Summary

In recent years, a number of maritime collision avoidance (COLAV) systems for autonomous surface vehicles (ASV) have been proposed. While many include solutions for obeying the International Regulations for Preventing Collisions at Sea (COLREGS), most of these are not proactive, in the sense that few make any active efforts to avoid COLREGS situations before they occur. This thesis proposes and analyzes the performance of a proactive COLAV system based on the neighbour course distribution method (NCDM), a long term prediction method based on historical Automatic Identification System (AIS) data. Test results suggest that it is able to use the AIS data to make proactive maneuvers to avoid COLREGS situations, but also demonstrate challenges with using the predictions in the current COLAV framework.

In addition, to improve the situational awareness of the proactive COLAV system, a method for automatically generating coastal constraints from map data is developed, and the resulting constraints included in the COLAV system.

Sammendrag

I de siste årene har det blitt utviklet flere maritime anti-kollisjonssystemer (COLAV-system) for autonome overflatefartøy (ASV). Mange metoder inkluderer løsninger for å følge Konvensjonen om internasjonale regler til forebygging av sammenstøt på sjøen (COLREGS), men ytterst få av disse prøver å aktivt unngå COLREGS-situasjoner før de finner sted. Denne oppgaven foreslår og tester ytelsen til et proaktivt COLAV system basert på neighbor course distribution method (NCDM), en prediksjonsmetode basert på Automatic Identification System (AIS) data. Testresultatene antyder at den klarer å utføre proaktive manøvre for å unngå COLREGS-situasjoner, men belyser også utfordringer rundt måten prediksjonene blir brukt i det nåværende COLAV-rammeverket.

For å forbedre situasjonsforståelsen til det proaktive COLAV-systemet, er det i tillegg blitt utviklet en metode for å generere begrensninger ved hjelp av kystlinjedata. Disse er blitt implementert i COLAV-systemet.

Preface

This thesis continues the work on the neighbor course distribution method (NCDM) [1], a prediction method based on historical automatic identification system (AIS) data. The method was further developed in [2], and it was deemed that a more extensive evaluation of the method's performance in a collision avoidance (COLAV) system was needed.

Originally intended for use in mid-level COLAV, which handles obstacles at relatively close range, one of the NCDM's main criticisms was its lack of compliance with the International Regulations for Preventing Collisions at Sea (COLREGS). The original problem description also included implementing COLREGS compliance in the COLAV; however, this would have made poor use of the method, as it would have to follow the COLREGS regardless of the prediction. After some discussion, it was decided to instead focus on early proactive action, by increasing the prediction horizon from 8 to 15 minutes, before any maneuver according to COLREGS is necessary. The focus of this thesis is therefore to evaluate NCDM's ability to proactively avoid COLREGS situations, whenever feasible.

The work done is largely based on the master theses of Simen Hexeberg [1] and Bjørnar R. Dalsnes [2], as well as an article co-authored by my co-supervisor Bjørn-Olav H. Eriksen [3]. I have also been provided with the MATLAB code from [2] and parts of [3]. The AIS data set used is the same as in the previous theses.

I would like to thank my supervisor Edmund F. Brekke, who also supervised in [1] and [2], as well as my co-supervisors Bjørn-Olav H. Eriksen and Giorgio D. K. M Kufoalor, for all their helpful advice and insightful discussions.

David H. Wu
Trondheim, July 11, 2019

Contents

List of Tables

List of Figures

Abbreviations

AIS	=	Automatic identification system
CAS	=	Collision avoidance system
COLAV	=	Collision avoidance
COLREGS	=	International Regulations for Preventing Collisions at Sea
CVM	=	Constant velocity model
MPC	=	Model predictive control
MMSI	=	Maritime Mobile Service Identity
NCDM	=	Neighbor Course Distribution Method
PCHIP	=	Piecewise Cubic Hermite Interpolating Polynomial
UTM	=	Universal transversal Mercator projection

Chapter 1

Introduction

1.1 Background

In recent years, there has been extensive research on the commercial use of autonomous surface vessels (ASV), as there are multiple benefits in terms of safety, efficiency and reduced CO_2 emissions. These benefits have also prompted many governments to look into ASVs, with nations such as Norway, Finland, Japan, the USA, and Singapore conducting research and trials.

Central to the operation of an ASV is a collision avoidance (COLAV) system. There are mainly two types of COLAV methods: reactive and deliberate. Reactive methods have little or no access to a priori information, relying only on sensor input. Few computations are required, and new trajectories are calculated often, making the system robust to unexpected situations. However, since information is relatively scarce, the resulting trajectory is usually only optimal in the local sense. The Dynamic Window approach (DW) [4] and the Velocity Obstacle (VO) method [5] are two widely used methods in this category. In the DW approach, a circular trajectory is calculated by limiting the search space to velocities that are safe with respect to the obstacle, and within the limits of the dynamics of the vehicle. The VO method, which assumes that the obstacle has a constant course and speed, avoids collision by considering safe velocities in the velocity space.

On the other hand, deliberate methods use a priori information such as weather forecasts, currents, coastlines, etc. This produces better trajectories in the global sense, at the cost of higher complexity and longer computation times. As a result, these methods are less robust to unexpected changes in the environment. Among these, A* [6] and Rapidly-exploring Random Tree [7] are frequently used.

Lately, there has also been much focus on creating systems that abide by the international regulations for preventing collisions at sea (COLREGS). The COLREGS define primarily three types of situations between two vessels at sea: Head-on, overtaking, and crossing, and COLAV systems in existing literature tend to focus on these. In [8], a method based on VO is proposed, which models the COLREGS situations (head-on, crossing, overtaking) as constraints in the velocity space. Another method based on model predictive control (MPC) is proposed in [9], where COLREGS-compliant trajectories are motivated by penalizing violations in the cost function of the MPC. Although these may be considered deliberate methods, they are not very proactive, in the sense that they do not make any active efforts to avoid COLREGS situations before they occur.

There are also many grey areas and ambiguities in the COLREGS, where the rules require the sailor to act according to “good seamanship”. This knowledge requires years to acquire, and historically there have been many accidents caused by confusion or disagreement as to which rules to apply. This can be even more problematic for autonomous ships, where direct communication between vessels may be impossible or at best delayed.

In order to develop a proactive system, an accurate long-term prediction method is needed. One method that may be suitable for this is the neighbour course distribution method (NCDM) [2], a data-driven prediction method based on historical automatic identification system (AIS) messages. By considering neighboring trajectories, it is able to identify sea lanes as well as how they branch. It was also demonstrated to be able to make predictions with reasonable accuracy and consistency in the 15-minute range.

Originally intended for mid-level COLAV, one of its main critiques was the lack of COLREGS compliance. In this thesis, its viability is therefore tested in a proactive COLAV system, where the aim is to avoid COLREGS situations whenever feasible. This is done by comparing it to CVM in 1000 randomly generated collision scenarios. A number of scenarios are then studied in more detail in a qualitative analysis, to attempt to uncover the strengths or weaknesses of either method.

Generally, however, it is not cost-efficient to attempt to avoid all COLREGS situations. A four-layer hybrid architecture (see Figure 1.1) is therefore proposed, which includes a COLREGS compliant COLAV system to handle these cases. The upper level consists of a high-level planner, which calculates an optimal route to the destination, with respect to static obstacles, currents, weather forecasts etc. Directly below is the proactive COLAV system, responsible for avoiding COLREGS situations whenever it is not too costly to do so. A COLREGS compliant level handles encounters with other vessels, and the reactive level is activated when there is a high risk of collision, overriding the COLREGS if necessary.

Finally, in addition to prediction of dynamic obstacles, static obstacles, i.e. land, has also been implemented, for mainly two reasons. First, situational awareness is key in a proactive COLAV system. Including land constraints allows for better decision-making in the early stages of a potentially dangerous situation, and gives a clearer picture of what can be reasonably expected from both the ownship and the target ship. This is especially true in a long-term COLAV system, where the new trajectory may deviate significantly from the original one, and it is more likely that planned trajectories go through land.

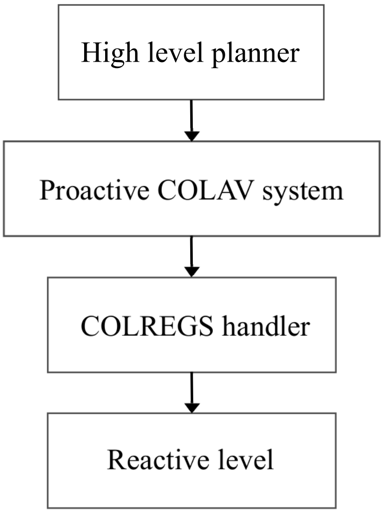


Figure 1.1: An example architecture for using the new proactive COLAV system. When activated, each level overrides the ones above it.

Second, the presence of land plays an important role in shaping the data distribution. In order to properly evaluate a data-driven proactive COLAV system and make a realistic assessment of its limitations, it is therefore important to perform the tests in an environment with similar constraints as in the one which shaped the data. For this purpose, a method for modelling coastline data as a series of interconnected ellipses has been developed, and integrated with the COLAV system as inequality constraints.

1.2 Contributions

The main contributions of this thesis are:

- Development of a method to automatically generate coastline constraints in the COLAV system using map data
- Qualitative and quantitative analysis of modified NCDM in a proactive COLAV system.
- Modification of parameters in NCDM and COLAV system to make it better suited for longterm proactive COLAV. Reduction of interpolation error in NCDM.
- Integration of NCDM with the improved COLAV system in [3], significantly reducing runtime.

1.3 Outline

The remainder of this thesis is structured as follows: Chapter 2 presents the AIS and coastline data sets, as well as a data structure developed in [2], and Chapter 3 contains background theory on NCDM and the COLAV system used. Chapter 4 presents methodology, while Chapters 5 and 6 deal with the results and discussion, respectively. Finally, the conclusion and future work are presented in Chapter 7.

Chapter 2

Data sets

2.1 AIS data

The original data set consists of 3 million AIS messages from ships in Trondheimsfjorden. The fields of the message relevant to this thesis are displayed in Table 2.1. Note that messages with speed over ground less than 0.5 knots have been removed.

Name	Explanation
MMSI	Unique identification number of ship.
Timestamp	Number of seconds since 1st of January 1900, 00:00:00 UTC time.
Longitude	Geographic coordinate. (-180, 180].
Latitude	Geographic coordinate. (-90, 90].
SOG	Speed over ground.

Table 2.1: AIS message fields used in this thesis.

2.1.1 Data structure

The data structure developed in [2] is used, which considers the recent trajectory of a vessel instead of single data points. Trajectories are represented as lists of length n sub-trajectories. Sub-trajectories are given by

$$\boldsymbol{S_i} = \begin{bmatrix} \boldsymbol{p_i} & \boldsymbol{p_{i+1}} & \cdots & \boldsymbol{p_{i+n-1}} \end{bmatrix},$$

where $\boldsymbol{p_i} = \begin{bmatrix} x_i^E & x_i^N \end{bmatrix}$ is a data point represented in north and east coordinates. A trajectory with length m is thus given by

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{S_1} \\ \boldsymbol{S_2} \\ \vdots \\ \boldsymbol{S_{m-n+1}} \end{bmatrix}, \tag{2.1}$$

a $(m - n + 1) \times n$ matrix. Finally, all N trajectories are collected in a list:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{T_1} \\ \boldsymbol{T_2} \\ \vdots \\ \boldsymbol{T_N} \end{bmatrix}. \tag{2.2}$$

A different interpolation method is used in this thesis than in the original method. The process of creating trajectories from the original data set is otherwise the same, and can be summed up in the following three steps:

1. Messages with the same MMSI and less than 15 minutes in between are collected. The time limit ensures that trajectories leaving and later entering the data window are considered separate during prediction.

2. The data points are converted to Universal Transversal Mercator (UTM) coordinates, and each trajectory is interpolated using Piecewise Cubic Hermite Interpolating Polynomial (PCHIP). The reasoning for using PCHIP is that ship trajectories most of the time follow straight lines (and the shortest route from a to b), which is better captured by PCHIP than for instance piecewise cubic spline interpolation (see Figure 2.2). Cubic spline interpolation is also particularly problematic for trajectories with large variations in sampling time, resulting in points that lie far outside the original data set (see Figure 2.1).

3. Each of the newly interpolated $m \times 1$ trajectories are reshaped into a new $(m-n+1) \times n$ matrix, where rows are shifted sub-trajectories of length n (see (2.1)). In this thesis, sub-trajectories have length $n = 3$, and the interpolation interval is one minute.

2.1.2 Coastal data

Map data for Nord-Trøndelag and Sør-Trøndelag (see Figure 2.3) was taken from the GADM database [10]. The data points are divided into land masses, and oriented counter-clockwise around each landmass. Originally in WGS84 datum, the data are converted to UTM coordinates.

