



Norwegian University of
Science and Technology

Long-term Vessel Prediction Using AIS Data

Bjørnar Ryen Dalsnes

Master of Science in Cybernetics and Robotics

Submission date: June 2018

Supervisor: Edmund Førland Brekke, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem description

For maritime collision avoidance (COLAV) systems it is useful to have some idea about how other ships are likely to move during a time horizon of several minutes. While such predictions hardly can be relied upon for a reactive COLAV method, they may nevertheless enable a proactive COLAV method to make good decisions so that dangerous situations never occur in the first place. This MSc thesis will contribute to the development of such prediction methods, building on a previous specialization project written by the candidate. Tasks for the MSc thesis include:

- Further refine the Neighbor Course Distribution Method (NCDM) to include the possibility that a vessel moves straight ahead.
- Develop metrics for deciding the number of components in the Gaussian mixture representing the uncertainty of the NCDM.
- Assess the suitability of the NCDM as part of a COLAV system.

Preface

This master thesis is written as part of the study program Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). I would like to thank my supervisor Edmund F. Brekke together with my co-supervisor Bjørn-Olav H. Eriksen for the advice and guidance they have provided me with throughout this project.

Much of the work in this thesis is based upon the master thesis of Simen Hexeberg [1] and an article written by my co-supervisor Bjørn-Olav and Morten Breivik [2]. I have therefore benefited from some resources from these earlier works. The same dataset that was utilized in [1] is also used in this thesis. I have also been provided with the MATLAB code from [2] and have used this as a foundation for further development.

For the most part I have been working independently on this thesis. My supervisors have primarily assisted me by providing feedback on the thesis.

Bjørnar Ryen Dalsnes
Trondheim, June 26, 2018

Abstract

In recent years there has been much research dedicated to the development of autonomous surface vehicles (ASVs), including large-scale autonomous ships. An important part of this research is to ensure that an ASV can safely operate in areas with other marine traffic. Therefore it is necessary that an ASV is equipped with a collision avoidance (COLAV) system. A vital part of this system is the ability to predict the trajectories of other vessels in order to avoid them. This thesis is focused on improving recently developed prediction techniques and investigate whether they are suitable for use within a COLAV system.

In this thesis, the neighbor course distribution method (NCDM), which makes predictions based on automatic identification system (AIS) data and represents obstacles as Gaussian mixture models (GMMs), is improved upon to make it more suitable for use in COLAV. This method has shown promise, but as it is a data-driven method it performs poorly when little data is available and is often overconfident. Therefore, a modified NCDM, which assumes near constant velocity in low data density areas, is developed in this thesis to mitigate these problems. The modified NCDM shows a significant improvement in covariance consistency, although there is some loss in accuracy.

Furthermore, the modified NCDM is used for obstacle prediction within a COLAV algorithm based on model predictive control (MPC). This is benchmarked against using the same COLAV algorithm, but assuming that the vessels maintain a constant velocity and modeling these obstacles as circular constraints. The new method is found to be advantageous in areas where vessels historically have maneuvered often. The results also show that the modifications to the NCDM makes the algorithm significantly more suitable for COLAV than the original NCDM in areas with low data density.

Sammendrag

I de siste årene har det vært mye utvikling dedikert til utviklingen av autonome overflate fartøyer (ASVer), inkludert autonome skip på stor skala. En viktig del av denne forskningen er å forsikre at en ASV opererer trygt i områder med annen marin trafikk. Derfor er det nødvendig at en ASV er utstyrt med et kollisjonsungåelsessystem (COLAV system). En vital del av dette systemet er evnen til å prediktere banene til andre skip for å kunne unngå dem. Denne oppgaven er fokusert på å forbedre nylig utviklede prediksjonsmetoder og undersøke om de er egnet for bruk i et COLAV system.

I denne oppgaven er metoden "neighbor course distribution method" (NCDM), som gjør prediksjoner basert på "automatic identification system" (AIS) data og representerer hindringer som Gaussiske mikstur modeller (GMMer), forbedret ved å gjøre den mer egnet for bruk i COLAV. Denne metoden har vist seg lovende, men siden den er en datadrevet metode så gir den dårlige resultater i områder med lav datatetthet og er ofte for selvsikker. Derfor er en modifisert NCDM, som antar nær konstant hastighet i områder med lav datatetthet, utviklet i denne oppgaven for å forminske disse problemene. Den modifiserte NCDM viser en betydelig forbedring av kovarianse konsistens, selv om det er noe tap av nøyaktighet.

Videre er den modifiserte NCDM brukt for prediksjon av hindringer i en COLAV algoritme basert på "model predictive control" (MPC). Dette er målt mot den samme COLAV algoritmen, men med antagelsen om at andre båter holder en konstant fart og der hindringene er modellert som sirkulære begrensninger. Den nye motoden er vist å være fordelaktig i områder der båter historisk har manøvrert mye. Resultatene viser at modifikasjonene til NCDM gjør algoritmen betydelig mer egnet for COLAV enn den originale NCDM i områder med lav datatetthet.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Background | 1 |
| 1.2.1 | Obstacle prediction | 1 |
| 1.2.2 | Collision avoidance | 2 |
| 1.3 | Contributions | 3 |
| 1.4 | Outline | 4 |
| | | |
| I | Vessel Prediction | 5 |
| | | |
| 2 | AIS dataset | 7 |
| 2.1 | Dataset | 7 |
| 2.2 | Data structures | 7 |
| 2.2.1 | Original structure | 9 |
| 2.2.2 | New structure | 9 |
| | | |
| 3 | Prediction algorithms | 11 |
| 3.1 | Prediction tree | 11 |
| 3.2 | Constant velocity model | 11 |
| 3.3 | Neighbor course distribution method | 12 |
| 3.3.1 | Interpolation of trajectories | 14 |
| 3.4 | Constant velocity model in the NCDM | 14 |
| | | |
| 4 | Gaussian mixture learning | 17 |
| 4.1 | Gaussian mixture models | 17 |
| 4.2 | Expectation maximization | 18 |
| 4.2.1 | Model selection | 18 |
| | Akaike information criterion | 18 |
| | Bayesian information criterion | 19 |
| | Component distance | 19 |
| 4.3 | Variational Bayes theoretical background | 19 |
| 4.3.1 | Bayesian inference | 19 |
| 4.3.2 | Variational calculus | 19 |
| 4.3.3 | Kullback-Leibler divergence | 20 |
| 4.4 | Variational Bayesian inference | 20 |
| 4.5 | Variational Bayes method for GMMs | 21 |
| | | |
| 5 | Prediction results | 23 |
| 5.1 | Performance measures | 23 |
| 5.1.1 | Root mean square error | 23 |
| 5.1.2 | Consistency | 23 |
| 5.2 | Test setup | 24 |

| | | |
|-----------|--|-----------|
| 5.3 | Testing of CVM | 25 |
| 5.4 | Testing the modified NCDM | 25 |
| 5.5 | Testing of Gaussian learning methods | 27 |
| 5.6 | Comparison of methods | 30 |
| 5.6.1 | RMSE | 30 |
| 5.6.2 | Consistency | 31 |
| 5.7 | Discussion | 31 |
| II | Collision Avoidance | 33 |
| 6 | NCDM in a COLAV system | 35 |
| 6.1 | Model predictive control | 35 |
| 6.2 | MPC-based COLAV | 35 |
| 6.2.1 | ASV modeling | 36 |
| 6.2.2 | Control objective | 36 |
| 6.2.3 | Optimization problem | 37 |
| 6.2.4 | Using GMMs as obstacles | 39 |
| 6.3 | Simulation | 39 |
| 7 | COLAV results | 41 |
| 7.1 | Scenario 1: Straight line crossing curved traffic area | 41 |
| 7.1.1 | Original COLAV method | 42 |
| 7.1.2 | New COLAV method with NCDM | 43 |
| 7.1.3 | COLAV with modified NCDM | 44 |
| 7.2 | Scenario 2: Head-on in low data density area | 44 |
| 7.2.1 | Original COLAV | 44 |
| 7.2.2 | Modified NCDM | 44 |
| 7.3 | Discussion | 44 |
| 8 | Conclusion and further work | 51 |
| A | Accepted paper for the 21th International Conference on Information Fusion 2018 | 53 |

List of Figures

| | | |
|-----|---|----|
| 2.1 | Scatter plot of the ship positions in the entire dataset | 8 |
| 2.2 | Elapsed time between subsequent AIS messages | 8 |
| 2.3 | Interpolated trajectory | 10 |
| 3.1 | Prediction tree structure | 12 |
| 3.2 | Prediction from sub-trajectory. | 13 |
| 3.3 | Interpolation of prediction tree | 14 |
| 3.4 | Predictions using the NCDM with low data density | 14 |
| 4.1 | Forward and reverse KL divergence | 20 |
| 5.1 | Performance measure | 24 |
| 5.2 | Box plot of the PDF-ratios for different noise parameters | 26 |
| 5.3 | Prediction plots of the CVM with different noise covariance parameters | 27 |
| 5.4 | Box plot of the PDF-ratios for different weight parameters. | 28 |
| 5.5 | Box plot of the PDF-ratios of the different learning methods. | 29 |
| 5.6 | Median RMSE over time | 30 |
| 5.7 | Box plot of the PDF-ratios for the method comparison tests. | 31 |
| 6.1 | Illustration of an MPC for a single-input single-output system. | 36 |
| 6.2 | GMM avoidance proof of concept | 40 |
| 7.1 | Scenario 1 | 42 |
| 7.2 | Scenario 1 with the original COLAV method | 43 |
| 7.3 | Scenario 1 with the new COLAV method and NCDM | 45 |
| 7.4 | Scenario 1 with the new COLAV method and modified NCDM. | 46 |
| 7.5 | Scenario 2 | 47 |
| 7.6 | Scenario 2 with the original COLAV method. | 48 |
| 7.7 | Scenario 2 with the new COLAV method and modified NCDM. | 49 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Description of AIS parameters | 7 |
| 5.1 | Decision parameters for NCDM | 24 |
| 5.2 | Parameters for learning methods | 27 |
| 5.3 | Results of learning method tests | 28 |
| 5.4 | Median RMSE for each test time | 30 |
| 7.1 | Decision parameters for the COLAV tests. | 41 |

Acronyms

AIC Akaike's information criterion

AIS automatic identification system

ASV autonomous surface vehicle

AUV autonomous underwater vehicle

BIC Bayesian inference criterion

CN close neighbor

COG course over ground

COLAV collision avoidance

COLREGS international regulations for preventing collisions at sea

CVM constant velocity model

DW dynamic window

EM expectation maximization

GMM Gaussian mixture model

KL Kullback-Leibler

MPC model predictive control

NCDM neighbor course distribution method

NLP nonlinear program

OU Ornstein-Uhlenbeck

PDF probability density function

RMSE root mean square error

ROT yaw rate

RRT rapidly-exploring random trees

SOG speed over ground

SPNS single point neighbor search

VB variational Bayes

VO velocity obstacle

Chapter 1

Introduction

1.1 Motivation

Ever since the industrial revolution humans workers performing simple tasks have been replaced by automated machines. In more recent years there has been much research dedicated to the automation of more complex tasks, such as driving. The motivation behind this research is twofold. First, there is the potential for increased safety. For example, many thousand die in traffic every year. This number could be drastically reduced with the introduction of reliable autonomous cars. Second, there is the economic benefits. Automation has greatly reduced the cost in manufacturing, allowing consumer to purchase cheaper products. In addition, people are freed up to spend their time on more useful or rewarding tasks.

There is reason to believe that similar advancements can be made in the maritime industry if autonomous surface vehicles (ASVs) begin to replace regular, manned vessels. It is expected that ASVs will be safer and cheaper to run [3, 4]. It is estimated that 75% - 96% of all marine accidents are due to human error [4]. It is therefore reasonable to assume that autonomous vessels can greatly reduce the number accidents and lives lost. Another problem facing the maritime industry is that there is shortage of skilled people willing to work on a ship for weeks or months at a time. There is also the problem of piracy which is not a serious threat for an ASV as the ship cannot be controlled from the ship itself and there are no potential hostages on board.

However, making an ASV safely operate together with other marine traffic is a major challenge. The vessel needs to comply with the international regulations for preventing collisions at sea (COLREGS), which serves as the "rules of the road" at sea. It is therefore necessary that ASVs are equipped with a collision avoidance (COLAV) system that not only avoids collisions with other ships, but also complies with the COLREGS. When performing COLAV with respect to other vessels, one must model how the vessels will maneuver in the future. The focus of this thesis is to investigate such prediction techniques and improve upon them, and to test whether these predictions are suitable for use within a COLAV system.

1.2 Background

1.2.1 Obstacle prediction

Most of the previous literature regarding collision avoidance for ASVs assumes that obstacles behave according to the constant velocity model (CVM), that is that they maintain a constant or near constant velocity [5, 2, 6]. However, such predictions might be sub-optimal for prediction horizons of 5 - 15 minutes, which is time horizon

of interest for COLAV.

Other prediction methods for maritime vessels from the literature are usually developed for longer-term predictions than are of interest in the context of COLAV, and instead focus on ship traffic monitoring and anomaly detection [7, 8, 9]. In [7], vessel trajectories are predicted by describing a vessel's velocity using a Ornstein-Uhlenbeck (OU) stochastic process. This was shown to greatly reduce the variance of the predicted position. Another approach for vessel movement prediction is to cluster trajectories based on historical automatic identification system (AIS) data and then assign an object's initial state to one of these clusters [9, 10, 11].

A more data-driven approach was first introduced in [12] with the single point neighbor search (SPNS). The SPNS algorithm uses historical AIS messages within an euclidean radius to obtain a predicted course and speed of the vessel. Historical messages with courses that deviate by a certain amount from the vessel's course are discarded in order to avoid influence from opposite moving vessels in the predictions. What remains are messages that have a similar position and course as that of the vessel. The median course and speed of these close neighbors are calculated and used as a predicted course and speed of the vessel. The predicted course and speed are then used to calculate the future position by a given step length parameter. The same process is then applied on the newly predicted state and this is then repeated until a trajectory of desired length is produced.

In order to handle branching of sea lanes and to estimate prediction uncertainty, the neighbor course distribution method (NCDM) was developed [1]. Whereas the output from the SPNS can be seen as a list of states which forms a single trajectory, the output from NCDM is a tree of states which forms several trajectories. Each individual trajectory is calculated in a similar manner as in the SPNS. The same set of close neighbors are used to predict the vessel's course and speed at each predicted position. However, while the SPNS predicts the course and speed as the median course and speed of the closest neighbors, the NCDM samples courses randomly from the closest neighbor set. The NCDM is thus able to predict trajectories in several branched sea lanes and it possess the ability to indicate an uncertainty measure of the predictions.

In the specialization project [13] the NCDM was further developed. The predicted position of the vessel was here represented as a Gaussian mixture model (GMM). A new data structure which utilizes the recent trajectory of the vessel was also introduced. This data structure significantly improved the consistency of the prediction, although there was no improvement in accuracy.

1.2.2 Collision avoidance

In the COLAV literature for ASVs a distinction between global and local COLAV methods is often made. A global method plans a desired path for the vessel based on the initial state, the goal state and the known obstacles at the initial time. The major drawback of global methods is the large computational cost which means a real-time implementation might not be feasible. Local methods, on the other hand, use far less computational power, and are therefore more dynamic and can be updated with new input more frequently. These methods have no overview of the global environment, but rely on sensory input from the local surroundings.

One common global method is the rapidly-exploring random trees (RRT) algorithm first introduced in [14]. This randomized algorithm grows a tree that explores the available state space. In the end, a sup-optimal, but sufficient path that avoids the specified obstacles is found from start to goal. A COLREGS compliant planner based on RRTs was developed in [15].

Local methods typically used for COLAV include the velocity obstacle (VO) method and the dynamic window (DW) algorithm. In [5] the VO method is used. The VO is the set of all velocities that the ASV can take that will eventually result in a collision, assuming that the obstacle velocity is constant. The method used considers four different scenarios where different COLREGS rules apply. A velocity outside the VO is selected based on which scenario the ASV is facing.

The DW algorithm, unlike the VO method, takes the limitations of the vehicle into account. A window containing the reachable combinations of forward and rotational speed is defined based on the state and dynamic constraints of the vehicle. The search space is thus reduced to velocities within this window. In addition, velocities that will lead to collisions with the obstacles are also restricted. The DW algorithm was developed in [16] for implementation with mobile robots. In [17] the DW algorithm was implemented for COLAV for autonomous underwater vehicles (AUVs). This algorithm was further developed for use with ASVs and tested in full-scale experiments with a radar-based tracking system in [18].

In [19] a hybrid method is implemented where one global and one local method are combined. A RRT is used to construct a path from start to goal, while the DW approach is used to make local adaptations when the sensors detect that the planned path runs into obstacles. This was demonstrated to give superior results compared to relying on either a global or local approach alone.

In [2] a COLAV method based on model predictive control (MPC) is developed. This algorithm is intended to be used in a mid-layer between a top, global layer and a bottom, local layer. The planned path from the global layer is used as input for the mid-level algorithm which outputs a new modified path for the local layer. The algorithm uses the predicted output from an MPC as a planned path. This process is described in further detail in Section 6.2. MPC is also used for COLAV in [6] with a compliance to more COLREGS rules than in [2], but the control behaviours of the ASV are limited to a finite set of discrete maneuvers as the method rely on brute force techniques. The MPC algorithm in [6] is tested in full-scale experiments using AIS for obtaining obstacle information in [20].

1.3 Contributions

The contributions of this thesis are:

- A review of previous position prediction techniques and COLAV methods.
- Modifications to the NCDM to make it rely on the CVM in areas with low data density.
- Investigation of different GMM learning techniques.
- Extensive testing of the modified NCDM with different parameters and GMM learning techniques.

- The modification of a MPC-based COLAV method which makes it able to threat a GMM as an obstacle.
- Testing of the NCDM within the modified COLAV method and a comparison with older methods.

1.4 Outline

This thesis is divided into two parts. Part I is about vessel trajectory prediction. This part begins with Chapter 2 which gives an overview of the AIS dataset used. Chapter 3 explains the prediction algorithms used in the thesis. In Chapter 4 different GMM learning methods are explored. Finally, the different prediction methods are tested and compared in Chapter 5. Part II is about the use of the prediction methods in a COLAV system. In Chapter 6 the NCDM is used for obstacle prediction within a MPC-based COLAV system. The implementation is then tested and compared to other methods in Chapter 7. A conclusion for both parts is given in Chapter 8.

Part I

Vessel Prediction

Chapter 2

AIS dataset

This chapter is heavily based on chapter 4 of [13], but is included here for completeness.

AIS is an automatic tracking system that is used for identifying and tracking vessels. It provides information such as a unique identification number, position, course and speed of a vessel. This information is shared between vessels and also transferred to base stations and satellites. The International Maritime Organization requires all international voyaging ships with gross tonnage of 300 or more, and all passenger ships to be fitted with an AIS system [21]. In addition, AIS is also used by many smaller vessels.

2.1 Dataset

In this thesis the same dataset as in [1] and [13] is used. The dataset is provided by DNV-GL and consists of almost 3 million AIS-messages from Trondheimsfjorden collected during 2015. There are six values from this dataset that are considered in this thesis. The six relevant values are explained in table 2.1.

TABLE 2.1: Description of AIS parameters

| Parameter | Description |
|--------------------|--|
| MMSI | Vessel identification number. Unique for all vessels. |
| Timestamp | Time when message was sent. Unit is days since 1.Jan 1900, 00:00. |
| Longitude | Geographic coordinate. $(-180^\circ, 180^\circ]$ |
| Latitude | Geographic coordinate. $[-90^\circ, 90^\circ]$ |
| Speed over ground | Speed of the vessel measured in knots. |
| Course over ground | Course of the vessel measured in degrees clockwise relative to true north. |

It is also worth mentioning that messages where ships are moving very slowly or standing still (speed over ground < 0.5 knots) have been removed from the dataset. The data is sorted by MMSI first and time second.

2.2 Data structures

In [13] two different data structures were used. The first of these structures was developed in [1] and contains data about a vessels' position, course and speed. The second structure was developed in [13] and contains data about a vessels' recent trajectory.

