Doctoral theses at NTNU, 2022:259

Trym Tengesdal

Risk-based Traffic Rules Compliant Collision Avoidance for Autonomous Ships

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Trym Tengesdal

Risk-based Traffic Rules Compliant Collision Avoidance for Autonomous Ships

Thesis for the Degree of Philosophiae Doctor

Trondheim, September 2022

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

© Trym Tengesdal

ISBN 978-82-326-5751-3 (printed ver.)
ISBN 978-82-326-6431-3 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:259

Printed by NTNU Grafisk senter

Summary

As the maritime traffic sector is expected to increase over the next years, effort should be invested into protocols and technology for ensuring safe and efficient voyage over the seas. Autonomous ships can here offer multiple benefits, such as increased safety and efficiency, lessened environmental impact and higher rates of reliability and consistency. Recently, there has been an increased focus on the development of such autonomous surface vehicle platforms, with examples such as Yara Birkeland and Zeabuz leading the way in Norway. For this development to be successful, there is a need for robust autonomy systems onboard the ship, which can make intelligent decisions in reasonable time in order to avoid collision and adhere to the International Regulations for Preventing Collisions at Sea (COLREGS), when faced with a challenging and uncertain environment involving multiple grounding hazards and dynamic vessels or obstacles. In this setting, a robust collision avoidance (COLAV) system is vital for the safety and efficacy of the autonomous ship. Topics within this field has been the focus in this thesis.

During the PhD, two sampling-based Collision Probability Estimators (CPEs) were developed, one based on a combination of Monte Carlo Simulation (MCS) and a Kalman-filter, and the other based on the Cross-Entropy (CE) method. Both CPEs estimate the probability that a ship will collide with a nearby dynamic obstacle, by taking the kinematic uncertainty of the obstacle into account. For the first CPE, samples from the obstacle Probability Density Function (PDF) involving both position and velocity are used to estimate the probability, which is then filtered through the KF to obtain the final estimate. The KF makes it possible to reduce the statistical variance introduced by the MCS. However, the method struggles with estimating low collision probabilities, as the assumption of constant obstacle velocities in the sampling and the curse of dimensionality makes evident. This was solved through the CE-based CPE, which can adaptively converge towards the optimal density to sample from using iterative optimization by minimizing the Kullback-Leibler divergence between the optimal and the current importance densities. Because of the adaptivity, the CE-based estimator can obtain low variance collision probability estimates at reasonably low computational cost. Simulation studies are shown to verify the methods.

Simultaneously, the Probabilistic Scenario-based Model Predictive Control (PSB-MPC) was introduced, based on the original Scenario-based MPC (SB-MPC), as an aim towards taking situational uncertainty into account for COLAV. The PSB-MPC employs a CPE to have probabilistic risk assessment, by using collision probability estimates in its cost function, which increases the COLAV situational awareness. The PSB-MPC was further enhanced throughout the thesis work to also consider intention uncertainty for nearby dynamic obstacles, by evaluating the probabilities of multiple different maneuvering scenarios for each obstacle. Ornstein-Uhlenbeck (OU) processes and Line-of-Sight (LOS) guidance based models have been used to improve its prediction scheme and allow for also taking into account this intention uncertainty. Simulation results are shown throughout the thesis chapters to prove that the method incrementally gets more and more viable for use as a means for robust COLAV. This is also evident after a parallelized and efficient implementation of the algorithm was developed, which facilitates both static and dynamic obstacle avoidance through the use of a Graphical Processing Unit (GPU) in the MPC cost evaluation. In the last part of the thesis work, the GPU-based PSB-MPC with a Dynamic Bayesian Net (DBN) for intention inference is validated in full-scale experiments, which demonstrates the viability of the method.

Lastly, a new approach to joint vessel destination inference and long-term kinematics prediction for dynamic obstacles was developed. The method uses a maritime graph in combination with a piecewise OU process and a so-called bridge model, in a Bayesian inference setting. The piecewise OU process utilizes maritime traffic pattern information through the maritime graph, for more accurate predictions of typical vessel trajectories along common sea lanes. After the vessel has reached the end of its trajectory along the maritime graph, the bridge model enables the continued vessel prediction to converge towards the considered destinations. It is demonstrated using historical Automatic Identification System (AIS) data that the method performs better than current state-of-the-art in terms of both destination inference and prediction quality. Improved vessel predictions, in addition to information on their most likely goal destinations, is valuable for ship autonomy applications, and the next step will be to also incorporate this in risk-based COLAV systems.

The thesis work has contributed to the progress of research within robust COLAV systems, for both the situational awareness and decision making aspects of the topic. Still, there are considerable challenges to solve. How to tune COLAV systems such as the PSB-MPC, with all involved modules, for use in a large variety of situations with different geographies and dynamic obstacle configurations is highly non-trivial. It will here be important to investigate methods for making such systems adaptive and capable of learning from experience, in order to properly adopt them on maritime vessels.

Samandrag

Ein ventar at den maritime sektoren vil auke i dei komande åra. Det bør derfor investerast i protokollar og teknologi for å gjere sjøtransport både sikrare og meir effektiv. Her kan autonome skip tilby fleire fordelar, som auka sikkerheit og effektivitet, mindre miljømessig påverknad samt ein i større grad påliteleg og repeterbar oppførsel. I det siste har det vore auka fokus på utvikling av slike platformar for autonome overflatefartøy, med eksempel som Yara Birkeland og Zeabus i Noreg. For at denne utviklinga skal vere vellukka, trengst det robuste autonomisystem på skipet, som kan ta intelligente avgjersler i rimeleg tid for å unngå kollisjon med andre båtar. Eit utfordrande og usikkert miljø med fleire grunningsfarar og dynamiske hindringar krev også at ein føyer seg etter det internasjonale regelverket for å forhindre kollisjon på sjøen (COLREGS). I denne samanhengen så er eit robust kollisjonsunngåingssystem viktig for sikkerheita og effektiviteten til det autonome skipet. Emne innan dette feltet har vore fokuset i denne avhandlinga.

I løpet av PhD-en blei to samplingsbaserte kollisjonssannsynsestimatorar utvikla. Den første er basert på ein kombinasjon av Monte Carlo-simulering og eit Kalmanfilter, medan den andre er basert på kryssentropi-metoden. Begge estimerer sannsynet for at eit skip vil kollidere med ei nærliggande dynamisk hindring, ved å ta omsyn til den kinematiske usikkerheiten til hindringa. For den første estimatoren, blir sampel av både posisjon og hastigheit frå sannsynstettleiken til hindringa brukt til å estimere sannsynet, som vidare blir filtrert gjennom Kalmanfilteret for å få det endelege estimatet. Kalmanfilteret gjer det mogleg å redusere den statistiske variansen som kjem frå bruk av Monte Carlo-simulering. På den andre sida slit metoden med å estimere låge kollisjonssannsyn, på grunn av at ein i samplinga reknar hastigheiten som konstant, samt den såkalla dimensjonalitetsforbanninga. Dette blei løyst gjennom bruk av den kryssentropibaserte estimatoren, som på ein adaptiv måte kan konvergere mot den optimale tettleiken å sample frå, ved bruk av iterativ optimalisering for å minimalisere Kullback-Leibler-divergensen mellom den optimale og nåverande viktigheitstettleiken. På grunn av adaptiviteten kan den andre estimatoren oppnå låg varians på kollisjonssannsynsestimata ved liten reknemessig kostnad. Simuleringsstudiar er her vist for å verifisera dei to estimatorane.

Samtidig blei den probabilistisk scenariobaserte modellprediktive regulatoren (PSB-MPC) introdusert, basert på den opprinnelege scenariobaserte MPC-en (SB-MPC), med målet om å kunne ta omsyn til situasjonsspesifikk usikkerheit for kollisjonsunngåing. PSB-MPC-en bruker ein kollisjonssannsynsestimator for å oppnå probabilistisk risikovurdering der kollisjonssannsynsestimata inngår i kostnadsfunksjonen, noko som igjen aukar situasjonsforståinga til systemet. Algoritmen blei vidare forbetra gjennom arbeidet i doktorgraden til å også kunne ta omsyn til intensjonsusikkerheit for nærliggande dynamiske hindringar. Dette var mogleg ved å evaluere sannsyna for fleire ulike manøverscenariar for kvar hindring. Ornstein-Uhlenbeck (OU)-prosessen og siktelinjemetoden blei brukt til å forbetre prediksjonsdelen av MPC-en, for å kunne ta omsyn til intensjonsusikkerheit. Simuleringsresultat er vist for å demonstrere at metoden gradvis gir meir robust kollisjonsunngåing. Dette blir endå meir tydeleg etter at ein parallellisert og effektiv implementasjon av algoritmen blei utvikla. Denne innlemma kollisjonsunngåing med omsyn til både statiske og dynamiske hindringar ved å ta i bruk ein grafikkprosessor i kostnadsevalueringa til MPC-en. Den paralleliserte versjonen av PSB-MPC-algorithmen blei til sist testa experimentelt ilag med eit dynamisk Bayesiansk nettverk brukt til intensjonsinferens for dynamiske hindringar, som demonstrerte at algoritmen kan brukast i praksis.

Sist blei ein ny måte å samtidig estimere eit skip sin planlagte destinasjon og langtidsprediksjon mot denne, utvikla. Metoden bruker ein maritim graf kombinert med ein stykkevis OU-prosess og ein såkalla brumodell i ein Bayesiansk setting. Den stykkevise OU-prosessen utnyttar informasjon om maritime trafikkmønster gjennom den maritime grafen, for meir nøyaktige prediksjonar av typiske skipsbanar langs vanlege sjøvegar. Etter at skipet har nådd enden av banen langs den maritime grafen, gjer brumodellen det mogleg å få baneprediksjonen til å konvergere mot vurderte destinasjonar. Ved bruk av historisk data frå eit automatisk identifikasjons system (AIS), blir det demonstrert at metoden yter betre enn den nåverande forskningsfronten på emnet, med tanke på destinasjonsinferens og kvaliteten på skipsprediksjonane.

Avhandlinga har bidratt til framgang på forskninga innan robuste system for kollisjonsunngåing, for både situasjonsforståing- og beslutningstakingsaspektet på feltet. Det finst derimot framleis betydelege utfordringar å løyse. Korleis ein skal justere parametrane for slike system, som for eksempel PSB-MPC-algoritmen, med alle involverte modular til bruk i eit stort spekter av situasjonar med forskjellig geografi og konfigurasjonar av dynamiske hindringar, er ikkje trivielt. Det vil her vere viktig å utforske moglege metodar for å gjere slike system adaptive og i stand til å lære frå erfaring, for å kunne ta dei i bruk på maritime fartøy.

