship directly.

Metric	Using CRWs	Without using CRWs
$\mathcal{P}_{delay}$	**0.16**	0.70
$\mathcal{P}'_{\Delta_{\chi app}}$	**0.26**	0.88
$\mathcal{P}_{safety}$	0.012	**0.011**

Table 4.5: COLREGs-specific performance metric scores with and without using windows of reduced cost. The metrics are calculated using the parameters given in Table 4.4. The best results are bold, although there is little difference between 0.012 and 0.011 in practice.

The COLREGs-specific metrics are however not all that matters in this work. The control error, energy consumption, actuator wear-and-tear and absolute acceleration are also important to take into account when assessing the quality of the methods. The IAEW, IADC, IAEW-WT and IAA metrics from the simulations are shown in Figure 4.9. From Figure 4.9a it is seen that the control error and power consumption is quite similar in both cases. It is however seen in Figure 4.9b and Figure 4.9c that the actuator wear-and-tear is higher when using windows of reduced cost. This is as expected, as the windows of reduced cost incentivizes the ownship to accelerate more in order to perform more readily apparent maneuvers. This is also shown by the IAA score in Figure 4.9d.

It is based on the above seen that the windows of reduced cost effectively increases the compliance with Rule 8 and Rule 16 of the COLREGs by enforcing earlier and more apparent maneuvers compared to when not using the windows. It does however come at the cost of more energy consumption and actuator wear-and-tear. As seen from the safety score in Table 4.5, the safety is not necessarily increased through introducing the windows of reduced cost. The safety score indicates a safe transit in both cases, at least with respect to the distance kept to the ownship. It therefore ends up in a trade-off between COLREGs-compliance and factors such as actuator wear-and-tear. It can however be argued that COLREGs-compliance leads to increased safety as well, as this leads to less confusion between the ownship and target ship at sea, which likely leads to less possibility of collisions or other unwanted close-quarter situations. The safety of the passengers onboard shall always be of highest priority, followed by factors such as expenses associated with energy consumption and increased actuator wear-and-tear.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.9: IAEW, IADC, IAEW-WT and IAA scores with and without using windows of reduced cost.

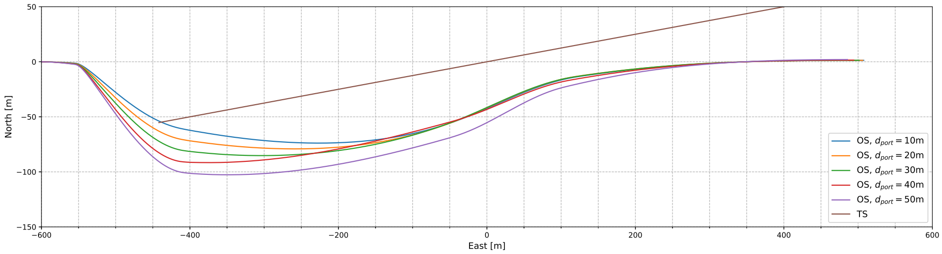


Figure 4.10: Trajectories of the ownship with five choices of d_{port} in the range from 10m to 50m and the target ship trajectory.

Adding the port-side constraint

In order to assess the effect of the port-side constraint presented in Section 3.8.3, the same head-on encounter as above is simulated, varying the distance d_{port} from the target ship to the constraint. The d_{port} parameter is varied from 10m to 50m, in steps of 10m between simulations. The resulting trajectories are shown in Figure 4.10. As seen from the figure, increasing d_{port} forces the ownship to make more aggressive maneuvers. As discussed in Section 3.8.3, the d_{port} parameter should not be larger than the l-parameter used to define the TS domain or the d_{crit} parameter used to define the windows of reduced cost. This statement is strengthened by Figure 4.10, as it is hard to see the need for more aggressive maneuvers than those made in the figure. Choosing d_{port} such that

$$d_{port} \leq 0.6 \times l \tag{4.2}$$

seems to be more than enough in order to force a readily apparent maneuver in compliance with Rule 8 and Rule 16 of the COLREGs.

The COLREGs-specific performance metric scores when using the parameters from Table 4.4 is shown in Table 4.6. As seen from the table, the $\mathcal{P}_{delay}$ and $\mathcal{P}'_{\Delta_{\chi_{app}}}$ scores are the same for all choices of d_{port}. The safety score is also more or less the same, and close to zero in all cases. This shows that increasing d_{port} from $0.2 \times l$ to l has no effect on these metrics. As seen from the $\mathcal{P}'_{\Delta_{\chi_{app}}}$ score, the maneuver is deemed readily apparent in all cases when the threshold of a readily apparent maneuver is set to 30°. When compared to Table 4.5, the table also shows that introducing the port-side constraint leads to a more readily apparent maneuver based on the $\mathcal{P}'_{\Delta_{\chi_{app}}}$-score, compared to when only using windows of reduced cost.

The choice of $\Delta_{\chi_{app}} = 30°$ is as mentioned based on [34], where 30° is said to be the minimum requirement of a readily apparent maneuver. It is however also stated in [34] that it is preferred that the maneuver is between 60° and 90°. Changing $\Delta_{\chi_{app}}$ to 60° gives the results shown in Table 4.7. This shows that increasing d_{port} leads to a positive effect on the metric, indicating the maneuver becomes more and more apparent as d_{port} is increased. The necessity of the constraint should however be assessed based on the environment the vessel is to maneuver in among other factors.

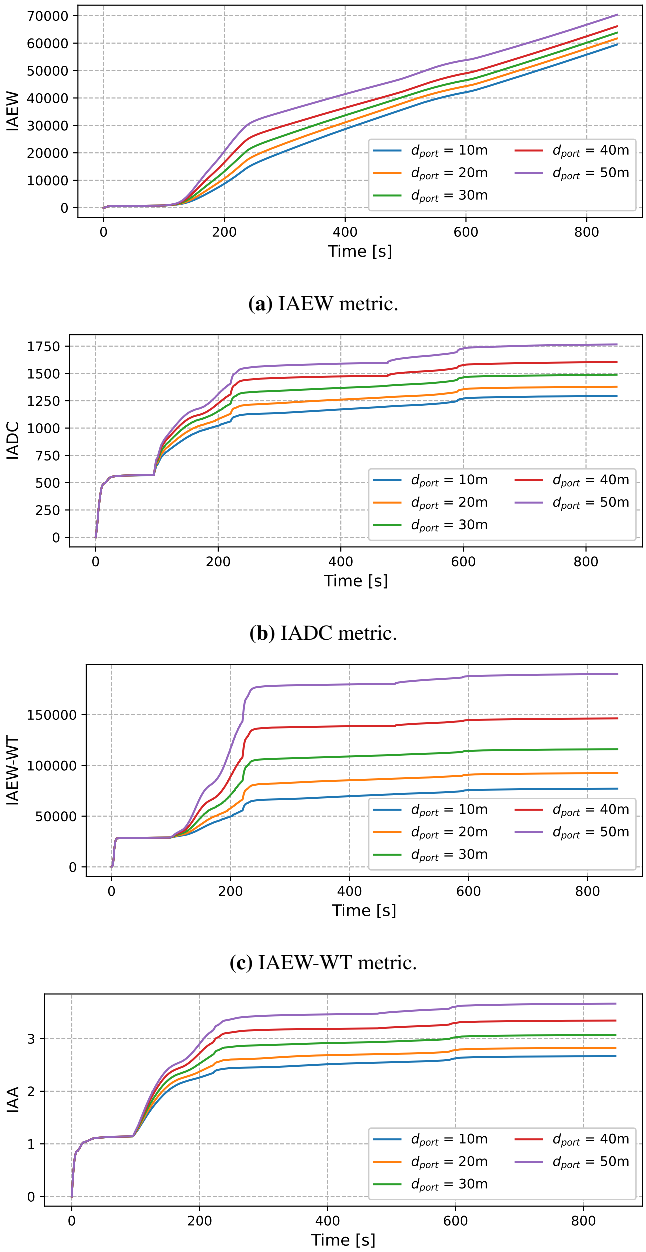
The necessity of the constraint can be further questioned when looking at Figure 4.11. In this figure it is seen that the IAEW, IADC, IAEW-WT and IAA metrics increases proportionally as the d_{port}-parameter is increased. This means the control error, power consumption, actuator wear-and-tear and the acceleration during the transit increases. This indicates the constraint might not be necessary or even desirable in the context of autonomous passenger ferries, where passenger safety is of highest importance. Adding the port-side constraint does however effectively make the maneuver more apparent as seen in Figure 4.10 and Table 4.7, demonstrating its efficiency as a tool that can be added to the COLREGs-compliant trajectory-planning toolbox when needed.

Metric	$d_{port} = 10$	$d_{port} = 20$	$d_{port} = 30$	$d_{port} = 40$	$d_{port} = 50$
$\mathcal{P}_{delay}$	0.16	0.16	0.16	0.16	0.16
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0	0	0	0	0
$\mathcal{P}_{safety}$	0.0057	0.0042	0.0038	**0.016**	0

Table 4.6: COLREGs-specific performance metric scores with a varying value of d_{port}. The metrics are calculated using the parameters given in Table 4.4.

Metric	$d_{port} = 10$	$d_{port} = 20$	$d_{port} = 30$	$d_{port} = 40$	$d_{port} = 50$
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0.62	0.50	0.37	0.21	**0.089**

Table 4.7: The $\mathcal{P}'_{\Delta_{\chi_{app}}}$-metric scores with $\Delta_{\chi_{app}} = 60°$ instead of $\Delta_{\chi_{app}} = 30°$.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.11: IAEW, IADC, IAEW-WT and IAA scores when varying the d_{port}-parameter when using the port-side constraint.

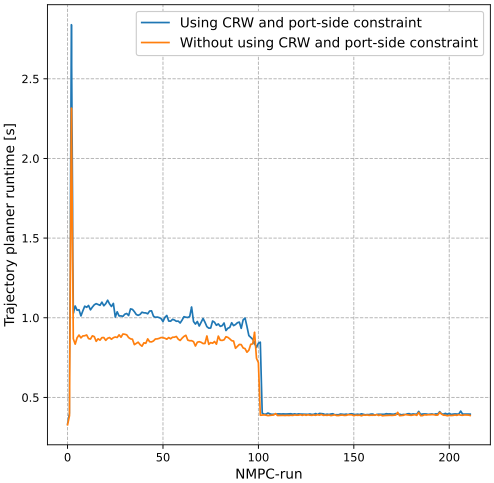


Figure 4.12: The runtime of the NMPC-planner with and without using cost-reduction windows (CRW) and the port-side constraint. The runtimes covers the time it takes for the NMPC-planner to set up the constraints of the NLP and solve it.

Another aspect to consider when adding tools such as windows of reduced cost and the port-side constraint to the trajectory planning pipeline is that they are likely to increase the total runtime of the algorithm. The runtimes of the trajectory planner each NMPC-run is shown in Figure 4.12, both with and without using windows of reduced cost and the port-side constraint in the HO_1 scenario. As seen from the figure, adding windows of reduced cost and the port-side constraint increases the runtime. The increase is however relatively small, meaning the increased runtime is a small price to pay compared to potential benefits of adding the tools. This is especially the case if one considers that the implementation used in this work is written in Python. Implementing the trajectory planner in e.g. C++ would likely decrease the runtime significantly. It should also be mentioned that this runtime is obtained from simulations, and not from physical experiments. The runtime shown in the figure also might not even be a problem, as what is an acceptable runtime is dependent on the operational domain of the trajectory planner.

4.1.3 Complex scenarios

This section presents the results from simulating more complex scenarios using the milliAmpere1 model and the low-speed trajectory planner. In this work, a complex scenario is a scenario that involves static obstacles and potentially multiple target ships. Two scenarios are simulated, both situated at Brattøra in Trondheim, and are set up in a manner that aims to challenge the methods in different ways. Both scenarios used in the complex simulations using the milliAmpere1 model uses the same encounter-specific parameters as shown in Table 4.3, and the rest of the static parameters used by the trajectory planner are given in Table 4.8. The parameters used to calculate the COLREGs-specific parameters in the complex simulations are given in Table 4.9. The port-side constraint is not used in either of the complex scenario simulations, as it is seen to cause problems when there are multiple target ships.

Parameter	Value	Unit
N_p	300	[-]
h	2	[s]
dt_{nmpc}	4	[s]
dt_{stat}	6	[s]
d_{stat}	8	[m]
K_k^p	2.5×10^{-6}	[m^{-2}]
K_k^a	30	[m^{-4}s^{4}]
κ	0.2	[-]
l	15	[m]
$t_{after\_passing}$	10	[s]
$t_{stand\_on\_reaction\_time}$	10	[s]
$t_{maneuver}$	20	[s]
d_{crit}	15	[m]
$d_{crit\_stand\_on}$	10	[m]
K_{CRW1}^p	10^{-5}	[-]
K_{CRW2}^p	10^{-5}	[-]
K_{CRW1}^a	0.008	[-]
K_{CRW2}^a	0.1	[-]

Table 4.8: Numerical values of static parameters used in the complex low-speed simulations.

Parameter	ϵ_{chi}	$\Delta_{\chi_{app}}$	R_{min}	R_{nm}	R_{col}	γ_{nm}	γ_{col}
Value	10	60	15	10	5	0.25	0.75
Unit	[deg]	[deg]	[m]	[m]	[m]	[-]	[-]

Table 4.9: Summary of the parameters needed to calculate COLREGs-specific performance metrics for the complex low-speed simulations using the milliAmpere1 ferry model.

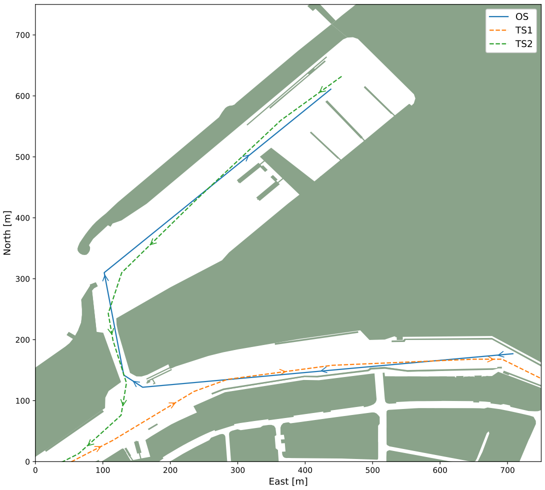


Figure 4.13: Overview of the BRATTØRA1 scenario. The arrows indicate the directions of the respective vessels.

BRATTØRA1 scenario

The BRATTØRA1 scenario is located at Brattøra in Trondheim, and aims to challenge the trajectory planner by introducing two consequent head-on encounters, where both target ships are detected late by the situational awareness system. The ZeabuzSimulator used in this work is as mentioned earlier improved with the possibility of performing complex simulations, and part of this is that it is possible to add functionality such that target ships are not detected if there is a polygon in the form of land between the ownship and target ship. In a real world setting it is of course possible to get the position of a target ship through AIS data, meaning the target ship does not need to be in line-of-sight of the ownship. It is however assumed in the rest of this work that the target ship is not considered by the trajectory planner until it is in line-of-sight of the ownship. It is also assumed that once a target ship has been in line-of-sight of the ownship, it will not be disregarded by the planner again.

An overview of the scenario is shown in Figure 4.13, which shows the trajectories of the ownship and two target ships based on the waypoints they are to follow. The waypoints the ownship is to follow are calculated using the high-level trajectory planner described in [20], and the target ship trajectories are set up manually. It should be noted that the high-level trajectory planner used to calculate the waypoints does not take the distance to be kept from land into account, nor does it take traffic lanes on water into account when planning. The two target ships are named TS1 and TS2, and both have a constant speed of 1.5 m/s throughout the simulation, while the ownship has a speed of 2 m/s. For this simulation, the static obstacle constraints are generated such that the trajectory planner

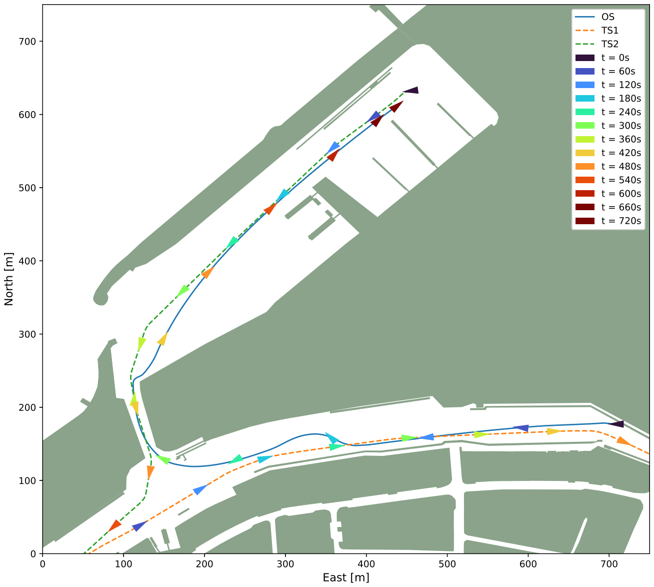


Figure 4.14: Resulting trajectories from simulations performed on the BRATTØRA1 scenario. The colors indicate the timestamps throughout the simulation, and the arrows indicate the heading angle of the respective vessels involved.

keeps a minimum distance of 8m to land-polygons.

The trajectories resulting from simulating the BRATTØRA1 scenario are shown in Figure 4.14. As shown in the figure, the first head-on encounter involves TS1, while the second involves TS2. TS1 is detected by the ownship at $t = 147$s, which is between the $t = 120$s and the $t = 180$s arrows in the figure. It is seen that the ownship gives way for the target ship in a way that complies with Rule 14 regarding head-on encounters. TS2 is detected by the ownship at $t = 350$s, which is approximately at the end of the narrow passage the ownship passes through about halfway through the simulation. As seen in the figure, this leads to a more aggressive maneuver by the ownship, as it is constricted by static obstacles on both sides at the time TS2 is detected. It is however seen that the trajectory planner is able to avoid collision with TS2, in a way that complies with Rule 14.

