Master's thesis

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

Tom Daniel Grande

PSB-MPC Collision Avoidance with Anti-Grounding

Master's thesis in Cybernetics and Robotics
Supervisor: Tor Arne Johansen
February 2021

NTNU
Norwegian University of
Science and Technology

Tom Daniel Grande

PSB-MPC Collision Avoidance with Anti-Grounding

Master's thesis in Cybernetics and Robotics
Supervisor: Tor Arne Johansen
February 2021

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

NTNU
Norwegian University of
Science and Technology

Chapter 1

Problem Description

Most algorithms for automatic collision avoidance assume open ocean conditions, i.e. they neglect that grounding and collision with static obstacles must be considered in coastal scenarios. The objective of this thesis is to extend the scenario-based Model Predictive Control (SB-MPC) approach to COLREGS compliant collision-avoidance Johansen et al. (2016) with anti-grounding. This thesis will address the following tasks:

- Literature review.
- Consider how to represent grounding hazards and static obstacles based on electronic navigational charts, keeping in mind that the data will be used to define an additional term in an SB-MPC cost function and the approach must therefore be computationally efficient in order to enable real-time computation.
- Implement an extended SB-MPC algorithm based on the existing code base and the proposed model and cost for anti-grounding.
- Consider how to formulate the method in a risk-based framework by combining probabilistic SB-MPC (PSB-MPC) formulation and probabilistic grounding risk.
- Test the method using simulations.

Summary

This thesis looks at autonomous surface vessels (SV) as the future of maritime industrial shipping. To obtain a fully autonomous SV, further work on a robust and complete collision avoidance system (CAS) were carried out. Main focus areas were to include anti-grounding system on top of a electronic navigational chart-map extraction environment; this with the aim of complementing the International Regulations for Preventing Collisions at Sea (COLREGS)-compliant probabilistic scenario-based model predictive control (PSB-MPC).

A wide area of simulation cases for testing the anti-grounding system on top of the ENC are carried out, including; bare bones simulation with targeted focus on the anti-grounding cost-term, simulation in challenging eviroments with the inclusion of rocks and bridge pylons and full scale system simulations with obstacle ships, challenging static obstacle scenarios. These simulations give a promising outlook for combining anti-grounding and anti-collision in the same MPC structure in future research and real life testing.

Preface

This thesis concludes my Master of Science degree in Cybernetics and Robotics, with a main profile in autonomous systems at the Norwegian University of Science and Technology (NTNU). This thesis investigates the possibility to implement an ENC-module for the addition of a anti-grounding term into the PSB-MPC developed in Johansen et al. (2016); Tengsedal et al. (2020); Tengesdal and Johansen (2020)

I would like to thank Tor Arne Johansen for guidance on both my project-thesis, on Modeling and Simulation of an LNG engine, and this Master Thesis on PSB-MPC Collision avoidance with anti-grounding for ASV. Furthermore I would like to thank my co-supervisors Trym Tengsedal and Simon Blindheim for help and support on the collision avoidance and anti-grounding, proof-reading and for answering questions during any time of day.

Finally I would like to thank my family and friends, especially Emil for helping me proof-read the thesis.

Table of Contents

List of Tables

List of Figures

Abbreviations

ASV	=	Autonomous Surface Vessel
CG	=	Center of Gravity
COLAV	=	Collision Avoidance
COLREGS	=	International Regulations for Preventing Collisions at Sea
CPA	=	Closest Point of Approach
DOF	=	Degrees Of Freedom
ECDIS	=	Electronic Chart Display and Information System
ENC	=	Electronic Navigational Charts
ESRI	=	Environmental Systems Research Institute
FGDB	=	ESRI File Geodatabase
IHO	=	International Hydrographic Prganization
IMO	=	International Maritime Organization
KF	=	Kalman Filter
LOS	=	Line-Of-Sight
MPC	=	Model Predictive Control
NED	=	North-East-Down
NLP	=	Non Linear Program
OCP	=	Optimal Control Problem
PSB-MPC	=	Probabilistic Scenario Based Model Predictive Control
SB-MPC	=	Scenario Based Model Predictive Control
UTM	=	Universal Transverse Mercator
WP	=	Waypoint
WPP	=	Waypoint Path Progression

Nomenclature

1.1 Greek

α	=	path tangential angle
η	=	pose given in earth fixed reference frame
δ_t	=	Simulation timestep
δ_u	=	penalty function for speed offset
δ_χ	=	penalty function for course offset
$\rho(\cdot)$	=	ad hoc risk cost function
ϕ	=	rotation about the x-axis in Euler angles
θ	=	rotation about the y-axis in Euler angles
ψ	=	rotation about the z-axis in Euler angles
τ	=	forces from vessel actuator
χ_d	=	desired course in guidance algorithm
χ	=	course offset
χ_m	=	modified optimal course from MPC
κ	=	COLREGS penalization term
ζ	=	grounding cost sensitivity constant
μ_1	=	Grounding cost term constant
μ_2	=	Grounding cost wind-term constant

1.2 Lowercase

e	=	cross track error
f(·)	=	cross track error
h(·)	=	chattering penalization function
l_r	=	distance from rudder to CG
m	=	mass
n_χ	=	number of course offset scenarios
n_u	=	number of propulsion commands
p	=	rotation about x-axis
p_i	=	obstacle tracked state
q	=	rotation about y-axis
r	=	rotation about z-axis
t	=	time
u	=	surge
u	=	input control signal in MPC
u_m	=	modified optimal control output from MPC
u_d	=	desired control output from mission planner
v	=	sway
w	=	heave
x	=	position in x-direction
y	=	position in y-direction
y	=	predicted horizon outputs in MPC
z	=	position in z-direction

1.3 Uppercase

C	=	System Coriolis matrix
D	=	System damping matrix
H	=	Optimization problem
I	=	Inertia
K	=	Moment about x-axis
K_p	=	Proportional controller gain
K_ω	=	Horizon weight parameter
K_{sgn}	=	manuvering-change constant
M	=	Moment about y-axis
M_s	=	System inertia matrix
N	=	Moment about z-axis
P	=	Collision probability
R	=	Rotation matrix
V_w	=	wind velocity relative to the ship's velocity
X	=	Force in x-direction
Y	=	Force in y-direction
Z	=	Force in z-direction

Chapter 2

Literature Review and Introduction

2.1 Introduction

This literature review serves the purpose of looking at contributions to the field of optimal control (mainly MPC) in collision avoidance systems for autonomous surface vessels (ASV). The investigation has furthermore the goal of reviewing the state of the art methods of optimization, guidance and control and finally the inclusion of both anti-collision and anti-grounding terms into the optimization problem.

2.2 Background and motivation

Autonomous shipping is considered to be the future of the maritime industry, and will likely play a large role in Norwegian industry in years to come. In the context of safe autonomous shipping; control for ASV's is a highly relevant, broad, important and complex field of research, this due to the fact that about 75% of accidents occur due to human errors. For a fully autonomous SV there is a number of factors to account for. Wind, Ship collision rules, currents, static and dynamic obstacles - to name some of the most important. The rules for ship collision avoidance is set by COLREGS (2020), "Convention on the International Regulations for Preventing Collisions at Sea" by the IMO, "International Maritime Organization". The COLREGS was developed in 1972 as a consequence of collisions becoming a problem during the 1960's. In addition to local rules, coastal traffic lights etc., also could also play a role in optimal control.

2.3 Literature review

2.3.1 General

The field of collision avoidance for ASV's stems from a series of articles: Goodwin (1975), Davis et al. (1980) and Coldwell (1983) to name a few. Goodwin (1975) proposed a setup

which discretely divided the ship domain into three zones. This concept can be seen as a building block for Davis et al. (1980) and Coldwell (1983), where respectively a circle action domain for necessity of evasive maneuvers and a scenario based ship domain for navigation in restricted waters were proposed.

From a more modern perspective one can divide the collision avoidance problem into two categories: Mathematical algorithms and Evolutionary algorithms. The scope of this work is as mentioned focused on investigating the former. Mathematical algorithms can also be divided into two categories: local and global methods. A general rule is that purely local methods handle dynamic obstacles well, but lack the ability to take static obstacles into account, and vice versa for purely global methods. As an ASV in a real maritime scenario both faces dynamic and static obstacles, the need for a merge of the global and local methods arises.

2.3.2 MPC and COLAV

In Johansen et al. (2016) a finite set of alternative control behaviours and a MPC concept compatible with COLREGS (2020) is reviewed. In the background section the article states the need for a concept which better scales with a large number of dynamic obstacles and simultaneously also accurately can take the dynamics of the ship into consideration. It aims to solve the problem of handling both dynamic and static obstacles at the same time. More specifically Johansen et al. (2016) proposes 13x4 (course-offset x propulsion commands) different control behaviours to comply with computation time available. The important ship dynamics is a 3DOF model based on models in Fossen (2011). The control behaviour of the ship is calculated by minimizing the hazard function $\mathcal{H}^k(t_0)$, which includes anti-collision terms based on the cost of collision with an obstacle, a cost for violating COLREGS, a cost for the choice of maneuvering effort and a grounding term specified to be ship-sensory and ENC-based.

Further work on Johansen et al. (2016) is carried out in Hagen et al. (2018), Tengesdal and Johansen (2020) Kufoalor et al. (2020) and Tengsedal et al. (2020). In Hagen et al. (2018), a strategy for an existing marine vessel's guidance system is proposed. Here the COLAV and the guidance module are separated, the number of control behaviour scenarios are 13x3 and the 3DOF dynamical ship model is replaced by a kinematic equation. The MPC COLAV was implemented and tested on Telemetron which is an RBB from Maritime Robotics. Results showed that the MPC COLAV proposed in Johansen et al. (2016) can be implemented in an already implemented guidance system architecture with satisfactory results, and in addition the transitional cost which incentives more continuous COLREGS maneuvers, neglecting oscillating behaviours displayed in overtaking and crossing situations.

In Kufoalor et al. (2020) a field verification of the COLAV MPC algorithm, in collaboration with Maritime Robotics AS, the authorities of Netherlands and Deltares (Institute for applied research) is carried out. The receding horizon implementation of the SB-MPC algorithm showed great results when it comes to abiding COLREGS in dangerous situation, while the structure of the algorithm itself allowed for more advanced dynamic models, i.e 3DOF instead of a single kinematic equation.

In Tengsedal et al. (2020), and Tengesdal and Johansen (2020) a probabilistic version of the SB-MPC is examined. Estimates of the probability of collision with all nearby obstacles is done using a combination of Monte Carlo simulation (MCS) and a Kalman Filter (KF). The model is shown to have a larger situational awareness and to be on par with the SB-MPC's path following. Further in Tengesdal and Johansen (2020) a generalized framework for obstacle intent inference is introduced, which further seeks to improve the SB-MPC's ability to obtain situational awareness.

2.3.3 Anti-grounding

In Blindheim et al. (2020), Bakdi et al. (2019) and Montewka et al. (2011), further focus on anti-grounding systems is presented. Blindheim et al. (2020) investigates emergency management during failure modes. This is done by forming a OCP which includes a path progression cost, control input costs and an ad hoc risk cost. The NLP is solved in a receding horizon fashion, yielding an MPC scheme. Furthermore, Blindheim et al. (2020) focuses on anti-grounding while COLAV is assumed to be natural extensions. The anti-grounding is included in the ad hoc risk function $\rho(X_k, \Theta_k)$, which is formulated as a non-explicit constraint yielding an evaluation of the risk of grounding even though they are very high. The ad hoc risk function takes the grounding obstacles which is represented as circles with a centre point and radius, wind relative speed and direction, into account in an exponentially weighted manner.

In Bakdi et al. (2019) an adaptive ship-safety-domain is proposed with spatial risk functions to identify both collision and grounding risk based on motion and maneuverability conditions for all vessels. The grounding risk of a vessel is calculated based on the intersection of a vessels safety domain with shore line polygon shapes. Where the grounding risk function decays exponentially with distance to the obstacle.

Montewka et al. (2011) considers oil tankers risk of collision and grounding in the Gulf of Finland. The study focuses on modeling the accident probabilities. The grounding model is based on gravitational-like interactions between a ship and the obstacles with which it is surrounded.

Mazaheri et al. (2013) two years later conducted a literature review on modeling the risk of ship grounding. A number of 13 different approaches from 1974 - 2011 were considered stemming from 90 articles, reports and dissertations, including Montewka et al. (2011). The different approaches was compared with regards to inputs, outputs, decision making potential, etc.

2.4 COLAV system

As investigated in the literature review, the choice of COLAV and anti-grounding could be carried out by a number of methods. Usually the methods used are split into two main categories, namely: Mathematical and Evolutionary, which distinguishes between environmental and mathematical descriptions and Artificial Intelligence trained on relevant data sets. Furthermore, a distinction into proactive and reactive algorithms could be made, where the latter includes Velocity obstacle (VO) Fiorini and Shiller (March 2018) and Dynamic window (DW) Fox and Burgard (April 1997); Eriksen et al. (March 2018)

approaches. These short term decision algorithms generally have a hard time including COLREGS, even though they could be modified to resemble COLREGS-compliant behavior.

Proactive-long term methods include the evolutionary methods which are based on neural networks and fuzzy logic. Hu et al. (June 2020); Statheros et al. (January 2008) These algorithms handle long-term planning and COLREGS-compliance well, but lack certain aspects in short temporal reactions, which the previously discussed reactive algorithms are better at.

Chapter 3

Basic Theory

3.1 Introduction

The theory chapter is heavily based on the work from Johansen et al. (2016) and **?**. The important theory is repeated here to make the thesis self-contained. The overall system architecture is demonstrated in a block diagram in figure 5.2

3.2 Reference frames

The coordinate frames used in the implemented framework is the NED reference frame and the BODY fixed frame. Notation utilized in the ship dynamics formulas are from SNAME and are shown in figure 3.1

3.3 Ship dynamics

This section described the theory used for modeling the ship dynamics used for the simulated ASV and the simulated obstacle ships. The model used is the 3DOF horizontal plane ship dynamics model. Fossen (2011).

Figure 3.1 The notation of SNAME (1950) for Marine vessels

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

$$\dot{\eta} = R(\psi)\nu \tag{3.1}$$

$$M\dot{v} + C(v)v + D(v)v = \tau + R(\psi)\tau_w \tag{3.2}$$

Where $\eta = \begin{bmatrix} N & E & \psi \end{bmatrix}^T$ is the pose given in the earth fixed NED reference frame, $R(\psi)$ is the rotation matrix transform from body-fixed velocities to world fixed given by equation 7.10 in Fossen (2011):

$$R(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

and $\nu = \begin{bmatrix} u & v & r \end{bmatrix}^T$. Furthermore M_s is the system inertia matrix and is defined:

$$M_s = M_A + M_{RB} \tag{3.4}$$

$$= \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} + \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{3.5}$$

C is the coreolis matrix defined as:

$$C(\nu) = C_A(\nu) + C_{RB}(\nu) \tag{3.6}$$

$$= \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \tag{3.7}$$

D is the damping matrix defined as:

$$D(\nu) = D + D_n(\nu) \tag{3.8}$$

$$= \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} + \begin{bmatrix} -X_{|u|u}|u| & 0 & 0 \\ 0 & -Y_{|v|v}|v| & -Y_{|v|r}|v| \\ 0 & -N_{|v|v}|v| & -N_{|v|r}|v| \end{bmatrix} \tag{3.9}$$

τ is the forces from the vessels actuator given here as:

$$\tau = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ l_r F_y \end{bmatrix} \tag{3.10}$$

where l_r is the distance from the rudder to the CG; which is the moment arm in which the moment about the z-axis is working. The results are obtained by the assumption that we have no vertical motion, homogeneous mass distribution, symmetry about the xz-plane and that the CO and the CG are set such that the distance between them in y-deriction is equal to zero i.e., $y_g = 0$.

3.4 PSB-MPC Controllers

For the Control input update in simulation-prediction two controllers are implemented. The controllers are a feedback linearizing controller on speed and a Proportional-Derivative Controller (PD) for course. Respectivly:

$$F_x = C_{vv}(0) + D_{vv}(0) + K_{p_u} m(u_d - u) \tag{3.11}$$
$$= (-mv + Y_{\dot{v}} v + Y_{\dot{r}} r)r - (X_u + X_{|u|u}|u| + X_{uuu} u)u + K_{p_u} m(u_d - u) \tag{3.12}$$

Where C is the Coriolis vector and D is the damping vector both described in section 4.3. K_{p_u} is the proportional gain vector.

$$F_y = \frac{K_{p_\chi} I_z \chi_\Delta - K_{d_\chi} r}{l_r} \tag{3.13}$$
$$\chi_\Delta = \chi_d - \chi \tag{3.14}$$

Here K_{d_ψ} and K_{p_ψ} are the controller gains. While I_z is the moment of inertia about z-axis and l_r the yaw moment acting arm.

Finally a saturation is carried out as follows:

$$F_{y_{min}} < F_y < F_{y_{max}} \tag{3.15}$$
$$F_{x_{min}} < F_x < F_{x_{max}} \tag{3.16}$$

3.5 Guidance Method

The Guidance law used in simulation is the LOS path-following algorithm. LOS-guidance (Line of Sight) is a path-following algorithm which calculates the systems desired course angle. Here the control objective is to minimize cross-track-error i.e., the shortest distance from the simulated ship to the desired path.

First the path tangential angle is calculated as follows:

$$\alpha = arctan2(WP_{y_{k+1}} - WP_{y_k}, WP_{x_{k+1}} - WP_{x_k}) \tag{3.17}$$
$$\tag{3.18}$$

Then the cross-track error is calculated and integrated:

$$e = -(N - WP_{x_{k+1}})sin(\alpha) + (E - WP_{y_{k+1}} cos(\alpha)) \tag{3.19}$$
$$e_i = \int_0^{\Delta t} e \, dt \tag{3.20}$$
$$\chi_d = \alpha + arctan2(-(e + K_{i_{LOS}} e_i), \Delta)) \tag{3.21}$$

Where Δ is the look-ahead distance, K_i is the integral gain, N and E is the northing and easting respectively.

Chapter 4

Electronic Navigational Chart Module

4.1 Introduction

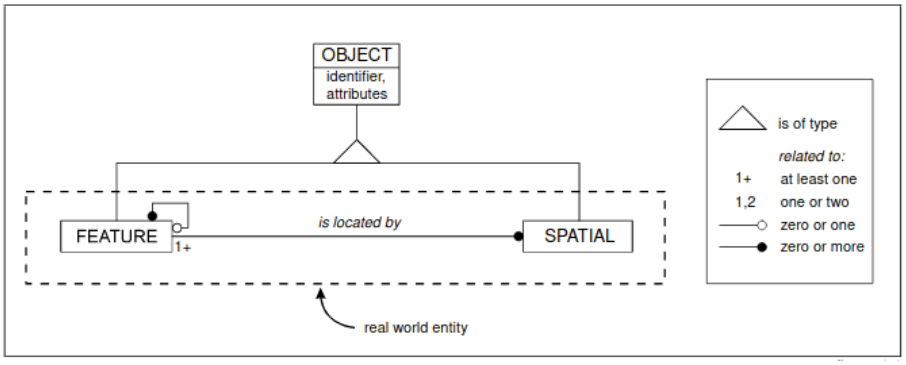
To assure a rigid simulation environment for the PSB-MPC algorithm with both collision avoidance and anti-grounding we need to establish a framework for representation of the various static obstacles faced in a simulation of a real life scenario. Thus, this chapter will introduce the background material and theories relevant for the anti-grounding system. Such a environment should be able to represent the static obstacles at sea, including location, size, shape and uncertainty. These obstacles include, but are not limited to, seabed, rocks, land-area, shore and shallows. To implement such a enviroment, one should use a collection of data describing the real world; an ENC.