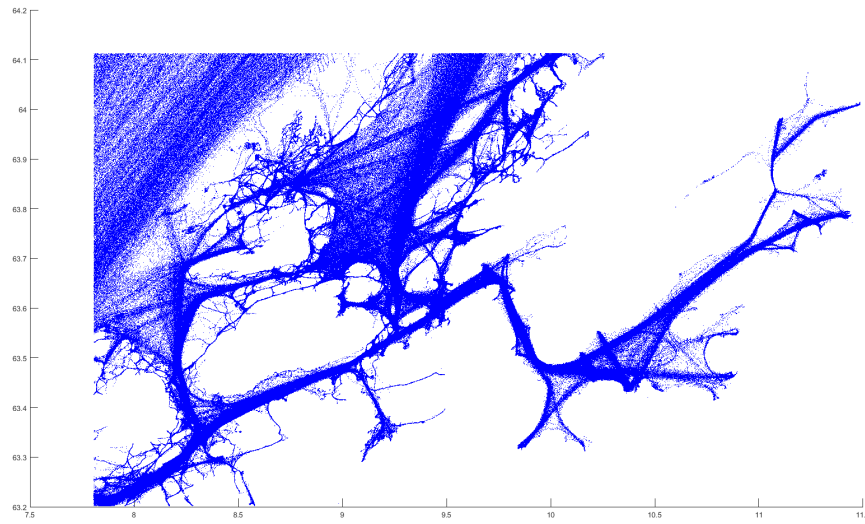


FIGURE 2.1: Scatter plot of the ship positions in the entire dataset. Shipping lanes and an outline of the geography around Trondheimsfjorden can clearly be seen.

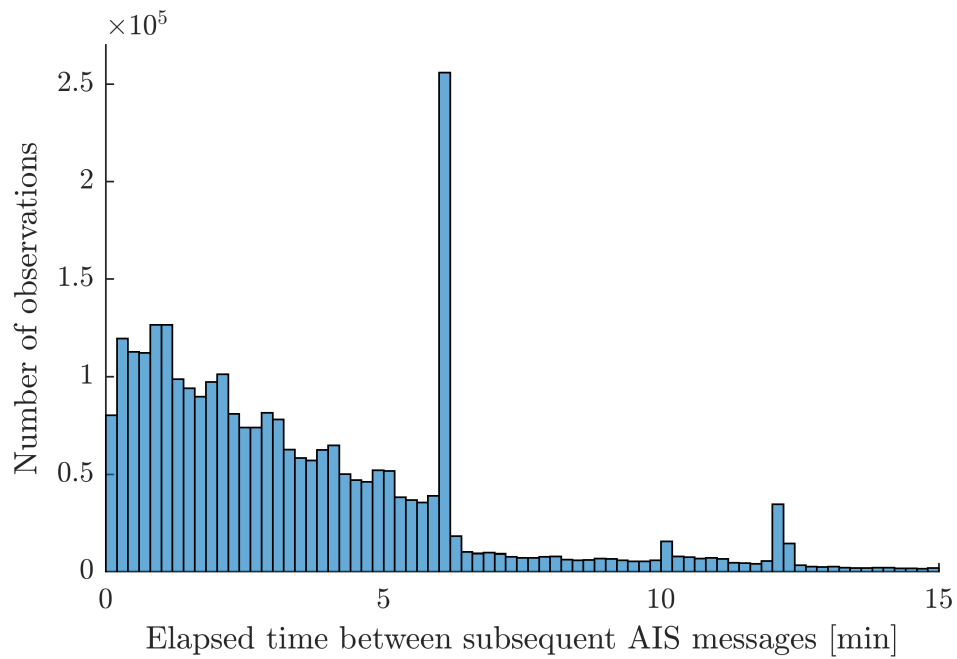


FIGURE 2.2: Elapsed time between subsequent AIS messages of the same MMSI. Courtesy of [1].

Only the second structure is used in this thesis as this structure was shown to have better covariance consistency properties [13]. However, the original structure from [1] is defined here as it is needed to derive the new data structure.

2.2.1 Original structure

The original structure consists as a list of AIS messages:

$$\mathbf{X}^A = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \dots \quad \mathbf{X}_n]^T \quad (2.1)$$

where each message is given as:

$$\mathbf{X}_i = [\text{MMSI}_i \quad t_i \quad \mathbf{p}_i^T \quad \chi_i \quad v_i] \quad (2.2)$$

where MMSI_i , t_i , \mathbf{p}_i^T , χ_i and v_i are the MMSI number, time stamp, position vector, course over ground (COG) and speed over ground (SOG), respectively. The position vector can be written as $\mathbf{p}_i^T = [\lambda_i \quad \phi_i]$ where λ_i and ϕ_i are the longitude and latitude in the WGS84 coordinate system.

2.2.2 New structure

The new structure takes the recent trajectory of a vessel into account. The data is here structured as a list of sub-trajectories. A sub-trajectory \mathbf{S} consists of n points:

$$\mathbf{S} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \dots \quad \mathbf{p}_n], \quad (2.3)$$

where \mathbf{p} is a point given by longitude and latitude. There is also an equal amount of time t elapsed between any two subsequent points. A trajectory is given by:

$$\mathbf{T} = [\mathbf{S}_1 \quad \mathbf{S}_2 \quad \dots \quad \mathbf{S}_n]^T, \quad (2.4)$$

and the entire dataset is given by a list of trajectories:

$$\mathbf{X}^B = [\mathbf{T}_1 \quad \mathbf{T}_2 \quad \dots \quad \mathbf{T}_n]^T \quad (2.5)$$

Reformatting the dataset from the original structure \mathbf{X}^A to the new structure \mathbf{X}^B involves three main steps:

1) *Find trajectories from the original dataset:* Points that belong to the same MMSI number and have less than 15 minutes between two subsequent points are considered as part of the same trajectory. The time limit makes sure that trajectories from vessels which leave and later enter the dataset window or which stay in port for a long time, are split into separate trajectories

2) *Interpolate trajectories to get new data points:* Cubic spline interpolation is used to extract new data points at a specified time interval. The new data points now form a new trajectory.

3) *Create sub-trajectories of n points:* The first sub-trajectory consists of the first n points of the new data points. This is the first row of the data structure. The next row is a new sub-trajectory shifted one point forward, thus creating a structure of partly overlapping sub-trajectories. This continues until the end of the trajectory. With this method a trajectory of k points will result in $k - n + 1$ sub-trajectories of

n points.

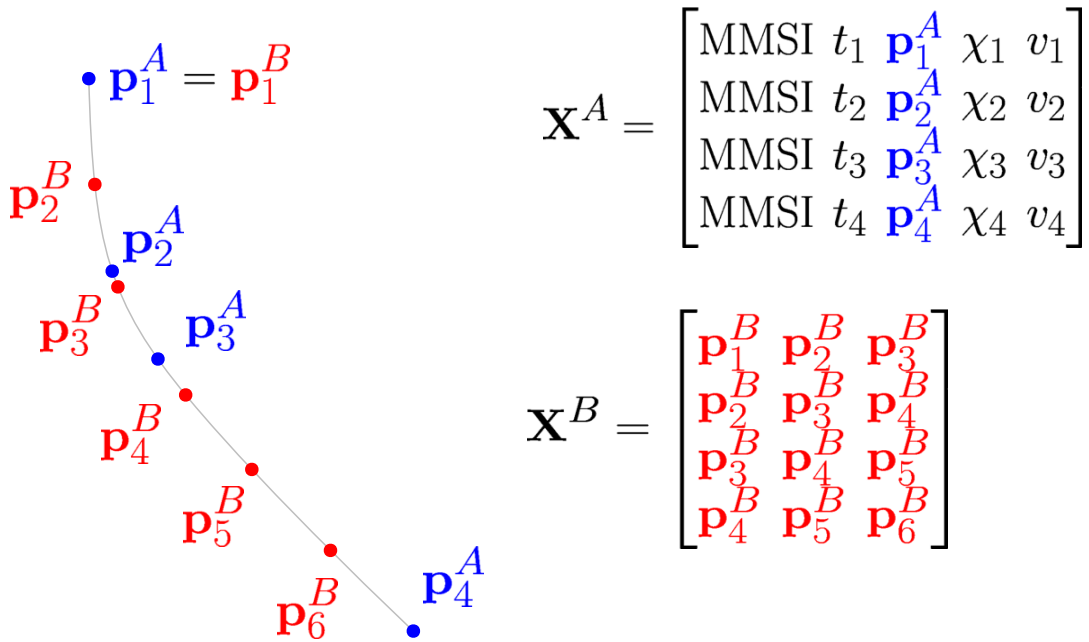


FIGURE 2.3: An example of a short trajectory represented with both data structures: The blue dots represents the ship's positions as given in the original AIS data. The red dots are the new data points obtained using interpolation. Here the sub-trajectories consist of $n = 3$ points.

Chapter 3

Prediction algorithms

In this chapter two methods for predicting the trajectories of vessels are presented. Firstly, the CVM, which assumes that the vessel will continue with a near constant course and speed, where the vessel's acceleration is modeled as a white noise process with a small variance [22]. Secondly, the NCDM, first introduced in [1], which utilizes historical AIS data. Lastly, a modified NCDM is introduced. This algorithm attempts to mitigate some weaknesses of the NCDM by relying on the CVM in certain cases.

The work done on the NCDM in [1] and in the specialization project [13] lead to a paper accepted to the 21th International Conference on Information Fusion. This paper is included in Appendix A.

3.1 Prediction tree

The NCDM gives its predictions of a vessel's future position as a prediction tree. In this thesis the CVM is also included in the tree structure to better compare and combine the two methods.

The prediction tree represents a vessel's position at time t_k by J^{max} predicted states. The predictions of these states are subject to a degree of randomness which means that states at the same level will be slightly different from each other. This is used to give a measure of the uncertainty of the prediction.

The initial state is the root node of the prediction tree. Each node represents a predicted state $\hat{\mathbf{X}}_{k,j}$, where k is the depth index and j is the width index of the tree. The value $N_{k,j}$ gives the number of child nodes for node (k, j) , while J_k is the number of nodes at level k . In this thesis the number of children for the root node is chosen to be $N_{1,1} = J^{max}$, while the number of children at all other levels are set to one. In other words, the tree has J^{max} branches which all originate from its root node. The tree is illustrated in Figure 3.1.

For a given level k in the prediction tree, a GMM can be fitted to these points to give a probabilistic, multimodal representation of the predicted position. This process is described in more detail in Chapter 4.

3.2 Constant velocity model

The discrete-time CVM method presented here assumes that a vessel will continue with its current velocity in the near future. However, some noise is also introduced to the model. The motion model from [23] is used with some minor changes to the noise

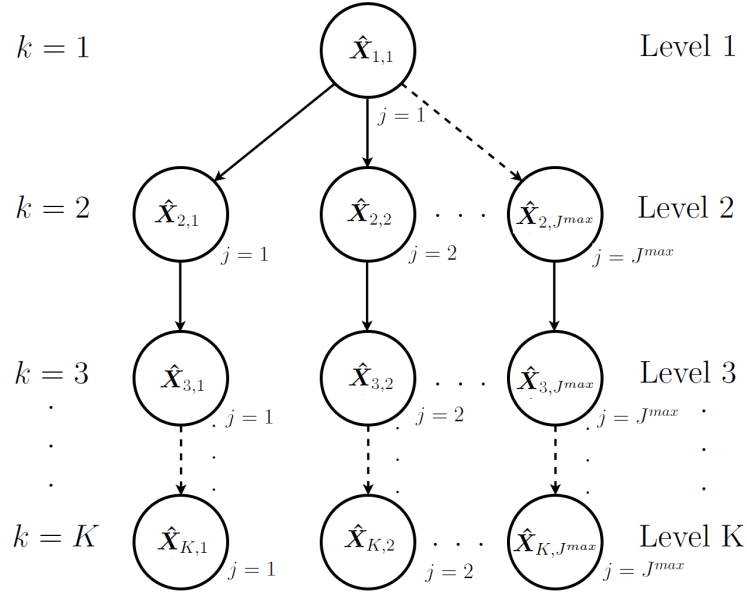


FIGURE 3.1: Prediction tree structure. $N_{1,1} = J^{max}$, otherwise $N_{k,j} = 1$. Courtesy of [1]

covariance matrix. The state of a vessel is given as:

$$\begin{bmatrix} N & V_N & E & V_E \end{bmatrix}, \quad (3.1)$$

where N , E , V_N and V_E are the positions and velocities of a vessel in the north and east directions in a stationary NED reference frame. The complete model can be written as:

$$x_{k+1} = F_T x_k + v_k \quad p(v_k) = \mathcal{N}(v_k; 0, Q_T) \quad (3.2)$$

where v is the process noise and \mathcal{N} is the normal distribution. The state transition matrix F_T and the noise covariance matrix Q_T are given as:

$$F_T = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q_T = \sigma_a^2 \begin{bmatrix} T^4/3 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/3 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix}, \quad (3.3)$$

where T is the time in seconds between each level in the prediction tree and σ_a is the noise covariance parameter.

3.3 Neighbor course distribution method

The NCDM is a data-driven algorithm that utilizes historical AIS data to make its predictions. To obtain the predicted position $\hat{\mathbf{p}}_{k+1,j}$ the set of close neighbors (CNS) at node (k, j) is considered. This set is defined as:

$$\mathbf{C}_{k,j} = \{\mathbf{X}_i | d(\hat{\mathbf{S}}_{k,j}, \mathbf{S}_i) \leq r_c, \mathbf{X}_i \in \mathbf{X}\}, \quad (3.4)$$

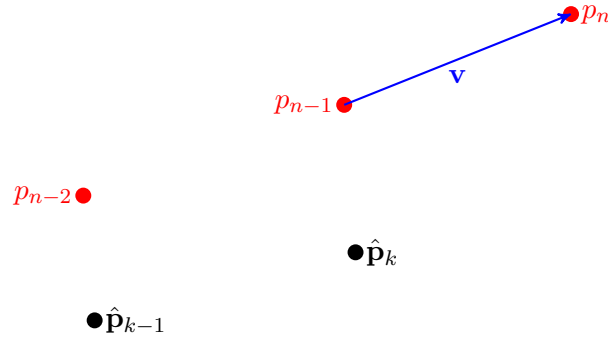


FIGURE 3.2: The sub-trajectory shown in red is a close neighbor of \mathbf{X}_1 if the Euclidean distance between $[\hat{\mathbf{p}}_{k-1} \ \hat{\mathbf{p}}_k]$ and $[\mathbf{p}_{n-2} \ \mathbf{p}_{n-1}]$ is less than r_c . The location of $\hat{\mathbf{p}}_{k+1}$ is determined by adding \mathbf{v} to $\hat{\mathbf{p}}_k$.

where \mathbf{X} is the \mathbf{X}^B data structure described in Chapter 2, r_c is a search radius and $d(\hat{\mathbf{S}}_{k,j}, \mathbf{S}_i)$ is the distance between the predicted sub-trajectory at node (k, j) and sub-trajectory \mathbf{S}_i in the dataset. The distance between two sub-trajectories of size n is found by representing each sub-trajectory as a point in $2 \times n$ dimensions and calculating the Euclidean distance between these two points. The initial state and predicted states consists of $n - 1$ points while the sub-trajectories in the dataset consist of n points. The predicted states are therefore only compared to the $n - 1$ first points of the sub-trajectories in the dataset. The remaining point is used to obtain the next predicted position.

A random sample is drawn from the CNs. From this sampled sub-trajectory the vector $\mathbf{v} = \mathbf{p}_n - \mathbf{p}_{n-1}$ is obtained. This vector is added to $\hat{\mathbf{p}}_{k,j}$ to get $\hat{\mathbf{p}}_{k+1,q}$. The state, which is a sub-trajectory, is updated by removing the first point from the sub-trajectory and adding the newly calculated one at its end.

Algorithm 1 Neighbor course distribution method

1: **Input parameters:**

- \mathbf{X}_1 ▷ Initial state
- $N_{k,j}$ ▷ Number of child nodes from node (k,j)
- K ▷ Total number of tree levels

2: Set $\hat{\mathbf{X}}_{1,1} = \mathbf{X}_1$

3: **for** $k = 1$ to $K - 1$ **do**

4: $q = 0$ ▷ Indexing variable at level k

5: **for** $j = 1$ to J_k **do**

6: Find close neighbors

7: **for** $N_{k,j}$ iterations **do**

8: $q = q + 1$

9: Obtain random sample

10: Calculate the next position

11: Update $\hat{\mathbf{X}}_{k+1,q}$ based on the latest prediction

12: **end for**

13: **end for**

14: **end for**

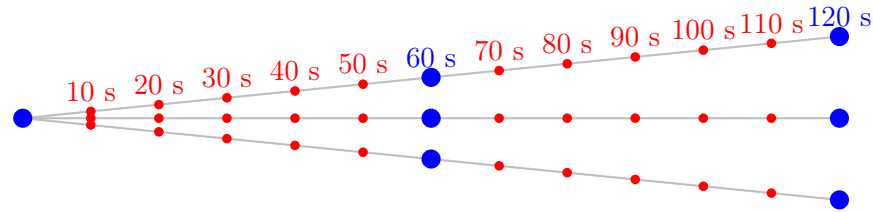


FIGURE 3.3: Interpolation of a simple prediction tree with two prediction steps of one minute each and $J^{max} = 3$. The new prediction tree has one level every 10 seconds. The old levels are shown in blue and the new levels in red.

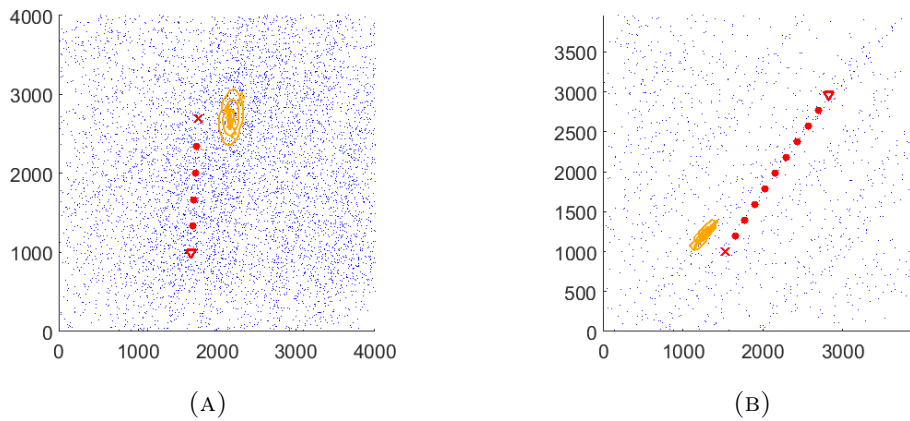


FIGURE 3.4: Predictions using the NCDM with low data density: The blue dots show the AIS data. The red dots show the real trajectory of the vessel with the triangle indicating its start and the cross indicating its end. A probability distribution of the vessel's end position as predicted by the NCDM is shown in orange.

3.3.1 Interpolation of trajectories

The choice of t , the time step between each point in a sub-trajectory, in \mathbf{X}^B is of major importance when applying the NCDM as changes in this parameter will influence both the granularity and accuracy of the predicted trajectories. In Chapter 6 it becomes evident that a smaller time step than the one minute used in [13] is necessary. However, a smaller time step seems to negatively affect predictions as sub-trajectories pointing in opposite directions might then be considered close neighbors. This is solved by linearly interpolating the vector \mathbf{v} , which defines the position of $\hat{\mathbf{p}}_{k+1,q}$ in relation to $\hat{\mathbf{p}}_{k,q}$, at desired intervals. This is illustrated in Figure 3.3.

3.4 Constant velocity model in the NCDM

A major drawback of the NCDM is that it often gives inaccurate and overconfident predictions in areas with low data density, as illustrated in Figure 3.4. In such areas relying on a data-driven approach does not make sense and it might be more reasonable to fall back on the CVM method.

This is solved by creating new sub-trajectories calculated using CVM and adding them to the set CNs. Given the state $[\mathbf{p}_1 \cdots \mathbf{p}_{n-1}]$, the new sub-trajectory will be given as: $[\mathbf{p}_1 \cdots \mathbf{p}_{n-1} \quad \mathbf{p}_n]$, where \mathbf{p}_n is calculated using the CVM as explained in Section 3.2.

The values N , E , V_N and V_E are obtained as follows:

$$[N \quad E] = \mathbf{p}_{n-1}, \quad (3.5a)$$

$$[V_N \quad V_E] = \frac{1}{t} [\mathbf{p}_{n-1} - \mathbf{p}_{n-2}], \quad (3.5b)$$

where t is the time step of the dataset. Note that using this method the CVM is only used to predict one time step into the future. An alternative is use the CVM for the entire prediction if there is no data nearby the initial state. However, this will fail to take advantage of new data that might appear nearby newly predicted states.

Several copies of this newly created sub-trajectory are added to the CNs. The number of copies are found according to the following formula:

$$W = \begin{cases} [\alpha \frac{1}{M}], & \text{if } M > 0 \\ 1, & \text{if } M = 0 \end{cases} \quad (3.6)$$

where W is the number of copies, $\alpha > 0$ is a weight parameter, M is the number of sub-trajectories in the CNs and $[\cdot]$ rounds to the nearest integer. Equation (3.6) is designed to provide balance between the use of NCDM and CVM depending on the data available. In areas with high data density, NCDM is favored, while CVM is favored in areas with low data density. Thus the sum $M + W$ is the total number of CNs, while the ratios $M/(M + W)$ and $W/(M + W)$ give the fraction of CNs that come from the NCDM and CVM, respectively.