Contents

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor (PhD) at the Norwegian University of Science and Technology (NTNU), Trondheim.

The work presented has been conducted at the Center for Autonomous Marine Operations and Systems (AMOS) at the Department of Engineering Cybernetics (ITK), NTNU. The project supervisor has been Professor Tor Arne Johansen and co-supervisor Associate Professor Edmund Brekke, both from the Department of Engineering Cybernetics. The work was supported by the Research Council of Norway (grant no. 223254) at AMOS.

Acknowledgements

I am grateful for the opportunity to take a PhD at ITK, with a big thanks to the support of my main supervisor Tor Arne Johansen and co-supervisor Edmund Førland Brekke throughout the PhD. Tor Arne, I am grateful for all the discussions and the advice I have received from you, and also the freedom to pursue my own ideas without constraints or pressure. Not to mention your speed of light email support when I required advice or feedback, especially when publication deadlines came close. For my co-supervisor Edmund Førland Brekke I also express my largest gratitude. You have given me the perfect mix of humble advice and feedback on my work, and motivation when I needed it. Without your connections, I would not have gotten the chance to go to La Spezia for my research stay. Both of you have been humble and non-constraining in your approach to supervision, which I have appreciated a lot.

I have had the enjoyment of working with a number of intelligent people throughout my PhD. I would like to thank Sverre Velten Rothmund for our fruitful collaboration on dynamic obstacle intent inference. Even though we had some heated discussions at times, your critical view and considerate opinions on our work has been very much appreciated and quality raising. I would also like to thank Erlend Andreas Basso for the collaboration we had on the road towards experimental testing of the collision avoidance system developed in this thesis. Your ideas, opinions and experience in the Robotic Operating System and hardware have been appreciated. I also want to thank Haakon Robinson and Bjørn-Olav Eriksen for good advice and help on relevant subjects.

The PhD would never have been the same without the outstanding work environment at ITK, where I am grateful for all the friends I have made during my three year period. You all know who you are. I would especially thank my office mates Haakon Robinson, Asbjørn Espe and Andreas Våge, and Erlend Torje Berg Lundby in the last part, for an unconstrained, weird and joyfilled local workspace where no inhibitions has existed. It has been a zoo which I have enjoyed being a part of.

A big thanks is in order, to my parents Eli and Morten, for your constant support and motivation throughout. You have always believed in me, and for that I am very grateful. Also I am grateful for my brothers Njål and Stig. You partially motivated me to do this PhD as I could not be the one loser brother without a doctoral title. Last but not least, I am grateful for Silje, for being there and always keeping me on my toes.

August 2022, Trondheim
Trym

Abbreviations

AGT Approximate Ground Truth 84, 86

AI Artificial Intelligence 5

AIS Automatic Indentification System 23, 56, 71, 127, 163 f., 166 f., 169, 173 f.

AUV Autonomous Underwater Vehicle 6

CARACaS Control Architecture for Robotic Agent Command and Sensing 12

CC COLREGS Compliance 59, 67, 69, 71

CDF Cumulative Density Function 75

CE Cross Entropy 21, 73, 77, 80–83, 85 f., 88, 91–94, 110, 113, 137, 158

COG Course Over Ground 166, 174

COLAV Collision Avoidance 8, 11–17, 19 f., 22 f., 34 ff., 39, 41 ff., 49 f., 53, 55 ff., 62, 73–77, 85, 92 ff., 101–105, 112, 114 f., 120, 122, 127–132, 134, 149, 151, 154, 156 f., 161, 163, 177–180

COLREGS Convention on the International Regulations for Preventing Collision at Sea 8, 10–17, 19 f., 22, 24, 37 f., 42 f., 46, 50, 53, 56–59, 61 f., 66 f., 71, 76, 91–94, 101–104, 109 ff., 113, 115, 121 ff., 127–131, 134–141, 144–153, 155–159, 161, 177, 179 f.

CPA Closest Point of Approach 13, 45 f., 48, 62, 64, 74, 83, 94, 103, 129, 138, 146 ff.

CPE Collision Probability Estimator i, 17, 19 ff., 55, 63, 74, 76 ff., 91–94, 177–180

CPU Central Processing Unit 113 f., 116 ff., 120

CT Constant Turn 15

CV Constant Velocity 14 ff., 36, 44, 46, 49, 103, 129, 134, 164, 174, 178

DARPA Defence Advanced Research Projects Agency 6

DBN Dynamic Bayesian Network 22, 24, 127, 130, 140 f., 148, 157, 161, 178 f.

DOF Degrees of Freedom 32, 43

DOII Dynamic Obstacle Intention Inference 140, 150

DP Dynamic Positioning 6, 134, 140, 159

DW Dynamic Window 12

EKF Extended Kalman Filter 34

EM Expectation Maximization . . . 164

EMSA European Maritime Safety Agency . . . 6

ENC Electronic Navigational Chart . . . 17, 21, 104, 132

ERV Equilibrium Reverting Velocity . . . 23, 165 f., 169, 172, 174

FCS Finite Control Set . . . 34, 36

FCS-MPC Finite Control Set Model Predictive Control . . . 34, 111

GMM Gaussian Mixture Model . . . 164

GNC Guidance Navigation and Control . . . 14, 26, 39, 180

GNSS Global Navigation Satelite Systems . . . 34, 131, 133 f., 158

GPS Global Positioning System . . . 27, 134

GPU Graphical Processing Unit . . . 21, 103, 112–118, 120, 122, 127, 130, 134 f.

IMM Interacting Multiple Model . . . 62

IMO International Maritime Organization . . . 8

INS Inertial Navigation System . . . 34

IS Importance Sampling . . . 25 f., 79

KF Kalman Filter 14, 19, 28, 41, 43–47, 49 f., 56, 62, 67, 75, 92, 129, 134, 169 f., 172, 174

LOS Line of Sight . . . 33, 36, 44, 59, 104, 106 f., 136

LSTM Long Short-Term Memory . . . 164

LTI Linear Time Invariant . . . 165

MC Monte Carlo . . . 13, 15, 25, 92, 103, 114, 128

MCS Monte Carlo Simulation . . . 19, 25, 42 f., 45 ff., 56, 62, 75 f., 82, 84 ff.

MINLP Mixed Integer Nonlinear Programming . . . 111

MPC Model Predictive Control . 13 f., 19, 21, 32, 34 f., 39, 43 f., 48 f., 62 ff., 66 f., 71, 91 ff., 101–104, 106, 108, 111–118, 122, 129 f., 135 f., 140, 151, 178

MSA Maritime Situational Awareness . . . 23

MTT Multi Target Tracking . . . 41

NCDM Neighbor Course Distribution Method . . . 164

NED North-East-Down . . . 27

NIS Normalized Innovation error Squared . . . 46

NLP Nonlinear Programming . . . 11

NMPC Nonlinear Model Predictive Control . . . 13, 103, 129

NTNU Norwegian University of Science and Technology . . . 6, 132

NTRIP Network Transport of RTCM data over IP . . . 133

OU Ornstein-Uhlenbeck 20, 23, 29, 31, 56, 63 f., 71, 77, 85, 106 f., 164, 166 f., 169, 174, 178

PD	Proportional-Derivative	44
PDAF	Probabilistic Data Association Filter	28, 41
PDF	Probability Density Function	19, 42, 44, 46, 76–80, 82 ff., 86
PED	Prediction Error Density	172 f.
PSB-MPC	Probabilistic Scenario-based Model Predictive Control	17, 19–23, 43, 48 ff., 52 f., 55 ff., 63–67, 69, 71, 76 f., 91–94, 102–106, 108–111, 113–116, 120, 122 f., 127, 130 ff., 134–137, 140, 147, 151, 155–158, 161, 177–180
RDP	Ramer-Douglas-Peucker	104, 108, 113, 118 ff.
RL	Reinforcement Learning	14
RMSE	Root Mean Square Error	84, 86, 88
RNN	Recurrent Neural Network	164
ROS	Robotic Operating System	134
RRT	Rapidly-exploring Random Tree	12, 14, 103, 129
RTK	Real-Time Kinematic	133
SB-MPC	Scenario-based Model Predictive Control	13, 19, 34 ff., 39, 43, 48 ff., 53, 66 f., 69, 76, 91 ff., 102, 130, 138
SDE	Stochastic Differential Equation	30 f., 166
SFI	Centre for Research-based Innovation	6
SOG	Speed Over Ground	166, 174
SPNS	Single Point Neighbor Search	164
USV	Unmanned Surface Vehicle	43
UTM	Universal Transvere Mercator	26
VO	Velocity Obstacle	11, 129 f.
VRS	Virtual Reference Station	133, 158
WGS	World Geodetic System	27

Chapter 1

Introduction

1.1 Motivation

The modern society we know today is dependent on automated technology or *automation.* Automation is found everywhere, ranging from the flushing mechanism in our toilet, display light adjustment on the smart phone screen in our pocket to robotic assembly lines and Artificial Intelligence (AI) based chat bots for customer service. The common denominator of automation is the minimal human intervention in a task, as the meaning of the word is to be self-governing, self-acting or moving on its own. Automation was first taken into common use after Henry Ford established an automation department in his car factory in 1947. The origins of the word however goes millennia back to ancient Greece, where the similar word *automaton* was used by Homer in the Iliad to describe self-governing machines or *automata.*

It was here, in ancient times that the first known automated technology inventions originated, where e.g. the Greek Ctesibius developed automatic water level control using floats [1]. During the Industrial Revolution, centrifugal governors were introduced as a feedback mechanism to control the speed of steam engines [2]. In the latter part of the 20th century, the large advances in computer technology, especially during the Space race between the US and the Soviet Union, paved the way for modern automatic control systems in a diverse set of applications such as electrical power plants, oil platforms, aircraft autopilots and space satellites [3]. The introduction of computerized control made automation realistic in all branches of society, and revived the concept of *autonomy* or *autonomous systems* first commonly used in the 17th century. Autonomy is an extension of automation to describe an independent system, which can act, learn and adapt when faced with new situations and uncertain environments. Another interpretation of autonomy is the automation of tasks commonly performed by operators today. In recent years it is starting to become a part of daily conversation when considering the progress within topics such as self-driving cars [4], robotics [5] and drones [6].

In 1922, the first success within maritime automation emerged with the installation of Elmer Sperry's *gyropilot* on the ship Munargo [7]. The invention introduced automatic ship heading control through the use of a *gyrocompass* for feedback [8]. Only 12 years after the first permanent installation, over 400 gyropilots were in service. Since this stage, in tact with improved vessel propulsion systems and maneuverability, they have gradually gotten more advanced automatic control systems, exemplified by the development of the dynamic positioning (DP) control system for accurate station-keeping [9].