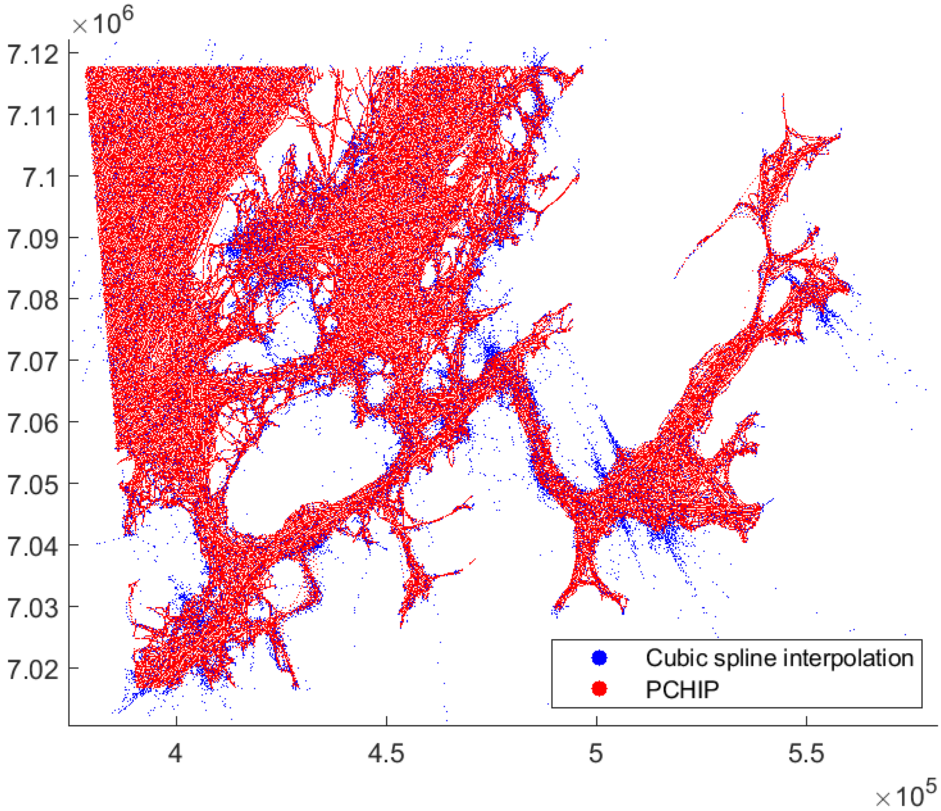


Figure 2.1: Interpolated AIS data set using PCHIP and piecewise cubic spline interpolation.

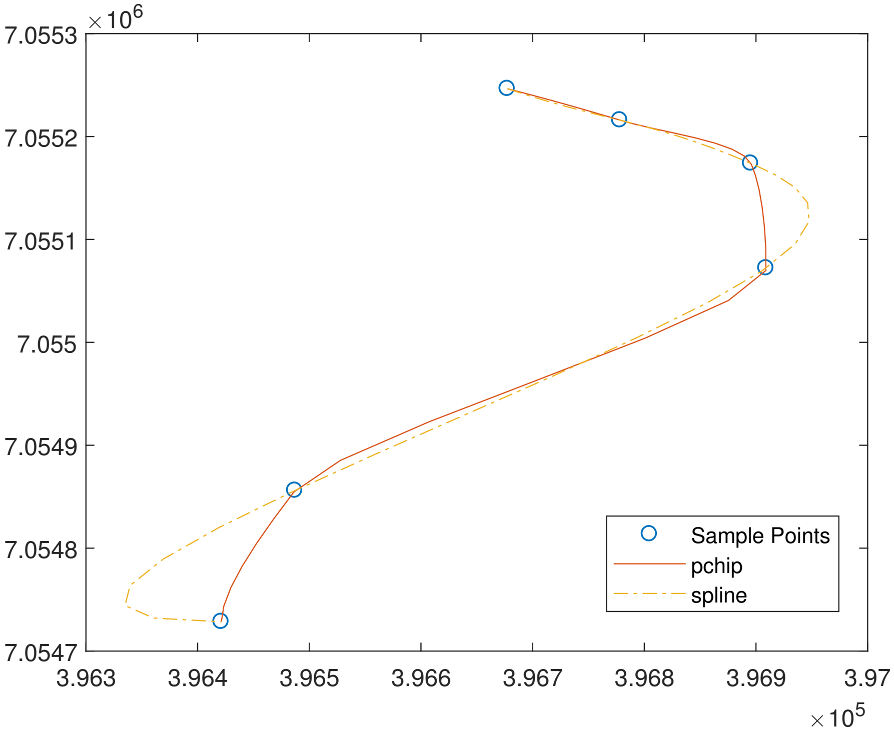


Figure 2.2: Sample points from a trajectory, interpolated using PCHIP and piecewise cubic spline interpolation. Unlike PCHIP, piecewise cubic spline interpolation leads to data points far outside the original data set.

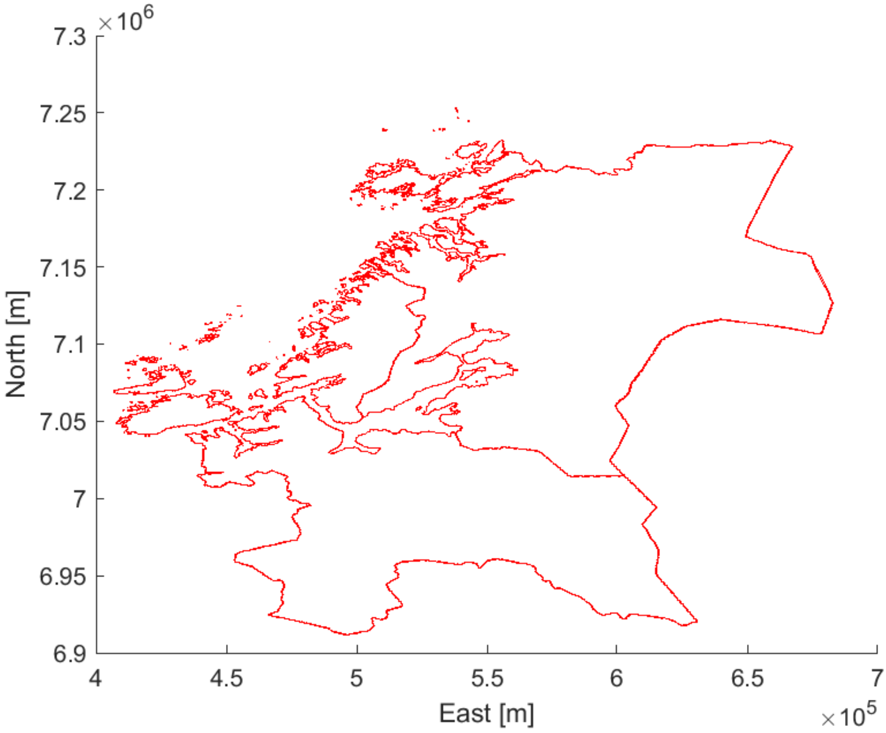


Figure 2.3: Map data for the coastline and borders of Nord-Trøndelag and Sør-Trøndelag.

Chapter 3

Background Theory

3.1 COLREGS

The Convention on the International Regulations for Preventing Collisions at Sea 1972 (COLREGS) [11] is a set of 41 rules regulating maritime traffic. Between power-driven vessels at sea, the rules describe three types of scenarios: Overtaking, head-on and crossing, which henceforth will be referred to as COLREGS situations. Although the goal is to avoid COLREGS situations, the analysis later on requires some knowledge of what they are and when they apply. The main contents of rules 13-17 are therefore briefly reiterated here:

Rule 13 covers overtaking situations, and states that the overtaking vessel is the give-way vessel, while the vessel being overtaken is the stand-on vessel. A vessel is considered to be overtaking another vessel "when coming up with another vessel from a direction more than 22.5°abaft her beam".

Rule 14 deals with head-on situations. When two power-driven vessels are moving towards each other such that a risk of collision exists, both vessels shall alter their course starboard such that they pass each other's port side. Here, both vessels are give-way vessels.

Rule 15 covers crossing situations, and states that the vessel which has the other vessel on its starboard side has to give way. If possible, it should avoid crossing in front of the other vessel.

Rule 16 dictates that the give-way vessel should, whenever possible, take early and substantial action to stay clear of the other vessel.

Rule 17 states that the stand-on vessel should keep her course and speed. However, if it is clear that the give-way vessel is not taking appropriate action to avoid collision, the stand-on vessel is required to make an evasive manoeuvre.

3.2 Generation of coastal constraints

A method for converting map data to ellipse-shaped inequality constraints has been developed. Since the coastline is non-convex, the functions generally need to be smooth in order for the solver to find the derivatives. An alternative is to use polytopes and lifting, such as in [12], however, this increases computation time significantly in scenarios with a long coastline. Therefore, the choice was made to use ellipses.

With ellipses, there are mainly two approaches. The first one is to cover each land mass with as few ellipses as possible. An existing method from computer vision does this, by fitting multiple connected ellipses to a silhouette [13]. The connections between the ellipses are represented by a tree structure, where the root node is the ellipse with the most connections and lower levels have fewer ones. Originally intended for use on human silhouettes, the paper shows that it does this with good results. However, no method exists yet for automatically determining the structure of the tree, i.e. the connections between the ellipses, and doing this manually for large amounts of map data is non-trivial and not within the scope of this thesis.

The new method is based on the second approach, which is to fit ellipses along the coastline. Although more costly in terms of number of ellipses, it is fast and fairly robust. The method fits ellipses to the data points, by encapsulating groups of points that lie roughly in a straight line in an ellipse. More specifically, it identifies groups of points that lie on a straight line within an error threshold, marking the first and last point of the group as *knot*. Ellipses are then fit using consecutive knots as focal points with the direct ellipse fit method [14], which is relatively fast and robust. Because the method in general yields thicker ellipses where the points are further away, the eccentricity is chosen to depend on their distance. Setting the eccentricity β equal to

$$\beta = \exp(-\frac{10}{||\boldsymbol{F_1}, \boldsymbol{F_2}||_2}),$$

where $\boldsymbol{F_1}, \boldsymbol{F_2}$ are focal points, is found to yield good results. Then, any points still lying outside the ellipses are marked as knots, and the nearby points are refit. Finally, the ellipses are added in the optimization problem as inequality constraints (see Section 3.8.4).

One problem with inequality constraints is that the optimization solver might plan a course that "jumps" over the ellipses. To prevent this, a second layer of ellipses is added, by copying and shifting the existing ellipses further in towards land. The ellipses in the new layer are then scaled by a set amount to better fill the narrow spaces from the outer layer (see Figure 3.1).

Since the data for each landmass is ordered in a counter-clockwise direction, given to focal points $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$, the shift for an ellipse is:

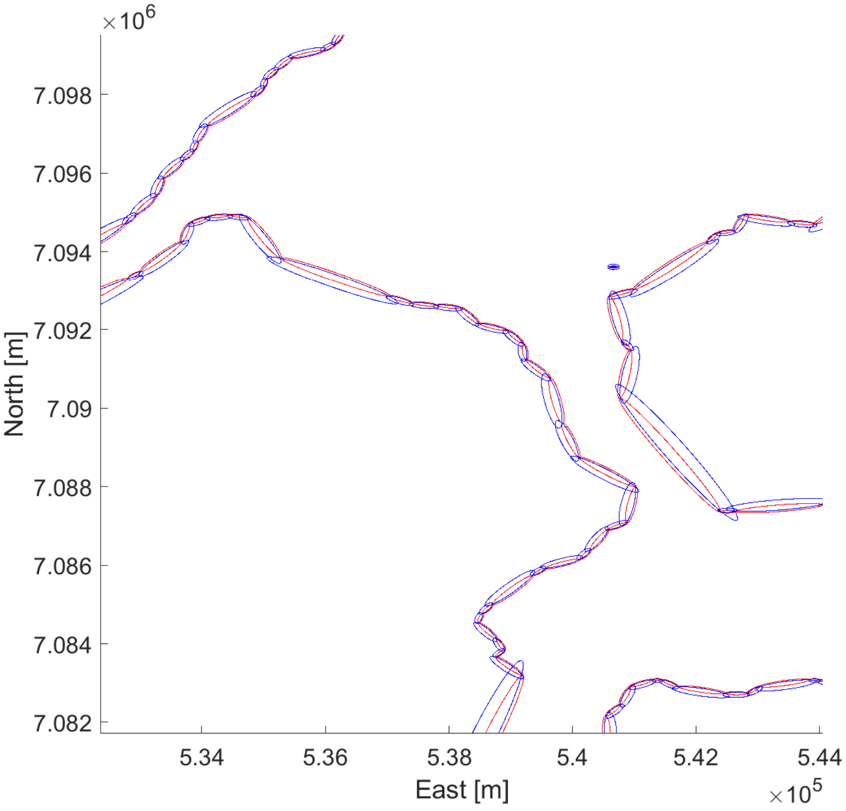


Figure 3.1: Coastline represented using two layers of ellipses. The blue ellipses represent the inner layer which are further in towards land, while the red ellipses are the inner layer closest towards sea.

$$\Delta = \psi \begin{bmatrix} \boldsymbol{F}_{2N} - \boldsymbol{F}_{1N} \\ \boldsymbol{F}_{2E} - \boldsymbol{F}_{1E} \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.1}$$

where the subscripts N,E denote the North and East coordinates in the NED frame, the operator × denotes the cross product, and ψ is a weighting parameter.

3.3 Model predictive control

Model predictive control (MPC) is a method used in a wide range of industrial applications, which requires a sufficiently accurate model of the system that is to be controlled. At each sampling instant, the control input is calculated by solving a finite horizon open loop optimal control problem, with the current state of the plant as the initial conditions. The first control input is then applied, and the process repeated [15]. A major advantage is that inequality constraints can be set on both process and input variables by defining them in the optimization problem.