The distance between the ownship and the two target ships is shown in Figure 4.15. The minimum distance used to set up the TS-domain is set to $l = 15$m, but it can be seen in the figure that the DCPA between the ownship and TS2 is 12.4m. This effectively means the ownship has entered the TS domain, which further means the trajectory planner could not have found a feasible solution at this particular point during the simulation. The ownship is however able to avoid collision by giving way to TS2. Regardless, this shows the downside with using optimization-based methods for trajectory planning, as there is always the possibility of not obtaining a feasible solution to the OCP. This did not cause big problems in this particular simulation, but it could possibly cause problems in a real-world experiment. It must also be noted that in a real setting, TS2 would likely slacken its

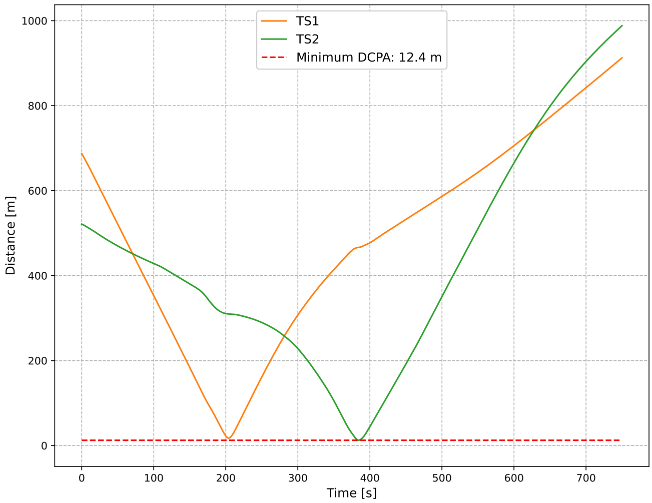


Figure 4.15: Distance between the ownship and target ships in the BRATTØRA1 scenario.

speed in this particular scenario in order to let the ownship finish the narrow passage, in compliance with good seamanship.

Using the timestamps for when the target ships were detected by the ownship, it is possible to calculate r_{det}, which further makes it possible to calculate the COLREGs-specific performance metrics in this scenario. The metrics has to be calculated w.r.t. each of the target ships, and the scores are shown in Table 4.10. As seen in the table, the scores for delayed action are at least under 0.5. The scores are however higher than what has been seen in earlier simulations, but the scores can also be expected to be higher in a narrow canal than at open sea. The first window of reduced cost is also activated a bit after the target ships is added to $\mathcal{N}_{pri}$, meaning starting the window right away could likely lead to an earlier maneuver. This could however also cause problems if the planner is not given any time to plan before the ownship is suddenly inside of CRW1.

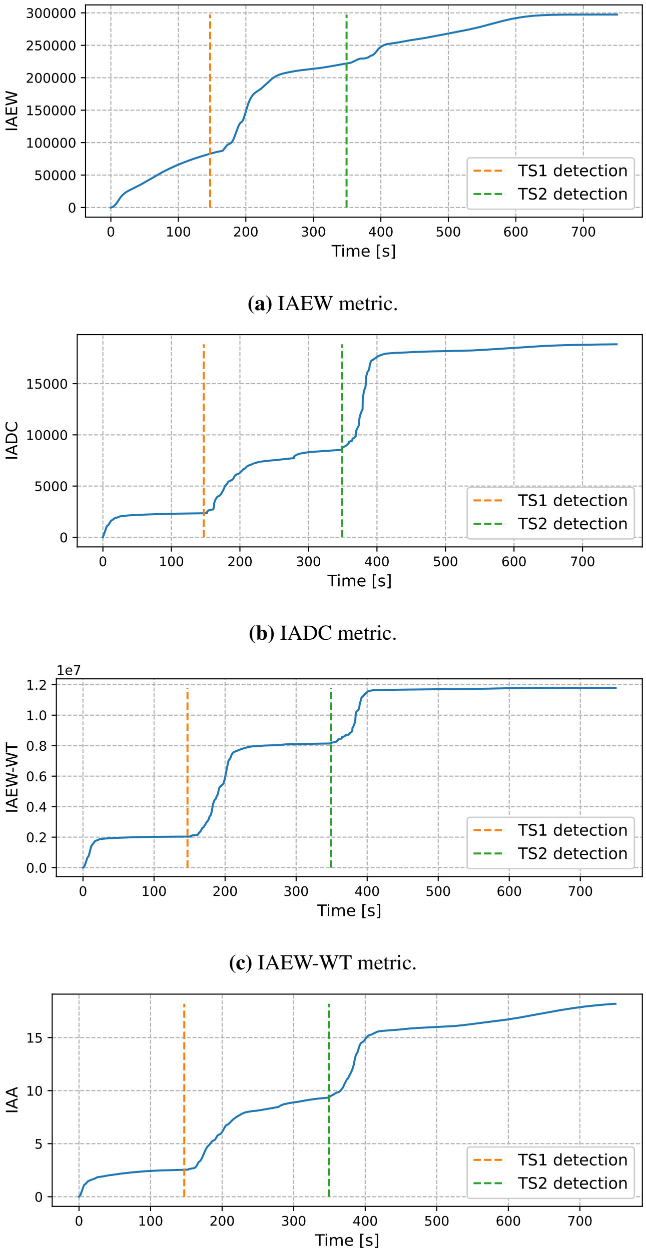
Metric	TS1	TS2
$\mathcal{P}_{delay}$	0.43	0.40
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0.17	0
$\mathcal{P}_{safety}$	0	0.12

Table 4.10: COLREGs-specific performance metric scores resulting from simulating the BRATTØRA1 scenario.

The score penalizing non-apparent maneuvers is low for both encounters. The score is actually 0 for the encounter with TS2, but this is as expected when the ownship has to make such an aggressive maneuver to avoid collision. For the TS2 encounter, it can also be seen that the $\mathcal{P}_{safety}$ score is higher than the scores seen in earlier simulations. This is because for TS2, the ownship actually comes closer than the minimum distance $R_{min} = 15$m, which has not been the case earlier.

The IAEW, IADC, IAEW-WT and IAA scores resulting from simulating the BRATTØRA1 scenario are shown in Figure 4.16. In these figures, the times where TS1 and TS2 are detected are indicated by vertical lines. It is clear from the figures that each detection leads to a subsequent increased power consumption, actuator usage and acceleration. It can also be seen that the metrics increase after a little delay from when the target ships are detected, which likely coincide with when the first window of reduced cost starts.

Simulating scenarios that consists of more than one target ship as well as static obstacles is expected to lead to an increased runtime of the trajectory planner. The runtimes from simulating the BRATTØRA1 scenario are shown in Figure 4.17. As seen in the figure, the runtime is higher in the presence of two target ships and static obstacles, as one would expect. The peaks are likely from when TS1 and TS2 are detected, which leads to a significantly increased runtime. Especially for the encounter with TS2, it is important to avoid large peaks in the runtime, as there is less time to maneuver.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.16: IAEW, IADC, IAEW-WT and IAA scores resulting from the BRATTØRA1 scenario. The times where the target ships are detected are indicated.

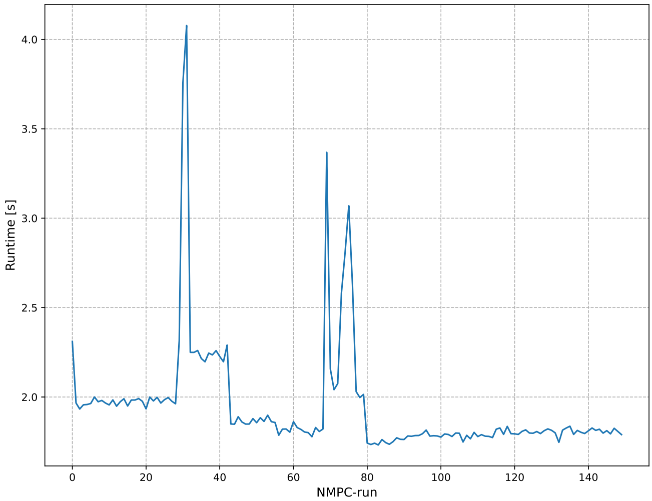


Figure 4.17: Trajectory planner runtime when simulating the BRATTØRA1 scenario.

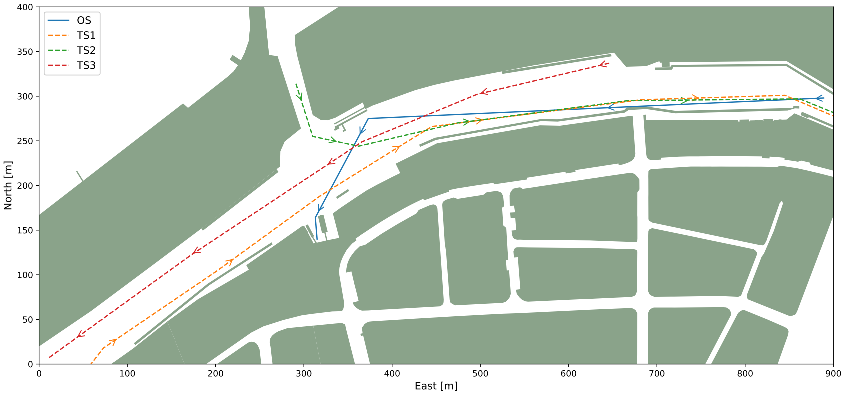


Figure 4.18: Overview of the BRATTØRA2 scenario. The arrows indicate the directions of the respective vessels.

BRATTØRA2 scenario

The BRATTØRA2 scenario is inspired by the first overtaking situation described in Section 2.1.1, where an overtaking encounter creates challenges when planning a trajectory. The scenario involves three target ships, where the target ships are set up such that

- TS1 creates a head-on encounter, and also poses a late detection as it is not in line-of-sight of the ownship until some time has passed.
- TS2 also creates a head-on encounter, but because the intent of the target ships is estimated like in this work, the encounter is not classified as a head-on encounter until some time has passed.
- TS3 creates an overtaking encounter, where the ownship has the choice of overtaking TS3 on either the port- or starboard side. This is the vessel that potentially causes the most problems, as it forces the ownship to consider whether or not to overtake the target ship, while also having to consider two head-on encounters.

The scenario is shown in Figure 4.18. As seen in the figure, the ownship is supposed to dock in the opposite end of the canal relative to its traffic lane. Docking is not part of this work, meaning the trajectory planner will not actually plan a trajectory that suits a docking scenario, at least not deliberately. It does however show how the situation is handled w.r.t. to the different encounters.

As seen in Figure 4.18, the canal has a natural curvature. This means it shares similarities with the first situation described in Section 2.1.1 as it means it might not be the most reasonable to overtake TS3 on the port side, as it is expected to make a constant port turn during the entire simulation.

In this scenario, the ownship has a desired speed of 1.5 m/s. TS1 and TS2 has a constant speed of 1.5 m/s, and TS3 has a speed of 1 m/s in order to give the ownship a speed advantage.

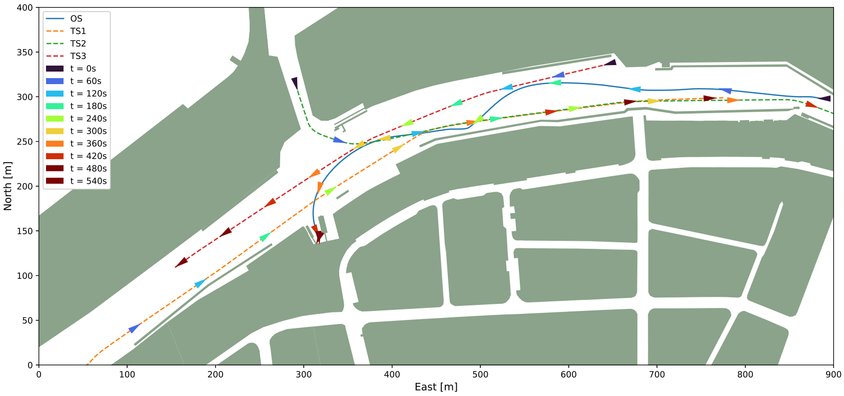


Figure 4.19: Trajectories for the BRATTØRA2 scenario. The colors indicate the timestamps throughout the simulation, and the arrows indicate the heading angle of the respective vessels involved.

The trajectories resulting from simulating the BRATTØRA2 scenario are shown in Figure 4.19. As shown in the figure, the ownship gives way to TS2 when it is clear that TS2 poses a head-on encounter. While giving way to TS2, the ownship maneuvers up behind TS3. After this, something that is not very COLREGs-compliant happens. According to results obtained from the simulation, TS1 is detected at $t = 208$s, which is after the green arrow indicating $t = 180$s and before the arrow indicating $t = 240$s. The ownship then goes on to turn towards its port side, instead of staying behind TS3 and passing TS1 on the target ship's port side. The ownship maneuvering towards the opposite side of the canal instead of passing TS1 on its port side is a violation of the COLREGs, and would cause a potentially dangerous situation in a real-world setting, as it would cause unnecessary confusion for especially TS1. The reason for this maneuver is however likely not the trajectory planner itself, but rather the way the intention of TS1 is predicted. An illustration of how the trajectory of TS1 is predicted around the time TS1 is detected is shown in Figure 4.20. This figure shows how the trajectory of TS1 is predicted based on the assumption it will keep its course and speed, and shows why the trajectory planner plans a trajectory that makes the ownship maneuver to its port side instead of its starboard side. It could be argued that the ownship still should have been able to plan a trajectory that would have passed TS1 on its port side, but the way TS1's trajectory is predicted at least doesn't help. Predicting TS1's trajectory based on the assumption that it will keep a constant distance from the center line of the canal could be an example of an improvement, but this is not trivial to do.

The distances between the ownship and the three target ships throughout the simulation is shown in Figure 4.21. As seen in the figure, the ownship once again enters the TS domain of a target ship, and somewhat surprisingly based on TS3 arguably being the target ship that poses the least risk of collision in this scenario, it looks like it is the target ship the ownship comes closest to. The ownship also comes close to TS1, and the DCPA between the ownship and TS1 is 15.2m, which is just outside of the TS domain of TS1. This again

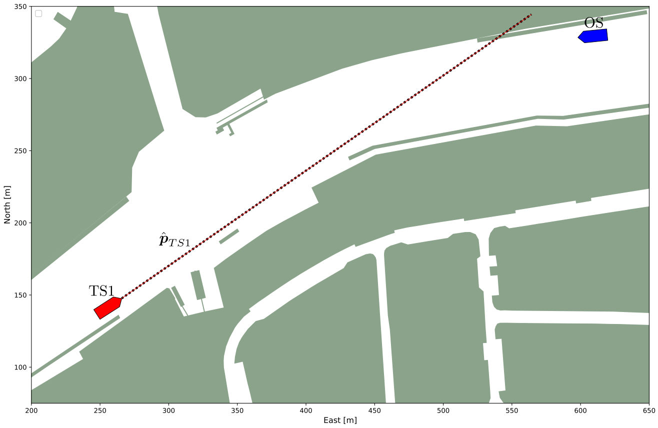


Figure 4.20: Illustration of how the trajectories of target ships are predicted, more specifically how the trajectory of TS1 in the BRATTØRA2 scenario is predicted. The predicted trajectory $\hat{\boldsymbol{p}}_{TS1}$ is shown as a black dotted line, TS1 is the red vessel and the ownship is shown in blue.

shows that the trajectory planner has similarly to in the BRATTØRA1 scenario resulted in a trajectory that is not feasible w.r.t. the OCP constraints. One interesting observation is seen by looking at the trajectories between $t = 300$s and $t = 360$s. It can be seen that the ownship is behind TS3 at $t = 300$s, but has overtaken it at $t = 360$s. By looking at Figure 4.21, this coincides with when the ownship and TS3 are closest to each other. This is not a very safe maneuver by the ownship, as it optimally should have just followed TS3 before turning port to its destination while still being behind TS3. In other words, TS3 should probably never have been overtaken in this scenario.

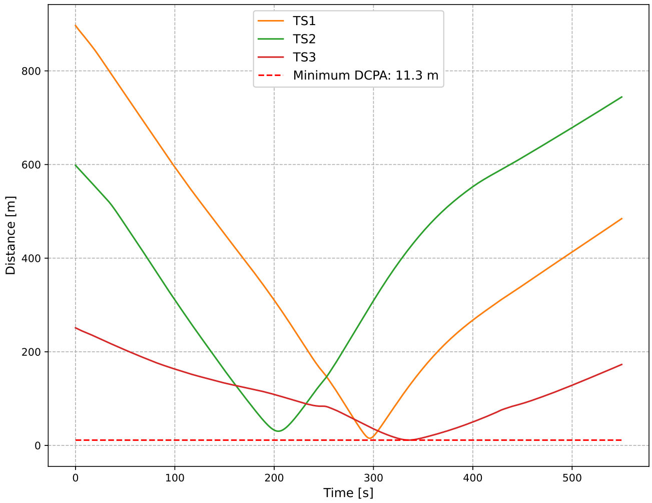


Figure 4.21: Distance between the ownship and target ships in the BRATTØRA2 scenario.

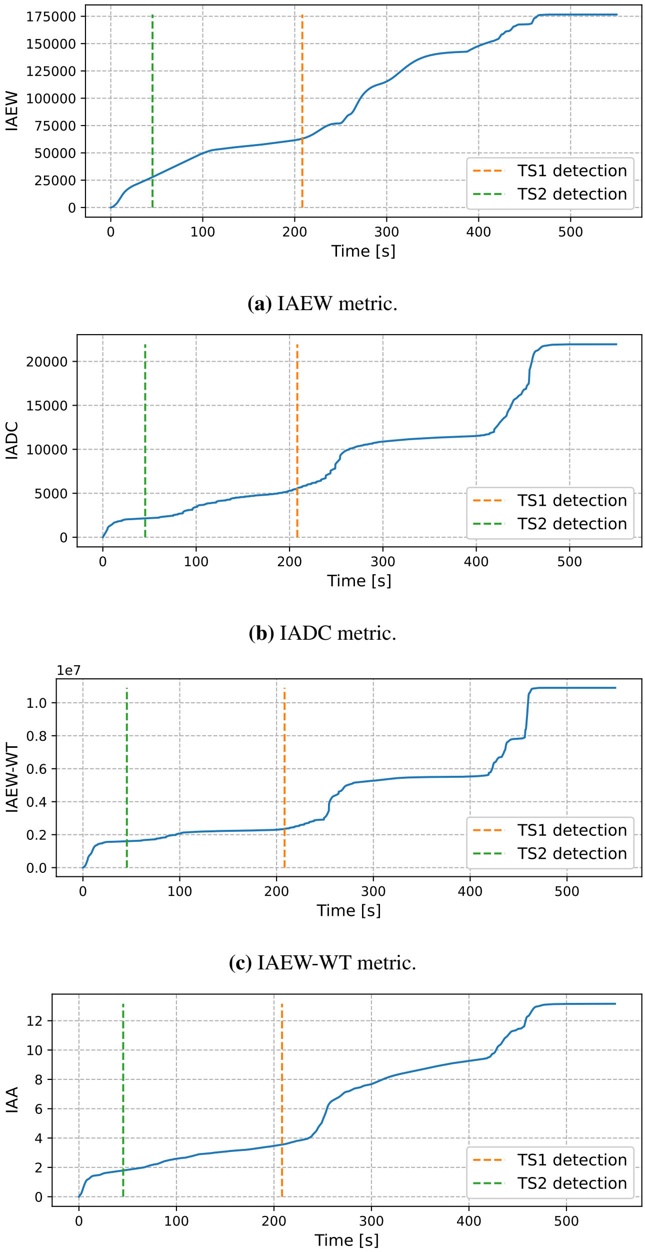
The COLREGs-specic performance metrics are shown in Table 4.11. For TS3, only the safety metric $\mathcal{P}_{safety}$ is calculated. The reason for this is that the other scores are deemed most relevant in either head-on or crossing encounters, and the parameters needed to calculate $\mathcal{P}'_{\Delta_{\chi_{app}}}$ and $\mathcal{P}_{delay}$ are not trivial to calculate for an overtaking encounter. As seen in the table, the maneuver made by the ownship to avoid collision with TS2 is not deemed to have been made in ample time, nor is it deemed very apparent using a threshold of 60°. For TS1, the maneuver is on the other hand deemed both readily apparent, and to have been made in ample time. The $\mathcal{P}_{safety}$ scores are as expected 0 for both TS1 and TS2, as the ownship keeps out of the TS domain of both vessels. For TS3 it is however seen that the maneuver is deemed more unsafe, which is reasonable based on Figure 4.19 and Figure 4.21.