Research on maritime autonomy has over the years mostly been focused on autonomous underwater vehicles (AUVs), as the difficulty of direct vehicle control underwater is high due to limitations on communication between the vehicle and the human operator. However, this is changing today, as increasing effort is being put also on autonomous surface ship technology. This origins from challenges within the growing maritime sector, one of them being safety. The European Maritime Safety Agency (EMSA) reported a total number of 15481 causality events with a ship from 2014 to 2020 [10], where collision, contact and grounding constituted 43% of these. The consequences of such events can be detrimental, with human casualties, environmental and property damage being the result. Humans are reported to be the main causality factor over 75% of the time [11]–[13], which speaks the case for autonomous ships taking over many aspects of maritime traffic, in order to increase safety at sea. Aside the safety aspect, autonomous ships offer a number of benefits, including increased reliability and efficiency, and lower costs during operation [13].

The emphasis on developing autonomous ship technology today is shown through the increased effort in the maritime domain for testing and commercializing self-driving vessels, both from the governmental and industrial side. In 2016, the Defense Advanced Research Projects Agency (DARPA) launched *Sea Hunter*, an unmanned surface vehicle (USV) designed to detect and track submarines [14]. Together with the Finnish Finferries, Rolls-Royce demonstrated autonomous operation of the car ferry *Falco* in 2018, with the ferry shown in Figure 1.1. Around the same time, the other Finnish company Wärtsilä conducts successful autonomous docking and voyage with its solution onboard the Norwegian *Folgefonna* ferry. Later in 2020 Kongsberg Maritime also demonstrated autonomous car ferry operation on the *Bastø Fosen VI* ferry in the Oslofjord [15], and launches the zero emission Yara Birkeland autonomous cargo ship project [16]. NTNU has also taken initiative of experiments with autonomous ferry transport as a low cost alternative to manned bridges through the Milliampere vessel [17], shown in Figure 1.2. Furthermore, the university recently launched the Autoship Centre for Research based Innovation (SFI) [18] focused on taking a leading role in autonomous ship development the next eight years.

There is, however, still a lack of trust in self-governing vehicles, which is why none are arguably permanently deployed without any form of human supervision today. This is well argued, as autonomous systems must be able to work in a large variety of operating conditions. For such self-governing systems, challenges related to

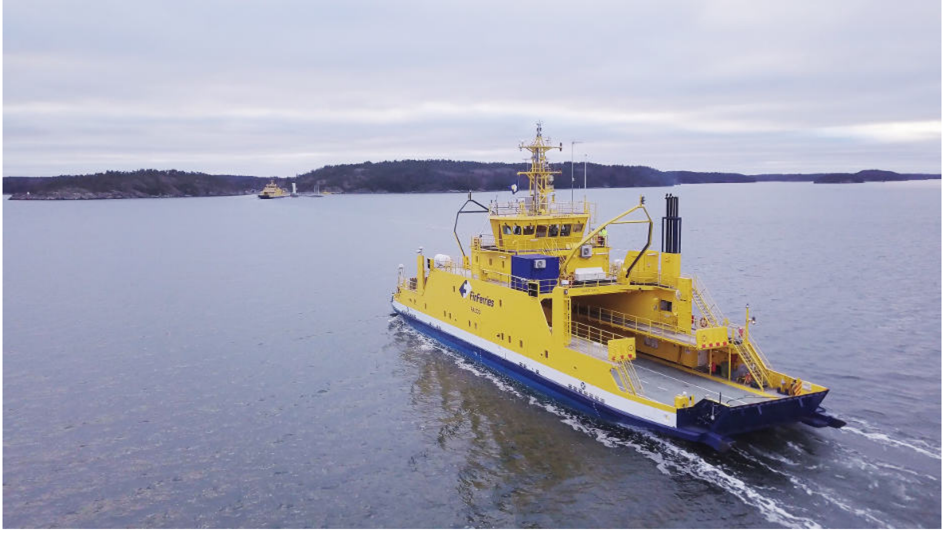


Figure 1.1: The Finnish car ferry Falco. Courtesy of The Engineer [19].

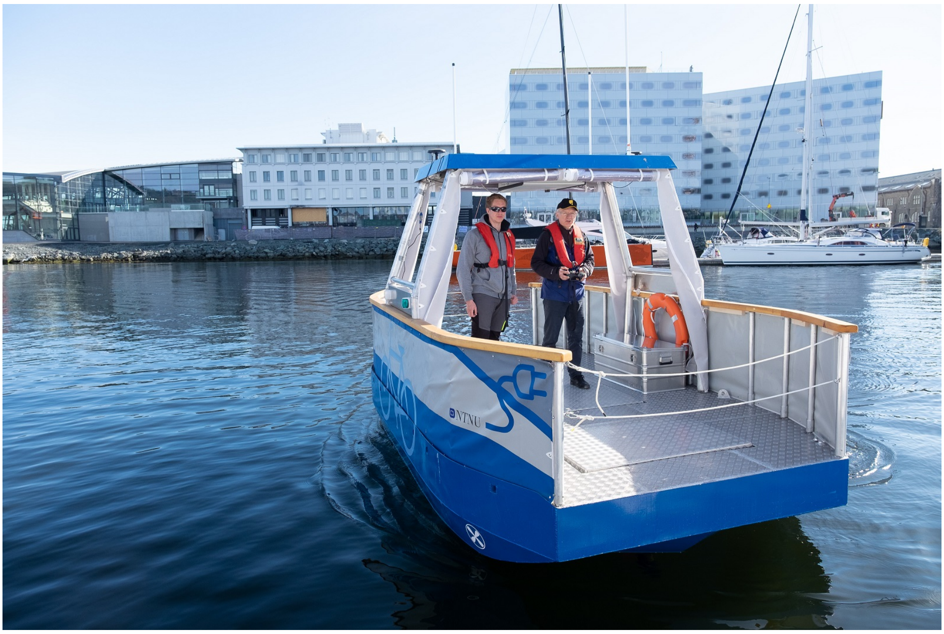


Figure 1.2: The autonomous ferry prototype Milliampere developed at NTNU. Courtesy of Kai Dragland [20].

cyber-security, self monitoring, fault tolerance and complex machinery need to be addressed in order to increase their thrustworthyness. In the context of autonomous ships, the sea is a challenging environment to operate in, as the ship must avoid collision with dynamic vessels, grounding hazards and other potentially harmful obstacles. Simultaneously, the ship should to adhere to the Convention on the International Regulations for Preventing Collision at Sea (COLREGS) [21]. In this setting, there exists a multitude of uncertain factors that the ship must deal with. One of the biggest uncertainties are associated with nearby vessels, where questions such as the following arises: Where is the other vessel is going? Will it adhere to the COLREGS? Has the vessel seen the autonomous ship? Furthermore, the sensor suite of the autonomous ship, used to detect and track nearby hazards, will never be able to perfectly detect and track the hazards in an environment, and the corresponding tracking uncertainty should also be taken into account. Thus, the autonomous ship needs adequate situational awareness in addition to robustness in its decision making algorithm, in order to tackle all sources of uncertainty while adhering to the COLREGS. A robust and deliberate COLAV system is needed for this, which has been the main research focus in this thesis. Here, the word deliberative refers to the COLAV algorithm planning efficient trajectories that adheres to the COLREGS and avoid collision well before risky situations occur. The thesis focus has been put specifically on how to capture uncertainties associated to nearby vessel intents and kinematics in a more informed collision risk assessment for autonomous ship collision avoidance. Furthermore, it has also been put focus on how to combine this risk assessment with automatic anti-grounding and probabilistic compliance to the COLREGS, all in a real-time feasible manner.

1.2 COLREGS

For car traffic there are fairly strict rule sets to follow, somewhat depending on the country one drives in. If these rules are violated, the consequences are human casualties, property and environmental damage. This is because of cars having strict constraints of following the set lanes on the road, being designed for high traffic density, leading to limited maneuvering space. For traffic at open sea, the case is often the opposite, where one can have large room for vessel maneuvering. Furthermore, sea voyage inland, in narrow waterways and near port areas give rise to specially challenging situations. The freedom of sea voyage gives many possibilities, but also challenges in determining how vessels will travel locally, especially in cases with other nearby vessels and land objects. An autonomous ship will greatly benefit from information on how nearby vessels intends to maneuver, both with regards to safety and efficiency.

The International Regulations for Preventing Collision at Sea were put forth by the International Maritime Organization (IMO) in 1972 [21], which helps in addressing the challenges with sea voyage by facilitating regulated sea traffic and to reduce the risk for collisions between maritime vessels. Many aspects of different existing rule sets were standardized by the introduction of COLREGS, which helped in removing many sources of inconsistencies often leading to incorrect maneuvering

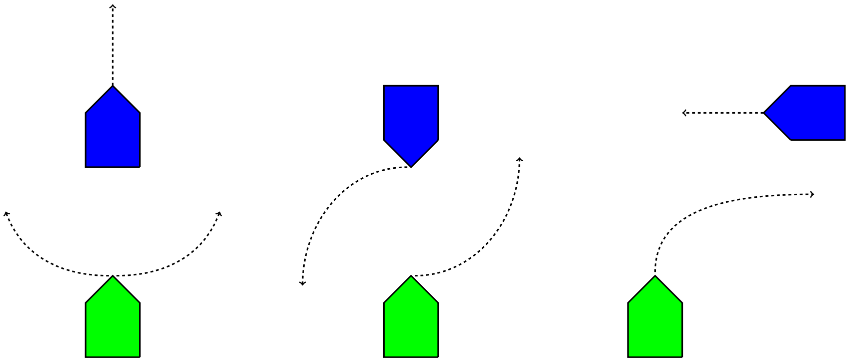


Figure 1.3: COLREGS situations. From left to right: Overtaking, head-on and crossing situations. The arrows indicate the correct behavior in each situation.

decisions and confusion at sea.

The COLREGS is divided into six parts (A to F) and has a total of 41 rules. Four Annexes detailing technical requirements for things such as ship lights and their shapes are also included. For collision avoidance at sea, Part B detailing rules on steering and sailing for power driven vessels should be considered. From this part, rules 6-10 and 13-18 are the most relevant. These are described below, with rules 13-15 illustrated in Figure 1.3.

Rule 6 *Safe speed*: Vessels shall travel with safe speeds in order to make maneuvering for collision avoidance easier and more effective, where one should take into account the current situational information such as environmental disturbances, visibility and traffic density.

Rule 7 *Risk of collision*: States that every vessel shall determine the if there is a risk of collision, based on available means and information from the current situation. It also explicitly states that assumptions shall not be made based on scanty or small amounts of information.