3.4 Gaussian Mixture Models

Gaussian Mixture Models [16] is an unsupervised machine learning algorithm, which fits the given data to a weighted sum of Gaussians:

$$p(\boldsymbol{x}; \lambda) = \sum_{i=1}^{L} \omega_i \boldsymbol{g}(\boldsymbol{x}; \mu_i, \boldsymbol{\Sigma}_i). \tag{3.2}$$

Here, $\boldsymbol{x}$ is a d-dimensional point, L is the number of components in the mixture, and $\boldsymbol{g}(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a d-variate Gaussian function, given by

$$\boldsymbol{g}(\boldsymbol{x}; \mu_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \mu_i) \right), \tag{3.3}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ denote the mean and covariance of Gaussian i, respectively. The variable λ is defined as

$$\lambda = \{\omega_i, \mu_i, \boldsymbol{\Sigma}_i\}_{i=1,\ldots,L}, \tag{3.4}$$

represents all the parameters.
In addition, each point is assigned a latent variable $\boldsymbol{z}$, which represents the component it belongs to. For a point $\boldsymbol{x}_k$ belonging to component i, the ith element of $\boldsymbol{z}$ is equal to one, while the others are zero.

The method is based on the Expectation-Maximization (EM) algorithm. In the expectation step, the membership weights are calculated, which indicates the probability that a point belongs to a certain component. The probability that point i belongs to component k is given by:

$$\tau_{ik} = \frac{\hat{\omega}_k \mathcal{N}(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{j=1}^{K} \hat{\omega}_j \mathcal{N}(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)} \tag{3.5}$$

where $\hat{\omega}_i$, $\hat{\boldsymbol{\mu}}_i$, and $\hat{\boldsymbol{\Sigma}}_i$ are the estimates of the weight, mean, and covariance of component i, respectively.

In the maximization step, these estimates are updated using the membership weights:

$$\hat{\omega}_k = \frac{\sum_i^N \tau_{ik}}{N} \tag{3.6}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_i^N \tau_{ik} \boldsymbol{x}_i}{\sum_i^N \tau_{ik}} \tag{3.7}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_i^N \tau_{ik} \boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^T (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)}{\sum_i^N \tau_{ik}}, \tag{3.8}$$

where N is the number of points. This procedure is repeated until the parameters converge.

3.5 Constant velocity model

The constant velocity model is used for predicting the future position of a moving object, by assuming that the velocity is constant between time steps. A small amount of noise is added to account for the uncertainty of the prediction. The same model is used as in [2].

Given a state vector $\boldsymbol{x_t} = \begin{bmatrix} N & V_N & E & V_E \end{bmatrix}^T$, where N, E denote a vessel's north and east coordinates in the NED frame, and V_N, V_N are corresponding velocities. The model is then given by

$$\boldsymbol{x_{t+1}} = \boldsymbol{A} \boldsymbol{x_t} + \boldsymbol{v_t}, \qquad p(\boldsymbol{v_k}) = \mathcal{N}(\boldsymbol{v_k}; 0, \boldsymbol{Q}),$$

where $\boldsymbol{v_k}$ is zero-mean Gaussian process noise. The covariance $\boldsymbol{Q_T}$ and state transition matrix $\boldsymbol{A}$ are given by

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{Q} = \sigma_a^2 \begin{bmatrix} T^4/3 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/3 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix}, \tag{3.9}$$

where T is the time step in seconds and σ_a is the noise covariance parameter.

3.6 Neighbor course distribution model

The neighbor course distribution model (NCDM) is a method originally developed in [1], and then further improved in [2]. It uses the data structure described in Chapter 2. NCDM makes use of the fact that ships typically travel along fixed sea lanes, using more or less

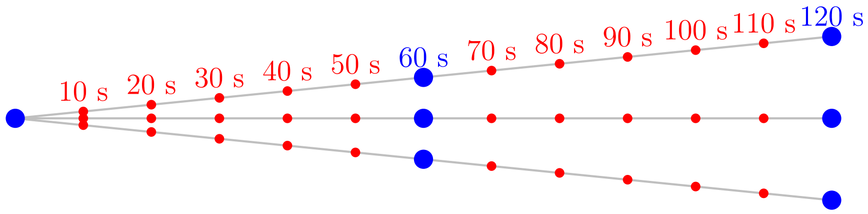


Figure 3.2: Interpolation of a prediction tree with three branches originating from the root node. The blue and red points represent the tree before and after interpolation, respectively.

the same routes as the ships before them. A prediction can then be made by assuming that the ship will have the same course and speed as nearby trajectories in the data set.

First, the state $\boldsymbol{X_k}$ of a vessel is defined as its most recent length $n-1$ sub-trajectory:

$$\boldsymbol{X_k} = \begin{bmatrix} \boldsymbol{p_{k-n+2}} & \cdots & \boldsymbol{p_{k-1}} & \boldsymbol{p_k} \cdot \end{bmatrix} \tag{3.10}$$

As before, n denotes the number of points in each sub-trajectory in the new data structure. A set of similar nearby trajectories can then be found, called close neighbors (CN). The set of CNs C is given by

$$C(\boldsymbol{X_k}) = \{\boldsymbol{S_i} | d(\boldsymbol{S_i}, \boldsymbol{X_k}) \le r_c, \boldsymbol{S_i} \in \boldsymbol{D}\},$$

where $\boldsymbol{S_i}$ is a sub-trajectory in the set of all trajectories $\boldsymbol{D}$, r_c is a scalar parameter and d is the distance function for two sub-trajectories of equal length:

$$d(\boldsymbol{S_i}, \boldsymbol{S_j}) = \sum_{w=1}^{m} \|\boldsymbol{p_{iw}} - \boldsymbol{p_{jw}}\|_2$$

Note that while the state of the vessel has length $n-1$, sub-trajectories in $\boldsymbol{D}$ have length n. In other words, only the $n-1$ first points of the candidate trajectories are considered when creating the set of CNs. The nth point is used to predict the position of the vessel at the next time step, given by $\boldsymbol{\hat{p}_{k+1}} = \boldsymbol{p_k} + (\boldsymbol{p_n} - \boldsymbol{p_{n-1}})$ (Figure 3.3). The new predicted state is

$$\boldsymbol{\hat{X}_{k+1}} = \begin{bmatrix} \boldsymbol{p_{k-n+1}} & \cdots & \boldsymbol{p_k} & \boldsymbol{\hat{p}_{k+1}} \cdot \end{bmatrix} \tag{3.11}$$

By sampling multiple trajectories from the set of CNs, different estimates can be obtained, and a prediction tree can be created (Figure 3.4). Let the current position of the vessel be the root node $\boldsymbol{X_{1,1}}$, and its child nodes the predictions of the state at the next time step. Node $\boldsymbol{\hat{X}_{k,j}}$ denotes the jth child node at depth k, i.e. $k-1$ steps into the future. After sampling, the branches are linearly interpolated (Figure 3.2).

Finally, at each level, the nodes are used to fit a GMM. Model selection, i.e. choosing the number of components in the Gaussian Mixture, is performed by increasing the number of components until that the Euclidean distance between the means becomes less than some

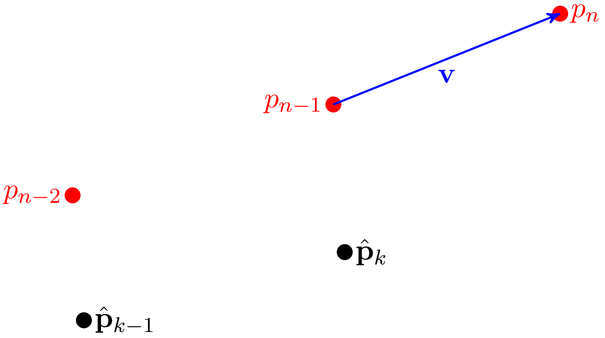


Figure 3.3: The red points belong to a trajectory in the data set, and the black points are estimates of the vessel's position at different time steps. The red trajectory is considered a close neighbor if the Euclidean distance between $\begin{bmatrix}\boldsymbol{p_{n-2}} & \boldsymbol{p_{n-1}}\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{\hat{p}_{k-1}} & \boldsymbol{\hat{p}_k}\end{bmatrix}$ is less than r_c. The new estimate $\boldsymbol{\hat{p}_{k+1}} = \boldsymbol{\hat{p}_k} + \boldsymbol{v}$. Illustration courtesy of [2].

margin M. The reason for using this criterion over others such AIC and BIC, is because NCDM aims to identify branching of sea lanes, represented by multiple components with significantly different means.

3.7 Modified NCDM

Modified NCDM was introduced in [2] in an effort to improve the consistency of the predictions, as well as make the method work in areas with sparse data. In areas with low data density, a sub-trajectory based on predictions using CVM is created. Copies of the new sub-trajectory is then added to the set of CNs. The number of copies scales with the density of the nearby AIS data, and is given by

$$W = \begin{cases} [\alpha \frac{1}{M}], & M > 0 \\ 1, & M = 0 \end{cases}$$

where W is the number of copies, $\alpha > 0$ is a weight parameter, M is the number of sub-trajectories in the set of CNs, and $[\cdot]$ rounds to the nearest integer. Essentially, α represents the trade-off between NCDM and CVM.

3.8 MPC-based COLAV

An MPC-based COLAV system was developed in [17], and adapted for use with modified NCDM in [2]. An updated version of the original framework [3] is used here, which includes relative trajectory tracking. Furthermore, coastal constraints created using the method in Section 3.2 have been included, using the existing method in [3] for adding static obstacles.

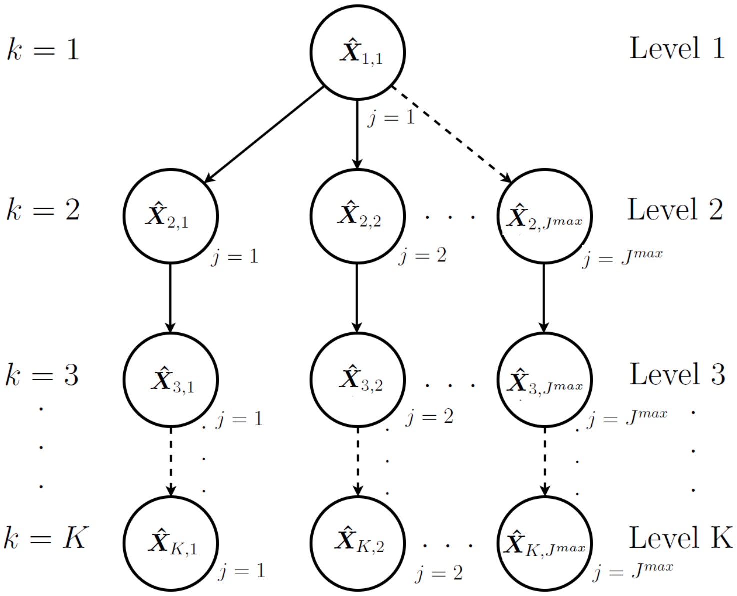


Figure 3.4: A prediction tree with $N_{1,1} = J^{max}$ and all other $N_{k,j} = 1$. In other words, the tree has J^{max} branches which all originate from the root node. Illustration courtesy of [1].

3.8.1 ASV model

For the ASV, a purely kinematic model is used, given by

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \cos(\Phi) & 0 \\ \sin(\Phi) & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}, \tag{3.12}$$

where $\eta = \begin{bmatrix} N & E & \Phi \end{bmatrix}^T$ is the pose of the ASV, with N and E representing the North and East position, respectively, and Φ is the clockwise angle with respect to north. The vector $\boldsymbol{u} = \begin{bmatrix} U & r \end{bmatrix}^T$, where U is the speed over ground (SOG) and r is the rate of turn (ROT) of the vessel. Side-slip and currents have been omitted from the model, as these will be handled by the reactive layer.

For underactuated vehicles, which typically depend on a rudder for course changes and forward thrust from the propeller, the ROT is heavily dependent on the SOG. The dependency between the variables is captured in the following constraints:

$$U_{min}(r_{min}) \leq U \leq U_{max}(r_{max}) \tag{3.13}$$

$$r_{min} \leq r \leq r_{max} \tag{3.14}$$

The definitions of the functions $U_{min}(r_{min})$ and $U_{max}(r_{max})$ can be found in [3].

3.8.2 Relative trajectory tracking

With regular trajectory tracking, the desired positions $p_{d_{1:N_p}}$, where N_p is the number of prediction steps, are fixed. This makes manoeuvres involving acceleration, such as overtaking, problematic, since it will later need to decelerate again to match the fixed desired positions. Relative trajectory tracking fixes this, by tracking the desired trajectory with a time offset t_b. This way, the relative trajectory can be adjusted after overtaking another vessel by increasing t_b. It is given by

$$\bar{\boldsymbol{p}}_d(t) = \boldsymbol{p_d}(t + t_b) \tag{3.15}$$

The time offset t_b is calculated at each time step, by finding the offset which minimizes the Euclidean distance from the ASV to the desired trajectory. At a given time step t_0:

$$t_b(t_0) = \text{arg}\,\min_{t_b} \|\boldsymbol{p_d}(t_0 + t_b) - \boldsymbol{p}(t_0)\|_2, \tag{3.16}$$

which can be solved using a line search algorithm.

3.8.3 Control objective

The control objective is to stay as close as possible to a desired trajectory $\boldsymbol{p_d} = [N_d(t) \quad E_d(t)]^T$, while avoiding static and dynamic obstacles. In addition, changes in course and speed should be clear and visible to nearby vessels, as per rule 8 of COLREGS:

Rule 8 (b): Any alteration of course and/or speed to avoid collision shall, if the circumstances of the case admit, be large enough to be readily apparent to another vessel observing visually or by radar; a succession of small alterations of course and/or speed should be avoided.

