Håvard Gjøsæter

A Method for Detecting Near-Misses from AIS Data

Master's thesis in Marine Technology
Supervisor: Bjørn Egil Asbjørnslett
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU
Norwegian University of
Science and Technology

Håvard Gjøsæter

A Method for Detecting Near-Misses from AIS Data

Master's thesis in Marine Technology
Supervisor: Bjørn Egil Asbjørnslett
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU
Norwegian University of
Science and Technology

Preface

This master thesis represents the end of my two year Master of Science degree with specialization in Marine Systems Design and Logistics at the Norwegian University of Science and Technology (NTNU). It has been written during the spring 2019 and corresponds to 30 ECTs.

The thesis proposes a method to identify near-misses from AIS data. All the data used in this thesis has been provided by the Norwegian Coastal Administration.

I would like to express my thanks to my supervisor prof. Bjørn Egil Asbjørnslett for the help trough this project. I would also thank Bjørnar Brende Smestad for his work within simplifying AIS data.

Trondheim, 10.June.2019

Håvard Gjøsæter

Håvard Gjøsæter

Summary

The consequences of maritime accidents can be enormous regarding loss of life, economical loss and environmental harm. Therefore, improvement of safety at sea has been a focus for many years, and various maritime risk assessments have been conducted. To calculate the safety level in maritime traffic it is important with correct historical accidents statistics of collisions between ship. To get an even more thoroughly analysis of the risk in marine areas near-misses should also be investigated in addition to real collision. Underreporting of near-misses is an issue, and a model to detect these has therefore been proposed using Automatic Identification System (AIS) data.

The model uses the concept of ship domain to detect near-misses. This is an area around the ship that the navigators want to keep free from other vessels, and if a ship enter another ship's domain it will be defined as a near-miss. The use of ship domain is a well established method to detect near-misses. However, the domains vary and some domains are only applicable for open waters while other for restricted fairways. In this thesis a new ship domain has been proposed for narrow straits. This is a small domain where the length of the ship combined with speed are the parameters which determine the size of the domain.

The model was applied in Karmsund, a narrow strait with high traffic density located in the western part of Norway. The model was first tested with a AIS database from 2010 to 2015, but due to high time intervals between the messages none near-misses were detected. A new database was acquired with data from August 2017, and in this period 1337 near-misses were detected.

It is concluded that most of the detected near-misses are only close encounters and can not be defined as a near-miss. The proposed ship domain is not applicable for Karmsund since the vessels are forced to sail closely to each other due to the narrow fairway, and for large vessels the size of the domain will cover the whole width of the strait.

Sammendrag

Konsekvensene av maritime ulykker kan være enorme med hensyn til tap av liv, økonomiske tap og miljøskader. Derfor har forbedring av sikkerheten til sjøs vært et fokus i mange år, og det har vært utført ulike maritime risikovurderinger. For å beregne sikkerhetsnivået i den maritime trafikken er det viktig med riktig ulykkesstatistikk for kollisjoner mellom skip. For å få en enda grundigere analyse av risikoen i de ulike marine områdene bør nestenulykker undersøkes i tillegg til ekte kollisjoner. Underrapportering av nestenulykker er en utfordring, og en modell for å oppdage disse har derfor blitt foreslått ved hjelp av data fra automatisk identifiseringssystem (AIS).

Modellen bruker konseptet skipsdomene for å oppdage nestenulykker. Dette er et område rundt skipet som navigatørene ønsker at andre skip ikke kommer inn i. Hvis et skip kommer inn i dette området, vil det bli definert som en nestenulykke. Bruken av skipsdomene er en veletablert metode for å oppdage nestenulykker. Disse domenene varierer, og enkelte domener gjelder kun for åpent farvann mens andre for begrensede områder. I denne oppgaven er det foreslått et nytt skipsdomene for smale farvann. Dette er et lite domene der lengden på skipet kombinert med hastigheten er parameterne som bestemmer størrelsen på domenet.

Modellen ble brukt i Karmsund, et smalt sund med høy trafikktetthet som er lokalisert i Vest-Norge. Modellen ble først testet med en AIS-database fra 2010 til 2015, men på grunn av høye tidsintervaller mellom meldingene ble det ikke oppdaget noen nestenulykker. En ny database ble anskaffet med data fra august 2017, og i denne perioden ble 1337 nestenulykker oppdaget.

Det konkluderes med at de fleste av de oppdagede nestenulykkene kun er passeringer mellom to skip der avstanden har vært kort, og de kan derfor ikke defineres som en nestenulykke. Det foreslåtte skipsdomenet er ikke anvendelig å bruke i Karmsund siden fartøyene er tvunget til å seile tett på hverandre på grunn av den smale farleden, og for store fartøyer vil størrelsen på domenet dekke hele bredden av sundet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Marine transport is the most economical way for transporting goods, and it is crucial for trading in a global perspective. Ships are carrying large quantities, and therefore the potential consequences of a single accident can be enormous regarding financial loss, fatalities and environmental harm. The transportation by sea has grown over the years and the world fleet is getting larger. Due to increasing traffic and the environmental focus the last decades, the safety at sea has become a priority for maritime authorities.

This focus has led to maritime risk assessments and implementing of risk reducing measures (RRM). When it comes to monitoring and analysis of ship traffic, historical data is essential. Therefore, the International Maritime Organization (IMO) introduced Automatic Identification System (AIS). This is a navigation aid to avoid collisions where the ships' properties are broadcasted live between vessels using VHF, and most of the world fleet is obligated to carry an AIS transponder.

AIS data for each vessel is stored in databases, and the data can be utilized for different purposes like monitoring the traffic pattern and identify high risk areas. With an increasing database and more reliable data it will become possible to reveal real historical near-misses using AIS data only. Then a more thorough risk assessment can be performed for different fairways and help authorities identify high risk areas.

IMO's definition of a near-miss is: *"A sequence of events and/or conditions that could have resulted in loss. This loss was prevented only by a fortuitous break in the chain of events and/or conditions. The potential loss could be human injury, environmental damage, or negative business impact (e.g. repair or replacement costs, scheduling delays, contract violations, loss of reputation)"* (IMO, 2008). This definition is part of IMO's guidance where they encourage companies to report near-misses, and they emphasize that there are no negative consequences for reporting and that nobody will be pursued. Reporting and investigation of near-misses are important in order to avoid future similar near-misses or serious accidents, and areas with a high probability of collisions can be identified. Implementing recommendations and risk reducing measures for these areas are essential for reducing the likelihood of collisions. Therefore near-misses should be investigated in line with real accidents.

An issue with near-misses is the lack of visual and physical evidences, and in many cases the only persons familiar with the situation are the crew on the involved vessels who often chose not to report. One of the involved vessel may consider the encounter as a near miss while the other one may not. There are also time consuming procedures for reporting which make it easier to continue the voyage and pretend that nothing happened.

1.2 Objective

How to identify and reveal near-misses and how historical AIS data can be utilizes to detect these are the big questions in this project. The objectives of the thesis are divided in three parts:

- Develop a model to detect near-misses between two vessels.
- Use historical AIS data to reveal near-misses using the proposed model.
- Identify if there are patterns for these encounters.

1.3 Structure of the Thesis

This thesis is divided in six chapters: introduction, theory, method, case study, discussion and conclusion. Chapter 2 consists of the theoretical background which covers the theory behind the developed model. Chapter 3 describes the proposed model and why it has been chosen. It also describes how it works. In chapter 4 the model have been tested in a case study for validation of the model. Here the results of the thesis are presented. In the discussion chapter pros and cons for the used method are discussed in addition to the results. The last chapter is a conclusion and recommendations to further work.

Chapter 2

Theory

A literature review is the foundation of a thesis. Since this paper is about risk in maritime traffic, it is important to get an understanding in the topic and get familiar with previous work and established models. The first part is an introduction to maritime traffic risk and the second part is previous work utilizing AIS data in risk assessment and in near-miss detection.

2.1 Risk Assessment in Maritime traffic

The safety at sea became a high priority in the 1970s, and the first well known work done within maritime traffic risk was performed by Fujii (1974) and Macduff (1974). Their pioneering work has been used as a foundation and been further developed in several studies. Fujii's model was a method to estimate the number of potential accidents in a fairway per time unit, in other words the frequency of accidents (Eq. 2.1). Macduff proposed a ship collision - and grounding model where the probability of a collision/grounding in fairways was conducted (Eq. 2.2). These two models are closely related and applies for traffic in distinct shipping lines.

$$F_{ship-ship} = N_a \times P_C \tag{2.1}$$

$$P = P_a \times P_C \tag{2.2}$$

N_a is the geometrical number of possible accident candidates, i.e. traffic intensity. P_a is the geometrical probability of an accident if no action is taken. Geometric parameters are the shape of the fairway, ship size, speed, course and traffic volume. P_C is the causation probability, which is the probability of a collision given a critical situation or accident scenario, or the probability of losing navigational control. This probability is set to be constant for different water areas, but the probability depends on the type of accident. There are three different types of accident situations in maritime traffic, which are collisions between ships, stranding and grounding. The focus in this thesis will be on collisions between ships, which can be divided into crossing-, head-on- and overtaking collisions. These are illustrated in Figure 2.1 and further described in Section 3.3.

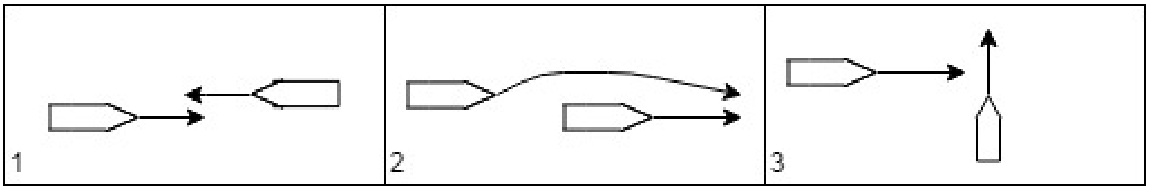


Figure 2.1: Collision scenarios: Head-on(1), Overtaking (2) and crossing collision(3)

A disadvantage in Fujii's and Macduff's models is that the vessels are randomly distributed in the waterways, while in reality they mostly follow specific lines. To avoid this Pedersen et al. (1995) introduced a Gaussian (normal) distribution of the ship traffic in waterways, where it is most likely that the vessels are sailing in the center of the waterway. Figure 2.2 shows a potential crossing collision scenario and illustrates two crossing lines with associated normal distribution, f$_j$, and the risk area. V is the vessel's speed and z is the distance from the centerline to shore.

There are several studies within this maritime traffic risk. Li et al. (2012) have provided a review of 87 various quantitative risk assessment (QRA) models for maritime fairways, and the conclusion was that human error is essential and cannot be excluded in risk analysis. Some examples of risk assessments are (Friis-Hansen, 2008) and COWI (2012). IWRAP MK II is a risk management tool from IALA, a further development of Pedersen's model, which used Bayesian Network to find the causation probability and a software toolbox to evaluating the risk in distinct geographical areas.

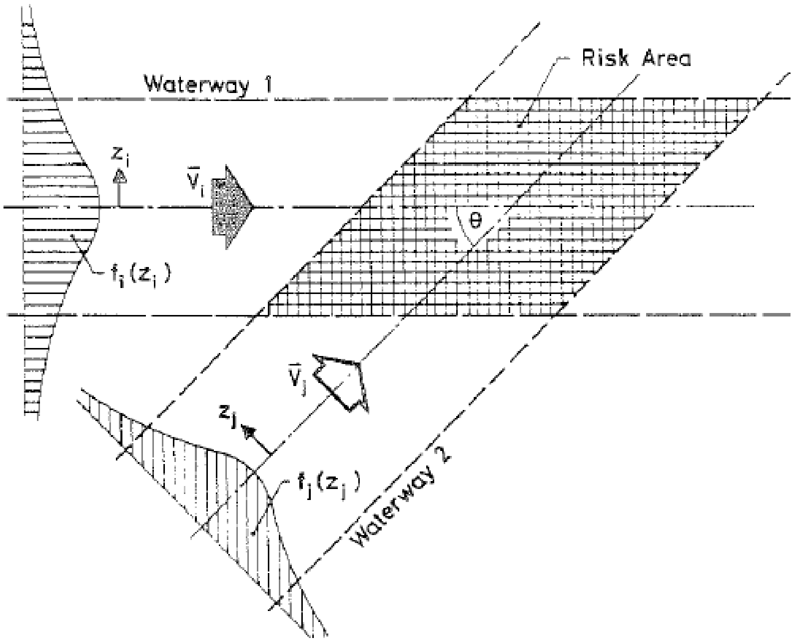


Figure 2.2: Illustration of crossing waterways (Pedersen et al., 1995)

COWI performed a risk analysis in the North Sea and the Baltic Sea, and used Fujii's model as a basis to calculate the frequency of accidents in a more detailed way.

2.2 Previous work within AIS

This section is a preview of some previous papers using AIS data, which is essential to get an insight in how AIS data can be utilized. The chosen papers are within maritime risk, but there are several other areas where AIS data can be used. Before 2010 there were some few studies within the subject, but most of the studies were performed later due to insufficient databases.

As mentioned in the previous section COWI (2012) performed a risk assessment, and they used AIS data to model the density of ship traffic. Figure 2.3 shows the density plot and is an example of how AIS data can be visualized. The red lines indicates high density. Silveira et al. (2013) performed a ship collision risk analysis off the coast of Portugal where available AIS data was used to calculate the expected number of collisions in a traffic separation schemes (TSS) based on Equation 2.1. The result was used to confirm that the causation probability, P_C, was reliable and that AIS data could be used to identify collision candidates. TSS is a risk reducing measure introduced by IMO which are dedicated lines that ships are obligated to follow to avoid head-on collisions, but the risk of overtaking collisions are still present. These are common in areas with high traffic density. In Figure 2.3 there are parallel lines from Russia to Germany which illustrate the TSS.

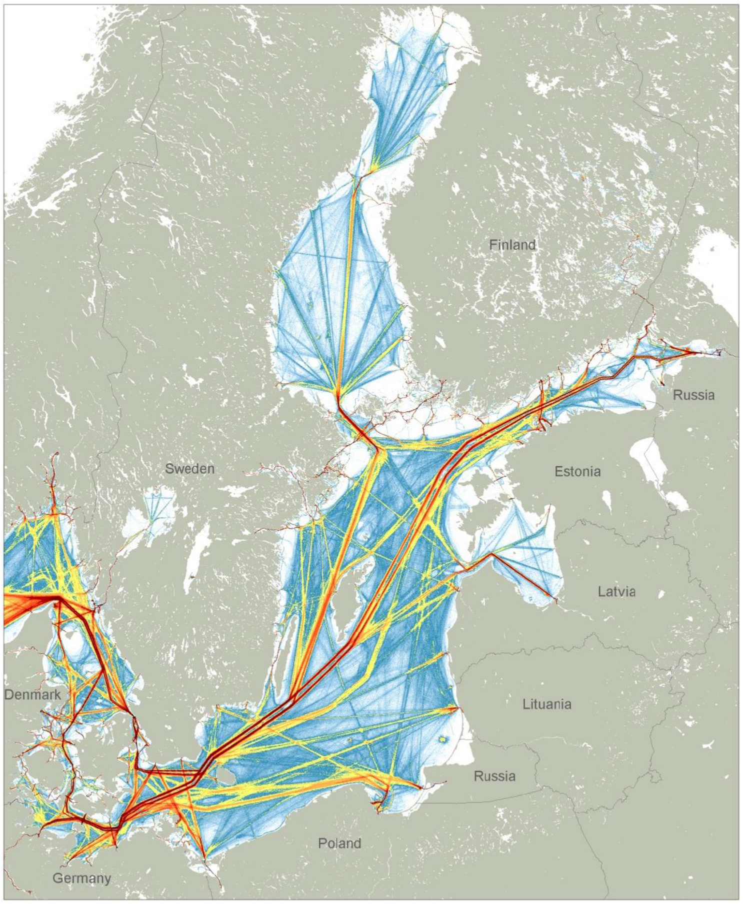


Figure 2.3: Traffic density plot for the Baltic Sea (COWI, 2012)

Montewka et al. (2011) did an assessment in the Gulf of Finland and used the minimum distance to collision (MDTC) to find the probability of ship collisions. AIS data was used to find collision candidates, but the calculated number of candidates did not comply with the observed values. The causation probability used in the study was not appropriate with the chosen model. Aarsæther and Moan (2009) demonstrated the application of AIS data to estimate navigation patterns. The study demonstrated how computer vision techniques could be applied to get statistics and grouping of ship traffic from AIS data. Eide et al. (2007b) developed a model which made it possible for vessel traffic service (VTS) centers to identify and monitor high risk oil tankers, including weather which made the risk level dynamic. This model was implemented by Vardø VTS, and allowed VTS operators to monitor individual tankers with a high risk level. Eide et al. (2007a) did also develop a dynamic model, including wind, current and wave for the North Norwegian coast which made it possible to identify, not only high risk vessels, but also high risk areas. The risk became a function of location and time, and the model was used as an aid to position tug vessels.