The IAEW, IADC, IAEW-WT and IAA scores from simulating the BRATTØRA2 scenario are shown in Figure 4.22, where the times of detecting TS1 and TS2 are indicated. As seen in the plots, detecting TS2 does not lead to a significant increase in any of the metrics, at least not compared to what is seen after TS1 is detected. This fits well with what is seen in Table 4.11, as the maneuver made to avoid TS2 is performed somewhat late, and is not deemed very apparent. It is on the other hand seen in the figure that detecting TS1 leads to a larger increase in all metrics, indicating an increased power consumption, actuator usage and absolute acceleration after TS1 is detected. This is also in line with the low scores for TS1 in Table 4.11.

The runtime of the trajectory planner when simulating the BRATTØRA2 scenario is shown in Figure 4.23. This figure shows that the runtimes when including one more target ship, and encounters that to some degree leads to conflicting rules, makes the trajectory planner significantly slower than what is seen earlier. It is for the most part able to set up and solve the OCP in under 4 seconds, but the overall runtime is above 2s for almost all runs. There is also a peak of over 7 seconds, which is not very well suited for online trajectory planning, at least not if such peaks happen frequently.

Metric	TS1	TS2	TS3
$\mathcal{P}_{delay}$	0.14	0.75	-
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0.16	0.68	-
$\mathcal{P}_{safety}$	0	0	0.18

Table 4.11: COLREGs-specific performance metric scores resulting from simulating the BRATTØRA2 scenario.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.22: IAEW, IADC, IAEW-WT and IAA scores resulting from the BRATTØRA2 scenario. The times where TS1 and TS2 are detected are indicated.

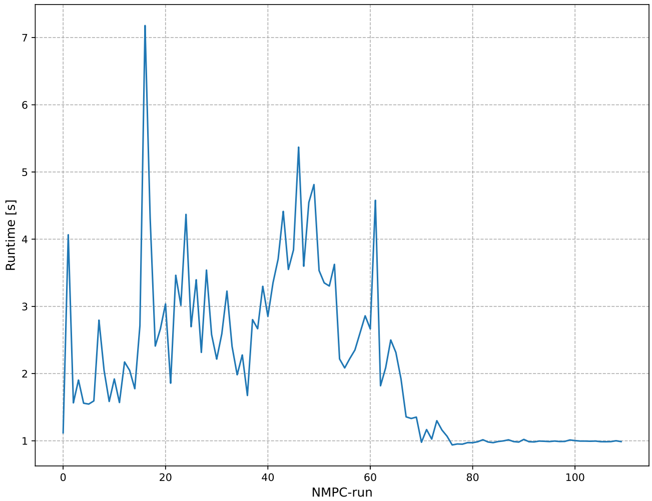


Figure 4.23: Trajectory planner runtime when simulating the BRATTØRA2 scenario.

4.1.4 Discussion

The batch simulations performed in this section shows a behaviour of the trajectory planner that at least can be argued to be COLREGs-aware. For the head-on-, give-way crossing- and overtaking encounters, the trajectory planner results in the ownship giving way to the target ship in all cases while satisfying the minimum distance to keep from the target ship. In order to avoid confusion at sea, the ownship shall show its intentions clearly and early in the encounter. In the batch simulated head-on encounters, the ownship shows its red sidelight early, and continues to give way to the target ship in a way that complies with Rule 14 and Rule 16 of the COLREGs. Also in the overtaking scenarios, the ownship is shown to give way to the target ship. In these scenarios, the ownship overtakes the target ship on the port side in some cases, and on the starboard side in others. This is in compliance with Rule 13 of the COLREGs. One problem that is seen in the overtaking scenarios is that the ownship does to some extent make itself a crossing vessel in some of the simulations, which is a violation of Rule 13. The side of the target ship the ownship shall pass should optimally be based on the course of the target ship to a larger extent than what is shown in the overtaking batch simulations, in order for the ownship to not become a crossing vessel during an overtaking.

For the give-way crossing encounters, it is harder to conclude that the behaviour is compliant with Rule 15 and Rule 16 of the COLREGs. In some of the simulations in the crossing encounters, the trajectory planner plans trajectories that forces the ownship to increase its speed and cross in front of the target ship. This can be said to be a violation to Rule 15 of the COLREGs, as a give-way vessel shall avoid crossing in front of the stand-on vessel. In practice there is however a threshold for when it is acceptable to cross in front of a stand-on target ship. It is hard to argue that a give-way vessel crossing e.g. 100m in front of a stand-on vessel is a violation of the COLREGs for the particular speeds considered in this section. There does however come a point where crossing in front of the stand-on vessel will lead to confusion and potentially dangerous situations at sea. The trajectory planner used in this work uses a constant minimum distance to keep from a target ship when defining the constraints, practically making up a circular distance about the target ship that should not be entered. In give-way crossing situations it can however be argued that this distance should be larger in front of the stand-on vessel, meaning the minimum distance used to define the TS domain etc. should have a more elliptical shape, with the major axis aligned with the course of the stand-on vessel.

The stand-on crossing batch simulations are as mentioned as much a test of how compliant the planner is with Rule 17 of the COLREGs as its compliance with Rule 2. In the results shown, it is seen that the ownship keeps its stand-on obligation, until it is clear that the target ship will not keep its give-way obligation. This result is of course dependent of what parameters like $d_{crit\_stand\_on}$ and $t_{stand\_on\_reaction\_time}$ is set to. It is however shown that as long as these parameters are set reasonably, the trajectory planner is capable of complying with the stand-on obligations stated by Rule 17, as well as departing from the rules in order to avoid collision in situations where it is needed, which is in compliance with Rule 2.

Following the batch simulations, the effect of introducing windows of reduced cost and the port-side constraint described in Section 3.8 is assessed. These results shows that introducing tools such as windows of reduced cost and the port-side constraint directly affects to what extent a maneuver planned by the trajectory planner is readily apparent, and made in ample time, at least according to the metrics used. Introducing windows of reduced cost seems to be a tool that is safe to use, and that positively affects the COLREGs-compliance of the trajectory planner. Increasing the COLREGs-compliance will in turn make the ownship's intentions clear at sea, which can be argued to make for a safer transit overall. The reasoning behind this is that clear intentions are as important at sea as when driving a car in traffic. If all vessels show clearly what they intend to do, the risk of accidents as a result of confusion between vessels is effectively reduced.

The port-side constraint is also shown to effectively affect how readily apparent a maneuver is. For this tool, it is however seen through relevant performance metrics that the control error, power consumption, actuator wear-and-tear and absolute acceleration during the transit is increased proportionally as the distance from the target ship to the port-side constraint is increased. This raises the question if the port-side constraint is worth using, and if the benefits from only introducing windows of reduced cost are enough. In the complex simulations performed using the milliAmpere1 model, the port-side constraint isn't used at all, as it turned out it is not applicable in the presence of multiple target ships. The reason for this is that the second window of reduced cost ends at the maximum t_{crit}^{exit}, and in the presence of multiple target ships this can be large. This means the port-side constraint is active too long, restricting the feasible region for the trajectory planner too much. Both windows of reduced cost and the port-side constraint are however shown to represent tools that effectively affects the trajectory planners compliance with Rule 8 and Rule 16 of the COLREGs. One or both of the tools might not be needed in all contexts, but can rather be seen as tools that are available in a larger toolbox when developing MPC-based trajectory planners in the context of COLREGs-compliance. The port-side constraint will of course need more work if it is to be applicable in the general sense.

For the complex scenarios presented in this simulation, the trajectory planner shows a COLREGs-compliant behaviour in most encounters. The exemption is the head-on encounter with TS1 in the BRATTØRA2 scenario. The ownship does in this encounter end up performing a maneuver that is compliant with Rule 14 regarding head-on encounters by passing TS1 on its port side. Before performing this maneuver, it is however not in compliance with e.g. Rule 8, as the intentions of the ownship are confusing during this encounter. The ownship does also come close to TS3 in this scenario, which is a vessel that *can* be overtaken. In an overtaking encounter, the ownship is not forced to overtake, but is rather allowed to do so if the circumstances allow it. It can in the overtaking scenario with TS3 be observed that the ownship switches between overtaking and not, which would likely be very confusing for TS3 in a real-life setting. The ownship does however avoid collisions in all encounters, and it keeps a good distance to TS1 and TS2. In the BRATTØRA1 scenario it can be argued that the ownship does what it should have in a real-life setting, and that it complies with the relevant rules.

Another aspect that becomes quite evident in the complex simulations is that the way the target ships' trajectories are predicted is not sufficient in more complex scenarios. In the simulations performed prior to the complex simulations, all target ships have a constant course and speed throughout the simulations, which fits the assumptions made when predicting the target ship trajectories perfectly. In the complex simulations, this assumption is however not met by any of the target ships. This is seen to cause problems, especially in the BRATTØRA2 scenario where this is likely the reason for some questionable planning by the trajectory planner. It can also be said that estimating the target ship trajectories accurately is important when using tools such as windows of reduced cost and the port-side constraint, as these tools are dependent on good estimates of t_{crit}^{enter} and t_{crit}^{exit}. Estimating the intent of target ships is an essential part of obtaining good trajectory planning, and is a relevant subject for future work. Obtaining good estimates of the target ships trajectories is however not trivial, and requires assumptions to be made carefully.

Through the complex scenarios it is also seen that the runtime of the trajectory planner is in the higher end of what is preferred, but as technology advances forward and new runtime-optimization tools become available, this problem might be reduced in the near future. The increased runtime can also be reduced just by implementing the trajectory planner in C++ instead of Python.

4.2 High-speed simulations: Scaled Otter USV

This section presents the simulation results from using the high-speed trajectory planner, i.e. the trajectory planner utilizing the nonlinear model described in Section 3.4.2. The simulations are run using the scaled Otter USV model, where the speed and heading of the vessel is controlled as described in Section 3.2.2. The references for the speed and heading controller are generated using the target tracking method described in Section 2.6.3.

4.2.1 Batch simulations

The high-speed batch simulations are performed similarly to the low-speed batch simulations. The exceptions are that the high-speed simulations are run over longer distances due to the higher transit speed, as the ownship now has a speed of 10 m/s, and the target ship speed is set to 7 m/s. Also here, batches of 60 simulations are run for two head-on-, overtaking-, give-way crossing- and stand-on crossing encounters. In the first simulation, the ownship is going from west to east, 400m north. The subsequent simulations, the ownship starts $\approx$ 13m south of its previous starting position. The initial conditions for the simulation batches are summarized in Table 4.12.

The static parameters used by the trajectory planner for the high-speed simulations are summarized in Table 4.13, and the encounter-specific parameters are the same as in Table 4.3 used in the low-speed simulations.

The static parameters related to the guidance and control of the scaled Otter USV described in Section 2.6.3 and Section 3.2.2 respectively are given in Table 4.14. All parameters in this table is tuned based on early simulations performed in this work.

Scenario name	TS starting position	TS goal position
HO_3	(215, 1740)	(-215, -1740)
HO_4	(-215, 1740)	(215, 1740)
OT_3	(215, -1740)	(-215, 1740)
OT_4	(-215, -1740)	(215, 1740)
GW_3	(-1620, 670)	(1620, -670)
GW_4	(-1620, -670)	(1620, 670)
SO_3	(1620, 670)	(-1620, -670)
SO_4	(1620, -670)	(-1620, 670)

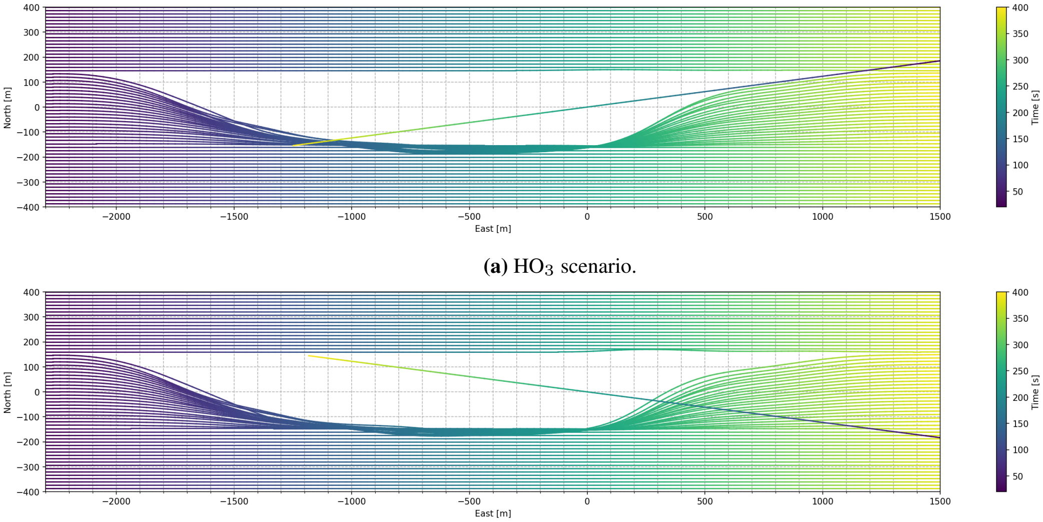
Table 4.12: Initial conditions for the batch simulated scenarios in this work performed using the scaled Otter USV. All position coordinates are given in the NED frame.

Parameter	Value	Unit
N_p	400	[-]
h	2	[s]
dt_{nmpc}	4	[s]
K_k^p	2.5×10^{-6}	[m^{-2}]
K_k^a	60	[m^{-4}s^4]
K_k^{tr}	30	[rad$^{-2}s^2$]
κ	0.1	[-]
l	150	[m]
$t_{after\_passing}$	30	[s]
$t_{stand\_on\_reaction\_time}$	30	[s]
$t_{maneuver}$	60	[s]
d_{crit}	150	[m]
$d_{crit\_stand\_on}$	75	[m]
K_{CRW1}^p	10^{-5}	[-]
K_{CRW2}^p	10^{-5}	[-]
K_{CRW1}^a	0.5	[-]
K_{CRW2}^a	1	[-]
K_{CRW1}^{tr}	0.05	[-]
K_{CRW2}^{tr}	0.7	[-]
d_{port}	30	[m]
$\alpha_{crossing}$	$\pi/3$	[rad]

Table 4.13: Parameters for the high-speed batch simulations.

Parameter	Value	Equation	Unit
Δ_p	150	(2.33)	[m]
ζ	0.7	(2.35)	[-]
k_p	25000	(3.8)	[Nm^{-1}s]
k_i	2800	(3.8)	[Nm^{-1}]
k_p	95000	(3.9)	[Nrad^{-1}]
k_d	100	(3.9)	[Nrad^{-1}s]
k_i	1000	(3.9)	[Nrad^{-1}s^{-1}]

Table 4.14: Numerical values of parameters related to the guidance and control of the scaled Otter USV model.



(a) HO$_3$ scenario.

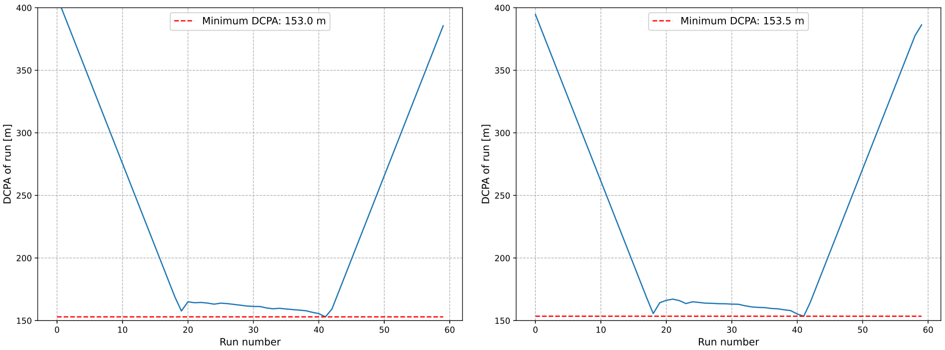
(b) HO$_4$ scenario.

Figure 4.24: High-speed head-on batch simulations.

Head-on encounter

The simulated head-on scenarios HO$_3$ and HO$_4$ are shown in Figure 4.24. It can be seen from the figure that the ownship makes a small turn early in the encounter, in order to give way to the target ship in both scenarios. This will, similarly to what was seen in the low-speed simulations, make the ownship show its red sidelight early on, which is in compliance with Rule 14 regarding head-on encounters, and Rule 16 regarding the action of a give-way vessel. It can also be seen in the figure that including the heading angle in the MPC-planner gives turns with a lower turn-rate, which makes the planner better suited for underactuated vessels moving at higher speeds. One of the trajectories at $\approx$ 150m-160m north in Figure 4.24b makes a very slight port turn around the 150m contour of the target ship formed by the constraints. This is however not a violation of the COLREGs, as the distance between the ownship and target ship is $>$ 150m. The intention of the ownship in this particular trajectory should also be clear, as it mostly shows its green sidelight and keeps its course and speed.