3.8.4 Optimization problem

The control objective can be formulated as an optimal control problem (OPC):

$$\begin{aligned} \text{minimize} \quad & \phi(\boldsymbol{\eta}(t), \boldsymbol{u}(t)) \\ \text{subject to} \quad & \dot{\boldsymbol{\eta}}(t) = \boldsymbol{F}(\boldsymbol{\eta}(t), \boldsymbol{u}(t)), \\ & \boldsymbol{h}(\boldsymbol{\eta}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}, \\ & \boldsymbol{\eta}(t_0) = \bar{\boldsymbol{\eta}}_{\boldsymbol{0}}, \end{aligned} \tag{3.17}$$

where ϕ is the objective function, $\boldsymbol{\eta}$ is the pose of the vehicle, $\boldsymbol{u}$ is the control input, and $\boldsymbol{F}$ represents the model of the ASV (3.12). The vector $\boldsymbol{h}$ contains the inequality constraints, and $\bar{\boldsymbol{\eta}}_{\boldsymbol{0}}$ is the initial pose of the ASV.

Since the continuous optimization is difficult to solve, a non-linear program (NLP) is defined by discretizing (3.17) using direct multiple shooting, resulting in an NLP with N_p prediction steps:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimize}} \quad & \phi(\boldsymbol{w}) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0}, \\ & \boldsymbol{h}(\boldsymbol{w}) \leq \boldsymbol{0}, \end{aligned} \tag{3.18}$$

where $\boldsymbol{w} = \begin{bmatrix} \boldsymbol{\eta}_0^T & \boldsymbol{u}_0^T & \dots & \boldsymbol{\eta}_{N_p-1}^T & \boldsymbol{u}_{N_p-1}^T & \boldsymbol{\eta}_{N_p}^T \end{bmatrix}^T$ is the vector of decision variables, and $\boldsymbol{g}$ is the vector of equality constraints.

Objective function

The objective function is given by:

$$\phi(\boldsymbol{w}), \boldsymbol{p}_{d_{1:N_p}}) = \phi_p(\boldsymbol{w}, \bar{\boldsymbol{p}}_d) + \phi_c(\boldsymbol{w}) + \phi_g(\boldsymbol{w}) \tag{3.19}$$

where $\boldsymbol{p}_{d_{1:N_p}} = \begin{bmatrix} \boldsymbol{p_{d_1}} & \boldsymbol{p_{d_2}} \dots \boldsymbol{p}_{dN_p} \end{bmatrix}^T$ is the vector of desired positions. The first two terms are given by

$$\phi_p(\boldsymbol{w}, \bar{\boldsymbol{p}}_d) = \sum_{k=1}^{N_p} (K_p q_p(\left\|\boldsymbol{p}_{k+1} - \bar{\boldsymbol{p}}_{d_{k+1}}\right\|_2; \delta)) \tag{3.20}$$

$$\phi_c(\boldsymbol{w}) = \sum_{k=0}^{N_p-1} (K_{\dot{U}} q_{\dot{U}}(\dot{U}_k) + K_r q_r(r_k)), \tag{3.21}$$

where q_p penalizes deviation from the desired trajectory, while $q_{\dot{U}}$ and q_r penalize changes in SOG and ROT that are not readily observable to nearby vessels. Finally, K_p, $K_{\dot{U}}$, and K_r are tuning parameters.

The deviation from the desired trajectory is captured using a Huber loss function, which is quadratic near origin and resembles the absolute value function for values greater than $\delta > 0$:

$$H(\rho) = \begin{cases} \frac{1}{2}\rho^2 & |\rho| \leq \delta \\ \delta(|\rho| - \frac{1}{2}\delta) & \text{otherwise.} \end{cases} \tag{3.22}$$

Using the Huber function instead of a quadratic loss function prevents position error from dominating the other terms when it is large. The loss function for position q_p is defined as:

$$q_p(\boldsymbol{p}, \bar{\boldsymbol{p}}_d) = H(x - \bar{x_d}) + H(y - \bar{y}_d). \tag{3.23}$$

Since the Huber function is only C^1, this leads to a discontinuous Hessian matrix in (3.18), which makes it difficult to solve. In [3], this is solved by implementing it as a quadratic program (QP):

$$\begin{aligned} \min_{v,\mu} \quad & \delta v + \frac{1}{2}\mu^2 \\ \text{subject to} \quad & -\mu - v \leq \rho \leq \mu + v, \\ & v \geq 0. \end{aligned} \tag{3.24}$$

Eq. (3.24) is a reformulation of (3.22), where slack variables are used to fix the discontinuity issues. The new NLP is given by:

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{v},\boldsymbol{\mu}}{\text{minimize}} \quad & \bar{\phi}_p(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\mu}) + \phi_c(\boldsymbol{w}) + \phi_g(\boldsymbol{w}) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{w},\boldsymbol{\eta}(t_0)) = \boldsymbol{0}, \\ & \boldsymbol{h}(\boldsymbol{w}) \leq \boldsymbol{0}, \\ & \bar{\boldsymbol{h}}_k(\boldsymbol{\eta_k},\boldsymbol{v_k},\boldsymbol{\mu_k},\bar{\boldsymbol{p}}_{\boldsymbol{d},\boldsymbol{k}}) \leq \boldsymbol{0} \quad \forall \quad k \in \{1,\dots,N_p\}, \end{aligned} \tag{3.25}$$

where $\boldsymbol{v} = \left[\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, \dots, \boldsymbol{v}_{N_p}^T\right]^T$ and $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \dots, \boldsymbol{\mu}_{N_p}^T\right]^T$ are slack variables. The new loss function $\phi_p(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\mu})$ is

$$\bar{\phi}_p(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\mu}) = \sum_{k=1}^{N_p} K_p(\delta\boldsymbol{1}^T\boldsymbol{v}_k + \frac{1}{2}\boldsymbol{mu}_k^T\boldsymbol{\mu}_k), \tag{3.26}$$

and $\bar{\boldsymbol{h}}_k(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\mu},\bar{\boldsymbol{p}}_{\boldsymbol{d},\boldsymbol{k}})$ represents the constraints in (3.24).

The terms $q_{\dot{U}}$ and q_r motivate readily observable maneuvers by penalizing a series of small maneuvers more than one large maneuver. This is done using a objective function based on a quadratic and a decaying exponential function:

$$q(p;a,b) = ap^2 + (1 - exp(-\frac{p^2}{b})), \tag{3.27}$$

which give

$$q_{\dot{U}}(\dot{U}) = \frac{100}{q(\dot{U}_{max};a_{\dot{U}},b_{\dot{U}})} q(\dot{U};a_{\dot{U}},b_{\dot{U}}) \tag{3.28}$$

$$q_r(\dot{U}) = \frac{100}{q(r_{max};a_r,b_r)} q(r;a_r,b_r) \tag{3.29}$$

The final term contains the GMMs for predictions made using NCDM. Adding the predictions there and not in the constraints makes better use of the representation of probability that the GMM provides. The GMM for a prediction at a given time is added to the objective function, which makes it more costly to travel in regions that have a high likelihood of containing an obstacle at that time. The new objective function is given by

$$\phi_g(\boldsymbol{w}) = \sum_{k=1}^{N_p} K_g g(\boldsymbol{p}_k;\boldsymbol{\theta}_k)), \tag{3.30}$$

Here, $K_g > 0$ is a tuning parameter and

$$g(\boldsymbol{p}_k|\boldsymbol{\theta}_k)) = \sum_{m=1}^{M} \pi_{km}\mathcal{N}(\boldsymbol{p}_k; \mu_{km}, \boldsymbol{\Sigma}_{km} + \boldsymbol{\Sigma}_{pad}) \tag{3.31}$$

is the value of the GMM at time step k and position $\boldsymbol{p}_k$. The number M is the number of components in the mixture, and θ_k contains the mixture parameters $\boldsymbol{\pi}_k$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, representing the relative weights in mixture, the means and the covariance matrices, respectively. The second term of the covariance matrix is for obstacle padding, and is given by

$$\boldsymbol{\Sigma}_{pad} = \lambda \boldsymbol{I}, \tag{3.32}$$

where $\lambda > 0$ is a weight parameter. One of the problems in [2] was that the predictions for the first time steps had very little uncertainty, leading to small GMMs and planned trajectories that were very to the target ship. This increases the size of the GMMs, which also prevents the COLAV system from planning a trajectory that "jumps" over the GMMs between time steps.

In order to make a fair comparison between NCDM and CVM, the same term ϕ_g is used for representing obstacles when using CVM. There, $M = 1$, i.e. each mixture only has one component, μ_k is the prediction from CVM at time step k, and the covariance matrix is $\boldsymbol{\Sigma}_{pad}$.

Direct multiple shooting

When discretizing using direct multiple shooting, constraints are needed to ensure that the input and state variables follow the dynamics of the ASV (3.12). It is done using 4th order Runge-Kutta:

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{F}(\boldsymbol{\eta}_k, \boldsymbol{u}_k) \\
\boldsymbol{k}_2 &= \boldsymbol{F}(\boldsymbol{\eta}_k + \frac{h}{2}\boldsymbol{k}_1, \boldsymbol{u}_k) \\
\boldsymbol{k}_3 &= \boldsymbol{F}(\boldsymbol{\eta}_k + \frac{h}{2}\boldsymbol{k}_2, \boldsymbol{u}_k) \\
\boldsymbol{k}_4 &= \boldsymbol{F}(\boldsymbol{\eta}_k + h\boldsymbol{k}_3, \boldsymbol{u}_k) \\
\boldsymbol{f}(\boldsymbol{\eta}_k, \boldsymbol{u}_k) &= \boldsymbol{\eta}_k + \frac{h}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4),
\end{aligned} \tag{3.33}$$

where h is the discretization time step. The state of the ASV at the next time step can now be defined as $\boldsymbol{\eta}_{k+1} = \boldsymbol{f}(\boldsymbol{\eta}_k, \boldsymbol{u}_k)$. The shooting constraints are added in the equality constraints as

$$\boldsymbol{g}(\boldsymbol{w}) = \begin{bmatrix} \bar{\boldsymbol{\eta}}_0 - \boldsymbol{\eta}_0 \\ \boldsymbol{f}(\boldsymbol{\eta}_0, \boldsymbol{u}_0) - \boldsymbol{\eta}_1 \\ \boldsymbol{f}(\boldsymbol{\eta}_1, \boldsymbol{u}_1) - \boldsymbol{\eta}_2 \\ \vdots \\ \boldsymbol{f}(\boldsymbol{\eta}_{N_{p-1}}, \boldsymbol{u}_{N_{p-1}}) - \boldsymbol{\eta}_{N_p} \end{bmatrix}. \tag{3.34}$$

Static obstacles

The coastline is modelled as a series of as elliptical inequality constraints. Each constraint is given by:

$$(\frac{\cos(\alpha)(x - x_c)}{x_a})^2 + (\frac{\sin(\alpha)(y - y_c)}{y_a})^2 \geq 1, \tag{3.35}$$

where (x_c, y_c) is the center of the ellipse in the NED frame and x_a, y_a describe the sizes of the major and minor axes, respectively. The variable α is the angle between the major axis and North. An equivalent representation of the inequality with better numerical properties is used (see [3]):

$$\begin{aligned} h_o(x, y, x_c, y_c, x_a, y_a, \alpha) = -\log & \left[\left(\frac{\cos(\alpha)(x - x_c) + \sin(\alpha)(y - y_c)}{x_a} \right)^2 \right. \\ & \left. + \left(\frac{-\sin(\alpha)(x - x_c) + \cos(\alpha)(y - y_c)}{y_a} \right)^2 + \epsilon \right] + \log(1 + \epsilon) \leq 0. \end{aligned} \tag{3.36}$$

The inequality constraint for the ith obstacle is given by

$$\boldsymbol{h}_{s_i} = \begin{bmatrix} h_o(x_1, y_1, x_{c,i}, y_{c,i}, x_{a,i}, y_{a,i}, \alpha_i) \\ h_o(x_2, y_2, x_{c,i}, y_{c,i}, x_{a,i}, y_{a,i}, \alpha_i) \\ \vdots \\ h_o(x_{N_p}, y_{N_p}, x_{c,i}, y_{c,i}, x_{a,i}, y_{a,i}, \alpha_i) \end{bmatrix} \leq \boldsymbol{0}, \tag{3.37}$$

and for S obstacles:

$$\boldsymbol{h}_s(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{h}_{s_1}(\boldsymbol{w}) \\ \boldsymbol{h}_{s_2}(\boldsymbol{w}) \\ \vdots \\ \boldsymbol{h}_{s_S}(\boldsymbol{w}) \end{bmatrix}. \tag{3.38}$$

Control input constraints

The constraints on the control input defined in (3.13),(3.14) are expressed in the following function:

$$\boldsymbol{h}_{u_i}(\boldsymbol{u}_i) = \begin{bmatrix} U_{min}(r_i) - U_i \\ -(U_{max}(r_i) - U_i) \\ r_{min} - r_i \\ -(r_{max} - r_i) \end{bmatrix}, \tag{3.39}$$

which form the inequality constraints:

$$\boldsymbol{h}_u(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{h}_{u0}(\boldsymbol{u}_0) \\ \boldsymbol{h}_{u1}(\boldsymbol{u}_1) \\ \vdots \\ \boldsymbol{h}_{u_{N_{p-1}}}(\boldsymbol{u}_{N_{p-1}}) \end{bmatrix}. \tag{3.40}$$

Finally, the inequality constraints in $\boldsymbol{h}_s(\boldsymbol{w})$ and $\boldsymbol{h}_u(\boldsymbol{w})$ are concatenated, giving

$$\boldsymbol{h}(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{h}_s(\boldsymbol{w}) \\ \boldsymbol{h}_u(\boldsymbol{w}) \end{bmatrix} \tag{3.41}$$

3.9 Evaluation metrics

The performance analysis of the proactive COLAV system will be based on mainly two metrics, namely the deviation from the planned trajectory, and the number of COLREGS situations avoided. The second metric will be discussed in more detail, as it is not straightforward from the COLREGS themselves when and where a COLREGS situation begins.