Rule 8 *Action to avoid collision*: Actions taken to avoid collision shall be readily observable for nearby vessels and taken in ample time, which implies that speed or course changes taken shall be large enough. Course changes should here be prioritized over speed changes for visibility, if enough free space is available.

Rule 9 *Narrow channels*: A vessel driving along a narrow channel or fairway must stay as near the starboard limit of the channel or fairway as is safe and practicable.

Rule 10 *Traffic separation schemes*: When using a traffic separation scheme, a vessel shall follow the general direction of the traffic lane it is associated with, keep clear of the separation line, and when leaving a lane, doing so at as small an angle to the general traffic direction as possible.

Rule 13 *Overtaking*: A vessel is classified as overtaking if coming up to another vessel from a direction more than 22.5 degrees abaft her beam. If this is the case, the overtaking vessel shall keep clear of the overtaken vessel.

Rule 14 *Head-on*: When two vessels meet on reciprocal or near reciprocal courses such that there is a risk of collision, each vessel shall change their course to starboard such that they pass each other with the other vessel on the port side.

Rule 15 *Crossing*: When two vessels are crossing with speed and course configuration such that there is a risk of collision, the vessel with the other on its starboard side is supposed to keep clear, and avoid crossing ahead of the other vessel if possible.

Rule 16 *Action by give-way vessel*: The vessel supposed to give-way shall if possible perform substantial actions early to keep well clear of the other vessel.

Rule 17 *Action by stand-on vessel*: The vessel with stand-on role shall nominally keep its course and speed, but should take action to avoid collision if the give-way vessel does not take appropriate action in order to comply with the rules. Furthermore, if the situation considered is crossing, the stand-on vessel shall if possible not alter its course to port when the other vessel is on its port side.

Rule 18 *Responsibilities between Vessels*: A power-driven vessel shall during voyage keep clear of:

(i) A vessel not under command
(ii) A vessel with restricted maneuvering capabilities
(iii) A fishing vessel
(iv) A sailing vessel

Of the listed rules, 6-7, 9-10 and 18 are included for completeness, which should be adhered to by a COLAV system in a fully functioning autonomous ship. In the literature, however, mostly rules 8 and 13-17 have been focused on.

Following these rules blindly in any type of situation is not sufficient, which was shown in [22]. Here, it was demonstrated that the existence of agreements between captains and local unwritten rules went contrary to the rules specified by COLREGS. Furthermore, COLREGS is open to disagreements making it unsafe to act only based on your own interpretation of the situation [23], [24]. This can be gauged by examining the list of rules described here, where 13-16 only involve two vessels and are vague with respect to quantifying how the speed and course of give-way vessels shall be decided.

Autonomous ships that are to operate at sea thus need to take into account the intentions of other vessels, when attempting to adhere to the COLREGS. For sea voyage, rigidly sticking to the rules can be dangerous and inefficient if the encountering vessel facing the autonomous ship violates them. Vessels involved in a COLREGS situation can have different interpretations on what rules to apply, and different inclinations towards following them. This makes it paramount to have adequate situational awareness for the autonomous ship with respect to inferring nearby obstacle ship intentions. One can then better assess the risk level in any hazardous situation in adherence with rule 7 and thus make more deliberate maneuvering decisions.

1.3 Literature Review

In this section, a brief historical overview on the results within the field of maritime COLAV is given, with the aim of providing context to the main research questions in this thesis. A more complete overview of the literature and proposed methods can be found in [25]–[31].

Research within maritime collision avoidance began after World War II in the 1960s [26], shortly after the newly emerged radar technology was also starting to be utilized on civilian vessels. At this stage and until the 1980s, studies were qualitative and focused on collision regulation interpretation, ship decision support and plot aid systems for officers on watch. An example is Mitrofanov [32], who developed an analogue computer which used manually input radar data to calculate a set of safe headings. The computation was based on a mathematical model using the involved vessel‘s speed and course, where the collision-free headings were represented as a non-shaded region on a heading display. Further ahead in the 1970s, the Sperry "collision avoidance system" also introduced interpretation of the radar data, to display predicted areas of danger for the officer [33].

From the academic side, more theoretical approaches to COLAV from within game theory [34]–[36], optimal control [37]–[39], and geometry-based models (the room-to-maneuvre principle) [40] were starting to emerge in the 1970s and 1980s. The latter approach was a precursor to the common Velocity Obstacle (VO) COLAV method [41], and was one of the first methods in the field to consider both heading and speed changes in COLAV. Furthermore, the game-theory based approach by Cannell [36] was to the author‘s knowledge the first COLAV study to also consider COLREGS. However, it was not until Dove and colleagues [42] proposed a method for collision avoidance at sea where the vessel could govern itself without human interaction, that the research expanded its focus from not only decision support systems but also to *automatic* COLAV for vessels.

More complex methods were proposed for COLAV during the 1990s and 2000s, using fuzzy logic [43], [44], Nonlinear Programming (NLP) [45], [46], genetic and evolutionary algorithms [47], [48], and articifial potential fields [49], thus advancing further the state-of-the-art. Sato [50] introduced a COLAV method which used radar and infrared imaging for measuring the course of nearby target vessels. Their approach demonstrated that such sensor technology can give improved situational awareness for collision avoidance.

A *sense-act* behaviour-based control architecture using multiobjective optimization for COLAV was proposed by Benjamin and Curcio in 2004 [51], and later demonstrated in 2006 as the first closed loop experimentally verified COLAV from academia [52]. They did however use communication between the vessels involved in the sea trials. For the method, each of the COLREGS rules 8 and 14-16 were captured by specific behaviours encompassed by their own objective functions. When multiple rules were active, the multiobjective optimization found the optimal weighting of the behaviours to decide which course and speed to apply for the vessel.

On-wards to the late 2000s, the concept of planning ahead in a *sense-plan-act* decomposition for deliberate and hybrid COLAV emerged [53]–[56]. Loe [55] developed a two-layered hybrid COLAV using A*-search and Rapidly-exploring Random Trees (RRTs) [57] as a top level global or deliberate planner, with the Dynamic Window (DW) algorithm [58] as a lower level local or reactive planner. The method was verified to comply with COLREGS rules 14-16. A three-layered approach was introduced by Casalino [56] in 2009, further decomposing the COLAV problem by having three layers with specialized focus on global static obstacle avoidance, local dynamic obstacle avoidance and local reactive static and dynamic obstacle avoidance, respectively. COLREGS were however not considered.

In 2014, Kuwata and colleagues developed the *sense-act* VO algorithm for maritime dynamic and static COLAV [41], which was experimentally verified to adhere to COLREGS rules 13-16. The method computes a feasible velocity set in which the own-ship does not collide with nearby vessels nor grounding hazards. COLREGS is taken into account by expanding the VO associated with an obstacle in specific directions to consider the relevant traffic rule. Kinematic uncertainty was handled by expanding the VO to account for a worst case scenario. Kuwata‘s COLAV was embedded into the Control Architecture for Robotic Agent Command and Sensing (CARACaS) autonomy suite developed at the NASA Jet Propulsion Laboratory [59], through the Autonomous Maritime Navigation (AMN) project [60]. A version of VO taking consideration of nearby vessel kinematic uncertainties were considered already in 2004 [61]. However, the avoidance behavior of the obstacles were assumed known, which will not be the case unless vessel-vessel communication is used.

Svec [62] introduced a two-layered approach with a deliberate lattice-based planner and a reactive COLREGS-compliant planner for COLAV in 2013. The method was demonstrated to also be COLREGS-compliant with rules 13-16. Their work was one of the first studies to consider probabilistic risk assessment in maritime COLAV, although limited details on the implementation nor methods used were given. A* search is also used by Blaich[63] and Schuster [64] on an occupancy grid to plan collision-free waypoints. The methods employ a non-symmetrical ship domain [65] in order to make the COLAV algorithm plan COLREGS-compliant routes. For [64], radar-based detection and tracking in COLAV was demonstrated in sea trials, although no convincing results with respect to COLREGS adherence were given. The work in [63] was later extended by Blaich in [66] to calculate occupancy probabilities for obstacles in two-dimensional space by using a numerical approximation, considering their kinematic uncertainty from tracking system information. As a large part of the contributions to maritime autonomy and COLAV came from the military side during the 2000s and 2010s [59], [60], [67], the studies [62], [63] was a breath of fresh air in terms of contributions to the topic from the academia again.

Agrawal and Dolan [68] came up with a field-test verified A* search path planner for COLAV in 2015, which attempts to find a COLREGS-compliant and collision-free path with respect to both dynamic and static obstacles. To predict nearby dynamic obstacle paths, the planner employs Monte-Carlo (MC) simulation using fuzzy logic and the path history of the obstacle to find a set of probable paths, where the most probable one is considered for collision avoidance. However, the method does not consider the prediction uncertainty associated with dynamic obstacles. With the deliberate aspects of COLAV gaining increased focus in the 2000s and 2010s, [68] were one of the studies not conforming to the common assumption of constant speed and course for nearby dynamic obstacles in the prediction horizon.

Predictions allowing maneuvering for dynamic obstacles were also utilized in [69], where A* search was applied to collision-free path planning. An intention based motion model was used for dynamic obstacles, using historic data in order to classify a vessel as COLREGS-compliant or not, and which also incorporated reactive COLAV. Details on this model were again however not given. This work was to the author‘s knowledge one of the first to consider dynamic obstacle intentions in maritime COLAV.

Johansen, Perez and Cristofaro [70] introduced the Scenario-based Model Predictive Control (SB-MPC) for COLAV in 2016, which employs brute force cost evaluation of a discrete set of own-ship trajectories or scenarios to find an optimal surge and course modification. These modifications are then used to alter the input references for the ship autopilot. In the study, the motion of dynamic obstacles are predicted with a deterministic CV model. However, the COLAV is versatile in that the MPC problem can be non-convex, nonlinear and mixed integer with constraints, as it does not depend on a complex solver for the solution. This gives freedom in parameterizing the problem, selecting cost functions and prediction models for the own-ship and obstacles. The algorithm has been experimentally verified in several closed loop sea trials [71]–[73] to be compliant with COLREGS rules 13-17. Other MPC-based approaches to COLAV in the later years include Abdelaal [74]–[76], which developed a trajectory tracking Nonlinear MPC for COLAV with dynamic and static obstacles as inequality constraints. How the NMPC scales with increasing situational information is not discussed however.