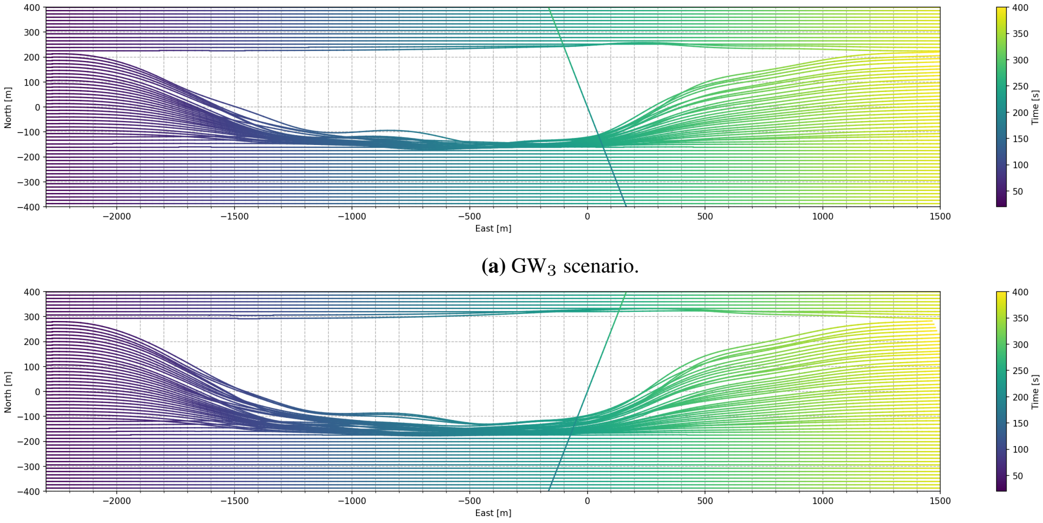
The DCPA between the ownship and target ship for all simulations is shown in Figure 4.25. As seen in these figures, the ownship keeps its minimum distance of 150m to the target ship in all simulations. The contour-hugging-like behaviour is also once again evident, as the distance bounces over the 150m line for the middle simulations. This again shows the strength of MPC-based trajectory planners, as keeping the minimum distance to the target is always ensured by hard constraints, as long as the control system paired with a suitable guidance strategy is able to make the ownship follow the trajectory closely.



(a) HO_3 scenario.

(b) HO_4 scenario.

Figure 4.25: DCPA between the ownship and target ship in the high-speed head-on batch simulations.



(a) GW_3 scenario.

(b) GW_4 scenario.

Figure 4.26: High-speed give-way crossing batch simulations.

Give-way crossing encounter

The results from batch simulating the high-speed give-way crossing scenarios are shown in Figure 4.26. As seen in this figure, the ownship gives way to the crossing target ship in both the GW_3- and GW_4-scenarios. One can also argue that the trajectories from the high-speed planner gives way in a more clear manner than for the low-speed planner. The maneuver starts early, and the trajectory planner plans trajectories that goes abaft the target ship to a larger degree than what was done in the low-speed simulations.

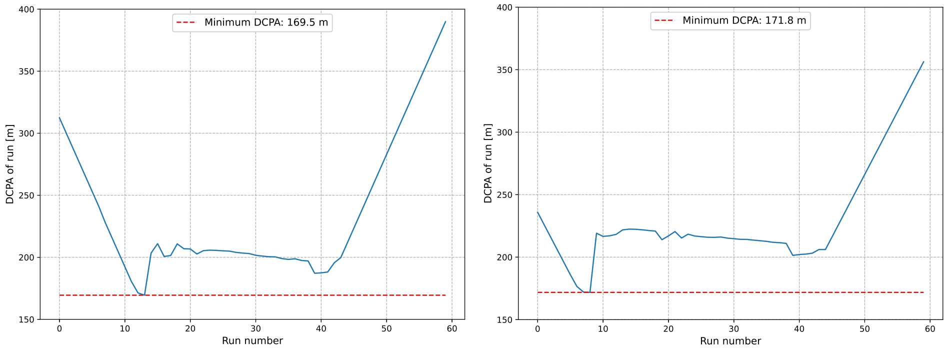
It should be mentioned that the behaviour of the ownship in Figure 4.26 resembles an under-damped system in some of the simulations, as the trajectories are quite wavy and the Otter USV is seen to struggle going straight forward. This might however not be a result of the trajectory planner itself, but possibly the scaled Otter USV model paired with the guidance and control strategy. Some of the added dynamics that stem from higher speeds are likely not transferable through the scaling, as discussed in Section 3.2. The scaled Otter USV is also 15m long and $\approx$ 8m wide, which at least is an unusual shape for a high-speed vessel.

A possible explanation for the wavy behaviour might also involve the trajectory planning itself, and the port-side constraint. The port-side constraint is seen to create problems for the high-speed planner, as mentioned in Section 3.8.3. In the give-way crossing scenario, this constraint to some extent sets up a wall in the horizon that the ownship is not allowed to pass before some point in time, t_{crit}^{exit} in this case. This wall is shifted counter-clockwise by some angle $\alpha_{crossing}$, but it is still a stricter constraint in this scenario compared to head-on scenarios. If the first estimate of this time is too short, and the ownship could have passed it earlier in the horizon, the constraint will lead to the ownship having to slow down. Because accelerations are penalized more than the turning rate in the planner, it might to some extent use turning instead of acceleration to buy itself some time. This once again shows a drawback with the port-side constraint, and that this constraint should be added to the pipeline with care. It might e.g. be suited only for head-on encounters when using the high-speed planer. The problem could also be that the $\alpha_{crossing}$-angle is too small, meaning the constraint should be shifted by a larger angle. Choosing this angle is however not trivial to do in a general sense, and shows a sensitivity to the parameter-setting in the trajectory planners proposed in this work. There is a significant amount of static parameters that needs to be set, and finding a good combination of parameters is not trivial, and should be done with care. Looking into a way of setting a subset of the parameters dynamically, i.e. based on the context and area of operation among other factors, should be considered in future work.

The DCPA between the ownship and target ship in the simulations shown in Figure 4.26 are shown in Figure 4.27. These distances does looks slightly different than the distances shown in earlier sections. There is a clear dip in the DCPA around the 13th simulation for the GW$_3$ scenario, and around the 8th simulation in the GW$_4$ scenario. By taking a look at Figure 4.26, these dips likely coincide with when the trajectory planner starts planning trajectories abaft the target ship, as it is clear this happens in earlier simulations for the GW$_4$ simulations than the GW$_3$ simulations. This means the DCPA in both scenarios likely stems from the last trajectory before the trajectory planner plans trajectories abaft the target ship. The DCPA in both scenarios is however well above the minimum distance l set to 150m.

Overtaking encounter

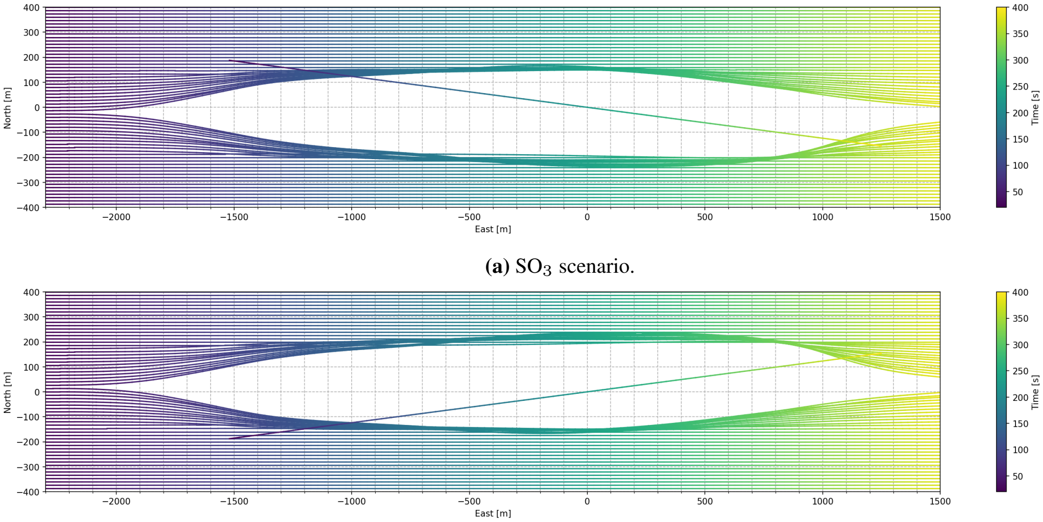
The batch simulations resulting from the overtaking scenarios OT$_3$ and OT$_4$ are shown in Figure 4.28. As seen in the figure, the trajectories are very similar to the low-speed overtaking scenarios from Figure 4.5. The only difference is that the maneuver starts earlier and lasts longer, as can be expected when the vessels are moving at significantly higher speeds compared to the low-speed simulations. All in all, this behaviour can be said to be in compliance with Rule 13 regarding overtaking situations. One problem that is still



(a) GW$_3$ scenario.

(b) GW$_4$ scenario.

Figure 4.27: DCPA between the ownship and target ship in the high-speed give-way crossing batch simulations.

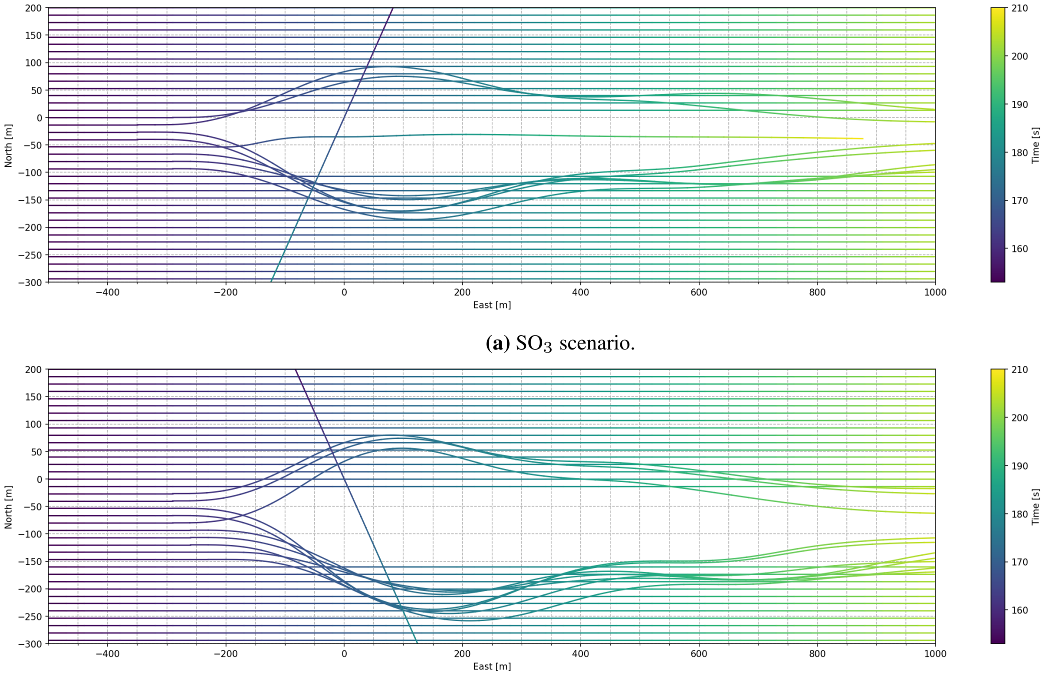


(a) SO$_3$ scenario.

(b) SO$_4$ scenario.

Figure 4.28: High-speed overtaking batch simulations.

apparent is however that the ownship can be said to become a crossing vessel at some of the southern overtakings in Figure 4.28a, and in some of the northern overtakings in Figure 4.28b. It is however hard to state the significance of this potential violation without experience at sea.



(a) SO$_3$ scenario.

(b) SO$_4$ scenario.

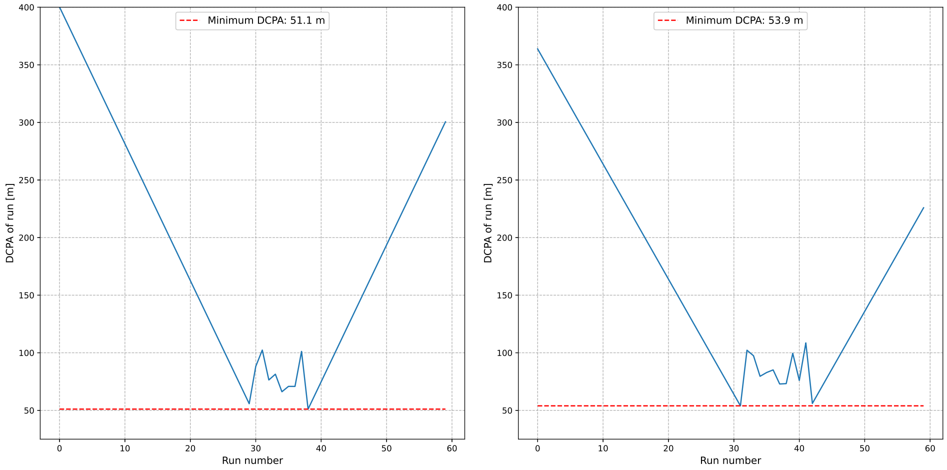
Figure 4.29: High-speed stand-on crossing batch simulations.

Stand-on crossing encounter

A subset of the high-speed stand-on crossing simulations are shown in Figure 4.29. From this figure, it is seen that the trajectory planner plans a maneuver approximately 300m before it crosses 0m east, as decided by the $t_{stand\_on\_reaction\_time}$ parameter which is set to 30s in these simulations. The DCPA in each of the simulations is shown in Figure 4.30, where it is clear that it keeps a distance above the minimum distance l of 50m in all simulations. The DCPA between the ownship and the target ship might also stem from the simulations where the ownship does not maneuver.

One trajectory in Figure 4.29a stands out from the rest. It is seen that one of the trajectories doesn't turn as much as the others, and seem to avoid collision by slowing down. This is found to be simulation number 34, and the speed and heading of simulation 34 can be compared to the speed and heading of simulation 35 in Figure 4.31, which shows the speed and heading of the ownship during two minutes surrounding the COLAV maneuver. From Figure 4.31a, it is clear that the trajectory planner slackens the speed of the ownship during simulation 34, instead of keeping it like in simulation 35. It is also clear from Figure 4.31b that simulation 35 involves more turning. The reason why this happens is hard to say, especially as it only happens in one of the simulations, where the simulations performed before and after does not lead to a similar behaviour.

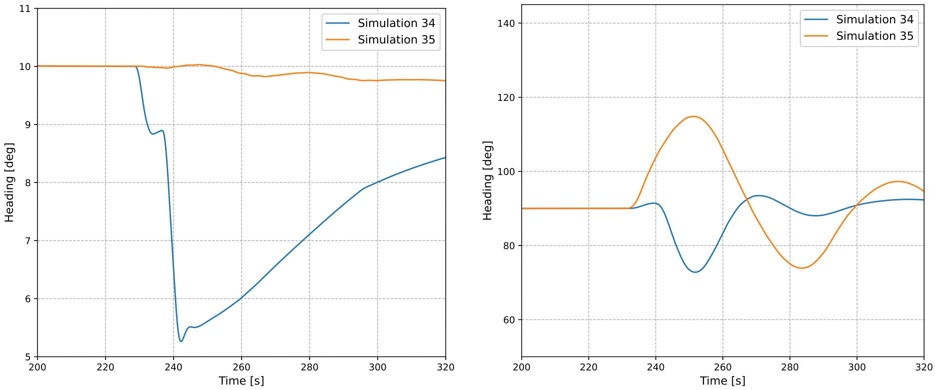
Whether it is best to change the speed or heading in these particular simulations can



(a) SO_3 scenario.

(b) GW_4 scenario.

Figure 4.30: DCPA between the ownship and target ship in the high-speed stand-on crossing batch simulations.



(a) Speed.

(b) Heading.

Figure 4.31: Comparison between the speed and heading resulting from simulations 34 and 35 in the high-speed stand-on crossing simulations.

be discussed. A maneuver in any way, whether it be a change in speed or heading, is a violation of the stand-on obligation of the ownship. Both maneuvers does however comply with Rule 2, as a collision is avoided in all simulations. Rule 17 does also allow for a departure from the stand-on obligations in order to avoid collision, as mentioned in Section 2.1.

4.2.2 Effect of windows of reduced cost and port-side constraint

Similarly to what was done for the low-speed simulations, this section presents the effect of introducing the windows of reduced cost and the port-side constraint in the high-speed trajectory planner. For the high-speed simulations, the HO_3 scenario is chosen to assess the effect of the windows of reduced cost and the port-side constraint. The reason for choosing this scenario is that the high-speed results are comparable to the low-speed results in this way.

The control-system-specific IAEW, IADC, IAEW-WT and IAA metrics are also calculated in a similar manner to what was done for the low-speed simulations, in accordance with what is described in Section 2.9.

The COLREGs-specific performance metrics are calculated in a similar manner to what was done for the low-speed simulations, the only difference being that the parameters used to calculate the safety score $\mathcal{P}_{safety}$ are changed to fit the context of higher speeds. When the ownship and target ship are moving at higher speeds, the distance between them should be higher than for lower speed to achieve the same safety score.

The parameters used to calculate the COLREGs-specific parameters are summarized in Table 4.15, and the rest of the parameters are the same as in Table 4.3, Table 4.13 and Table 4.14.

Introducing windows of reduced cost

The trajectories of the ownship and target ship with and without using windows of reduced cost are shown in Figure 4.32. This figure shows a similar result to what was obtained by introducing the windows of reduced cost in the low-speed trajectory planner, as seen in Figure 4.8. It is clear that introducing windows of reduced cost has the intended effect of making the trajectory planner plan action earlier in the encounter, which is in compliance with Rule 8 and Rule 16.

The resulting scores of the COLREGs-specific metrics are shown in Table 4.16. As seen from the table, introducing windows of reduced cost has a significant effect on the $\mathcal{P}_{delay}$-score, meaning the maneuver is made in ample time to a larger degree than without using windows of reduced cost. It is however seen that the maneuver is not deemed readily apparent, as the $\mathcal{P}'_{\Delta_{\chi_{app}}}$ is very similar in both simulations. This is likely a result of the high-speed trajectory planner penalizing the turning rate of the vessel, which is not done directly in the low-speed trajectory planner. The high-speed trajectory planner is intended to generate trajectories that are more easily followed by an underactuated vessel, but an underactuated vessel would likely be able to follow the trajectory even if the turning rate was higher than shown in Figure 4.32. It is also worth mentioning that the scores

Parameter	ϵ_{chi}	$\Delta_{\chi_{app}}$	R_{min}	R_{nm}	R_{col}	γ_{nm}	γ_{col}
Value	10	30	150	50	20	0.25	0.75
Unit	[deg]	[deg]	[m]	[m]	[m]	[-]	[-]

Table 4.15: Summary of the parameters needed to calculate COLREGs-specific performance metrics for the high-speed simulations using the scaled Otter USV model.