3.9.1 Total deviation from planned trajectory

While avoiding COLREGS situations is the main objective of the proactive COLAV system, having too much distance means the ship has to deviate significantly from the planned trajectory. This again leads to extra costs in time and fuel consumption.
It is given by

$$\sum_{i=1}^{N} \text{dist}(\boldsymbol{x_i}, \boldsymbol{T_d}),$$

where N is the number of time steps, $\boldsymbol{T_d}$ is the desired trajectory, $\boldsymbol{x_i}$ is the ownship's position at time step i, and dist is the Euclidean distance function for a point to a trajectory.

3.9.2 Ability to avoid COLREGS situations

An implementation of a COLREGS detector will be used, courtesy of the author of [3]. It is implemented as a state machine, using the position and velocity of the ownship and target ships to determine which COLREGS situation the ships are in. It is based on the notion of closest point of approach (CPA).
CPA is the point where the distance between the vessels will reach its minimum value, given that the vessels maintain their current velocity. The time at closest point of approach (TCPA) is the time until the vessels are at CPA, and distance at closest point of approach (DCPA) is the distance between the vessels at CPA. DCPA and TCPA are metrics commonly used in assessing collision risk.

Choice of DCPA and TCPA

In order to choose a TCPA for the COLREGS detector, it is necessary to first determine the point at which the steering rules of COLREGS apply. In other words, at what point vessels are deemed to be in a COLREGS situations, and required to steer accordingly. Formally, this is defined in rule 11:

Rule 11: "Rules in this Section shall apply to vessels in sight of one another."

Here, “this Section” refers to Section II (Rules 11-18). In other words, according to the rules, the line of sight is the determining factor.

This is, however, very conservative and often unfeasible, and in practice, factors such as the distance, speed, and size of ships are taken into account instead. Typically, much emphasis is placed on acting according to *good seamanship*. In [18], it is referred to a statement in the U.K. Marine Accident Investigation Branch [19]: “The old rule of thumb that avoiding action should be taken in the four to six mile range bracket still holds good for large ships in open waters.” A cargo ship typically travels at 30-46 km/h [20], and six miles equals a bit under 10 kilometers.

For two cargo ships each travelling at 30 km/h towards each other, the rule of thumb of 10 km applies 10 minutes before collision. It further adds that “For small vessels, especially in confined waters, give-way action is, of necessity, taken at shorter ranges, but the same philosophy applies.”

The ownship is a vessel of medium size, assumed to operate under calm weather conditions at around 6m/s. Because of its size, it is also reasonable to assume that it has better maneuverability than the aforementioned large ships. Based on this, a TCPA of 270 s is chosen.

For DCPA, Rule 8d of COLREGS states that:

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

Again, the exact distance is not defined, as it varies on a case-by-case basis. In [21], it is referred to a court case involving two large ships in the U.S. Fifth Circuit, where it was deemed that a 600 yard (548m) distance is sufficient in a fairway. Based on this, and the fact that Trondheimsfjorden is relatively narrow in some areas, a fairly conservative DCPA of 900 m is chosen.

Chapter 4

Method

The scenarios were simulated using the MPC-based COLAV system described in Chapter 3, with the goal of assessing NCDM's strengths and weaknesses compared to CVM in a proactive long term COLAV system.

4.1 Generating cases

For the quantitative analysis, $N = 1000$ scenarios are generated. First, starting points $\boldsymbol{x_1}$ are sampled using latin hypercube sampling. A fixed length s is then chosen to be 9000 m. For an ownship travelling at 6 m/s, each scenario will take at least fifteen minutes, ensuring that the entire dynamics of the COLAV system is captured. Endpoints $\boldsymbol{x_2}$ are then found by sampling an angle θ between 0 and 2π for each starting point, again using latin hypercube sampling, giving

$$\boldsymbol{x_2} = \boldsymbol{x_1} + s \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix}$$

A desired trajectory T_d is created from each waypoint pair, assuming constant course and nominal speed U_d. If the trajectory intersects land, it is discarded. Otherwise, it is matched to an obstacle trajectory T_{obs} that satisfies

$$dist(\boldsymbol{T_d(t)}, \boldsymbol{T_{obs}(t)}) < 250,$$

and

$$dist(\boldsymbol{T_d(0)}, \boldsymbol{T_{obs}(0)}) > 5000,$$

where dist denotes the minimum euclidean distance in meters between the two trajectories. The first criterion ensures that a maneuver has to be made in order to avoid a COLREGS situation, while the second ensures that the scenario does not begin in one.

Finally, the scenarios are classified according to the behavior of the obstacle, namely:

1. The obstacle moves in a straight path
2. The obstacle makes a steady turn
3. The obstacle makes a sharp turn

To determine the category, the maximum course change of the obstacle over five time steps ϕ_{max} and the net course change ϕ_{net} are evaluated, given by

$$\phi_{max} = \text{max}(|\phi_{i+5} - \phi_i|), \quad \forall i \in [1, (K-5)],$$

$$\phi_{net} = |\phi_k - \phi_1|$$

where $\phi_i \in [-\pi, \pi)$ is the course at time step i, and K is the number of time steps in the simulation. Then,

$$\text{Category} = \begin{cases} \text{Straight} & \phi_{max} < 1^\circ \quad \wedge \quad \phi_{net} < 10^\circ \\ \text{Sharp turn} & \phi_{max} > 10^\circ \\ \text{Steady turn} & \text{otherwise} \end{cases}$$

This makes it possible to differentiate steady turns from sharp turns, and removes some of the noise that may occur from one time step to the next. The distribution of the speed and max course change can be found in Figures 4.1 and 4.2, respectively.

A closer look will also be taken at the number of close neighbors of the target ship in each scenarios, for two reasons. The first one is that more CNs to sample from should in theory lead to better predictions for NCDM. The second reason is that a larger number of CNs indicates more clearly defined sea lanes, where ships move in a fairly predictable manner. On the other hand, target ships with few or no CNs often have sudden and unpredictable changes in course and speed.

4.2 Decision parameters

The decision parameters for the simulation are displayed in Table 4.2. Most of them are the same as in the original modified NCDM [2], but a few modifications have been made. In particular, the parameter α controlling the weighting between NCDM and CVM has been changed. Whereas $\alpha = 5000$ was chosen based on a consistency and accuracy analysis in [2], which may be suitable in COLREGS situations, this is not optimal in a long-term proactive COLAV system for several reasons.

First, a larger prediction horizon means larger uncertainties to begin with, reducing the significance of inconsistency. Second, because this is intended for a more proactive system, accuracy is arguably more important than consistency, and only a rough estimate is needed; as an example, CVM is consistent and good for short-term predictions, but its good consistency properties are not very useful in a long-term COLAV, where its prediction might be off be several kilometers. Finally, there is also a larger margin for error, as

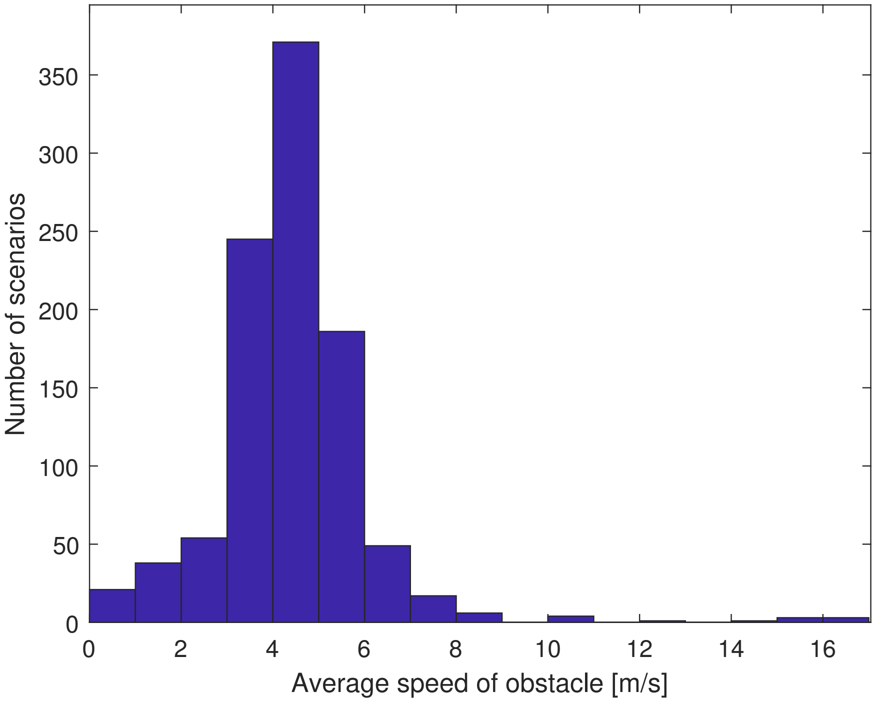
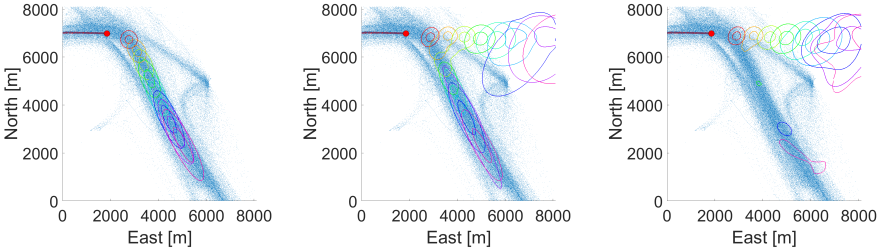


Figure 4.1: Distribution of average speed of the obstacles for the simulated test set.

the COLAV will have plenty of time to change the course should the long-term prediction turn out to be wrong. Through trial and error, $\alpha = 100$ was found to be suitable for long-term prediction. In general, a smaller prediction horizon requires a larger α, and vice versa. Figure 4.3 shows a comparison between different α.



(a) $\alpha = 100$. **(b)** $\alpha = 2000$. **(c)** $\alpha = 5000$.

Figure 4.3: Comparison of NCDM with different choices of α.

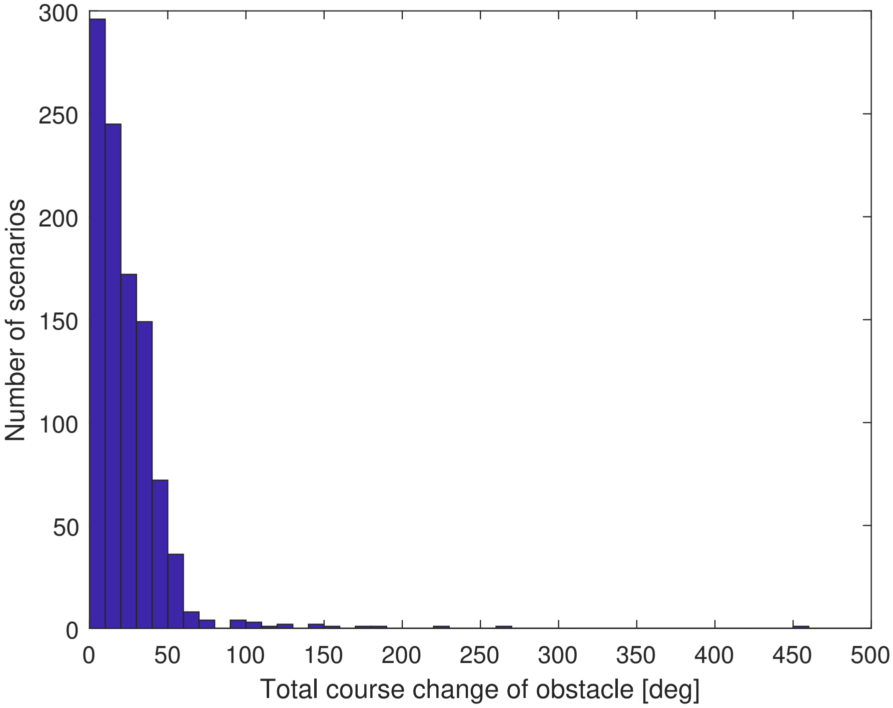


Figure 4.2: Distribution of total course change of the obstacles for the simulated test set.

Another major change was the choice of weights for the MPC, making it slightly more proactive without making it too conservative. Also, another set of weights were necessary for CVM, since its GMMs are spherical and do not vary in size nor shape.

Both methods were tuned iteratively using the test set, until a satisfactory and somewhat similar performance with respect to number of COLREGS situations avoided was achieved. With one of the metrics fixed, a fair comparison between NCDM and CVM could then be made by considering the total deviation from the desired path.
Although tuning using the test set may in some cases lead to "overfitting", the sample size is fairly large and should thus be representative of the method's actual performance.

Finally, the frequency at which the MPC is updated has been changed to $f_{mpc} = \frac{1}{300}\ \frac{1}{\mathrm{s}}$, i.e. every five minutes.