Chapter 4

Gaussian mixture learning

Learning the parameters of a GMM can be done by applying the expectation maximization (EM) algorithm, which fits the maximum likelihood GMM to the given set of data points [24]. Since the EM algorithm needs the number components as an input, fitting the model must be done using a two stage implementation: First, several models are fitted to the data with an increasing component number k . Second, one model is selected based on a criteria such as Akaike's information criterion (AIC) [25] or Bayesian inference criterion (BIC) [26]. In [13] the number of components was selected by applying a minimum threshold for the distance between means of each component.

However, this approach may suffer from a large computational cost because a new model has to be learned for each k , with more parameters to be learned each time k increases. Another problem with the EM algorithm is the presence of singularities in the likelihood function [27].

One solution to this problem is using automatic model selection in which the number of components are found automatically during parameter learning. In this thesis the variational Bayes (VB) method introduced in [27] is tested for this task. This chapter first explains the essential theory of this method before exploring the method in depth.

4.1 Gaussian mixture models

A GMM consists of a sum of a finite number of Gaussian distributions [28]. In a mixture with K components each component will have a weight π_k , a mean $\boldsymbol{\mu}_k$ and a covariance matrix $\boldsymbol{\Sigma}_k$. The weights are defined so that they sum to 1, i.e $\sum_{i=1}^K \pi_k = 1$. A Gaussian mixture is defined by the following equation:

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (4.1)$$

where $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is a multivariate Gaussian.

Each data point \mathbf{x}_n also has an associated latent variable \mathbf{z}_n which is a binary vector of length K . The vector \mathbf{z}_n indicates which component the point \mathbf{x}_n originates from. If \mathbf{x}_n comes from component k then $z_k = 1$ while all other elements in \mathbf{z}_n are equal to zero.

4.2 Expectation maximization

The EM algorithm fits the maximum likelihood GMM to a set of points. The algorithm starts with an initial mixture and iterates until convergence is found. Each iteration can be divided into one expectation step and one maximization step. Before applying the EM algorithm the number of components in the mixture has to be known.

The expectation step consists of computing the *membership weights*. This measure, w_{ik} , gives the probability of variable \mathbf{x}_i being generated by component k . The weights are computed as

$$w_{ik} = \frac{\hat{\pi}_k \mathcal{N}(\mathbf{x}_i | \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{j=1}^K \hat{\pi}_j \mathcal{N}(\mathbf{x}_i | \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}, \quad (4.2)$$

where $\hat{\pi}_i$, $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\Sigma}}_i$ are estimates of the weight, mean, and covariance matrix respectively for component i . These membership weights are computed for all variables \mathbf{x}_i , $1 \leq i \leq N$ and all components $1 \leq k \leq K$. This gives an $N \times K$ matrix where each row will sum to 1.

In the maximization step, the estimates for all K components are updated using the membership weights. The following equations are used to calculate the new estimates for the mixture weights, means and covariance matrices:

$$\hat{\pi}_k = \frac{\sum_i w_{ik}}{N} \quad (4.3a)$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_i w_{ik} \mathbf{x}_i}{\sum_i w_{ik}} \quad (4.3b)$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_i w_{ik} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)}{\sum_i w_{ik}}. \quad (4.3c)$$

These two steps are repeated until convergence.

4.2.1 Model selection

The EM algorithm takes the number of components in a mixture as an input. Several methods exist for determining this number.

Akaike information criterion

AIC is a measure of how well a statistical model fits to a set of data. The smaller the AIC, the better the fit. It is defined as follows:

$$\text{AIC} = 2k - 2 \ln(\hat{L}), \quad (4.4)$$

where k is the number of parameters and \hat{L} is the maximum value of the likelihood function for the model. The second term of the equation rewards goodness of fit, while the first term penalizes models with a large number of parameters in order to avoid overfitting.

Bayesian information criterion

The BIC has the following definition:

$$\text{BIC} = \ln(n)k - 2 \ln(\hat{L}), \quad (4.5)$$

where k and \hat{L} are the same values as defined above, and n is the number of data points. As can be seen, the BIC is very similar to the AIC. The only difference is in the penalty term for the number of parameters. A comparison between AIC and BIC is made in [29].

Component distance

In [13], the component distance (CD) method was used which fits a new GMM with an increasing component number to the data until the distance between the means of two components is shorter than a set distance. This was done because the NCDM is mostly concerned about finding components with significantly different means as a way to handle branching of sea lanes.

4.3 Variational Bayes theoretical background

This section explains the theory behind variational Bayesian inference.

4.3.1 Bayesian inference

Bayesian inference is the process of updating a probability distribution as more data becomes available. This is done by applying Bayes' theorem:

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}. \quad (4.6)$$

The prior distribution $p(Y)$, the marginal likelihood $p(X)$ and the likelihood function $p(X|Y)$ are needed to get a posterior distribution $p(Y|X)$, which gives the distribution of Y after the data X is taken into account.

4.3.2 Variational calculus

A function is defined as an operator that takes a value as input and returns a value as output. Similarly, a functional is defined as an operator that takes a function as input and returns a value. Variational calculus uses variations, small changes in a functional's value due to a small change in its input function, to find maxima and minima of functionals. The concept of variations is analogous to differentiation in conventional calculus.

Functional derivatives are found by applying the Euler-Lagrange equation. This equation states that a functional on the form:

$$F[y] = \int_a^b L(x, y(x), y'(x)) dx, \quad (4.7)$$

has the functional derivative:

$$\frac{\partial F}{\partial y(x)} = \frac{\partial L}{\partial y} - \frac{d}{dx} \frac{\partial L}{\partial y'}. \quad (4.8)$$

Maxima and minima are found, as in conventional calculus, by setting the derivative equal to zero.

4.3.3 Kullback-Leibler divergence

Kullback-Leibler (KL) divergence is a measure of the difference between two probability distributions, p and q . In the context of VB, p is the true posterior distribution, while q is its approximation. KL divergence is defined by the following equation:

$$\text{KL}(q||p) = - \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} d\mathbf{Z}, \quad (4.9)$$

where \mathbf{Z} are the latent variables and \mathbf{X} are the data points.

Note that the KL divergence is non-symmetric, that is $\text{KL}(q||p) \neq \text{KL}(p||q)$. To distinguish the two, $\text{KL}(q||p)$ is called forward KL divergence, while $\text{KL}(p||q)$ is called reverse KL divergence.

From Equation (4.9) it can be seen that $\text{KL}(q||p) \rightarrow \infty$ if $q(\mathbf{Z}) \rightarrow 0$ while $p(\mathbf{Z}|\mathbf{X})$ is large. This means the divergence between two distributions can be widely different whether the forward or the reverse KL divergence is used. This is illustrated in Figure 4.1. Figure 4.1a shows a scenario where $\text{KL}(q||p)$ is small while $\text{KL}(p||q)$ is large. This is mostly due to the tails of $q(\mathbf{Z})$ where $p(\mathbf{Z}) \approx 0$. In Figure 4.1b the opposite is true. Here $\text{KL}(q||p)$ is very large, due to the peak of the rightmost component of $p(\mathbf{Z})$ being at a point where $q(\mathbf{Z}) \approx 0$, while $\text{KL}(p||q)$ is small.



(A) Scenario where $\text{KL}(q||p)$ is small and $\text{KL}(p||q)$ is large. (B) Scenario where $\text{KL}(q||p)$ is large and $\text{KL}(p||q)$ is small.

FIGURE 4.1: Two probability distributions $p(\mathbf{Z})$ and $q(\mathbf{Z})$. The forward and reverse KL divergences give very different similarity between the two.

4.4 Variational Bayesian inference

Given a Bayesian model where all parameters have prior distributions, all parameters and latent variables are denoted by \mathbf{Z} , while the observed data is denoted by \mathbf{X} . The goal is to find the posterior distribution $p(\mathbf{Z}|\mathbf{X})$. The marginal likelihood $p(\mathbf{X})$ is often difficult to obtain and Bayes' theorem can therefore not be used directly in such situations. Instead the variational Bayesian inference solves this problem by finding

the parameters of a distribution $q(\mathbf{Z})$ which approximates $p(\mathbf{Z}|\mathbf{X})$. The similarity between the two distributions is measured using the KL divergence. A good approximation therefore means minimizing the KL divergence.

The log marginal probability can be decomposed as follows [28]:

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}(q||p), \quad (4.10)$$

where $\mathcal{L}(q)$ is called the *lower bound* and is defined as:

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z}. \quad (4.11)$$

Since the log marginal probability is constant, maximizing the lower bound is equivalent to minimizing the KL divergence. Notice that $\mathcal{L}(q)$ is a functional and it is therefore necessary to use variational calculus to maximize it.

The elements of \mathbf{Z} are partitioned into disjoint groups such that the approximate distribution gets the form:

$$q(\mathbf{Z}) = \prod_{i=1}^M q_i(\mathbf{Z}_i). \quad (4.12)$$

The goal now is to find the distribution $q(\mathbf{Z})$ which maximizes the lower bound $\mathcal{L}(q)$. By plugging (4.12) into (4.11) the following expression can be obtained [28]:

$$\ln q_j^*(\mathbf{Z}) = \mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Z})] + \text{const}, \quad (4.13)$$

where $q_j^*(\mathbf{Z})$ is the distribution which maximizes the lower bound and $\mathbb{E}_{i \neq j}[\dots]$ gives the expectation across all variables except j .

4.5 Variational Bayes method for GMMs

Variational Bayesian inference can be used to infer the parameters of a GMM. This application of VB is analogous to the EM algorithm. The difference is that instead of using maximum likelihood estimates to find the most probable value of each parameter, the VB method computes an approximation of the entire posterior distribution of the parameters and latent variables. The VB method has a similar two-step, alternating structure as the EM algorithm. These two steps are:

1. A variational E-step that computes the *responsibilities* (analogous to the membership weights of EM) based on the current parameter estimates of the mixture.
2. A variational M-step that computes new estimates for the parameters based on the responsibilities.

Unlike the EM algorithm, the VB method does not need a parameter specifying the number of components in the mixture, but infers this during the parameter learning from a given value for the maximum number of components allowed in the mixture.

Chapter 5

Prediction results

This chapter presents the results of tests performed on the CVM method and the NCDM, with and without modifications. Tests are conducted to determine the optimal parameters for the methods. Lastly, the methods are compared to see which one is the most suitable for vessel position prediction. All tests measuring computational time were run on an Intel(R) Xeon(R) CPU 3.40 GHz processor with 8 GB RAM and Windows 10 64-bit operating system.

5.1 Performance measures

Two performance measures are used to determine the quality of a set of predictions: root mean square error (RMSE) and a generalized concept of filter consistency in the sense of [30]. These are the same performance measures used in [13], but their explanation is repeated here.

The terms accuracy and precision will be used extensively in this analysis and should be clarified: Accuracy is a value indicating how far a predicted mean is from the true value. Precision is a measure of the predicted variance. High precision means that a prediction has low variance and vice versa.

5.1.1 Root mean square error

The RMSE is used to measure the accuracy of a prediction. The smaller the RMSE, the more accurate is the prediction. It is here defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{J^{max}} \|\hat{\mathbf{p}} - \mathbf{p}_i\|^2}{J^{max}}}, \quad (5.1)$$

where $\hat{\mathbf{p}}$ is the true position, \mathbf{p}_i is the predicted position at iteration i and J^{max} is the number of individual predictions as defined in Section 3.1. This measure will give an idea of how close the mean of the predictions is to the true value. The RMSE has some problems when it comes to multimodal distributions (GMMs with more than one component) which might be produced by the prediction methods. This is because the mean of a multimodal distribution might be in an empty area between components. However, this is a rare scenario so RMSE can still serve as a good quality measure. An alternative to RMSE is the median error, but this is not used here.

5.1.2 Consistency

The second performance measure is used to determine if the GMM produced is consistent. Filter consistency is colloquially explained as follows in [30]: The estimation

error should have magnitude commensurate with the corresponding covariance that is yielded by the estimator. In order to deal with multimodality, consistency of the prediction methods is measured using the same method as in [31]. First, the probability density function (PDF) value as given by the GMM for the true position is noted for each individual test. This value, called f , is then compared to the maximum value of the same PDF, called f_{max} . The ratio $f/f_{max} \in [0, 1]$ thus serves as a quality measure where a single prediction is better the closer the ratio is to one. The distribution of f/f_{max} for N predictions will then give an idea of the consistency. The values f and f_{max} are illustrated in Figure 5.1.

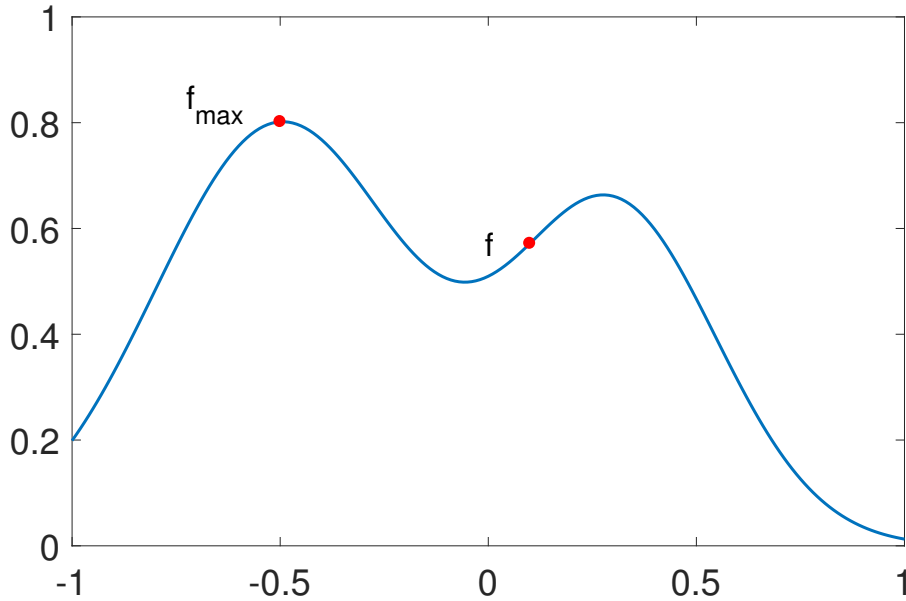


FIGURE 5.1: Performance measure illustrated using one-dimensional data. The true value $x = 0.1$. In this case $f/f_{max} \approx 0.57/0.8 \approx 0.71$.

5.2 Test setup

The dataset is initially divided into a training set and a test set. The first 90% of the data points are used as training data, while the remaining 10% are used for tests. The NCDM is always run with the prediction tree described in Section 3.1 where $J^{max} = 200$, i.e the tree has 200 branches which all originate in its root node. All tests were done on $N = 600$ initial states randomly pulled from the test set. A test of a given method for a given time on N initial states will from here be referred to as a 'test' while a test of a single initial state will be referred to as an 'individual test'. The parameters for the NCDM are given in Table 5.1.

TABLE 5.1: Decision parameters for NCDM

| Decision parameter | Value | Description |
|--------------------|-------|---|
| n | 3 | Number of points in each sub-trajectory |
| t | 60s | Time between each point |
| r_c | 100m | Search radius for CNs |

The initial states and their corresponding trajectories must fulfill one requirement: The time at the end of the trajectory minus the time of the initial state must be larger than or equal to the test time. This ensures that there is a true position in the test trajectory that the prediction can be compared to. An individual test is also discarded if the GMM learning algorithm fails to fit a GMM to the points. This may happen when the data is highly correlated.

5.3 Testing of CVM

In this section the CVM is tested with various values for the process noise covariance parameter σ_a and evaluated according to their resulting consistency as defined in Section 5.1. In [23] the authors conclude that $\sigma_a = 0.05$ gives the best results for large vessels with little maneuverability, while $\sigma_a = 0.5$ gives better results for vessels with more maneuverability. With these results as a starting point, values between 0.05 and 0.5 for the process noise covariance are tested.

The natural logarithms of the f/f_{max} ratios are displayed in Figure 5.2 in order to investigate the consistency. The far-right box plot consists of values sampled from a Gaussian distribution. This plot should therefore exhibit ideal consistency properties and serves as a comparison for the box plots to the left. A consistent prediction method would have a box plot similar to the one from the Gaussian distribution.

From Figure 5.2 it appears that the predictions with noise covariance parameter $\sigma_a = 0.1$ has the box plot most similar to that of the Gaussian and therefore is the prediction with the best consistency properties. Using $\sigma_a = 0.05$ gives by far the worst consistency properties. This is because a low amount of noise leads to very precise predictions that will miss if the vessel makes a slight turn. The two remaining predictions give variances that are too high.

Figure 5.3 further explains the results. Although $\sigma_a = 0.05$ makes an accurate prediction in this scenario, the prediction is overconfident as the ratio f/f_{max} will be very low if the vessel deviates from its initial velocity. It also becomes apparent that increasing σ_a quickly leads to very uncertain predictions.

5.4 Testing the modified NCDM

In this section the modified NCDM is tested. Tests are run for 8 minutes into the future as before. Each test is run with a different value of α to determine the one which leads to predictions with the best consistency properties.

Several different values for α have been tested and it appears that a α of around 5000 gives the best consistency properties. In Figure 5.4 the f/f_{max} ratios of some of the tested values are displayed. It becomes apparent from the box plot for $\alpha = 0$ that modifications to the NCDM result in a significant improvement in the consistency properties. Finer tuning of the α parameter is challenging as the box plots are difficult to compare and because of the stochastic nature of the NCDM.

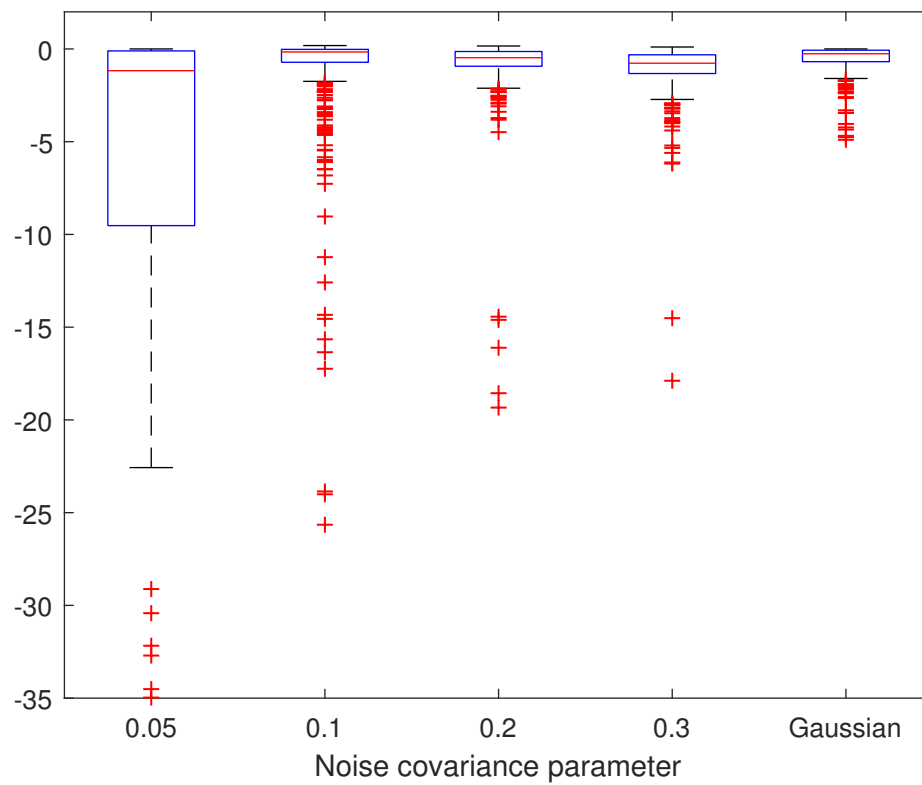


FIGURE 5.2: Box plot of the PDF-ratios for different noise parameters: The central line represents the median. The bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. All data points not considered outliers are between the whiskers, while outliers are plotted individually as crosses. Note that outliers below -35 on the y-axis are not plotted.