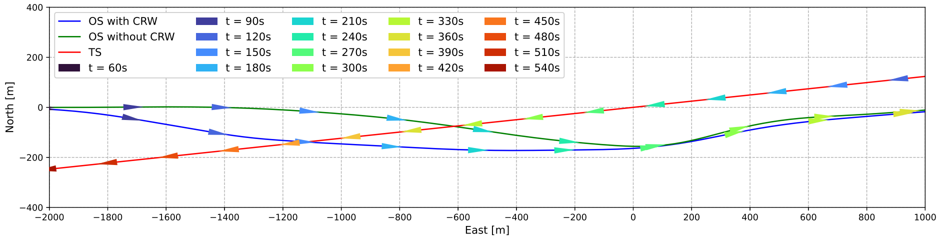


Figure 4.32: Trajectories of the ownship with and without using windows of reduced cost, as well as the target ship trajectory.

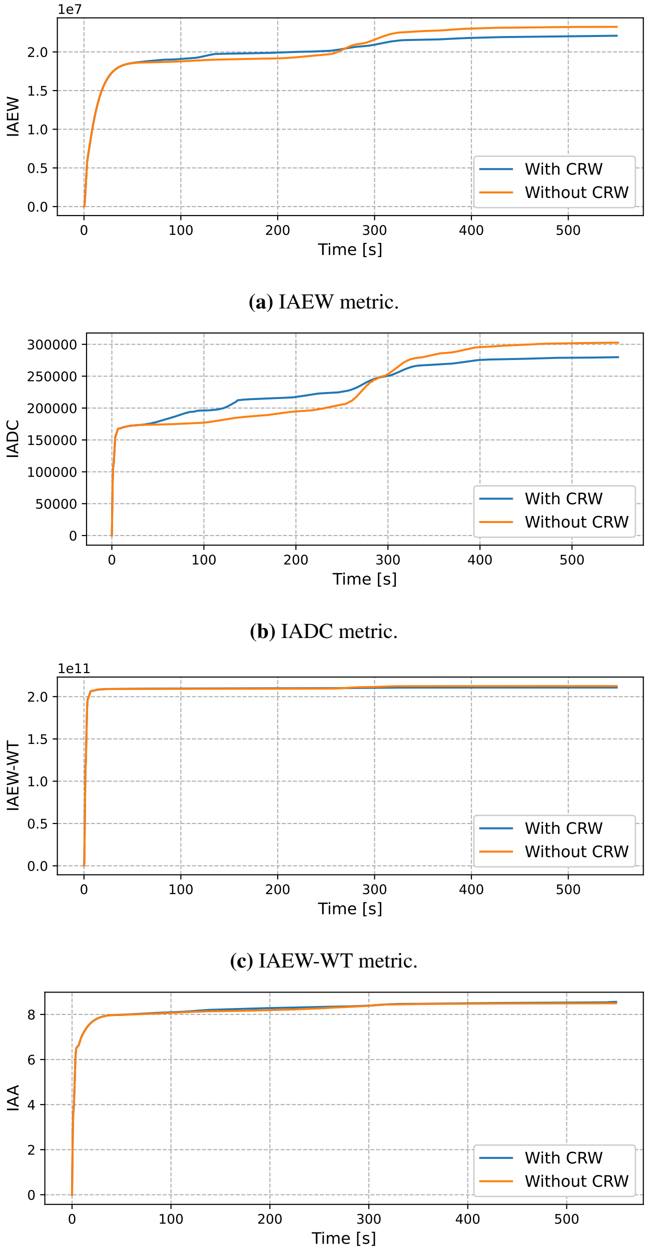
shown for $\mathcal{P}'_{\Delta_{\chi_{app}}}$ in Table 4.16 are calculated using a threshold of only 30° for what a readily apparent maneuver is. This is as argued in [34] in the lower end of what would be accepted as a readily apparent maneuver, meaning the high-speed trajectory with the static parameters chosen as in Table 4.13 penalizes the turning rate too much.

Another result that is evident from Table 4.16 is that the $\mathcal{P}_{safety}$ score is perfect. This once again shows a strength of MPC-based trajectory planning that allows for defining the distance to be kept from other vessels directly in the constraints.

The IAEW, IADC, IAEW-WT and IAA metrics from the simulation performed with and without using windows of reduced costs are shown in Figure 4.33. From these plots, it is shown that introducing windows of reduced costs does not have as large of an effect on these metrics compared to in the low-speed simulations. Contrary to in the low-speed simulations, the control error, power consumption and actuator wear-and-tear is actually lower when introducing windows of reduced costs, as seen by the IAEW and IADC plots. This is likely because the maneuvering is to a large degree done by slightly turning in the high-speed simulations. In the low-speed trajectory planner, maneuvering is forced to be done by accelerating. Accelerating can mean also mean turning, but in the low-speed planner there is no possibility of directly incentivizing the trajectory planner to only turn instead of changing the speed. This can also be seen from the IAA plot in Figure 4.35d, which shows that most of the acceleration is done early during the transit.

Metric	Using CRWs	Without using CRWs
$\mathcal{P}_{delay}$	**0.30**	0.77
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	**0.81**	0.86
$\mathcal{P}_{safety}$	0	0

Table 4.16: COLREGs-specific performance metric scores with and without using windows of reduced cost for the high-speed simulation. The metrics are calculated using the parameters given in Table 4.15. The best results are bold.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.33: IAEW, IADC, IAEW-WT and IAA scores with and without using windows of reduced cost in high-speed simulations.

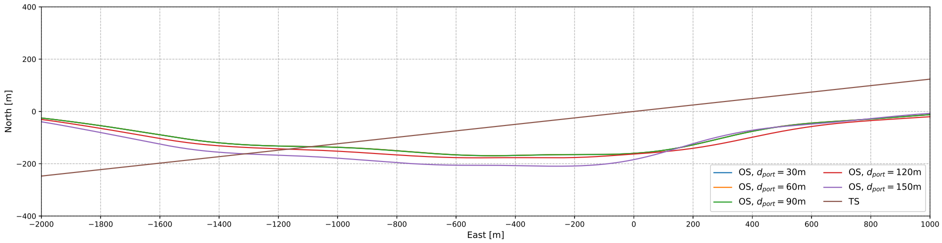


Figure 4.34: Trajectories of the ownship with five choices of d_{port} and the target ship trajectory.

Adding the port-side constraint

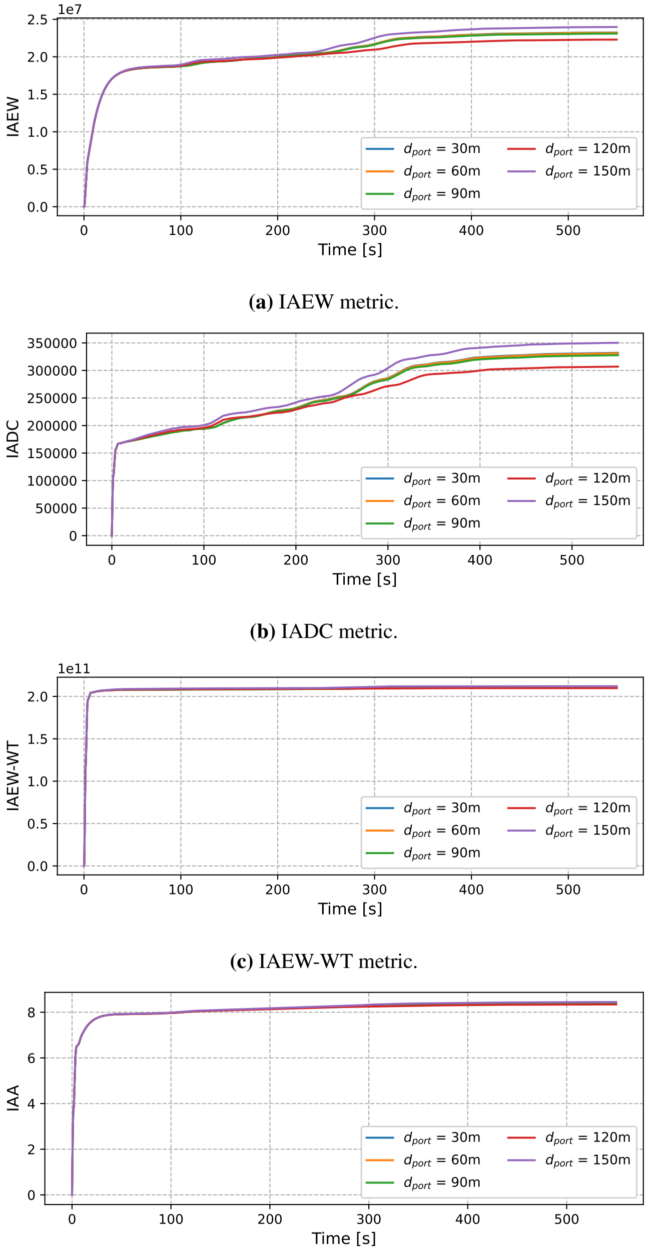
For the high-speed simulations, the effect of adding the port-side constraint is assessed in a similar manner to what was done for the low-speed simulations, except the d_{port} parameter is higher. For the high-speed simulations, the d_{port} parameters is varied from 30m to 150m, in steps of 30m. Figure 4.34 shows the result of introducing the port-side constraint for these choices of d_{port}. As seen in the figure, increasing d_{port} from 30m to 150m has next to no effect on the trajectories, and some of the trajectories completely overlap each other. The only effect can be seen when $d_{port} = 150$m, but the trajectory in this case is very similar to the ones resulting from other choices of d_{port}.

The COLREGs-specific performance metrics resulting from the choices of d_{port} are shown in Table 4.17. As seen from the table, the performance metrics are very similar, as expected from Figure 4.34. Also as expected from the figure, the best result comes from choosing $d_{port} = 150$m, although the $\mathcal{P}_{delay}$ score is very close to the other metrics. It can however be seen by comparing Table 4.17 to Table 4.16 that using the port-side constraint with $d_{port} = l = 150$m leads to better $\mathcal{P}_{delay}$ and $\mathcal{P}'_{\Delta_{\chi_{app}}}$ scores than when only using windows of reduced cost. It is however evident that the high-speed trajectory planner leads to less apparent maneuvers overall compared to the low-speed planner, which is likely due to the constraint being active for a shorter period of time for the high-speed planner.

The IAEW, IADC, IAEW-WT and IAA metrics obtained when varying d_{port} are shown in Figure 4.35. As seen from the figure, and as expected based on the other results from introducing the port-side constraint, there is not much difference in these performance metrics when varying d_{port}. The only simulation that stands out slightly is the simulation where $d_{port} = 150$m, where a slightly higher IAEW and IADC can be seen, meaning there is more actuator wear-and-tear when introducing the port-side constraint. This is reasonable, since the ownship maneuvers further to starboard in this simulation.

Metric	$d_{port} = 30$	$d_{port} = 60$	$d_{port} = 90$	$d_{port} = 120$	$d_{port} = 150$
$\mathcal{P}_{delay}$	0.26	0.26	0.26	0.23	**0.19**
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0.84	0.84	0.84	0.83	**0.58**
$\mathcal{P}_{safety}$	0	0	0	0	0

Table 4.17: COLREGs-specific performance metric scores with a varying value of d_{port}. The metrics are calculated using the parameters given in Table 4.15.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.35: IAEW, IADC, IAEW-WT and IAA scores with and without using windows of reduced cost and the port-side constraint with a varying d_{port} in high-speed simulations.

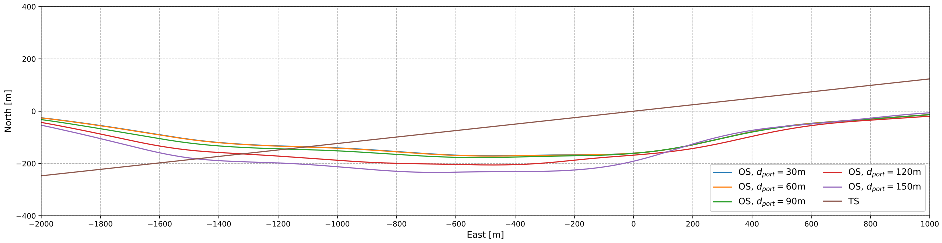


Figure 4.36: Trajectories of the ownship with five choices of d_{port} and the target ship trajectory, but with the port-side constraint being introduced earlier during CRW2 compared to Figure 4.34.

As mentioned in Section 3.8.3, the port-side constraint was during early simulations in this work shown to cause problems for the high-speed trajectory planer. This is why it is mostly added to the high-speed planner in order to ensure the correct side to pass the target ship is chosen by the planner. As described in Section 3.8.3, the constraint is active halfway into the second window of reduced cost, CRW2. Similar simulations to what was shown in Figure 4.34 was however run, but in these simulations the port-side constraint was activate 10% earlier in CRW2. The resulting trajectories from these simulations are shown in Figure 4.36, which shows a slightly bigger difference between the trajectories. These simulations also led to the performance metrics shown in Table 4.18. As seen in this table, introducing the port-side constraint earlier during CRW2 has a positive effect on the metrics for the higher choices of d_{port}, although the $\mathcal{P}_{delay}$-score is approximately the same. The $\mathcal{P}'_{\Delta_{\chi_{app}}}$-score is however improved significantly. This shows that the port-side constraint is a tool that can be used to effectively alter how apparent a maneuver is, as was the case for the low-speed planner. It must however be said that even though the $\mathcal{P}'_{\Delta_{\chi_{app}}}$-score is lowered by introducing the constraint earlier, the score is still significantly higher than some of the scores obtained by the low-speed planner. For the low-speed planner, a threshold of 30° led to $\mathcal{P}'_{\Delta_{\chi_{app}}}$ being zero for all choices of d_{port}. When using a threshold of 30° for the high-speed planner, some of the scores are still higher than when using a threshold of 60° on the low-speed planner, as seen in Table 4.7. The high-speed planner is however used in this work with the intention of planning trajectories with lower turning rate, so the scores can be expected to be higher.

Metric	$d_{port} = 30$	$d_{port} = 60$	$d_{port} = 90$	$d_{port} = 120$	$d_{port} = 150$
$\mathcal{P}_{delay}$	0.26	0.26	0.22	0.18	**0.16**
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0.84	0.84	0.83	0.74	**0.50**
$\mathcal{P}_{safety}$	0	0	0	0	0

Table 4.18: COLREGs-specific performance metric scores with a varying value of d_{port}, obtained by introducing the port-side constraint earlier during CRW2.

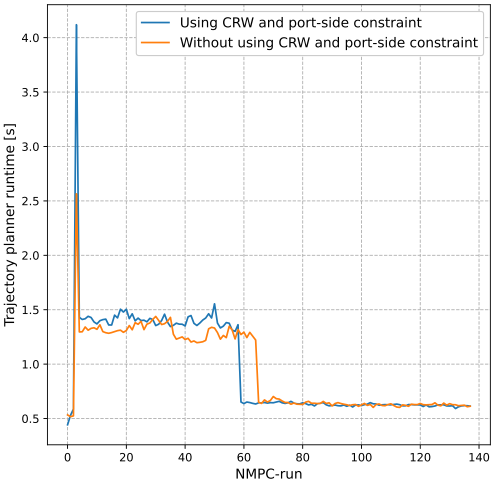


Figure 4.37: Runtime for problem formulation and solving for the high-speed trajectory planner with and without using cost-reduction windows (CRW) and the port-side constraint.

The runtime of the high-speed trajectory planner with and without using windows of reduced cost and the port-side constraint is shown in Figure 4.37, where the runtimes are from the simulations shown in Figure 4.34. As seen from the figure, the runtimes are not changed much by introducing the windows of reduced cost and port-side constraint. When using these tools, the planner is however seen to use significantly longer time at the beginning of the planning. This is most likely when the target ship is added to the set of prioritized vessels $\mathcal{N}_{pri}$. Compared to the runtimes of the low-speed planner seen in Figure 4.12, the runtime of the high-speed planner is higher. The low-speed planner without windows of reduced cost sets up the OCP and solves it in under one second per NMPC-run, and just above one second when using these tools. For the high-speed planner, the runtime is well over one second for all NMPC-runs until the ownship has passed the target ship and the encounter is classified as SF. The runtime of the high-speed planner is arguably acceptable, but it would be beneficial to reduce it in some way. This is because the high-speed planner in this work is run with a horizon of $N_p = 400$ steps and a time between discreatization steps of $h = 2$s. It is likely that a finer discretization would lead to a better behaviour for this planner, as the speed of the vessel is much higher than for the low-speed simulations. It could e.g. be beneficial to instead use a horizon of $N_p = 800$ steps and a discretization step of $h = 1$s, in order to get a smoother trajectory for the ownship to follow. This would however double the number of discretized steps, meaning even higher runtimes than what is shown in Figure 4.37 are expected. The most important is that the planning horizon is long enough, such that the complete encounter is covered within it.

4.2.3 Complex scenarios

Also for the high-speed simulations using the scaled Otter USV model, complex scenarios have been simulated in order to test the high-speed trajectory planner in the presence of multiple target ships and static obstacles in the form of land. The complex scenarios defined in this section are in a larger area than for the low-speed complex simulations, and involve larger vessels. It should be noted that the sizes of the target ships are not considered by the trajectory planner, meaning the distance to keep from target ships is defined by the l-parameter used to define the TS domain. Two scenarios are simulated using the high-speed scaled Otter USV, namely the MOSS scenario and the KIREKØY scenario. These scenarios differ in that the MOSS scenario is to a large degree in open sea, as only a couple static obstacles are considered. For the KIRKEØY scenario, there is however significantly more land-obstacles to consider. Due to this difference, some of the distance parameters used by the trajectory planner are changed from the MOSS scenario to the KIRKEØY scenario, in order to fit the environment better. The updated parameters for both scenarios are shown in Table 4.19.

It can be seen from the table that a distance of 50m should be kept to land in both scenarios. For the MOSS scenario, 200m should be kept to other vessels, because this scenario is meant to involve larger target ships. It can be argued that 200m is well above what could have been set as a minimum safety distance to keep from all target ships in this scenario, but the distance is however defined in this way in order to not only keep a *safe* distance to other vessels, but also a somewhat *comfortable* distance. For the KIRKEØY scenario the l-distance is smaller, due to the scenario not involving as large vessels as the MOSS scenario, as well as the presence of significantly more land-obstacles. It can also be seen that the static constraints are defined more frequently in the high-speed simulations due to the speed of the vessel being higher. The encounter-specific parameters are still the same as shown in Table 4.3, and the control- and guidance-specific parameters are as shown in Table 4.14. The COLREGs-specific parameters are calculated based on parameters given in Table 4.20.