A global and local path planner was developed by Candeloro and colleagues [77] in 2017, using Voronoi Diagrams to generate a set of static obstacle collision-free waypoints, from where a continuous path is generated using Fermat‘s Spiral. The method considers local replanning windows for taking detected dynamic and static obstacles into account, and predicts dynamic obstacle motion with the Constant Velocity (CV) model [78]. A convex hull representing the dynamic obstacle uncertainty up until time to Closest Point of Approach (CPA) is created from using the position estimates and error covariances from a Kalman filter (KF), which is then regarded as an area to avoid in the planner. This may however be overly conservative, due to the unrealistic uncertainty growth in the CV model [79]. How the local replanning run-time scales with increasing window size, dynamic and static obstacles is not considered.

Deep Reinforcement Learning (RL) obstacle avoidance for COLAV was proposed by Cheng in 2018 [80], where obstacle collisions were formulated as negative rewards in the RL agent, although no COLREGS was considered. RL or deep learning approaches were also proposed in [81], [82]. Also in 2018, Chiang and Tapia [83] introduced a static and dynamic obstacle considerate path planner with COLREGS-compliant COLAV based on RRTs, where a joint simulator is used to predict both the own-ship and dynamic obstacle motion. Potential fields are used in the prediction to ensure that all the vessels have collision-free paths with respect to each other and static obstacles. The method is shown to have beneficial run-times feasible for real-time. However, the underlying assumption in the prediction is that ships will always perform deterministic COLREGS-compliant maneuvers if possible, which is not necessarily true in practice. Zaccone and colleagues [84] also developed a RRT-based COLAV, with the main aim of it being used as decision support. Newer proposals for hybrid COLAV based on MPC has been shown in [85] with compliance to COLREGS rules 8, and 13-17.

The work in this PhD started in 2019, and at this point the research has become more mature, with a rich set of algorithms developed for automatic COLAV, where multiple of these are experimentally verified in closed loop tests [41], [52], [62], [68], [69], [73], [86]. One of the aspects still missing in COLAV is the situational awareness part [28], [87], as most studies have put focus on the control part or own-ship modelling part. The majority of algorithms proposed for maritime COLAV assume constant velocity for dynamic obstacles in their predictions. Only a few considers the prediction uncertainty [66], [69], [77], [88]. Furthermore, actually considering what the intent of nearby dynamic obstacles are in any hazardous situation has only been done in a few recent studies [69], [89], [90]. For air traffic and road vehicles however, there exist studies on intention-aware COLAV, such as in [91], [92].

1.4 Research Questions

Based on the above literature overview on maritime COLAV, three open research questions **RQ1 - RQ3** are formulated below. These are related to the topics of collision risk assessment, dynamic obstacle predictions and intention inference, and real-time feasibility of COLAV.

RQ-1: Collision Risk Assessment

Previous studies up until 2019 had little focus on adequate collision risk assessment in COLAV considering uncertainties present in GNC [28], except from a couple of studies [66], [69], [77], [88], [89]. For these studies, there are a few limitations. The CV assumption is used in [66], [77], [88], which is conservative and not valid in most hazardous situations, as vessels are expected to maneuver in order to avoid grounding and to comply with COLREGS. This assumption will thus give conservative risk estimates in a deliberate COLAV, especially when considering longer time horizons, and could lead to infeasibility as the uncertainty propagation blows up with time [79].

Cho [89] uses a Constant Turn (CT) model for obstacles in the risk assessment, which will have the same issue with less realistic trajectory predictions and large increases in the uncertainty propagation. However, the method represents a step towards more informed risk assessment, as dynamic obstacle intention inference is also considered, and combined with a reachable set based collision probability estimation approach. A priori precomputed uncertainties for a given scene, using MC simulation and polynomial regression are used as basis for the collision risk estimation in [69]. This is also a limitation, as the uncertainty will not be representable in other collision situations with varying configurations of static and dynamic obstacles.

Thus, there are improvements to be made in collision risk assessment for COLAV, with respect to considering maneuvering vessel models with adequate uncertainty representations not exploding with time, and also the trade-off with the computational efficiency of the method. The latter topic has not been discussed in any of the above mentioned studies, and will be important when evaluating the risk with increasing numbers of own-ship trajectories, static and dynamic obstacles. Addressing these limitations can result in better adherence with COLREGS rule 7, which states that scanty information shall not be used for performing collision risk assessment.

RQ-2: Predictions and intention inference

The situational awareness part of COLAV, with the improvement of nearby dynamic obstacles predictions and consideration of their intentions and associated uncertainty, has mostly been neglected in the literature, except from a few studies [68], [69], [77], [89]. This is closely related to the above research question on collision risk assessment. Again there are limitations related to modelling and uncertainty consideration in the predictions. Agrawal [68] predicts multiple probable obstacle path, and chooses the most likely one based on ad hoc set fuzzy weights. However, no uncertainty is considered for the path. The problem with precomputed uncertainties as in Shah [69] was mentioned under **RQ-1**. Furthermore, little details on the obstacle motion model is provided in [69]. Cho [89] considers a CT model with uncertainty, but the predictions have limited validity over longer time horizons. This is, as discussed in the above section, also the case for CV-based predictions with or without uncertainty, as in [66], [77], [88].

These limitations creates a gap to be filled with COLAV methods that can more accurately predict multiple alternative obstacle maneuvering scenarios and use the associated intention and kinematic uncertainty, resulting in elevated situational awareness. Since vessels are expected to maneuver in hazardous situations, and more so when constrained by grounding hazards and multiple nearby dynamic obstacles, this is required for properly informed autonomous ship decision making. Thus, improving on the prediction part in a deliberate COLAV to take into account vessel maneuvering, uncertainty and also their intents, has been a central focus in this thesis work. This can give better capabilities for planning safer and more efficient trajectories, avoiding nearby hazardous objects while adhering to the COLREGS when the situation admits it.

RQ-3: Real-time feasibility

Lastly, the topic of real-time feasible deliberate COLAV is considered. This is especially relevant for COLAV where longer prediction horizons and larger amounts of information should be considered. How deliberate COLAV algorithms scale with increasing amounts of possible avoidance decisions, situational information such as nearby vessels behaving in multiple various prediction scenarios, and grounding hazards, have not been formally addressed in the studies [68], [70], [74], [76], [77], [85], [93], except from [83]. Chiang [83] shows run-time results compared with other methods for increasing dynamic obstacles, and also for varying time steps considered. The other mentioned methods do not consider such an analysis with respect to changes in the environment. Furthermore, as many of these methods employ a simplistic CV prediction for dynamic obstacles [70], [74], [76], [77], [85], [93], the computational demand will be lower than if more sophisticated predictions are used, as e.g. the joint forward simulator in [83].

Thus, there is work to be done on thorough investigation of run-time properties for a deliberate COLAV method which explicitly takes into account increasing amounts of both own-ship decisions and situational information.

1.5 Autonomous Ship System Architecture

The challenges put forth by grounding hazards and uncertainties in dynamic obstacle tracking, vessel intentions, COLREGS situation interpretation and adherence, put requirements on how the system architecture for the autonomous ship should be designed.

A simplified example architecture for the autonomous ship which addresses these challenges is shown in Figure 1.4. Here, the COLAV system is composed of three layers, as in a hybrid structure. It uses feedback from a situational awareness system and the vessel sensor and navigation suite. The top level COLAV planner is typically run at the start of the mission, and calculates a global path for the ship to follow, using the objective and e.g. static obstacle information. Information on dynamic obstacles and updated static obstacles are used by the mid level COLAV planner to modify the nominal trajectory output from the top level planner if necessary. Lastly, this modified path goes through the low-level reactive COLAV which can plan emergency avoidance maneuvers if necessary. The sampling time in this COLAV hierarchy typically decreases as one moves down the layers. The top level planner is typically being run only once per mission or if new objectives are received, while the mid-level planner can be run regularly depending on the prediction horizon it considers. The low-level COLAV has the fastest sampling time due to it being designed for reacting to cases or new information the mid-level COLAV has not accounted for.

It is also possible to go towards a more integrated approach, with one COLAV module for considering multiple aspects of the problem, such as a merged middle and lower-level COLAV which takes in a nominal static obstacle free trajectory from a top level planner. This has been the main focus for COLAV in this thesis work, where the Probabilistic Scenario-based Model Predictive Control (PSB-MPC) has been developed [94]. The PSB-MPC can serve as both mid and low-level COLAV depending on its prediction horizon and sampling time.

For the situational awareness part, it is necessary to have a module that can detect and track dynamic obstacles reliably, typically named the target tracking system in the literature [95], originating from its military applications. The track estimates on targets or dynamic obstacles and other information is then used by an inference module which estimates their intent towards e.g. COLREGS adherence and destination target [96]. This can then be used in the mid and low-level COLAV for better decision making. With respect to grounding hazards, there is also the need for a managing unit which can process sea map data in typical Electronic Navigational Chart (ENC) format, vessel sensor and navigation data, and weather information to output updated static obstacle data for the autonomous ship.

Other aspects not mentioned in this architecture includes modules such as a self-monitoring system for cyber-security, ship integrity, sensor and navigation systems. This is omitted here as the focus in this thesis has been more exteroceptive, as COLAV is considered.

1.6 Contributions and Outline

The thesis is structured into nine chapters based on the contributions made during this PhD. The outline is given in chronological order based on the six publications where I have been the main author, which comes after this introductory chapter and Chapter 2 on preliminaries. Lastly, a discussion with concluding remarks on the thesis work and how it relates to the research questions, reflections on the work and suggestions for future research directions are given in Chapter 9. The following text gives an overview of the chapters, including the publications on which they are based on, their topic, and the main contributions therein.