2.3 Near-Miss Detection

Near accidents are considerably more common than actual accidents and can give a more accurate statistically evaluation of maritime safety. Therefore, a need to identify these near-misses is present, and with AIS this may be possible. AIS makes it possible to detect close encounters, but the question is which of these encounters are near-misses. There have been preformed some studies within this topic, but there are no recognized methods to reveal near-misses. A common way to detect them using AIS is to use the so called ship domain, an area around the ship that can't be occupied by other vessels. The issue with the ship domain is that there are a lot of different domains, and they differ in the different studies. The shapes of the domains are circles, ellipses, sector-domains or fuzzy domains. In this section some of these studies are compared and discussed, and it makes a foundation for the rest of this thesis regarding to define a near-miss.

The concept of ship domain was introduced by Fujii in 1971, illustrated in Figure 2.4. The shape of the domain is an ellipse where the half-width (a) is the length times 1.6 and the half-length (b) the length times 4.0 taken from the center of the ship. Goodwin (1975) defined a ship domain as: *"A ship domain may be thought of as the sea around his ship which the navigator would like to keep free, with respect to other ships and fixed objects."*.

Szlapczynski and Szlapczynska (2017) discussed different ship domains and made a thorough review. The domains were divided into three section: domains developed by theoretical analysis,

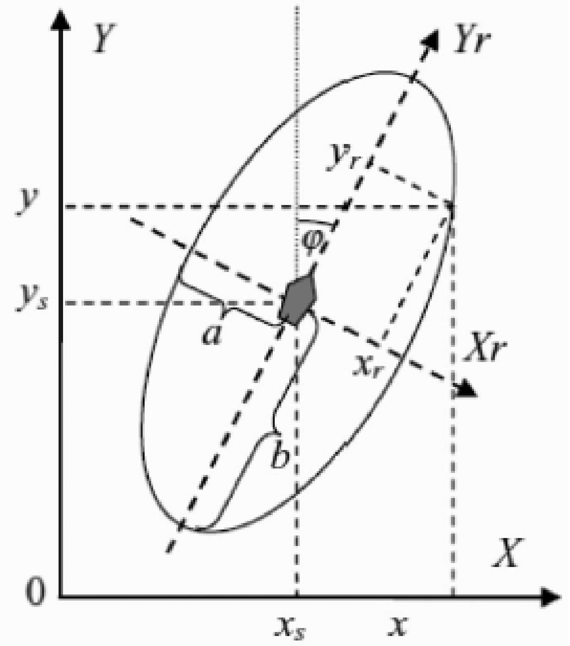


Figure 2.4: Fujii's ship domain. (Goerlandt et al., 2012)

domains based on expert's knowledge and domains determined empirically, but these have also been combined. Empirical domains are simple and are suitable for determine capacity of fairways, but not for collisions. A combination of expert judgement and analytical approach is preferred in the detection of near-misses. The empirical domain depends on the fairway or water area and is determined by data-based methods. This means that in a narrow waterway the domains will become smaller and the percentage of collision types will be different. Most of the domains are applicable for all areas, but there are some domains dedicated to only open waters or restricted waters. Szlapczynski and Szlapczynska (2017) concluded that ship domains are a success when it comes to near-miss detection, but not collision avoidance due to complexity. When using ship domains, there are four different ways of defining an encounter which are listed below and illustrated in Figure 2.5:

- a) Own ship's (OS) domain should not be violated by a target ship (TS)
- b) A target ship's (TS) domain should not be violated by the own ship.
- c) neither of the ship domains should be violated (a conjunction of the first two conditions)

- d) ship domains should not overlap - their areas should remain mutually exclusive (the effective spacing will be a sum of spacing resulting from each domain).

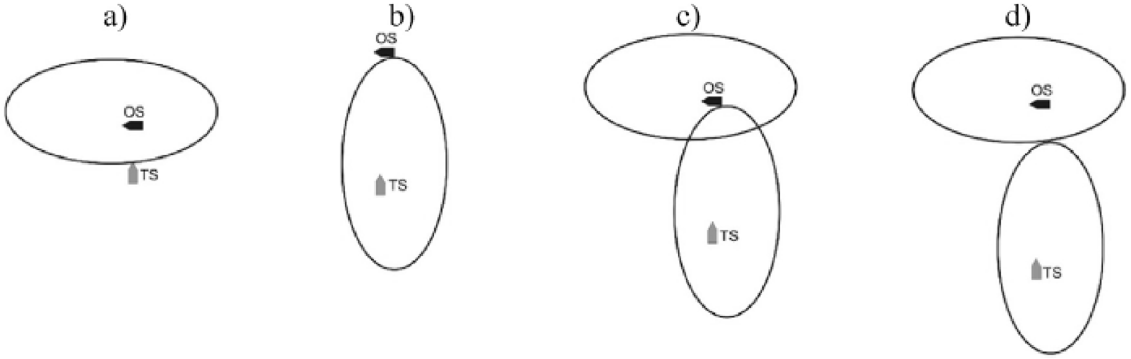


Figure 2.5: Different domain-based safety criteria (Szlapczynski and Szlapczynska, 2017)

Goerlandt et al. (2012) initiated a study of near collisions in the Gulf of Finland. This is one of several studies of near-misses performed at Aalto University in Finland. In the study AIS data were utilized together with Fujii's ship domain to identify and evaluate near-misses for crossing-, overtaking- and head-on collisions. When a ship entered another ship's domain it was defined as a near-miss. This is illustrated in Figure 2.6. In the figure both vessels have violated the domains, but to be classified as a near-miss it was sufficient that only one vessel had violated the domain. This conform with c) in Figure 2.5. To find these encounters they developed an algorithm which could trace the vessels and find the overlapping trajectories at the same time in the same location, and then save the two involved vessels' data like the speed and course when the situation occurred, but also type of vessels and flag state. These data could be used to find statistics of which vessels and flag states that have the highest contribution to near accidents.

Zhang et al. (2015) and Zhang et al. (2016) did also perform a study of near-misses in the Gulf of Finland. They used a method that ranked encounters from AIS data using a Vessel Conflict Ranking Operator (VCRO) which filters the data and the highest ranked encounters were evaluated by experts to determine if they were near-misses or not. The experts based their conclusion on the distance between the involved ships, the relative speed between them and the difference in heading. Zhang et al. (2016) used the ship domain to detect the encounters which were further analysed. Also here Fujii's domain was used. They also added the Minimum Distance to Collision

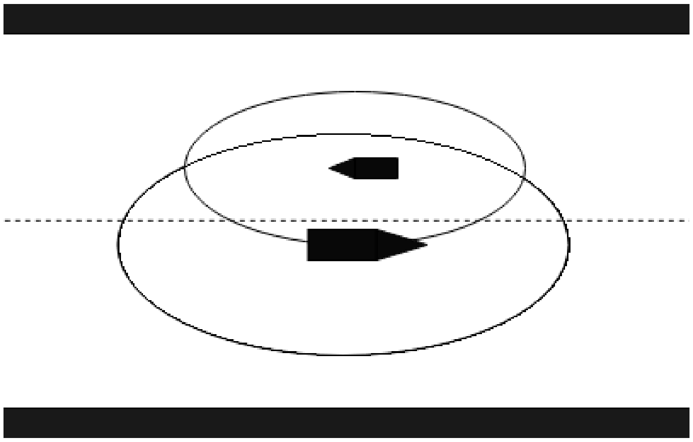


Figure 2.6: Domain violation

(MDTC) which made it easier to separate the high risk encounters from the lesser ones.

Mestl et al. (2016) demonstrated how the vessels' trajectories from historical AIS data can be used in a collision investigation. They performed a case study of a real collision in Norway between a ferry and a fishing vessel to demonstrate the use of rate of turn (ROT), see Figure 2.7. High resolution AIS data (2-12 second sampling time) was utilized to identify evasive manoeuvres and high ROT values. They considered ROT as the most valuable AIS parameter. A high ROT value is a non-normal manoeuvring and may indicate that the navigators are trying to avoid an obstacle or a ship right before the collision or near collision in the hope to avoid it. In Figure 2.7 the red line in the ferry's trajectories shows a high ROT and it is an indication that the ferry tried an evasive manoeuvre just before the collision. The fishing vessel followed a straight line which may indicate that they did not observe the ferry.

The comfort limits for cruise vessels driving in 20 kn is 10 deg/min in ROT. This value varies between the type of vessels. Historical AIS data can be used to reveal high ROT values, and then be further analysed to find the causes for these high values. If there have been other vessels at the same location, it might have been a near-miss. In heavy sea the ROT will be naturally higher than in calm sea. In addition to ROT, there is centripetal acceleration (CA) which is associated to ROT and is connected to the passenger comfort. A drawback using ROT to reveal near-misses is that a lot of vessels do not have an AIS transponder and it is not mandatory for vessels below 50 000 GT to carry a ROT indicator. Only 5 % of the world fleet (2016) are properly logging ROT. In addition,

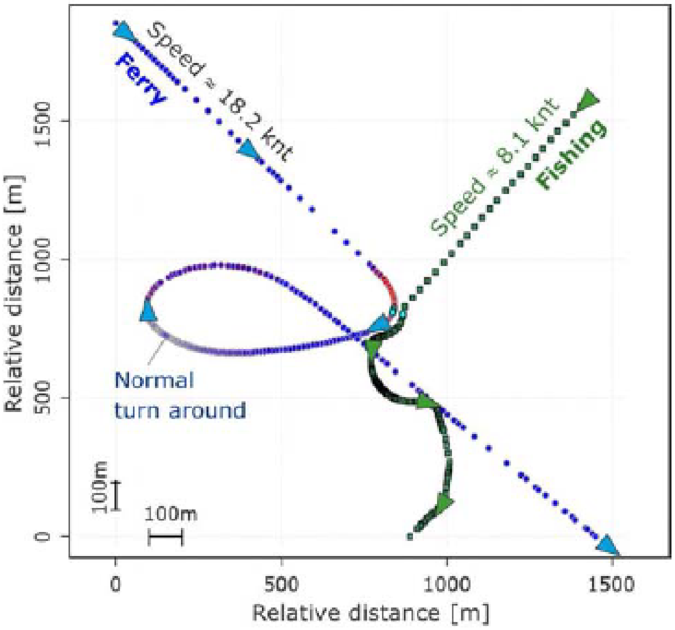


Figure 2.7: Visualization of trajectories from high resolution AIS data

there are data quality issues. It is also very hard to see large ROT values for large vessels due to their moment of inertia (Mestl et al., 2016).

van Westrenen and Ellerbroek (2017) used AIS data to analyse the traffic in the Netherlands. The objective was to investigate local conflict complexity as a contributor to collision risk. The focus was how near collisions were affected by the complexity of the local traffic. To find the safe space around respective vessels, conflict resolution constraints were determined based on extrapolation of the current traffic. Ship domain (Fujii) was used to represent the uncertainty and not the risk. Since the domain only represent the uncertainty, a simple domain would suffice. There were two types of near-misses in the study: A complex situation with a small resolution space, and the second with only two ships who intentionally minimize the passing distance due to economic matters. The risk of a collision was high when the ratio between acceptable and unacceptable solutions within this space was low. The complexity increased when the resolution space decreased. Resolution space was the safe space where conflicts had been avoided.

Iperen (2015), at the Maritime Research Institute Netherlands (MARIN), has developed a method to classify crossing, head-on and overtaking encounters in the North Sea from AIS Data. The paper discusses how to classify encounters and distinguish between normal, exceptional encounters and near-misses. He defined an encounter as: ”*An encounter is defined as the tracks of two ships having a speed of at least 1 knot, that at certain moments during their approach, are expected to pass each other within 3 nautical miles within 20 minutes, based on their speed and course*”. The method uses Distance at Closest Point of Approach (DCPA) and Time to Closest Point of Approach (TCPA), in addition to ship domain, to find encounters with abnormal patterns. The domain was determined empirically, and it confirmed that an elliptical form were applicable for head-on and overtaking collisions.

Wang (2010) proposed a new ship domain called quaternion ship domain (QSD) and a fuzzy QSD (FQSD) with four radii/lengths: fore, aft, starboard and port. These lengths were determined by several factors, like speed, course and manoeuvring capability. The shape of the domain can be a quadrangle or a combined ellipse, see Figure 2.8. R_{fore} and R_{aft} are the longitudinal radius while R_{port} and R_{starb} are the lateral radius. These domains are not symmetrical. The fore and starboard radii are longer than the aft and port radii respectively. This is due to COLREGs regulations which is described in section 3.3. The FQSD, where uncertainty and fuzzy information have been added, is more practical and convenient for navigators. Since these lengths are a function of several factors, the size of the domain will vary. For example, when the speed is 15 knots, the domain will be similar to Fujii's domain in the fore and port side while it will differ in aft and starboard.

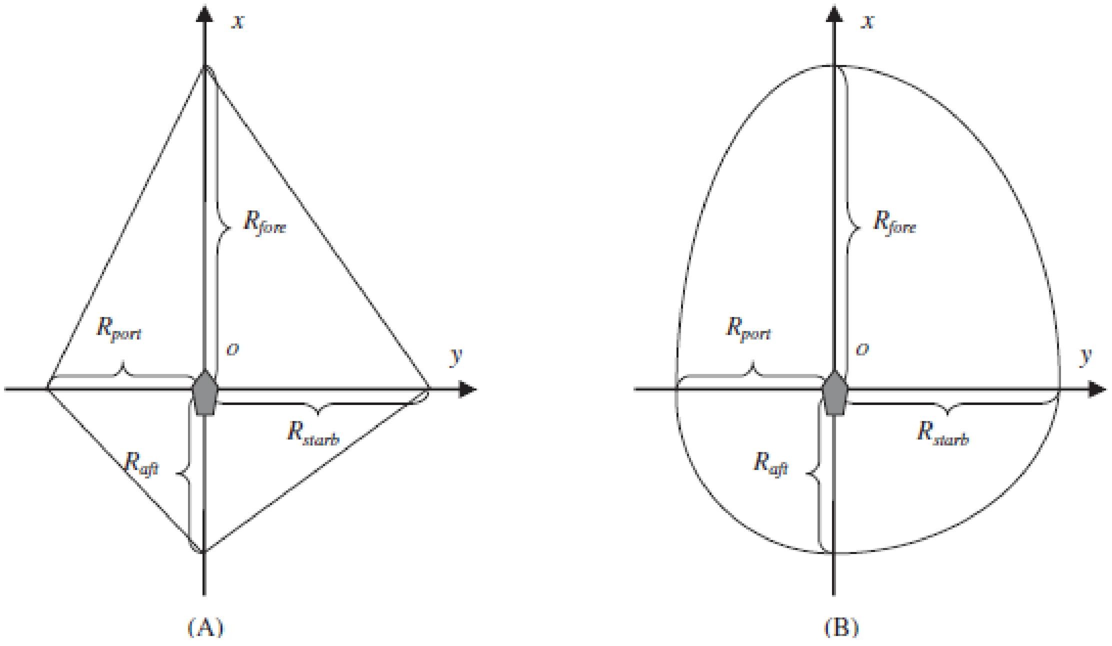


Figure 2.8: A. Quadrangle domain. B. Elliptical domain. (Wang, 2010)

Chapter 3

Method

3.1 Near-Miss Detection Algorithm

The algorithm to detect near-misses in this thesis is shown in Figure 3.1. The proposed method is based on the algorithm used by Goerlandt et al. (2012) and Nordkvist (2018). The idea is to detect events in the database where a ship has violated the domain of the other ship, which will be classified as a near-miss. Nordkvist (2018) did not classify these encounters as near-misses, but did further analyses to find high risk encounters. The main different in the two methods is that Goerlandt et al. (2012) look at concurrent trajectories with an interval of five minutes, while in the Nordkvist (2018) and in proposed method all time steps are independently treated.

The algorithm starts with evaluating if the vessels' trajectories occur in an overlapping timeframe. The next step is to find the closest distance between the vessels. The event is defined as a near-miss if the distance between the vessels is less than the length to the boundary of own domain. This is illustrated in Figure 2.6 and the proposed domain is described in Section 3.4. If the domain is violated, the information like time of occurrence, the vessels ID and dimensions, the distance between them, the speed, the course and the position (long and lat) will be stored in an own table for further analysis. The algorithm is coded in Python 3.7. The proposed method has been applied in a case study in Section 4. This method is only applicable for near collision between two vessels and not grounding or stranding.

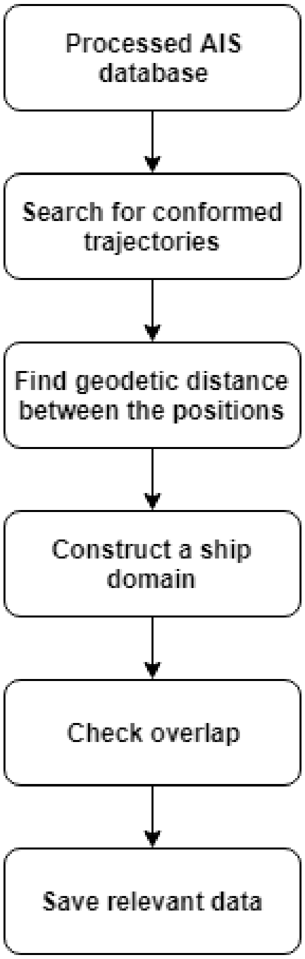


Figure 3.1: Algorithm to reveal near-misses

3.2 AIS Data

This section is an introduction and description of AIS data. The first part is an introduction of AIS while the second part is a description of how to proceed from raw AIS data to a useful database for a chosen problem. The data used in the case study in this thesis will also be described in this section.