Parameter	**MOSS scenario**	**KIRKEØY scenario**	**Unit**
N_p	400	-	[-]
h	2	-	[s]
dt_{nmpc}	4	-	[s]
dt_{stat}	2	-	[s]
d_{stat}	50	-	[m]
K_k^p	2.5×10^{-6}	-	[m^{-2}]
K_k^a	60	-	[m^{-4}s^4]
K_k^{tr}	30	-	[rad$^{-2}s^2$]
κ	0.1	-	[-]
l	200	75	[m]
$t_{after\_passing}$	30	-	[s]
$t_{stand\_on\_reaction\_time}$	30	15	[s]
$t_{maneuver}$	60	30	[s]
d_{crit}	200	75	[m]
$d_{crit\_stand\_on}$	75	40	[m]
K_{CRW1}^p	10^{-5}	-	[-]
K_{CRW2}^p	10^{-5}	-	[-]
K_{CRW1}^a	0.5	-	[-]
K_{CRW2}^a	1	-	[-]
K_{CRW1}^{tr}	0.05	-	[-]
K_{CRW2}^{tr}	0.7	-	[-]

Table 4.19: Numerical values of static parameters used in the high-speed complex scenario simulations. For the KIREKØY scenario, '-' indicates that the parameter is unchanged from the MOSS scenario.

Parameter	ϵ_{chi}	$\Delta_{\chi_{app}}$	R_{min}	R_{nm}	R_{col}	γ_{nm}	γ_{col}
MOSS scenario	10	30	200	75	30	0.25	0.75
KIRKEØY scenario	10	30	75	40	20	0.25	0.75
Unit	[deg]	[deg]	[m]	[m]	[m]	[-]	[-]

Table 4.20: Summary of the parameters needed to calculate COLREGs-specific performance metrics for the complex high-speed simulations using the scaled Otter USV model.

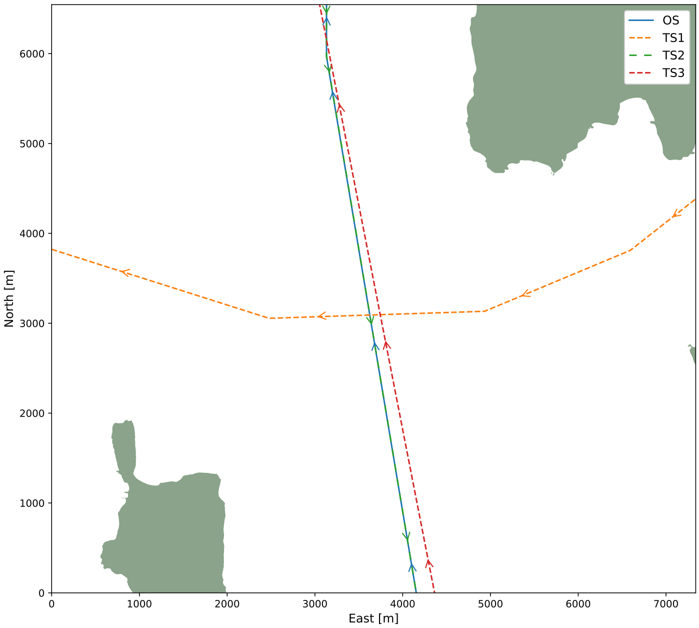


Figure 4.38: Overview of the MOSS scenario. The arrows indicate the directions of the respective vessels. The Bastøy island can be seen to the left, and the Jeløy island can be seen to the right.

MOSS scenario

The MOSS scenario could have been called MOSS or HORTEN, as the scenario is placed between Moss and Horten in Norway. The scenario is based on the second scenario described in Section 2.1.1, i.e. the scenario visualized in Figure 2.4. The scenario consists of three target ships, where the target ships are set up in the following way:

- TS1 is based on the BASTØ IV ferry that transfers cars and people between Moss and Horten in Norway [1]. The trajectory of the ferry is based on trajectories that are seen on the maps from the Norwegian website Gule Sider [2]. The speed of the ferry is set to 6 m/s, and is based on data from MarineTraffic.com [3]. TS1 poses a give-way crossing encounter for the ownship.
- TS2 is based on the COLOR FANTASY ferry that transfers cars and people from

[1] BASTØ IV passenger ferry: https://www.marinetraffic.com/en/ais/details/ships/shipid:4492086/mmsi:257845600/imo:9771420/vessel:BASTO_IV (Accessed 27/05/2023)

[2] Gule Sider map: https://kart.gulesider.no/?c=59.384895,10.587044&z=12 (Accessed 27/05/2023)

[3] MarineTraffic.com: https://www.marinetraffic.com/en/ais/home/shipid:4492086/zoom:14 (Accessed 27/05/2023)

Oslo in Norway to Denmark [4]. The trajectory of this ferry is also found based on the map available at Gule Sider, which shows the trajectory of the ferry. The speed of the ferry is set to 9 m/s based on data from MarineTraffic.com, and the target ship poses an head-on encounter for the ownship.

- TS3 is intended to be a leisure vessel that is overtaking the ownship, and therefore has a speed of 12 m/s.

The initial waypoints to be followed by the ownship is not calculated using a high-level planner for this scenario, but are set up manually to ensure the scenario poses the intended challenges. For the MOSS scenario, the ownship is set to follow the same trajectory as the COLOR FANTASY ferry (TS2), but in the opposite direction at a speed of 10 m/s. The scenario is summarized in Figure 4.38.

[4] COLOR FANTASY ferry: https://www.marinetraffic.com/en/ais/details/ships/shipid:304932/mmsi:257182000/imo:9278234/vessel:COLOR_FANTASY (Accessed 27/05/2023)

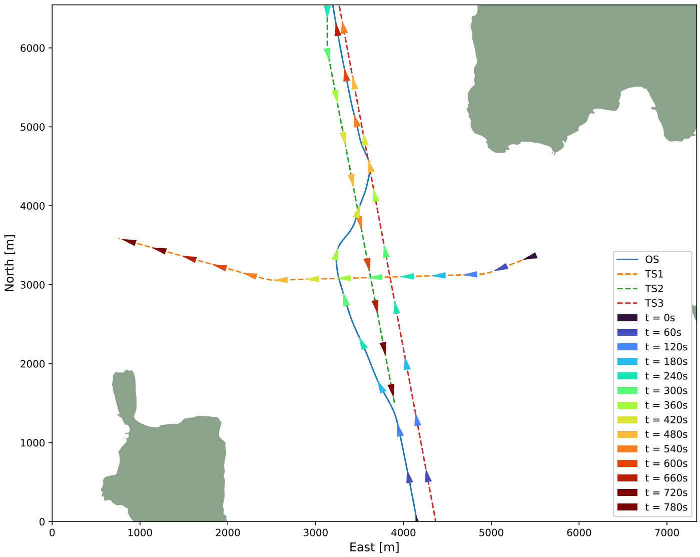


Figure 4.39: Trajectories for the the MOSS scenario. The colors indicate the timestamps throughout the simulation, and the arrows indicate the heading angle of the respective vessels involved.

The trajectories resulting from simulating the MOSS scenario are shown in Figure 4.39. It can be seen that the ownship starts a maneuver to avoid collision with the ferry TS1 quite early. The collision between the ownship and TS1 would happen somewhere between $t = 240$s and $t = 300$s, and the maneuver is initiated between $t = 120$s and $t = 180$s, which is fairly early. In order to comply with Rule 15 regarding crossing situations, the trajectory planner should have planned a trajectory that passed abaft TS1. There is however a vessel overtaking the ownship, namely TS3. This target ship probably makes it hard for the ownship to plan a path abaft TS1, as it leads to a situation with somewhat conflicting rules, as discussed in Section 2.1.1. The trajectory planner plans a trajectory that crosses in front of TS1, which it should avoid according to Rule 15. This again comes down to the question of what distance it is allowed to cross in front of a target ship. In practice, the ownship has a speed advantage to TS1, but this is likely not a good excuse for crossing in front of it.

One thing that is different with the MOSS scenario than the scenario visualized in Figure 2.4 is that the MOSS scenario does not include a vessel that comes up behind the ownship, forcing the ownship to keep its speed. It is however seen from Figure 4.39 that the trajectory planner does not try to slacken the speed of the ownship in order to avoid collision with TS1, even if there is nothing keeping it from doing so.

After having crossed in front of TS1, the trajectory planner plans a trajectory such that the ownship passes TS2 on the target ship's port side, which is in compliance with Rule 14. At this point of the scenario, the ownship has also been overtaken by TS3. During the overtaking, the ownship can be said to have kept its obligations w.r.t. Rule 13, as it has never made itself a crossing vessel for TS3. Overall the trajectory planner is seen to plan

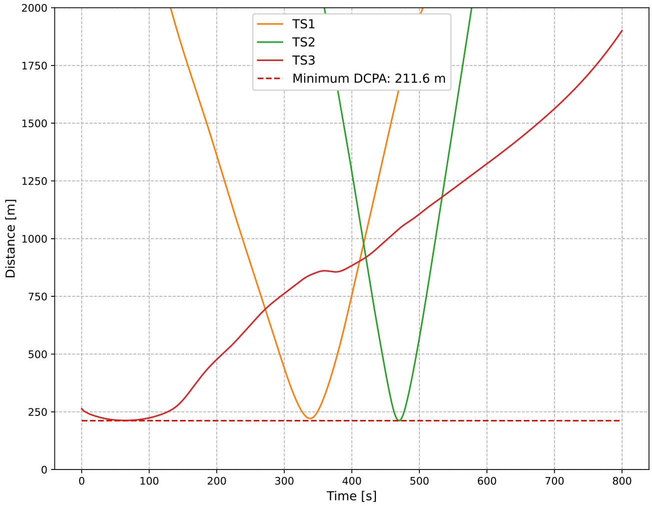


Figure 4.40: Distance between the ownship and target ships in the MOSS scenario.

trajectories that avoids collision with all target ships, while adhering to the COLREGs, at least w.r.t. TS2 and TS3. For the TS1 encounter, the maneuver is more questionable, but might be a result of conflicting rules inflicted by TS1 and TS3.

The distances kept between the ownship and the three target ships during the simulation are shown in Figure 4.40. As seen in this figure, the ownship keeps a distance of $>$ 211m from all target ships. This also means that the ownship crosses at least 211m in front of TS2, which is $\approx$ 35 seconds in front of TS1 as it has a speed of 6 m/s. The fact that the ownship keeps a distance of over 211m when crossing in front of TS1 is however a result of choosing $l = 200$ when defining the TS domain. This means that even if 211m is an acceptable distance to cross in front of TS1 (which it might not be), this distance would likely be lower if l was chosen differently. If l was set to 100m, and the ownship still crossed in front of TS1, this would mean it crosses $\approx$ 16s in front of TS1, which is certainly a violation of both Rule 8 and Rule 15 of the COLREGs. Using a tool like the port-side constraint in order to ensure the ownship passes abaft TS1 could be useful in this encounter, as seen in the give-way crossing batch simulations in Figure 4.26. The port-side constraint would however need some more work, as the way it is described in Section 3.8.3 does not work well in the presence of multiple target ships. Instead of defining the port-side constraint with respect to the second window of reduced cost, it should have been defined w.r.t. each individual target ship's t_{crit}^{enter} and t_{crit}^{exit}. It must also once again be underlined that planning a path abaft TS1 would require letting TS3 overtake the ownship before performing the maneuver, and it seems like the trajectory planner is not eager to slacken the speed of the ownship in order to resolve encounters like these. This could be different with a different set of parameters than those shown in Table 4.19, as the parameters used in this simulation penalizes acceleration more than turning.

The COLREGs-specific metrics are harder to calculate in this scenario compared to the complex low-speed simulations using the milliAmpere1. The $\mathcal{P}_{safety}$ score is trivial to calculate, but the r_{det} parameter used to calculate $\mathcal{P}_{delay}$ and $\mathcal{P}'_{\Delta_{\chi_{app}}}$ is not trivial to determine. The reason for this is that in the MOSS scenario, all target ships are detected from the start ($t = 0$s in Figure 4.39). For TS1 it is possible to calculate all metrics, and use the distance at $t = 0$s as a basis to calculate the metrics. For TS2 it is however harder, as it doesn't make much sense to calculate this score based on when TS2 is detected, because the first maneuver is not a maneuver that aims to avoid collision with TS2. It is however in this scenario proposed to use the distance between the ownship and TS2 at $t = 330$s, which is the point where the ownship crosses in front of TS1. Using this as a basis of calculating the metrics, as well as the parameters given in Table 4.20, results in the scores given in Table 4.21. Only the safety score is calculated for TS3, as the ownship doesn't have a give-way role w.r.t. to this target ship.

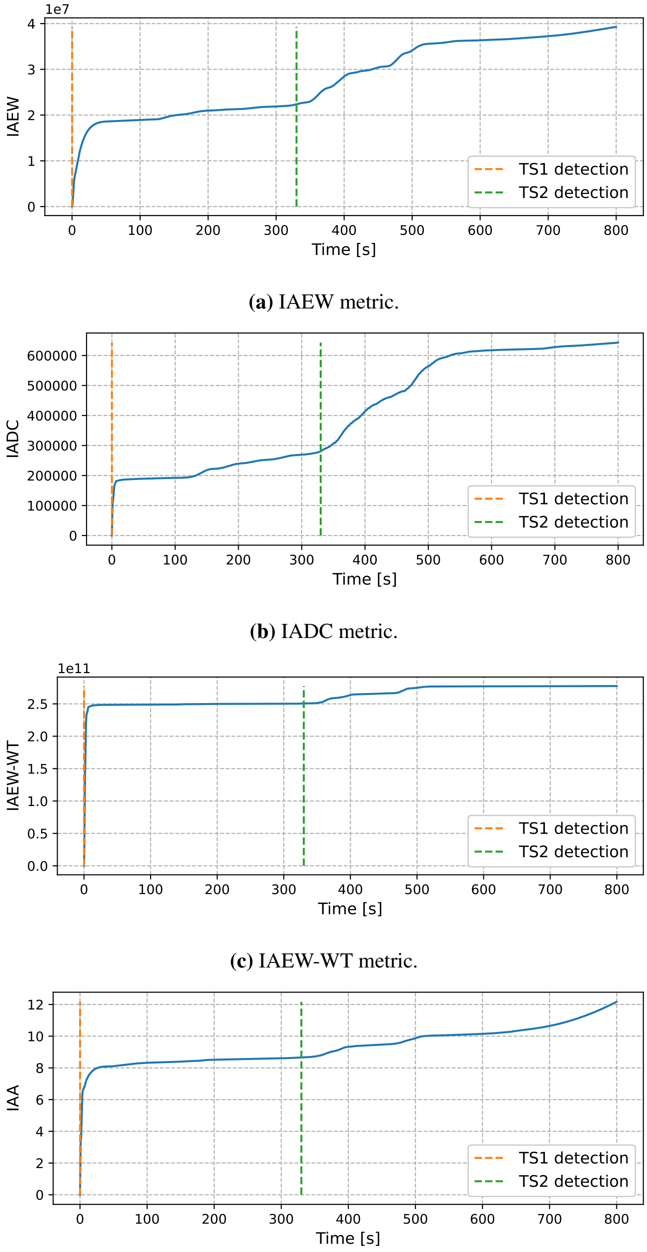
As seen from Table 4.20, the maneuver made to avoid collision with TS1 is not low or high. The scores indicate that the maneuver made w.r.t. TS1 is not made early, nor late when using a threshold of 10° to decide if a maneuver is initiated. The $\mathcal{P}'_{\Delta_{\chi_{app}}}$-score is in the higher end, even when using a threshold of 30° as what can be deemed an apparent maneuver. These scores are expected based on Figure 4.39, and are also expected from the high-speed trajectory planner that aims to plan trajectories with less turning involved than the low-speed planner. This ends up as a trade-off between getting good scores, and planning trajectories that are feasible to follow for an underactuated, high-speed vessel, and it is not trivial to decide what should be weighted higher of the two. A minimum should of course be that the vessel is able to follow the trajectory, but it might have been able to even with more turning than shown in Figure 4.39.

The IAEW-, IADC-, IAEW-WT- and IAA metrics from simulating the MOSS scenario are shown in Figure 4.41, where the times of detecting TS1 and TS2 are indicated. Also here, $t = 330$s is used as a time of detecting TS2, although this is not entirely correct as discussed above. As seen in the plots, all metrics increase in the beginning, which likely coincides with the start of the first window of reduced cost activated by the detection of TS1. It can also be seen on the IAEW- and IADC metric plots that these increase when the maneuver to avoid collision with TS2 starts just after the ownship has crossed in front of TS1.

The runtimes of the high-speed trajectory planner from simulating the MOSS scenario are shown in Figure 4.42. What is surprising with this figure is that the runtime of the high-speed trajectory planner is actually lower than the runtime of the low-speed planner when simulating the BRATTØRA2 scenario, even though there are three target ships in both scenarios. One reason can however be that the MOSS scenario involves significantly less static obstacles than the BRATTØRA2 scenario, which makes it reasonable to assume the runtime will be lower for setting up and solving the OCP.

Metric	**TS1**	**TS2**	**TS3**
$\mathcal{P}_{delay}$	0.48	0.17	-
$\mathcal{P}'_{\Delta_{\chi_{app}}}$	0.57	0	-
$\mathcal{P}_{safety}$	0	0	0

Table 4.21: COLREGs-specific performance metric scores resulting from simulating the MOSS scenario.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.41: IAEW, IADC, IAEW-WT and IAA scores resulting from the MOSS scenario. The times where TS1 and TS2 are detected are indicated.

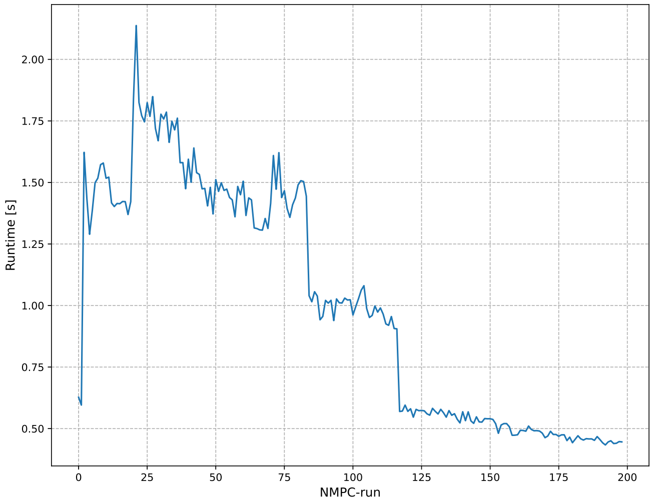


Figure 4.42: Trajectory planner runtime when simulating the MOSS scenario.