Chapters 3 - 5 represent the incremental development of Collision Probability Estimators (CPEs), for use in COLAV that has been done throughout the PhD work, as a way of taking into account uncertainties present in situational awareness in hazardous situations. Furthermore, Chapter 4 represent a first step for using dynamic obstacle intent information in an MPC framework, which Rothmund et. al. embodies in an intent inference module in [96]. Chapters 3 - 7 represent incremental

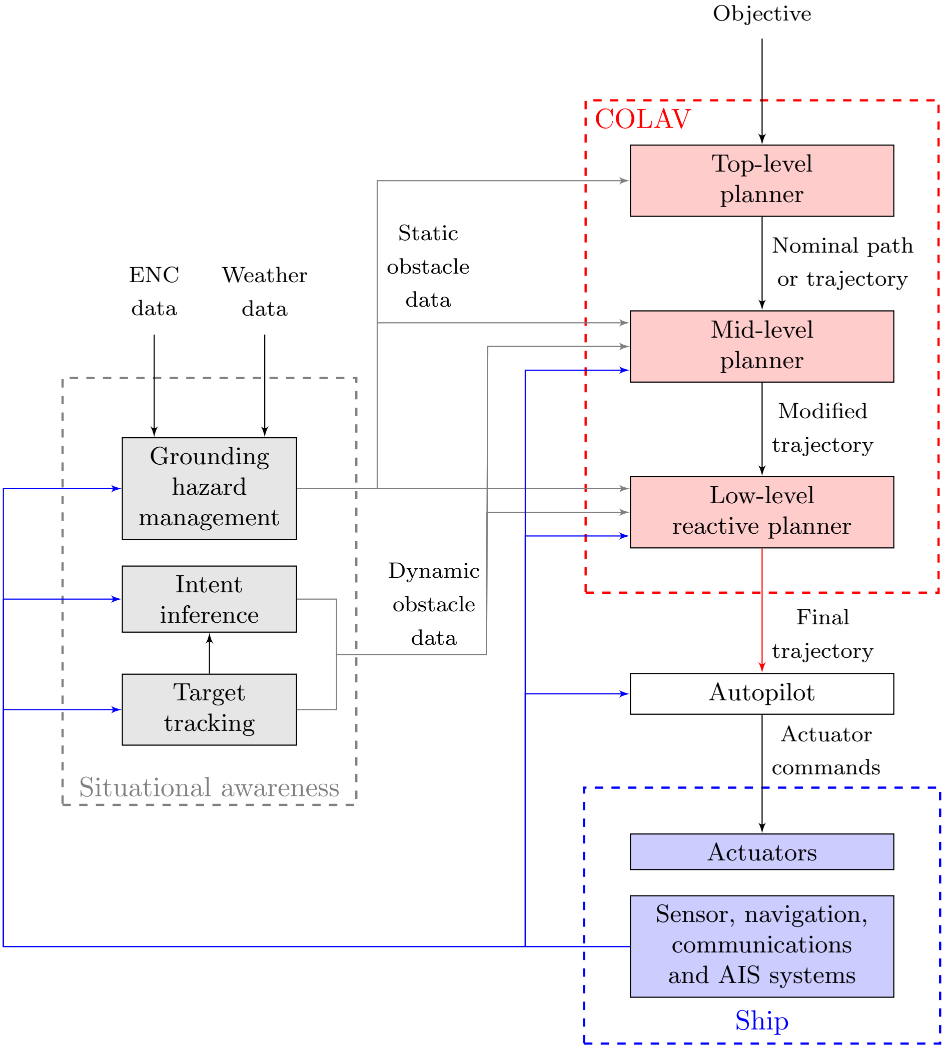


Figure 1.4: Example system architecture for the autonomous ship.

improvements of the Probabilistic Scenario-based Model Predictive Control (PSB-MPC) as steps towards a risk-based COLREGS-compliant COLAV with adequate situational awareness for real-time use on maritime vessels, experimentally verified in Chapter 7. Lastly, Chapter 8 details the work on destination inference and long-term prediction for maritime vessels, was pursued during the author's research stay in La Spezia during the Spring 2022.

1.6.1 Preliminaries (Chapter 2)

This chapter contains information on preliminary aspects used throughout the thesis. This includes mathematical concepts used throughout the text, an overview of different models used in the PhD, information on sampling based estimation techniques, and an overview of the Scenario-based MPC (SB-MPC) on which large parts of this work is based on.

1.6.2 First Edition of Collision Risk Assessment in a Probabilistic Scenario-based MPC (Chapter 3)

[97] **T. Tengesdal**, E. F. Brekke and T. A. Johansen, "On collision risk assessment for autonomous ships using scenario-based MPC", IFAC-PapersOnLine, vol. 53, no. 2, pp. 14509-14516, 2020, *21st IFAC World Congress, Berlin.*

The chapter introduces a way of dealing with the kinematic uncertainty present when relying on track estimates of nearby dynamic obstacles in COLAV. This uncertainty must be accounted for in robust COLAV systems to ensure both safe and efficient operation of the vessel in accordance with the traffic rules. To enable this, a COLAV system built on the Scenario-based Model Predictive Control (SB-MPC) with dynamic probabilistic risk treatment is presented. The system estimates the probability of collision with all nearby obstacles using a combination of Monte Carlo simulation (MCS) and a Kalman Filter (KF). A probabilistic collision cost is then used in the MPC to penalize risk-taking maneuvers. Simulation results show that the proposed method may provide increased robustness due to increased situational awareness, while also being able to efficiently follow the nominal path and adhere to the traffic rules.

The novelty presented in this chapter is related to the developed Collision Probability Estimator (CPE), which can take dynamic obstacle kinematic uncertainty in both position and velocity into account. A strategy based on using MCS for estimating a defined collision probability over a time step, and then filtered through a KF to reduce statistical sampling noise is new in the literature. The probability is defined through the event of collision between the own-ship and an obstacle at any time in the future, thus also making the probability predictive. This requires sampling both position and velocity in the MCS from the four-dimensional Probability Density Function (PDF) describing the obstacle state, where constant velocities until the time of collision is assumed. Because the MCS output is filtered by the

KF, it reduces the need for large sampling numbers in order to reduce statistical variance, making the CPE fast. The first version of the Probabilistic SB-MPC (PSB-MPC) then uses these probability estimates in a less ad-hoc manner in its collision risk assessment.

1.6.3 Second Edition of Collision Risk Assessment in a Probabilistic Scenario-based MPC with Obstacle Intent Consideration (Chapter 4)

[98] **T. Tengesdal**, T. A. Johansen and E. F. Brekke, "Risk-based autonomous maritime collision avoidance considering obstacle intentions", in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020.

A robust and efficient COLAV system for autonomous ships is dependent on a high degree of situational awareness. This includes inference of the intent of nearby obstacles, including compliance with traffic rules such as COLREGS, in order to enable more robust decision making for the autonomous agent. In this chapter, a generalized framework for obstacle intent inference is introduced. Different obstacle intentions are then considered in the second version of the PSB-MPC COLAV algorithm using an examplatory intent model, when statistics about traffic rules compliance and the next waypoint for an obstacle are assumed known. Simulation results show that the resulting COLAV system is able to make safer decisions when utilizing the extra intent information.

The chapter presents several contributions. The first is the definition of a generalized framework which can be used for maritime obstacle intent inference, through the use of Bayesian networks. Furthermore, the CPE introduced in the previous chapter is improved to take into account vessel maneuvering in the estimation, by defining the collision probability through the event that a collision occurs over the next prediction time step in the PSB-MPC. This makes the collision probability estimation less conservative.

The main contribution is the introduction of the new PSB-MPC version, which can use the updated CPE and obstacle intention information for better decision making. This is because it can take into account obstacle kinematic uncertainty for maneuvering obstacles, a priori obstacle COLREGS compliance information and future waypoint information. A side contribution here is the enhanced prediction scheme for the PSB-MPC, which in order to accomodate intent information allows for multiple alternative maneuvering scenarios for each dynamic obstacle, through the use of an Ornstein-Uhlenbeck (OU) process.

1.6.4 Third Edition of Collision Risk Assessment in a Probabilistic Scenario-based MPC Using The Cross-Entropy Method (Chapter 5)

[94] **T. Tengesdal**, E. F. Brekke and T. A. Johansen, "Ship collision avoidance utilizing the cross-entropy method for collision risk assessment", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11148-11161, 2022.

In this chapter, a new approach for ship-ship collision probability estimation based on the Cross-Entropy (CE) method is introduced, which can be treated as an adaptive importance sampler. It has the advantage of attaining low variance estimates of small collision probabilities, which will most often be the case in realistic scenarios. The CE-based CPE is used in the risk assessment of the PSB-MPC, and tested in a simulation study, where the total system is validated. Simulation results show that the MPC is able to utilize the new CPE for better risk assessment than the original version, in order to make safer decisions in close quarter situations and cases where nearby obstacles make unexpected maneuvers. It is also shown that when all vessels involved use the PSB-MPC, situations are resolved according to the traffic rules in a safe manner.

The CE-based CPE is the main contribution, which compared to the current state of the art can produce low variance probability estimates at reasonable computational cost, especially for smaller collision probabilities. The method is benchmarked against other CPEs, such as the one developed in the start of this PhD work in Chapters 3 and 4, and other sampling based methods. Because of the low variance estimate property, it will lead to decreased sensitiveness in decision making when used in the COLAV system for risk assessment, which is desirable.

1.6.5 Real-time Feasible Probabilistic Scenario-based MPC (Chapter 6)

[99] **T. Tengesdal**, T. A. Johansen, T. D. Grande and S. Blindheim, "Ship Collision Avoidance and Anti Grounding Using Parallelized Cost Evaluation in Probabilistic Scenario-based Model Predictive Control", *IEEE Access*, 2022. Submitted.

The ability to effectively process large amounts of information in reasonable time will be important for robust pro-active collision avoidance (COLAV) algorithms. Failure to do so can lead to collision, and can be compared to lack of proper supervision from officers on watch. An implementation of the PSB-MPC on a Graphical Processing Unit (GPU) is here presented for static and dynamic obstacle avoidance. The MPC can read and preprocess ENC data into a low computational cost representation of static obstacles for anti-grounding purposes. Simulation results demonstrate that the COLAV method can comply with COLREGS and keep safe distance to grounding hazards and nearby vessels with relatively low computational cost. Corresponding run-time results presented demonstrates that the algorithm utilizing parallel processing performs better than the alternative for increasing numbers of own-ship control behaviours, nearby static and dynamic obstacles, and dynamic obstacle prediction scenarios considered.

The contribution in this chapter is thus the parallelized implementation of the PSB-MPC on a GPU platform which incorporates both dynamic obstacle avoidance and anti-grounding. The parallelized computation makes it possible for the COLAV to consider larger amounts of situational awareness information in the form of obstacle prediction scenarios considered, and possible own-ship avoidance decisions.

1.6.6 Full-scale Experiments With an Obstacle Intention-Aware Probabilistic Scenario-based MPC (Chapter 7)

[100] **T. Tengesdal**, Sverre V. Rothmund, Erlend A. Basso, T. A. Johansen, and H. Schmidt-Didlaukies, "Obstacle Intention Awareness in Automatic Collision Avoidance: Full Scale Experiments in Confined Waters", *Field Robotics*, 2022. Submitted.

In this chapter, full-scale experiments with a dynamic obstacle intention-aware PSB-MPC-based COLAV system are presented. The COLAV system consists of the PSB-MPC for trajectory planning, dynamic obstacle avoidance, and anti-grounding, with a Dynamic Bayesian Net (DBN) for inferring obstacle intentions online. The experiments put emphasis on hazardous situations where intention information is both useful and necessary to avoid high collision risk. Experimental results demonstrate the validity of the proposed COLAV scheme, with adherence to the COLREGS rules 7-8 and 13-17 in a diverse set of situations.