3.2.1 Introduction to AIS

Automatic identification system (AIS) is a tracking and communication system used in maritime traffic as a navigation tool to improve the safety at sea. The ships are carrying a transponder where the data are broadcasted using the very high frequency (VHF) system. The purpose is to improve the safety of navigation and protect the environment. To to this the AIS is used in three different ways:

- In a ship-ship mode to avoid collisions.
- To help vessel traffic service (VTS) centres to control and manage the traffic.
- To provide the authorities with information about the cargo and vessels, which makes it possible to monitor vessels with hazardous cargo.

The International Maritime Organization (IMO) introduced AIS in the early 2000. It was implemented in the International Convention for the Safety of Life at Sea (SOLAS) in 2002 with the general requirement that all vessels above 300 gross tonnage in international waters, cargo ships above 500 gross tonnage not engaged on international voyages and all passenger ships irrespective of size are obligated to carry a AIS transponder. Table 3.1 shows a more detailed description for which vessels that are required to carry an AIS transponder. Even though it was introduced in 2002 it was not fully implemented until July 1. 2007. There are two classes of AIS transponders, A and B. The vessels in table 3.1 require to have class A while class B are voluntary and are often used by smaller fishing vessels and pleasure crafts (DNV-GL, 2014).

The VTS centres have a limited range of 40-60 nautical miles which is the range of VHF. In Norway there are 60 VTS centers and the AIS network was first established by the Norwegian Coastal Administration (NCA) in 2005 (Kleppe, 2016). To receive messages outside the range and for extending the coverage satellite AIS (S-AIS) are utilized. The satellites provide coverage in areas without signals and far off the coast. There are four AIS satellites in Norway, owned by the NCA. The first one, AISSat-1, was launched in 2011, AISSat-2 in 2014 and the two last ones, NorSat-1 and NorSat-2 were launched in 2017. With the two newest satellites the NCA could detect 60 % more vessels. The data from the satellites is sent directly to Vardø VTS which is responsible for the ship traffic and rescue centers along the Norwegian Coast. Since the data is sent directly, the rescue missions can be faster and more coordinated (Kleppe, 2017).

Vessel type	Requirenments
Tankers	All in international voyage All in voyage inside EU/EEA
Passenger vessels	All in international voyage Above 300 GT in voyage inside EU/EEA High speed vessels above 150 GT in national voyage
Cargo ships	Above 300 GT in international voayage Above 300 GT in voayge inside EU/EEA
Fishing vessels	Above 300 GT or 45 meters inside EU/EEA

Table 3.1: IMO's requirements for AIS transponders

There are 27 different types of messages in the AIS data defined by the International Telecommunication (ITU) (Series, 2010). In this thesis there will be a focus on message type 1, position report, and message type 5, static vessel and voyage related data. 72.5 % of all AIS data is from message type 1 (Smestad, 2015). Each message type consists of different information, and the information for massage type 1 and 5 are shown in Table 3.2 and Table 3.3 respectively.

Message Type 1	Description
Unixtime	Seconds since 01.01.1970
MMSI number	Maritime Mobile Service Identity (Vessel ID)
Position	Coordinates in latitudes and longitudes
Course over ground (COG)	Current course
Speed over ground (SOG)	Current speed
Rate of turn (ROT)	Change of course
Status	Navigational status

Table 3.2: Information in Message type 1

Message Type 5	**Description**
Unixtime	Seconds since 01.01.1970
MMSI number	Maritime Mobile Service Identity (Vessel ID)
Destination	Destination of current voyage
Dimension	Length and width
Draught	Current draught in meter
ETA	Estimated time of arrival in unixtime
IMO number	International Maritime Organization number
Name	Name of the vessel

Table 3.3: Information in Message type 5

Message 1 consists of dynamic data, except the user ID. Each vessel has a unique IMO number and Mobile Service Identity (MMSI) for identification. The dynamic data are sent every 2-10 seconds and consists of position, course and speed. The time interval for sending data depends on the speed, and at anchor the data is sent every three minutes, see Table 3.4. The static data is recorded every sixth minutes unless there are any changes. Then the data should be sent immediately, but since they are changed manually errors may occur (IALA, 2016).

Vessel status	**General reporting interval**
Vessel at anchor	3 min
Vessel at 0-14 knots	12 sec
Vessel at 0-14 knots changing course	4 sec
Vessel at 14-23 knots	6 sec
Vessel at 14-23 knots changing course	2 sec
Vessel $>$ 23 knots	3 sec
Vessel $>$ 23 knots changing course	2 sec

Table 3.4: Reporting intervalls for dynamic data

3.2.2 Data Quality

Together with the radar the vessel tracking and the positional accuracy can be improved, the manoeuvring data becomes near real time and the AIS gives information in radar shadow area like behind islands and bends. Even though it is a great aid, there are also some limitations: smaller vessels are not obligated to carry a transponder, there are areas without signals and the transponders don't have sufficient power to send the signals. There are also human error since the AIS can be turned off and the manually data like draught, destination and estimated time of arrival (ETA) can be incorrect. Since there are several limitations with AIS, it should be used together with the radar - which is still the main instrument to avoid collision - and other navigation aids (IALA, 2016).

The quality of AIS data has been discussed and covered in previous master thesis. Some examples are Smestad (2015) and Næss (2018). According to Smestad (2015) there are several errors in S-AIS data: several thousands vessels have erroneous data like incorrect MMSI numbers or the dimensions of the vessels are incorrect. He also discovered that 4.95 % of the world fleet was missing from the AIS data. The satellites can also have problems with interference due to a high amount of transponders in a small geographic area. Low orbiting rates over the areas will contribute to gaps in the data set. Leonhardsen (2017) discovered that there are more MMSI numbers in the database than vessels in the world fleet.

3.2.3 Data Handling

The AIS data used in this thesis was provided by the Norwegian Coastal Administration in 2015. The raw S-AIS data in the period 2011-2015 was utilized by Smestad (2015) and Leonhardsen (2017) in their master thesis. Smestad (2015) developed an algorithm in Python to decode the raw AIS messages into readable information and the data was extracted to an SQLite database.

All the data handling and visualization in this thesis have been done by using Python 3.7 which is compatible with SQLite. SQLite is a database motor used to extract the data, and the DB browser for SQLite is used to visualize the database. The table with the identified near-misses is saved as a comma separated values (CSV) file which can be imported and easily visualized in the DB browser. CSV files have been chosen instead of writing directly to the database due to errors writing to a single database from several processes (Nordkvist, 2018). Basemap at matplotlib is the library used to plot the findings, while the Numpy and Pandas libraries are used to work with and manipulate data.

3.3 COLREGs Definitions of Encounters

Convention of the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs) are rules developed by IMO for navigating at sea and replaced the Collision Regulations of 1960. Rules 11-18 in COLREGs describe how vessels in sight of one another should act, and rule 13-15 describe overtaking, head-on and crossing situations (ecolregs, 2018). Due to complexity only rule 13-15 are applied in this thesis. These are listed below:

- Rule 13: An overtaking situation occur when a vessel is coming up with another vessel from a direction more than 22.5° abaft her stern, and the overtaking vessels are obligated to give way. 22.5 degrees are due to the masthead lights shine from straight ahead to a point 22.5 degrees abaft.
- Rule 14: In a head-on situation with two power-driven vessels meeting on reciprocal courses where a risk of collision is present each vessel shall alter the course to starboard. If the vessel is able to see both the sidelights (green and red) together with the masthead of the meeting ship, it will be defined as a head-on situation.
- Rule 15: A crossing situation is defined when to power-driven vessels are crossing and there is a risk of collision involved. The vessel which has the other ship on starboard side shall keep out of the way and try to avoid crossing ahead of the other ship.

Iperen (2015), mentioned in Section 2.3, used relative heading, φ, to distinguish the types of encounters, see Equation 3.1 where COG_1 and COG_2 are the courses of the two vessels. Using max/min values between the courses gives positive results where the course of the vessels are in the range of 0-360°. Table 3.5 specifies in what range the different encounters are defined. These values will be used in the code to determine the type of encounter. The relative bearing, α, is the direction of vessel B seen from A, defined as the angle of the line between the center of the two ships and the course line of A (COG), see Equation 3.15. The angle is measured clockwise from the COG from own ship (A). Figure 2.5 illustrates the different types of encounters and for which headings that separates them. Vessel A is own ship, and in this situation ship B1 and C should give way to A. Vessel B1 should give way since A is located on starboard side of vessel B1, while C should give way due to overtaking. Vessel A should give way to B2. For vessel D both A and D should alter the course to starboard. The head-on sector has been given a value of 30 degrees which is the same as Iperen (2015), while the overtaking value is chosen with respect to Rule 13 in COLREGs.

$$\varphi = max(COG_1, COG_2) - min(COG_1, COG_2) \tag{3.1}$$

Type of encounter	**Section**
Head-on	$015° \geq \varphi \geq 345°$
Overtaking	$112.5° \leq \varphi \leq 247.5°$
Crossing, stand on vessel Crossing, give way vessel	$015° < \varphi < 112.5°$ $247.5° \leq \varphi \leq 345°$

Table 3.5: Defining encounters

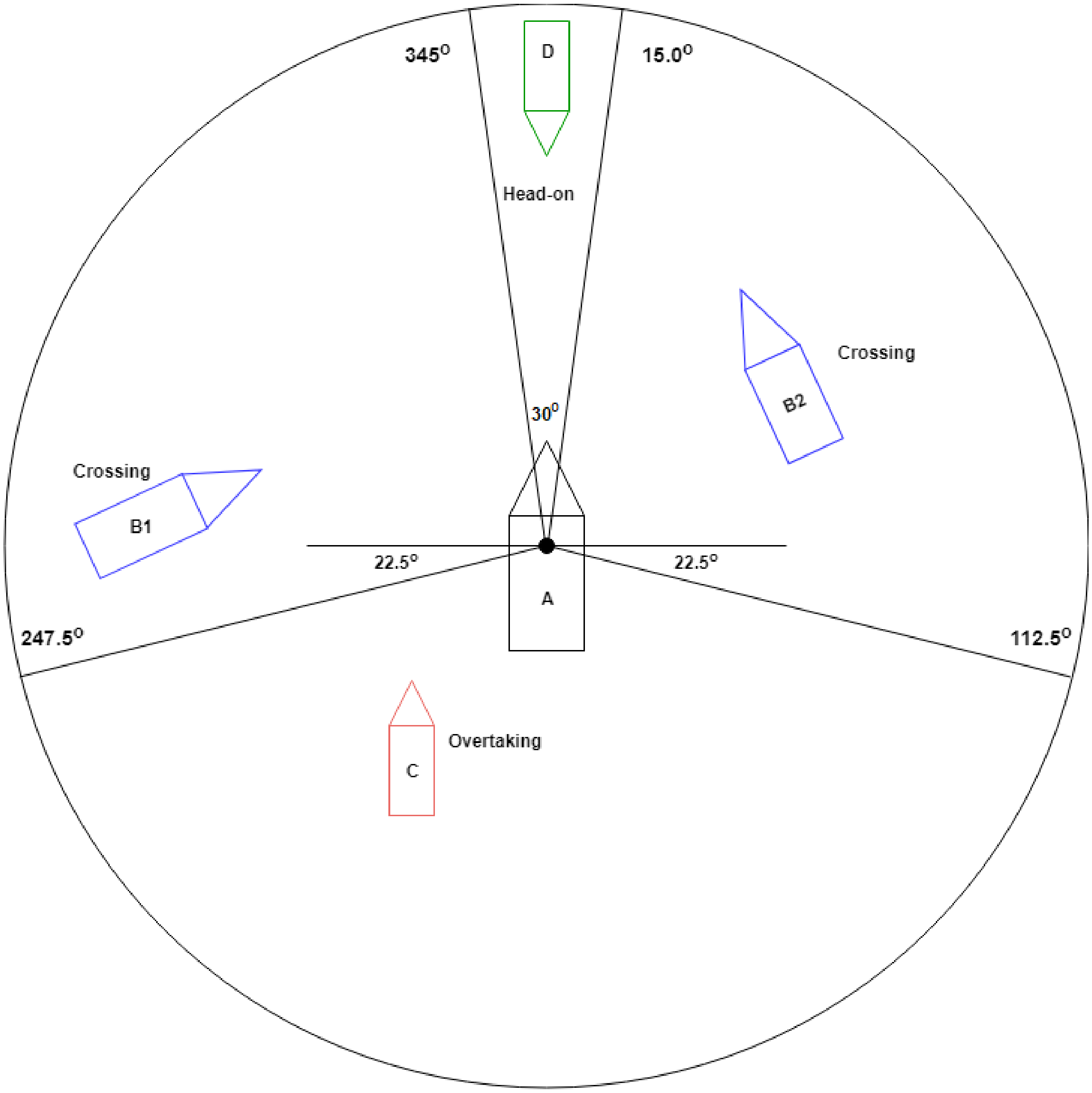


Figure 3.2: Division of encounter sections

3.4 Ship Domain

As explained in Section 2.3 most previous work within the topic use ship domain to identify near-misses from AIS data. Ship domain is also used in this thesis to reveal near-misses. The proposed method is for narrow fairways and it is therefore most appropriate to use a narrow domain. The model will also be applicable for TSS where overtaking encounters are going to dominate while head-on situation should not occur and crossing encounters are rare. COLREGs allows overtaking encounters on both port and starboard side in TSS and therefore it is sufficient to use a symmetrical domain (Szlapczynski and Szlapczynska, 2017).

The chosen domain is a variant of the elliptical quaternion ship domain (QSD) developed by Wang (2010), see Figure 2.8B. The domain is a crisp and not a fuzzy domain, and it will be a function of the ship's length, speed and manoeuvrability. It is reasonable to chose a domain where speed is a parameter since the probability of a collision increases with higher speed. The shape of the domain used by Wang (2010) is described in Equation 3.2, which is the function of the domain's boundary. This is also the same domain that Nordkvist (2018) used in his master thesis to detect encounters in open waters. The QSD is too wide for narrow straits and it has therefore been reduced. The domain will also be symmetrical since it is allowed to pass on both sides in TSS. This means that the lengths R_{port} and R_{starb} are equal and they are smaller than in the original QSD. The aft length, R_{aft}, has also been reduced since it is not possible to collide into a ship located behind you when moving forward.

$$\begin{aligned} f_{ce}(x,y,Q) = &\left(\frac{2x}{(1+sgnx)R_{fore}-(1-sgnx)R_{aft}}\right)^2 \\ &+\left(\frac{2y}{(1+sgny)R_{starb}-(1-sgny)R_{port}}\right)^2 \end{aligned} \tag{3.2}$$

Here Q is the quaternion (i.e. R_{fore}, R_{aft}, R_{port} and R_{starb}) which determines the size of the domain. The sign function sgn(.) is defined in Equation 3.3, and Equation 3.4 shows how the radii from the original QSD are calculated:

$$sgn(x) = \begin{cases} \text{1, x} \leq 0 \\ \text{-1, x} < 0 \end{cases} \tag{3.3}$$

$$\begin{cases} R_{fore} = (1 + 1.34\sqrt{k_{AD}^2 + (k_{DT}/2)^2})L \\ R_{aft} = (1 + 0.67\sqrt{k_{AD}^2 + (k_{DT}/2)^2})L \\ R_{starb} = (0.2 + k_{DT})L \\ R_{port} = (0.2 + 0.75k_{DT})L \end{cases} \tag{3.4}$$

Here L is the length of own ship while k_{AD} and k_{DT} represent the maneuverability, given in Equation 3.5. k_{AD} is the advance and k_{DT} is the tactical diameter while V_{own} is the speed of own ship in knots. As seen from the equation the domain only depends on the length and speed. It should be mentioned that Szlapczynski and Szlapczynska (2017) pointed out that using maneuverability in defining shape and size of the domain has not been sufficiently documented, and Equation 3.5 can therefore be questioned.

$$\begin{aligned} k_{AD} = A_D/L = 10^{0.3591*lg*V_{own}+0.0952} \\ k_{DT} = D_T/L = 10^{0.5441*lg*V_{own}-0.0795} \end{aligned} \tag{3.5}$$

The proposed radii in the new domain are shown in Equation 3.6, where R_{fore} is the same while R_{aft}, R_{port} and R_{starb} have been reduced. The new domain is illustrated in Figure 3.3 and is a function of the four radii. It is a composition of two ellipses, one in the fore end and one in the aft end with R_{fore} and R_{aft} as the longitudinal radii respectively. Figure 3.4 illustrates the domain for different speeds and sizes. In a) the length of the ship is 150 meters while the speed varies between 5 and 20 knots. These domains are compared with Fujii's domain (dotted line) and they are quite similar in the front and the sides, but they differ in the aft end since Fujii's domain is symmetrical around the axes. The breadth of the Fujii's domain corresponds with a speed of 20 knots in the new domain. In b) the speed is 10 knots and the length varies between 100 and 200 meters. Here there are larger differences, which means that the domain deeps more on the length than the speed.