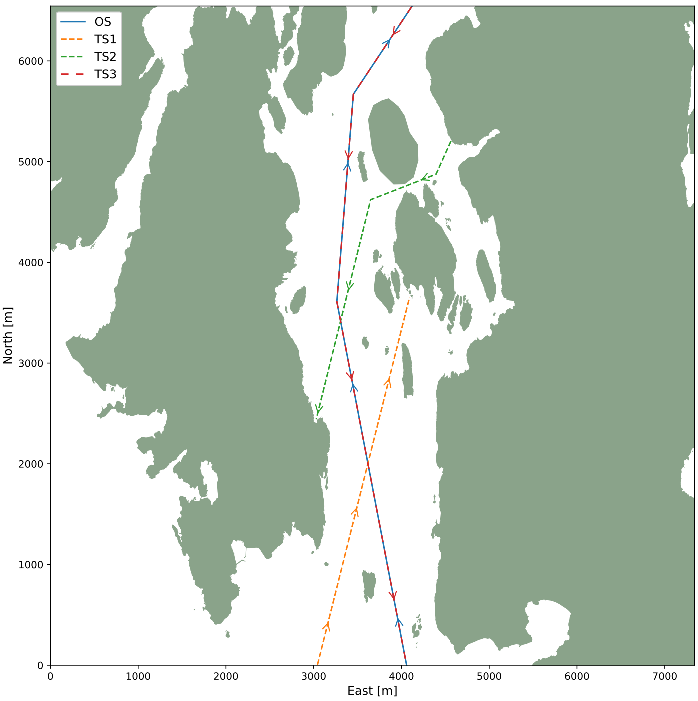


Figure 4.43: Overview of the KIRKEØY scenario. The arrows indicate the directions of the respective vessels. The KIRKEØY island can be seen as the large island to the right.

KIRKEØY scenario

The KIRKEØY scenario is located between the Kirkeøy- and Asmaløy islands in Norway, south-east of the MOSS scenario. The scenario involves three target ships, where the target ships are set up as follows:

- TS1 is thought to be a leisure vessel that imposes a stand-on crossing encounter for the ownship. The target ship is set up such that it shall pass just abaft the ownship, and the ownship shall therefore not perform any maneuver to avoid collision with TS1. This target ship is set up to have a constant speed of 6 m/s.
- TS2 is also thought to be a leisure vessel, but this vessel ends up imposing a head-on encounter for the ownship. This target ship is however set up such that it is not in line-of-sight of the ownship until some time into the simulation, and therefore challenges the trajectory planner by representing a somewhat late detection. The intention of this target ship should also not be clear until it passes its second way-

point, in order to challenge the trajectory planner based on the way it predicts the trajectory of target ships. This target ship also has a constant speed of 6 m/s.

- TS3 is based on a ferry route from Strømstad to Fredrikstad in Norway, that can be seen on OpenStreetMap. TS3 therefore follows this ferry route, and is assumed to have a speed of 7 m/s throughout its transit in the absence of relevant data from MarineTraffic.com. This vessel imposes a head-on encounter for the ownship, and is also detected after a time has passed, as it is not in line-of-sight of the ownship to begin with.

The ownship is in this scenario defined such that it follows the same ferry route as TS3, but in the opposite direction at 10 m/s. This means the waypoints the ownship is supposed to follow are set up manually in this scenario, similarly to what was done in the MOSS scenario. The scenario is summarized in Figure 4.43.

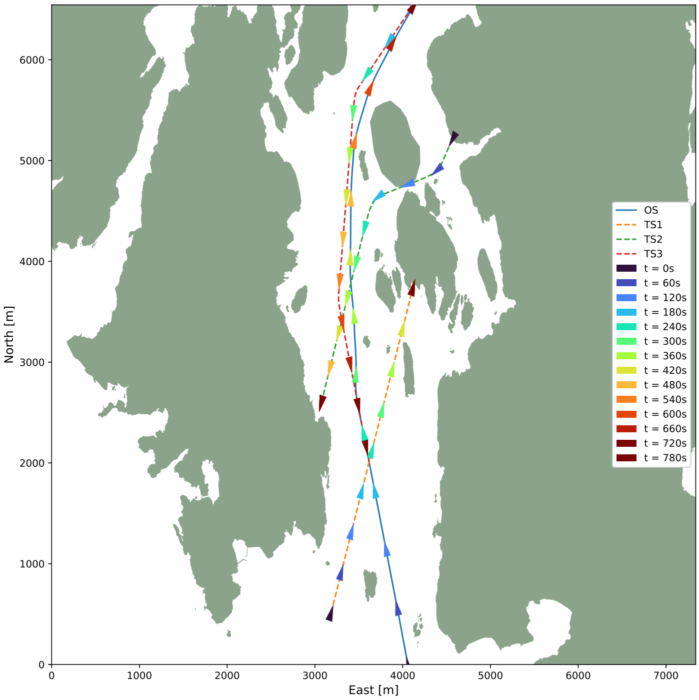


Figure 4.44: Resulting trajectories from simulations performed on the KIRKEØY scenario. The colors indicate the timestamps throughout the simulation, and the arrows indicate the heading angle of the respective vessels involved.

The trajectories resulting from simulating the KIRKEØY scenario are shown in Figure 4.44. It can be seen that the ownship starts by keeping its stand-on obligation w.r.t. TS1, as it is not seen to perform any maneuvers to avoid collision with the target ship before the target ship crosses abaft the ownship. This planning is in compliance with Rule 17. It can then be seen that the ownship starts a maneuver to avoid collision with TS2 between $t = 240\text{s}$ and $t = 300\text{s}$. This is not a large maneuver, but it is likely enough for the ownship to show its red sidelight, signalling to TS2 that it intends to maneuver towards starboard. This is in line with Rule 14, and to some degree in line with Rule 8. The reason it is hard to conclude that it is in line with Rule 8 is that the maneuver might not be as apparent as one would wish in this encounter.

After the ownship has met TS2 port-to-port, it is not seen to make any maneuvers in order to increase its distance to TS3. It is seen that it mostly just keeps its course, before also passing TS3 port-to-port. Both the encounter with TS2 and TS3 are therefore in compliance with Rule 14. Whether the ownship should perform a maneuver w.r.t. TS3 is arguable, but it does at least not make any maneuvers that could lead to confusion.

It can also be seen that the trajectory of the ownship is overall very suitable for a high-speed, underactuated vessel, and one could argue the resulting trajectory involves too little turning. It could e.g. be beneficial to make a more apparent maneuver w.r.t. TS2 and TS3.

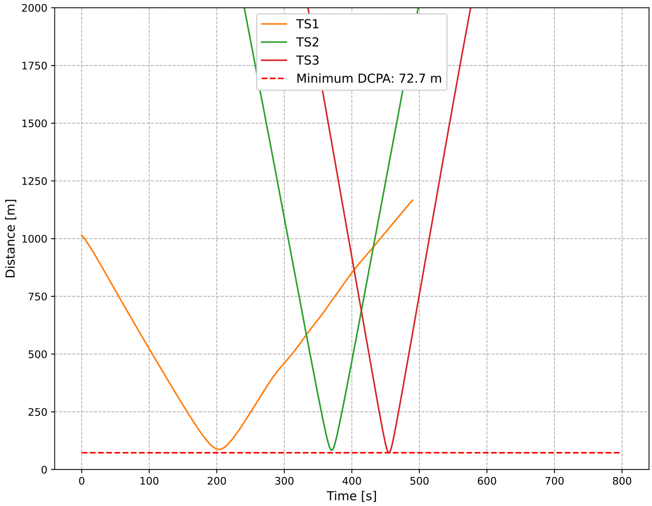


Figure 4.45: Distance between the ownship and target ships in the KIRKEØY scenario.

The distances between the ownship and the three target ships are shown in Figure 4.45. It can be seen that the minimum DCPA is 72.7m, and that this DPCA is between the ownship and TS3, which means the ownship has entered its TS domain. This could be a result of the trajectory planner itself, but it could also be a result of the Otter USV model not following the intended trajectory as accurately as it should. The fact that there is a lot of room on the east side of TS3, and that there is no late detection involved, makes it hard to believe the trajectory planner has not been able to find a feasible solution that passes $>$ 75m from TS3. It is also confirmed from data obtained from the simulation that the DCPA between the ownship and TS1 and TS2 is 87m and 83m, respectively.

The COLREGs-specific performance metrics calculated based on the KIRKEØY scenario simulation are shown in Table 4.22. The $\mathcal{P}_{delay}$- and $\mathcal{P}'_{\Delta_{\chi app}}$-scores are not calculated for TS1, as this is not a give-way encounter for the ownship. For TS2 and TS3, the detection times are obtained from simulation data at $t = 191$s and $t = 264$s, respectively. As seen from the table, all scores except from the safety scores are in the higher end of what one would prefer. For the head-on encounter with TS2, it is seen that both scores are above 0.5, which indicate the action is made relatively late, and is not very apparent

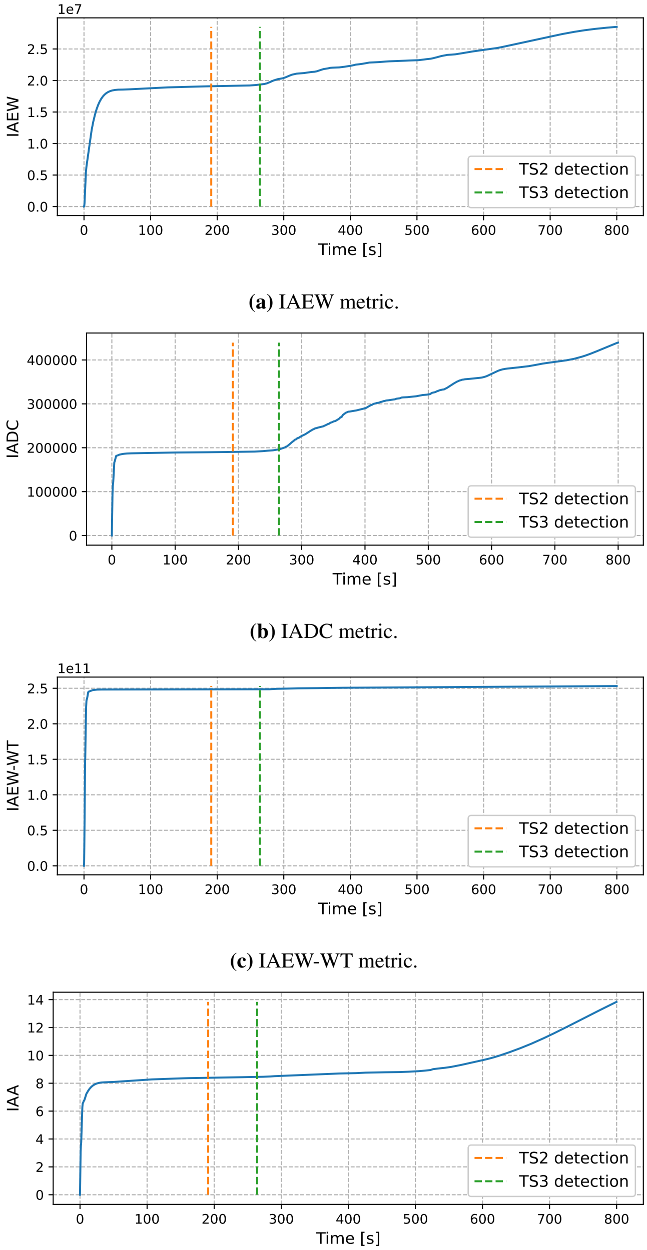
Metric	TS1	TS2	TS3
$\mathcal{P}_{delay}$	-	0.52	0.89
$\mathcal{P}'_{\Delta_{\chi app}}$	-	0.63	0.99
$\mathcal{P}_{safety}$	0	0	0.016

Table 4.22: COLREGs-specific performance metric scores resulting from simulating the KIRKEØY scenario.

based on the simulation data. For TS3, the scores are as expected based on Figure 4.44 almost 1, meaning the action is not readily apparent, and not made in ample time. This is as expected, as the ownship is not seen to make any specific maneuver to avoid collision with TS3, and the maneuver made is probably first and foremost made to avoid collision with TS2.

The IAEW, IADC, IAEW-WT and IAA metrics throughout the simulation are shown in Figure 4.46, where the detection of TS2 and TS3 are indicated. From these figures it is clear that there is not much happening in any of the metrics until after the ownship has detected TS3. It is at this point in the trajectory the ownship starts turning in order to avoid collision with TS2, before passing TS3 and completing the transit. This is seen in the figures, as both the IAEW and IADC scores increase just before $t = 300$s, which indicates an increased control error, power consumption and actuator usage in this period. This is as expected, as this is really the only part of the trajectory where a maneuver is made to avoid collision. The fact that the metrics keeps increasing after having passed TS3 might be an indication of the last part of the trajectory involving a long turn that involves more turning than earlier in the transit, although the turning made by the ownship during the transit is overall not much.

The runtimes of the trajectory planner when simulating the KIRKEØY scenario are shown in Figure 4.47. From this figure it is clear that the trajectory planner uses more time setting up and solving the OCP compared to what it did in the MOSS scenario. This is as expected, as the KIRKEØY scenario involves the same amount of target ships as the MOSS scenario, but has significantly more land-obstacles to consider when planning. It can also be seen that the high-speed trajectory planner uses more time setting up and solving the OCP than the low-speed trajectory planner for the BRATTØRA and BRATTØRA2 scenarios. This is also as expected, as the high-speed trajectory planner uses the nonlinear model (3.28) instead of the linear model (3.22) used by the low-speed planner.



(a) IAEW metric.

(b) IADC metric.

(c) IAEW-WT metric.

(d) IAA metric.

Figure 4.46: IAEW, IADC, IAEW-WT and IAA scores resulting from the MOSS scenario. The times where TS2 and TS3 are detected are indicated.

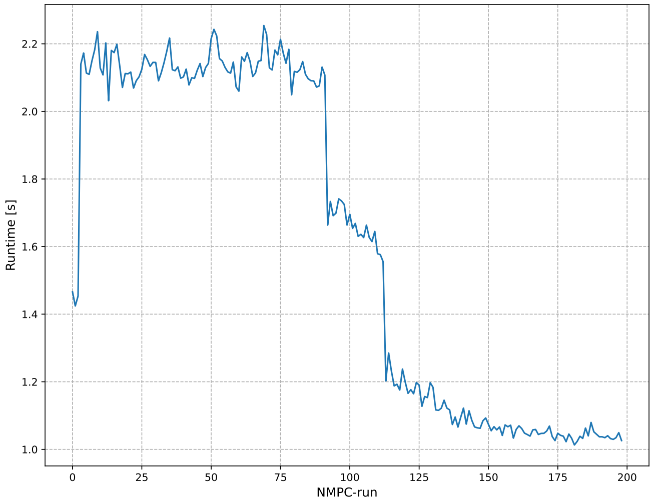


Figure 4.47: Trajectory planner runtime when simulating the KIRKEØY scenario.

4.2.4 Discussion

As seen from the results from the high-speed simulations in this section, the high-speed trajectory planner results in trajectories that are partly COLREGs-compliant, and that is better suited for an underactuated vessel to follow in higher speeds. The batch simulations show trajectories that are similar to the ones obtained by the low-speed planner in Section 4.1.1. The simulated head-on encounters show similar trajectories to those obtained by the low-speed planer, but that involves less turning by the ownship overall. The trajectory planner plans maneuvers early in the encounter, and avoids collision while keeping the desired minimum distance to the target ship, and the trajectories can be said to be in compliance with Rule 14 and Rule 16 of the COLREGs.

The give-way crossing encounters batch-simulated in this section also shows a behaviour that might actually be more in line with Rule 15 and Rule 16 than those obtained in Section 4.1.1. The reasoning behind this statement is that the high-speed planner to a larger degree plans trajectories that passes abaft the target ship, where the low-speed planner planned paths that arguably violated Rule 15 by crossing ahead of the target ship. Also in the give-way crossing scenarios, the ownship keeps well above the desired distance from the target ship.

The overtaking batch simulations shows a very similar behaviour to what was obtained by the low-speed planner, the difference being that the maneuver stretches over a longer area by the high-speed planner. This is reasonable, as the speed is significantly higher. It is however still possible to argue some of the trajectories are not completely in line with Rule 13 regarding overtaking encounters. This is because some of the trajectories can be said to make the ownship (or the target ship) a crossing vessel, which is not permitted by Rule 13. It is however difficult to say when it is allowed for the ownship to finish its overtaking and maneuver in front of the target ship in order to continue its route. The overall results from the batch-simulated overtaking scenarios can however be said to be in compliance with Rule 13, as the ownship always gives way to the target ship by overtaking it on either side.

The stand-on encounters shown in the batch simulations can be said to both comply with Rule 17 and Rule 2 of the COLREGs. This is because the ownship keeps its course and speed until it is clear the target ship will not give way. This makes the ownship violate its stand-on obligations, in order to avoid collision. This is in line with Rule 2 as well as with Rule 17, as Rule 17 allows the ownship to violate its stand-on obligations if there is no other way to avoid collision.

Introducing windows of reduced cost is seen to have the same benefits for the high-speed planner as for the low-speed planner, especially when it comes to initiating a maneuver early in compliance with Rule 8. One difference between the high-speed planner and the low-speed planner is however that introducing windows of reduced cost does seemingly not lead to increased power-consumption, actuator usage or accelerations during the transit. This is based on the IAEW, IADC, IAEW-WT and IAA scores actually being higher for the simulations performed without using windows of reduced cost. This might be due to the low-speed trajectory planner using accelerations to a larger degree when avoiding collision, while the high-speed trajectory planner is incentivized to avoid collision first and foremost through turning. It can however be argued that the high-speed planner plans trajectories that involve too little turning based on the COLREGs-specific metrics.