4.2 ENC

Electronic Navigational Charts (ENC) is the international description for official navigational charts and are used for navigation in the electronic chart-system ECDIS (Electronic Chart Display and Information System). The ENC is furthermore a vector-map which is produced in compliance with the ”S-57” standard by the IHO (International Hydrographic Organization). In Norway ”Kartverket” is the sea-chart authority and are responsible for production and maintenance of the Norwegian ENC’s.

ENC’s are similar to regular paper-charts, in that they are edited in given scales, but usually contains more detailed seabed and depth information but less detailed presentation of land-areas.

Figure 4.1 Model Overview of S-57 standard from Organization (Edition 3.1 - November 2000)



4.3 S-57 Model standard

S-57 is the format for the data used in transfer of digital hydrographic data between national hydrographic offices (like Kartverket), and for its distribution to users of the data. The S-57 ENC product specification defines how hydrographic offices will construct an Electronic Navigational Chart. By meeting these specifications, hydrographic offices will ensure that all electronic charts contain all the chart information that is necessary for safe navigation. Within this model these sets of characteristics are defined in terms of feature objects and spatial objects Organization (Edition 3.1 - November 2000). The model overview can be seen in 4.1. There are four types of feature objects:

- Meta - Feature object which contains information about other objects.
- Cartographic - Feature object which contains information about the cartographic representation.
- Geo - Feature object which carries the descriptive characteristics of a real world entity.
- Collection - Feature object which describes the relationship between other objects.

As the S-57 standard contains a very detailed list of objects in the real world which could potentially be used in navigation, a simplification and specification of interest areas for anti-grounding have to be made.

4.4 Shape file generation

A script for reading and processing the ENC-files, more spesifically ESRI File Geodatabase (FileGDB), was used from https://github.com/simbli/seacharts, and adapted to the purpose of this thesis to create a shapefile.

4.4.1 Shapefile

A shapefile is a simple, nontopological format for storing the geometric location and attribute information of geographic features. Geographic features in a shapefile can be represented by points, lines, or polygons (areas).

The shapefile is in fact a grouping of several files formatted to represent different aspects of geodata:

- .shp — shape format; the feature geometry itself.
- .shx — shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly.
- .dbf — attribute format; columnar attributes for each shape, in dBase IV format.

The format used in this thesis is the FGDB 10.0 format.

4.4.2 European Terrestrial Reference System 1989

To accurately represent and measure properties in the ENC a geodetic datum is used. A geodetic datum is a reference point that a coordinate system relies on and refers to. EUREF89 (European Terrestrial Reference System 1989) is the official geodetic datum in Norway, and are the datum used in the shapefile generation module.

Universal Transverse Mercator

The official projection used by Kartverket is the UTM-projection. UTM is a Gauss-Krüger-projection with 6° zone-width. Because of the large zone-width there are defined a measurement-factor: 0.9996 along the limit meridians. This means that a distance measured in the terrain have to be corrected with up to 4 cm per 100 m. This can be neglected within this thesis. Furthermore the UTM sone 33, 2d are chosen for the area we are interested in.

4.4.3 Supported Features in ENC-module

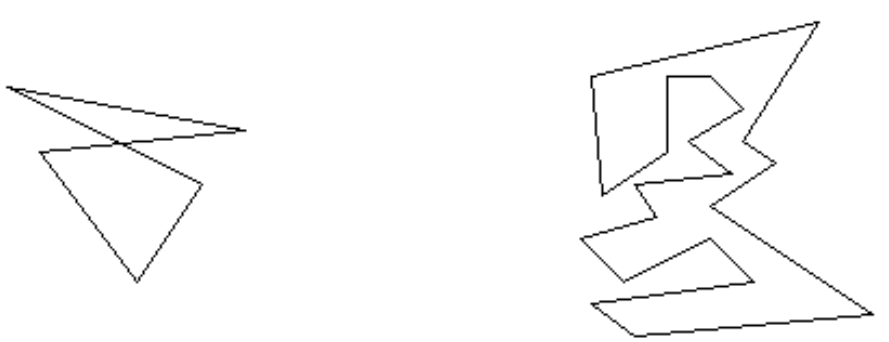
Terrain

The ENC-module python script supports the following terrain-types: seabed, land, rocks, shallows and shore.

Geometry

The geometric representation of the terrain are stored as points and polygons. Furthermore the ENC module saves the different supported types of relevant geometric data in separate shapefiles. In the parsing of the data a folder is made for each archetype of data. For the purpose of simulation and implementation an offline handling of the data should be made separate from the simulation loop.

Figure 4.2 Simple (to the right) versus complex (to the left) polygon illustration



4.5 Polygons and the Boost Library

As the shapefiles representation of the terrain types are stored as points and polygons, a rigid framework for geometric representation and manipulation in C++ have to be developed.

4.5.1 Polygons

For the manipulation and representation of polygons the Boost-Polygon library is used. The Boost-Polygon library provides algorithms focused on manipulating planar polygon geometry data.

Classification

The classifications of the polygons used are ”simple”:

- A polygon encloses a region (called its interior) which always has a measurable area.
- The line segments that make up a polygon (called sides or edges) meet only at their endpoints, called vertices.
- Exactly two edges meet at each vertex.
- The number of edges always equals the number of vertices.
- Has no holes

Furthermore the polygons can both be convex and concave. These classifications is to ensure that the polygons handled by the Boost library are classified properly and errors are neglected. A representation of a complex versus a simple polygon can be viewed in figure 4.2

Figure 4.3 Map area used in ENC module, with origin marked ”C” at: E/N=(307065, 7075541)

4.6 Implementation

4.6.1 Python Shapefile generation

The python script for shapefile-generation has the following settings:

- origin measured in easting/northing (UTM zone 33N)
- window size in easting/northing in meters
- Chosen from the list of supported regions

For simulation the following parameters were used:

- origin = (307065, 7075541)
- window size in easting/northing in meters
- region = ’Trøndelag’

which is the coastal fjord area outside of Levanger/Ytterøya seen in figure 4.3

Table 4.1: ENC-Module choice of parameters

Description	Parameter	Value
Origin	O	(307065, 7075541)
Window-size	(E,N)	(20000, 16000)
Region	T	'Trøndelag'
data	-	'land'

4.6.2 C++ data manipulation

As mentioned in section 5.4.3 the supported types of data is stored in different shapefile-structure folders. Furthermore in section 5.5 we see that these are saved as polygons. The idea and purpose of this is that one can now use the Boost-Polygon library to extract the relevant features for our simulation environment.

Algorithm 1 Pseudocode of ENC data manipulation

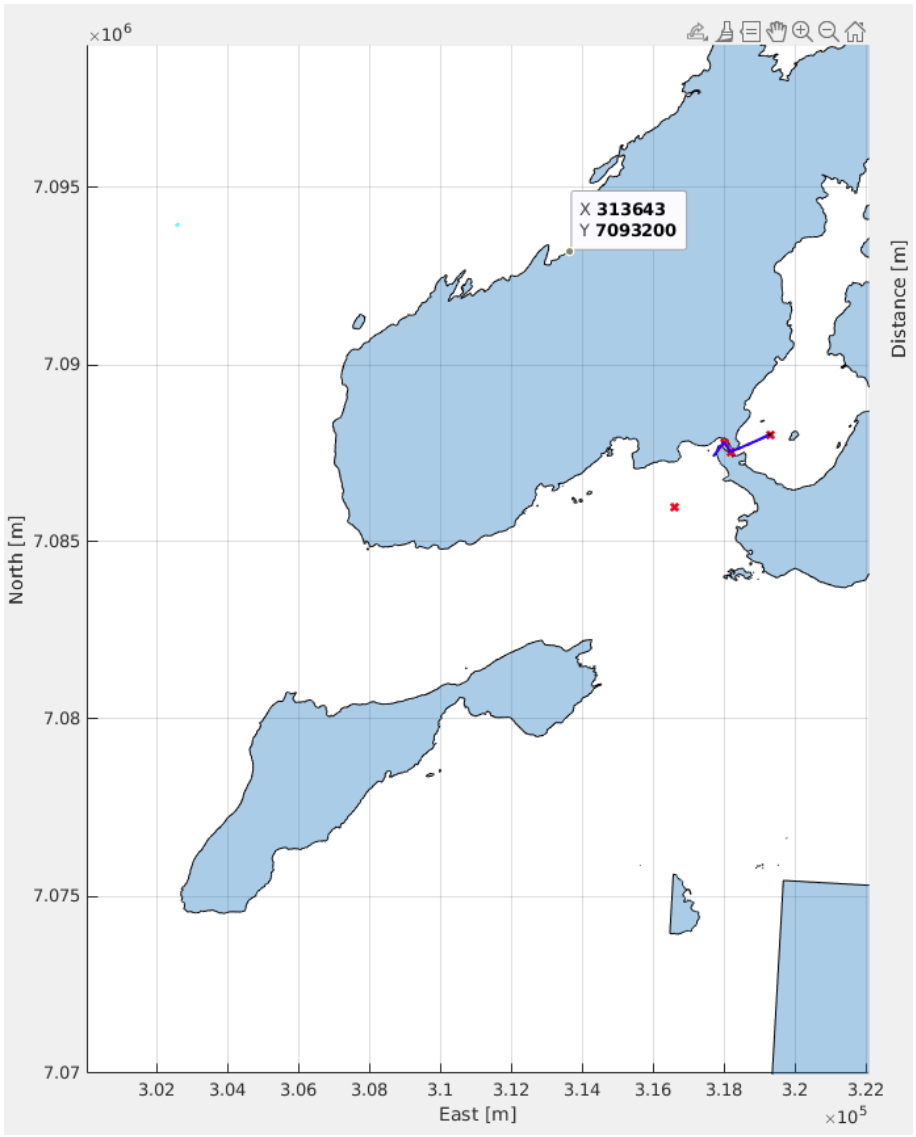
```
Load data objects from ENC python module shapefiles
Define Polygon type from Boost Polygon framework
convert Shape file Object to Defined Boost Polygon "Object"
for Object n=1 Object n= N do
    Add Objects that are simple noholed polygons to polygonlist
    calculate distances from preplanned path to polygons
    if distance>Threshold then Pop Polygon from list for calculation efficiency
    end if
end for
```

4.7 Polygon map overview

Figure 4.4 shows a plot of the implemented, shapefile-generated static obstacle polygons. Note that this is simply a plot of the polygons parsed with the ENC-module parameters described in table **??**.

The chosen parameters here implies that no rocks, sea-depth or other static obstacle choices are accounted for in this particular plot. However, as all added extra features are implemented as simple polygons, there is no loss off generality, nor complexity with respect to the simulation only using land-area.

Figure 4.4 Electronic map extraction of Land area surrounding Beitstadfjorden in Trøndelag. Blue marked areas are simple polygons making up land-area. White text-box next to grey dot indicates a reference position in easting/northing. Red dots are check-points in the CAS. The blue line is the CAS predicted trajectory.



Chapter 5

Collision and grounding avoidance

5.1 Introduction

To make an ASV autonomous one need an complete anti-collision and anti-grounding system. Such a system should have safety and predictability as its main focuses in maneuvering. The CAS commonly uses real time sensory inputs, pre-defined rulesets and pre-computed geographical data. Real time sensory inputs include positional, speed, shallow ground sensors, etc. Pre-defined rule sets includes COLREGS, light-signals among others and pre-computed data involves sea-depth, land area and rocks to name a few. The International Regulations for Preventing Collisions at Sea (COLREGS) determines the ASV's intended actions during interaction with other vessels and could in many ways be seen as the baseline for designing a complete and robust anti-collision algorithm.
To tackle the factors discussed in the introduction the MPC approach to solving the collision avoidance problem is chosen in this thesis. Collision avoidance based on MPC is thoroughly discussed in chapter 3, and the main implementation and concept is based on Tengsedal et al. (2020). Here the algorithm system output is a modified optimal course and surge offset.

5.2 International Regulations for Preventing Collisions at Sea (COLREGS)

In the introduction a baseline rule set for preventing collisions at sea were mentioned as one of the most important inclusions in a complete CAS for ASV's. This section includes the modern international rule set, namely the COLREGS, which came into effect in 1972. The COLREGS ruleset is devided into six parts:

- Part A - General (Rules 1-3)
- Part B - Steering and Sailing rules (Rules 4-19)

- Part C - Lights and Shapes (Rules 11-18)
- Part D - Sound and Light Signals (Rules 20-31)
- Part E - Exemptions (Rule 38)
- Part F - Verification of Compliance with the Provisions of the (Rule 39-41) Convention

In addition the COLREGS contains four annexes:

- Annex I - Positioning and technical details of lights and shapes
- Annex II - Additional signals for fishing vessels fishing in close proximity
- Annex III - Technical details of sound signal appliances
- Annex IV - Distress signals, which lists the signals indicating distress and need of assistance.

For the purpose of this thesis, Part B; containing the rules for sailing and steering, are repeated. More specifically Rules 6,8, 13-17.

Rule 6

Safe Speed; Every vessel shall at all times proceed at a safe speed so that she can take proper and effective action to avoid collision and be stopped within a distance appropriate to the prevailing circumstances and conditions. Visibility, traffic density, weather, water depth and more must be taken into account when determining a safe speed.

Rule 8

Action to avoid collision; For a vessel in risk of collision, action to avoid it shall be taken with accordance to the rules of part B. Any alteration of course and/or speed to avoid collision shall be large enough to be readily apparent and made in ample time. The action shall result in passing at a safe distance.

Rule 13

Overtaking; Any vessel overtaking any other shall keep out of the way of the vessel being overtaken. It is deemed an overtake vessel when it comes up with another vessel from a direction more than 22.5 degrees abaft her beam.

Rule 14

Head-on situation; When two power-driven vessels are meeting on reciprocal courses we have a head-on situation and each shall alter her course to starboard, so they both pass on port side

Rule 15

Crossing situation; There is a crossing situation if two power-driven vessels are crossing in a way that involve risk of collision. Then the vessel which has the other on her starboard side, is deemed the give-way vessel and should avoid collision as well as try to avoid passing in front of the other vessel.

Rule 16

Action by give-way vessel; A give-way vessel shall as far as possible take a substantial and early action to avoid collision.

Rule 17

Action by stand-on vessel; The stand-on vessel should keep her speed and course. However when it becomes apparent that the give-way vessels actions alone is not enough to avoid collision, the stand-on vessel should take action as well. This rule do not relieve the give-way vessel of her obligations.

5.3 Probabilistic Scenario-based Model Predictive Control

This section will include all the necessary background material to fully contextualize the PSB - MPC strategy used to solve the collision avoidance and anti-grounding problem stated.

5.3.1 Model Predictive Control

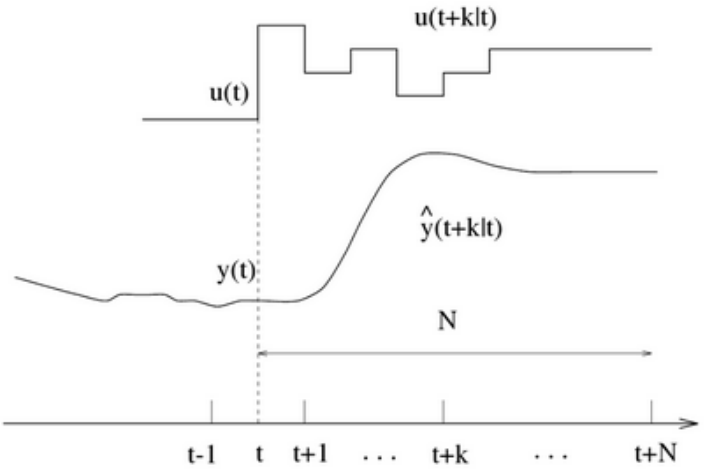
The methodology of controllers in the MPC familiy as represented in 5.1 is as follows:

The predicted horizon outputs $y(t+k|k)$ is predicted at each instant t using the process model over the determined prediction horizon N. These outputs for $k = 1, \ldots, N$ depends on the known values up to instant t, which are the past inputs and outputs, and on the future control signals $u(t + k|t), k = 0, \ldots, N - 1$, which are those to be sent to the system for calculation.

Furthermore, these control signals are calculated by optimizing a determined criterion to keep the process optimized with respect to the reference trajectory. This criterion is usually in the form of a quadratic function of the errors between the predicted output signal and the predicted reference trajectory. If there are no explicit solution, iterative models could be used.

Lastly, the control signal $u(t|t)$ is sent to the process while the next control signal which are calculated are rejected, this due to the next output instant $y(t + 1)$, which as already known, leading to all sequences brought up to date. Thus, $u(t + 1|t + 1)$ is calculated and is in principle different from $u(t + 1|t)$ because of the receeding horizon. Eduardo F. Camacho (2013)

Figure 5.1 Illustration of MPC strategy from Eduardo F. Camacho (2013)



5.4 The Probabilistic Scenario-based MPC

Model predictive control is a versatile tool in optimization and modeling as it incorporates the possibility to utilize mathematical models of both the systems and its surroundings. One MPC optimization method, and the method used in this thesis, is to optimize over a finite predetermined set of possible control behaviors. The strategy of this method is based on picking between a discrete number of outputs, based on the comparison of the cost of said outputs. Maiworm et al. (December 2015). Johansen et al. (2016)

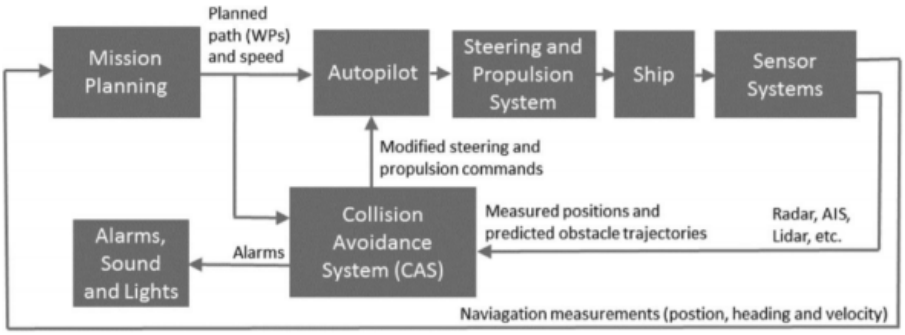
The term probabilistic refer to the solution of solving the track estimate of nearby obstacles. This uncertainty should be accounted for in a robust COLAV system in order to ensure safe operation in accordance to the traffic rules (COLREGS). In Tengesdal and Johansen (2020) a set of two events $\mathbb{A}_i$ and $\mathbb{B}_i$, where the former is the event of a collision occurring at some time $t_c \geq t_k$, and the latter is the event of a collision not occurring at any time $t_c \geq t_k$, are defined to determine the collision probability between the own-ship and obstacle ship. Collision is set to be the breach of a safety-region with radius, d_{safe}. The probability of collision is thus set to be $\mathbb{P}^i_{c,k} = P(\mathbb{A}_i) = 1 - P(\mathbb{B}_i)$. Finally the collision probability is found by integrating the obstacle tracked state PDF $p^i(\boldsymbol{x}; t_k)$:

$$\mathbb{P}^i_{c,k} = \int_S p^i(\boldsymbol{x}; t_k) dx \tag{5.1}$$

where S is a region which include all straight line trajectories.

The calculated collision probability is then filtered recursively using a Kalman Filter (KF). Further reading and understanding of the probabilistic SB-MPC is found in Tengsedal et al. (2020); Tengesdal and Johansen (2020).

Figure 5.2 The system architecture overview, denoting system flow between modules inspired by Johansen et al. (2016)



5.4.1 Collision Probability

5.5 Control system scenarios

The CAS outputs a course angle offset and a modified propulsion command that are given to the autopilot. These outputs are defined based on a set of scenarios, where the scenarios are made out of the state of the ship to be simulated, the desired path of the obstacle ships and the desired control behaviours.
Johansen et al. (2016) states a set of typical course angle offsets and modified propulsion commands as follows:

- Course offsets, $\chi \in$ (-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90) degrees
- Propulsion commands, u $\in$ (1,0.5,0) factor

The optimal control behavior then modifies the course angle offsets and modified propulsion commands as follows:

$$\chi_c = \chi_m + \chi_d \tag{5.2}$$
$$u_c = u_m u_d \tag{5.3}$$

where χ_c, u_c are the optimal course angle and optimal propulsion command, χ_m, u_m are the output from the PSB-MPC optimal control strategy and χ_d, u_d are the desired references from the mission planner.