$$\begin{cases} R_{fore} = (1 + 1.34\sqrt{k_{AD}^2 + (k_{DT}/2)^2})L \\ R_{aft} = 0.5R_{fore} \\ R_{port} = R_{starb} = (0.375k_{DT})L \end{cases} \tag{3.6}$$

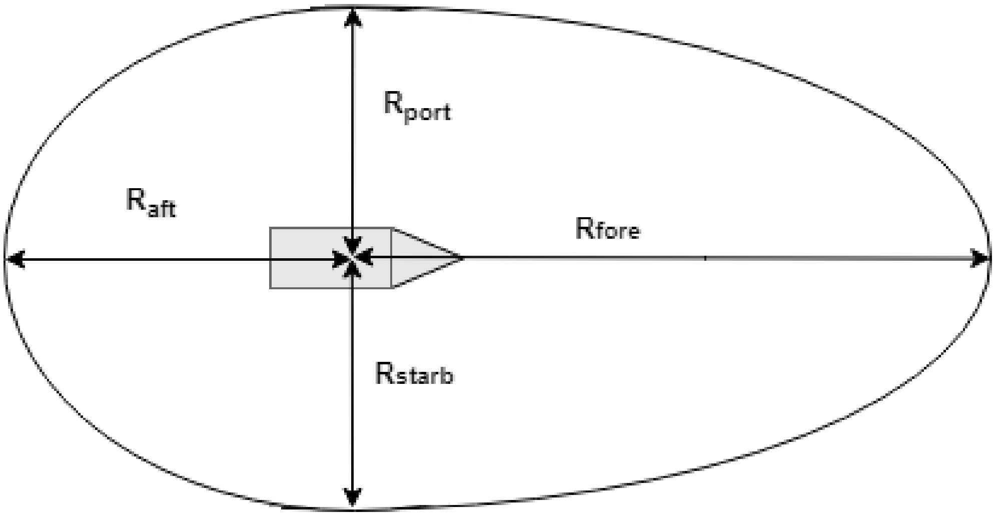
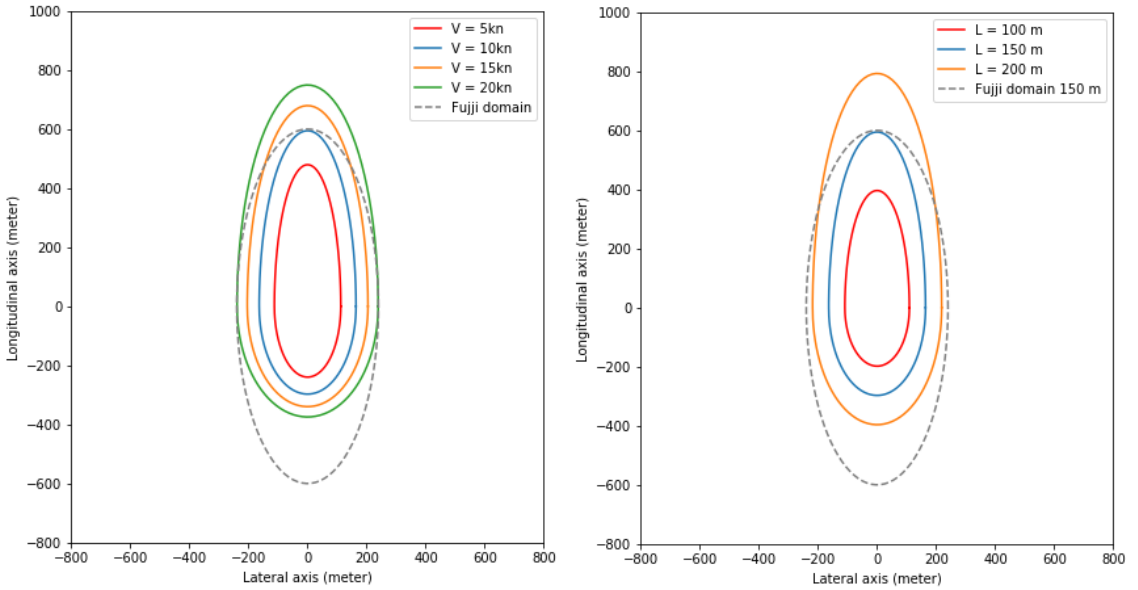


Figure 3.3: The proposed domain is shaped like an ellipse, and Equation 3.6 defines the four radii.



a) b)

Figure 3.4: QSD for different speeds (a) and sizes (b) compared to Fujii's. In figure a) the domain is a function of speed with a constant length of 150 m, while in b) the domain is a function of length with a speed of 12 knots.

A near-miss is detected when the distance between the two involved vessels is less than the radius of the largest of the two domains. This corresponds with definition c in Figure 2.5 from Szlapczynski and Szlapczynska (2017) where neither of the ship domains should be violated. The distance is taken from the center of the ship. Therefore, it is essential to find the lengths to any points along the boundary. The formula for finding the length from center to boundary in an ellipse is given in Equation 3.7 where S is the lateral radius and R the longitudinal radius.

$$L = \left(\frac{1 + tan^2\alpha}{\frac{1}{s^2} + \frac{tan^2\alpha}{R^2}} \right)^{1/2} \tag{3.7}$$

Zhang et al. (2016) use this equation to find the length to the boundary of the domain where the angle α is the relative bearing between the vessels. In the new domain S and R will be switched since Zhang et al. (2016) have a different definition of α. The distance from the ship center to the boundary, L_α, is calculated from Equation 3.8.The relative bearing is illustrated in Figure 3.5.

$$l_\alpha = \begin{cases} \left(\frac{1+tan^2\alpha}{\frac{1}{R^2_{fore}}+\frac{tan^2\alpha}{R^2_{starb}}} \right)^{1/2} & if\ \frac{3}{2}\,\pi < \alpha \leq \frac{\pi}{2} \\ \left(\frac{1+tan^2\alpha}{\frac{1}{R^2_{aft}}+\frac{tan^2\alpha}{R^2_{starb}}} \right)^{1/2} & if\ \frac{\pi}{2} < \alpha \leq \frac{3}{2}\,\pi \end{cases} \tag{3.8}$$

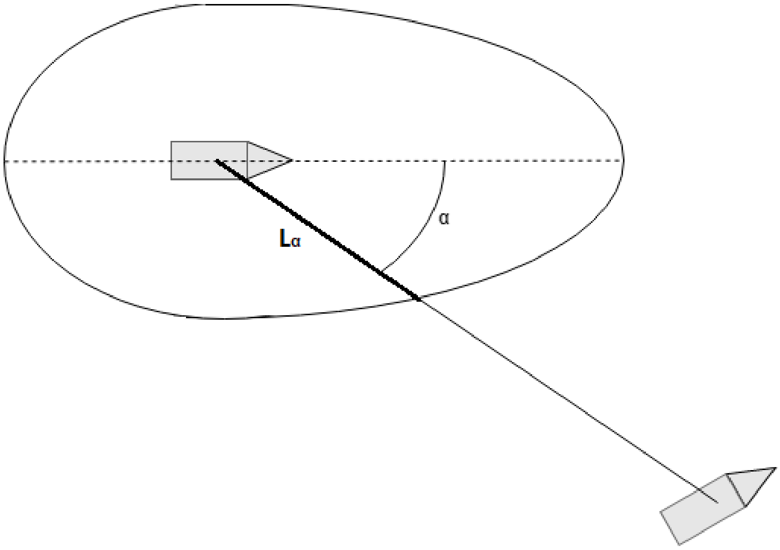


Figure 3.5: The distance between the ship and boundary, L_α, where α is the relative bearing

3.5 Finding True Center

The AIS transponders are not necessarily located in the center of the ships, and it is therefore essential to find the true center in order to place the domain correctly with respect to the ship. The method to find true center is the same method used by Nordkvist (2018). Equation 3.9 shows how to find the true position of the ship's center, where the parameters are the coordinates of the AIS transponder, azi1 and the distance s12. This is done by solving a geodesic problem which is to calculate distances and angles on the surface of the earth. To find the distances and angles on an ellipsoid in Python the package Geographiclib is used, where the angles are in the range of 0-360°. Geographiclib is compatible with the WGS84 coordinate reference system which is the same coordinate system used by Global Positioning System (GPS) (Geography, 2018). Azi1 is the azimuth angle measured in degrees, and is the clockwise angle from 0° North, see Equation 3.10. β is the clockwise angle from the transponder to the center of the ship standing on the position of the transponder and looking straight ahead. s12 is the distance from the AIS transponder to the bow, aft, port and starboard perpendiculars, found by Equation 3.12-3.14.

$$TrueCenter = Direct.WGS84.Dierct(Latitude, Longitude, azi1, s12) \tag{3.9}$$

$$azi1 = COG + \beta \tag{3.10}$$

$$\beta = \begin{cases} 0, & \text{if } d_{bow} > d_{aft} \wedge d_{starb} = d_{port}. \\ 180, & \text{if } d_{bow} < d_{aft} \wedge d_{starb} = d_{port}. \\ 90, & \text{if } d_{bow} = d_{aft} \wedge d_{starb} > d_{port}. \\ -90, & \text{if } d_{bow} = d_{aft} \wedge d_{starb} < d_{port}. \\ arctan\frac{dist_y}{dist_x}, & \text{if } d_{bow} > d_{aft} \wedge d_{starb} > d_{port}. \\ 360 - arctan\frac{dist_y}{dist_x}, & \text{if } d_{bow} > d_{aft} \wedge d_{starb} < d_{port}. \\ 180 + arctan\frac{dist_y}{dist_x}, & \text{if } d_{bow} < d_{aft} \wedge d_{starb} < d_{port}. \\ 180 - arctan\frac{dist_y}{dist_x}, & \text{if } d_{bow} < d_{aft} \wedge d_{starb} > d_{port}. \end{cases} \tag{3.11}$$

$$s12 = \sqrt{dist_x^2 + dist_y^2} \tag{3.12}$$

$$dist_x = \begin{cases} (d_{bow} + d_{aft})/2 - d_{aft} & \text{if } d_{bow} > d_{aft} \\ (d_{bow} + d_{aft})/2 - d_{bow} & \text{if } d_{bow} < d_{aft} \end{cases} \tag{3.13}$$

$$dist_y = \begin{cases} (d_{starb} + d_{port})/2 - d_{port} & \text{if } d_{starb} > d_{port} \\ (d_{starb} + d_{port})/2 - d_{starb} & \text{if } d_{starb} < d_{port} \end{cases} \tag{3.14}$$

$$\begin{cases} \alpha = 360 - (COG - azi1) & \text{if azi1} \leq \text{COG} \\ \alpha = azi1 - COG & \text{if azi1} > \text{COG} \end{cases} \tag{3.15}$$

Figure 3.6 illustrates the geometries that are calculated to find the distance from the transponder to the center, s12, and the azi1 angle. If the transponder is located at point J the length, s12, and azi1 will become:

$$\begin{gathered} dist_{x2} = (|JN| + |JM|)/2 - |JN| \\ dist_{y2} = (|JP| + |JO|)/2 - |JP| \\ \text{s12} = \sqrt{dist_{x2}^2 + dist_{y2}^2} \\ \text{azi1} = \text{COG} + 360 - \arctan \frac{dist_{y2}}{dist_{x2}} \end{gathered} \tag{3.16}$$

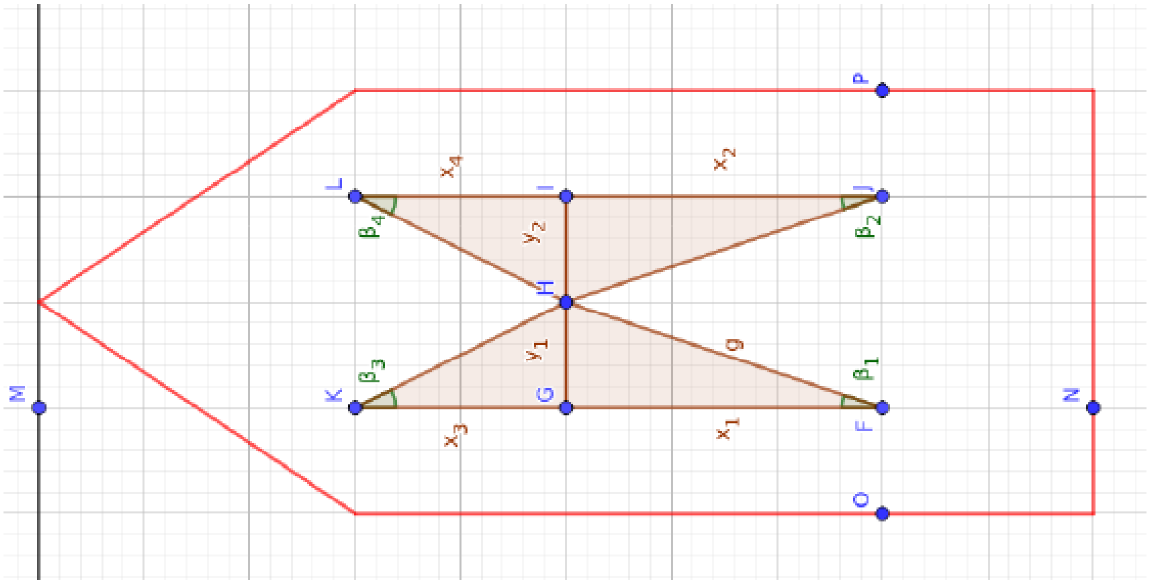


Figure 3.6: Geometries to find the ship's true center (H) where F, K, L and J are representing the location of the AIS transponder (Nordkvist, 2018).

Chapter 4

Case Study

To validate the model a case study has been performed. The chosen fairway is Karmsund in Norway which is a narrow strait with a high traffic density located in western part of Norway, between the island Karmøy and the mainland, shown in Figure 4.2. The length of the fairway is approximately 30 km from Haugesund in north to Boknafjord in south (DNV-GL, 2015), amd in some areas the strait is only a few hundred meters wide. The obstacles trough the strait are well marked and there is a TSS present in the sothern part with a width that varies between 0.7 - 2.0 nautical miles (Slotsvik, 2014). The objective of this thesis is to develop a model to reveal near-misses in narrow fairways. Therfore, Karmsund has been chosen to validate the proposed ship domain.

There are several industry sites located in the area. There is a liquefied natural gas (LNG) distributor located in Avaldsnes operated by Gasnor. Kårstø, a gas terminal, is located east of Karmsund. This means that large tankers are sailing trough the strait. The consequences of an accidents involving a tanker are catastrophic regarding both pollution and loss of life. A VTS center is located in Kvitsøy covering the whole area. The main duty is to monitor the traffic in connection to Kårstø, in addition to surveillance of the coastal traffic (NCA, 2011).

In addition to Karmsund, a part of Boknafjord has been implemented to include the ferry line Mortavika - Arsvågen, see Figure 4.1. The chosen area is limited by coordinates and a time period from 07.July.2010 - 31.Desember.2015. This is the time period for the database used in the case study. The latitude is in the range 59.13 - 59.44 and the longitude 5.20 - 5.60.

The Norwegian Maritime Authorotiy (NMA) is responsible to store every reported accidents and near accidents in Norway (NMA, 2016). The only registered collision between two vessels in the area within the time period was in 2015 between the patrol vessel ”Karm Frøya” and the catamaran ”Ingunn”. This accident is marked as a red star in Figure 4.3. Unfortunately the catamaran did not carry an AIS transponder, and it is therefore not possible to follow the trajectories of the vessel. The trajectory of the patrol vessel can be investigated to identify any change in course or speed when the situation occurred. It should also be mentioned that the accident was not in Karmsund, but still in chosen area.

Figure 4.1: Ferry line Mortavika-Arsvågen

Figure 4.2: Overview of Karmsund from Havbase.no

4.1 Data

The available data used in the case study is processed raw AIS. The raw AIS data was given by the Norwegian Coastal Administration in 2015 and Smestad (2015) processed the raw data into a readable database. The database used in the study is satellite-AIS data and is called ”AISNOR”, which is an excerpt from the global database used by Smestad (2015). The size of the database is

5.2 GB. This is data from all over Norway and up to Svalbard within the time period. For a time period of 4.5 years the database is quite small.

Only the data in the chosen area has been used in the thesis. All the data points within the coordinates of latitude 59.13 - 59.44 and longitude 5.20 - 5.60 have been stored into a CSV file and uploaded into a new database using the database browser for SQLite. The new database is called ”Karmsund” and the size is only 0.1 GB. All the data points have been plotted in Figure 4.3 together with the registered collision. For this area there are 1072 unique MMSI numbers and 22580 messages. This means that each vessel have an average of 21 messages.