4.3 Simulation

The simulations were done using the parameters in Table 4.2, using an Inter Core i7-6700 processor @ 3.40GHz, running Matlab 2018a on a 64-bit operating system. The NLP is

Parameter	Value	Description
h	20 s	Time step
N_p	45	Prediction steps
N_c	45	Control intervals
f_c	1/300 1/s	Frequency of trajectory update
U_{min}	0 m/s	Minimum SOG
U_{max}	18 m/s	Maximum SOG
K_p	0.02	Position error weight
$K_{\dot{u}}$	1	Surge rate weight
K_r	100	Yaw rate weight
$[a_{\dot{U}}, b_{\dot{U}}]$	$[8, 2.5 \times 10^{-4}]$	SOG rate penalty parameter
$[a_r, b_r]$	$[112, 6.25 \times 10^{-4}]$	Yaw rate penalty parameter

Table 4.1: Decision parameter values for the COLAV system.

Parameter	Value	Description
n	3	Number of points in each sub-trajectory
t_{sub}	60 s	Time between points in each sub-trajectory
t_{pred}	20 s	Time between points in prediction tree
r_c	100 m	Search radius for CNs
σ_a	0.1 m	Noise covariance parameter
$N_{1,1}$	200	Number of children for root node in prediction tree
$N_{k,j}$	1	Number of children for all other nodes
λ_{ncdm}	25000	Obstacle padding weight parameter for NCDM
λ_{cvm}	70000	Obstacle padding weight parameter for CVM
c_{max}	8	Maximum number of components in GMM
M	500 m	Minimum Euclidean distance between means

Table 4.2: Decision parameter values for the NCDM.

solved using the CASADI [22] framework with the IPOPT solver.

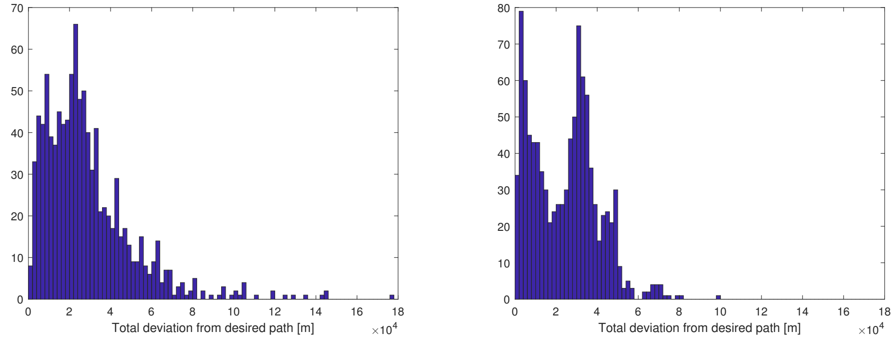
Chapter 5

Results

5.1 Quantitative proactive COLAV assessment

The overall results are displayed in Table 5.1. After tuning, both NCDM and CVM managed to avoid COLREGS situations roughly 70% of the time. Overall, it seems that CVM performs slightly better than NCDM, avoiding 15 more COLREGS situations with 6000m less average total deviation.

The median total deviation for NCDM is smaller than for CVM, suggesting that for the set of parameters used, CVM generally keeps a greater distance to the target ship. The much larger average total deviation in NCDM is likely due to the method having much larger variance in the shapes of the GMMs, leading to more extreme values on the far end of the scale. This is also reflected in Figure 5.1, which shows the distribution of the total deviation for both NCDM and CVM.



(a) NCDM.

(b) CVM.

Figure 5.1: Distribution of total deviation from path for both NCDM and CVM.

The large number of unavoided COLREGS situations for both methods can in part be explained by irregular and unpredictable obstacle behavior. Particularly in areas with sparse data, sudden changes in course and speed are not uncommon, and out of 1000 scenarios, 59 had a course change greater than 175° over the span of 100s. Since the proactive COLAV system uses an update period of 5 minutes, such scenarios require an extremely conservative set of weights to avoid a COLREGS situation. Another reason is the presence of static obstacles, which may force the trajectory of the ownship to be closer to the target ship than normal. Several other factors will be further investigated in the qualitative section. The following sections will investigate the three categories individually, to attempt to uncover strengths or weaknesses of either method.

Metric	CVM	NCDM
COLREGS situations avoided	269/1000	286/1000
Median total deviation from path	26252m	23823m
Average total deviation from path	24614m	28523m

Table 5.1: Overall test results for CVM and NCDM for 1000 test scenarios.

5.1.1 Straight path scenarios

Because ships tend to take the most direct route to minimize fuel consumption, the straight path category is by far the largest one. The results in Table 5.2 show that CVM performs significantly better than NCDM; in fact, it is the category with the largest difference in performance. This is not very surprising, as the CVM's predictions are near perfect regardless of what the AIS data might suggest, barring changes in speed of the target vessel. For the total deviation from path, the trend is the same as the overall results; a large average and smaller median for NCDM, and the opposite for CVM.

Metric	CVM	NCDM
COLREGS situations avoided	161/685	183/685
Median total deviation from path	28938m	24470m
Average total deviation from path	26484m	29490m

Table 5.2: Test results for CVM and NCDM for the scenarios where the obstacle moves in a straight line.

5.1.2 Steady turn scenarios

The NCDM has a higher average total deviation than CVM, but otherwise the performances of the methods are fairly similar (see Table 5.3). This is because compared to the straight category, the accuracy of CVM starts to fall off. The percentage of scenarios resulting in COLREGS situations is higher for both methods.

Metric	CVM	NCDM
COLREGS situations avoided	38/125	41/125
Median total deviation from path	24368m	23526m
Average total deviation from path	24322m	27811m

Table 5.3: Test results for CVM and NCDM for the scenarios where the obstacle makes a steady turn.

5.1.3 Sharp turn scenarios

The sharp turn category is the only one where NCDM seems to perform better than CVM, with 62 to 70 COLREGS situations. This corresponds well with intuition, since it can predict turns where CVM cannot, as well as indicate a larger uncertainty where irregular behavior is common. Also, unlike in the other categories, Table 5.4 show that the median and average total deviation is significantly smaller for CVM than NCDM, which suggests that CVM often reacted too late to the turn.

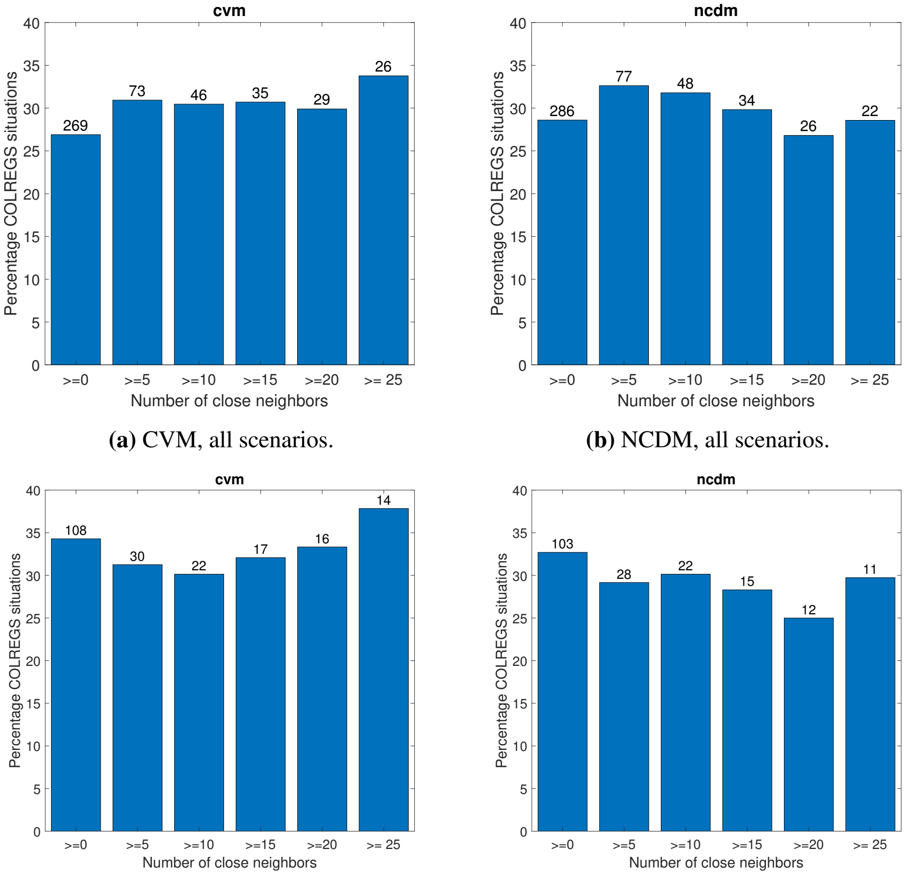
Metric	CVM	NCDM
COLREGS situations avoided	70/190	62/190
Median total deviation from path	13881m	21209m
Average total deviation from path	18064m	25601m

Table 5.4: Test results for CVM and NCDM for the scenarios where the obstacle makes a sharp turn.

5.1.4 Effect of number of close neighbors

Figures 5.2a, 5.2b shows the number of close neighbors at the first time step vs. percentage of COLREGS situations, for both CVM and NCDM. Overall, it may look like the NCDM performs slightly better in scenarios where the target ship has more close neighbors. However, the difference is marginal, and may be due to statistical noise.

In Figures 5.2c, 5.2d, the straight-path scenarios have been removed, which represent roughly two-thirds of the scenarios. CVM shows no improvement with increasing number of neighbors, in fact, the percentage of COLREGS situations increases slightly. NCDM seems to improve slightly with more neighbors, however, the sample size is relatively small, especially for scenarios with a large number of CNs, and the decrease is not very significant. It seems that having more close neighbors does little or nothing to improve NCDM in a proactive COLAV system.



(a) CVM, all scenarios.

(b) NCDM, all scenarios.

(c) CVM, excluding straight path scenarios.

(d) NCDM, excluding straight path scenarios.

Figure 5.2: Number of close neighbors vs. percentage of scenarios resulting in COLREGS situations. The number above each bar is the number of COLREGS situations. For NCDM, there seems to be a slight decrease in percentage of COLREGS situations for a larger number of close neighbors.

5.2 Qualitative proactive COLAV assessment

In this section, a closer look is taken at a handful of scenarios which highlight advantages and shortcomings of NCDM compared to CVM. One thing to be aware of for these simulations is that some of the scenarios show the ownship dangerously close to the target ship. This will never happen in practice, however, since the method is intended to be used as part of the four-layer architecture described in Chapter 1. When close, the lower level will recognize and correctly deal with COLREGS situations according to their respective rules.

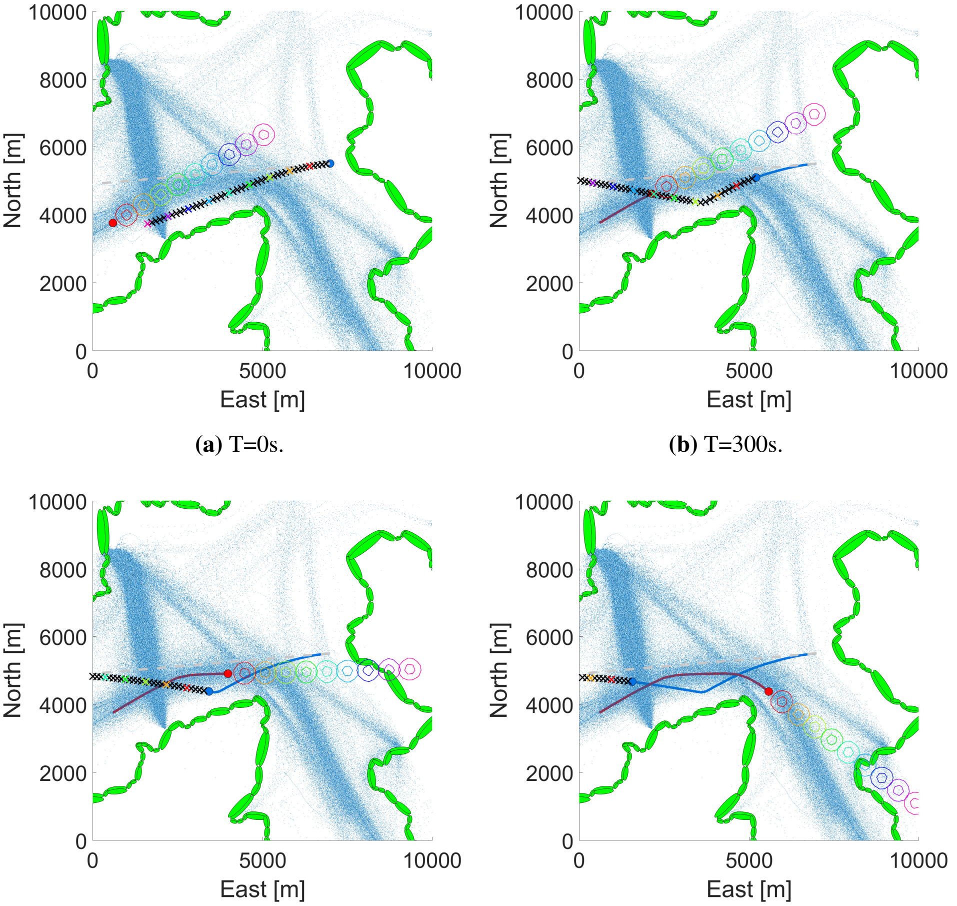
In the figures of the following scenarios, the small blue dots are AIS messages, the blue dot is the ownship, and the red dot is the target ship. X-es mark the positions planned for the MPC at different control intervals, and the colored contours of corresponding color are the predictions of the future positions of the target ship at the same time step. The inner and outer contour represent 1 and 2 standard deviations for the predictions, respectively. The small green ellipses are the inequality constraints representing the coastline, and the grey line represents the desired trajectory.