With the above described scenarios there is a total of $n_{cb} = n_\chi$ * n_u = 3*13 = 39 control behaviours.

5.5.1 Multiple Sequential Avoidance Maneuvers

Scenario maneuvers is introduced to prevent conservative solutions on the prediction horizon. This increases the solution space complexity by $n_{cb}^{n_m}$ where n_{cb} is the number of control behaviours as described = 39, and n_m is the number of sequential avoidance maneuvers. In the original code-base these maneuvers were based on a maneuver scheme from Tengsedal et al. (2020), where the first avoidance maneuver always were made at t_0. The subsequent avoidance maneuvers were made immediately after the closest obstacle in the current collision situation or at CPA (Closest point of approach). However, a change to this had to be made due to the introduction of static obstacles and anti-grounding. Maneuver times were instead set at pre-defined time-intervals.

5.6 Cost Function

The cost-function used in this thesis is based on Johansen et al. (2016) and further changed to fit to the Probabilistic SB-MPC in Tengsedal et al. (2020), and finally fit to include the anti-grounding scheme introduced in this thesis.

5.6.1 General cost function

The general SB-MPC formulation to evaluate all possible scenarios, is based on Johansen et al. (2016) and are formulated as follows:

$$\mathcal{H}^k = \max_i \max_{t \in D(t_0)} (\mathcal{C}_i^k(t)\mathcal{R}_i^k(t) + \kappa_i \mu_i^k + \lambda_i \mathcal{T}_i^k(t) + f(P^k, \chi_{ca}^k)) + g(P^k, \chi_{ca}^k)) \quad (5.4)$$

where $f(\cdot)$ is a function containing a penalty function for keeping nominal speed and course. $g(\cdot)$ is the anti-grounding cost term. $\kappa_i \mathcal{M}_i^k$ is the COLREGS associated cost.$\lambda_i \mathcal{T}_i^k(t)$ is the COLREGS transitional cost. Finally $\mathcal{C}_i^k(t)\mathcal{R}_i^k(t)$ is the representation of the collision hazard risk cost.

5.7 PSB-MPC cost function

However, to account for multiple obstacle intentions and own-ship maneuvers in the prediction horizon; the PSB-MPC cost-function for the own-ship control behavior with index l is modified to

$$\mathcal{H}^l(t_0) = \max_i \sum_{a=1}^{n_a} P_a^i(t_0) C_a^{l,i} + g(\cdot) + \frac{1}{n_M} \sum_{M=1}^{n_M} f(u_{m,M}^l, u_{m,M-1}^l, \chi_{m,M}^l, \chi_{m,M-1}^l) \quad (5.5)$$

$$+ \frac{1}{n_M - 1} \sum_{M=2}^{n_M} h(t_M^l - t_{M-1}^l, \chi_{m,M}^l, \chi_{m,M-1}^l) \quad (5.6)$$

Tengsedal et al. (2020) where

$$C_a^{l,i} = \sum_{M=1}^{n_M} \frac{w^{i,s}}{n_{ps}^i(a)} \max_{t \in D(t_0)} [\mathcal{C}_i^{l,s}(t) P_c^{l,s,i}(t) + \kappa_i \mu_i^{l,s}] \tag{5.7}$$

The index a = 1, 2, ..., n_a is here defined as the obstacle intention. Sub- and superscripts l, i and s refers to control behaviour sequence, obstacle and prediction scenario respectively. Furthermore the term $\mathcal{C}_i^{l,s}$ is the collision cost, $P_c^{l,i,s}$ is the collision probability and ${\kappa_i}_i^{l,s}(t)$ is the COLREGS penalization term. $h(\cdot)$ is a penalization function on chattering, and $f(\cdot)$ are a control deviation penalization function.

The cost function and its terms can thus be divided into the following:

- The COLREGS-penalization cost
- The collision risk cost
- The grounding cost
- The control deviation cost
- The maneuvering-change chattering cost.

5.7.1 The COLREGS penalization cost

The COLREGS penalization term $\kappa_i \mu_i^{l,s}$, where κ is a tuning parameter, and $\mu_i^{l,s}$ is a boolean function determining there is a COLREGS-violating cost or not. This calculation is based on the predicted velocity of the own-ship and obstacle ship, $\vec{v}_o^k(t)$, $\vec{v}_i(t)$ where **i** denotes obstacle with index **i** , the unit LOS direction vector $\vec{L}_i^k(t)$, the predicted distance between the ships, and the minimum allowance distance $d_{o,i}(t)$ and d_i^{cl} respectively.

Furthermore the following definitions is used for the COLREGS-boolean calculation:

- Close: when $d_{o,i}(t) \leq \mathrm{d}_i^{cl}$
- Is overtaken: when $\vec{v}_o^k(t) \cdot \vec{v}_i(t) > cos(\phi_{overtaken}) |\vec{v}_o^k(t)||\vec{v}_i(t)|$
- Is Ahead: when $\vec{v}_o^k(t) \cdot \vec{L}_i^k(t) > cos(\phi_{ahead}) |\vec{v}_o^k(t)|$
- Is Starboard: when bearing angle of the unit LOS direction vector $\vec{L}_i^k(t)$ is larger then the own-ship heading
- Is head-on: when "Close" to ownship and $|\vec{v}_o^k(t)| < 0.05$, $\vec{v}_o^k(t) \cdot \vec{v}_i(t) < -cos(\phi_{headon}) |\vec{v}_o^k(t)||\vec{v}_i(t)|$ and Is Ahead
- Crossing: when "Close" and $\vec{v}_o^k(t) \cdot \vec{v}_i(t) > cos(\phi_{crossing}) |\vec{v}_o^k(t)||\vec{v}_i(t)|$

The final output to determine if the main COLREGS rules 13-15 have been violated is thus determined by the boolean expression:

Table 5.1: COLREGS violation parameters

Description	Parameter	Value
Angle for calculating if ship is ahead	ϕ_{ahead}	68.5
Angle for calculating if ship is overtaken	$\phi_{overtaken}$	68.5
Angle for calculating if ship is head on	ϕ_{headon}	22.5
Angle for calculating if ship is crossing	$\phi_{crossing}$	68.5
Cost of not complying with COLREGS constant	κ	3.0
COLREGS Transitional cost constant	κ_{TC}	3.0

$$\mu_i^{l,s} = \text{COLREGS rule 14}||\text{COLREGS rule 15} \tag{5.8}$$

$$\mu_i^{l,s} = (\text{Close } \& \text{ Is Starboard } \& \text{ Is Head-on}) \tag{5.9}$$

$$||(\text{Close } \& \text{ Is Starboard } \& \text{ Is Crossing } \& \text{ Is not overtaken}) \tag{5.10}$$

Rule 13 is here indirectly included in rule 14. Table 5.1 includes the relevant simulation parameters for the COLREGS violation-cost.

5.7.2 Collision cost

PSB-MPC formulation with probabilistic collision cost as described in equation 5.7. Here we have already discussed the COLREGS violation part.

The collision cost is calculated:

$$\mathcal{C}_i^{l,s} = K_{coll} \cdot |\vec{v}_o^k(t) - \vec{v}_i(t)| \tag{5.11}$$

where K_{coll} is a tuning parameter.Furthermore the weights described in 5.7 are defined as:

$$w^{i,s} = \begin{cases} Pr(CC^i), & \text{if i is CC} \in s \\ 1 - Pr(CC^i), & \text{otherwise} \end{cases} \tag{5.12}$$

Here, the weights are a check on whether the obstacle is CC (Colregs Compliant) in the given scenarios, which is done by determining wheter it breaches COLREGS given the own-ship keeps its course.

5.7.3 The Grounding cost

This section investigates the anti-grounding term to be implemented in the PSB-MPC framework. First the main theoretical ideas are discussed in the general cost function formulation subsection

General cost formulation formulation

The grounding cost term is primarily based on Blindheim et al. (2020), where the grounding cost term is described as an ad hoc risk cost function:

$$\rho(\boldsymbol{x}_k, \boldsymbol{\Theta}_{\boldsymbol{k}}) = \sum_{j=1}^{n_J} (\mu_1 + \mu_2 \chi_j V_w^2) e^{-\frac{1}{\zeta^2}(||\boldsymbol{c}_j - \boldsymbol{p}_k|| - r_j)^2} \tag{5.13}$$

The inputs to the function are the system state vector $\boldsymbol{x}_k$ and the control setting vector $\boldsymbol{\Theta}_{\boldsymbol{k}}$. For further information on the function inputs the reader is referred to article Blindheim et al. (2020). For the cost - function, μ_1 and μ_2 are positive tuning parameters. Furthermore $\chi_j = \max(0, \hat{\boldsymbol{i}}_j \cdot \hat{\boldsymbol{w}})$ describes the unit vector $\hat{\boldsymbol{i}}$ from the ship to each static obstacle center and $\hat{\boldsymbol{w}}$ is the unit wind direction vector. V_w is the wind velocity relative to the ship's velocity $\boldsymbol{c}_j$ and $\boldsymbol{r}_j$ are the centerpoint and radius of the static obstacles, respectivly.

Using exponential terms for the grounding risk costs will make the algorithm strongly dominate the other objectives in the cost function, heavily favoring staying safe from grounding obstacles. This formulation is not explicit such that the algorithm acknowledge that grounding risk may still be evaluated even if the risk of grounding is very high. One could argue that this is one of the safest way to implement anti-grounding. In certain situations this could affect performance both when it comes to tracking desired speed and course, which is necessary.

An illustration of the concept is shown in figure 5.3. Here the distance from own-ship to the static obstacle center is marked in addition to the obstacle centerpoint c_j and obstacle radius r_j

PSB-MPC cost function formulation

For implementation in the PSB-MPC framework a modification of 5.7.3 were carried out:

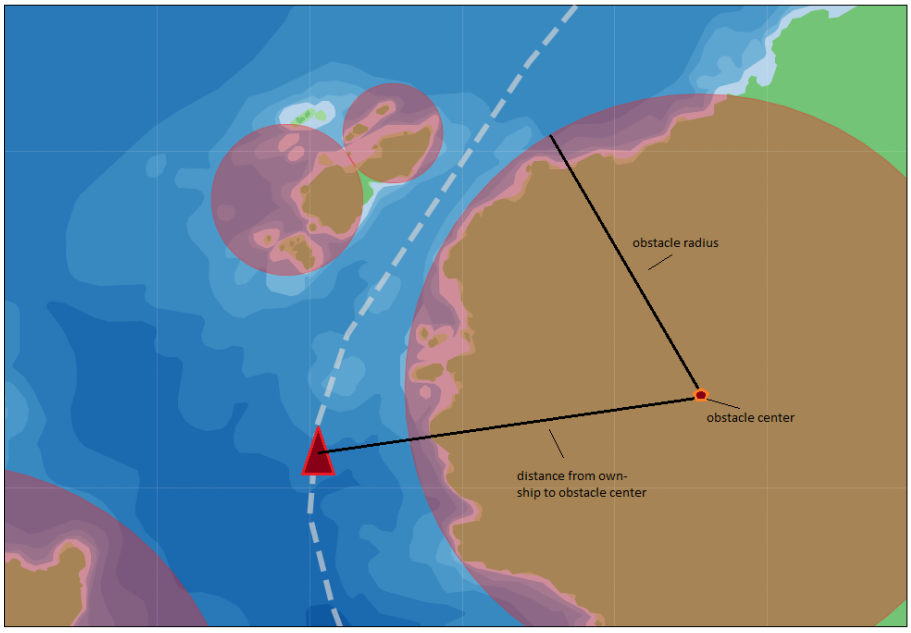
$$g(\cdot) = \frac{1}{n_M n_{ST}} \sum_{ts=1}^{n_{ts}} \sum_{M=1}^{n_M} \sum_{j=1}^{n_j} (\mu_1 + \mu_2 \chi_j V_w^2) e^{-\frac{1}{\zeta^2}(d_p + \omega)^2} \tag{5.14}$$

where

$$\omega = K_\omega \cdot n_{ts} \tag{5.15}$$

The index $ts = 1, 2, ..., n_{ts}$ is here defined as the own-ship prediction horizon trajectory samples. n_J is the number of static obstacles and n_M is the number of sequential avoidance maneuvers. μ_1 and μ_2 are positive tuning parameters. $\chi_j = \max(0, \hat{\boldsymbol{i}}_j \cdot \hat{\boldsymbol{w}})$ describes the unit vector $\hat{\boldsymbol{i}}$ from the ship to each static obstacle and $\hat{\boldsymbol{w}}$ is the unit wind direction vector. V_w is the wind velocity relative to the ship's velocity. Finally ω is a new tuning parameter included to hamper the relevancy of grounding far away in the prediction horizon. In essence: the further away in the prediction horizon, the larger numerical value of n_{st}.

Figure 5.3 Figure of ad-hoc exponential risk cost grounding function from Blindheim et al. (2020) Here the big red circles indicates the static obstacles. The red triangle is the own-ship with the dotted line as its trajectory.



Thus then increases the distance between the own-ship position and the edge of the static obstacle polygon at that instance.

Figure 5.4 shows an illustration of the PSB-MPC implementation of the anti-grounding algorithm. Here we see an illustratory predicted trajectory at three arbitrary point a, b and c on the prediction horizon trajectory n_{ts}. The black lines from the ships toward the edge of the polygons illustrate the distance d_p in equation 5.7.3.

Here, the calculated cost is summed over the prediction horizon and the number of static obstacles n_{st} and the number of maneuvers. The distance calculation from own-ship to the static obstacle are calculated as the distance from own-ship to the perimeter of the polygon making up the static obstacle. Furthermore, as mention in section 5.6.2 we manipulate the static obstacles set to contain only the static obstacles in a certain perimeter of the pre-defined own-ship path. This is done to prevent unnecessary operations, as the nested loops are heavy on computational power.

5.7.4 The control deviation cost

The control deviation cost: $f(\cdot)$ is the cost associated with keeping nominal speed and course and are formulated more spesifically as:

$$f(u^l_{m,M}, u^l_{m,M-1}, \chi^l_{m,M}, \chi^l_{m,M-1}) = k_u(1 - u^l_{m,M}) + k_\chi \chi^l_{m,M} \tag{5.16}$$

$$+ \delta_u(u^l_{m,M} - u^l_{m,M-1}) + \delta_\chi(\chi^l_{m,M} - \chi^l_{m,M-1}) \tag{5.17}$$

where k_u and k_χ are tuning parameters > 0 for keeping nominal speed and course and δ_u and δ_χ are penalty functions. Thus we make sure that the cost-function take account for unnecessary offsets in nominal states.

The course penalty function δ_χ,δ_u and the parameter k_χ are furthermore defined as:

$$k_\chi = \begin{cases} K_{chi_{strb}} \cdot \chi^2, & \text{if } \chi > 0 \\ K_{chi_{port}} \cdot \chi^2, & \text{otherwise} \end{cases} \tag{5.18}$$

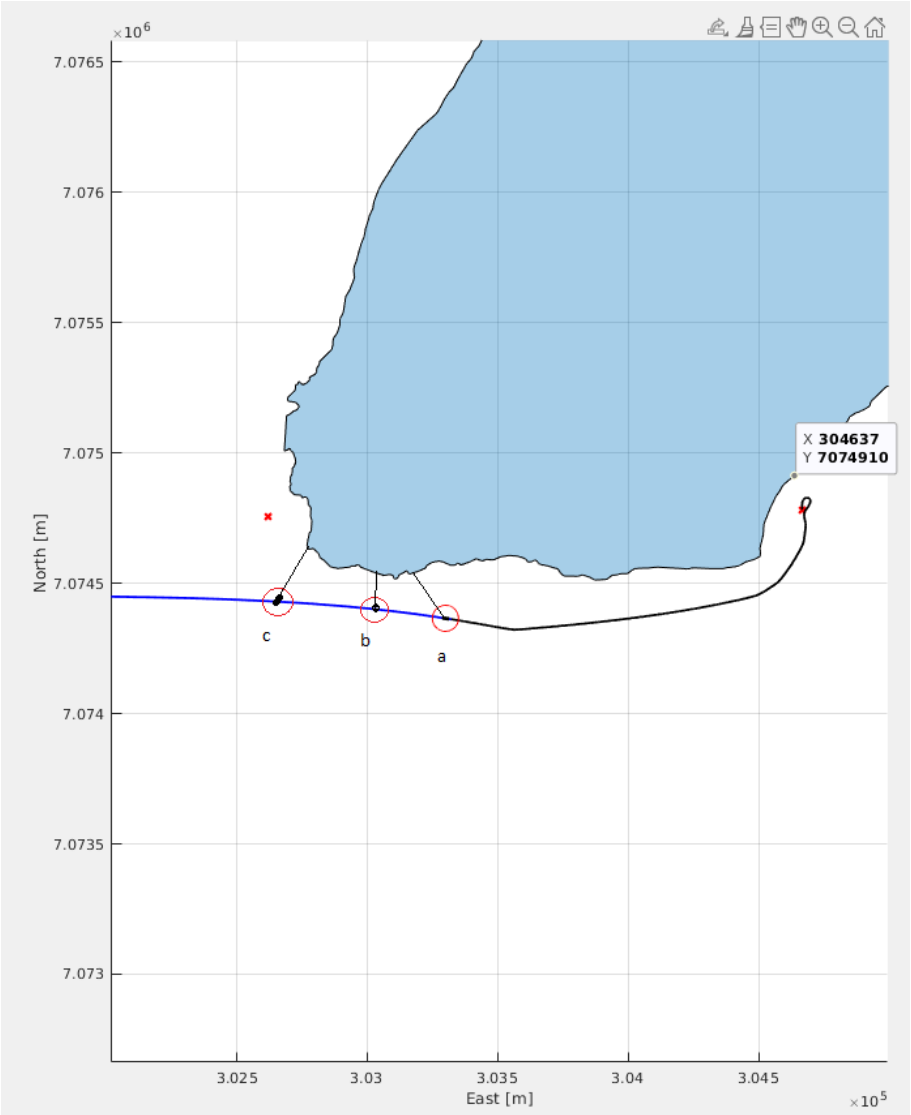
$$\delta_\chi = \begin{cases} K_{chi_{strb}} \cdot |\chi^l_{m,M} - \chi^l_{m,M-1}|^2, & \text{if } \chi > 0 \\ K_{chi_{port}} \cdot |\chi^l_{m,M} - \chi^l_{m,M-1}|^2,, & \text{otherwise} \end{cases} \tag{5.19}$$

$$\delta_u = K_{du} \cdot |u^l_{m,M} - u^l_{m,M-1}| \tag{5.20}$$

$$\tag{5.21}$$

which is to ensure that a favoring of maneuvering to starboard side is done. This is with respect to COLREGS.

Figure 5.4 Figure of ad-hoc exponential risk cost grounding function from Blindheim et al. (2020) Here the big red circles indicates the static obstacles. The red triangle is the own-ship with dotted line its trajectory.



5.7.5 The maneuvering-change chattering cost

When the number of maneuvers is greater than one, the last cost-function term to be introduced goes active. This is the maneuvering-change chattering cost, which aims to penalize chattering behaviour in course throughout the prediction horizon. The term is implemented as follows:

$$h(t_M^l - t_{M-1}^l, \chi_{m,M}^l, \chi_{m,M-1}^l) = \begin{cases} K_{sgn} e^{-\frac{t}{T_{sgn}}}, & \text{if } sign(\chi_{m,M}^l) \neq sign(\chi_{m,M-1}^l) \\ 0, & \text{otherwise} \end{cases} \tag{5.22}$$

Note that this term is inactive when simulating with $n_M = 1$.