As mentioned in Section 3.2.2 there are some errors in AIS data. Some observed errors in the database are listed below.

- Two vessels, a research vessel and a supply vessel, have a speed of 102 knots.
- 150 vessels without a ship type number, NaN.
- 15 vessels with 0 as ship type number.
- Vessels with wrong or several ship type numbers.
- Vessels without dimensions.
- Wrong dimensions. One vessel with breadth = 469 m
- Missing IMO numbers.
- Vessels with less than three AIS messages

The vessels’ dimensions in the database are the distance from the AIS transponder to the ship perpendiculars. These dimensions are used to find the true center and the length of the ship, and it is therefore essential that these number are correct. The total length of the vessel is dim-bow + dim-aft, and if these dimensions are zero, the ship will have no domain since the domain depends on the length. The errors in starboard and port will only affect the true center and the domain can still be used. The true center will only be calculated if the bow and aft dimension are present. All vessels with missing lengths are excluded. There are also some vessels above 400 meters which are erroneous data. These are also excluded.

For the vessels missing a ship type or IMO number, the missing values are ignored. The ship type numbers are only used in statistics and not for near-miss detection. If there should occur missing COG values, the domain will become circular with R_{port} as radius. In the database there are 9261 out of 22580 messages where the speed over ground is zero. It is hard to tell if they are vessels at anchor or erroneous values, and the erroneous data is therefore kept as zero.

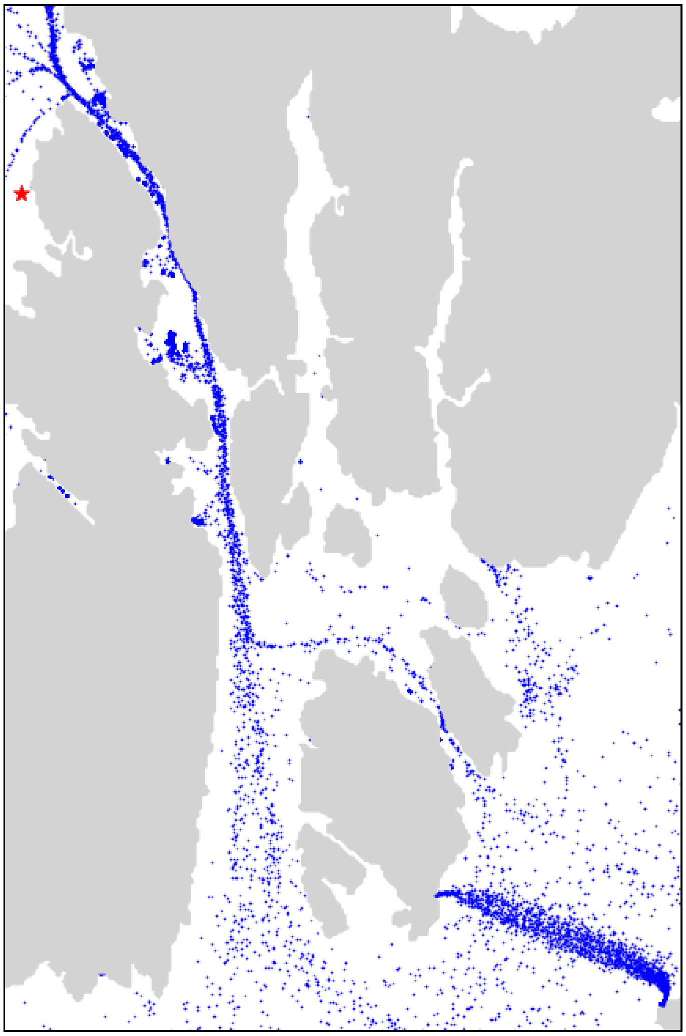


Figure 4.3: AIS plot of registered messages. The red star represent the location of a real collision

The two vessels with a speed of 102.3 knots are removed. In the "AISNOR" database several vessels have the exact speed of 102.3 knots. Smestad (2015) indicated that this might be a systematic error. Nordkvist (2018), who used another database, did also get values of 102.2 knots and pointed out that this was an error. There are several high speed vessels in the area with the highest recorded speed of 36.7 knots. To handle the erroneous SOG, all values above 40 knots have been excluded.

To determine the type of vessel each ship has its own code in the AIS. The main categories are listed in Table 4.1. Each ship type consists of two digits, the first digit represent the general category of vessels, while the second digit represent, in some cases, information regarding the hazardous lever of the cargo (MarineTraffic, 2018).

AIS vessel code	Vessel type
10	Reserved
20	Wing in ground
30	Fishing
40	High-speed craft
50	Special category
60	Passenger
70	Cargo
80	Tankers
90	Other

Table 4.1: AIS vessel type and main group codes

Figure 4.4 shows the distribution of the vessel types in the ”Karmsund” database. The total number of vessels are 1155. Cargo vessels have the highest contribution with 38 % followed by Fishing vessels (16 %) and tankers (12 %). The number of total vessels should match the number of unique MMSI numbers of 1072. The reason for this mismatch is due to an error in the database. Some vessels have more than one ship type number in different categories. One example is a tanker with a ship type number 81, which is the correct number, but it also has the number 70 that indicates a cargo vessel. Several vessels have zero as the ship type number in addition to the correct value. Therefore, some ships are counted two times.

As mentioned previously, vessels below 300 GT are not obligated to carry an AIS transponder which means that for example smaller fishing vessels are not included in the database or in the distribution. Those vessels in the Reserved category have 0 as ship type number, which is an error. For example, several of these ships are cargo vessels and should have 70 instead of 0.

The longest ship in the database is a container ship with a length of 300 meters. This ship was sailing to Kårstø trough Boknafjord and did not sail in Karmsund. The longest vessel sailing in Karmsund is a tanker with a total length of 277 meters. If this tanker sailed in 10 knots, the total breadth of the domain is 605 meters which is larger the the total width of the fairway in some areas.

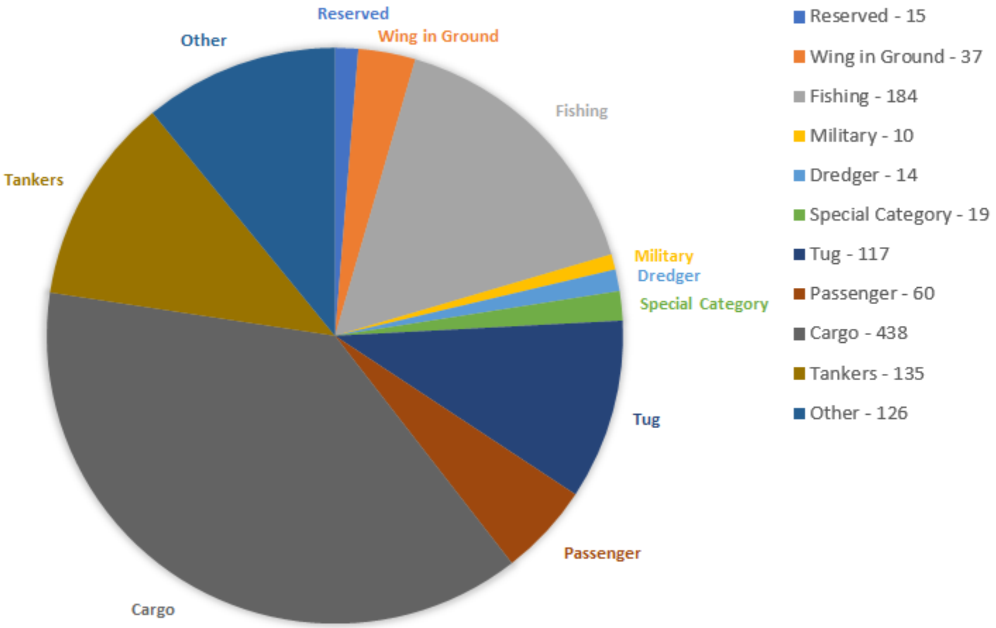


Figure 4.4: Distribution of vessel types in the database

Figure 4.5 shows the trajectories of two arbitrarily cargo vessels. In a) there are nine data points while there are 15 in b). The markers in the figures are the registered AIS messages and the lines demonstrate linearly interpolation between the messages. The use of linear interpolation between the messages is a common method to estimate the vessel's trajectories and for replacing erroneous or missing values (Aarsæther and Moan, 2009). Goerlandt et al. (2012) used interpolation to reconstruct the trajectories for each second to detect near-misses. To use a time interval of just one second will lead to a huge database, but it will be easier to detect overlapping trajectories and near-misses.

As Figure 4.5 shows there are long distances between the data points, which leads to incorrect trajectories where the lines are crossing land. The paths are clearly wrong when the time interval between the AIS messages is large. Therefore, interpolation has not been used in the proposed method to detect near-misses.

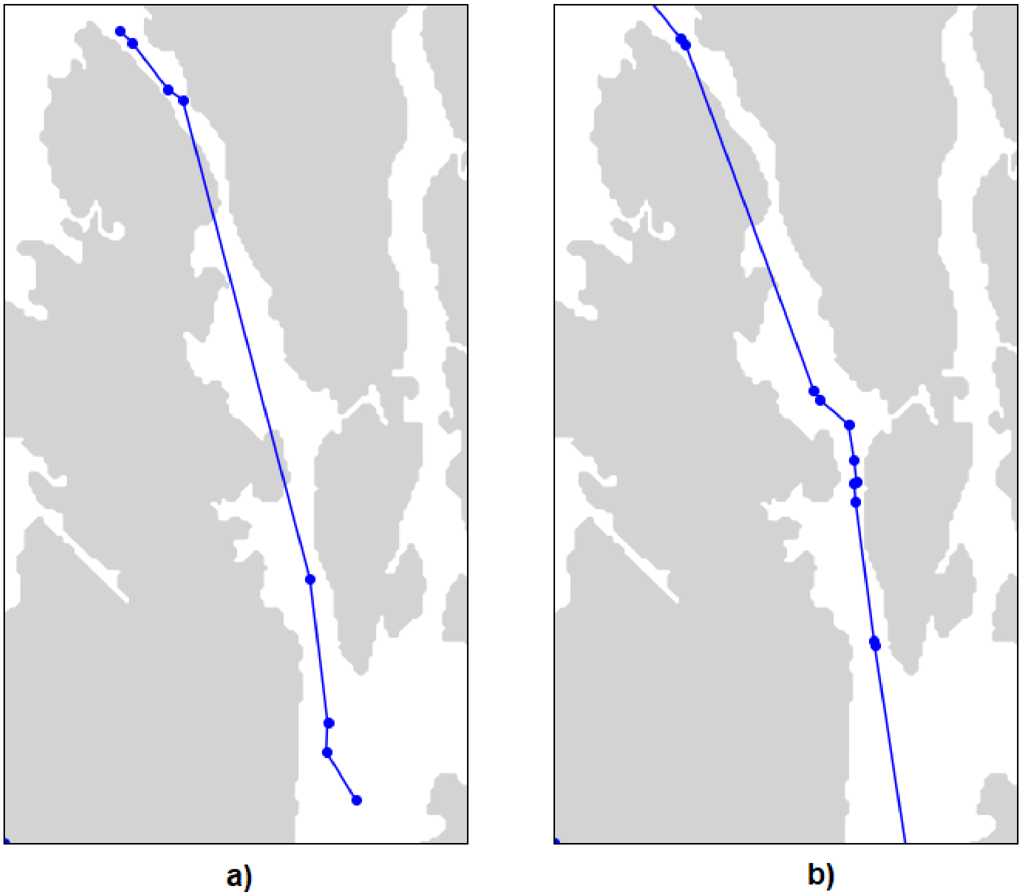


Figure 4.5: Interpolation of two arbitrarily vessels' trajectories

4.2 Results From the Case Study

The proposed method gives zero detected near-misses in the area. The result from the case study implies that there are no near-misses in Karmsund or Boknafjord between 07.July.2010 and 31.Desember.2015. A hypothesis for getting zero results is that the used database is insufficient with too few messages and too long time interval between the messages. To check this hypothesis, a new database has been acquired.

4.3 Case Study With New Database

To validate the method a new database has been used to reveal near-misses for the same area. The new database is called ”BasestationAUG2017” which is data from August 2017 only, with a total size of 40 GB. The database was provided by the Norwegian Coastal Administration, and the raw AIS data was processed before it was given to the author of this thesis 03.June.2019.

In the new case study Boknafjord is excluded and only near-misses inside Karmsund are identified. This has been chosen since the processing time is several hours, and it is sufficient to only study Karmsund for validation of the model. The area is confined by the latitude 59.20 - 59.42 and longitude 5.225 - 5.365. All the messages from Karmsund is stored in a new database called ”2017” with a size of 1.0 GB. In August 2017 there were no registered collisions or near collision in the area which could be used to validate the model.

There are 387 unique MMSI numbers and a total of 4 692 589 messages in the ”2017” database. This is 208 times as many messages as in the database used in the previous case. Several of these 387 vessels have incorrect values regarding the length: 13 vessels have no length and 5 vessels have a incorrect length above 400 meters. There are also a dozen of erroneous MMSI numbers. The longest ship with a correct length is a container vessel with a length of 400 meters. The vessels' dimensions in the ”BasestationAUG2017” are only given in length and breadth, unlike in the ”AISNOR” database where the dimensions to the perpendiculars from the AIS transponder were given. It is therefore not possible to find the true center, and it is assumed that the transponder is located in the center of the ship.

4.3.1 Detected Near-misses

With the new database the proposed algorithm for detecting near-misses stored over 100 000 messages. Over half of these were empty rows, and the same encounters were stored several times. With further processing and removing the erroneous data, 1337 unique near-misses were identified within the time period, which gives an average of 3.4 near-misses per vessel. 15 % of the vessels in the database are above 100 m. However, these vessels contribute to 478 out of 1337 near-misses. This means that vessels above 100 meters are involved in 35 % of the near-misses even tough there are only 15 % in the database.

The locations for the revealed near-misses are plotted in a density plot in Figure 4.6. As the figure shows they are not evenly distributed trough the whole strait, but are located in clusters where most of them are located in the narrow parts of the strait. The red area, with highest density, is located in the narrowest part of Karmsund with a breadth of only 200 meters. The distribution of collision types is listed in the table below, where overtaking and head-on encounters dominate with 55 % and 41 % respectively, while crossing encounters are only 4 % .

Collision types	**Number of near-misses**
Overtaking	739
Head-on	552
Crossing	46
Total	1337

Table 4.2: Distribution of near-misses

The distribution of the three collision types is plotted in Figure 4.7. For overtaking and and head-on encounters the plots are quite similar where the near-misses are located trough whole Karmsund. In the southern part there is difference between overtaking and head-on. A TSS is present in this part of the strait and head-on encounters should not occur there. There are still few examples, but they are much rarer than overtaking encounters. The crossing encounters are also distributed trough the whole strait, but are most common in the northern inlet of strait and in the narrowest part.

A major error with the revealed near-misses is that for some encounters there have not been any encounter at all. When a near-miss is detected, the time of occurrence is given, and in some cases one of the involved vessels was not in the area during the time. The trajectories are correct, but the time may have been several hours, or even days later. It is uncertain how many incidents this applies to.

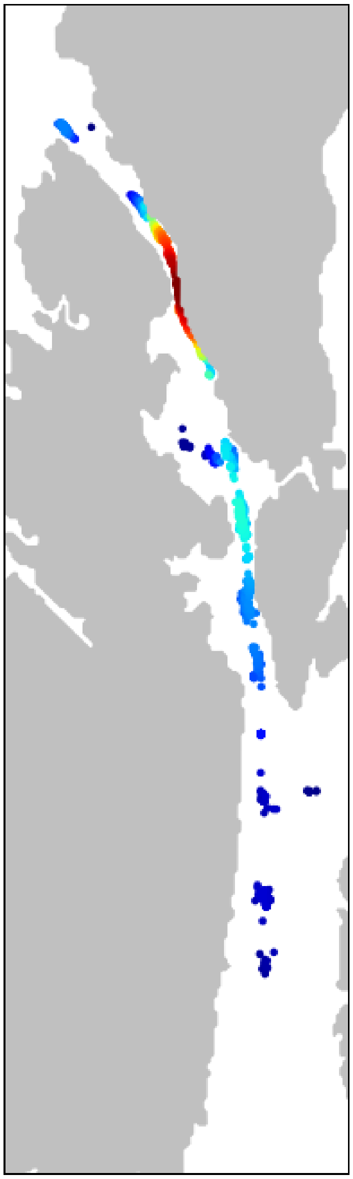


Figure 4.6: Density plot for the identified near-misses in August 2017.