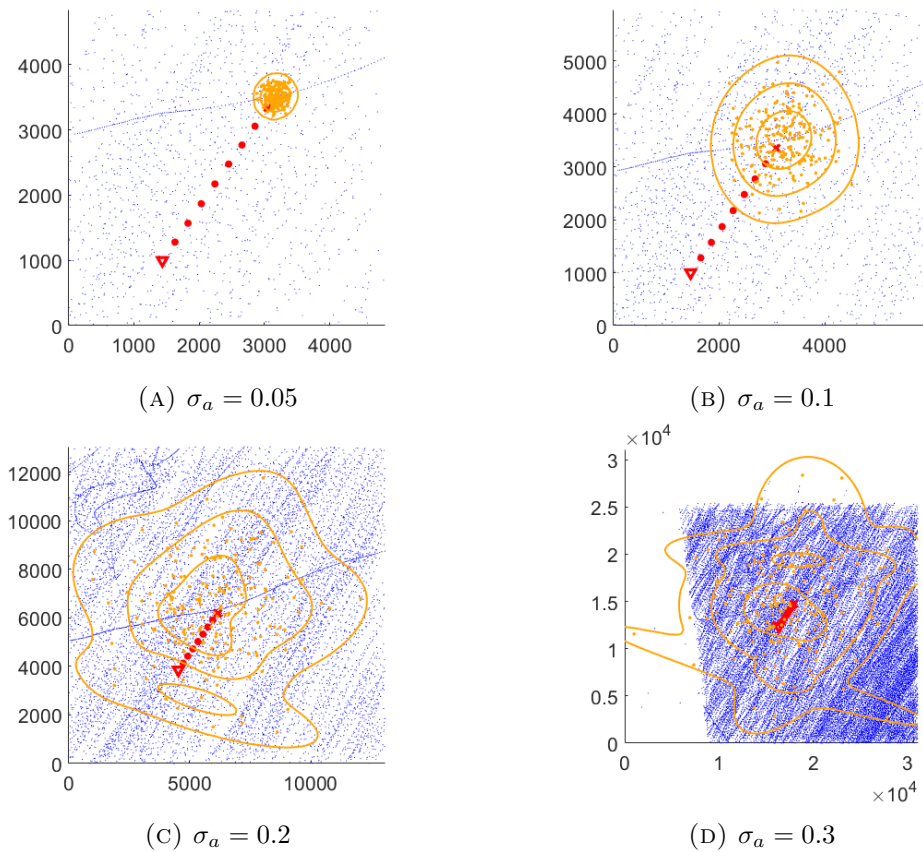


FIGURE 5.3: Prediction plots of the CVM with different noise covariance parameters: The blue dots are reported ship positions from the AIS dataset. The red dots indicate the true trajectory of a ship with the triangle representing the initial position and the cross the true position at the time the position is predicted. The orange dots are predicted positions and the orange lines correspond to the 1, 2 and 3 standard deviation equi-probability contours of the GMM fitted to them. The axes are in meters.

5.5 Testing of Gaussian learning methods

In this section the VB method for automatic model selection is tested. The results taken into account are: consistency, computational time and average number of components. This is compared with same results from EM algorithm, using AIC, BIC and the CD method for model selection.

TABLE 5.2: Parameters for learning methods

| Decision parameter | Method | Value |
|--------------------|--------|-------|
| d_{min} | CD | 500m |
| c_{max} | VB | 8 |
| π_{min} | VB | 0.01 |

Some of these methods require decision parameters that are given in Table 5.2. The distance d_{min} gives the minimum distance allowed between the means of any two components when using the CD method. The VB takes in a maximum number of

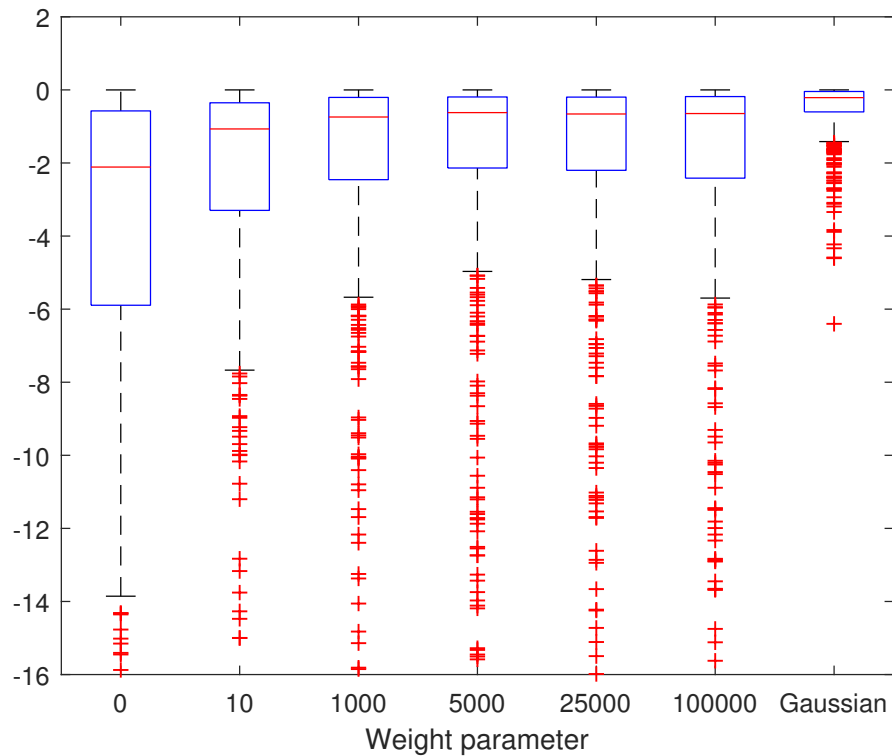


FIGURE 5.4: Box plot of the PDF-ratios for different weight parameters.

components which is given by c_{max} . This method often produces GMMs with many small components. However, these are unlikely to represent any real trend and increased complexity is often undesirable. Therefore the components with a weight of less than π_{min} are set to zero with their weight distributed among the remaining components.

From Figure 5.5 it can be seen that the new learning methods introduced to the NCDM in this thesis do not lead to an improvement in consistency properties. In particular, the VB method gives a poor performance. Table 5.3 also show that VB gives no improvement in computational time and in fact performs a lot worse than the EM algorithm. The information criteria for the EM, AIC and BIC, also perform worse than the CD method, both in consistency and computational time. Therefore, the means method continues to be used in the further testing.

TABLE 5.3: Results of learning method tests

| Method | Mean computational time | Mean number of components |
|--------------|-------------------------|---------------------------|
| Means method | 0.048 | 1.20 |
| AIC | 0.16 | 2.57 |
| BIC | 0.10 | 2.09 |
| VB | 0.68 | 2.88 |

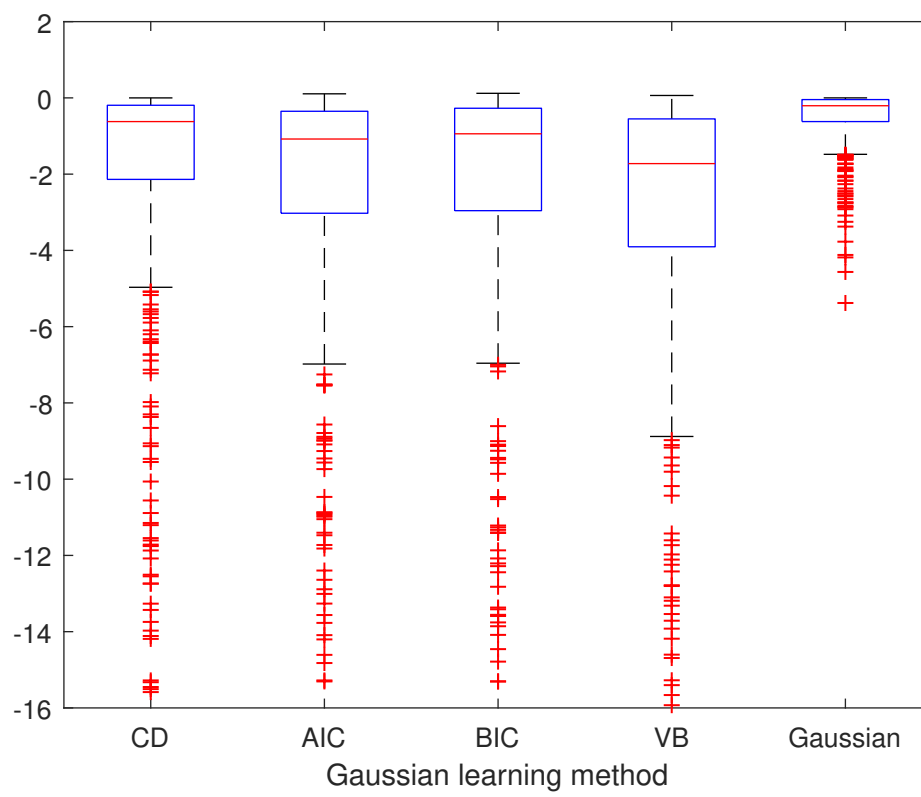


FIGURE 5.5: Box plot of the PDF-ratios of the different learning methods.

5.6 Comparison of methods

This section compares the three different prediction methods discussed in this thesis: the CVM, the NCDM and the modified NCDM. All methods are tested for 5, 10 and 15 minutes into the future. They are evaluated by the performance measures introduced in Section 5.1.

5.6.1 RMSE

The median values of the RMSE of all the methods for each test time is given in Table 5.4. The same information is plotted in Figure 5.6. As can be seen, the old NCDM is the most accurate method, followed by the modified NCDM and the CVM is the least accurate. The fact that the old NCDM is the most accurate is expected, as the modifications to the NCDM were not made to improve its accuracy, but rather its consistency. In any case, the results show that the NCDM, with or without modifications, is significantly more accurate than the CVM.

TABLE 5.4: Median RMSE for each test time

| | CVM | Old NCDM | Modified NCDM |
|--------|-------|----------|---------------|
| 5 min | 344m | 242m | 271m |
| 10 min | 978m | 582m | 718m |
| 15 min | 1805m | 955m | 1268m |

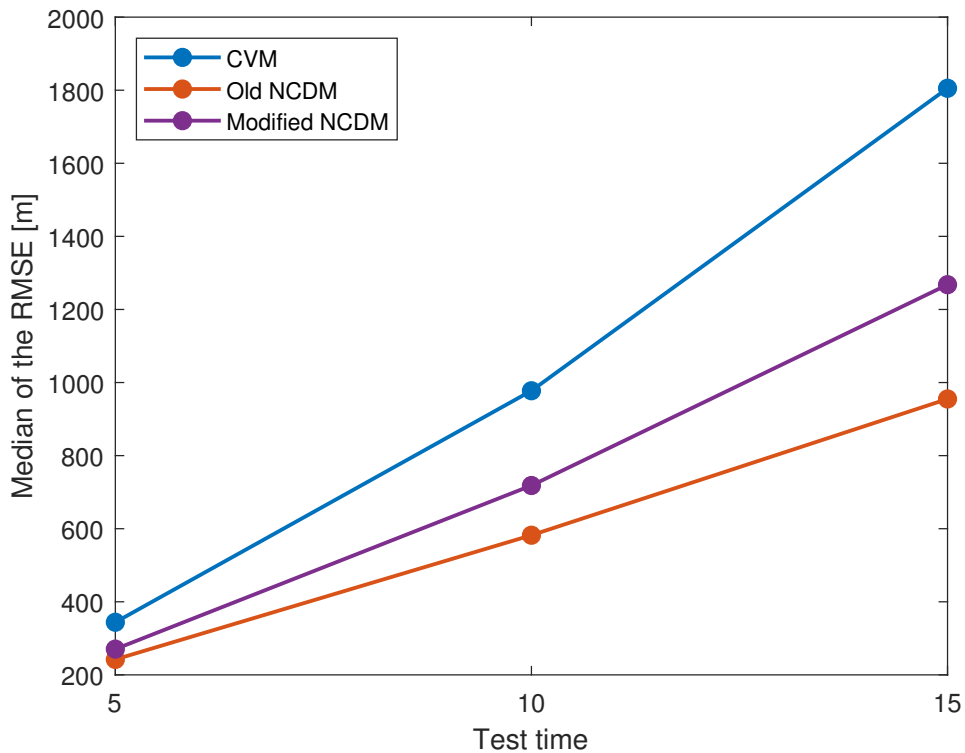


FIGURE 5.6: Median RMSE over time

5.6.2 Consistency

Figure 5.7 a consistency box plot for each method at each of the test times. The old NCDM stands out as the method with the worst consistency properties, while both the CVM and the modified NCDM offer a significant improvement. In addition, it can be seen that the CVM slightly outperforms the modified NCDM.

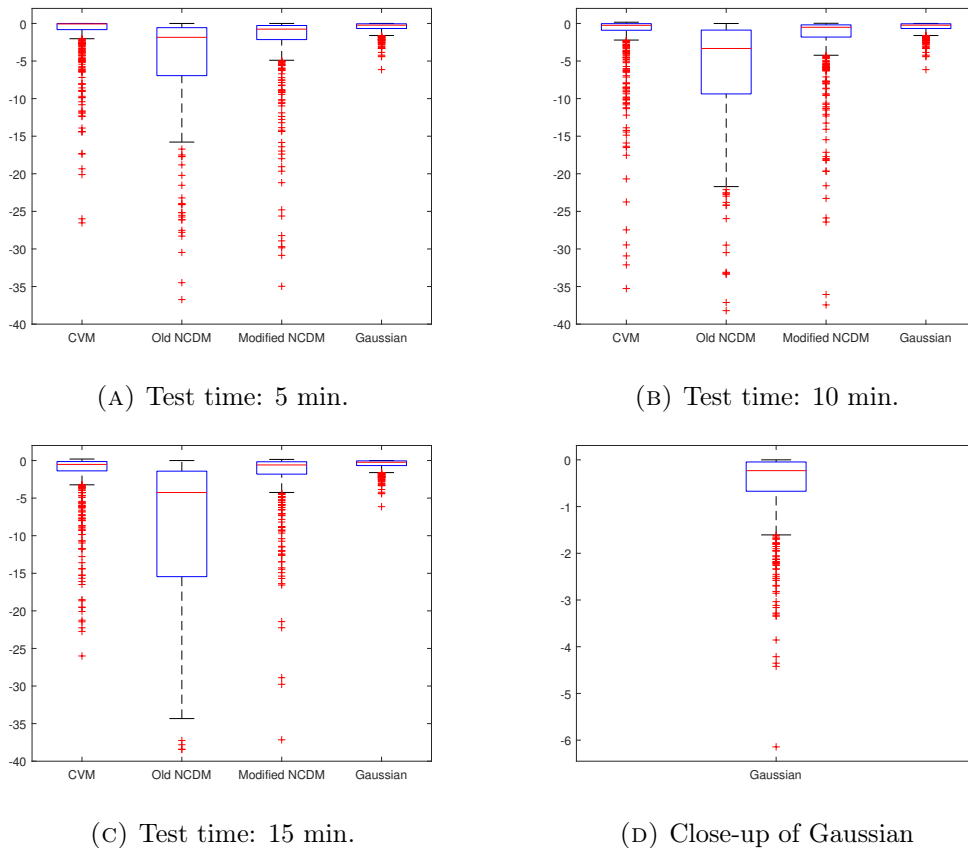


FIGURE 5.7: Box plot of the PDF-ratios for the method comparison tests.

5.7 Discussion

The learning of a GMM using VB was investigated in this thesis since the number of components in the mixtures that represent the predictions is unknown. However, different vessels usually do not deviate much from each other, neither in course nor speed, over the prediction horizons used. Thus, the best fit usually has very few components and a two stage implementation of the EM is therefore not very computationally expensive.

The modifications to the NCDM were introduced in order to improve the algorithm's performance in areas with low data density. The NCDM often makes overconfident and inaccurate predictions in such areas, which resulted in poor consistency properties. The results show that the modifications significantly improve the consistency properties of the predictions, although the CVM still have better consistency properties than the NCDM. On the other hand, the CVM is less accurate than the NCDM. Thus, the modified NCDM can be seen as a compromise between the CVM and the

old NCDM, where there is a trade-off between consistency and accuracy. The choice of α in the modified NCDM decides of much importance is placed on accuracy and consistency, where a small α will weight the algorithm towards the former and a large α will weight it towards the latter.

Part II

Collision Avoidance

Chapter 6

NCDM in a COLAV system

This chapter investigates whether the NCDM can be employed in a COLAV system. The COLAV algorithm used is presented in [2] and is based on MPC.

6.1 Model predictive control

MPC is a control technique that is commonly used for multi-variable control problems that have constraints on both input and output variables, as well as the states of the system [32]. MPC can be used when there exists a reasonably accurate model of the process that is to be controlled.

Figure 6.1 illustrates the principle behind a discrete MPC. The controller considers an optimization problem. At time step k it predicts the optimal output values for a prediction horizon P , by calculating the inputs over a control horizon $M < P$ that achieve these outputs.

Although MPC calculates M future inputs, only the first one is implemented, and the process is repeated at the next time step. This is done due to possible disturbances or changes in the optimization problem. The remaining future inputs are used for further calculations before a new sequence of inputs are produced. MPC therefore has a receding prediction horizon where it always calculates M steps ahead.

6.2 MPC-based COLAV

In [2] a MPC-based COLAV algorithm for ASVs is developed. The algorithm minimizes an objective function which penalizes deviation between the vessel trajectory and the desired trajectory, as well as changes in speed and yaw rate. Inequality constraints are imposed on the position of the vessel to disallow it being in the same area as the obstacles. In addition, constraints are imposed on the yaw rate and speed of the vessel to ensure that it behaves in a realistic manner. The output of the MPC is an optimal trajectory for the ASV. The first step of this trajectory is implemented, before a new optimal trajectory is calculated at the next time step.

A modified version of this algorithm is used in this thesis for COLAV. Whereas the obstacles in [2] are modeled using inequality constraints predicted by the CVM, the obstacles are now modeled in the objective function as GMMs produced by the NCDM.

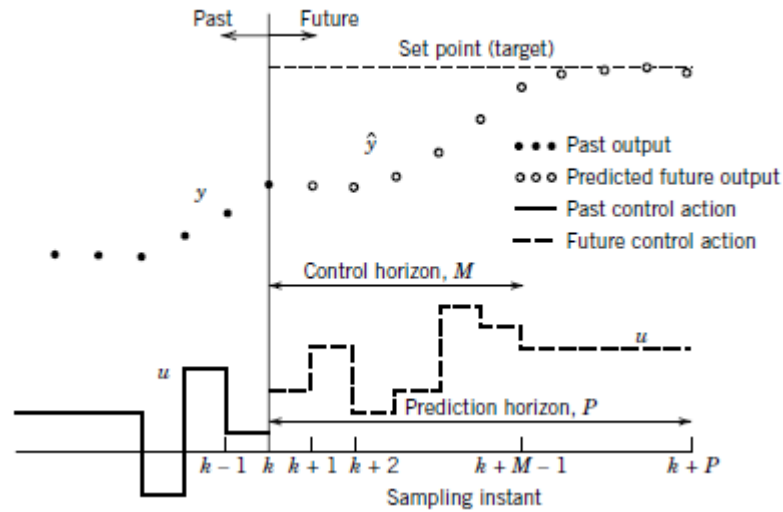


FIGURE 6.1: Illustration of an MPC for a single-input single-output system. Courtesy of [32].

6.2.1 ASV modeling

The algorithm uses a purely kinematic model for ASVs. It can be formulated as follows:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}, \quad (6.1)$$

where $\boldsymbol{\eta} = [N \ E \ \psi]^T$ is the vehicle pose with N and E representing the north and east coordinates respectively of the vessel position and ψ representing its yaw angle (heading with respect to north). The vector $\mathbf{u} = [U \ r]^T$, where U is the SOG and $r = \dot{\psi}$ is the yaw rate (ROT). Note that vehicle kinetics and ocean currents are neglected in this model. This can be done due to the long prediction horizon compared to the dynamics of the vessel, and also because a local, low-level COLAV algorithm is intended to also be used in the full COLAV system.

In addition, limitations are placed on the SOG and ROT to ensure that the vessel behaves in a feasible manner:

$$U_{min}(r) \leq U \leq U_{max}(r), \quad (6.2a)$$

$$r_{min} \leq r \leq r_{max}. \quad (6.2b)$$

Note that the limitations for U depend on r , while the limitations for r are constant.

6.2.2 Control objective

The algorithm assumes a desired trajectory $\mathbf{p}_d(t) = [N_d(t) \ E_d(t)]$ for the vessel. The objective is to make the vessel trajectory $\mathbf{p}(t)$ follow this desired trajectory as closely as possible, while at the same time avoiding areas where obstacles are expected.

A final objective is to make the vessel maneuver in a manner that complies with the COLREGS. These regulations give a set of 38 rules that are intended to prevent collisions between vessels. These rules have a varying degree of relevance for COLAV. Due to the significant challenges of implementing these rules in a COLAV system, they are often ignored in the literature. The algorithm from [2] is designed to make the vessel comply with 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.

6.2.3 Optimization problem

An important step in the MPC is to solve the optimization problem. First, the general optimization problem is defined as:

$$\begin{aligned}
& \text{minimize} && \phi(\boldsymbol{\eta}(t), \mathbf{u}(t)) \\
& \text{subject to} && \dot{\boldsymbol{\eta}}(t) = \mathbf{F}(\boldsymbol{\eta}(t), \mathbf{u}(t)) \\
& && \mathbf{h}(\boldsymbol{\eta}(t), \mathbf{u}(t)) \leq \mathbf{0} \\
& && \boldsymbol{\eta}(t_0) = \bar{\boldsymbol{\eta}}_0,
\end{aligned} \tag{6.3}$$

where ϕ is the objective function. $\mathbf{F}(\boldsymbol{\eta}(t), \mathbf{u}(t))$ is the kinematic model defined in (6.1), $\mathbf{h}(\boldsymbol{\eta}(t), \mathbf{u}(t))$ gives the inequality constraints, while $\bar{\boldsymbol{\eta}}_0$ is the initial state of the vessel.