Chapter 6

System Simulation

6.1 Introduction

This section contains a simulation study of the PSB-MPC system with anti-collision and anti-grounding. The simulation is mainly focused on testing the anti-grounding addition to the algorithm with gradual increase in complexity. Increased complexity refers to different maneuvering situations with the addition of Multiple Sequential Avoidance Maneuvering and ad-hoc.

6.2 Software and setup

The main part of the code are written in c++ on Ubuntu 20.04.1 LTS with a compilation scheme made through CMAKE 3.16.3 with GNU 9.3.0 . Plotting are done with matlab engine library ”engine.h,enginge.c” in MATLAB R2020b.

The anti-grounding algorithm is parsed and structured in python from the work of Blindheim et al. (2020). Furthermore, inclusion of generated shapefile geometries are manipulated by Boost polygon-library ”boost/lib.hpp”.

6.3 Simulation Parameters

This section includes all the main simulation - parameters used in simulation carried out in this thesis. The parameters are divided in general parameters and own-ship spesific parameters in their respective tables.

6.3.1 General Simulation Parameters

The general system parameters used in the simulation are shown in Table 5.1 and Table 6.1, in addition the own-ship dynamics are presented in Table 6.2. In simulations where changes are made to these parameters, the change is stated explicitly in the given section.

Table 6.1: Simulation parameters

Description	Parameter	Value
Simulation time	T_{sim}	1000
Number of Sequential Avoidance Maneuvers	n_M	2
Prediction horizon	T	240
Timestep	δ_t	0.5
Obstacle ship in range	d_{init}	1500
Obstacle ship is close	d_{close}	500
Obstacle ship safety distance	d_{safe}	25
Obstacle ship safety distance	d_{safe}	25
Chattering cost constant	K_{sgn}	25
Obstacle ship safety distance	d_{safe}	25
Chattering cost timestep	T_{sgn}	200
Grounding cost constant 1	μ_1	0.15
Grounding cost constant 2	μ_2	1
Grounding risk sensitivity constant	ζ	50
Cost of speed offset	K_u	100
Cost of course offset to starboard	$K_{chi_{strb}}$	1.3
Cost of course offset to port	$K_{chi_{port}}$	1.6
Cost of course offset to port	$K_{chi_{port}}$	1.6
Trajectory relative weight in grounding cost term	K_ω	50

6.3.2 Owh-ship simulation parameters

The current section is included to give a brief overview of our ownship type and parameters. Also added are the controller parameters for the LOS - guidance and simple PD-controller.

6.3.3 Obstacle-ship Simulation Parameters

The obstacle ship is implemented similar to the own-ship. It is a copy of the own-ship with regards to the parameters. For the simulations in this chapter the obstacle uncertainty is set to be constant for simplicity:

$$P_0 = \begin{bmatrix} 25 & 0 & 0 & \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 0.025 & 0 \\ 0 & 0 & 0 & 0.025 \end{bmatrix} \tag{6.1}$$

Furthermore the own-ship is simulated as a dumb component, meaning that it only travels between the given way-points, without taking grounding hazards, COLREGS nor obstacle-ships into account.

Table 6.2: Own-ship Simulation parameters

Description	Parameter	Value
Distance from rudder to CG	l_r	4.0
Distance from rudder to CG	A	5
Distance from rudder to CG	B	5
Distance from rudder to CG	C	1.5
Distance from rudder to CG	D	1.5
Own-ship mass	m	3980
Own-ship moment of intertia about z-axis	I_z	19703
Added mass	$X_{\dot{u}}$	0
Added mass	$N_{\dot{v}}$	0
Added mass	$Y_{\dot{v}}$	0
Added mass	$Y_{\dot{r}}$	0
Added mass	$N_{\dot{r}}$	0
Linear damping	X_u	-50.0
Linear damping	Y_v	-200
Linear damping	Y_r	0
Linear damping	N_v	0
Linear damping	N_r	-1281
Maximum integral correction in LOS guidance	$e_{int_{max}}$	$\frac{20\pi}{180}$
Waypoint acceptance radius	R_a	20.0
LOS Lookahead Distance	LOS_{LD}	500.0
LOS integral gain	LOS_{K_i}	0
PD-Controller proportional gain term	K_{p_u}	1
PD-Controller proportional gain term	K_{p_ψ}	5
PD-Controller derivative gain term	K_{d_ψ}	1
PD-Controller proportional term	K_{p_r}	8

6.4 Proof of concept simulations

This thesis focuses on including anti-grounding into the PSB-MPC scheme, and thus a natural first step is to simulate the system without any obstacle ships, and disregarding the COLREGS-term and the collision cost-term in the cost function in section 6.7.3. First simulations are carried out without the use of wind. i.e., setting $V_w = 0$. This can be done without loss of generality as this only affects the magnitude of the anti-grounding cost in the target wind direction.

6.4.1 Simple anti-grounding test with no obstacle ships

A bare-bones simulation with only chattering-cost, control deviation cost and grounding cost can be shown in Figure 6.3, 6.4 and 6.5. Here we see 3 figures at 3 different time-steps in the simulation of our own-ship sailing from N,E = (7074782,304666) to N,E = (7074757,302622). Which is the areas outside Ytterøya at The fjord of Trondheim in Trøndelag, showcased with a satellite photo in figure 6.1. In addition to simulating without the COLREGS-term and the collision cost-term we set $n_m = 1$, meaning we simulate without multiple sequential avoidance maneuvers. Hence the maneuvering-change chattering cost-term should also not contribute to the calculated cost, and we obtain a total of $n_\chi \cdot n_u \cdot n_M = 3 \cdot 13 \cdot 1 = 39$ control behaviours. Before simulation, the grounding cost-term is tuned such that the range of cost is similar to that of the control deviation -cost, and lastly we weight, with a tuning-parameter, the grounding cost by 95% such that the dominating factor in this bare bones simulation is the grounding cost. This is done to provide insight into the behaviour of the grounding cost-term.

As we see in Figure 6.2 the first iteration of the simulation the anti-grounding term is by far the most dominant. The maneuvering-change chattering cost is 0 across the whole control behaviour leaf-node tree as expected since we use $n_m = 1$ sequential avoidance maneuvers. Furthermore we can easily see the change in speed offsets between the ranges 0-13, 13-26 and 26-39 in 6.2, which corresponds to keeping nominal speed, changing speed to 0.5 times current speed and halting.

As the ship's starting position is with its bow towards north, a -90 degree angle is pointing straight east. This can also be seen from 6.2, which corresponds to 0 contribution to the anti-grounding term.

6.4.2 Anti-grounding with no obstacle ships in challenging enviroments

Before testing the algorithm with obstacle ships, a more strict test with the addition of rocks and bridge pylons in narrow convoluted waters are carried out. Figures 6.7, 6.8 and 6.9 showcases a simulation in challenging enviroments outside Straumen in Inderøy. A satellite-photo of the area is also included in Figure 6.6. The ship path is set to be from the fjord-area outside Straumen through the strait and into the local small harbor.

In figure 6.7 We see the ship starting its path through the strait with its horizon pointing straight through the end-point location. This figure also showcases the strength of including the weighting term ω on the horizon.

Figure 6.1 Sattelitephoto of the waters around ytterøya at N,E = (7074782,304666)

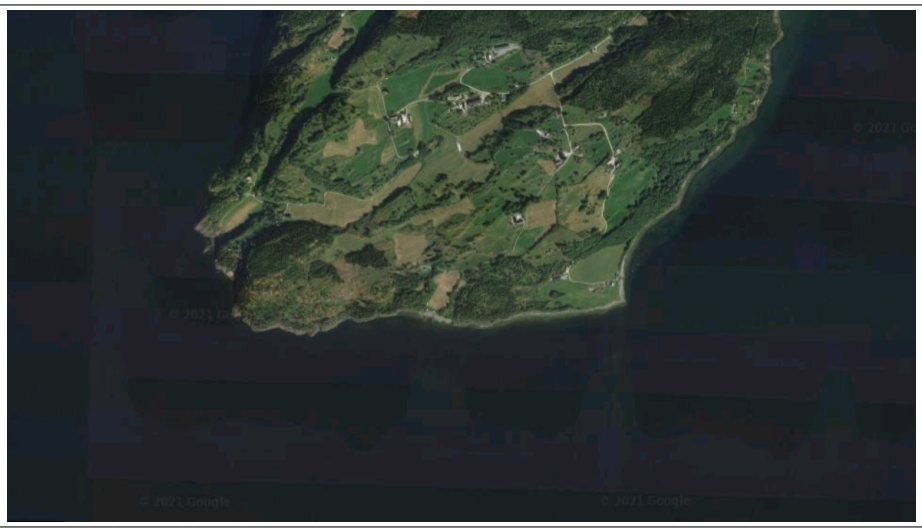


Figure 6.2 Plot of relative cost for each cost-term in the optimization-function over 1 iteration of the PSBMPC algorithm at T=0. The x-axis is in course offsets, and a change in propulsion commands are marked 1,0.5 and 0 respectivly.

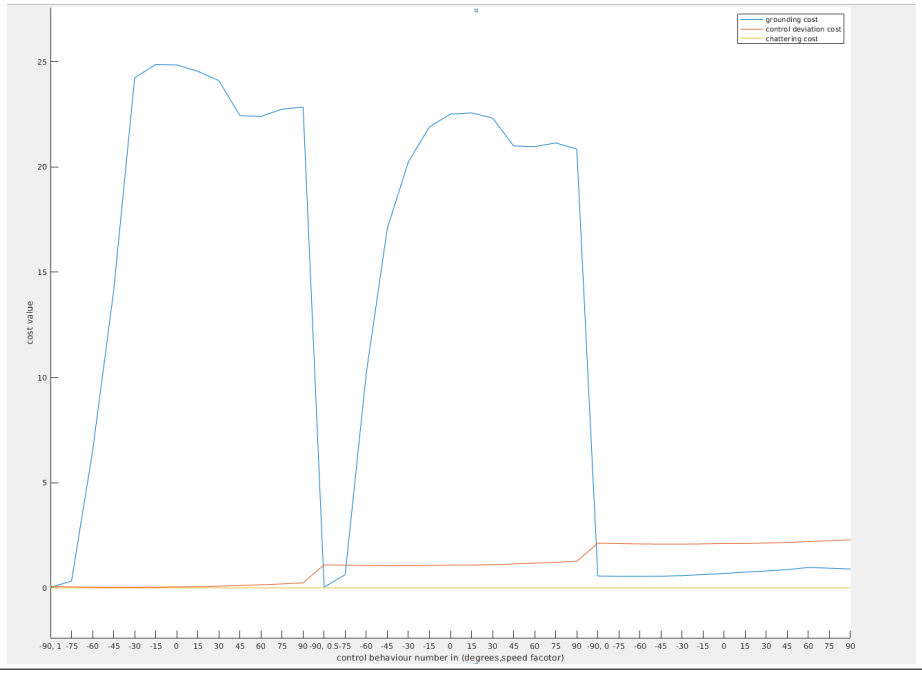


Figure 6.3 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, at T=50. Blue marked area is the land-area making up ”Ytterøya” at Coordinates approximatly Blue line is the predicted own-ship trajectory, black line is the followed trajectory and red circle is the own-ship safety distance.

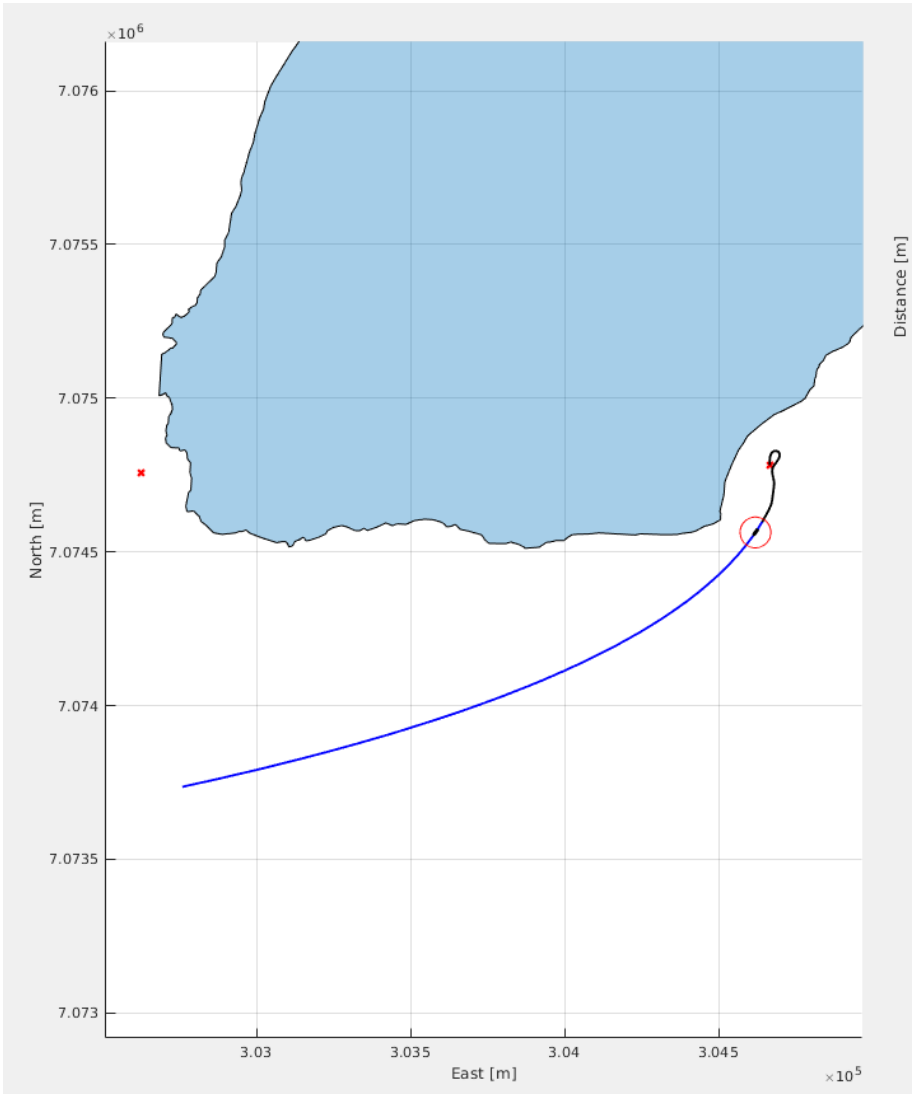


Figure 6.4 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, at T=300.

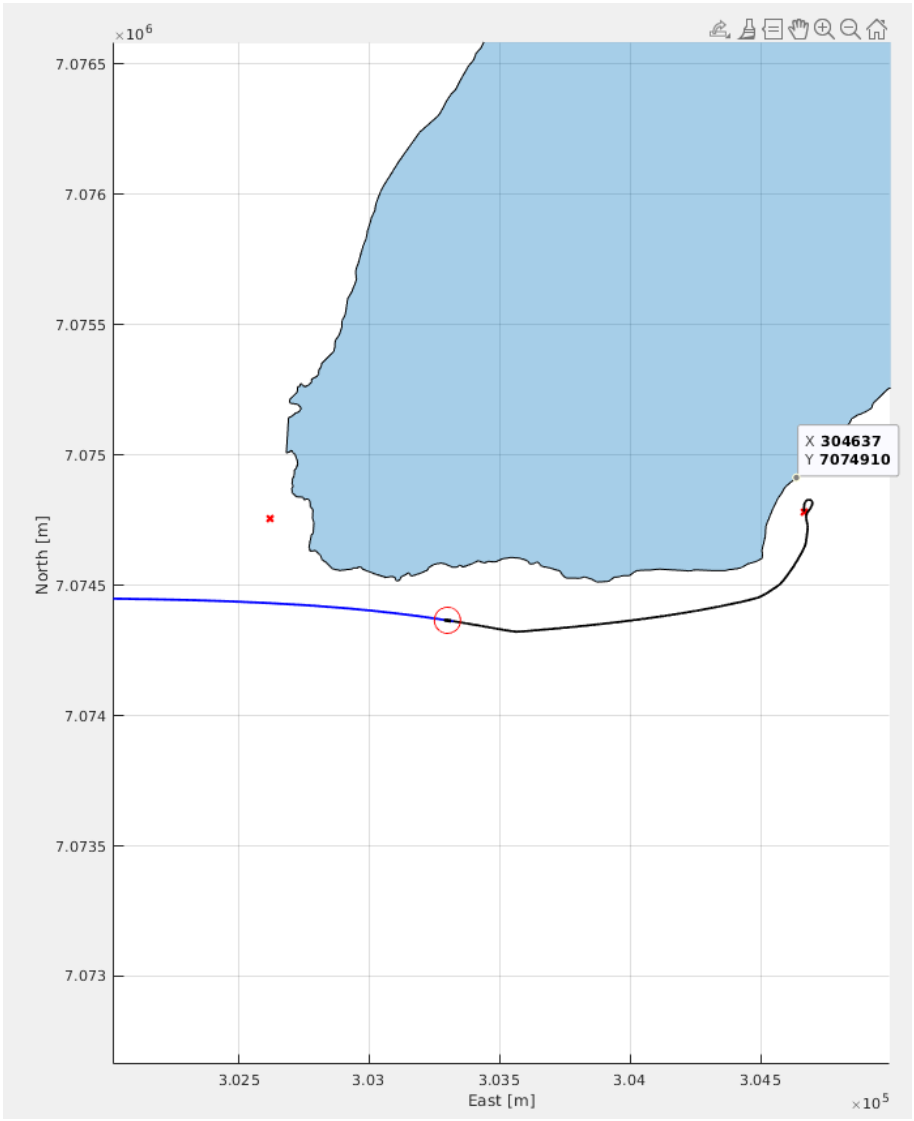


Figure 6.5 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, at T=400.

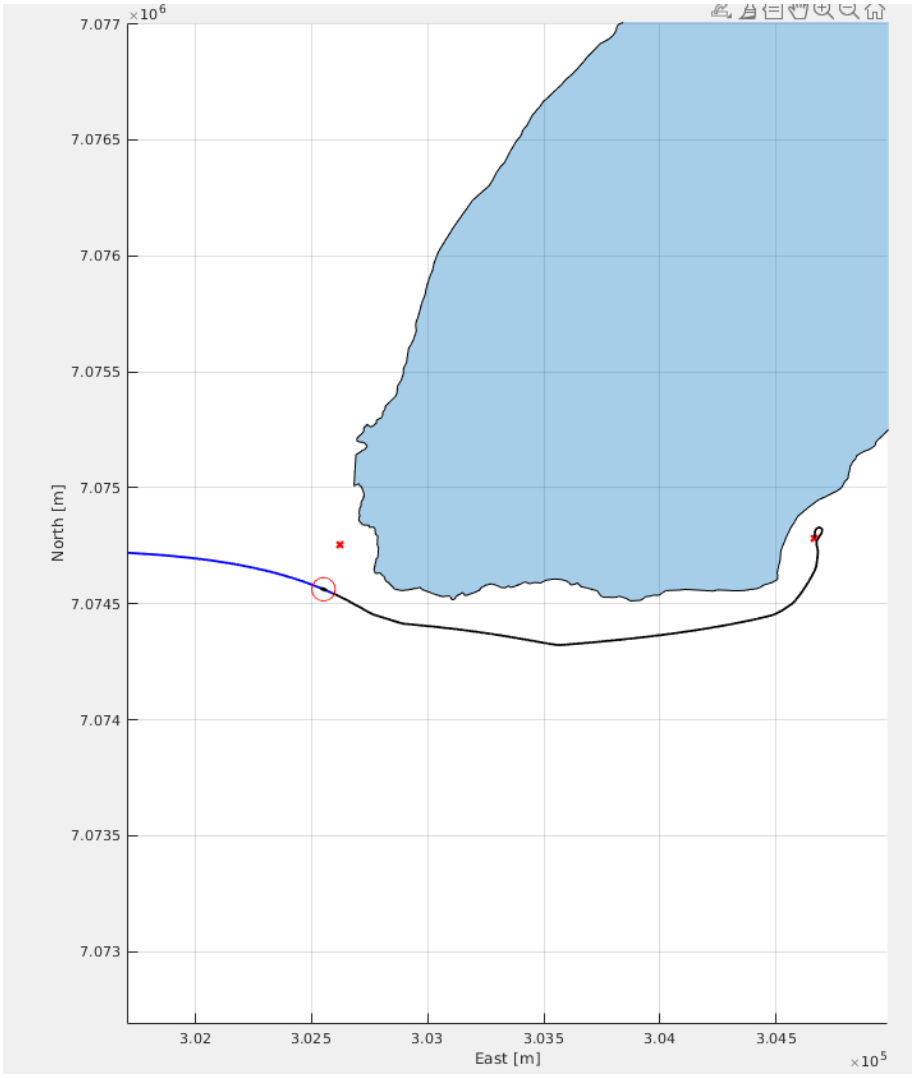


Figure 6.6 Google maps satellite photo of simulation location. Here we can see that the two ”rocks” in the middle of the strait in the simulation are bridge pylons

Carrying on, we see that as soon as our own-ship reaches the first large rock in figure 6.8. It makes a maneuver to port side, and carries on into the strait. Here it perfectly passes between the two bridge pylons, before it in figure 6.10 reaches it end-point destination outside the local small harbor. Lastly in figure 6.11 we see the whole driven path through the strait. When carrying out the same challenging environment simulation with $K_\omega = 0$ we see the own-ship completely avoiding the strait area and fails to reach the end-point destination. This is included in figures 6.12 and figure 6.13.