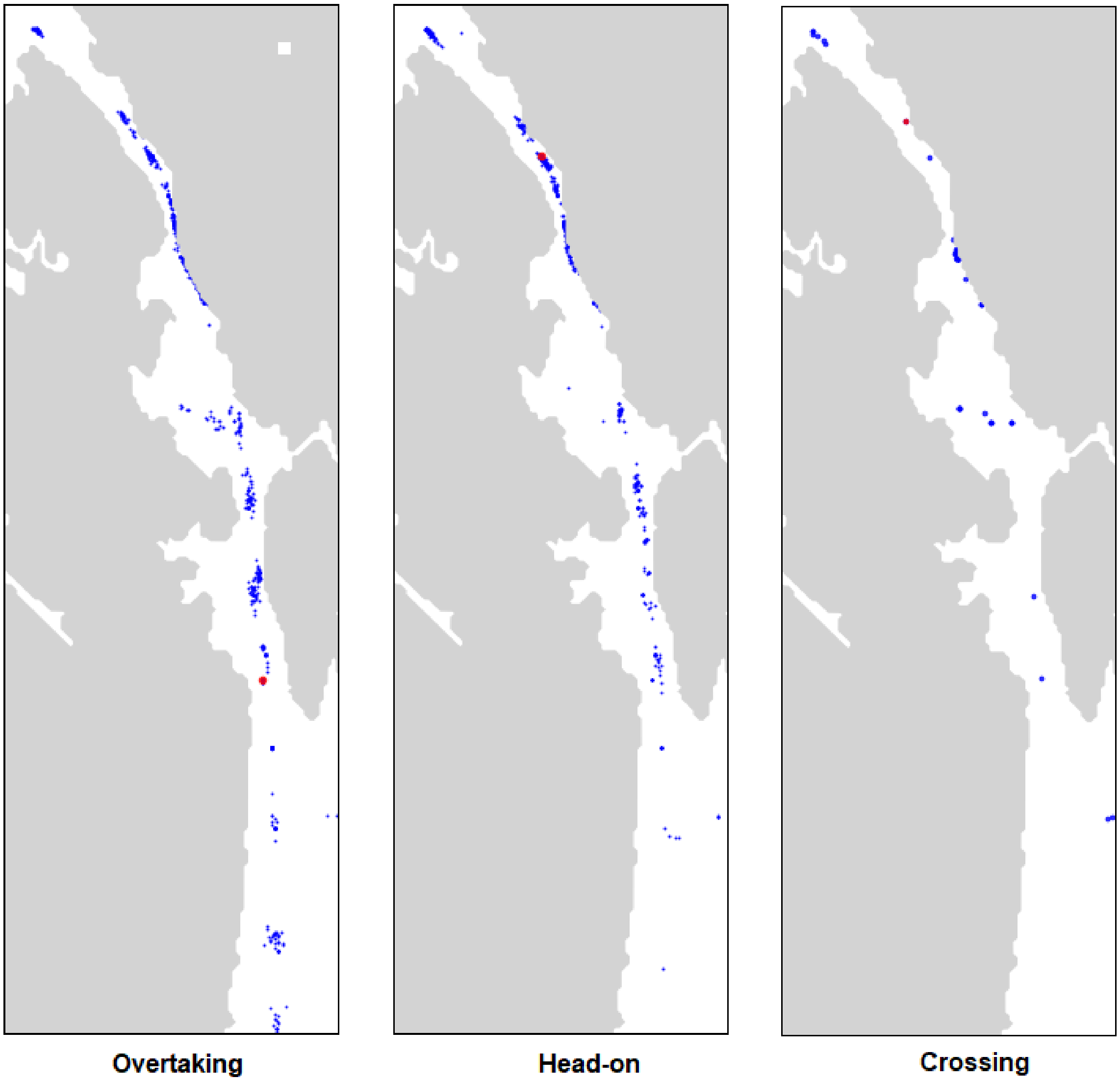


Figure 4.7: The figures show the distribution of near-misses for overtaking, head-on and crossing encounters. The red dots are the encounters used for further analysis

4.3.2 Examples of Detected Near-misses

In this section some examples of the identified near-misses are shown. There are three examples, one of each collision types. These have been randomly chosen from the table of detected near-misses. In Figure 4.8 the overtaking encounter has been plotted. The blue line is a 89 m long offshore supply vessel (OSV) sailing in 9.1 knots when the domain was breached and the near-miss was detected. This gives a domain length and breadth of 456 m and 92 m respectively. The red line is the overtaking vessel, a cargo ship with a length of 130 m sailing in 14 knots. Here the domain length and breadth are 767 m and 170 m respectively.

The largest domain will always be the violated domain, which is the cargo vessel in this case. The distance between the centers is 349 meters and the domain is violated by 22 meters. The relative bearing between the vessels, α, seen from the cargo vessel is $352°$. It should be mentioned that this distance is not necessarily the closest distance between the vessels, but it is the registered distance when the violation first is detected.

As the figure shows the vessels keep a steady course and they have no abrupt movements, which means that there are no indications that this is a near-miss even though they are sailing closely to each other and it is defined as a near miss in this case study. The OSV's domain is not violated and the OSV should not give way to the cargo vessel since the cargo vessel is the overtaking ship. Therefore, this situation may be seen as a safe encounter in the OSV's point of view.

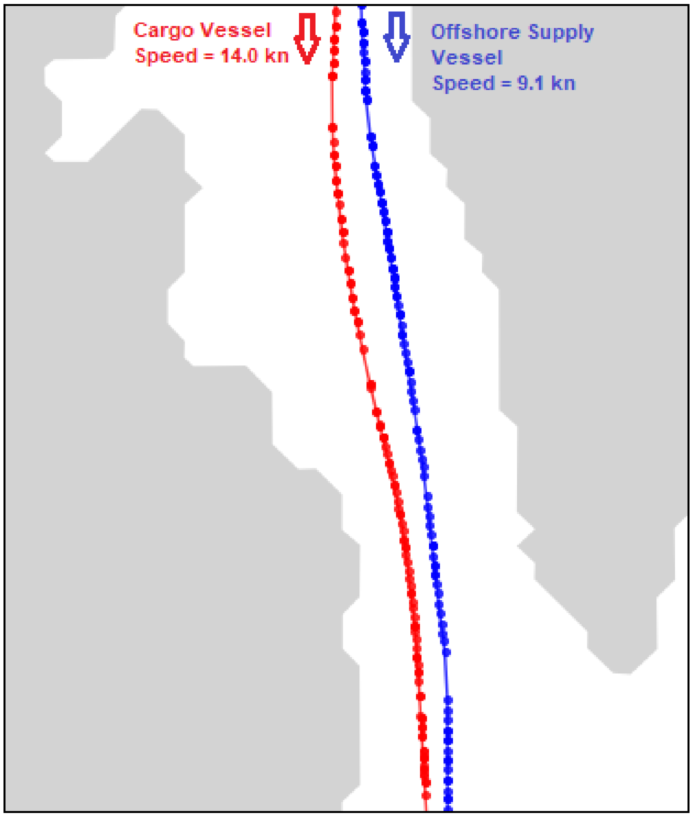


Figure 4.8: Example of overtaking encounter

The examples of head-on and crossing encounters are plotted in Figure 4.9 where a) is the head-on and b) the crossing. The head-on example is located in an area with a high density of head-on encounters, while the crossing example is the only one in that area. See Figure 4.7 for the locations.

The head-on encounter is between to cargo vessels. The blue line is the trajectory of a vessel with length 75 meters and speed 12.8 knots. The other vessel has a length of 107 meters sailing in 7.7 knots. The distance between the vessels is 227 meters. The northbound vessel is 64 meters inside the domain of the southbound vessel. It is evident from the figure that the red line follows a steady course, while the blue line must alter the course to starboard to avoid collision. The change of course was initiated early and there are no high ROT values before the encounter. If there was a high ROT value, it may be an indicator that this encounter is a near-miss, but since the change of course was initiated early the two vessels had most likely the situation under control.

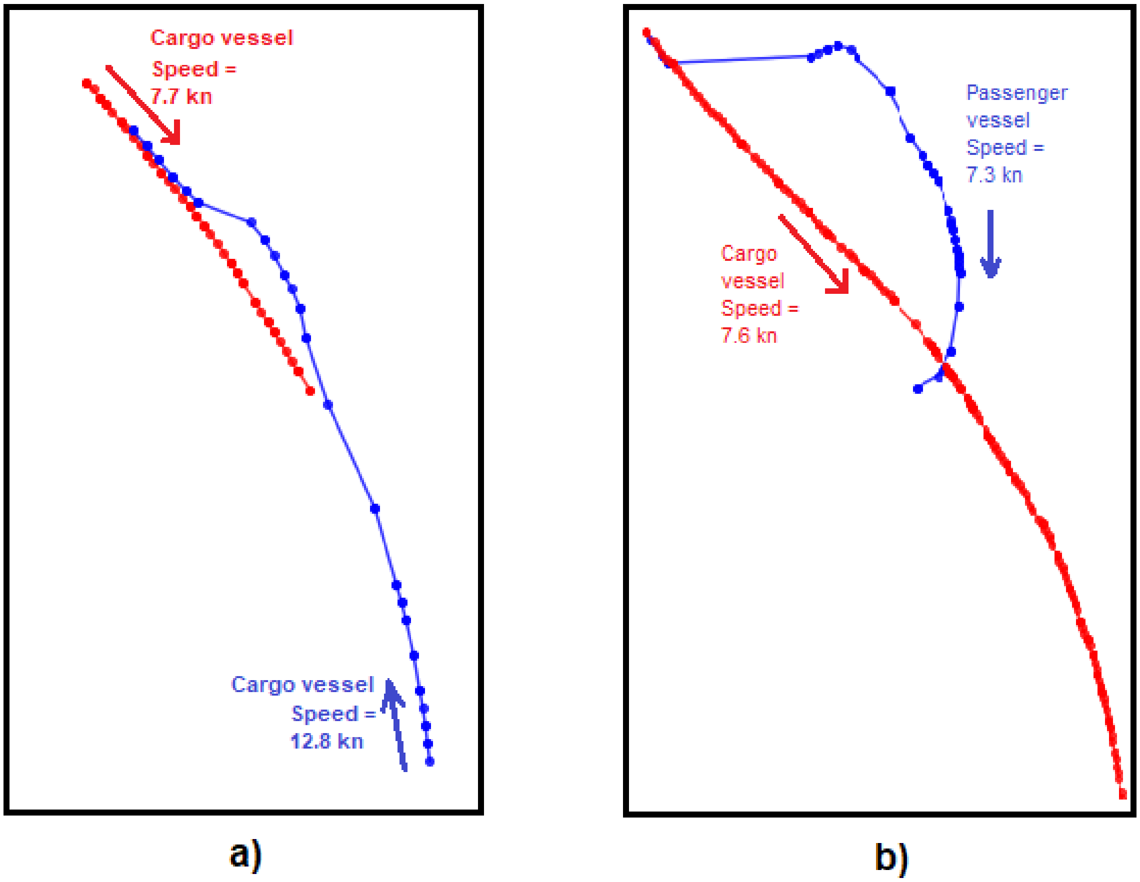


Figure 4.9: Example of head-on (a) and crossing (b) encounters

The crossing situation is a an encounter between a cargo and a passenger vessel with a length of 77 m and 20 m and a speed of 7.6 kn and 7.3 kn respectively. The distance between the ship centers is 173 m and the largest domain is violated with 13 meters. From the figure it looks like the passenger vessel is sailing in an odd pattern, but it is following a ferry rout inside Karmsund. The passenger vessel should give way to the cargo vessel and cross behind since the cargo vessel is located on starboard side. This is preserved by the passenger vessel and it does cross behind. Therefore, the cargo vessel can sail in a straight course and does not need to take action. The relative bearing, seen from the cargo vessel, is 177°, which means that the domain has been violated in the aft end. It can also in this example be discussed if this can be defined as a near-miss.

Chapter 5

Discussion

5.1 AIS Data

As mentioned in the Method chapter and observed in the case study, there are several errors in the AIS messages. The model in this thesis highly depends on the speed and length of the vessels to get a correct domain. The vessels dimensions are also important regarding finding true center. Correct position and time are essential to find concurrent overlapping trajectories. With all these parameters the model is vulnerable to errors in the AIS messages.

In some studies the vessels with erroneous data have been included, and missing or wrong values have been set to a standard value. Those vessels with extreme erroneous data have been excluded in this thesis. Extreme values are missing numbers, zero values or unrealistically high values. It has been chosen to exclude them since standard values will give wrong results, and a wrong result will be of the same value as no result. The course over ground is the only parameter where erroneous data have been included since this only affect the shape of the domain. The domain will then be a circle with a minimum size.

The model is not only dependent on reliable data, but also an adequate database. The initiated case study clearly demonstrated the importance of a large database where the time interval between the messages should be low. Without high resolution data it is hard to get correct trajectories. Several studies use interpolation between the messages, and the advantage is that overlapping trajectories are easier to detect which is used to detect near-misses. When the time interval or the distance between the messages become large, it is not sufficient to interpolate between the messages. This is evident in Figure 4.5 where these lines are crossing land. The different results from the two case study showed the importance of high resolution data and a large database, especially since the model in the first case study was not able to detect a single near-miss in a five year time period, while the second case study 1337 near-misses were detected in just one month.

5.2 The Method for Detecting Near-misses

The proposed method uses ship domain to define and identify near-misses, which is the most common and recognized method as described in the introduction. There have been several studies the last years regarding the use of ship domain in maritime risk assessments and there is still a need for further development.

From the second case study 1337 near-misses were detected with an average of 3.4 near-misses per vessel. This is unrealistically high for just one month, and the definition of near-misses can be questioned. Even though a small domain has been chosen, it still too large for the chosen area. For large vessels the domain will be wider then the total width of the fairway. This will lead to all encounters are classified as a near-miss. To avoid this an even smaller domain should be used. In narrow straits the navigators need to pay more attention than for open waters, and they are therefore more aware to other vessels. This means that they can sail closely to each other without be defined as a near-miss. Furthermore, the proposed ship domain is therefore not applicable alone to define near-misses for the chosen area. Another drawback with using Karmsund as a case study is that there are no registered collision or near collision there. The model can therefore not identify familiar situations which could be of high value when it comes to validation of the model.

To use ship domain to reveal close encounters utilizing AIS data has been proven in both this and other studies to be an excellent tool. However, to classify a domain violation as a near-miss is not sufficient. This is evident from the three examples in Section 4.3.2 where the domains have been breached, but there are no other indications that they are near-misses. There are many factors involved in a near-miss, and further analysis like expert judgement should be done to define close encounters as near-misses. This has been done among others by Zhang et al. (2015). Mestl et al. (2016) and Nordkvist (2018) use high rate of turn values to define a near-miss, which is the second step after the domain violations have been revealed. High ROT values may indicate that the navigators are trying to avoid a collision, and using ROT as the second step will exclude the normal encounters which again will reduce the workload for the expert judgement.

The model used in this thesis has shown that it is capable of detecting encounters with domain violation, but a major drawback with the algorithm is that a lot of the identified encounters never took place, which is described in Section 4.3.1. The extend of this problem is uncertain, and it has not thoroughly been investigated due to limited time between the date the new database was given to the author to the deadline day.

In the second case study it was not possible to find the true center since the only length given in the database was the total length of the vessel. The lengths to the fore, aft, starboard and port perpendiculars from the AIS transponder were missing. The consequences of not finding the true center is that the domain will be staggered relative to the vessel.

Chapter 6

Conclusion

The objective of this thesis has been to develop a model to detect near-misses between two vessels using AIS data and to identify patterns between them. The proposed method is a combination of previous studies within the topic, and a new ship domain has been developed to detect near-misses in narrow fairways. The model has been tested in a case study for two different AIS databases. In the first case a database from 2010-2015 was used, and the model was not able to detect any near-misses due to high time intervals between the messages. In the second case study a database with high resolution AIS data was used and 1337 near-misses were detected in August 2017 only.

The identified patterns between the near-misses are that overtaking and head-on encounters are dominating for the chosen area while crossing encounters are rear. This matched the initiated hypothesis. The other identified patterns were that most of the near-misses were located in the narrow parts of the fairway, and large vessels had a high contribution.

The two case studies have proved that the model is able to detect near-misses, but for high resolution data only. They have also proven that the proposed domain is too large and not applicable for Karmsund, even though it is quite smaller than other well known ship domains. The domain is too large since most of the identified near-misses were just close encounters and could not be classified as a near-miss. Ship domain should not be used alone without further analysis to define near-misses. It is well suited to identify close encounters, but not near-misses.

6.1 Recommendations for Further Work

It is recommended to continue the research within near-miss detection. To use ship domain for identifying near-misses need more research, but also new methods due to the drawbacks with ship domains should be investigated. To validate the proposed domain a case study in a new area is recommended, and the model should be used to detect encounters for further analysis where for example rate of turn values are included. The use of ship domain alone to detect near-misses is not sufficient, but it should be an early step in the process to identify them.

The algorithm is working well to find close encounters, but the error regarding false encounters should be investigated before using it, especially since the extent of this error is unknown.

Bibliography

Karl Gunnar Aarsæther and Torgeir Moan. Estimating navigation patterns from ais. *The Journal of Navigation*, 62(4):587–607, 2009.

COWI. Be-aware methodology note, 2012.

DNV-GL. Analyse av sannsynligheten for ulykker med tap av menneskeliv og akutt forurensning fra skipstrafikk i norske farvann. *Report No. 2014-1060, Rev. B*, 2014.