Since continuous optimization problems are difficult to solve it is more practical to define a nonlinear program (NLP) by discretizing (6.3). The discretization is done using the direct multiple shooting technique. This technique involves discretizing both the future input and the future state, as opposed to only discretizing the future input as is done with single shooting. The discrete solutions are pieced together with the shooting constraints. The new problem, with N_p prediction steps, is now defined as:

$$\begin{aligned}
& \text{minimize} && \phi(\mathbf{w}) \\
& \text{subject to} && \mathbf{g}(\mathbf{w}) = \mathbf{0} \\
& && \mathbf{h}(\mathbf{w}) \leq \mathbf{0},
\end{aligned} \tag{6.4}$$

where $\mathbf{w} = \left[\boldsymbol{\eta}_0^T \quad \mathbf{u}_0^T \quad \cdots \quad \boldsymbol{\eta}_{N_p-1}^T \quad \mathbf{u}_{N_p-1}^T \quad \boldsymbol{\eta}_{N_p}^T \right]$ are the decision variables.

The objective function is defined as:

$$\phi(\mathbf{w}, \mathbf{p}_{d_1:N_p}) = \sum_{k=1}^{N_p} \left(\frac{K_p}{2} C(\|\mathbf{p}_k - \mathbf{p}_{d_k}\|_2; \delta) + K_{\dot{U}} q_{\dot{U}}(\dot{U}_{k-1}) + K_r q_r(r_{k-1}) \right), \tag{6.5}$$

where p_k and p_{d_k} are the actual and desired position of the vessel at time step k , respectively. $K_p, K_{\dot{U}}, K_r > 0$ are tuning parameters. The function C is the pseudo-Huber cost function. This function is usually linear, but approximates a quadratic function close to zero, and is used to keep large errors from dominating the cost

function. It is defined as:

$$C(x; \delta) = 2\delta^2 \left(\sqrt{\left(1 + \frac{x}{\delta}\right)} - 1 \right), \quad (6.6)$$

where 2δ is the slope of the linear part of the function.

The last two terms of the objective function ensure that the vessel maneuvers in compliance with rule 8 of COLREGS, that is the vessel should make large, visible movements when avoiding other vessels. Therefore, these functions heavily penalize small changes in speed and yaw, but penalize large changes more lightly. The functions are defined as:

$$q_r(r; r_{max}) = \frac{100}{q(r_{max}; a_r, b_r)} q(r; a_r, b_r) \quad (6.7a)$$

$$q_{\dot{U}}(\dot{U}; \dot{U}_{max}) = \frac{100}{q(\dot{U}_{max}; a_{\dot{U}}, b_{\dot{U}})} q(\dot{U}; a_{\dot{U}}, b_{\dot{U}}), \quad (6.7b)$$

where r_{max} and \dot{U}_{max} are the maximum expected values for r and \dot{U} , respectively. The function q defines the shape of the penalty terms and is given as:

$$q(\zeta; a, b) = a\zeta^2 + (1 - e^{-\frac{\zeta^2}{b}}), \quad (6.8)$$

where a and b are tuning parameters.

Due to the use of the multiple shooting technique it is necessary to include shooting constraints to ensure that the control input and vessel states satisfy (6.1). An integrator function is defined using Runge-Kutta of order 4:

$$\mathbf{k}_1 = \mathbf{F}(\boldsymbol{\eta}_k, \mathbf{u}_k) \quad (6.9a)$$

$$\mathbf{k}_2 = \mathbf{F}\left(\boldsymbol{\eta}_k + \frac{h}{2}\mathbf{k}_1, \mathbf{u}_k\right) \quad (6.9b)$$

$$\mathbf{k}_3 = \mathbf{F}\left(\boldsymbol{\eta}_k + \frac{h}{2}\mathbf{k}_2, \mathbf{u}_k\right) \quad (6.9c)$$

$$\mathbf{k}_4 = \mathbf{F}(\boldsymbol{\eta}_k + h\mathbf{k}_3, \mathbf{u}_k) \quad (6.9d)$$

$$\mathbf{f}(\boldsymbol{\eta}_k, \mathbf{u}_k) = \boldsymbol{\eta}_k + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \quad (6.9e)$$

where h is the discretization time step. Given a vessel state $\boldsymbol{\eta}_k$ and a control input \mathbf{u}_k it is now possible to obtain the vessel state for next time step since $\boldsymbol{\eta}_{k+1} = \mathbf{f}(\boldsymbol{\eta}_k, \mathbf{u}_k)$. This results in the following equality shooting constraints for the optimization problem:

$$\mathbf{g}(\mathbf{w}) = \begin{bmatrix} \bar{\boldsymbol{\eta}}_0 - \boldsymbol{\eta}_0 \\ \mathbf{f}(\boldsymbol{\eta}_0, \mathbf{u}_0) - \boldsymbol{\eta}_1 \\ \mathbf{f}(\boldsymbol{\eta}_1, \mathbf{u}_1) - \boldsymbol{\eta}_2 \\ \vdots \\ \mathbf{f}(\boldsymbol{\eta}_{N_p-1}, \mathbf{u}_{N_p-1}) - \boldsymbol{\eta}_{N_p} \end{bmatrix}. \quad (6.10)$$

Each control input is subject to limitations defined in (6.2) which is defined in the following function:

$$\mathbf{h}_{u_i}(\mathbf{u}_i) = \begin{bmatrix} U_{min}(r_i) - U_i \\ -(U_{max}(r_i) - U_i) \\ r_{min} - r_i \\ -(r_{max} - r_i) \end{bmatrix}, \quad (6.11)$$

which again forms the inequality constraints:

$$\mathbf{h}_u(\mathbf{w}) = \begin{bmatrix} \mathbf{h}_{u_0}(\mathbf{u}_0) \\ \mathbf{h}_{u_1}(\mathbf{u}_1) \\ \vdots \\ \mathbf{h}_{u_{N_p-1}}(\mathbf{u}_{N_p-1}). \end{bmatrix} \quad (6.12)$$

In [2] the obstacles are also included in the inequality constraints.

6.2.4 Using GMMs as obstacles

Since the NCDM represents obstacles as GMMs it is better to not model obstacles as constraints. This would not have taken advantage of the representation of probability that the GMM provides. Instead the GMM for a prediction at a given time step is included as a term in the objective function and the vessel will thus be penalized when moving over regions with a high likelihood of containing an obstacle at that time. The new objective function is given as:

$$\begin{aligned} \phi(\mathbf{w}, \mathbf{p}_{d_1:N_p}) &= \sum_{k=1}^{N_p} \left(\frac{K_p}{2} C(\|\mathbf{p}_k - \mathbf{p}_{d_k}\|_2; \delta) \right. \\ &\quad \left. + K_{\dot{U}} q_{\dot{U}}(\dot{U}_{k-1}) + K_r q_r(r_{k-1}) + K_g g(\mathbf{p}_k | \boldsymbol{\theta}_k) \right) \end{aligned} \quad (6.13)$$

where $K_g > 0$ is a tuning parameter and

$$g(\mathbf{p}_k | \boldsymbol{\theta}_k) = \sum_{m=1}^M \pi_{km} \mathcal{N}(\mathbf{p}_k | \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}) \quad (6.14)$$

is the value of the predicted GMM at time step k evaluated at position \mathbf{p}_k . The number M is the number of components in the mixture and $\boldsymbol{\theta}_k$ contains the mixture parameters π_k , $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. Figure 6.2 shows a simulation where this new objective function is used when navigating around a static GMM.

6.3 Simulation

The simulated ASV starts with an initial state and a desired trajectory, consisting of a set of waypoints, as well as a desired speed. The simulation is run for a given number N_p of prediction steps with a given step size. At each of these steps the NCDM is run. It is important that the resulting prediction tree from the NCDM has the same time step between levels as the step size of the simulation. This means that each point \mathbf{p}_k in the predicted trajectory from the MPC avoids a corresponding GMM representing the obstacles at time k . Thus, at each step the optimization problem is solved with a new objective function. The optimization problem is solved using the CASADI [33] framework for MATLAB with the IPOPT solver.

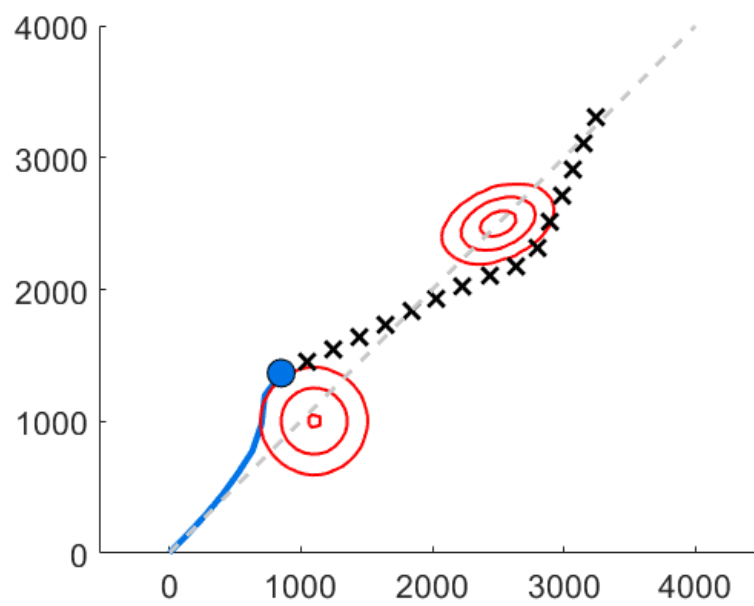


FIGURE 6.2: GMM avoidance proof of concept: The figure shows a simulated ASV in blue successfully avoiding a static GMM with two components. The red contours correspond to the 1, 2 and 3 standard deviation equi-probability contours for the mixture. The crosses indicate the trajectory predicted by the MPC.

Chapter 7

COLAV results

In this chapter a series of qualitative tests of the COLAV algorithm are performed. Two "close-to-collision" scenarios are picked from the dataset. The first scenario is chosen to see whether the new COLAV method with NCDM for obstacle prediction is advantageous compared to the original COLAV method. The second scenario is chosen to investigate whether the modified NCDM is able to handle low data density areas. In each scenario one vessel from the dataset is chosen as the obstacle and the desired trajectory for the ASV is defined near or across the trajectory of the obstacle. First, the original COLAV method from [2], where the CVM is used and the obstacles are represented as constraints, is tested. Then the new COLAV method described in Section 6.2.4 is tested, where the NCDM as described in Section 3.3 is used for vessel prediction. Lastly, the same method is tested, but now using the the modified NCDM described in Section 3.4 for vessel prediction. The parameters used in the tests are shown in Table 7.1.

TABLE 7.1: Decision parameters for the COLAV tests.

| Parameter | Value | Description |
|------------------------------|------------------------------|--|
| h | 20 s | Time step interval |
| N_p | 24 | Prediction steps |
| U_{min} | 0 m/s | Minimum SOG |
| U_{max} | 17 m/s | Maximum SOG |
| K_p | 0.04 | Position error scaling |
| $K_{\dot{u}}$ | 0.6 | SOG-derivative penalty term scaling |
| K_r | 0.5 | ROT penalty term scaling |
| $[a_{\dot{u}}, b_{\dot{u}}]$ | $[8, 2.5 \times 10^{-4}]$ | SOG-derivative penalty term parameters |
| $[a_r, b_r]$ | $[112, 6.25 \times 10^{-4}]$ | ROT penalty term parameters |
| r | 200 m | Obstacle radius (original COLAV method) |
| K_g | 10^{14} | GMM penalty term scaling (new COLAV method) |
| n | 3 | Number of points in each sub-trajectory (NCDM) |
| t | 60s | Time between each point (NCDM) |
| r_c | 100m | Search radius for CNs (NCDM) |
| α | 5000 | Weight parameter for (modified NCDM) |
| σ_a | 0.1 | Noise covariance parameter (modified NCDM) |

7.1 Scenario 1: Straight line crossing curved traffic area

The following scenario takes place around a turning sea lane with a relatively high amount of traffic. The desired trajectory goes in a straight line that tangents the turn,

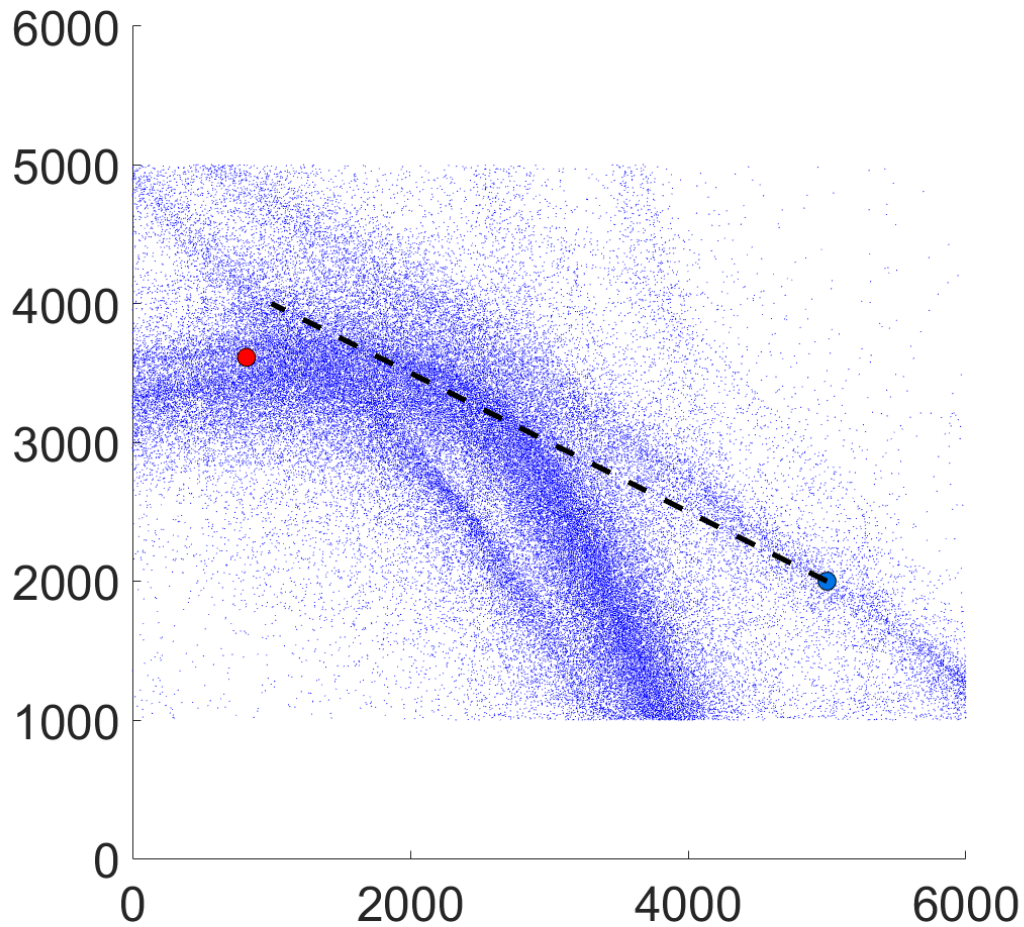


FIGURE 7.1: Scenario 1: The small blue dots are the underlying AIS data, the big, lighter blue dot is the ASV, and the red dot is the obstacle. The black line shows the desired trajectory.

and the desired speed is 10 m/s. In this scenario the ASV risks a head-on collision if it follows its desired trajectory. The scenario is shown in Figure 7.1. Note that, since this is a high data density area, the NCDM is equivalent to the modified NCDM with a low weight parameter α .

7.1.1 Original COLAV method

In Figure 7.2 the simulation of this method is shown at various time steps. Only every third constraint is plotted to simplify the figure. The predicted positions of the ASV are plotted as crosses, where the colors correspond to the constraints. In other words, a cross should not be able to be inside a circle of the same color.

The simulated ASV manages to avoid the obstacle. Figure 7.2b shows that the planned position for one minute into future deviate from the desired path because it cannot be inside the red circle. This also happens at prediction steps that are not plotted. In

the end, the ASV makes a slight turn to port to avoid the other vessel. COLREGS dictates that a vessel should turn to starboard when facing a head-on situation, but this is something this algorithm does not take into account and it is therefore arbitrary which side the ASV passes on.

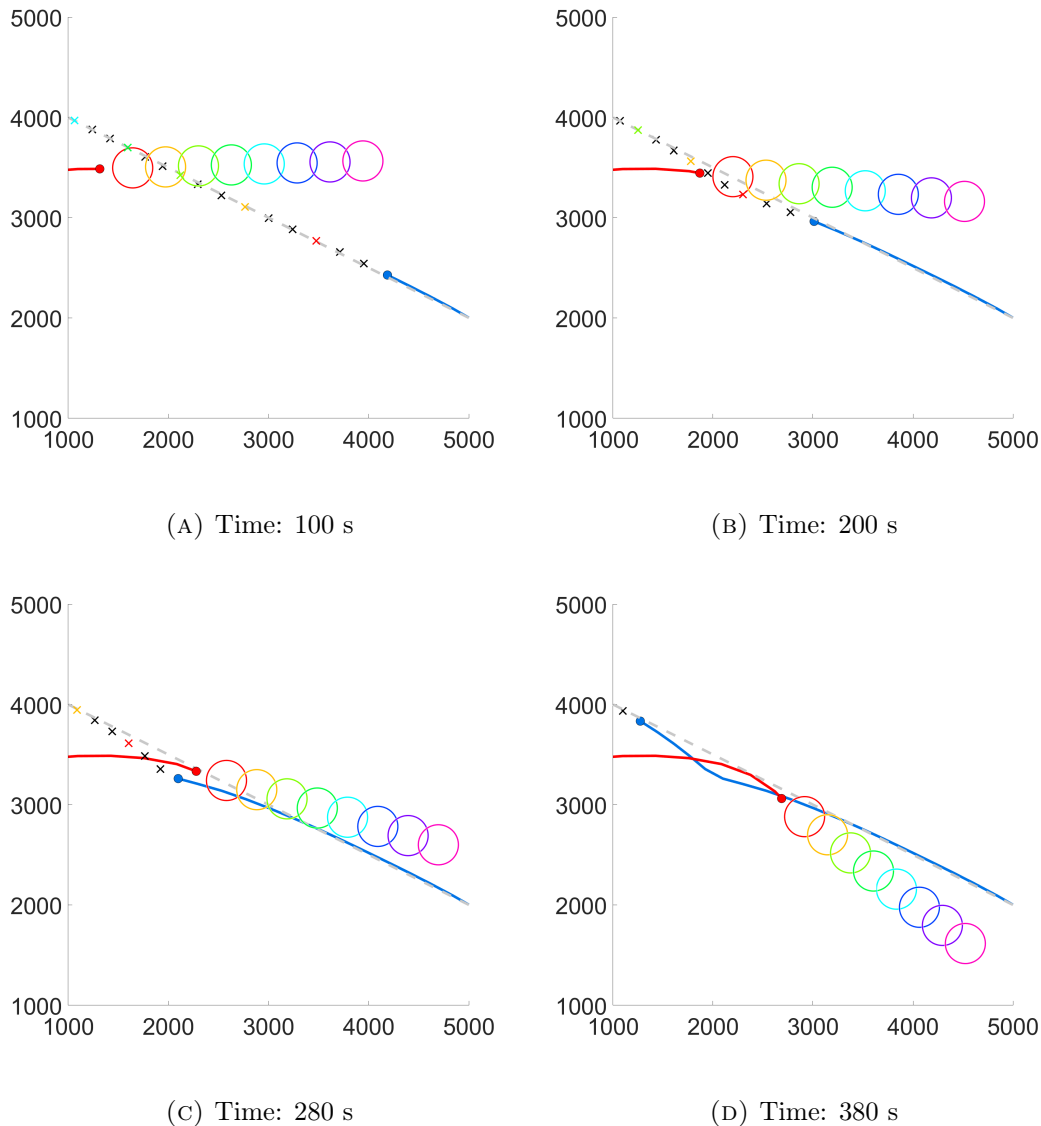


FIGURE 7.2: Scenario 1 the with original COLAV method: The blue and red lines show the ASV and the obstacle, respectively. The crosses show the predicted positions of the ASV. The multi-colored circles are the constraints. The axes are in meters.

7.1.2 New COLAV method with NCDM

Figure 7.3 is similar to Figure 7.2 with the exception that the 1 and 2 standard deviation equi-probability contours of the predicted GMMs at various time steps are plotted instead of the constraints. The crosses can be inside the contours plotted, but this will be more expensive.