6.5 Complex simulation cases

AS we saw in the previous section our MPC anti-grounding system performed in challenging simulation environments. The next logical step is to test the whole PSB-MPC system with COLAV and anti-grounding.

6.5.1 COLAV and anti-grounding with one obstacle ship in challenging envioments

A simulation of the whole PSB-MPC system with all cost-terms included were carried out, and are illustrated in Figures 6.14, 6.15 and 6.16.

In Figure 6.14 we see our own-ship avoding the obstacle-ship by turning starbord of the large rock, too avoid both the static obstacle and the obstacle-ship.

Figure 6.7 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, at T=50. Showcasing ship horizon pointing directly at trajectory end-point.

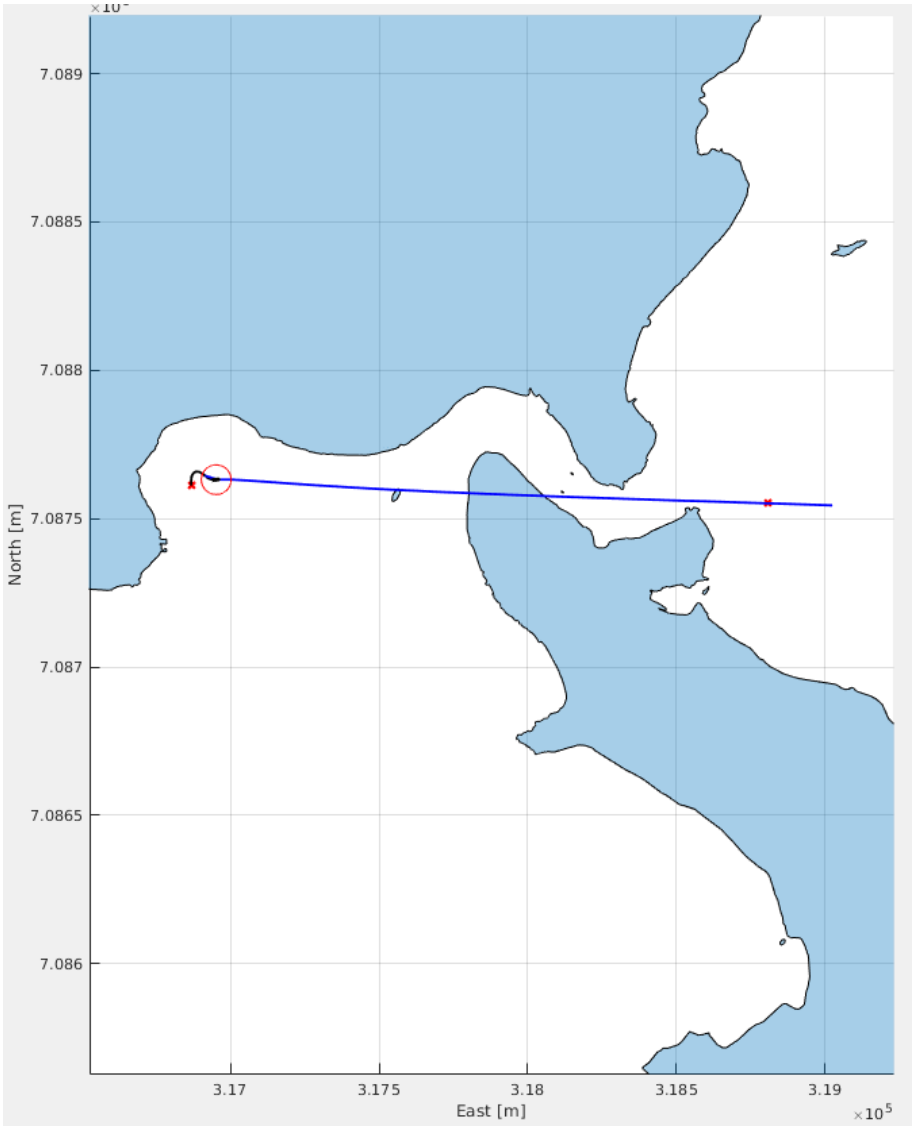


Figure 6.8 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, near first large rock obstacle on path through strait outside Straumen, Inderøya.

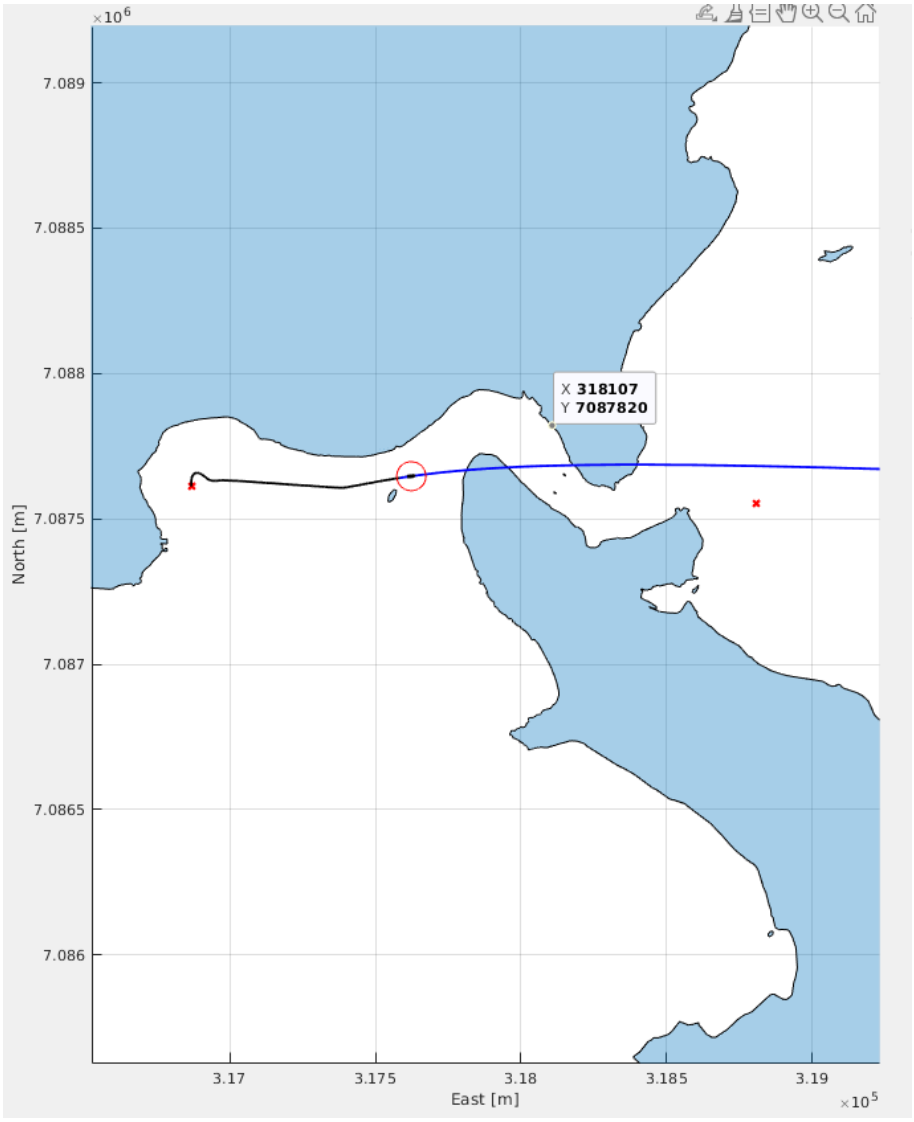


Figure 6.9 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, sailing in between two bridge polygons outside Straumen, Inderøya.

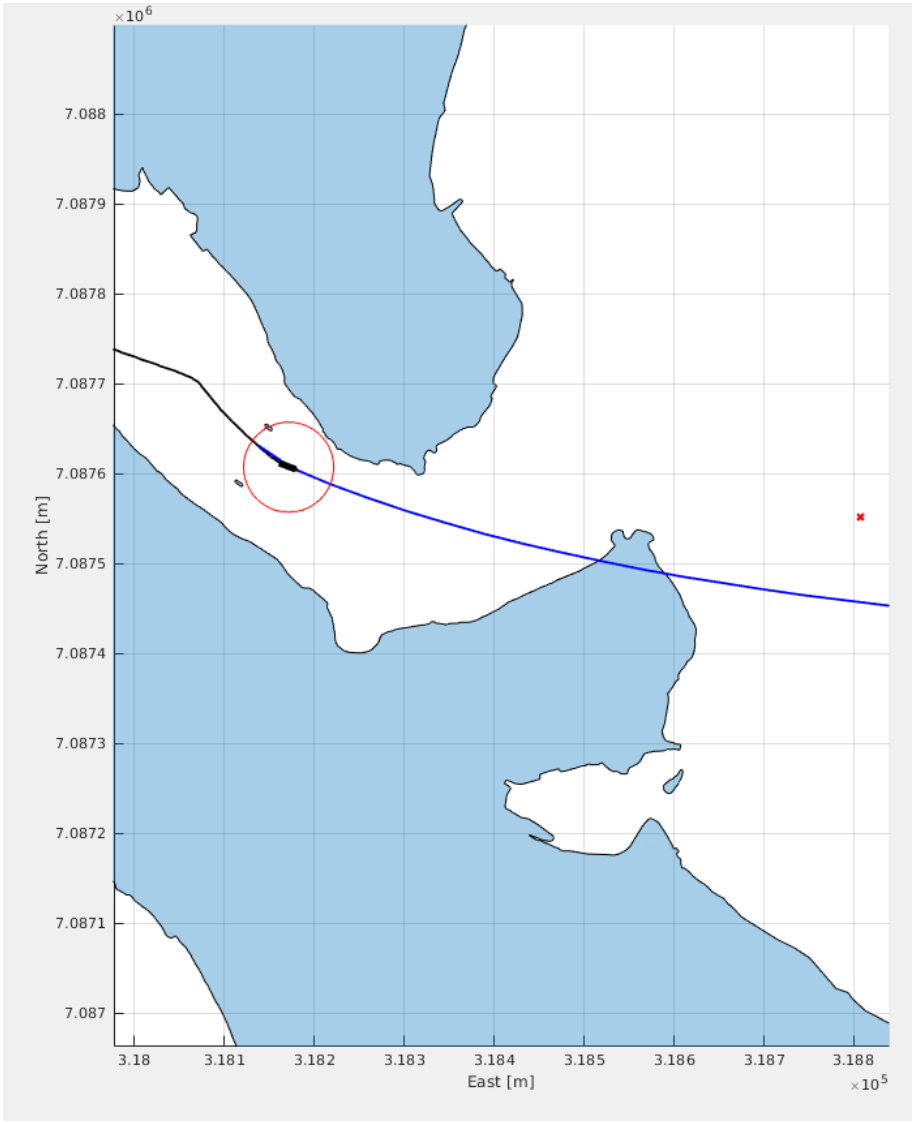


Figure 6.10 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, sailing in toward the waypoint outside Straumen, Inderøya.

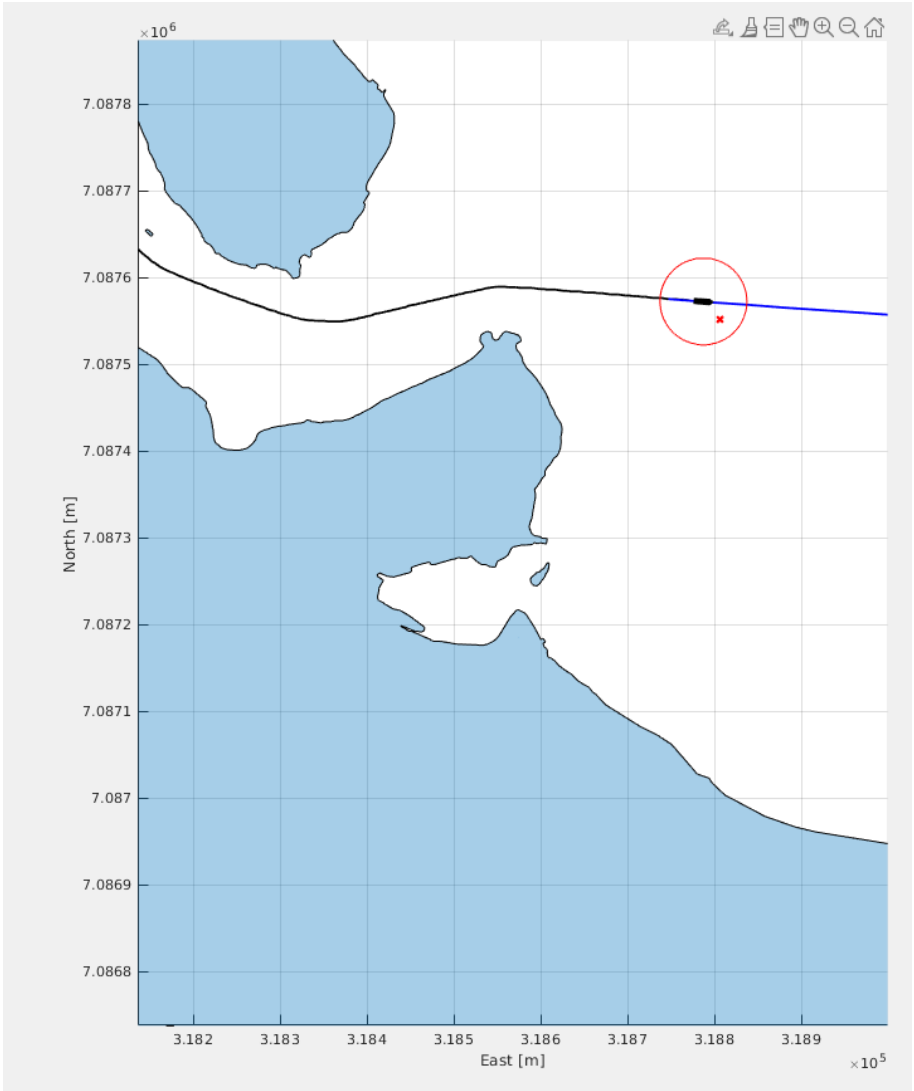


Figure 6.11 Illustration of ship-path using anti-grounding cost-term in PSB-MPC algorithm, final driven route overview of the challenging enviroment simulation case.

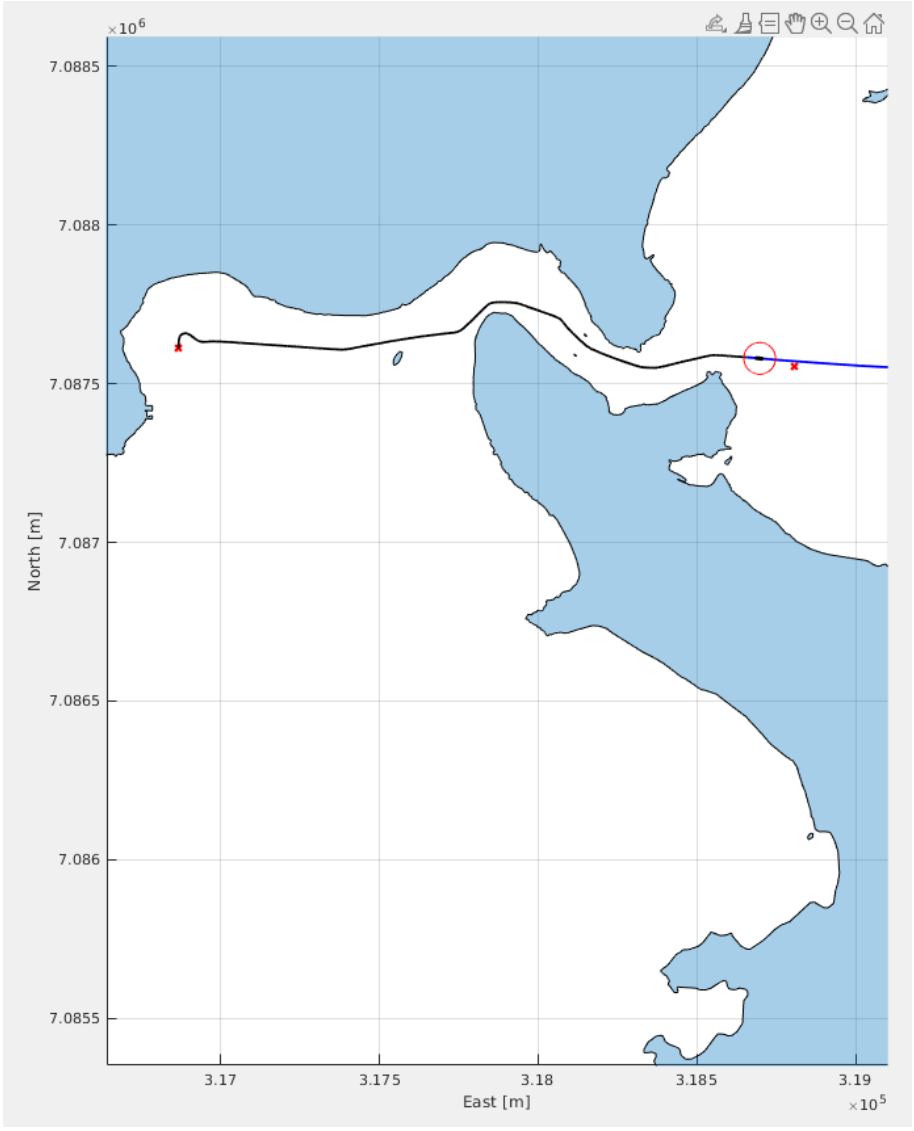


Figure 6.12 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, where $K_{=}0$. Here the ship completely avoids the strait as the calculated cost of grounding is too high without weighting down static obstacles far away.