DNV-GL. Lengdebegrensninger ved bruk av farledsbevis. *Risikoanalyse av farleder*, 2015.

ecolregs. Colregs rule13, 2018. URL `https://ecolregs.com/index.php?option=com_k2&view=item&layout=item&id=55&Itemid=388&lang=en`.

Magnus S Eide, Øyvind Endresen, Øyvind Breivik, Odd Willy Brude, Ingrid H Ellingsen, Kjell Røang, Jarle Hauge, and Per Olaf Brett. Prevention of oil spill from shipping by modelling of dynamic risk. *Marine Pollution Bulletin*, 54(10):1619–1633, 2007a.

Magnus S Eide, Øyvind Endresen, Per Olaf Brett, Jon Leon Ervik, and Kjell Røang. Intelligent ship traffic monitoring for oil spill prevention: Risk based decision support building on ais. *Marine pollution bulletin*, 54(2):145–148, 2007b.

P Friis-Hansen. Iwrap mk ii, 2008.

Y Fujii. Some factors affecting the frequency of accidents in marine traffic. *Journal of Navigation*, pages 239–252, 1974.

GIS Geography. World geodetic system (wgs84), 2018. URL `https://gisgeography.com/wgs84-world-geodetic-system/`.

Floris Goerlandt, Jakub Montewka, Heikki Lammi, and Pentti Kujala. Analysis of near collisions in the gulf of finland. *Advances in Safety, Reliability and Risk Management*, pages 2880–86, 2012.

Elisabeth M Goodwin. A statistical study of ship domains. *The Journal of navigation*, 28(3): 328–344, 1975.

IALA. Iala guideline 1082 – an overview of ais, 2016. URL `https://www.iala-aism.org/product/an-overview-of-ais-1082/`.

IMO. Guidance on near-miss. *MSC-MEPC.7/Circ.7*, 2008.

WH Iperen. Classifying ship encounters to monitor traffic safety on the north sea from ais data. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 9, 2015.

B Kleppe. Automatic identification system, ais, 2016. URL `https://www.kystverket.no/en/EN_Maritime-Services/Reporting-and-Information-Services/Automatic-Identification-System-AIS/`.

B Kleppe. Satellite-based ais, 2017. URL `https://www.kystverket.no/Maritime-tjenester/Meldings--og-informasjonstjenester/AIS/AISSat-1-og-AISSat-2/`.

Jon Hovem Leonhardsen. Estimation of fuel savings from rapidly reconfigurable bulbous bows-exemplifying the value of agility in marine systems design. Master's thesis, NTNU, 2017.

Suyi Li, Qiang Meng, and Xiaobo Qu. An overview of maritime waterway quantitative risk assessment models. *Risk Analysis: An International Journal*, 32(3):496–512, 2012.

T Macduff. The probability of vessel collisions. *Ocean Industry*, 9(9), 1974.

MarineTraffic. What is the significance of the ais shiptype number?, 2018. URL `https://help.marinetraffic.com/hc/en-us/articles/205579997-What-is-the-significance-of-the-AIS-Shiptype-number-`.

Thomas Mestl, Ken Tore Tallakstad, and R Castberg. Identifying and analyzing safety critical maneuvers from high resolution ais data. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 10, 2016.

J Montewka, P Krata, F Goerlandt, A Mazaheri, and P Kujala. Marine traffic risk modelling–an innovative approach and a case study. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 225(3):307–322, 2011.

NCA. Vts centres, 2011. URL `https://www.kystverket.no/en/EN_Maritime-Services/Vessel-Traffic-Service/VTS-Centres/`.

NMA. Accident statistics, 2016. URL `https://www.sdir.no/sjofart/ulykker-og-sikkerhet/ulykkesstatistikk/`.

Haakon Akse Nordkvist. An advanced method for detecting exceptional vessel encounters in open waters from high resolution ais data. Master's thesis, NTNU, 2018.

Patrick Andre Næss. Investigation of multivariate freight rate prediction using machine learning and ais data. Master's thesis, NTNU, 2018.

Preben Terndrup Pedersen et al. Collision and grounding mechanics. *Proceedings of WEMT*, 95 (1995):125–157, 1995.

M Series. Technical characteristics for an automatic identification system using time-division multiple access in the vhf maritime mobile band, 2010.

PAM Silveira, AP Teixeira, and C Guedes Soares. Use of ais data to characterise marine traffic patterns and ship collision risk off the coast of portugal. *The Journal of Navigation*, 66(6): 879–898, 2013.

Kirsti Slotsvik. Farledsnormalen. *The Norwegian Coastal Administration*, 2014.

Bjørnar Brende Smestad. A study of satellite ais data and the global ship traffic through the singapore strait. Master's thesis, NTNU, 2015.

Rafal Szlapczynski and Joanna Szlapczynska. Review of ship safety domains: Models and applications. *Ocean Engineering*, 145:277–289, 2017.

Fulko van Westrenen and Joost Ellerbroek. The effect of traffic complexity on the development of near misses on the north sea. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(3):432–440, 2017.

Ning Wang. An intelligent spatial collision risk based on the quaternion ship domain. *The Journal of Navigation*, 63(4):733–749, 2010.

Weibin Zhang, Floris Goerlandt, Jakub Montewka, and Pentti Kujala. A method for detecting possible near miss ship collisions from ais data. *Ocean Engineering*, 107:60–69, 2015.

Weibin Zhang, Floris Goerlandt, Pentti Kujala, and Yinhai Wang. An advanced method for detecting possible near miss ship collisions from ais data. *Ocean Engineering*, 124:141–156, 2016.

Appendix A

Appendix

A.1 Acronyms

AIS	Automatic Identification System
CA	Centripetal Acceleration
COLREGs	Convention of the International Regulations for Preventing Collisions at Sea
CSV	Comma Separated Values
DCPA	Distance at Closest Point of Approach
ETA	Estimated Time of Arrival
IALA	International Association of Lighthouse Authorities
IMO	International Maritime Organisation
ITU	International Telecommunication
MDTC	Minimum Distance to Collision
MMSI	Maritime Mobile Service Identity
NCA	Norwegian Coastal Administration

NMA	Norwegian Maritime Authorotiy
OS	Own Ship
OSV	Offshore Supply Vessel
QRA	Quantitative Risk Assessments
QSD	Quaternion Ship Domain
ROT	Rate of Turn
RRM	Risk-reducing measures
S-AIS	Satellite Automatic Identification System
TCPA	Time to Closest Point of Approach
TS	Target Ship
TSS	Traffic Separation Schemes
VCRO	Vessel Conflict Ranking Operator
VHF	Very High Frequency
VTS	Vessel Traffic Service

A.2 Plotting Domains

```
# Plotting domain for different speeds

import numpy as np
import math
from matplotlib import pyplot as plt
from math import pi

v = np.array([5,10,15,20]) #speed in knots
L=150 #Length of own ship
k_AD=10**(0.3591*np.log10(v)+0.0952)
k_DT=10**(0.5441*np.log10(v)-0.0795)
Rf = (1+1.34*np.sqrt(k_AD**2+(k_DT/2)**2))*L
Ra = 0.5*Rf
Rs = (0.75/2*k_DT)*L           # Radius on the x-axis
Rp = Rs
t = np.linspace(0, 2*pi, 100)

bA0 = (Rf[0]+Ra[0])/2          # b = Radius on the y-axis
bB0 = (Rf[0]+Ra[0])/4
s0 = np.sin(t)
s0[0:50] = bA0*s0[0:50]
s0[51:100] = bB0*s0[51:100]

bA1 = (Rf[1]+Ra[1])/2
bB1 = (Rf[1]+Ra[1])/4
s1 = np.sin(t)
s1[0:50] = bA1 * s1[0:50]
s1[51:100] = bB1*s1[51:100]
```

```
bA2 = (Rf[2]+Ra[2])/2
bB2 = (Rf[2]+Ra[2])/4
s2 = np.sin(t)
s2[0:50] = bA2 * s2[0:50]
s2[51:100] = bB2*s2[51:100]

bA3 = (Rf[3]+Ra[3])/2
bB3 = (Rf[3]+Ra[3])/4
s3 = np.sin(t)
s3[0:50] = bA3 * s3[0:50]
s3[51:100] = bB3*s3[51:100]

fig = plt.figure(figsize=(5,6))
ax = fig.add_axes([0,0,1,1])
ax.plot(Rs[0]*np.cos(t) , s0,c='r',label='V = 5kn')
ax.plot(Rs[1]*np.cos(t) , s1,label='V = 10kn')
ax.plot(Rs[2]*np.cos(t) , s2,label='V = 15kn')
ax.plot(Rs[3]*np.cos(t) , s3,label='V = 20kn')
ax.plot(1.6*L*np.cos(t), 4*L*np.sin(t),ls='--',c='gray',label='Fujji domain')
ax.legend(loc=0)
ax.set_aspect('equal')
ax.set_xlim(-800, 800)
ax.set_ylim(-800, 1000)
ax.figsize=(30,30)
plt.xlabel('Lateral axis (meter)')
plt.ylabel('Longitudinal axis (meter)')
#Plotting domains for different lengths

v = 10 #speed in knots
L=np.array([100,150,200]) #Length of own ship
k_AD=10**(0.3591*np.log10(v)+0.0952)
```

```
k_DT=10**(0.5441*np.log10(v)-0.0795)
Rf = (1+1.34*np.sqrt(k_AD**2+(k_DT/2)**2))*L
Ra = 0.5*Rf
Rs = (0.75/2*k_DT)*L            # Radius on the x-axis
Rp = Rs
t = np.linspace(0, 2*pi, 100)

bA0 = (Rf[0]+Ra[0])/2           # b = Radius on the y-axis
bB0 = (Rf[0]+Ra[0])/4
s0 = np.sin(t)
s0[0:50] = bA0*s0[0:50]
s0[51:100] = bB0*s0[51:100]

bA1 = (Rf[1]+Ra[1])/2
bB1 = (Rf[1]+Ra[1])/4
s1 = np.sin(t)
s1[0:50] = bA1 * s1[0:50]
s1[51:100] = bB1*s1[51:100]

bA2 = (Rf[2]+Ra[2])/2
bB2 = (Rf[2]+Ra[2])/4
s2 = np.sin(t)
s2[0:50] = bA2 * s2[0:50]
s2[51:100] = bB2*s2[51:100]

fig = plt.figure(figsize=(5,6))
ax = fig.add_axes([0,0,1,1])
ax.plot(Rs[0]*np.cos(t) , s0,c='r',label='L = 100 m')
ax.plot(Rs[1]*np.cos(t) , s1,label='L = 150 m')
ax.plot(Rs[2]*np.cos(t) , s2,label='L = 200 m')
```

```
ax.plot(1.6*150*np.cos(t), 4*150*np.sin(t),ls='--',c='gray',label='Fujji domain
ax.legend(loc=0)
ax.set_aspect('equal')
ax.set_xlim(-800, 800)
ax.set_ylim(-800, 1000)
ax.figsize=(30,30)
plt.xlabel('Lateral axis (meter)')
plt.ylabel('Longitudinal axis (meter)')
```

A.3 Preparing the Database

```
# Finding the messages from Karmsund only and save them in an own dtabase
databasepath = ('AISNOR.db')
con = lite.connect(databasepath)
inputquery1 = "SELECT * FROM MessageType1 \
WHERE latitude < 59.44 and latitude > 59.20 and longitude < 5.60 \
and longitude > 5.20"

with con:
        AIS_Data = pd.read_sql_query(inputquery1, con)
con.close()
AIS_DataMT1.to_csv('Karmsund_MT1.csv', index=False)

#Import the csv file to the database using DB Browser
databasepath = ('Karmsund.db')
con = lite.connect(databasepath)
inputquery2 = "SELECT * FROM Karmsund_MT1 \
LEFT JOIN MessageType5 ON Karmsund_MT1.userid = MessageType5.userid"
with con:
        AIS_DataMT5 = pd.read_sql_query(inputquery2, con)
con.close()
KarmMT5= AIS_DataMT5[['unixtime','latitude','longitude','userid','ship_type',
                      'dim_bow','dim_port','dim_starboard','dim_stern','imo']]
KarmMT5.to_csv('Karmsund_MT5.csv', index=False)
Karm5 = karm5[['unixtime','latitude','longitude','userid','ship_type',
               'dim_bow','dim_port','dim_starb','dim_stern','imo']]
Name = ['unixtime','delete','latitude','longitude','userid','delete1',
        'ship_type','dim_bow','dim_port','dim_starb','dim_stern','imo']
karm5 = pd.read_csv('Karmsund_MT5.csv', names=Name, header=0)
# Karm5 is the table used to detect near-misses
```

A.4 Preparing the New Database

```
# This script collects data for Karmsund from "BasestationAUG2017"
# which is stored in a new database called "2017"

import sqlite3 as lite
import numpy as np
import pandas as pd
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import datetime as datetime
import seaborn as sns
%matplotlib inline

# Choosing data from Karmsund only:
databasepath = ('basestationAUG2017.db')
con = lite.connect(databasepath)
inputquery = "SELECT unixtime, mmsi, latitude, longitude, sog, cog \
FROM messagetype1 \
WHERE latitude < 59.415 and latitude > 59.2 and longitude < 5.365 \
and longitude > 5.225"
inputquery2 = "SELECT distinct mmsi, length FROM messagetype5"

with con:
        AIS_Data = pd.read_sql_query(inputquery, con)
        DistMMSI = pd.read_sql_query(inputquery2, con)
con.close()
A = AIS_Data.drop_duplicates()
# The CSV files are manually added into the database
# using DB browser for SQLite
A.to_csv('MT1.csv', index=False)
```

```
DistMMSI.to_csv('Dist_MMSI.csv', index=False)

# Make a new table with all the information needed to detect a near-miss
databasepath = ('2017.db')
inputquery2 = "SELECT MT1.unixtime,MT1.mmsi,sog,longitude,latitude,cog, \
length FROM MT1 LEFT JOIN Dist_MMSI ON MT1.mmsi = Dist_MMSI.mmsi \
WHERE MT1.unixtime < 1504223998"
con = lite.connect(databasepath)
with con:
        AISa = pd.read_sql_query(inputquery2, con)
con.close()
AISa.drop_duplicates()
AISa['Time']= AISa['Time'].astype('datetime64[s]') # Timestamp to date
AISa.to_csv('AIS_2017a.csv', index=False)
```