From Figure 7.3 it is evident that a deviation is planned earlier than with the original COLAV method. A large deviation is planned in the beginning, but the planned deviation get smaller as time passes. In the end a deviation of similar size to the one using the original COLAV method is made.

Unlike the test with the original COLAV method, the ASV now turns to starboard to avoid the obstacle. This is because turning to port is now more expensive as the obstacle is predicted to be at the port side of the ASV in short time. The fact that this maneuver is COLREGS compliant and the maneuver performed with the original COLAV method is not comes down to luck as neither method defines which way to turn in different situations. However, the maneuver performed with the new COLAV method is safer as the ASV now avoids crossing in front of the moving obstacle, and avoids crossing the obstacle path altogether.

7.1.3 COLAV with modified NCDM

Figure 7.4 shows that predictions are now more uncertain, as the possibility that the vessel move straight ahead in addition to following historical trajectories is considered. It is therefore now more expensive to turn starboard and a turn to port is planned. In the end, a maneuver similar to the one from the test with the original COLAV method is performed.

7.2 Scenario 2: Head-on in low data density area

This scenario takes place in a low data density area as shown in Figure 7.5. The ASV now faces a head-on situation where both vessels are driving in a straight line towards each other. The desired speed is 10 m/s. The scenario is illustrated in Figure 7.5. The NCDM without modifications is not included in these tests since it is not able to produce predictions in areas with such low data density.

7.2.1 Original COLAV

Figure 7.6 shows that the ASV makes a small maneuver to avoid the other vessel. One might argue that maneuver happens too late or is not large enough. However, increasing the radius the circular constraints will solve this problem.

7.2.2 Modified NCDM

As seen in Figure 7.7 the method initially plans a rather large turn to port. However, this turn is later flipped to the other side and greatly reduced in size. In the end, the ASV ends up taking a very small turn and passes very close to the obstacle.

The problem the new COLAV method faces in this scenario is that the GMMs for the obstacle at small prediction steps have very little uncertainty. Therefore, the obstacles for small prediction steps cover a very small area and the ASV needs only make small turns to avoid them.

7.3 Discussion

In the first scenario the new COLAV method with the original NCDM appears to give the best performance. Using the modified NCDM gives a similar result to the original

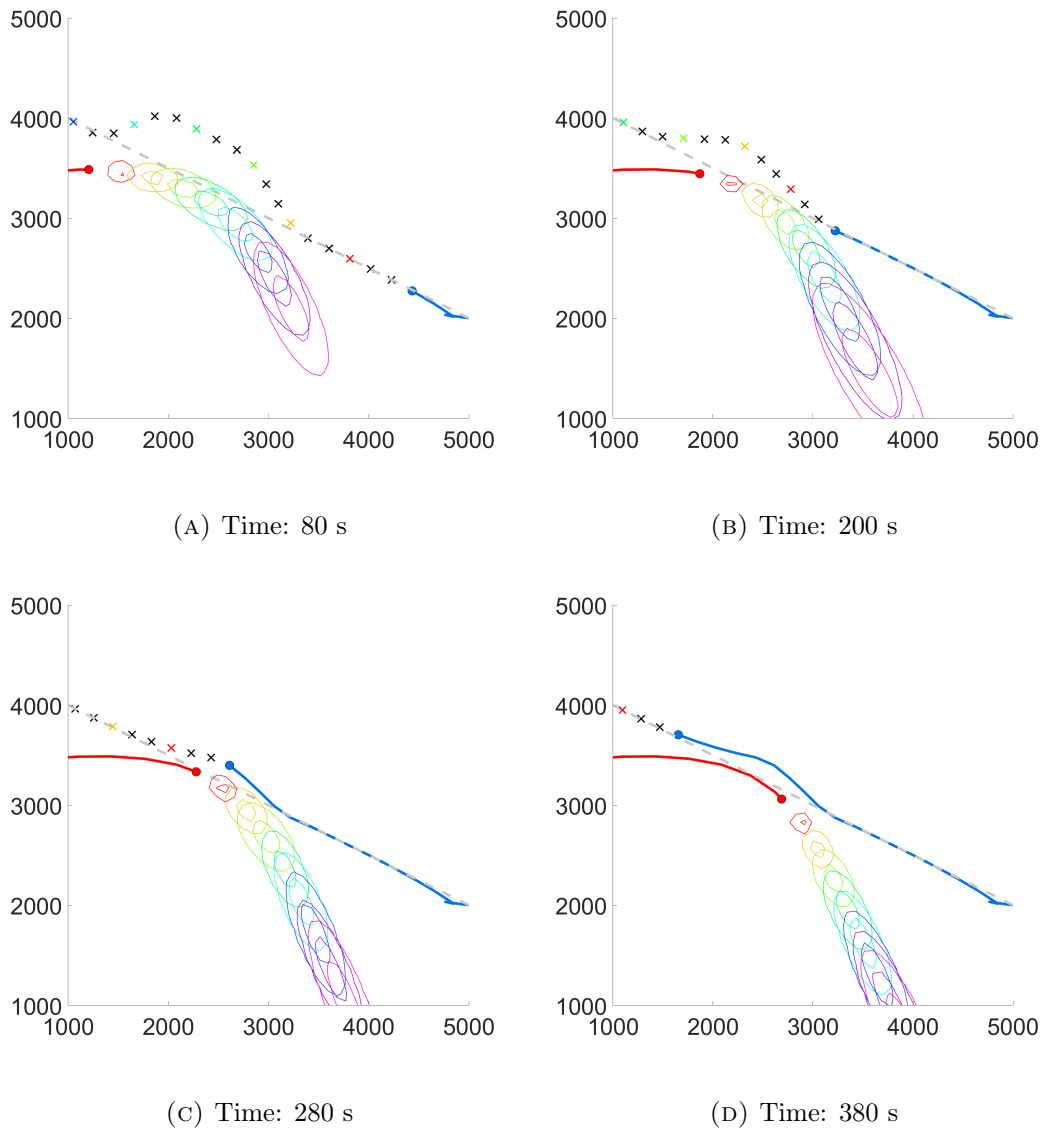


FIGURE 7.3: Scenario 1 with the new COLAV method and NCDM: The blue and red lines show the ASV and the obstacle, respectively. The crosses show the predicted positions of the ASV. The multi-colored contours are the 1 and 2 standard deviation equi-probability contours of the predicted GMMs.

COLAV method. Comparing the AIS data in Figure 7.1 to the plots in Figure 7.4 it might appear as if the modified NCDM overestimates the probability that the vessel maintains a constant velocity in this scenario, as there is plenty of data available. The modified NCDM was developed primarily to make the method usable in areas with low data density. Although an a weight parameter of $\alpha = 5000$ gave the most consistent results in Section 5.4, it might seem that a lower α is more suitable in a COLAV system.

In the second scenario the original COLAV method gave the better performance. It appears that the NCDM still struggles in low data density areas even after the modifications, although it is a vast improvement as the original NCDM cannot operate at all in such areas. An option to improve the results is to increase the covariance

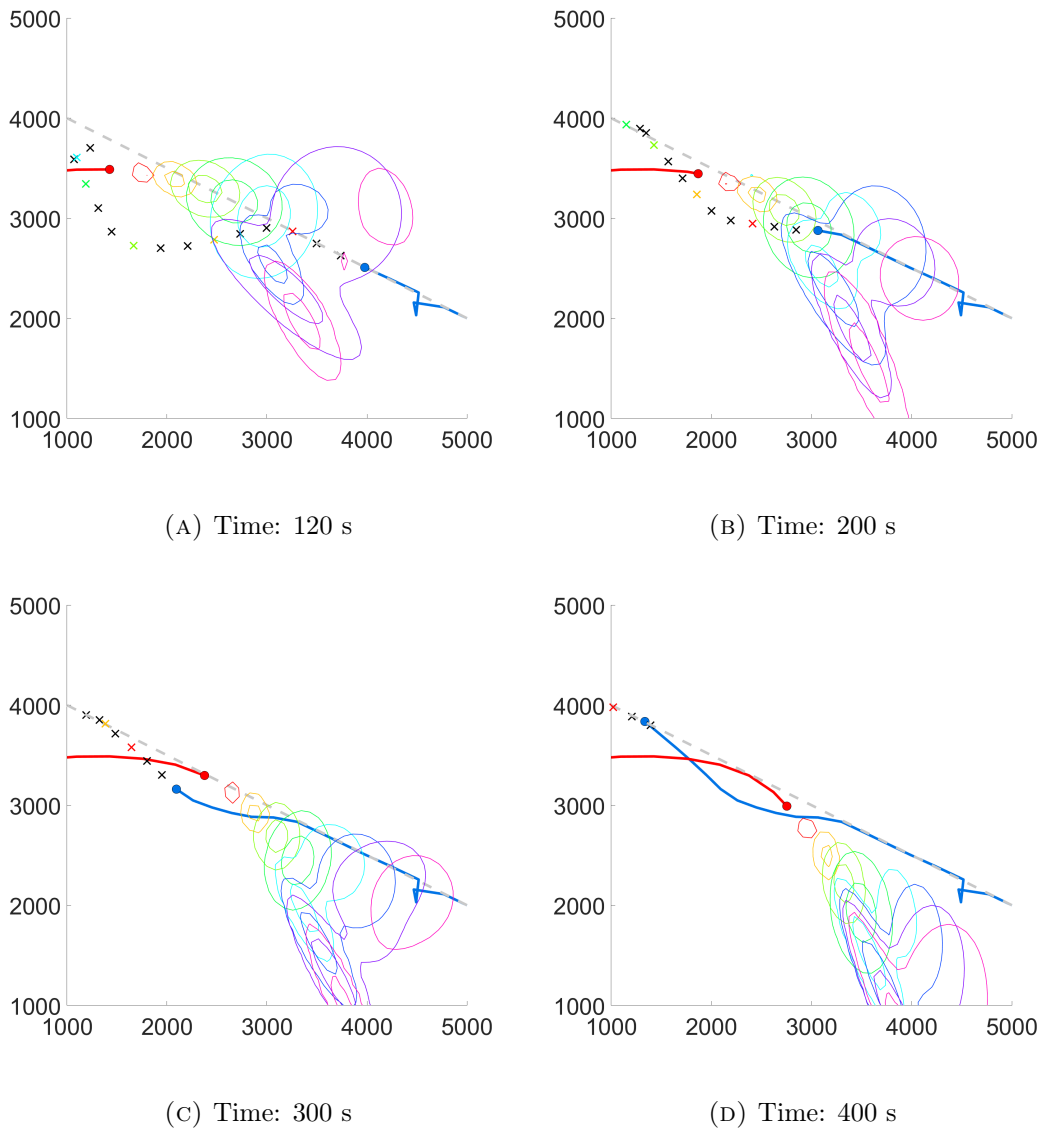


FIGURE 7.4: Scenario 1 with the new COLAV method and modified NCDM.

noise parameter, but this gives even more uncertainty at larger prediction steps as the uncertainty region of the CVM increases quadratically over time [7]. In addition, a higher noise parameter was shown to give worse consistency properties in Section 5.3.

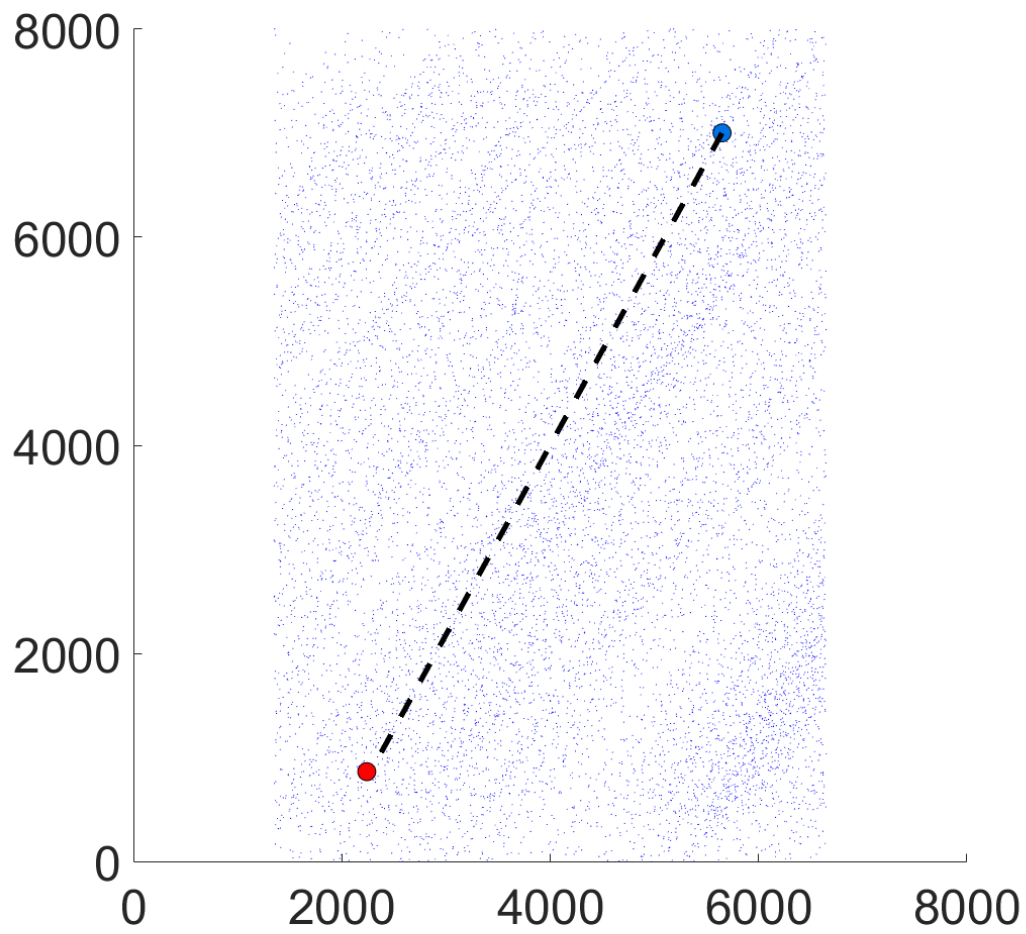
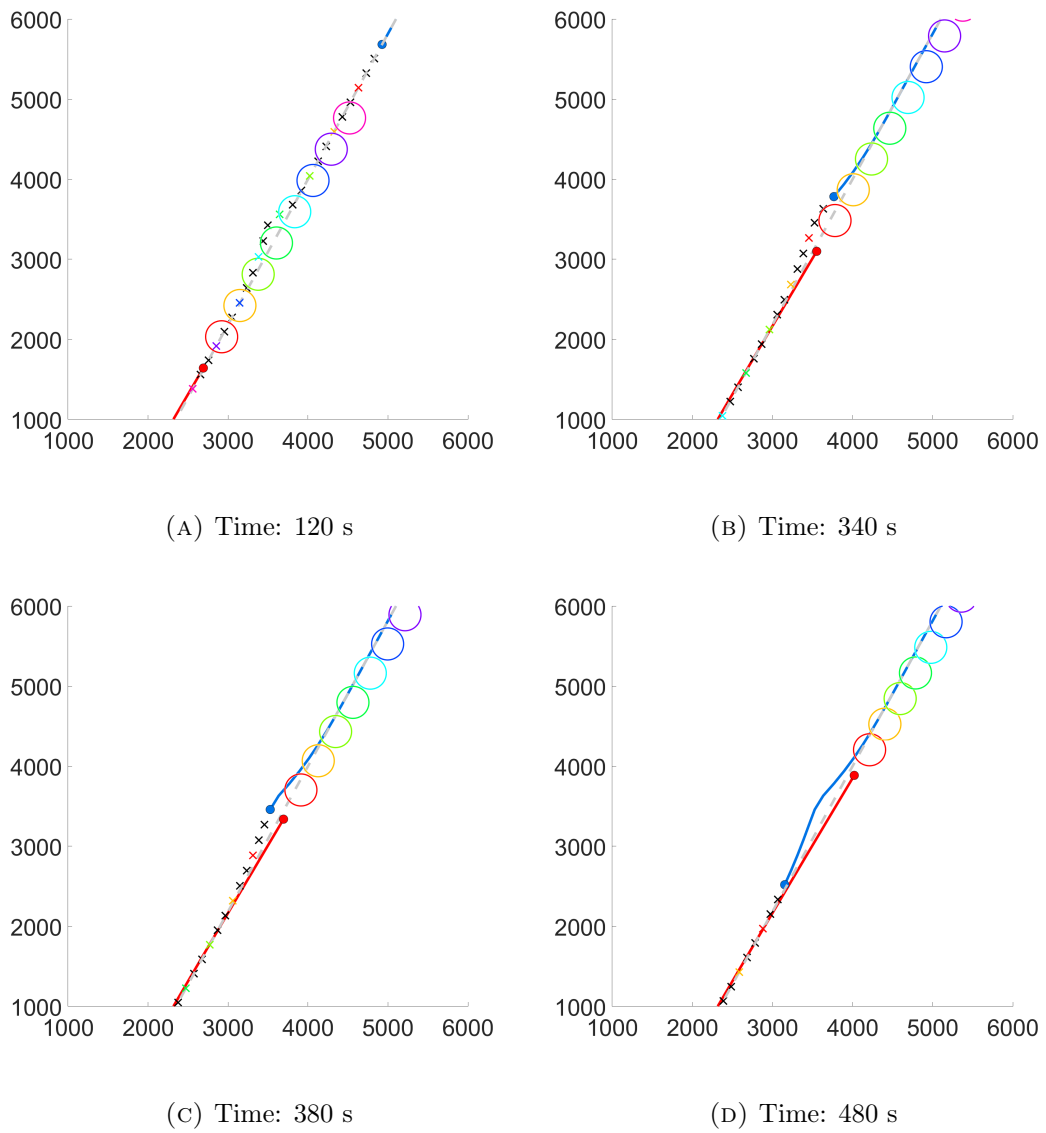
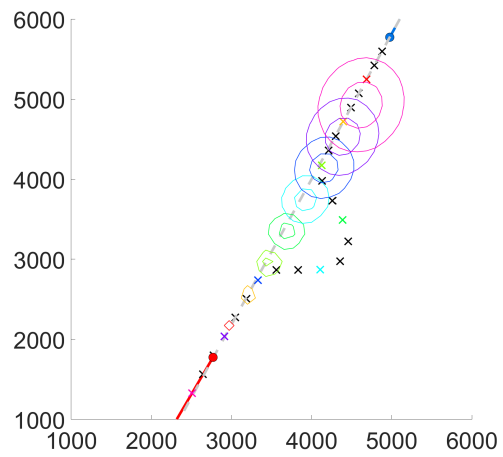
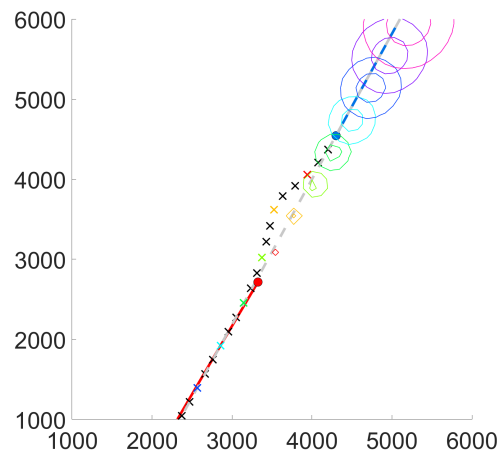


FIGURE 7.5: Scenario 2: The small blue dots are the underlying AIS data, the big, lighter blue dot is the ASV, and the red dot is the obstacle. The axes are in meters.

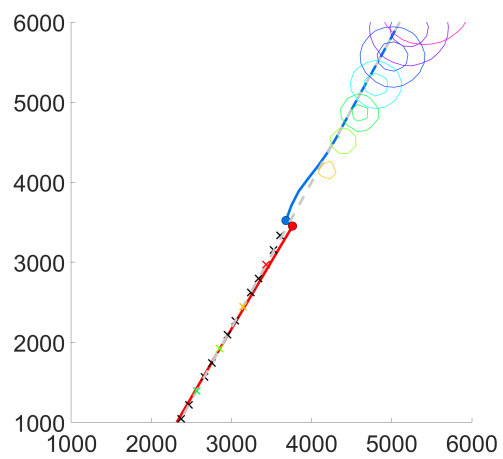




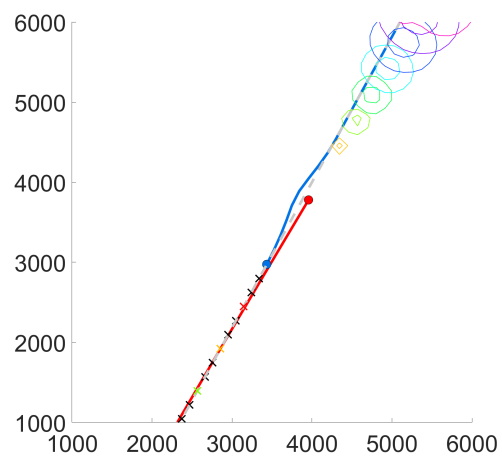
(A) Time: 140 s



(B) Time: 280 s



(C) Time: 400 s



(D) Time: 460 s

FIGURE 7.7: Scenario 2 with the new COLAV method and modified NCDM.