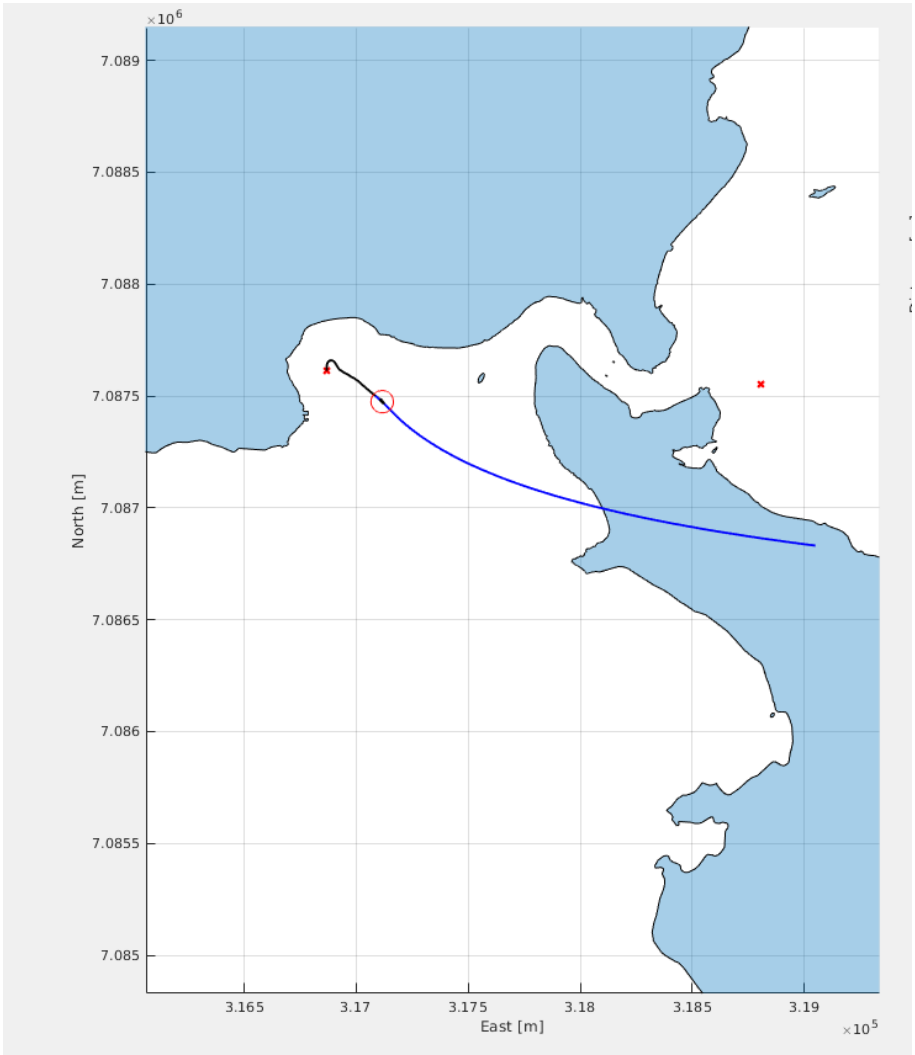
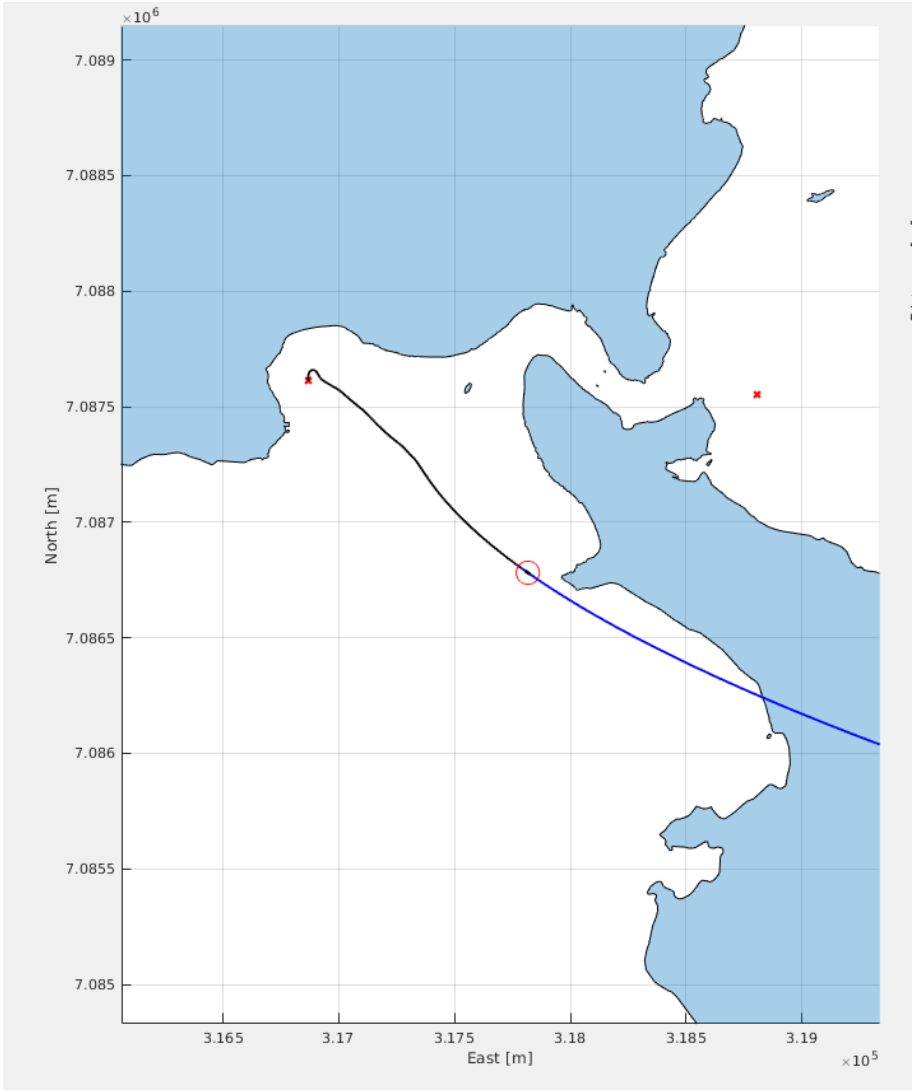


Figure 6.13 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, where $K_{=}0$. Here the ship completely avoids the strait as the calculated cost of grounding is too high without weighting down static obstacles far away.



Furthermore, in Figures 6.15 and figure 6.16 we see the ship making a large maneuver towards north to maneuver safely into the strait, while simultaneously avoiding the bridge pylons before it reaches its endpoint destination.

In Figure 6.17 we see the calculated costs during 1 iteration of the PSBMPC algorithm, early in the simulation. Here we can see that the grounding cost is negligible compared to the other costs. And as we Simulate with $n_m = 1$ the chattering cost is 0. Furthermore, in Figure 6.18, which is a illustration of the cost in the PSB-MPC algorithm right before we reach the large rock we see grounding cost has a larger effect on the optimal solution.

6.5.2 COLAV and anti-grounding with one obstacle ship in challenging environments, scenario:Overtaking

A simulation with the same parameters as in the previous section, but with a different COLAV - scenario, namely:"Overtaking", can be shown in figures 6.19, 6.20 and 6.21.

In figure 6.20 we see our own-ship and our obstacle-ship starting their drive mission heading north, before turning in on the right path toward east. Further, in figure 6.21 We see the own-ship safely surpass the obstacle ship on starboard side which is in accordance to COLREGS.

Lastly figures 6.22 and 6.23 are included for redundancy showing that the ship with the current configuration is able to pass the large rock, and through the strait with the two large bridge pylons. Note that the obstacle ship (green trajectory) is a dumb component and are driving in a straight line without stopping, even for land areas.

6.5.3 COLAV and anti-grounding with one obstacle ship in challenging environments, scenario:Head-On

A simulation with the same parameters as in the previous section, but with a different COLAV - scenario, namely:"Head-On scenario" is shown in figures 6.24, 6.25 and6.26.

We see this situation unfolds similar to the overtaking situation. The own-ship makes a maneuver to avoid the obstacle-ship, before it seamlessly drives through the strait.

Figure 6.14 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. The obstacle ship waypoints are marked with a purple cross, and it's trajectory are green.

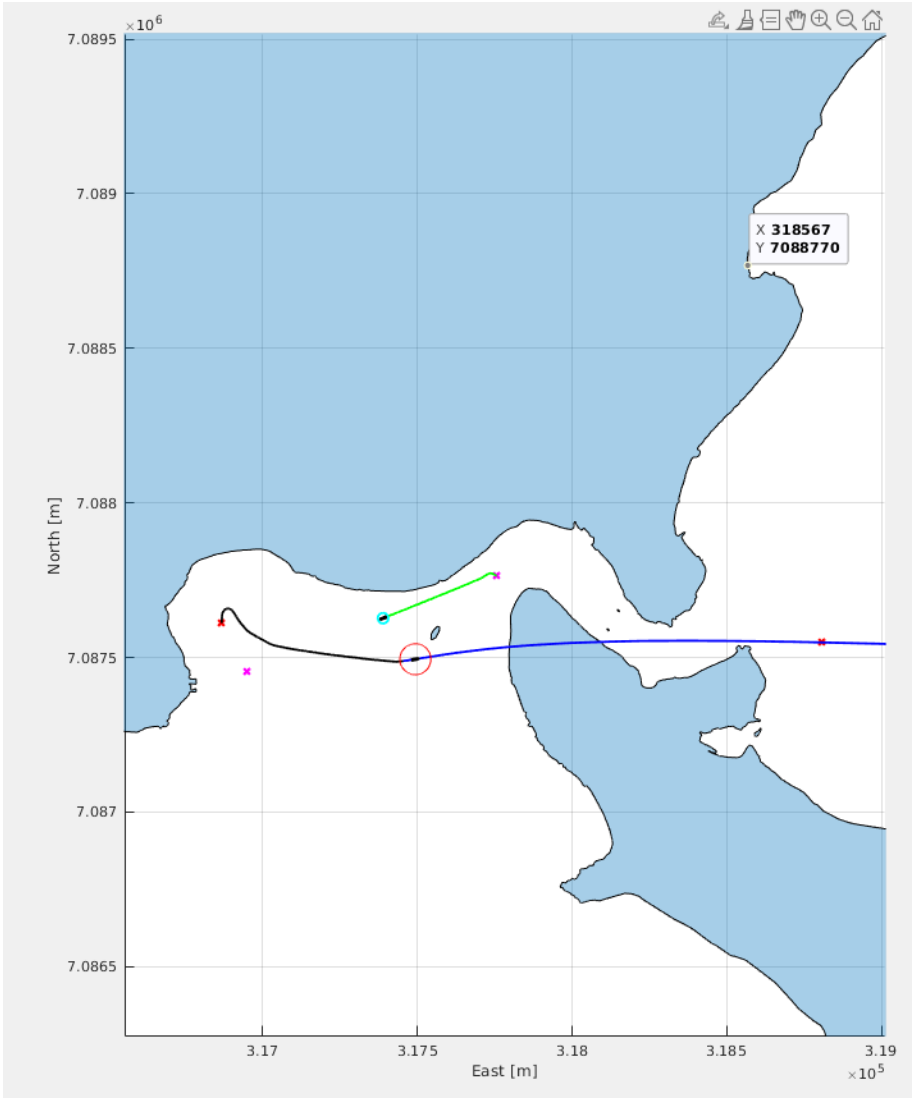


Figure 6.15 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship.

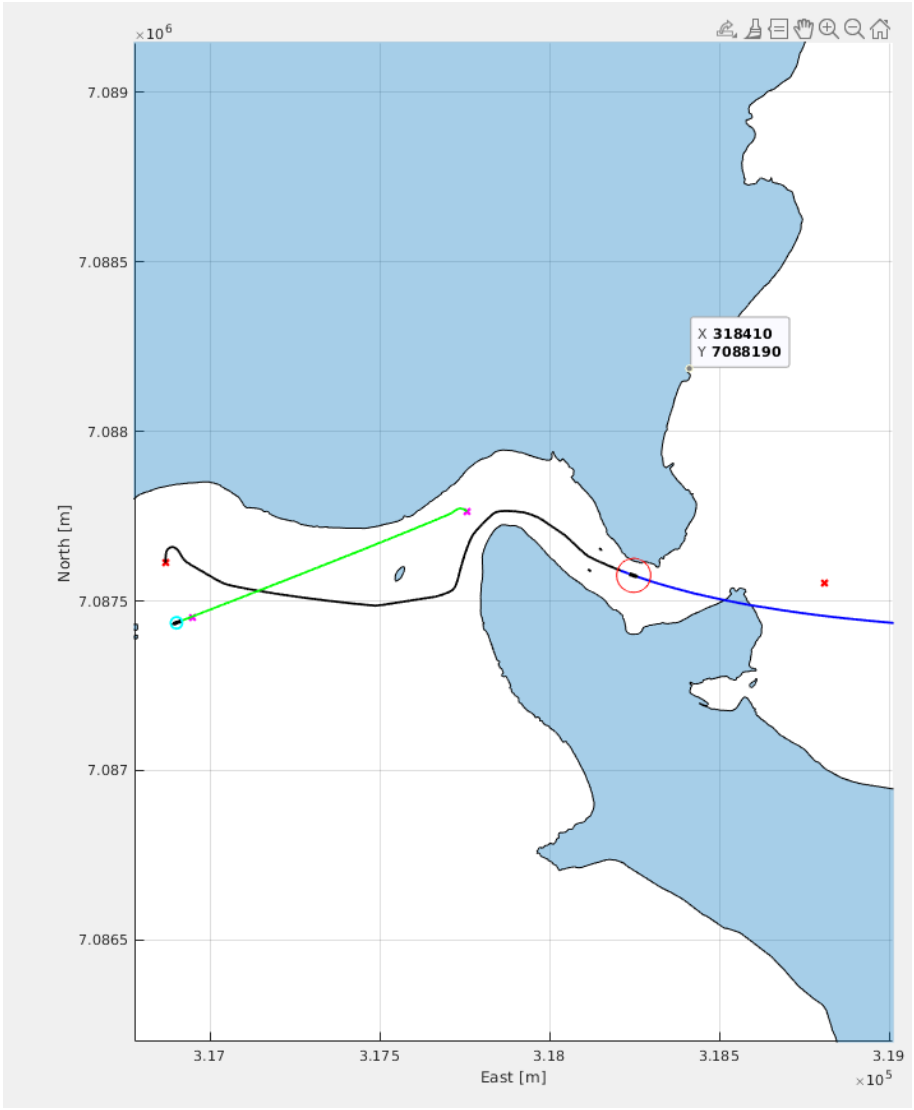


Figure 6.16 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship.

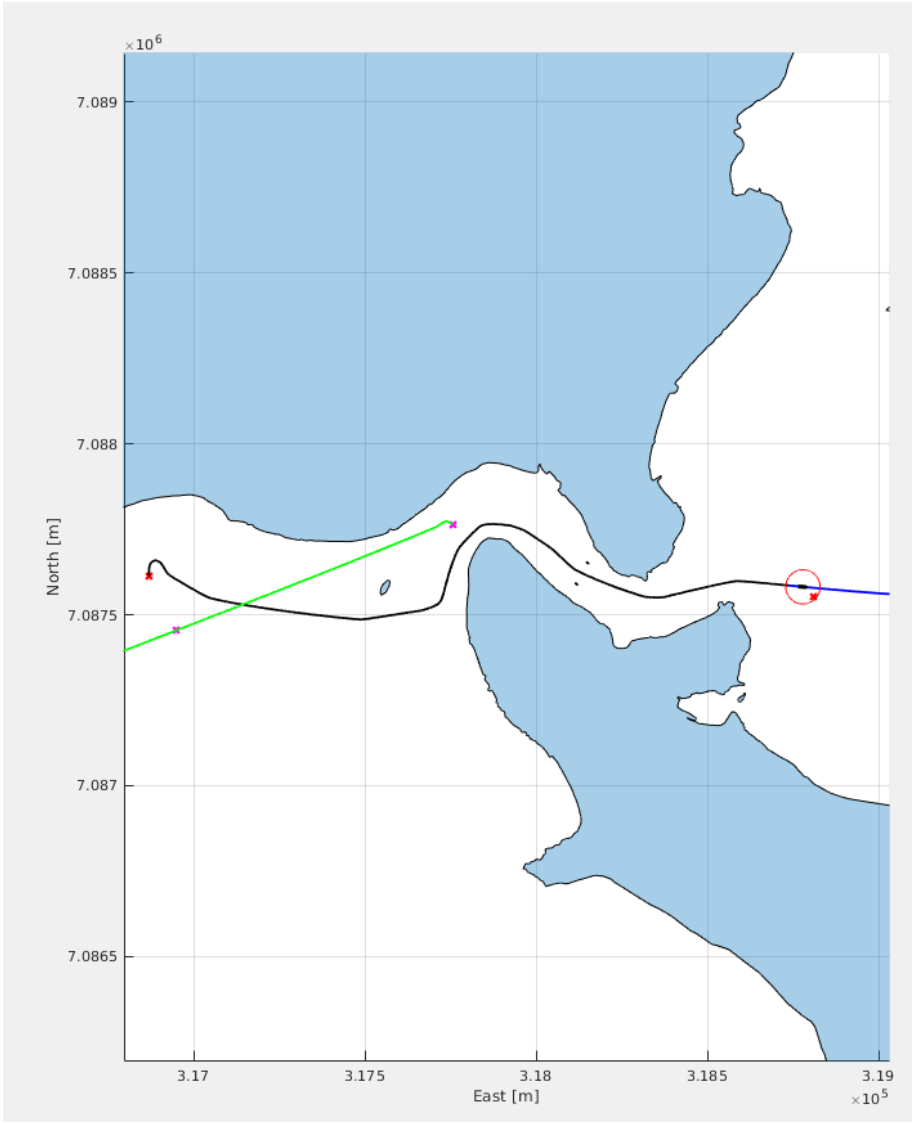


Figure 6.17 Illustration of the cost terms in cost-function at the start of the simulation

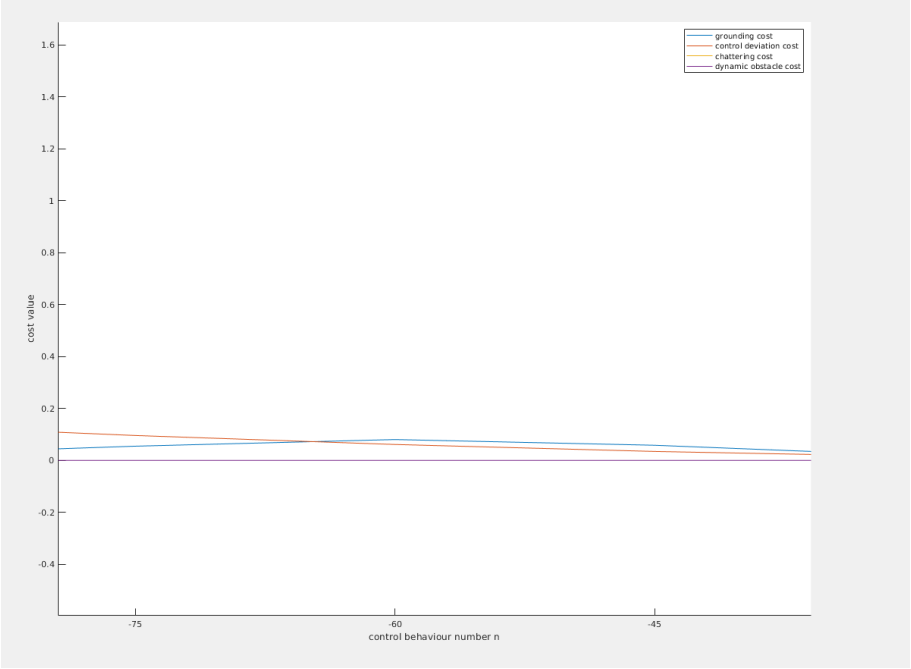


Figure 6.18 Illustration of the cost terms in cost-function right before meeting the large rock in the simulation.