The port-side constraint does not have much effect on the high-speed trajectory planner, as is also expected. This constraint is not added as early in the planning horizon for the high-speed trajectory planner, which means less action is demanded of the planner during the first window of reduced cost. Similarly to the low-speed planner, the port-side constraint was not used in the complex simulations, due to it not being well suited for encounters including multiple target ships. This could be fixed by defining the port-side constraint based on each individual target ship's t_{crit}^{enter} and t_{crit}^{exit}, and might be worth looking into as the constraint is still seen to directly affect the COLREGs-specific scores measuring early and apparent action.

As mentioned, the high-speed trajectory planner seems to generate trajectories that arguably involves too little turning. The trajectory planner is designed to be used by underactuated, high-speed vessels, but such vessels are likely to handle more turning than what is shown in the results from the complex scenario simulations. This ends up in a trade-off between COLREGs-compliance and feasibility for underactuated vessels. It is also possible to argue that in open-sea environments covering larger areas, there doesn't need to be as much turning by the ownship. One little turn might be enough to show the ownship's red sidelight in for example head-on encounters, indicating its intention for other vessels. One usage of the high-speed trajectory planner can therefore be as part of a larger, hybrid system for trajectory planning. In such a system, the low-speed trajectory planner could be used in parts of the transit that involve narrow environments like canals, before switching to the high-speed planner in parts of the trajectory that are out on the open sea. Implementing such a hybrid system should be looked into, but switching between trajectory planners is not trivial, and might lead to unwanted effects during the actual switch.

To decide if either of the trajectory planners used in this work are COLREGs-compliant, at least with a subset of the rules, is not trivial. Based on the discussions regarding Rule 2 in Section 2.1.1, a trajectory planner can be argued to be COLREGs-compliant as long as it avoids collision, because Rule 2 allows for a departure from the rules in order to avoid collision. All results shown in this work indicate that the trajectory planner is able to plan trajectories that avoids collisions with other vessels. Saying that a trajectory planner is COLREGs-compliant based on it avoiding collisions, and therefore complying with Rule 2, is however a far stretch, and would probably not fare well in a court if something was to happen. Even if the trajectory planner avoids collisions between the ownship and other target ships, collision might still happen. If the ownship makes a maneuver that is not expected, this can lead to confusion for the target ship, and might lead to the target ship colliding in some way due to the confusion. In a court, this might cause problems for the autonomous vessel. With this in mind, it is possible to argue that Rule 2 of the COLREGs is not a rule that applies to autonomous vessels as much as manned vessels. If a manned target ship observes an unexpected maneuver made by an autonomous vessel, knowing the vessel is autonomous, this will probably lead to more confusion than if the same maneuver was made by another manned vessel. If a manned vessel made the same maneuver, the target ship could to a larger degree rely on the maneuver being made based on for example the practice of good seamanship. It will require more for a person to rely on an autonomous vessel maneuvering in compliance with good seamanship, as autonomous vessels are still in most peoples eyes relatively new and unknown. To conclude if an autonomous vessel is COLREGs-compliant is therefore hard, and COLREGs-aware might be a more suited term at this point in time and with the current formulation of the COLREGs.

Chapter 5

Conclusions and further work

Conclusions

In this work, two COLREGs-aware MPC-based mid-level trajectory planners that considers static and dynamic obstacles are presented, where one of the trajectory planners are intended for fully actuated, low-speed passenger ferries and the other is intended for underactuated, high-speed vessels. Through an extensive simulation study, both trajectory planners are shown to avoid collision with both static and dynamic obstacles, in a way that partly complies with a subset of the COLREGs.

Batch simulations are performed to assess the overall ability of the trajectory planners to plan trajectories that avoid collisions in single-vessel encounters, in ways that comply with a subset of the COLREGs. The subset considered in this work are the encounter-specific rules, i.e. Rule 8 and Rule 13 - Rule 17. The batch simulations show an overall COLREGs-compliant trajectory planning w.r.t. these rules, that keeps the intended minimum distance to the target ship.

Two tools for MPC-based trajectory planning are presented as part of the work. The first tool is introducing windows of reduced cost, and the second is a simple linear constraint placed on the port side of target ships, referred to as the port-side constraint. Both tools are assessed based on COLREGs-specific performance metrics, as well as performance metrics intended to measure the control error of the motion control system onboard the ownship, the power consumed by the ownship, the actuator wear-and-tear and the absolute acceleration performed by the ownship. The windows of reduced cost are through COLREGs-specific metrics shown to incentivize both trajectory planners to plan early action, as well as action that is more apparent for other vessels, in compliance with Rule 8 and Rule 16 of the COLREGs. This is however shown to come at the cost of higher energy consumption, actuator usage and absolute acceleration when using the trajectory planner intended for low-speed passenger ferries. For the trajectory planner intended for high-speed, underactuated vessels, the windows of reduced cost are not seen to increase these factors.

The port-side constraint is also shown to effectively increase the compliance with Rule

8 and Rule 16 in both planners, but this constraint is also seen to cause problems, indicating this tool at least needs further work to be utilized fully. Both tools are either way seen to effectively affect how well both trajectory planners comply with Rule 8 and Rule 16, and are shown as tools that can be added to a larger trajectory-planning toolbox when working with MPC-based planners.

Two complex scenarios are simulated for each trajectory planner, each within the intended operational context of the respective planner. From these simulations, the trajectory planners are shown to comply with the chosen subset of the COLREGs to a large degree. The simulations involve violations of the COLREGs, which indicates further work should be done, especially when it comes to planning the correct side to pass a target ship. On the other hand, all simulations show that the trajectory planners are able to avoid collision with all static and dynamic obstacles, while also being able to avoid collision by departing from a subset of the rules if necessary, in compliance with Rule 2 of the COLREGs.

Further work

The following list summarizes the further work proposed based on this work:

- Real-life experiments should be performed on the experimental platforms milliAmpere1 or milliAmpere2 in order to test the methods in a real-life environment, using a real situational awareness system. This will help see how robust the methods in this work are to external disturbances, as well as a situational awareness system that does not provide perfect information at all times.

- More work should be done on intent prediction of the target ship. The intent of the target ship is in this work based on the assumption that a target ship will keep its measured course and speed. This is not a very good assumption in a real-world environment, but predicting the intent of a target ship is not trivial. Examples could be assuming the target ship will keep a constant distance to the edge of a canal in canal environments, or at least a constant ratio between the center line of the canal and the edge of the canal. This method will require taking a look at traffic separation schemes (TSS) at sea, and including these in the intent estimate.

- Runtime improvements should be made on both the low-speed- and high-speed trajectory planners. Runtime improvements can be obtained by different means, and one example is by utilizing metamodeling on the nonlinear model used by the high-speed trajectory planner. In short, metamodeling is a method for creating models from models [46]. Using statistical methods, metamodeling can be used to generate multiple simplified (linear) models from a nonlinear model.

  Another way of improving the runtime of code written in Python is a tool called Codon, which is a high-performance Python compiler said to improve the runtime of Python code with a speedup in the magnitude of 10-100x or more [1]. This speedup magnitude is stated by the creators of Codon, but it is likely that Codon could be utilized in order to at least improve the runtime of parts of this work significantly.

[1] Codon Github repository: https://github.com/exaloop/codon (Accessed 23/05/2023)

- The low-speed- and high-speed trajectory planners presented in this work can as discussed be seen as parts of a hybrid trajectory planning system, where e.g. the low-speed trajectory planner can be used in canal environments, and the high-speed trajectory planner can be used on open sea during the same transit. Switching between the two planners by switching models, objective functions and shooting constraints should be tested in future work, both through simulations and physical experiments involving both canal and open sea environments. In this work, theory from the field of hybrid systems should be considered. One example of such hybrid collision avoidance systems is presented in [47].

- There is a lot of static parameters that needs to be set in order to run the trajectory planners presented in this work, and it is hard finding an 'optimal' way to set all parameters without rigorous simulation studies. One way of setting the static parameters could however be obtained through utilizing machine learning techniques, or more specifically Reinforcement Learning (RL). In this way, simulations could possibly be run, where the RL agent is trained to find an optimal parameter combination based on both COLREGs-specific performance metrics, as well as performance metrics related to the motion control system.

  Another way of handling this problem is by setting some of the parameters in a more dynamic manner, based on information about the environment and sizes of the vessels involved. The l-parameter used to define the TS domain can e.g. be calculated based on the size of the target ship, as done in [18].

  Especially the high-speed trajectory planner proposed in this work is through simulations seen to be somewhat sensitive to different parameter setting. An underactuated, faster vessel is more dependent on the resulting trajectory being feasible to follow compared to a fully actuated, low-speed vessel. Because of this, it is especially important to consider finding another way of setting some of the parameters used by the high-speed trajectory planner.

- More work should generally be done on trajectory planning for high-speed, underactuated vessels. The methods and results from this work can be used as a starting point, but further work is needed. First, a vessel model that more accurately describes a high-speed vessel should be utilized. It is also seen that the high-speed trajectory planner in this work struggles to choose the correct side to pass the target ship in give-way crossing encounters, which should be considered in future work. One way of enforcing the correct side of the target ship being chosen as the passing side could e.g. be to introduce similar constraints as the port-side constraint, but this should be done with care, as the port-side constraint is seen to pose challenges for the high-speed trajectory planner in this work.

- More work should be done on utilizing and possibly developing more performance metrics to assess COLREGs-compliance. One example could be to combine COLREGs-specific performance metrics in the same way IAEW and IADC is combined into IAEW-WT.

Bibliography

[1] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2021.

[2] T. I. Fossen and T. Perez, "Marine systems simulator (MSS)." Available at: https://github.com/cybergalactic/MSS, 2004.

[3] I. Haugan, "NTNU trials world's first urban autonomous passenger ferry," *Norwegian SciTech News*, 2022. Available at: https://norwegianscitechnews.com/2022/09/ntnu-trials-worlds-first-urban-autonomous-passenger-ferry/ (Accessed 27/09/2022).

[4] E. H. Thyri, *COLREGs-aware Trajectory Planning and Collision Avoidance for Autonomous Surface Vessels*. PhD thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2022.

[5] H.-T. L. Chiang and L. Tapia, "COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2024–2031, 2018.

[6] F. Deng, L. Jin, X. Hou, L. Wang, B. Li, and H. Yang, "COLREGs: Compliant Dynamic Obstacle Avoidance of USVs Based on the Dynamic Navigation Ship Domain," *Journal of Marine Science and Engineering*, vol. 9, no. 8, 2021.

[7] A. Tsolakis, D. Benders, O. de Groot, R. Negenborn, V. Reppa, and L. Ferranti, "COLREGs-aware Trajectory Optimization for Autonomous Surface Vessels," in *Proceedings of the 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2022*, vol. 55, (Kongens Lyngby, Denmark), pp. 269–274, 2016.

[8] H. Lyu and Y. Yin, "Fast path planning for autonomous ships in restricted waters," *Applied sciences*, vol. 8, no. 12, p. 2592, 2018.

[9] Queensland Government: Maritime Safety Queensland, "Navigation Lights." Available at: https://www.msq.qld.gov.au/safety/navigation-lights. (Accessed 31/03/2023).

[10] R. Jassal, "COLREGs Rule 2: Here is the Simple Explanation to the Most Confusing Rule." Available at: https://www.myseatime.com/blog/detail/colregs-rule-2. (Accessed: 06/05/2023).

[11] Ø. Engelhardtsen, "COLREG-compliance - a tough nut to crack," 2023. Part of SFI AutoShip Webinar held on 22/03/2023 regarding COLREGs in the context of autonomous COLAV.

[12] SBG Systems, "Reference coordinate frames." Available at: https://support.sbg-systems.com/sc/kb/latest/underlying-maths-conventions/reference-coordinate-frames. (Accessed 05/10/2022).

[13] T. Perez, *Ship Motion Control*. Springer London, 2005.

[14] A.-L. Aakervik, "Førerløs over kanalen." Available at: https://ntnudiscovery.no/forerlos-over-kanalen/, December 2018. (Accessed: 15/04/2023).

[15] Maritime Robotics, "THE OTTER." Available at: https://www.maritimerobotics.com/otter. (Accessed: 07/02/2023).

[16] M. E. N. Sørensen, *Topics in Nonlinear and Model-based Control of Ships*. PhD thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2021.

[17] E. Thyri and M. Breivik, "A domain-based and reactive COLAV method with a partially COLREGs-compliant domain for ASVs operating in confined waters," *Field Robotics*, vol. 2, p. 637–677, 05 2022.

[18] E. H. Thyri and M. Breivik, "Collision avoidance for ASVs through trajectory planning: MPC with COLREGs-compliant nonlinear constraints," *Modeling, identification and control*, vol. 43, no. 2, pp. 55–77, 2022.

[19] B. Foss and T. A. N. Heirung, "Merging Optimization and Control," tech. rep., Norwegian University of Science and Technology (NTNU), 2016.

[20] A. Steinsmo, "*High-level Path Planner for Autonomous Crew Transfer Vessels*." Specialization project thesis. Norwegian University of Science and Technology (NTNU). Trondheim, Norway, 2022.

[21] Brødrene Aa, "Brødrene Aa is building autonomous ferry." Available at: https://www.braa.no/news/brodrene-aa-is-building-the-worlds-first-driverless-ferry. (Accessed 14/04/2023).

[22] Maritime and Coastgurd Agency, "MGN 324 (M+F) Amendment 2." Available at: https://www.gov.uk/government/publications/mgn-324-mf-amendment-2-navigation-watchkeeping-safety-use-of-very-high-frequency-vhf-radio-and-automatic-identification-system-ais/mgn-324-mf-amendment-2#use-of-vhf-to-aid-collision-avoidance. (Accessed 14/04/2023).

[23] NTNU, "The NTNU digital transformation initiative." Available at: https://www.ntnu.edu/digital-transformation. (Accessed 25/10/2022).

[24] NTNU, “Autoferry - autonomous all-electric passenger ferries for urban water transport.” Available at: https://www.ntnu.edu/autoferry. (Accessed 25/10/2022).

[25] G. Bitar, *Optimization-based Trajectory Planning and Automatic Docking for Autonomous Ferries*. PhD thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2021.

[26] IMO - International Maritime Organization, “COLREG - Preventing collisions at sea.” Available at: https://www.imo.org/en/OurWork/Safety/Pages/Preventing-Collisions.aspx. (Accessed 05/03/2023).

[27] B.-O. H. Eriksen and M. Breivik, “MPC-Based mid-level collision avoidance for ASVs using nonlinear programming,” in *Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA)*, (Hawaii, USA), pp. 766–772, 2017.

[28] H.-C. Burmeister and M. Constapel, “Autonomous collision avoidance at sea: A survey,” *Frontiers in Robotics and AI*, vol. 8, pp. 739013–739013, 2021.

[29] H. Myre, “Collision avoidance for autonomous surface vehicles using velocity obstacle and set-based guidance,” Master’s thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2016.

[30] P. K. E. Minne, “Automatic testing of maritime collision avoidance algorithms,” Master’s thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2017.

[31] K. Woerner, *Multi-contact protocol-constrained collision avoidance for autonomous marine vehicles*. PhD thesis, Massachusetts Institute of Technology (MIT), Department of Mechanical Engineering, Cambridge, MA, USA, 2016.

[32] I. B. Hagen, *Topics on Marine Collision Avoidance*. PhD thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2022.

[33] K. Woerner, M. R. Benjamin, M. Novitzky, and J. J. Leonard, “Quantifying protocol evaluation for autonomous collision avoidance: Toward establishing COLREGS compliance metrics,” *Autonomous robots*, vol. 43, no. 4, pp. 967–991, 2019.

[34] A. Cockcroft and J. Lameijer, *A guide to the collision avoidance rules: International regulations for preventing collisions at sea*. Elsevier, 2003.

[35] Skipssikkerhetsloven, “Lov om skipssikkerhet.” Available at: https://lovdata.no/dokument/NL/lov/2007-02-16-9/KAPITTEL_2#6, 2007. (Accessed 22/05/2023).

[36] K. Wróbel, M. Gil, Y. Huang, and R. Wawruch, “The Vagueness of COLREG versus Collision Avoidance Techniques - A Discussion on the Current State and Future Challenges Concerning the Operation of Autonomous Ships,” *Sustainability*, vol. 14, no. 24, 2022.

[37] E. F. Brekke, E. Eide, B.-O. H. Eriksen, E. F. Wilthil, M. Breivik, E. Skjellaug, K. Helgesen, A. M. Lekkas, A. B. Martinsen, E. H. Thyri, T. Torben, E. Veitch, O. A. Alsos, and T. A. Johansen, "milliAmpere: An autonomous ferry prototype," *Journal of Physics. Conference series*, vol. 2311, no. 1, 2022.

[38] M. Breivik, V. E. Hovstein, and T. I. Fossen, "Straight-Line Target Tracking for Unmanned Surface Vehicles," *Modeling, Identification and Control*, vol. 29, no. 4, pp. 131–149, 2008.

[39] M. E. N. Sørensen, E. S. Bjørne, and M. Breivik, "Performance comparison of backstepping-based adaptive controllers for marine surface vessels," in *Proceedings of the 2016 IEEE Conference on Control Applications (CCA)*, (Buenos Aires, Argentina), pp. 891–897, 2016.

[40] J. Andersson, J. Gillis, G. Horn, J. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, 2018.

[41] P. G. B. Torvund, "Nonlinear Autonomous Docking and Path-Following Control Systems for the Otter USV," Master's thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2020.

[42] Goa Shipyard Limited, "15M. High Speed Patrol Boat." Available at: https://goashipyard.in/products/product-history/15-m-high-speed-patrol-boat/. (Accessed 14/03/2023).

[43] A. A. Pedersen, "Optimization Based System Identification for the milliAmpere Ferry," Master's thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2019.

[44] J. Tamimi and P. Li, "Nonlinear Model Predictive Control Using Multiple Shooting Combined with Collocation on Finite Elements," in *Proceedings of the 7th IFAC Symposium on Advanced Control of Chemical Processes*, vol. 42, (Istanbul, Turkey), pp. 703–708, 2009.

[45] G. Bitar, A. B. Martinsen, A. M. Lekkas, and M. Breivik, "Two-Stage Optimized Trajectory Planning for ASVs Under Polygonal Obstacle Constraints: Theory and Experiments," *IEEE Access*, vol. 8, pp. 199953–199969, 2020.

[46] K. Tøndel and H. Martens, "Analyzing complex mathematical model behavior by partial least squares regression-based multivariate metamodeling," *WIREs Computational Statistics*, vol. 6, no. 6, pp. 440–475, 2014.

[47] A. Uttisrud, "Hybrid Collision Avoidance for Autonomous Passenger Ferries," Master's thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2020.

NTNU

Norwegian University of
Science and Technology