Chapter 8

Conclusion and further work

In this thesis a modified version of the NCDM which partly relies on the CVM is introduced and tested. The modified NCDM is, unlike the original NCDM, able to make predictions in areas with little or no AIS data available. The modifications introduced leads to significantly better covariance consistency properties, although the method's accuracy is somewhat reduced. However, adjusting a weight parameter decides how much the predictions rely upon either the original NCDM or the CVM, and the method is thus very flexible.

Further improvements to the NCDM have been attempted by investigating other methods for learning the parameters for the GMMs. The use of a method based on variational Bayesian inference to replace the EM algorithm did not improve the performance of the method. The tests conducted in this thesis show that the VB method results in worse covariance consistency properties and also an increase in computational time. Using the information criteria AIC and BIC to decide the number of components in a GMM also failed to show any improvement over the original criteria which considers the distance between the means of the GMM components.

The NCDM has been tested for dynamic obstacle prediction in a COLAV algorithm. These tests give an example of where the new MPC-based COLAV algorithm makes a safer maneuver than the original method when avoiding a turning vessel, although both methods are able to steer the ASV clear of the obstacle. A second example show that the new COLAV method struggles in areas with low data density. As the the original COLAV method is not dependent on AIS data, it performs better than the new method in this case.

There are still several issues to be solved regarding the use of the NCDM in a COLAV system. Therefore the following tasks are suggested to continue the work of this thesis:

- Investigate whether using a OU stochastic process to predict the position of a vessel can replace the use of the CVM in the modified NCDM.
- Conduct more qualitative tests in different collision situations: Investigate how the COLAV methods perform in crossing and overtaking situations.
- Conduct quantitative tests to measure and compare the performance of the different COLAV methods.

Appendix A

Accepted paper for the 21th
International Conference on
Information Fusion 2018

The Neighbor Course Distribution Method with Gaussian Mixture Models for AIS-based Vessel Trajectory Prediction

Bjørnar R. Dalsnes^{*†}, Simen Hexeberg[‡], Andreas L. Flåten^{*}, Bjørn-Olav H. Eriksen^{*} and Edmund F. Brekke^{*}

^{*} Department of Engineering Cybernetics

^{*} Norwegian University of Science and Technology (NTNU)

[‡] BearingPoint Norway AS

[†] bjornrda@stud.ntnu.no

Abstract—When operating an autonomous surface vessel (ASV) in a marine environment it is vital that the vessel is equipped with a collision avoidance (COLAV) system. This system must be able to predict the trajectories of other vessels in order to avoid them. The increasingly available automatic identification system (AIS) data can be used for this task. In this paper, we present a data-driven approach to predict vessel positions 5-15 minutes into the future using AIS data. The predictions are given as Gaussian Mixture Models (GMMs), thus the predictions give a measure of uncertainty and can handle multimodality. A nearest neighbor algorithm is applied on two different data structures. Tests to determine the accuracy and covariance consistency of both structures are performed on real data.

I. INTRODUCTION

In order for an autonomous surface vessel (ASV) to operate safely it is essential that it is equipped with a robust collision avoidance (COLAV) system. A major component of this system is the prediction of the future positions of other vessels. A simple solution to this problem is to assume that nearby vessels will continue with constant velocity. However, such predictions might not be sufficient for longer prediction horizons. In recent years automatic identification system (AIS) data has become more available and has been used to improve the trajectory predictions of vessels.

In previous work, vessel trajectories have been predicted by describing a vessel's velocity using a Ornstein-Uhlenbeck (OU) stochastic process. This was shown to give better results than to assume a near constant velocity [1].

Another approach for vessel movement prediction is to cluster trajectories based on historical AIS data and then assign an object's initial state to one of these clusters [2], [3], [4]. This approach can typically be divided into four sequential steps:

- 1) Cluster trajectories based on historical data.
- 2) Classify a new object to one of the clusters found in step 1.
- 3) Generate a representative trajectory for each cluster.
- 4) Predict the movement of the object along the representative trajectory found in step 3.

The trajectory clustering (TRACCLUS) algorithm presented in [5] is widely used for trajectory clustering. This algorithm uses density based clustering to detect trajectories. The algorithm partitions trajectories into smaller line segments and then clusters these line segments. The algorithm is further improved in [6] where it is made less sensitive to its decision parameters.

As opposed to TRACCLUS, the traffic route extraction for anomaly detection (TREAD) algorithm [2] was specifically designed for AIS data predictions. This algorithm defines waypoints that later are clustered instead of clustering trajectories directly. The waypoints are defined when a vessel enters or exits a pre-defined bounding box with minimal movement, as well as when it stays within the box for a certain amount of time. The algorithm is principally developed to detect low-likelihood anomalies that deviate from the main trajectories.

Clustering based methods like TRACCLUS and TREAD, as well as the OU method from [1], were developed for much larger prediction horizons than what is of interest in a COLAV system. In addition, the OU method requires the estimation of process parameters, which may not be straightforward. Therefore, a radically different and more data-driven approach was proposed in [7]. The key concept in this approach was a single point neighbor search (SPNS). Given an AIS message, the SPNS algorithm considers historical messages within a given radius, called close neighbors, to predict a course and speed of the vessel. Historical messages with courses that deviate by a certain amount from the vessel's course are discarded in order to avoid influence from opposite moving vessels in the predictions. What remains are messages that have a similar position and course as that of the vessel. The median course and speed of these close neighbors are calculated and used as a predicted course and speed of the vessel. The predicted course and speed is then used to calculate the future position by a given step length parameter. The same process is then applied on the newly predicted state and this is then repeated until a trajectory of desired length

is produced. The output of the SPNS algorithm is an array of waypoints with equal distances between the positions of any two subsequent messages.

The inability to estimate prediction uncertainty and the inability to handle branching of sea lanes are two of the SPNS algorithm's main shortcomings. The neighbor course distribution method (NCDM) was developed in the MSc thesis [8] to account for this. Whereas the output from the SPNS can be seen as a list of states which forms a single trajectory, the output from NCDM is a tree of states which forms several trajectories. Each individual trajectory is calculated in a similar manner as in the SPNS. The same set of close neighbors are used to predict the vessel's course and speed at each predicted position. However, while the SPNS predict the course and speed as the median course and speed of the closest neighbors, the NCDM samples course from the distribution of the neighbors' courses. The NCDM is thus able to predict trajectories in several branched sea lanes and it possess the ability to indicate an uncertainty measure of the predictions.

This paper builds upon the work done in [7] and [8]. We extend the NCDM by introducing a Gaussian Mixture Model (GMM) to represent the position of a vessel. The NCDM is implemented using both the original data structure that was used in [8] and a new data structure proposed in this paper, and the two implementations are tested on real AIS data.

The outline of this paper is as follows: Section II introduces the NCDM using both data structures. In Section III the methods are tested on a comprehensive set of AIS data from Trondheimsfjorden, Norway. Lastly, conclusions and suggestions for further work are given in Section IV and Section V.

II. NEIGHBOR COURSE DISTRIBUTION METHOD

This section describes the NCDM introduced in [8]. First the prediction tree is explained. Then the two different data structures are introduced: the AIS message structure from [8] and a new structure which utilizes the recent past trajectory of the vessel.

A. Prediction tree

The NCDM takes an initial state \mathbf{X}_1 of a ship as input. This initial state is the root node of the prediction tree, where each node represents a predicted state $\hat{\mathbf{X}}_{k,j}$. The tree has a depth index k and a width index j . $N_{k,j}$ gives the number of child nodes for node (k,j) , while J_k is the number of nodes at level k . In this paper we choose the number of children for the root node to be $N_{1,1} = J^{max}$, while the number of children at all other levels are set to one. In other words the tree has J^{max} branches which all originate in its root node. A predicted position $\hat{\mathbf{p}}_{k,j}$ is calculated from a random sample at every node in the tree. The method for obtaining the predicted position varies depending on the data structure used. The prediction

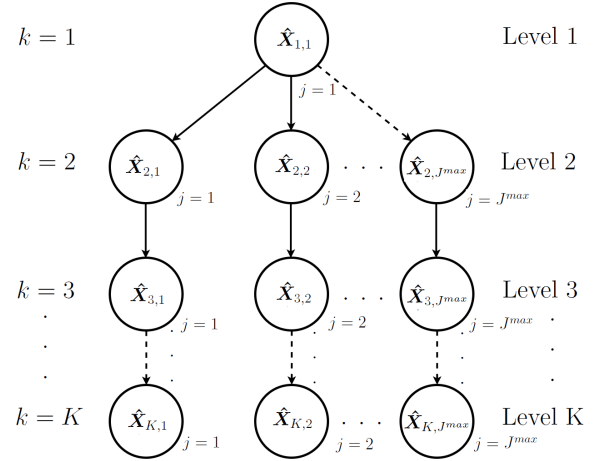


Fig. 1: Prediction tree structure. $N_{1,1} = J^{max}$, otherwise $N_{k,j} = 1$.

tree is illustrated in Figure 1.

Algorithm 1 Neighbor Course Distribution Method

1: **Input parameters:**

- \mathbf{X}_1 ▷ Initial state
- $N_{k,j}$ ▷ Number of child nodes from node (k,j)
- K ▷ Total number of tree levels

2: Set $\hat{\mathbf{X}}_{1,1} = \mathbf{X}_1$

3: **for** $k = 1$ to $K - 1$ **do**

4: $q = 0$ ▷ Indexing variable at level k

5: **for** $j = 1$ to J_k **do**

6: Find close neighbors

7: **for** $N_{k,j}$ iterations **do**

8: $q = q + 1$

9: Obtain random sample

10: Calculate the next position

11: Update $\hat{\mathbf{X}}_{k+1,q}$ based on the latest prediction

12: **end for**

13: **end for**

14: **end for**

B. Old data structure

The first structure, referred to as \mathbf{X}^A , is used in [7] and [8]. It consists of a list of AIS messages:

$$\mathbf{X}^A = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \dots \quad \mathbf{X}_n]^T, \quad (1)$$

where each message is given as

$$\mathbf{X}_i = [\text{MMSI}_i \quad t_i \quad \mathbf{p}_i \quad \chi_i \quad v_i], \quad (2)$$

where MMSI_{*i*}, *t_i*, *p_i*, *χ_i* and *v_i* are the unique vessel identification number, time stamp, position vector, course over

ground (COG) and speed over ground (SOG), respectively. The position vector can be written as $\mathbf{p}_i = [\lambda_i \ \phi_i]$ where λ_i and ϕ_i are the longitude and latitude in the WGS84 coordinate system.

The set of close neighbors (CNs) at step (k, j) is defined as

$$\mathbf{C}_{k,j} = \{\mathbf{X}_i | d(\hat{\mathbf{p}}_{k,j}, \mathbf{p}_i) \leq r_c, \chi_i \in I, \mathbf{X}_i \in \mathbf{X}\}, \quad (3)$$

where $d(\hat{\mathbf{p}}_{k,j}, \mathbf{p}_i)$ is the Euclidean distance between the predicted position and a position in the dataset calculated with the Haversine formula [9]. Further, r_c is the search radius and I is the interval of accepted course angles given by

$$I = [\chi_{k,j} - \Delta\chi, \ \chi_{k,j} + \Delta\chi], \quad (4)$$

where $\Delta\chi > 0$ is the maximum course angle deviation. In other words a state is considered a close neighbor to the predicted state if its position is within a radius r_c from the predicted position and if its course is within $\Delta\chi$ of the last predicted course.

A random sample is drawn from the CNs and the predicted course and speed are obtained from this sample. These values are used to update the state $\hat{\mathbf{X}}_{k+1,q}$. The position of the next state is given by:

$$\hat{\mathbf{p}}_{k+1,q} = \hat{\mathbf{p}}_{k,j} + \Delta l [\sin(\hat{\chi}_{k,j}) \ \cos(\hat{\chi}_{k,j})], \quad (5)$$

where $\hat{\mathbf{p}}_{k,j}$ is the ship's predicted position at node (k, j) , $\hat{\chi}_{k,j}$ is the predicted course at node (k, j) and Δl is the fixed step parameter. The time stamp is updated using the following equation:

$$\hat{t}_{k+1,q} = \hat{t}_{k,j} + \frac{\Delta l}{\hat{v}_{k,j}}. \quad (6)$$

This means that the new state is given as

$$\hat{\mathbf{X}}_{k+1,q} = [\text{MMSI}_i \ \hat{t}_{k+1,q} \ \hat{\mathbf{p}}_{k+1,q} \ \hat{\chi}_{k,j} \ \hat{v}_{k,j}], \quad (7)$$

where $\hat{\chi}_{k,j}$ and $\hat{v}_{k,j}$ are the randomly drawn course and speed respectively.

C. New data structure

The second structure, referred to as \mathbf{X}^B , structures the data as a list of sub-trajectories. A sub-trajectory \mathbf{S} consists of n points

$$\mathbf{S} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \cdots \ \mathbf{p}_n], \quad (8)$$

where \mathbf{p} is a point given by longitude and latitude. There is an equal amount of time elapsed between any two subsequent points. These points are found using cubic spline interpolation on the points given in the AIS messages. A trajectory is given by:

$$\mathbf{T} = [\mathbf{S}_1 \ \mathbf{S}_2 \ \cdots \ \mathbf{S}_M]^T, \quad (9)$$

and the entire dataset is given by a list of trajectories:

$$\mathbf{X}^B = [\mathbf{T}_1 \ \mathbf{T}_2 \ \cdots \ \mathbf{T}_N]^T. \quad (10)$$

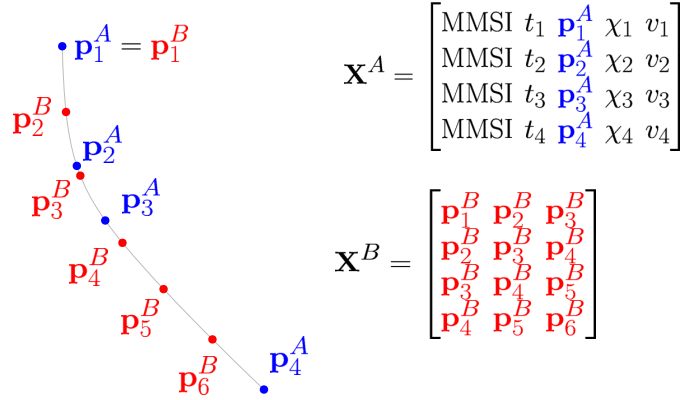


Fig. 2: An example of a short trajectory represented with both data structures: The blue dots represents the ship's positions as given in the original AIS data. The red dots are the new data points obtained using interpolation. Here the sub-trajectories consist of $n = 3$ points.

A comparison between the old and the new structure can be seen in Figure 2. Reformating the dataset from the original structure \mathbf{X}^A to the new structure \mathbf{X}^B involves three main steps:

1) *Find trajectories from the original dataset:* Points that belong to the same MMSI number and have less than 15 minutes between two subsequent points are considered as part of the same trajectory. The time limit makes sure that trajectories from vessels which leave and later enter the dataset window or which stay in port for a long time, are split into separate trajectories

2) *Interpolate trajectories to get new data points:* Cubic spline interpolation is used to extract new data points at a specified time interval. The new data points now form a new trajectory.

3) *Create sub-trajectories of n points:* The first sub-trajectory consists of the first n points of the new data points. This is the first row of the data structure. The next row is a new sub-trajectory shifted one point forward, thus creating a structure of partly overlapping sub-trajectories. This continues until the end of the trajectory. With this method a trajectory of k points will result in $k - n + 1$ sub-trajectories of n points.

Sub-trajectories can be seen as $2 \times n$ dimensional points, where n is the number of points of the sub-trajectories in \mathbf{X}^B . The distance between sub-trajectories is thus the Euclidean distance between these points. Close neighbors are found as in (3) where $d(\hat{\mathbf{p}}_{k,j}, \mathbf{p}_i)$ now is replaced by the distance between the two sub-trajectories. A key difference now is that the course tolerance parameter $\Delta\chi$ is no longer needed. Instead of choosing an arbitrary parameter for accepted courses, sub-trajectories with similar courses to the initial state are selected because these sub-trajectories

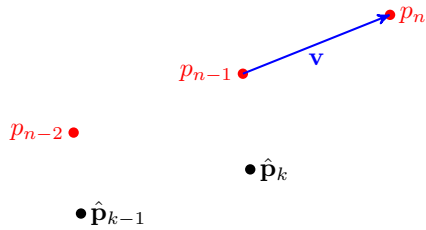


Fig. 3: The sub-trajectory shown in red is a close neighbor of \mathbf{X}_1 if the Euclidean distance between $[\hat{\mathbf{p}}_{k-1}, \hat{\mathbf{p}}_k]$ and $[\mathbf{p}_{n-2}, \mathbf{p}_{n-1}]$ is less than r_c . The location of $\hat{\mathbf{p}}_{k+1}$ is determined by adding \mathbf{v} to $\hat{\mathbf{p}}_k$.

are closer to the initial state. Two nearby sub-trajectories that point in opposite directions will still have a large distance between each other while two sub-trajectories that point in roughly the same direction will have a small distance between each other. Sub-trajectories with similar courses will thus be considered close neighbors. The step length parameter Δl is also no longer needed as predictions are made with fixed time steps as defined by the data structure.

The input in this case is a sub-trajectory of $n - 1$ points where the last point will be the first of the predicted trajectory. This state is compared to the $n - 1$ first columns of \mathbf{X}^B to find close neighbors. A random sample (a sub-trajectory from \mathbf{X}^B) is drawn from the CNs. From this sub-trajectory a vector $\mathbf{v} = \mathbf{p}_n - \mathbf{p}_{n-1}$ is obtained. As shown in Figure 3, this vector is added to $\hat{\mathbf{p}}_{k,j}$ to obtain $\hat{\mathbf{p}}_{k+1,q}$. The state, which is a sub-trajectory, is updated by removing the first point from the sub-trajectory and adding the newly calculated one at its end.

An advantage with the new data structure is that vessels with similar speeds are more likely to be considered close neighbors. This is because sub-trajectories of points are compared instead of just single points. Intuitively, sub-trajectories of similar length will be considered closer than sub-trajectories of different length.

D. Gaussian mixture representation

For both structures the predicted future position of a vessel is given by a number of J^{max} points taken from the desired level of the prediction tree. A GMM is then fitted to these points to give a probabilistic model of the future position. This is done using the Expectation Maximization (EM) algorithm which will fit the maximum likelihood GMM for the given points. The number of components used are increased until the means of two components are less than a pre-specified distance apart, then one less component is used. It is also possible to consider a model selection criterion such as AIC [10] or BIC [11] instead of looking at the distance between the means, but in the context of vessel trajectories we are mostly interested

in a multimodal distribution to enable predictions in branching sea lanes.

III. TESTS AND RESULTS

The NCDM is tested on AIS data gathered during 2015 in Trondheimsfjorden, Norway.

A. Test setup

The dataset is initially divided into a training set, from where the and a test set. The first 90% of the data points are used as training data, while the remaining 10% are used for tests. Both structures are tested to predict vessel positions 5, 10 and 15 minutes into the future from an initial state. We use the same prediction tree as described in Section II-A with $J^{max} = 200$, i.e., the tree has 200 branches which all originate in its root node. All tests were done on $N = 400$ initial states randomly sampled from the test set without replacement. The close neighbors of this sample are obtained from the training set. A test of a given method for a given time horizon on N initial states will from here be referred to as a 'test' while a test of a single initial state will be referred to as an 'individual test'.

The initial states and their corresponding trajectories must fulfill one requirement: The time at the end of the trajectory minus the time of the initial state must be larger or equal to the test time. This ensures that there is a true position in the test trajectory that the prediction can be compared against.

An individual test will be discarded if more than 25% of the predicted positions at the desired level of the prediction tree are identical. This is an indication that there is not enough data in the area of the initial state for the algorithm to make any reasonable predictions, and it is also difficult to fit a GMM to the points in this case.

Table I shows the decision parameters used for the test of the NCDM using data structure \mathbf{X}^A . The trajectories produced using this structure have a fixed step length between subsequent states and each state has a predicted time stamp. Most likely they will not have time stamps equal to the test time (5, 10 or 15 minutes). It is therefore necessary to interpolate the trajectories to obtain a position at the desired time.