The main contribution of the chapter is the experimental validation of an intention inference module for use in deliberate COLAV planning, bridging the gap between the situational awareness and decision making aspects of the problem at a higher degree. Intention information in the form of probabilities for different alternative obstacle prediction scenarios, its inclinations of giving way and performing COLREGS-compliant maneuvers, are used in the PSB-MPC for more informed decision making.

1.6.7 Vessel Destination and Kinematics Prediction Using a Maritime Traffic Graph (Chapter 8)

[101] **T. Tengesdal**, L. Millefiori, P. Braca, and E. F. Brekke, "Joint Stochastic Prediction of Vessel Kinematics and Destination based on a Maritime Traffic Graph", in *2nd International Conference on Electrical, Computer, Com-*

munications and Mechatronics Engineering (ICECCME), Maldives, 2022. In press.

Maritime traffic increases every year, and therefore also the amount of traffic data from the Automatic Identification System (AIS). Utilizing these data on traffic patterns and possible destinations for long-term vessel prediction is an important way of gaining maritime situational awareness (MSA) for use in the decision making in autonomous ships, such as collision avoidance algorithms, which can reduce collision risk during voyage. In this chapter, a destination inference method based on piecewise Ornstein-Uhlenbeck (OU) processes for predicting vessel motions toward a set of destinations is detailed. The mean velocities of the processes are inferred through the creation of a maritime graph that represents the major traffic patterns in the area of consideration. Tested on a real-time AIS dataset, the method is shown to perform better than current state-of-the art methods in destination inference as it indirectly takes land and passed destinations into account.

The usage of a maritime graph for capturing traffic patterns in historical AIS data, in a Bayesian vessel destination inference and kinematic state prediction method, is the main contribution. Furthermore, the combination of a traffic pattern informed piecewise OU process and an Equilibrium Reverting Velocity (ERV) bridge model for the vessel prediction is new, and allows for taking land indirectly into account. This is because the OU model part predicts the vessel kinematics along the maritime graph, which captures major sea lanes in the considered geographical area of interest. The ERV bridge model then predict the vessel convergence towards the destination. As current state of the art [102], [103] have only used a bridge model for the vessel from the start to endpoint, the errors for long-term prediction will be larger than for the proposed approach, as is shown in the chapter.

The work in this chapter is not directly linked to the development of the PSB-MPC presented in the foregoing chapters, but showcases how maritime traffic information can be combined with Bayesian inference to improve vessel kinematics prediction for increased situational awareness. A natural step-up would then be to incorporate the method in the prediction framework of the PSB-MPC, for deliberate COLAV planning at sea with longer time horizons.

1.6.8 Other Publications

I also contributed on the following paper

[96] S. V. Rothmund, **T. Tengesdal**, E. F. Brekke, and T. A. Johansen, "Intention modelling and inference for autonomous collision avoidance at sea", Preprint submitted to the Journal of Ocean Engineering, 2022.

which builds upon results in Chapter 4 for developing an intention inference module for use in the PSB-MPC and other compatible COLAV methods. It is included in the Appendix of this thesis.

The method proposed in the above publication uses a dynamic Bayesian network (DBN) to model and infer the intentions of other ships in open waters based on their observed real-time behavior. Multiple intention nodes are included to describe the different ways a ship can interpret and conflict with the behavioral rules outlined in COLREGS. The prior probability distributions of the intention nodes are adapted to the current situation based on observable characteristics such as location and relative ship size. The resulting model is able to identify situations that are prone to cause misunderstandings and infer the state of multiple intention variables that describe how the ship is likely to behave.

Chapter 2

Preliminaries

2.1 Introduction

This chapter provides descriptions of mathematical concepts, models, systems and algorithms commonly used or referred to in this thesis.

2.2 Sampling-based Integration

For continuous random variables $\boldsymbol{x}$ distributed according to some PDF $p(\boldsymbol{x}, \boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$, integrals are a means for extracting interesting properties from the distribution, such as the mean and variance. In the context of this thesis, estimating probabilites of collision has been central, which also involve an integral over the considered PDF. Consider the following integral

$$I = \int_{\mathcal{R}} \boldsymbol{g}(\boldsymbol{x}) p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{x}, \tag{2.1}$$

where $\boldsymbol{g}(\boldsymbol{x})$ is a vector-valued function of $\boldsymbol{x} \in \mathbb{R}^{n_x}$ with dimension n_x, and $\mathcal{R}$ is the domain of integration. If $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ and $\mathcal{R} = \mathbb{R}^{n_x}$, the value of I will be the ubiquitous expected value of $\boldsymbol{x}$ under the PDF $p(\boldsymbol{x}, \boldsymbol{\theta})$. On the other hand, if the function is equal to one, i.e. $\boldsymbol{g}(\boldsymbol{x}) = 1$, the value of (2.1) will be a *probability* [104], [105]. As the latter case has been central in this PhD topic, integrals on the form

$$I_p = \int_{\mathcal{R}} p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{x}, \tag{2.2}$$

is considered here. For multivariate distributions, solving (2.2) analytically or numerically can often be intractable. The choice fell on using sampling-based estimators in this thesis, due to their flexibility and simplicity, with MCS and Importance Sampling (IS) being two central methods here, outlined below.

2.2.1 Monte Carlo Simulation

MCS or MC integration [106] is a common way of estimating the value of (2.1) and (2.2), when analytical solutions are hard to find or intractable, or when numerical approximations are inaccurate. Applied to (2.2), the method involves sampling N_s independent and identically distributed random variables (i.i.d.) $\{\boldsymbol{x}\}_{z=1}^{N_s}$ directly from the PDF $p(\boldsymbol{x}, \boldsymbol{\theta})$, and estimate the integral as

$$\hat{I}_p^{MCS} = \frac{1}{N_s} \sum_{z=1}^{N_s} \mathcal{I}\{\boldsymbol{x}^z \in \mathcal{R}\} \tag{2.3}$$

where $\mathcal{I}\{\boldsymbol{x}^z \in \mathcal{R}\} \in \{0, 1\}$ is an indicator variable determining whether or not the sample $\boldsymbol{x}^z$ is inside the integration domain $\mathcal{R}$. The estimator is unbiased, as increasing sampling numbers will in theory ensure convergence to the true value, i.e. $N_s \to \infty$, $\hat{I}_p \to I_p$. The downside of MCS is the requirement of being able to sample directly from $p(\boldsymbol{x}, \boldsymbol{\theta})$, which can be intractable. Furthermore, a large sample number N_s is often required to ensure convergence.

2.2.2 Importance Sampling

In many cases, depending on how the distribution of $\boldsymbol{x}$ looks, it can be inefficient or intractable to use MCS, sampling directly from $p(\boldsymbol{x}, \boldsymbol{\theta})$. The Importance Sampling (IS) approach can then be a viable approach [107]. Note that the probability (2.2) can be written as

$$I_p = \int_{\mathcal{R}} \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{\lambda(\boldsymbol{x}, \boldsymbol{\nu})} \lambda(\boldsymbol{x}, \boldsymbol{\nu}) d\boldsymbol{x} \tag{2.4}$$

where $\lambda(\boldsymbol{x}, \boldsymbol{\nu})$ is the importance density parameterized by $\boldsymbol{\nu}$ and $\omega^z(\boldsymbol{\theta}, \boldsymbol{\nu})$. Then, sampling $\{\boldsymbol{x}\}_{z=1}^{N_s}$ from the importance density instead of the original integrand PDF, the IS estimate can be obtained as

$$\hat{I}_p^{MCS} = \frac{1}{N} \sum_{z=1}^{N} \mathcal{I}\{\boldsymbol{x}^z \in \mathcal{R}\} \omega^z(\boldsymbol{\theta}, \boldsymbol{\nu}) \tag{2.5}$$

where $\omega^z(\boldsymbol{\theta}, \boldsymbol{\nu}) = \frac{p(\boldsymbol{x}^z, \boldsymbol{\theta})}{\lambda(\boldsymbol{x}^z, \boldsymbol{\nu})}$ are the so-called importance weights. The importance density should be chosen such that it has support everywhere $p(\boldsymbol{x}, \boldsymbol{\theta}) > 0$ inside $\mathcal{R}$. The downside with IS is that it can be hard to select a feasible importance density, with proper support and ease of sampling from.

2.3 Coordinate Frames

States and measurements required for automatic control of the *own-ship* along desired trajectories, are referenced to specific coordinate systems. When using GPS, position measurements are given relative to an earth-centered frame, whereas measurements of nearby dynamic obstacles from a radar are given in a vessel-fixed coordinate frame. Furthermore, static obstacles read in from map data can be specified in a local Universal Transverse Mercator (UTM) projection for a given zone [108]. Thus, there is a need for different coordinate frames and transformations between them, when considering GNC for autonomous ships, target tracking of dynamic obstacles, and static obstacles from georeferenced map data in COLAV.

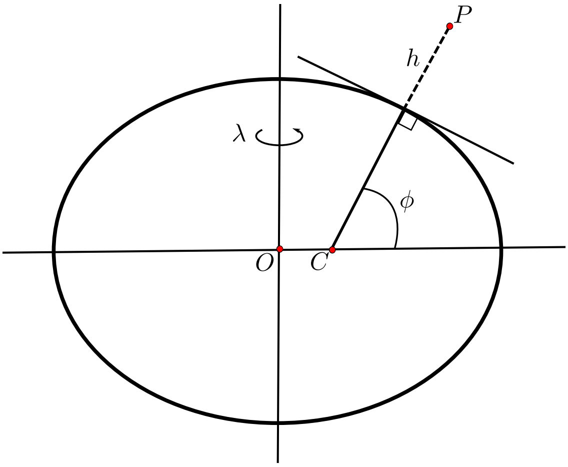


Figure 2.1: Illustration of the WGS ellipsoid with the definition of latitude ϕ, longitude λ and height h. P is an arbitrary point, O is the ellipsoid centroid, C the point on the equatorial surface when projected normally downwards from P through the shown surface tangential plane.

2.3.1 The World Geodetic System

For mapping and satelite-based navigation such as GPS, the World Geodetic System (WGS) is the common standard used as a reference today [109]. The WGS-84 revision is the latest version, and defines a reference ellipsoid and a horizontal datum which coordinate frames are defined relative to. The reference ellipsoid is an approximation of the earth‘s surface, and includes definitions of fundamental constants such as the origin at the earth‘s center of mass, equatorial radius and flattening factor. The datum then describes how positions are measured on the earth with the standard.

From this standard, positions on the earth‘s surface are described through the latitude ϕ, longitude λ and ellipsoidal height h, illustrated in Fig. 2.1.