5.2.1 Case 1: Predicting a turn in a popular sealane

This case presents modified NCDM's ideal and intended behavior, highlighting the advantage of using historical data when making proactive maneuvers. The ownship is heading in the direction of the target ship, with $U_d = 6$ m/s. In other words, if no further actions are taken, it will enter a head-on situation with the other ship.
The trajectory of the ownship using CVM is shown in Figure 5.3. Based on the target ship's velocity at T=0, the algorithm opts for an initial left turn, which is reasonable given the current course and speed of the other ship. However, as seen in the second picture on the same figure, the prediction by CVM turns out to be poor, as the target ship makes a steady turn to the right. This eventually leads to a head-on situation. The scenario was tested with a large range of different weights, all leading to a head-on situation with the other ship.
The same scenario using modified NCDM is shown in Figure 5.4. The AIS data reveal that the target ship is traveling along a popular sea lane, and the method predicts that it will make a starboard turn, prompting the COLAV system to plan a proactive maneuver towards starboard. As a result, the ownship passes the target ship at a large distance, removing the necessity for later maneuvering according to a head-on situation.
Had the target ship kept its course or maneuvered to the left, the situation would have been the opposite, and the modified NCDM case would have resulted in a COLREGS situation. However, by design, the NCDM would be correct most of the times in these types of situations.

5.2.2 Case 2: Prediction of branching sea lanes

The scenario in Figure 5.5 illustrates modified NCDM's ability to identify a split in sea lanes, as well as problems that may arise during optimization when the GMMs become more complex. By $T = 300$s, approximately ten minutes before the target ship has reached the branching, it has already been identified. As time passes and the target ship keeps its



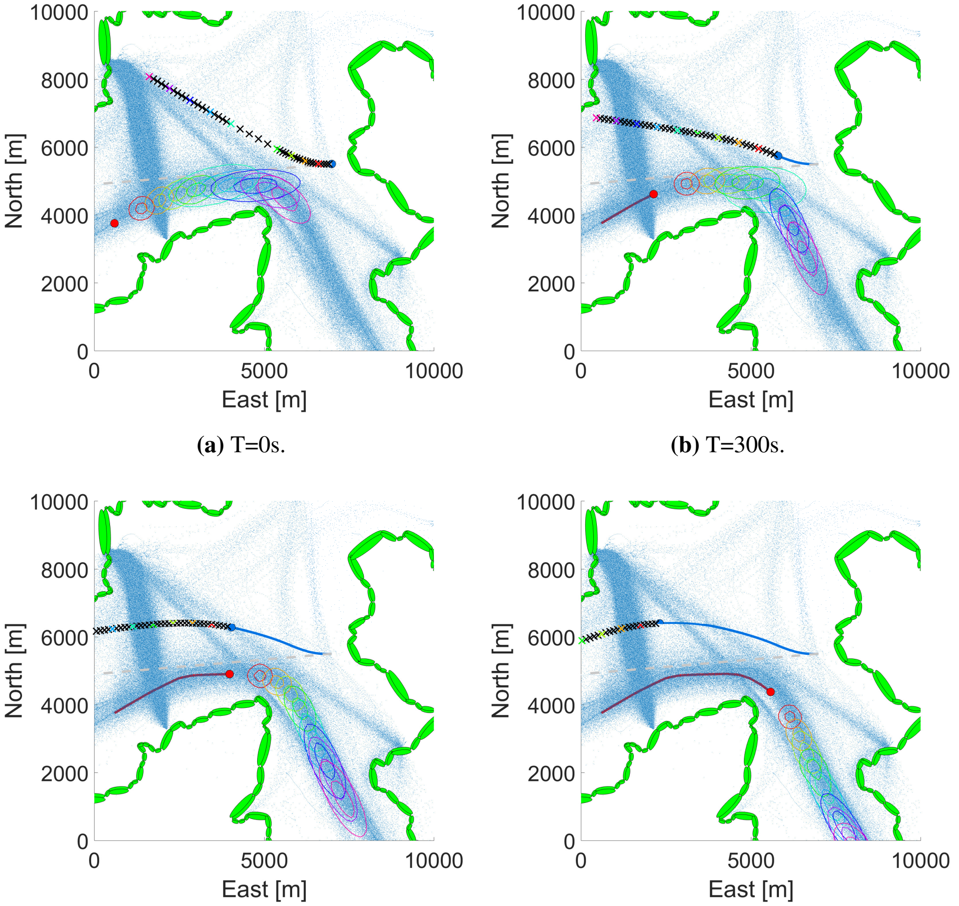
(a) T=0s.

(b) T=300s.

(c) T=600s.

(d) T=900s.

Figure 5.3: Scenario using the CVM method. The the COLAV system plans and executes a turn to port, a relatively safe maneuver at that distance. However, the target ship makes a turn, and the ships end up in a COLREGS situation. Small blue dots are data points, the large blue dot is the ownship, the red dot is the target ship and green ellipses are constraints representing the coastline. The desired trajectory is marked in gray. Shown here are time T=0s, T=300s and time=600s.



(a) T=0s.

(b) T=300s.

(c) T=600s.

(d) T=900s.

Figure 5.4: The NCDM method is able to exploit the pattern revealed by the AIS data, and predicts that the target ship likely intends to make a turn to starboard. The COLAV system responds by making a turn to starboard as well, avoiding any COLREGS situation. Small blue dots are data points, the large blue dot is the ownship, the red dot is the target ship and green ellipses are constraints representing the coastline. The desired trajectory is marked in gray. Shown here are time T=0s, T=300s and time=600s.

course, the method deems a turn to port less and less likely, which is also reasonable given the data at hand.

Interestingly, the COLAV system plans a path right between the sea lanes, a maneuver which is neither proactive nor intuitive for surrounding ships to understand. A closer inspection reveals that the planned path is only a local optimum, a common problem in non-convex optimization, and that a much better route with lower cost can be found by choosing a different initial guess (see red dotted line in Figure 5.6). It seems that with modified NCDM, the COLAV system would benefit from solving the optimization problem with multiple initial guesses.
The same scenario with CVM is shown in Figure 5.7. The COLAV system plans a turn to port already in T=0s, and as the target ship and the GMMs get closer, the planned trajectory is forced further away from the desired path. The ownship crosses in front of the target ship at T=900s, but keeps a clear distance and manages to avoid any COLREGS situation.

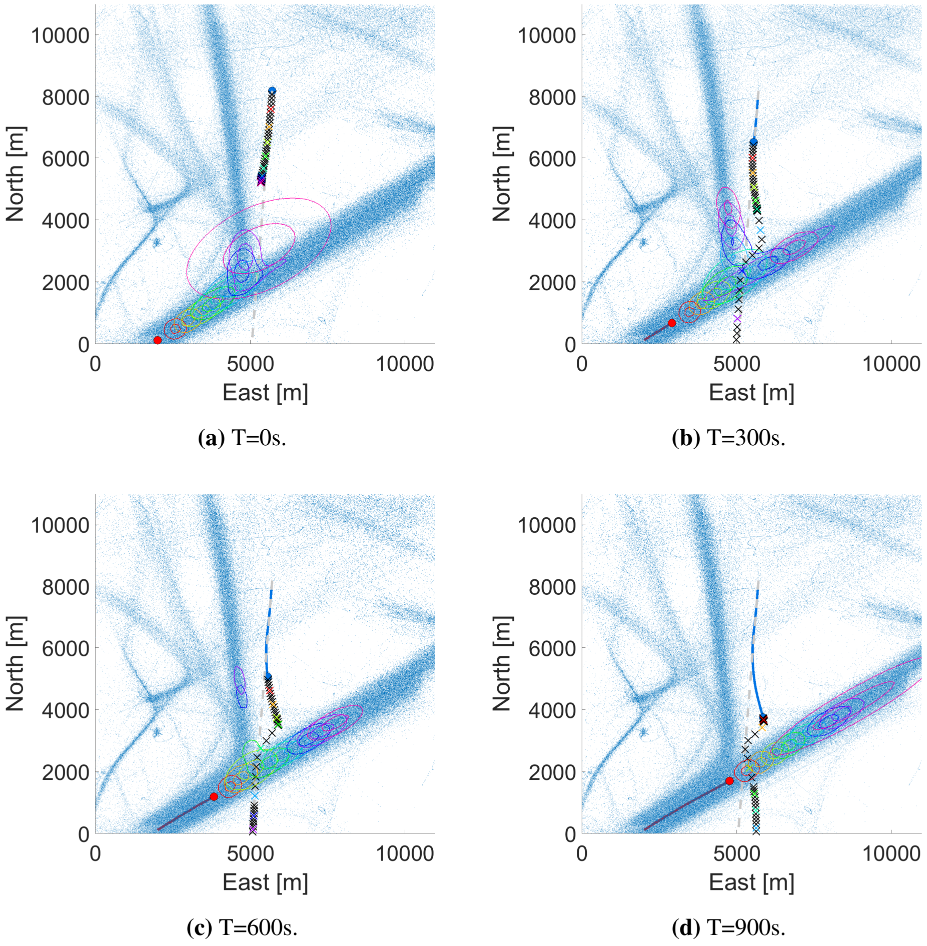
5.2.3 Challenges with framework

There are also some general issues related to the framework surrounding NCDM and the method itself. Several are shown in Figure 5.8, which is a particularly challenging scenario; the target ship moves unpredictably at 10-30 m/s, and there are a large number of small islands present.

The first issue is that there are AIS messages on the islands themselves, which is caused by interpolation errors and inaccuracies in the messages themselves. This can be seen in the bottom two pictures, where the obstacle seemingly passes straight through land. Another interpolation scheme involving obstacles, e.g. using A*, could help solve this problem, as well as data cleaning to remove messages on land.

The second one has to do with the land constraints, in that with sufficient speed, they can be "jumped" in a single time step. This can be seen in the third picture, where the ownship reduces its speed significantly in order to make it less costly to increase it to pass through the constraint. It is a problem that is inherent with using inequality constraints to represent obstacles. One possible solution is to decrease the size of the time step, at the cost of extra computations.

Finally, since it has no notion of COLREGS rules and concepts such as stand-on and give-way, it will always act as the give-way vessel. This is particularly problematic when another ship approaches from behind, as in Figure 5.8c. Essentially, the ownship will stray from its desired trajectory as long as the target ship is travelling along the same sea lane at roughly the same speed, even when it is far away from entering an overtaking situation.



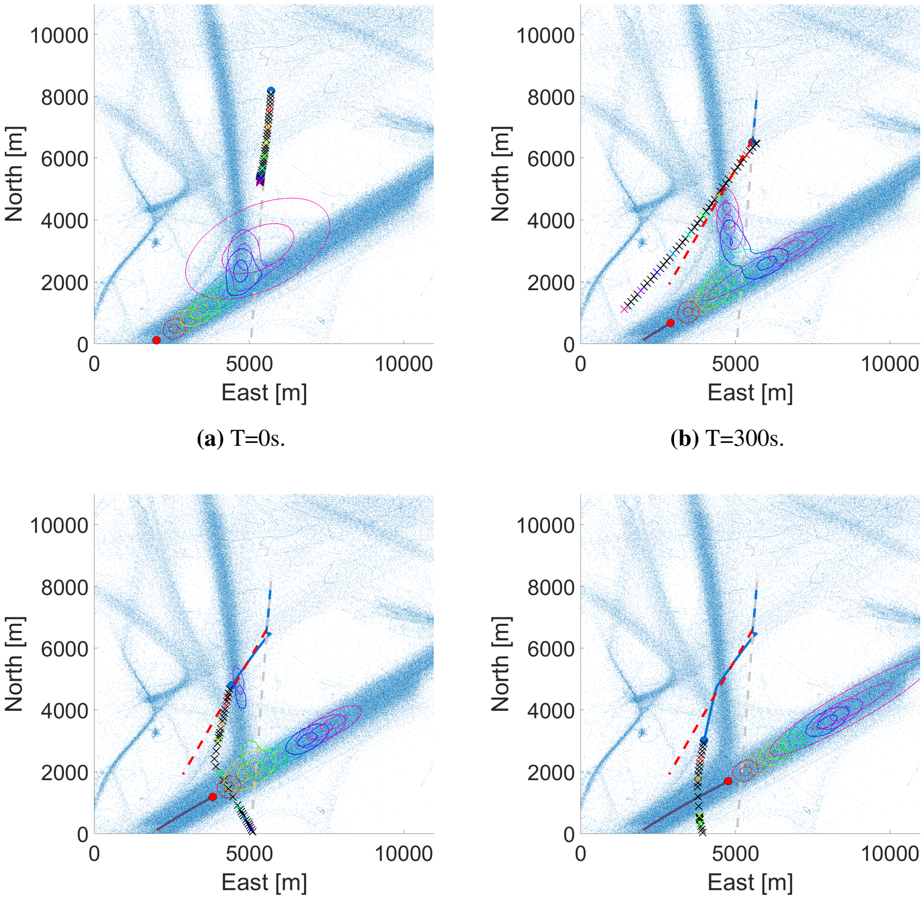
(a) T=0s.

(b) T=300s.

(c) T=600s.

(d) T=900s.