A.5 Detection of Near-misses

```
# This is the code for detection of near-misses for the 2017 database. Finding
# true center has been included here, but is removed when running the script.

import sqlite3 as lite
import numpy as np
import pandas as pd
import csv
import multiprocessing
import matplotlib.pyplot as plt
import seaborn as sns
import time
import math
from math import pi
from mpl_toolkits.basemap import Basemap
from datetime import datetime
from datetime import timedelta
from joblib import Parallel, delayed
from dateutil.relativedelta import relativedelta
from geographiclib.geodesic import Geodesic

# Angle from heading of own ship to position of other ship
def alpha(lon_0, lat_0, lon_2, lat_2, cog_0):
    azi1 = Geodesic.WGS84.Inverse(lat_0, lon_0, lat_2, lon_2)['azi1']
    if type(cog_0) == str:
        if azi1 < 0:
            return np.radians(360 + azi1)
        else:
            return np.radians(azi1)
    else:
```

```
        if azi1 < 0:
            azi1 = 360 + azi1
        if cog_0 >= azi1:
            return np.radians(360 - (cog_0 - azi1))
        else:
            return np.radians(azi1 - cog_0)

# Distance to domain boundaries
def l_a(length_0, lat_0, lon_0, lat_2, lon_2, cog_0, v_0):
    k_AD=10**(0.3591*np.log10(v_0)+0.0952)
    k_DT=10**(0.5441*np.log10(v_0)-0.0795)
    # course unknown circular domain
    if type(cog_0) != float:
        R_fore = R_aft = R_starb = R_port = (0.375*k_DT)*length_0
        cog_0 = 0
    elif type(v_0) != float:
        R_fore = R_aft = R_starb = R_port = (0.375*k_DT)*length_0
    else:
        R_fore = (1+1.34*np.sqrt(k_AD**2+(k_DT/2)**2))*length_0
        R_aft = 0.5*R_fore
        R_port = (0.375*k_DT)*length_0
        R_starb=R_port
    if (3 / 2) * np.pi < alpha(lon_0, lat_0, lon_2, lat_2, cog_0) and \
         alpha(lon_0, lat_0, lon_2, lat_2, cog_0) <= np.pi/2:
        l = (((1 + np.tan(alpha(lon_0, lat_0, lon_2, lat_2, cog_0)) ** 2) /
              ((1 / R_fore ** 2) +
               (np.tan(alpha(lon_0, lat_0, lon_2, lat_2, cog_0))
                                    ** 2) / R_starb ** 2)) ** 0.5)
    else:
        l = (((1 + np.tan(alpha(lon_0, lat_0, lon_2, lat_2, cog_0)) ** 2) /
              ((1 / R_aft ** 2)
```

```
                +(np.tan(alpha(lon_0, lat_0, lon_2, lat_2,cog_0)) ** 2)
                / R_starb ** 2)) ** 0.5)
    return l

# dir_x = 1 towards bow and 2 towards aft, dir_y = 1 towards
# starboard and 2 towards port
# When running the algorithm for the 2017 databse, True_center is removed.
def True_center(lat_0, lon_0, d_bow, d_aft, d_starb, d_port, cog_0):
    if type(cog_0) == str:
        return lat_0, lon_0
    elif type(d_bow) == str:
        return lat_0, lon_0
    elif type(d_aft) == str:
        return lat_0, lon_0
    elif type(d_starb) == str:
        d_starb = d_port = 0
    elif type(d_port) == str:
        d_port = d_starb = 0
    else:
        ##hack for 2013/2014 missing d_starb
        d_starb = d_port
        try:
            if d_bow > d_aft:
                dist_x = (d_bow + d_aft) / 2 - d_aft
                dir_x = 1
                if d_starb > d_port:
                    dist_y = (d_starb + d_port) / 2 - d_port
                    dir_y = 1
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
                elif d_starb < d_port:
                    dist_y = (d_starb + d_port) / 2 - d_port
```

```
                    dir_y = 2
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
                elif d_starb == d_port:
                    dir_y = 0
                    dist_y = 0
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
            elif d_bow < d_aft:
                dist_x = (d_bow + d_aft) / 2 - d_bow
                dir_x = 2
                if d_starb > d_port:
                    dist_y = (d_starb + d_port) / 2 - d_port
                    dir_y = 1
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
                elif d_starb < d_port:
                    dist_y = (d_starb + d_port) / 2 - d_port
                    dir_y = 2
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
                elif d_starb == d_port:
                    dir_y = 0
                    dist_y = 0
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
            elif d_bow == d_aft:
                dist_x = (d_bow + d_aft) / 2
                dir_x = 0
                if d_starb > d_port:
                    dist_y = (d_starb + d_port) / 2 - d_port
                    dir_y = 1
                    s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
                elif d_starb < d_port:
                    dist_y = (d_starb + d_port) / 2 - d_port
                    dir_y = 2
```

```
        s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
    elif d_starb == d_port:
        dir_y = 0
        dist_y = 0
        s_12 = np.sqrt(dist_x ** 2 + dist_y ** 2)
if dir_x == 0 and dir_y == 0:
    t_lon_0 = lon_0
    t_lat_0 = lat_0
elif dir_x == 1 and dir_y == 0:
    vector = Geodesic.WGS84.Direct\
    (lat_0, lon_0, cog_0, s_12, outmask=1929)
    t_lon_0 = vector['lon2']
    t_lat_0 = vector['lat2']
elif dir_x == 2 and dir_y == 0:
    vector = Geodesic.WGS84.Direct\
    (lat_0, lon_0, (cog_0 + 180), s_12, outmask=1929)
    t_lon_0 = vector['lon2']
    t_lat_0 = vector['lat2']
elif dir_x == 0 and dir_y == 1:
    vector = Geodesic.WGS84.Direct\
    (lat_0, lon_0, (cog_0 + 90), s_12, outmask=1929)
    t_lon_0 = vector['lon2']
    t_lat_0 = vector['lat2']
elif dir_x == 0 and dir_y == 2:
    vector = Geodesic.WGS84.Direct\
    (lat_0, lon_0, (cog_0 - 90), s_12, outmask=1929)
    t_lon_0 = vector['lon2']
    t_lat_0 = vector['lat2']
elif dir_x == 1 and dir_y == 1:
    vector = Geodesic.WGS84.Direct\
    (lat_0, lon_0, (cog_0 + np.degrees
```

```
                              (np.arctan(dist_y / dist_x))),\
               s_12, outmask=1929)
              t_lon_0 = vector['lon2']
              t_lat_0 = vector['lat2']
          elif dir_x == 1 and dir_y == 2:
              vector = Geodesic.WGS84.Direct(lat_0, lon_0, (cog_0 + \
              (360 - np.degrees(np.arctan(dist_y / dist_x)))),
                                              s_12, outmask=1929)
              t_lon_0 = vector['lon2']
              t_lat_0 = vector['lat2']
          elif dir_x == 2 and dir_y == 2:
              vector = Geodesic.WGS84.Direct(lat_0, lon_0, (cog_0 + \
              (180 + np.degrees(np.arctan(dist_y / dist_x)))),
                                              s_12, outmask=1929)
              t_lon_0 = vector['lon2']
              t_lat_0 = vector['lat2']
          elif dir_x == 2 and dir_y == 1:
              vector = Geodesic.WGS84.Direct(lat_0, lon_0, (cog_0 + \
              (180 - np.degrees(np.arctan(dist_y / dist_x)))),
                                              s_12, outmask=1929)
              t_lon_0 = vector['lon2']
              t_lat_0 = vector['lat2']
          return (t_lat_0, t_lon_0)
      # if dimensions are missing
      except:
          return lat_0, lon_0

def violations(i):
    print(i)
    if i.date().day == 1:
        d = i + timedelta(days=3)
```

```
elif i.date().day == 4:
    d = i + timedelta(days=3)
elif i.date().day == 7:
    d = i + timedelta(days=3)
elif i.date().day == 10:
    d = i + timedelta(days=3)
elif i.date().day == 13:
    d = i + timedelta(days=3)
elif i.date().day == 16:
    d = i + timedelta(days=3)
elif i.date().day == 19:
    d = i + timedelta(days=3)
elif i.date().day == 22:
    d = i + timedelta(days=3)
elif i.date().day == 25:
    d = i + timedelta(days=3)
elif i.date().day == 28:
    d = i + timedelta(days=3)
elif i.date().day == 31:
    # first of next month
    d = i + relativedelta(months=+1, day=1)
multiprocessing.Lock().acquire()

conn = lite.connect(f'2017.db')
c = conn.cursor()
# select all entries in one month starting with date i
c.execute(f"SELECT * FROM AIS_2017a WHERE length > 1 and length < 400 \
and sog > 1 and sog < 40 and Time >= '{i}' and Time < '{d}' \
order by Time;")
entries = c.fetchall()
```

```
for entry_0 in range(len(entries)):
    timestamp_0 = datetime.strptime(entries[entry_0][0],
                                    "%Y-%m-%d %H:%M:%S")
    mmsi_0 = entries[entry_0][1]
    v_0 = entries[entry_0][2]
    lon_0 = entries[entry_0][3]
    lat_0 = entries[entry_0][4]
    cog_0 = entries[entry_0][5]
    length_0 = entries[entry_0][6]
    for entry_2 in range((entry_0 + 1), len(entries)):
        timestamp_2 = datetime.strptime(entries[entry_0][0], \
                                        "%Y-%m-%d %H:%M:%S")
        # only cheak ais transmissions at the same time instance
        if timestamp_2 > timestamp_0:
            break
        t_lat_0 = lat_0
        t_lon_0 = lon_0
        mmsi_2 = entries[entry_2][1]
        v_2 = entries[entry_2][2]
        lon_2 = entries[entry_2][3]
        lat_2 = entries[entry_2][4]
        cog_2 = entries[entry_2][5]
        length_2 = entries[entry_2][6]
        t_lat_2 = lat_2
        t_lon_2 = lon_2
        # domain length in direction from ship 0 to ship 2 [meter]
        dom_len_0_2 = l_a(length_0, t_lat_0, t_lon_0, t_lat_2,
                          t_lon_2, cog_0, v_0)
        #print(dom_len_0_2)
        # domain length in direction from ship 2 to ship 0 [meter]
        dom_len_2_0 = l_a(length_2, t_lat_2, t_lon_2, t_lat_0,
```

```
                  t_lon_0, cog_2, v_2)
#            print(dom_len_2_0)
# distance center to center [meter]
distance_center = Geodesic.WGS84.Inverse\
(t_lat_0, t_lon_0, t_lat_2, t_lon_2)['s12']
#print(distance_center)
# if distance between true ship centers are less than
#safety domain of ship_0
if dom_len_0_2 > distance_center and mmsi_0 != mmsi_2:
    if mmsi_0 < mmsi_2:
        if dom_len_2_0 < distance_center:
            Violated_domain = mmsi_0
        else:
            Violated_domain = 2
        ID = int(str(mmsi_0) + str(mmsi_2))
        domain_overlap_0_2 = distance_center - dom_len_0_2
        domain_overlap_2_0 = distance_center - dom_len_2_0
        fields = [ID, timestamp_0, mmsi_0, mmsi_2,
                  Violated_domain, distance_center,
                  domain_overlap_0_2, domain_overlap_2_0, v_0,
                  v_2, cog_0, cog_2, t_lat_0, t_lon_0, t_lat_2,
                  t_lon_2, length_0, length_2]
        multiprocessing.Lock().acquire()
        with open(csvfile, 'a') as f:
            writer = csv.writer(f)
            writer.writerow(fields)
    else:
        if dom_len_2_0 < distance_center:
            Violated_domain = mmsi_0
        else:
            Violated_domain = 2
```

```
                    ID = int(str(mmsi_2) + str(mmsi_0))
                    domain_overlap_0_2 = distance_center - dom_len_0_2
                    domain_overlap_2_0 = distance_center - dom_len_2_0
                    fields = [ID, timestamp_0, mmsi_2, mmsi_0, Violated_domain,
                              distance_center,
                              domain_overlap_2_0, domain_overlap_0_2, v_2, v_0,
                              cog_2, cog_0, t_lat_2, t_lon_2, t_lat_0,
                              t_lon_0, length_2, length_0]
                    multiprocessing.Lock().acquire()
                    with open(csvfile, 'a') as f:
                        writer = csv.writer(f)
                        writer.writerow(fields)
                # if distance between true ship centers are less than safety
                #domain of ship_2
            elif dom_len_2_0 > distance_center and mmsi_0 != mmsi_2:
                if mmsi_0 < mmsi_2:
                    if dom_len_0_2 < distance_center:
                        Violated_domain = mmsi_2
                    else:
                        Violated_domain = 2
                    ID = int(str(mmsi_0) + str(mmsi_2))
                    domain_overlap_0_2 = distance_center - dom_len_0_2
                    domain_overlap_2_0 = distance_center - dom_len_2_0
                    fields = [ID, timestamp_0, mmsi_0, mmsi_2, Violated_domain,
                              distance_center, \
                              domain_overlap_0_2, domain_overlap_2_0, v_0, v_2,
                              cog_0, cog_2, t_lat_0, t_lon_0, t_lat_2, \
                              t_lon_2, length_0, length_2]
                    multiprocessing.Lock().acquire()
                    with open(csvfile, 'a') as f:
                        writer = csv.writer(f)
```

```
                        writer.writerow(fields)
                else:
                    if dom_len_0_2 < distance_center:
                        Violated_domain = mmsi_2
                    else:
                        Violated_domain = 2
                    ID = int(str(mmsi_2) + str(mmsi_0))
                    domain_overlap_0_2 = distance_center - dom_len_0_2
                    domain_overlap_2_0 = distance_center - dom_len_2_0
                    fields = [ID, timestamp_0, mmsi_2, mmsi_0, Violated_domain,
                              distance_center,
                              domain_overlap_2_0, domain_overlap_0_2, v_2, v_0,
                              cog_2, cog_0, t_lat_2, t_lon_2, t_lat_0,
                              t_lon_0, length_2, length_0]
                    multiprocessing.Lock().acquire()
                    with open(csvfile, 'a') as f:
                        writer = csv.writer(f)
                        writer.writerow(fields)

startyear = 2017
startmonth = 8
endyear = 2017
endmonth = 8
csvfile = f"NM.csv"

d_1 = [datetime(m // 12, m % 12 + 1, 1) for m \
       in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_4 = [datetime(m // 12, m % 12 + 1, 4) for m \
       in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_7 = [datetime(m // 12, m % 12 + 1, 7) for m \
       in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
```

```
d_10 = [datetime(m // 12, m % 12 + 1, 10) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_13 = [datetime(m // 12, m % 12 + 1, 13) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_16 = [datetime(m // 12, m % 12 + 1, 16) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_19 = [datetime(m // 12, m % 12 + 1, 19) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_22 = [datetime(m // 12, m % 12 + 1, 22) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_25 = [datetime(m // 12, m % 12 + 1, 25) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_28 = [datetime(m // 12, m % 12 + 1, 28) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]
d_31 = [datetime(m // 12, m % 12 + 1, 28) for m \
        in range(startyear * 12 + startmonth - 1, endyear * 12 + endmonth)]

# creates list of dates
d_1.extend(d_4)
d_1.extend(d_7)
d_1.extend(d_10)
d_1.extend(d_13)
d_1.extend(d_16)
d_1.extend(d_19)
d_1.extend(d_22)
d_1.extend(d_25)
d_1.extend(d_28)
d_1.extend(d_31)
d_1.sort()

conn = lite.connect(f'2017.db')
```

```
c = conn.cursor()
c.execute("""CREATE TABLE IF NOT EXISTS Domain_violation_2 \
(ID INT,Time DATETIME, mmsi_0 INT,mmsi_2 INT, Violated_domain INTEGER, \
distance_center float, domain_overlap_0_2 float, domain_overlap_2_0 float,
v_0 float,v_2 float,cog_0 float, cog_2 float,t_lat_0 float,t_lon_0 float,\
t_lat_2 float,t_lon_2 float, length_0 float,length_2 float)""")
index1 = (f"CREATE INDEX IF NOT EXISTS mmsi_index_reindex ON AIS_2017a(mmsi);")
c.execute(index1)
index2 = (f"CREATE INDEX IF NOT EXISTS Time_index_reindex ON AIS_2017a(Time);")
c.execute(index2)
write_column_names = ['ID', 'timestamp_0', 'userid_0', 'userid_2',
                      'Violated_domain', 'distance_center',
                      'domain_overlap_0_2', 'domain_overlap_2_0',
                      'v_0', 'v_2', 'cog_0', 'cog_2', 't_lat_0',
                      't_lon_0', 't_lat_2', 't_lon_2', 'length_0',
                      'length_2']
with open(csvfile, 'a') as f:
    writer = csv.writer(f)
    writer.writerow(write_column_names)
par = Parallel(n_jobs=16, verbose=10)
do_something = delayed(violations)
par(do_something(i) for i in d_1)
```

```
import sqlite3 as lite
import numpy as np
import pandas as pd

# The erroneus data is removed:
databasepath = ('2017.db')
NM = "SELECT distinct ID, timestamp_0, userid_0, userid_2, \
Violated_domain, distance_center,\
domain_overlap_0_2, domain_overlap_2_0, v_0, v_2, cog_0, cog_2, t_lat_0, \
t_lon_0, t_lat_2, t_lon_2, length_0, length_2  FROM NM \
WHERE Violated_domain > 10000000"
con = lite.connect(databasepath)
with con:
        A = pd.read_sql_query(NM, con)

con.close()

A = A.drop_duplicates(subset='ID',keep='first')
A = A.drop_duplicates()
A.to_csv('NM1.csv', index=False)
```

A.6 Distribution of Overtaking-, Head-on- and Crossing Encounters

```
# Count types of encounters:

import sqlite3 as lite #sql
import numpy as np
import pandas as pd

databasepath = ('2017.db')
```

```
overtaking = "SELECT ID, t_lat_0, t_lon_0 FROM NM1 \
WHERE 292.5 <= (max(cog_0, cog_2)-min(cog_0, cog_2)) \
OR (max(cog_0, cog_2)-min(cog_0, cog_2)) <= 67.5"

HeadOn = "SELECT ID, t_lat_0, t_lon_0 FROM NM1 \
WHERE 165 < (max(cog_0, cog_2) - min(cog_0, cog_2)) \
and (max(cog_0, cog_2) - min(cog_0, cog_2)) < 195"

crossing = "SELECT ID, t_lat_2, t_lon_2 FROM NM1 \
WHERE (165 > (max(cog_0, cog_2) - min(cog_0, cog_2)) \
and (max(cog_0, cog_2) - min(cog_0, cog_2)) > 67.5) \
OR ((max(cog_0, cog_2) - min(cog_0, cog_2)) > 195 \
and (max(cog_0, cog_2) - min(cog_0, cog_2))< 292.5)"

HeadOn1 = "SELECT ID, t_lat_0, t_lon_0 FROM NM1 \
WHERE userid_0 = 258062000 and userid_2 = 304010253 "
overtaking1 = "SELECT ID, t_lat_0, t_lon_0 FROM NM1 \
WHERE userid_0 = 219016713 and userid_2 = 311059100 "
crossing1 = "SELECT ID, t_lat_0, t_lon_0 FROM NM1 \
WHERE userid_0 = 257398700 and userid_2 = 375103000"


con = lite.connect(databasepath)
with con:
    overtaking = pd.read_sql_query(overtaking,con)
    overtaking1= pd.read_sql_query(overtaking1,con)
    HeadOn = pd.read_sql_query(HeadOn, con)
    HeadOn1= pd.read_sql_query(HeadOn1, con)
    crossing = pd.read_sql_query(crossing, con)
    crossing1=pd.read_sql_query(crossing1, con)
con.close()
```

```
print(overtaking.count())
print(HeadOn.count())
print(crossing.count())
```

NTNU
Norwegian University of
Science and Technology