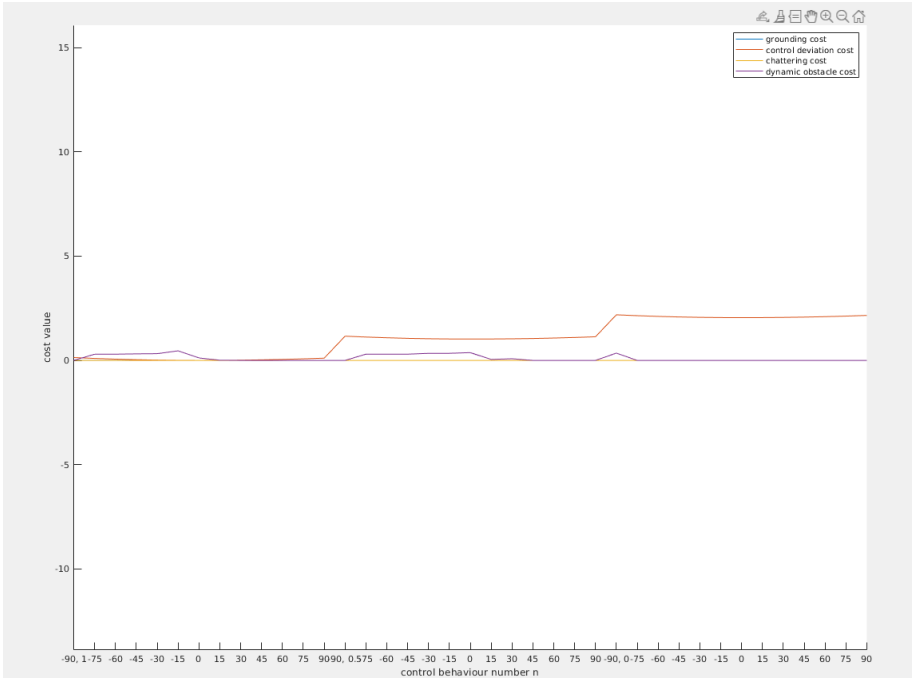


Figure 6.19 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on overtaking scenario. Simulation time T=0.

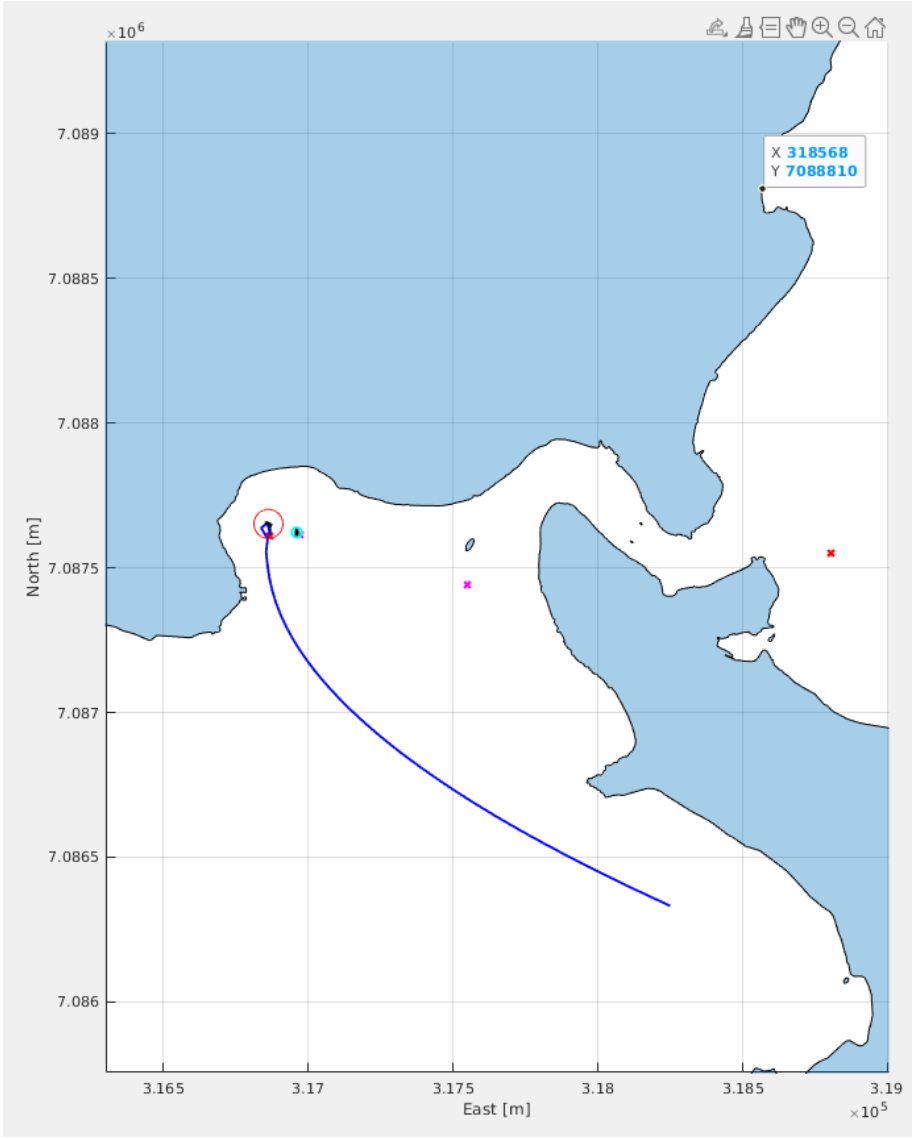


Figure 6.20 Overtaking scenario closeup. Simulation time T=0.

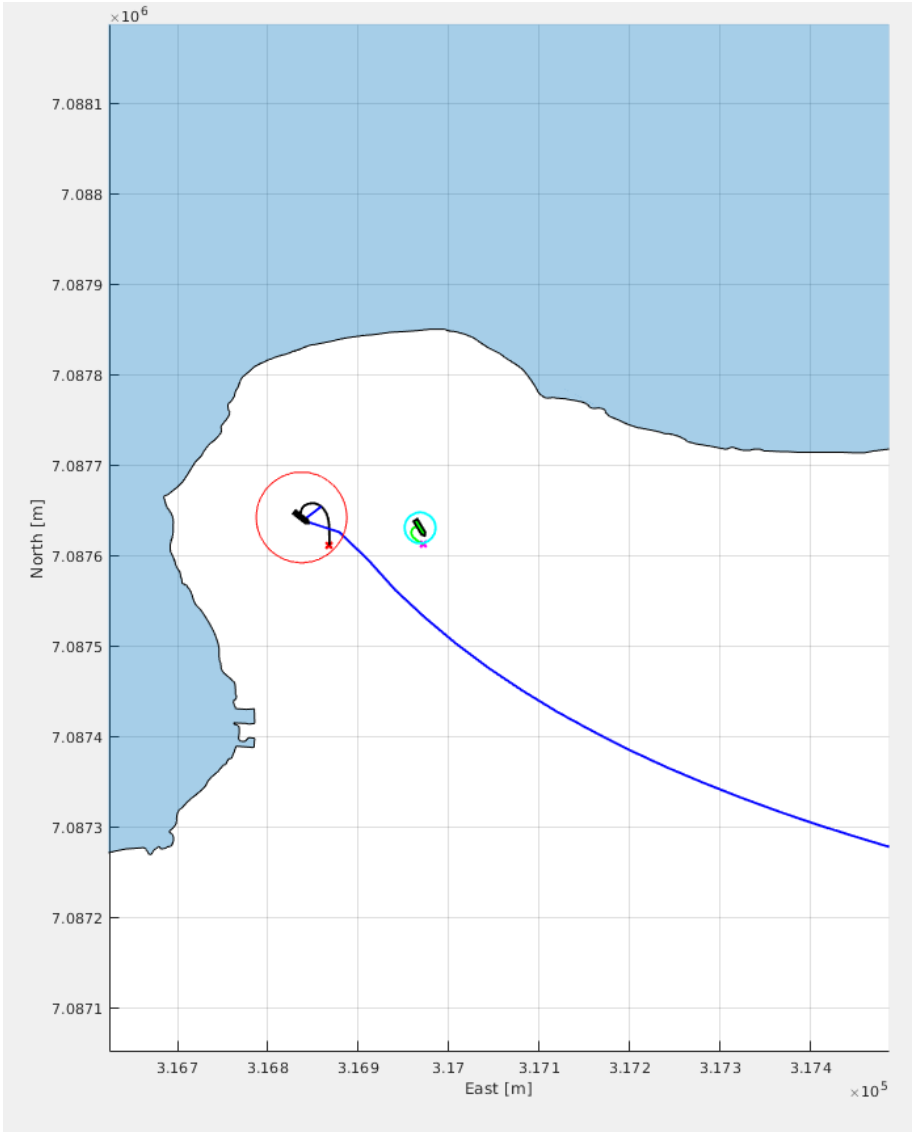


Figure 6.21 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on overtaking scenario. Simulation time T=100. Own-ship has safely driven past obstacle-ship.

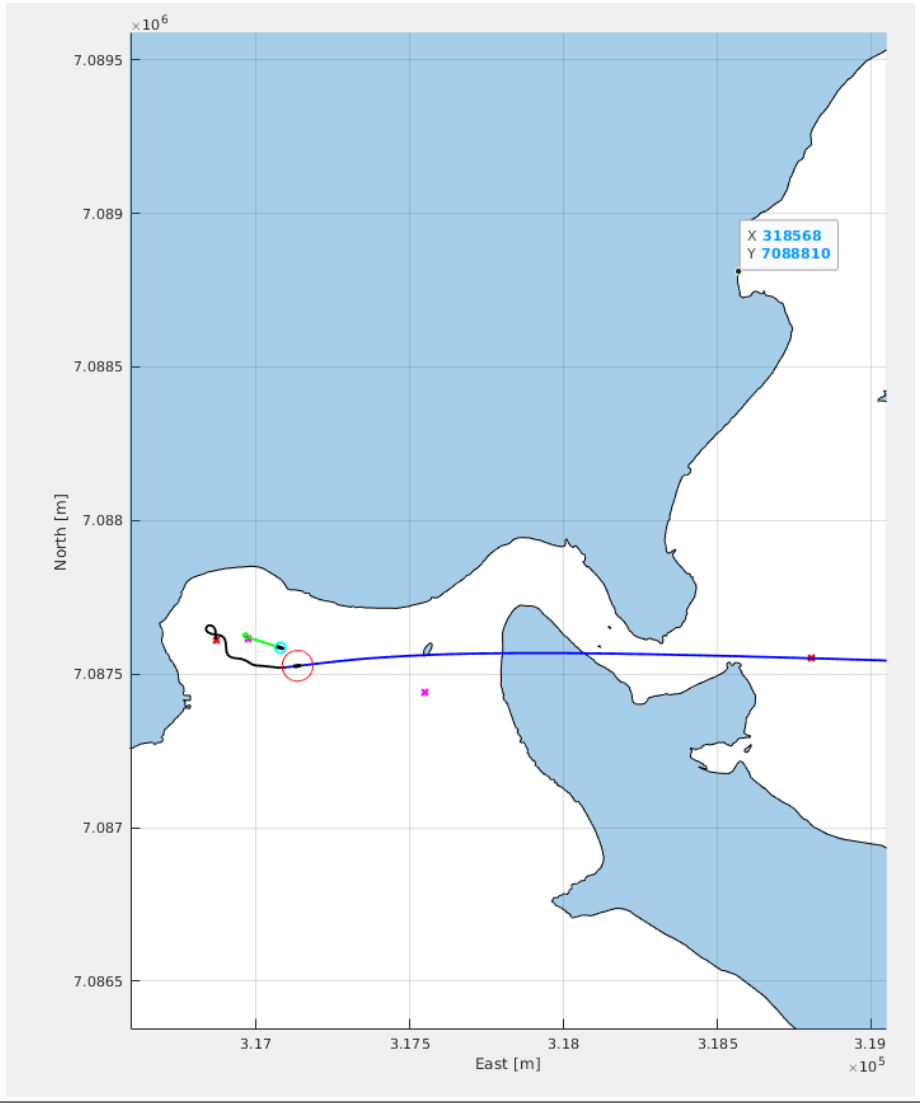


Figure 6.22 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on overtaking scenario. Simulation time T=100. Own-ship navigates past large rock.

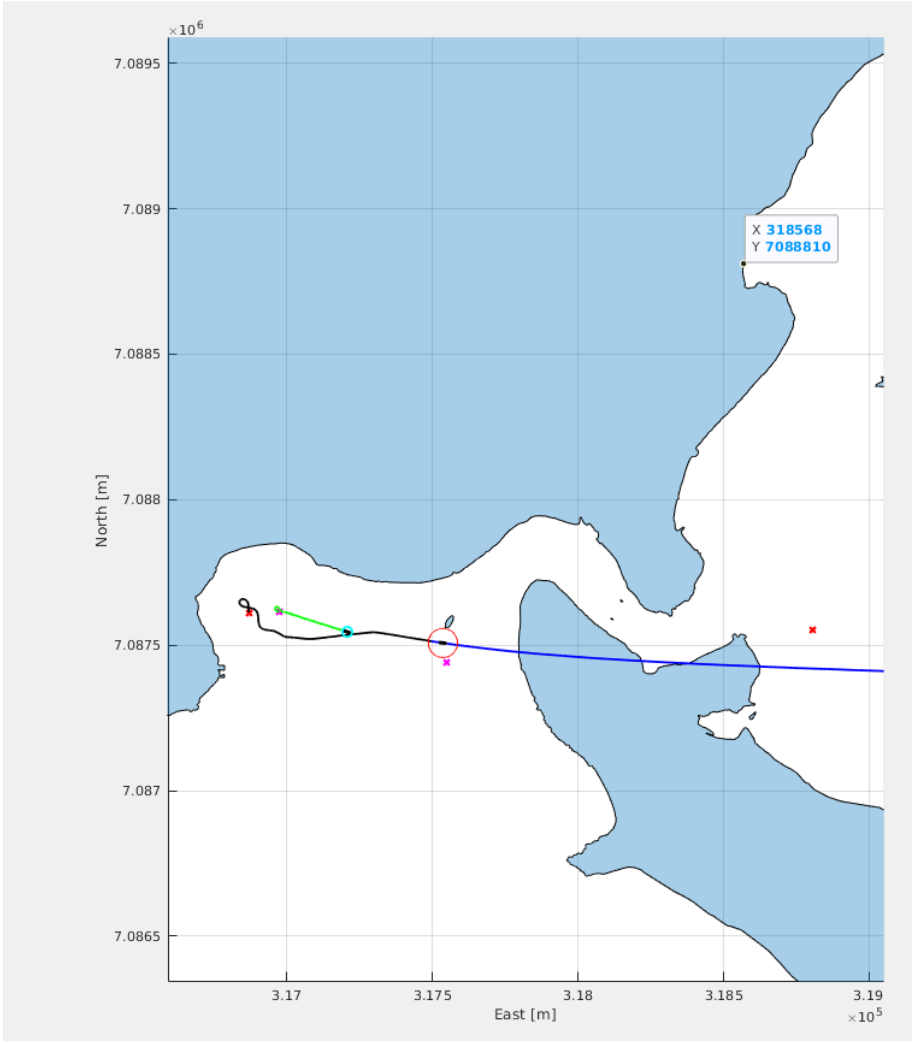


Figure 6.23 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on overtaking scenario. Simulation copmlete. Own-ship navigates past end-point.

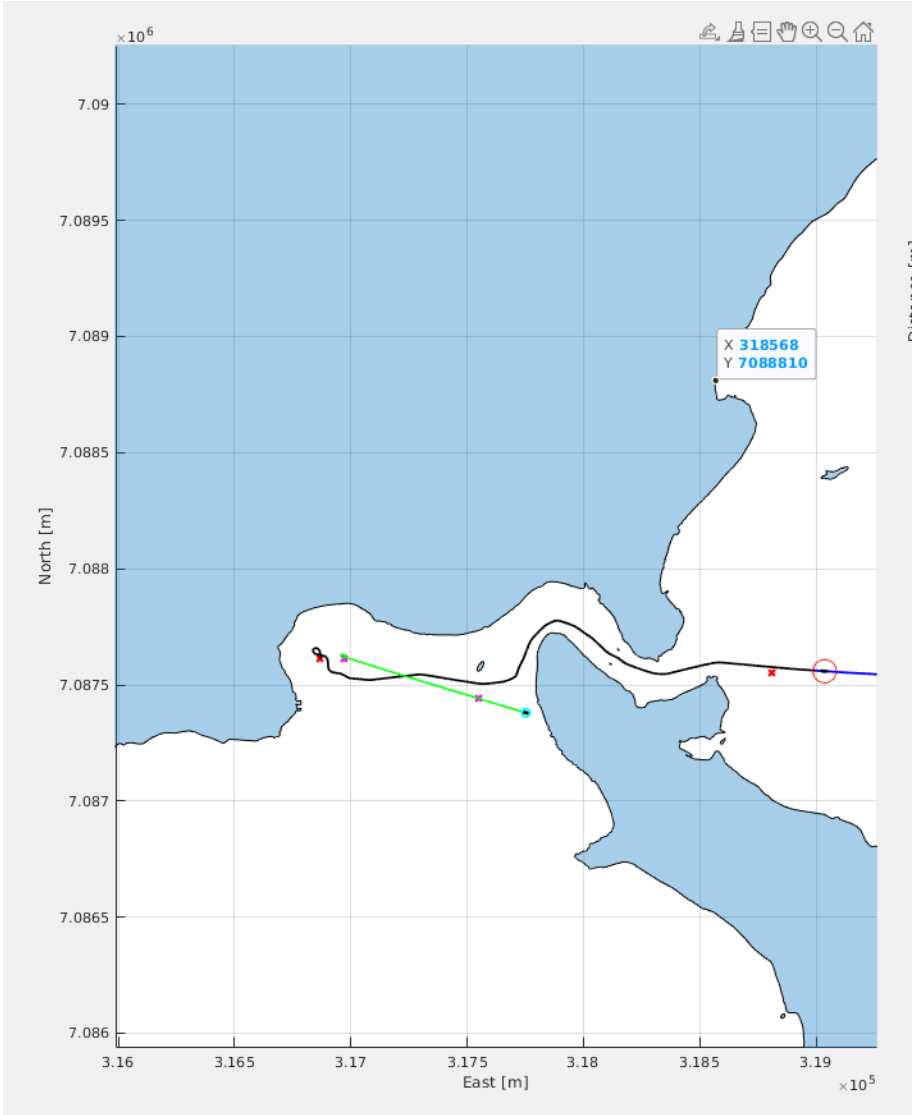


Figure 6.24 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on Head-On scenario. T=0. Own-ship navigates past end-point.