Figure 5.5: NCDM is able to detect branching sea lanes. In this scenario, however, the COLAV system has reached a local optimum, leading to a sub-optimal trajectory. A different initial guess produces a much better result (see Figure 5.6. Small blue dots are data points, the large blue dot is the ownship, and the red dot is the target ship. The desired trajectory is marked in gray. Shown here are time T=0s, T=300s, T=600s and time=900s.



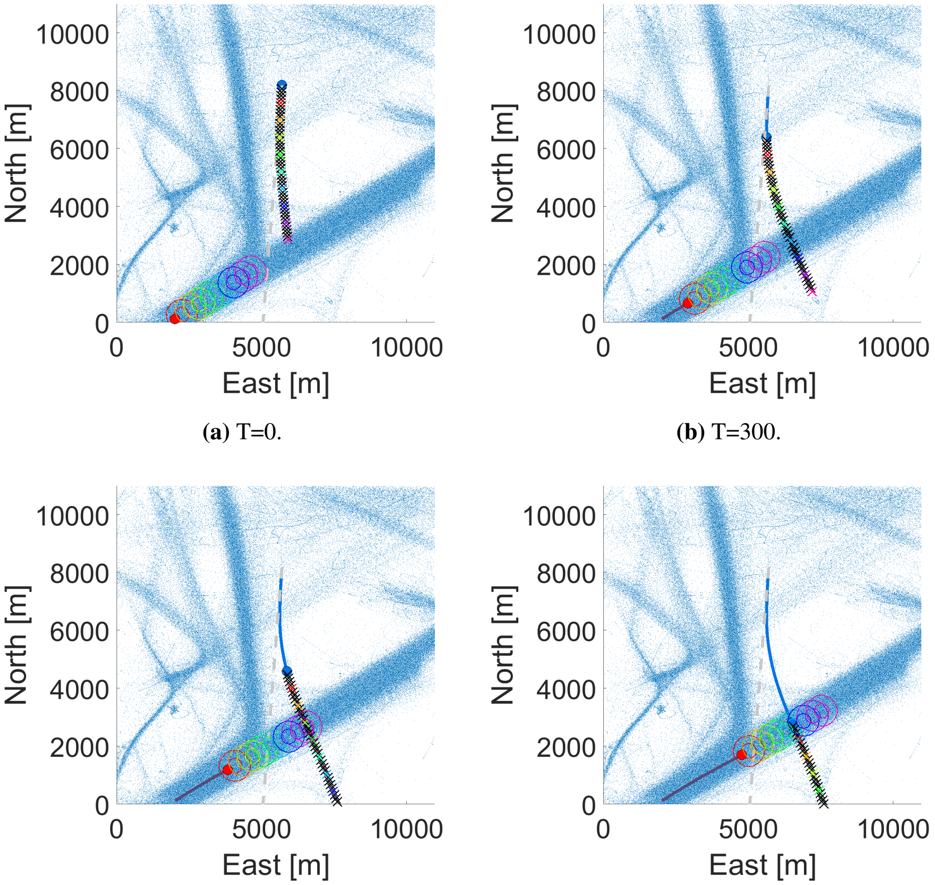
(a) T=0s.

(b) T=300s.

(c) T=600s.

(d) T=900s.

Figure 5.6: Scenario using the modified NCDM method, and a new initial guess at T=300s (red dotted line). The new resulting trajectory has a lower cost, and is much better. Small blue dots are data points, the large blue dot is the ownship, and the red dot is the target ship. The desired trajectory is marked in gray. Shown here are time T=0s, T=300s, T=600s and time=900s.



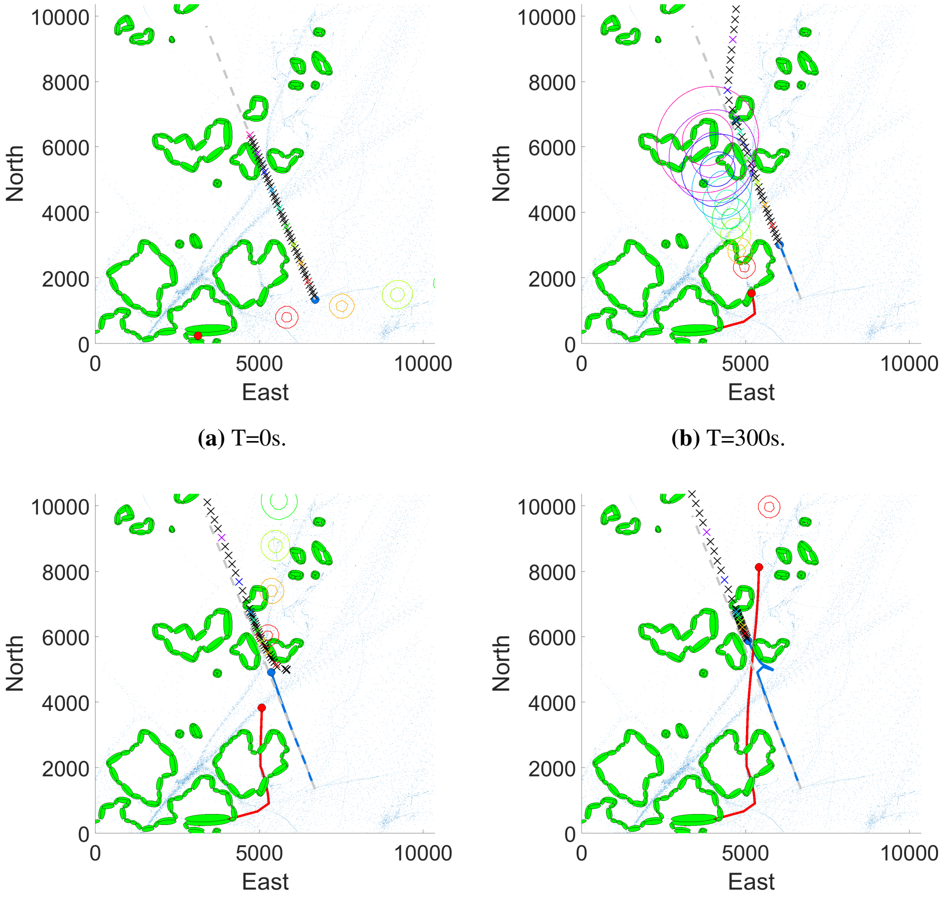
(a) T=0.

(b) T=300.

(c) T=600.

(d) T=900.

Figure 5.7: Scenario using CVM. The method cannot predict the branching sea lane, but makes an acceptable manoeuvre nonetheless. Small blue dots are data points, the large blue dot is the ownship, and the red dot is the target ship. The desired trajectory is marked in gray. Shown here are time T=0s, T=300s, T=600s and time=900s.



(a) T=0s.

(b) T=300s.

(c) T=600s.

(d) T=900s.

Figure 5.8: A challenging scenario which highlights some of the weaknesses of the framework. In particular, interpolation error leading to the target vessel (red) travelling on land, and the COLAV system planning a route which "jumps" over the green elliptical constraints. Also, the ownship (blue) reacts pre-emptively to being overtaken, which is not desirable.

Chapter 6

Discussion

There are inherent challenges with comparing methods which require tuning of parameters, since a better set of weights may have been used for one of the methods. Tuning is particularly difficult when one considers how the effect of the parameters are coupled, the large variance in types of scenarios, and that the goal is to avoid COLREGS situations in a "proactive" manner. It is therefore difficult to draw any definite conclusions about the overall performance of NCDM vs. CVM. Nevertheless, the results highlight some of their main strengths and weaknesses.

In the quantitative tests, CVM performed better than NCDM overall. This is not surprising, as the straight path category represented two-thirds of the scenarios. In other words, much of CVM's good results is attributed to the nature of ship movement, rather than its predictive power. This was also reflected in the results of the sharp turn category, where NCDM was shown to have better performance.

However, taking into account the long prediction horizon, and the fact that CVM can only predict straight lines, the difference between the methods was relatively small. NCDM was shown to have far better predictive power than CVM in [2], thus the biggest area of improvement is likely within the COLAV system itself.

One possible reason for this is the way the predictions are embedded in the COLAV system. The MPC uses the GMMs directly in the cost function, which leads to trajectories that are a set distance from the target ship. While this sometimes is sufficient, the current control objective does not match the metrics used, leading to cases where it keeps a safe distance, but fails anyway because it has no concept of COLREGS situations. Thus, in order to make full use of the predictions in a proactive COLAV system, it might be necessary to develop a cost term which maps the GMMs to DCPA and TCPA.

Another problem with using GMMs directly in the cost function, is that they vary significantly in size and shape. This makes tuning the weights in the cost function K_p, $K_{\dot{U}}$, K_r, and K_{gmm} difficult, since even similar scenarios can have very different GMMs depending on the underlying data distribution. A large weight must be chosen to accommodate for narrow GMMs, leading to very conservative trajectories for larger weights. This was also evident in the quantitative results, where NCDM consistently had a much larger average than median in all categories, caused by a few very conservative scenarios. With that being said, the proactive COLAV system does show some promise through case 1 and case 2, where it makes proactive manoeuvres based on the underlying data, and successfully avoids any COLREGS situations.

Finally, the testing of the method does not factor in how other ships might react. This means that some of the test results are conservative with respect to reality, since it is unlikely that the target ship would suddenly alter its course towards the ownship, creating a COLREGS situation.

In the future, there is potential to extend the method to multi-target scenarios. With the current framework, the implementation aspect is fairly straightforward. However, the problem of representing predictions in the COLAV system needs to be addressed first. Tuning is already difficult for single-target scenarios, and for multiple ships and potentially overlapping GMMs, the task quickly becomes unfeasible.

In general, making accurate long-term predictions also becomes increasingly challenging with more ships involved, as their actions are likely to depend on each other. There is also the problem of solving the NLP; as already seen in the case with branching of sea lanes, multiple-component GMMs makes the optimization problem considerably more difficult, and requires multiple initial guesses to obtain a globally optimal solution. However, considering the long prediction horizon, the accuracy and quality of the solution becomes less crucial. Overall, the aim is to recognize potentially difficult situations and take proactive action to avoid them.

Chapter 7

Conclusion and future work

The performance of a proactive COLAV system based on NCDM has been tested, and a new method for generating coastal constraints from map data has been developed.
The COLAV system's ability to make proactive maneuvers was then evaluated and compared to the performance of CVM, by testing both methods on 1000 randomly sampled collision scenarios, and measuring the number of COLREGS situations avoided and the total deviation from the desired path. The results suggest that CVM performs significantly better in scenarios where the target ship moves in a straight line, while NCDM is only slightly better in scenarios where the target ship has sharp turns. On average, NCDM tends to be more conservative.

The qualitative analysis highlighted some of the strengths and weaknesses of using NCDM in a proactive COLAV system. Its biggest issue is that the control objective in the MPC does not match one of the metrics used, namely number of COLREGS situations avoided. This leads to situations where the NCDM correctly predicts the course change, but steers into a COLREGS situation anyway because it has no notion of DCPA and TCPA. Nevertheless, the qualitative analysis demonstrated situations where a COLAV system based on NCDM can be used to act proactively where CVM cannot, as well predict and respond reasonably to branching of sea lanes.
For future work, the following tasks are proposed:

- Development of a new method to map GMMs to TCPA and DCPA.
- Testing of proactive COLAV system in a multi-layer architecture.

Bibliography

[1] S. Hexeberg, "AIS-based Vessel Trajectory Prediction for ASV Collision Avoidance," *NTNU*, 2017.

[2] B. R. Dalsnes, "Long-term Vessel Prediction Using AIS Data," *NTNU*, 2018.

[3] B. O. H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, "Hybrid Collision Avoidance for ASVs Compliant with COLREGs Rules 8 and 13–17," 2019. To be submitted to Frontiers in Robotics and AI.

[4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, pp. 23–33, 1997.

[5] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, pp. 760–772, 1998.

[6] P. E. Har, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, 1968.

[7] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Technical Report. Computer Science Department, Iowa State University*, October 1968.

[8] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, pp. 110–119, Jan 2014.

[9] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 3407–3422, Dec 2016.

[10] Database of Global Administrative Areas (GADM), "Administrative areas in Norway, level 1," 2019.

[11] International Maritime Organization, “Convention on the International Regulations for Preventing Collisions at Sea, 1972,” 2016.

[12] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, “Autonomous Parking Using Optimization-Based Collision Avoidance,” in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4327–4332, Dec 2018.

[13] R. Y. Xu and M. Kemp, “Fitting multiple connected ellipses to an image silhouette hierarchically,” *IEEE Trans Image Process*, pp. 1673–1682, Dec 2010.

[14] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher., “Direct Least Squares Fitting of Ellipses,” *IEEE Trans. PAMI*, vol. 21, pp. 476–480, 1999.

[15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, , and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36 (6), pp. 789–814, 2000.

[16] A. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 2018.

[17] B. O. Eriksen and M. Breivik, “MPC-based Mid-level Collision Avoidance for ASVs using Nonlinear Programming,” 08 2017.

[18] P. K. E. Minne, “Automatic testing of maritime collision avoidance,” *NTNU*, 2017.

[19] U.K. Marine Accident Investigation Branch, “Safety digest 3/2000,” *Marine Accident Investigation Branch*, 2000.

[20] M. McNicholas, *Maritime security, An Introduction, Second Edition*. Elsevier - Health Sciences Division, 2016.

[21] C. H. Allen, *Farwell’s Rules of the Nautical Road, 8th Edition*. Naval Institute Press, 2005.

[22] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, 2018.