TABLE I: Decision parameters for NCDM with \mathbf{X}^A

| Decision parameter | Value | Description |
|--------------------|-------|----------------------------------|
| r_c | 100m | Search radius for CNs |
| Δt | 100m | Prediction step length |
| $\Delta \chi$ | 35° | Maximum course deviation for CNs |

Data structure \mathbf{X}^B was tested using the decision parameters in Table II. The search radius r_c is set to the same as for the first data structure. Parameters n and t are chosen when generating \mathbf{X}^B as described in Section II-C. The choices of n and t are somewhat arbitrary. The minimum value for n is three and was chosen for simplicity. Higher values of n were tried

with no visible improvement, but this has not been extensively tested. The average SOG for the messages in the dataset is roughly 5.4m/s. If similar step length as with \mathbf{X}^A was to be chosen the time step would be $t = (100/5.4)s \approx 18.5s$. This was tried, but with such small sub-trajectories it was found that sub-trajectories pointing in opposite directions often were considered close neighbors. Therefore, t was increased to 60 seconds.

TABLE II: Decision parameters for NCDM with \mathbf{X}^B

| Decision parameter | Value | Description |
|--------------------|-------|---|
| r_c | 100m | Search radius for CNs |
| n | 3 | Number of points in each sub-trajectory |
| t | 60s | Time between each point |

B. Performance measures

Two performance measures are used to determine the quality of a set of predictions: the root mean square error (RMSE) and a generalized concept of filter consistency, also sometimes referred to as credibility.

The RMSE is used to measure the accuracy of a prediction. The lower the RMSE, the better is the prediction. We define the RMSE as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{J^{max}} \|\mathbf{p} - \hat{\mathbf{p}}_i\|^2}{J^{max}}}, \quad (11)$$

where \mathbf{p} is the true position, $\hat{\mathbf{p}}_i$ is the predicted position at iteration i and J^{max} is the number of individual predictions as defined in Section II-A. This measure gives an idea of how close the mean of the predictions is to the true value. The RMSE is generally not very well suited for multimodal distributions. The mean of a multimodal distribution might be in an empty area between components. This is, however, a rare scenario.

The second performance measure is used to determine if the GMM produced is consistent. Filter consistency is colloquially explained as follows in [12]: The estimation error should have magnitude commensurate with the corresponding covariance that is yielded by the estimator. In order to deal with multimodality, consistency of the prediction methods is measured using the same method as in [13]. First, the Probability Density Function (PDF) value as given by the GMM for the true position is noted for each individual test. This value, called f , is then compared to the maximum value of the same PDF, called f_{max} . The ratio $f/f_{max} \in [0, 1]$ thus serves as a quality measure where a single prediction is better the closer the ratio is to one. The distribution of f/f_{max} for N predictions will then give an idea of the consistency. The values f and f_{max} are illustrated in Figure 4.

C. Results

The median values of the RMSE of all the methods for each test time is given in Table III. The same information is plotted

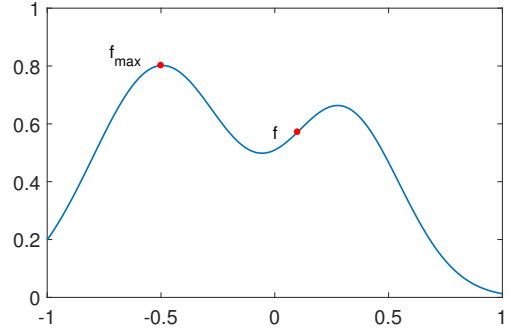


Fig. 4: Performance measure illustrated using one-dimensional data. The true value is $x = 0.1$. In this case $f/f_{max} \approx 0.57/0.8 \approx 0.71$.

in Figure 5. As can be seen, the accuracy of the NCDM is roughly the same using both data structures. It can seem as if \mathbf{X}^B has a slight advantage for short prediction horizons while \mathbf{X}^A is more accurate for longer horizons. However, the sample size might be too small to draw any definitive conclusions.

TABLE III: Median RMSE for each test time

| | NCDM with \mathbf{X}^A | NCDM with \mathbf{X}^B |
|--------|--------------------------|--------------------------|
| 5 min | 282m | 269m |
| 10 min | 509m | 537m |
| 15 min | 661m | 758m |

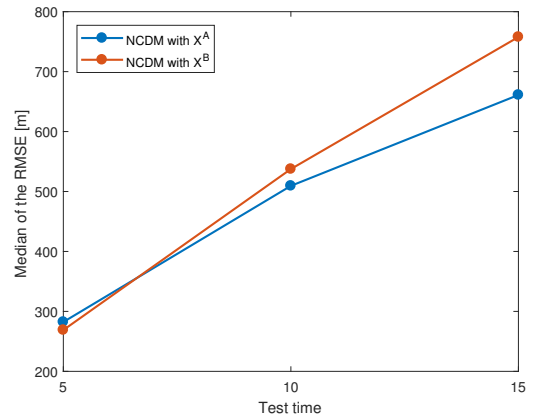
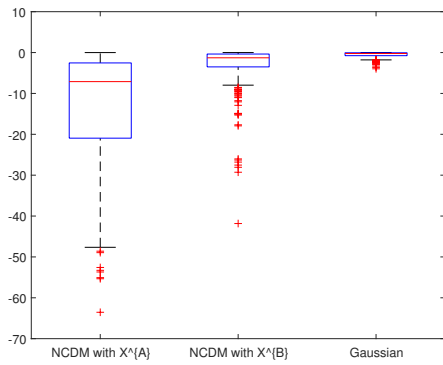
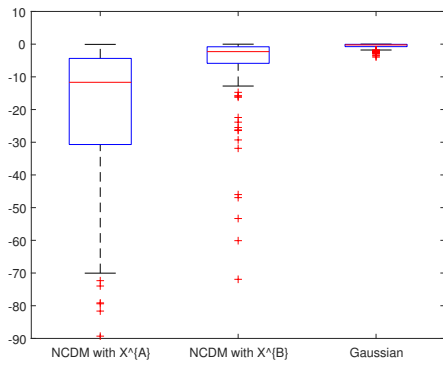


Fig. 5: Median RMSE over time

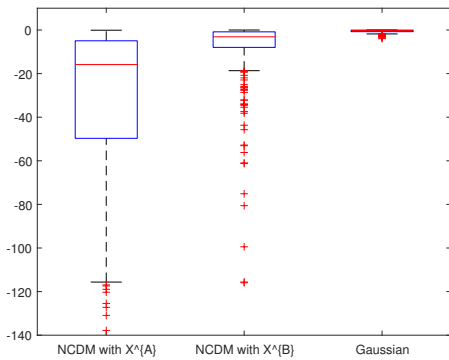
The natural logarithms of the f/f_{max} ratios are displayed in Figure 6 in order to investigate the consistency. The far-right box plots are values sampled from a Gaussian distribution (also shown in Figure 6d). This plot should therefore exhibit ideal consistency properties and will serve as a comparison for the box plots to the left. A consistent prediction method would have a box plot similar to the one from the Gaussian distribution.



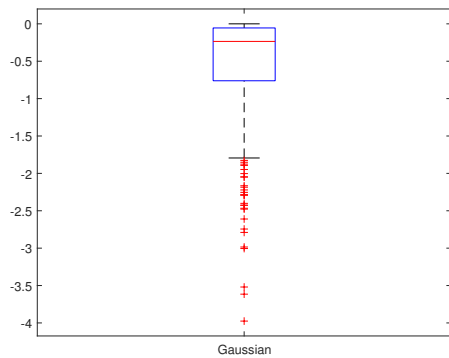
(a) Test time: 5 min.



(b) Test time: 10 min.



(c) Test time: 15 min.



(d) Close-up of Gaussian

Fig. 6: Box plot of the PDF-ratios

The consistency behavior deteriorates as the test time increases across all methods (notice the change of y-axis). It can be seen that NCDM using \mathbf{X}^B over \mathbf{X}^A results in a significant improvement in the consistency. However, compared to the Gaussian it is evident that the consistency properties of \mathbf{X}^B still remains far from ideal.

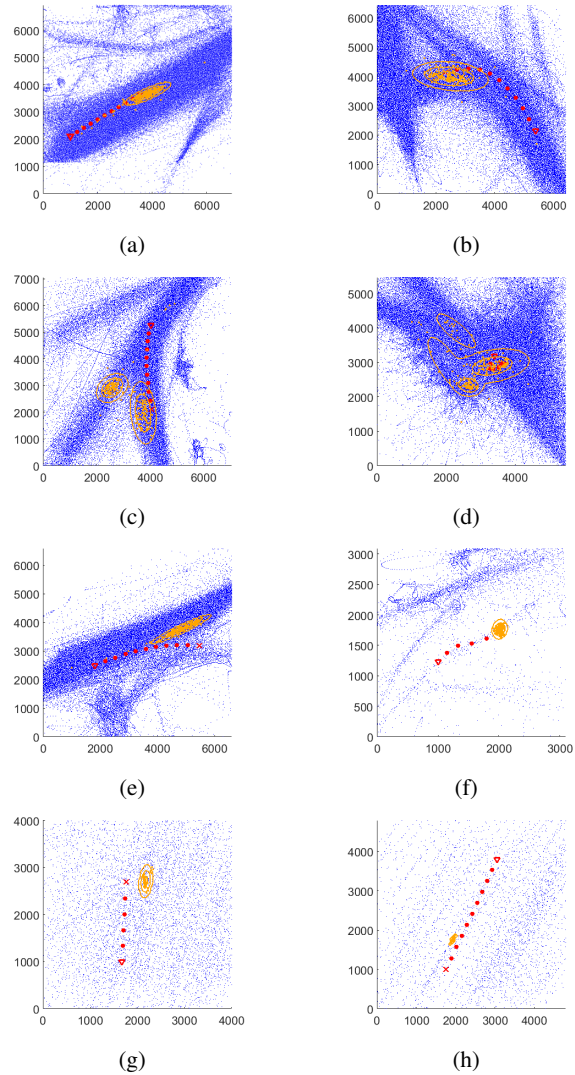


Fig. 7: The blue dots are reported ship positions from the AIS dataset. The red dots indicate the true trajectory of a ship with the triangle representing the initial position and the cross the true position at the time the position is predicted. The orange dots are predicted positions and the orange lines correspond to the 1, 2 and 3 standard deviation equi-probability contours of the GMM fitted to them. The axis are in meters.

Figure 7 shows plots where the true trajectories of vessels are compared to their predicted positions produced by the NCDM. They are not intended to show a representative selection of predictions or to give a comparison of the two structures, but are chosen to highlight the strengths and

weaknesses of the NCDM in general.

Figure 7a and Figure 7b show two typical outcomes where relatively accurate predictions are made. Figure 7c and Figure 7d show cases where the algorithm has produced multimodal predictions, Figure 7c in particular gives a good example of how the algorithm is able to handle branching. However, Figure 7e shows an example where the algorithm fails to produce a component for the less traveled lane. Figure 7f, Figure 7g and Figure 7h show predictions made in areas with sparse data density. The first produces a good prediction, while the last two highlight a major weakness of the algorithm. In areas with sparse data density it often makes overconfident and inaccurate predictions. The plots also show that there is usually more uncertainty in the speed prediction than in the prediction of the course (best illustrated in Figure 7a). The same conclusion was also reached in [8].

IV. CONCLUSION

The NCDM is able to give a probabilistic position prediction of vessels. The predicted position distribution can be multimodal and the algorithm is thus able to handle branching. A new data structure was developed to use data from the recent trajectories of vessels in the predictions. The accuracy of the predictions produced using the NCDM with the new structure are similar to the ones using the old structure. However, the uncertainty evaluations of the predictions are significantly more reliable with the new structure. There is still a clear potential for improvements in consistency, although it should be noted that the algorithm is intended to be used in a highly proactive manner, i.e., only for suggesting how other ships possibly may move 5-15 minutes into the future. This may relax the requirements to consistency somewhat and future research is needed to determine what is acceptable.

V. SUGGESTIONS FOR FUTURE WORK

The performance of the NCDM in areas with low data density has to be improved. A way of compensating for the lack of data may be to include the possibility that the vessel moves straight ahead with constant speed in such areas. Furthermore, the current method for determining the number of components for the EM algorithm is not ideal and may be improved by using automatic model selection [14]. The next step is to then assess the suitability of the NCDM as part of a COLAV system [15]. Another approach for trajectory prediction using Gaussian Process Regression can also be investigated.

ACKNOWLEDGMENT

This work was supported by the Research Council of Norway through projects 223254 (Centre for Autonomous Marine Operations and Systems at NTNU) and 244116/O70 (Sensor Fusion and Collision Avoidance for Autonomous Marine Vehicles). The authors would like to express great gratitude to DNV GL, especially Øystein Engelhardtson and Hans Anton Tvete, for providing real-world AIS data. The work of the last author is funded by DNV GL.

REFERENCES

- [1] Leonardo M Millefiori et al. "Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.5 (2016), pp. 2313–2330.
- [2] Giuliana Pallotta, Michele Vespe, and Karna Bryan. "Traffic Knowledge Discovery from AIS Data". In: *16th International Conference on Information Fusion (Fusion)*. IEEE, Istanbul, Turkey, 2013.
- [3] Fabio Mazzarella, Michele Vespe, and Carlos Santamaria. "SAR Ship Detection and Self-Reporting Data Fusion Based on Traffic Knowledge". In: *IEEE Geoscience and Remote Sensing Letters* vol. 12, no. 8 (Aug. 2015), pp. 1685–1689.
- [4] Fabio Mazzarella, Virginia Fernandez Arguedas, and Michele Vespe. "Knowledge-based vessel position prediction using historical AIS data". In: *2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, Oct. 2015.
- [5] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. "Trajectory clustering". In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*. Association for Computing Machinery (ACM), 2007.
- [6] Chen Jiashun. "A new trajectory clustering algorithm based on TRACCLUS". In: *Proceedings of 2nd International Conference on Computer Science and Network Technology*. IEEE.
- [7] Simen Hexeberg et al. "AIS-based vessel trajectory prediction". In: *20th International Conference on Information Fusion (Fusion)*. IEEE, Xi'an, China, 2017.
- [8] Simen Hexeberg. "AIS-based Vessel Trajectory Prediction for ASV Collision Avoidance". MA thesis. Norwegian University of Science and Technology, 2017.
- [9] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185.
- [10] Hirotugu Akaike. "A new look at the statistical model identification". In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.
- [11] Gideon Schwarz et al. "Estimating the dimension of a model". In: *The annals of statistics* 6.2 (1978), pp. 461–464.
- [12] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*. Vol. 19. YBs London, UK: 1995.
- [13] Edmund Brekke and Mandar Chitre. "Bayesian multi-hypothesis scan matching". In: *OCEANS*. IEEE, Bergen, Norway, 2013.
- [14] Lei Shi, Shikui Tu, and Lei Xu. "Learning Gaussian mixture with automatic model selection: A comparative study on three Bayesian related approaches". In: *Frontiers of Electrical and Electronic Engineering in China* 6.2 (2011), pp. 215–244.

- [15] Bjørn-Olav Holtung Eriksen and Morten Breivik. “MPC-based Mid-level Collision Avoidance for ASVs using Nonlinear Programming”. In: *Proc. of IEEE CCTA - to appear*. Kohala Coast, Hawai’i, 2017.

References

- [1] Simen Hexeberg. “AIS-based Vessel Trajectory Prediction for ASV Collision Avoidance”. MA thesis. NTNU, 2017.
- [2] Bjørn-Olav Eriksen and Morten Breivik. “MPC-Based mid-level collision avoidance for asvs using nonlinear programming”. In: *Control Technology and Applications (CCTA), 2017 IEEE Conference on*. IEEE. 2017, pp. 766–772.
- [3] Hans-Christoph Burmeister et al. “Can unmanned ships improve navigational safety?” In: *Proceedings of the Transport Research Arena, TRA 2014, 14-17 April 2014, Paris*, 2014.
- [4] Oskar Levander. “Autonomous ships on the high seas”. In: *IEEE Spectrum* 54.2 (2017), pp. 26–31.
- [5] Yoshiaki Kuwata et al. “Safe maritime autonomous navigation with COLREGS, using velocity obstacles”. In: *IEEE Journal of Oceanic Engineering* 39.1 (2014), pp. 110–119.
- [6] Tor Arne Johansen, Tristan Perez, and Andrea Cristofaro. “Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment”. In: *IEEE transactions on intelligent transportation systems* 17.12 (2016), pp. 3407–3422.
- [7] Leonardo M Millefiori et al. “Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction”. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.5 (2016), pp. 2313–2330.
- [8] Pasquale Coscia et al. “Multiple Ornstein-Uhlenbeck Processes for Maritime Traffic Graph Representation”. In: *IEEE Transactions on Aerospace and Electronic Systems* (2018).
- [9] Giuliana Pallotta, Michele Vespe, and Karna Bryan. “Traffic Knowledge Discovery from AIS Data”. In: *The 16th international conference on information fusion*. Istanbul, Turkey, 2013.
- [10] Fabio Mazzarella, Michele Vespe, and Carlos Santamaria. “SAR Ship Detection and Self-Reporting Data Fusion Based on Traffic Knowledge”. In: *IEEE Geoscience and Remote Sensing Letters* vol. 12, no. 8 (Aug. 2015), pp. 1685–1689.
- [11] Fabio Mazzarella, Virginia Fernandez Arguedas, and Michele Vespe. “Knowledge-based vessel position prediction using historical AIS data”. In: *2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, Oct. 2015.
- [12] Simen Hexeberg et al. “AIS-based vessel trajectory prediction”. In: *Information Fusion (Fusion), 2017 20th International Conference on*. IEEE. Xi’an, China, 2017, pp. 1–8.
- [13] Bjørnar R. Dalsnes et al. “AIS-based vessel trajectory prediction with Gaussian mixture models”. In: *5th year specialization project at Engineering Cybernetics*. NTNU. 2017.
- [14] Steven M LaValle. “Rapidly-exploring random trees: A new tool for path planning”. In: (1998).

- [15] Hao-Tien Lewis Chiang and Lydia Tapia. “COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 2024–2031.
- [16] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. “The dynamic window approach to collision avoidance”. In: *IEEE Robotics & Automation Magazine* 4.1 (1997), pp. 23–33.
- [17] Bjørn-Olav H Eriksen et al. “A modified dynamic window algorithm for horizontal collision avoidance for AUVs”. In: *Control Applications (CCA), 2016 IEEE Conference on*. IEEE. 2016, pp. 499–506.
- [18] Bjørn-Olav Holtung Eriksen et al. “Radar-based Maritime Collision Avoidance using Dynamic Window”. In: *Proc. of IEEE Aerospace*. Big Sky, Montana, USA, 2018.
- [19] Øivind Aleksander G Loe. “Collision avoidance for unmanned surface vehicles”. MA thesis. Institutt for teknisk kybernetikk, 2008.
- [20] Inger Berge Hagen et al. “MPC-based Collision Avoidance Strategy for Existing Marine Vessel Guidance Systems”. In: *Proc. of IEEE ICRA*. 2018.
- [21] International Maritime Organization. *Automatic Identification Systems (AIS)*. 2017. URL: <http://www.imo.org/en/OurWork/safety/navigation/pages/ais.aspx> (visited on 12/18/2017).
- [22] X Rong Li and Vesselin P Jilkov. “Survey of maneuvering target tracking. Part I. Dynamic models”. In: *IEEE Transactions on aerospace and electronic systems* 39.4 (2003), pp. 1333–1364.
- [23] Erik F Wilthil, Andreas L Flåten, and Edmund F Brekke. “A Target Tracking System for ASV Collision Avoidance Based on the PDAF”. In: *Sensing and Control for Autonomous Vehicles*. Springer, 2017, pp. 269–288.
- [24] Richard A Redner and Homer F Walker. “Mixture densities, maximum likelihood and the EM algorithm”. In: *SIAM review* 26.2 (1984), pp. 195–239.
- [25] Hirotugu Akaike. “A new look at the statistical model identification”. In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.
- [26] Gideon Schwarz et al. “Estimating the dimension of a model”. In: *The annals of statistics* 6.2 (1978), pp. 461–464.
- [27] Adrian Corduneanu and Christopher M Bishop. “Variational Bayesian model selection for mixture distributions”. In: *Artificial intelligence and Statistics*. Vol. 2001. Morgan Kaufmann Waltham, MA. 2001, pp. 27–34.
- [28] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.
- [29] Kenneth P Burnham and David R Anderson. “Multimodel inference: understanding AIC and BIC in model selection”. In: *Sociological methods & research* 33.2 (2004), pp. 261–304.
- [30] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*. Vol. 19. YBs London, UK: 1995.
- [31] Edmund Brekke and Mandar Chitre. “Bayesian multi-hypothesis scan matching”. In: *OCEANS, Bergen, 2013 MTS/IEEE*. IEEE. 2013.
- [32] Dale E Seborg et al. *Process dynamics and control*. John Wiley & Sons, 2010.

-
- [33] Joel Andersson. “A General-Purpose Software Framework for Dynamic Optimization”. PhD thesis. Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium: Arenberg Doctoral School, KU Leuven, Oct. 2013.