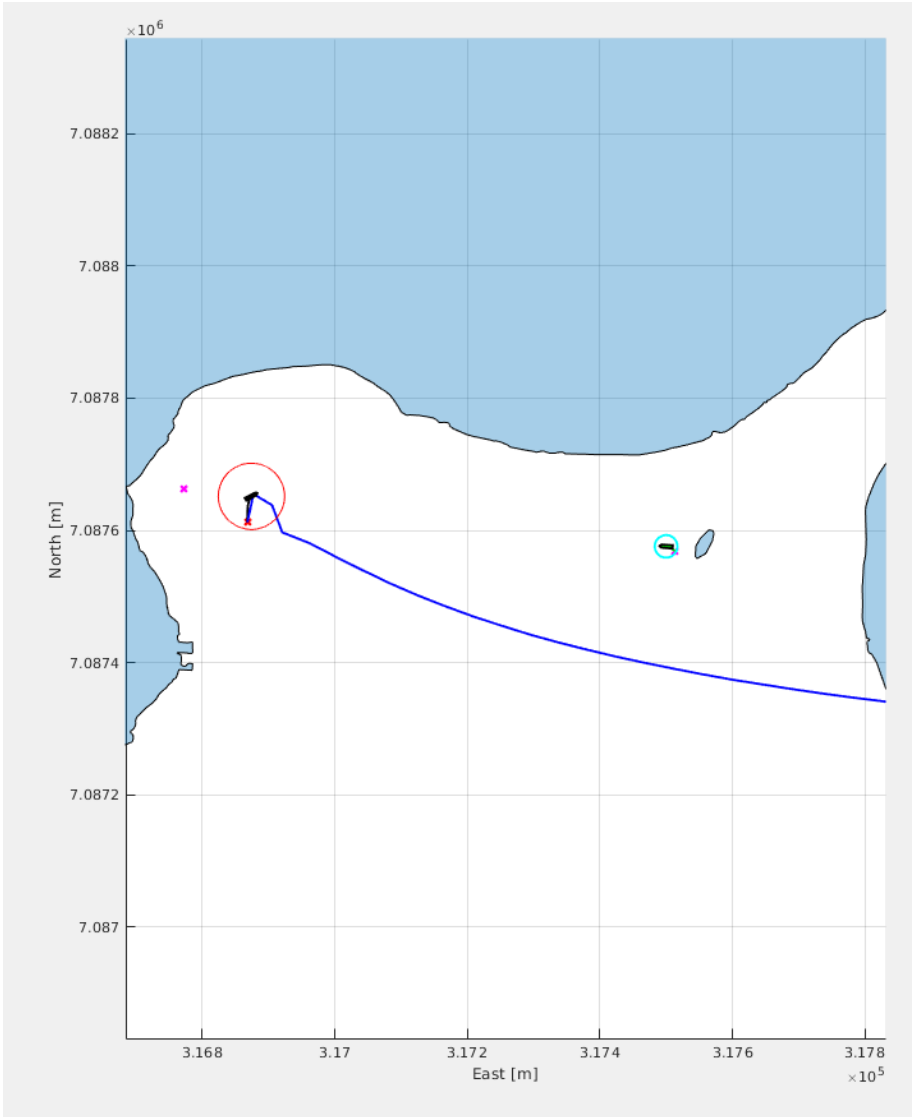


Figure 6.25 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on Head-On scenario. In the middle of Head-On situation. Own-ship navigates past end-point.

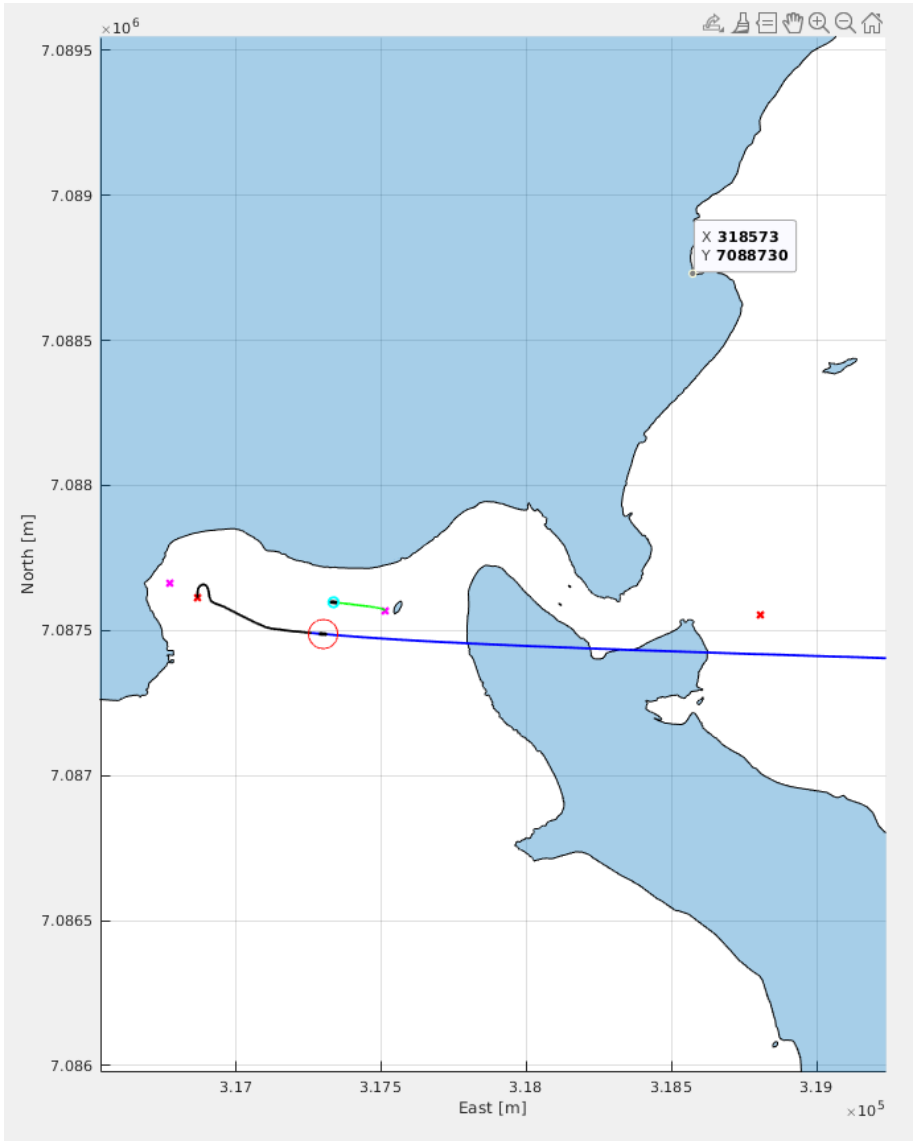
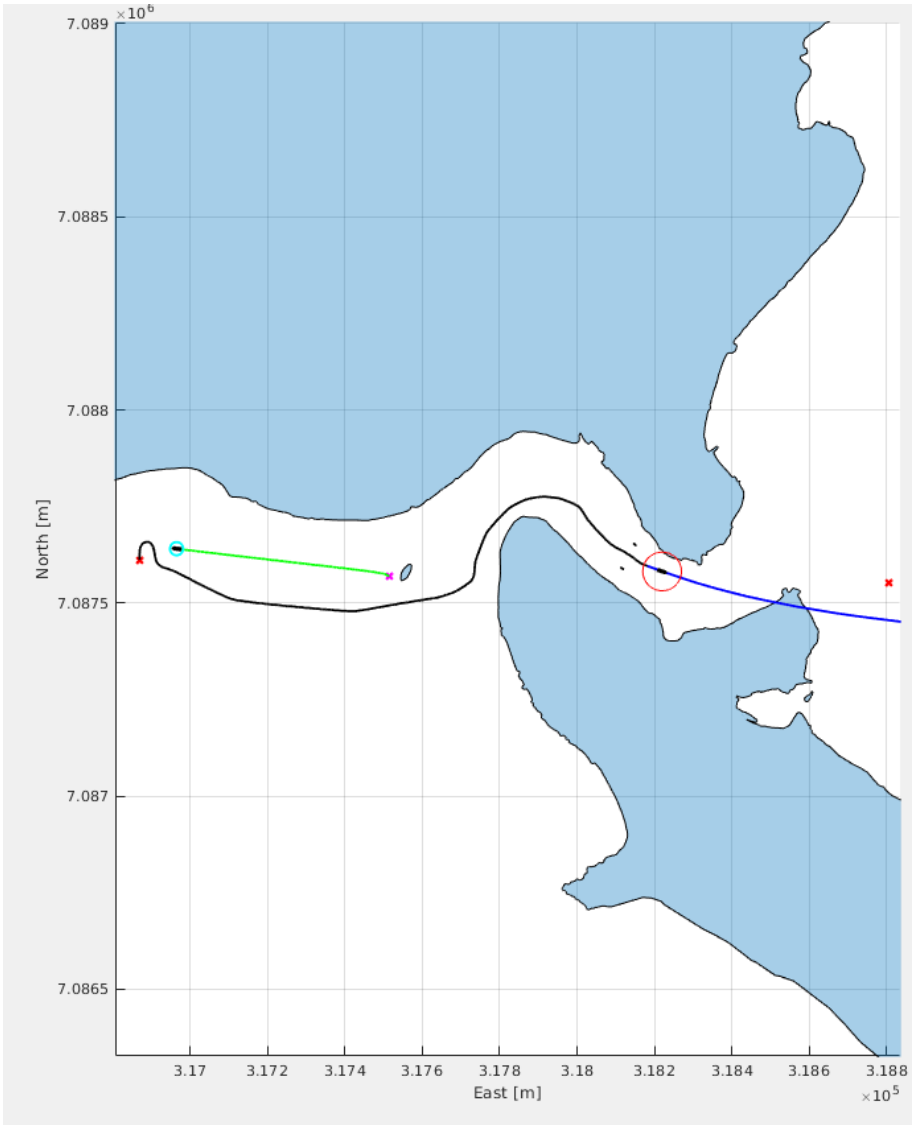


Figure 6.26 Illustration of own-ship simulation using anti-grounding cost-term in PSB-MPC algorithm, with the addition of a obstacle ship. Here with simulation focus on Head-On scenario. Simulation copmlete. Own-ship navigates past end-point.



Chapter 7

Discussion

7.1 Introduction

This section contains a short discussion of the simulation results and furthermore, a list of further work with a discussion on justification of the list.

7.2 Simulations

The simulations in section 6.4 showed proof of concept simulations. Here we saw the own-ship managing to keep a safe distance ti the static obstacles, i.e., the bridge pylons and the land-area specified. However, the own-ship trajectory were not perfect with respect to optimal path, meaning; one could easily imagine a line safer and less time-consuming than the one chosen. Here one could easily point to a couple of factors. The first one being the overall tuning of the cost-function terms, relative to each other. Secondly the tuning-parameter K_ω denoting the weight of far away points in the horizon, could be tuned for better performance. These steps were not tried out due to the limitation on time and since the results were satisfactory enough for further simulation.

The next simulation section, 6.5, handled more complex simulation cases. In section 6.5.1 COLAV and anti-grounding with one obstacle ship in challenging environments were shown. Whats interesting here is that, in contrast to the first simulation in challenging environments, the own-ship path trough the strait is better, and does not drive too close to the shore-line. Between these simulations a small tuning were carried out, decreasing the control deviation cost-term influence on the optimal problem by a factor of 2. This is possibly one of the reasons this happened.

Furthermore in section 6.5.2 and 6.5.3 two additional simulation cases; Head-On and Overtaking were carried out. Both scenarios showed that the own-ship was able to maneuver in a pleasing fashion to avoid the obstacle-ship. However, before concluding that this was in all cases; a success, a further analysis with additional system plots should be carried out. This is discussed further in 7.3.

7.3 Further work

As this is a large system with a big code-base and great complexity there are a lot of things I did not get the time to test. In table 7.3 a comprehensive list of objectives to further be investigated are gathered. Following below a short discussion of why these elements should be further investigated.

- Simulate cases with the inclusion of different types of weather
- Simulate the algorithm for other ship-types.
- Include more COLAV-cases, and possibly with added obstacle-ships.
- Added plotting and analytics features.

The first item on the list should be carried out with the focus on wind in the anti-grounding algorithm. This is included in the table for the sake of redundancy and completeness with respect to the work in article Blindheim et al. (2020). As the anti-grounding algorithm in the latter article is developed for robustness with respect to wather and loss of power this should be of great focus.

Second item is included since this algorithm is simulated only for the parameters of the Revolt ship discussed in the simulation chapter.

Third item should be a focus area in further work as the system was not tested in all COLAV cases, w.r.t. COLREGS. A simple head-on and overtaking scenario was tested but in addition one should investigate obstacle - ship crossing from both sides. Also a scenario where own-ship is being overtaken should be tested.

Fourth item is included for better system understanding. Plots that should be added for better analytics are:

- Distance between owh-ship and obstacle over time. Both for dynamic obstacles, i.e., other vessels, and static obstacles.
- Path tracking plots
- Surface plot of MPC risk costs.

Chapter 8

Conclusion

The implementation of an ENC - module and a anti-grounding cost term to the PSB-MPC algorithm, were carried out in this thesis. By adding a weight to hamper the relevancy of polygons far away in the prediction horizon, the anti-grounding algorithm performed in challenging environments and was able to steer away from large rocks and bridge pylons. This also while simultaneously adding obstacle ships, sailing in the proximity.

In the many simulations carried out some occurrences of non-optimal behavior in complex simulation studies were captured due to lack of optimal tuning, especially when simulating with all 4 cost-terms. However, the algorithm showed robust performance in all simulations not leading to any dangerous situations. Overtaking, Crossing and Head-On simulations all showed promising results.

The implementation of a ENC module with a working anti-grounding algorithm were a important progression in pushing the system into becoming a fully autonomous surface vessel.

Bibliography

Bakdi, A., Glad, I.k., Vanem, E., Engelhardtsen, , 2019. Ais-based multiple vessel collision and grounding risk identification based on adaptive safety domain. Journal of Marine Science and Engineering .

Blindheim, S., Johansen, T.A., Gros, S., 2020. Risk-based model predictive control for autonomous ship emergency management. 21st IFAC World Congress .

Coldwell, T., 1983. Marine traffic behaviour in restricted waters. The Journal of navigation 36, 430–444.

COLREGS, 2020. Colregs. URL: https://www.google.no/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi8i9yg9aHrAhXhs4sKHTvwDH8QFjAAegQIBRAB&url=http%3A%2F%2Fwww.mar.ist.utl.pt%2Fmventura%2FProjecto-Navios-I%2FIMO-Conventions%2520(copies)%2FCOLREG-1972.pdf&usg=AOvVaw2jZf2KZwBfJoFWTJ75obOb.

Davis, P., Dove, M., Stockel, C., 1980. A computer simulation of marine traffic using domains and arenas. The Journal of navigation 33, 215–222.

Eduardo F. Camacho, C.B.A., 2013. Model Predictive Control. Springer.

Eriksen, B., Withil, E., Brekke, E., Flåten, A., March 2018. Radar-based maritime collision avoidance using dynamic window. IEEE Aerospace Conference .

Fiorini, P., Shiller, Z., March 2018. Motion planning in dynamic environments using velocity obstacles. The International Journal of Robotics Research 17, 760–772.

Fossen, T.I., 2011. Handbook of marine craft hydrodynamics and motion control. Wiley.

Fox, D., Burgard, W., April 1997. The dynamic window approach to collision avoidance. IEEE Robotics Automation Magazine 4(1):23 - 33 .

Goodwin, E., 1975. A statistical study of ship domains. The Journal of navigation 28, 328–344.

Hagen, I.B., Kufoalor, D.K.M., Brekke, E.F., Johansen, T.A., 2018. Mpc-based collision avoidance strategy for existing marine vessel guidance systems. IEEE International Conference on Robotics and Automation , 7618–7623.

Hu, Y., Meng, X., Zhang, Q., Park, G., June 2020. A real-time collision avoidance system for autonomous surface vessel using fuzzy logic. IEEE Access 8, 108835–108846.

Johansen, T.A., Perez, T., Cristofaro, A., 2016. Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. IEEE Transactions on Intelligent Transportation Systems 17, 3407–3422.

Kufoalor, D.K.M., Johansen, T.A., Brekke, E.F., Hepsø, A., Trnka, K., 2020. Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios. J. Field Robotics 37, 387–403.

Maiworm, M., Bätlige, T., Findeisen, R., December 2015. Scenario-based model predictive control: Recursive feasibility and stability. IFAC-PapersOnLine 48.

Mazaheri, A., MOntewka, J., Kujala, P., 2013. Modeling the risk of ship grounding - a literature review from a risk management perspective. WMU Journal of Maritime Affairs .

Montewka, J., Krata, P., Goerlandt1, F., Mazaheri, A., Kujala, P., 2011. Marine traffic risk modelling – an innovative approac hand a case study. Journal of Risk and Reliability , 225–307.

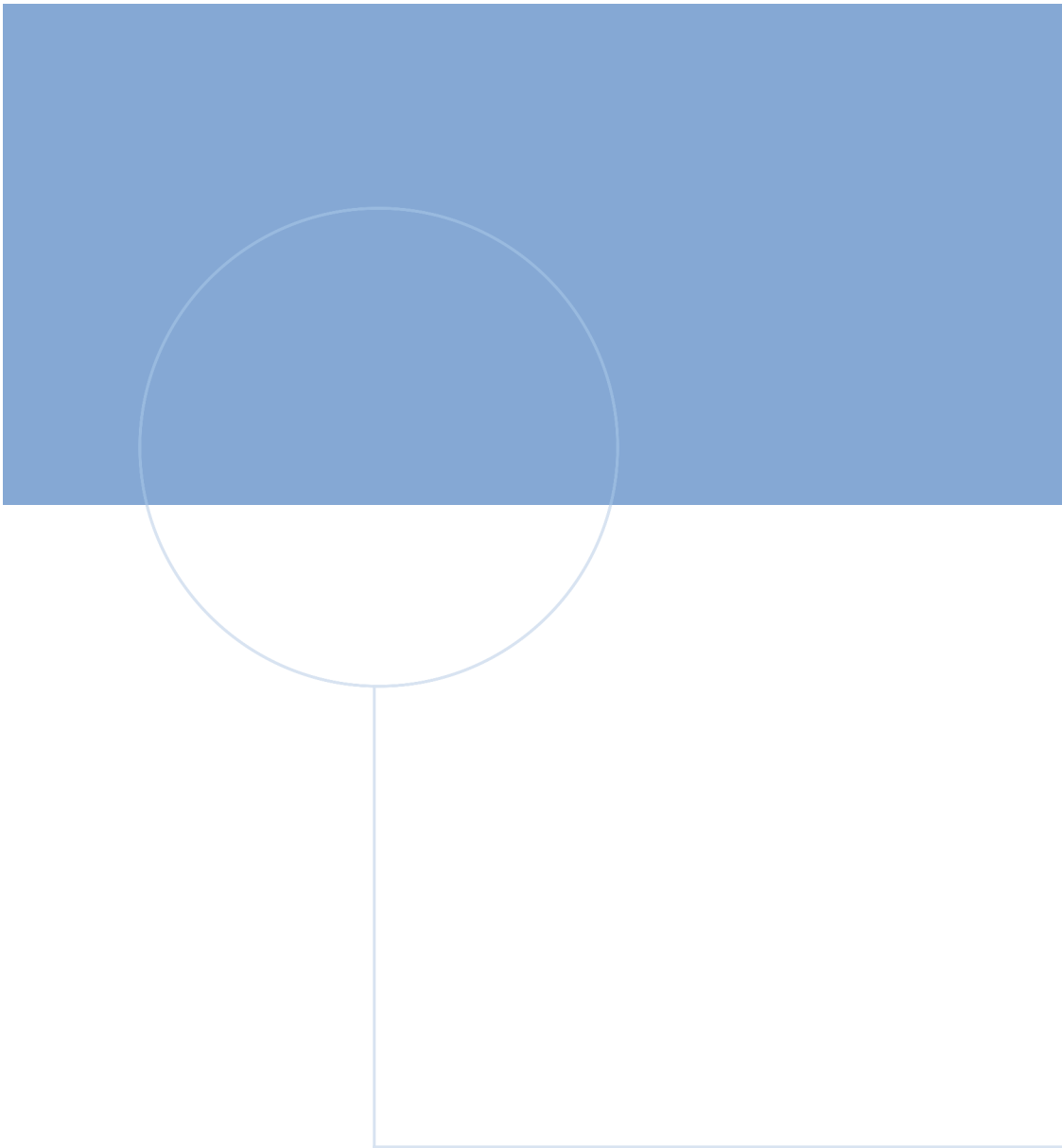
Organization, I.H., Edition 3.1 - November 2000. International hydrographic organization (iho) transfer standard for digital hydrographic data. URL: `https://iho.int/uploads/user/pubs/standards/s-57/31Main.pdf`.

Statheros, T., McDonald-Maier, K., Howells, G., January 2008. Autonomous ship collision avoidance navigation concepts, technologies and techniques. Journal of Navigation 61, 129 – 142.

Tengesdal, T.and Brekke, E.F., Johansen, T.A., 2020. On collision risk assessment for autonomous ships using scenario-based-mpc. IFAC .

Tengsedal, T., Johansen, T., Brekke, E., 2020. Risk-based autonomous maritime collision avoidance considering obstacle intentions .

Appendix



NTNU

Norwegian University of
Science and Technology