North-East-Down and BODY

In this work, mainly the North-East-Down (NED) and ship centered BODY frame have been used. The NED frame is defined with a chosen origin on the earth surface, relative to the reference ellipsoid from WGS-84. It is the most common coordinate system, used extensively in our daily lives, and essentially defines a tangent plane with the x-axis pointing towards the true north, the y-axes towards east, and the z-axis pointing in a normal downwards from the earth surface as in Fig. 2.1. A reference latitude ϕ_0 and longitude λ_0 defines the NED frame origin relative to the WGS-84 earth approximation. The NED-frame is typically used in *flat Earth navigation*, valid for vessels travelling only within a local area such as the in the Trondheimsfjord.

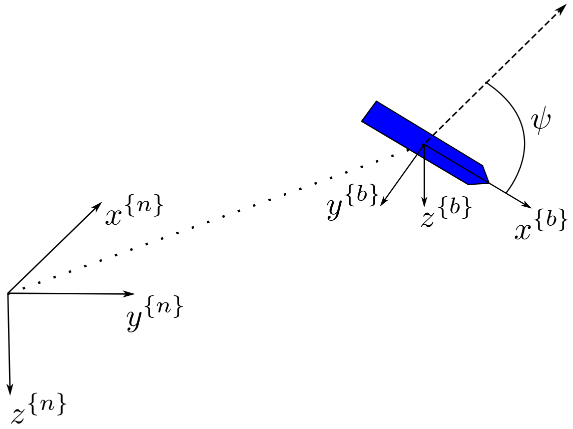


Figure 2.2: Illustration of the relationship between NED and BODY through a translation and rotation of ψ from NED to BODY for the blue vessel. The axes of NED and BODY are denoted with superscript $\{n\}$ and $\{b\}$, respectively.

The BODY frame on the other hand, is a coordinate system defined relative to the NED-frame, with origin fixed to the vessel, x-axis along the major vessel length, y-axis along its width, and z-axis pointing downwards [110]. The relationship between the NED frame and BODY frame is shown in Fig. 2.2.

Then, given a vector $\boldsymbol{x}^{\{n\}}$ described in the NED frame, it can be transformed to the BODY frame using the rotation matrix $\boldsymbol{R}_n^b \in \mathrm{SO}(3)$ between BODY and NED through

$$\boldsymbol{x}^{\{b\}} = \boldsymbol{R}_n^b \boldsymbol{x}^{\{n\}} \tag{2.6}$$

where the special orthogonal group of order 3 is defined as [110]

$$\mathrm{SO}(3) = \{\boldsymbol{R} \quad | \quad \boldsymbol{R} \in \mathbb{R}^{3\times 3}, \quad \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}\} \tag{2.7}$$

2.4 Vessel Modelling

Different models have been used for maritime vessels throughout the PhD work, where the type and usage depend on the application. Some example cases for different model requirements are given below:

- The tracking system providing the COLAV system with nearby dynamic obstacle estimates commonly use a stochastic model such as the Constant Velocity (CV) model [78], [111] for prediction, in combination with a tracker such as a KF or PDAF. As the prediction is only for a small time step, the constant velocity assumption can be feasible and thus often used [95], [111], [112]. A stochastic model is here used to have an uncertainty measure for the track estimates.

- For the own-ship trajectory prediction in the MPC, a deterministic model without uncertainty description can be used, as one can normally use GNSS and IMUs to have accurate navigation estimates compared to that of nearby tracked vessels [113]. Furthermore, as the modelling error will only increase with larger prediction horizons, it can be argued that a simplistic prediction model should be used, where one assumes that the vessel motion control system can compensate for complex modelling objects such as environmental disturbances.
- The dynamic obstacle trajectory prediction in the MPC can, on the same note as for the own-ship prediction, use a simplistic model, but should be stochastic in the attempt to capture realistic future maneuvering uncertainty for the obstacle. This is why an Ornstein-Uhlenbeck (OU) process has been used in the PSB-MPC for prediction, as it is simple and can produce realistic levels of uncertainties for vessels with estimated or assumed known mean velocities [79]. Having multiple prediction scenarios or trajectories allows for taking into account multiple intention scenarios for the obstacle, and thus more uncertainty.

2.4.1 Constant Velocity Model

Commonly used in target tracking [78], the CV model captures the kinematics of an obstacle i with state vector $\boldsymbol{x}^i = [x^i, V_x^i, y^i, V_y^i]^T$, where x^i, y^i are the position coordinates, and V_x^i, V_y^i are velocity components, all typically given in the NED or North-East frame, respectively. The model in discrete time for obstacle i is given by

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{F}\boldsymbol{x}_k^i + \boldsymbol{\epsilon}_k^i \tag{2.8a}$$

$$\boldsymbol{y}_k^i = \boldsymbol{H}\boldsymbol{x}_k^i + \boldsymbol{w}_k^i \tag{2.8b}$$

where $\boldsymbol{F}$ and $\boldsymbol{H}$ are the transition and measurement matrix, respectively. The vector $\boldsymbol{\epsilon}_k^i$ is the process noise affecting obstacle i, and $\boldsymbol{w}_k^i$ the measurement noise affecting the measurement at discrete time instant t_k. Lastly, the vector $\boldsymbol{y}_k^i$ contains the noise corrupted measurement, containing position and/or velocity information, depending on the available information, reflected through the matrix $\boldsymbol{H}$. The system matrices $\boldsymbol{F}$ and $\boldsymbol{Q}$ are given as

$$\boldsymbol{F} = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{Q} = \sigma_a^2 \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} & 0 & 0 \\ \frac{T_s^2}{2} & T_s & 0 & 0 \\ 0 & 0 & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ 0 & 0 & \frac{T_s^2}{2} & T_s \end{bmatrix} \tag{2.9}$$

where $T_s = t_{k+1} - t_k$ is the sampling time for the linear prediction, and could be time varying. The process noise and measurement noise are assumed to be zero mean, white, mutually independent and Gaussian. The process noise strength σ_a is chosen based on the expected maneuverability of the vessel [114]. Note that other prediction- and measurement models are also possible, for instance the constant turn rate model or the constant acceleration model [78].

2.4.2 Ornstein-Uhlenbeck Process

The OU process can be used to represent the motion of non-maneuvering vessels at open sea, as first described in [115]. The model has the property that an obstacle will have a tendency to revert towards its mean velocity with time. As commercial traffic mostly keep a constant speed, the property is valuable for prediction purposes. Furthermore, the model has beneficial uncertainty propagation properties compared to the CV model [79].

Here, $\boldsymbol{x}_k = \boldsymbol{x}(t_k) = [\boldsymbol{s}_k^T \quad \dot{\boldsymbol{s}}_k^T]^T$ is the state vector at discrete time t_k consisting of the $r = 2$ two-dimensional position s_k and velocity $\dot{\boldsymbol{s}}_k$ of the maritime vessel in a suitable coordinate system. Denoting the variables as function of the time t, the model has its basis from the following Stochastic Differential Equation (SDE)

$$d\boldsymbol{s}(t) = \boldsymbol{A}_{OU}\boldsymbol{s}(t) + \boldsymbol{G}_{OU}\boldsymbol{v}dt + \boldsymbol{B}_{OU}d\boldsymbol{w}(t) \tag{2.10}$$

where $\boldsymbol{v} = [v_x, v_y]^T$ is the assumed constant mean velocity or long-run OU-process mean with velocity components in a planar reference system such as the north-east frame. The system matrices are given as

$$\boldsymbol{A}_{OU} = \begin{bmatrix} \mathbf{0}_{r\times r} & \boldsymbol{I}_{r\times r} \\ \mathbf{0}_{r\times r} & -\boldsymbol{\Theta} \end{bmatrix} \tag{2.11}$$

$$\boldsymbol{B}_{OU} = \begin{bmatrix} \mathbf{0}_{r\times r} & \boldsymbol{C} \end{bmatrix} \tag{2.12}$$

$$\boldsymbol{G}_{OU} = \begin{bmatrix} \mathbf{0}_{r\times r} & \boldsymbol{\Theta} \end{bmatrix} \tag{2.13}$$

with $\mathbf{0}_{r\times r}$ and $\boldsymbol{I}_{r\times r}$ being the r-dimensional null matrix and identity matrix, and where $\boldsymbol{C}$ and $\boldsymbol{\Theta}$ are generic bi-dimensional matrices. The revertion effect of the process towards its mean velocity is captured by the matrix $\boldsymbol{\Theta}$, normally assumed positive definite and decomposed with an affine transformation as $\boldsymbol{\Theta} = \boldsymbol{J}\boldsymbol{\Gamma}\boldsymbol{J}^{-1}$ where $\boldsymbol{J}$ is a transformation matrix and $\boldsymbol{\Gamma}$ is a diagonal matrix of revertion strength $\boldsymbol{\gamma} = [\gamma_x, \gamma_y]$ in each dimension. The target state at discrete times $\boldsymbol{x}_k$ can be found from the first moment of the solution to (2.10) [116], and can be written as

$$\boldsymbol{x}_k = \boldsymbol{\Phi}(h)\boldsymbol{x}_{k-1} + \boldsymbol{\Psi}(h)\boldsymbol{v} + \boldsymbol{w}_k \tag{2.14}$$

with

$$\boldsymbol{\Phi}(h) = \begin{bmatrix} \boldsymbol{I}_{r\times r} & (\boldsymbol{I}_{r\times r} - e^{-\boldsymbol{\Gamma}h})\boldsymbol{\Gamma}^{-1} \\ \mathbf{0}_{r\times r} & e^{-\boldsymbol{\Gamma}h} \end{bmatrix} \tag{2.15}$$

$$\boldsymbol{\Psi}(h) = \begin{bmatrix} h\boldsymbol{I}_{r\times r} - (\boldsymbol{I}_{r\times r} - e^{-\boldsymbol{\Gamma}h})\boldsymbol{\Gamma}^{-1} \\ \boldsymbol{I}_{r\times r} - e^{-\boldsymbol{\Gamma}h} \end{bmatrix}, \tag{2.16}$$

where we then assume $\boldsymbol{\Theta} = \boldsymbol{\Gamma}$ is diagonal and thus $\boldsymbol{J} = \boldsymbol{I}_{r\times r}$. The covariance prediction in the OU process from a time t_k to t_{k+1} is given through

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k + \boldsymbol{\Sigma}_1 \circ \boldsymbol{\Sigma}_2(t_{k+1} - t_k) \tag{2.17}$$

where $\boldsymbol{P}_k$ is the covariance at t_k, and

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} \frac{\sigma_x^2}{\gamma_x^3} & \frac{\sigma_{xy}}{\gamma_x\gamma_y} & \frac{\sigma_x^2}{2\gamma_x^2} & \frac{2\sigma_{xy}}{\gamma_x} \\ \frac{\sigma_{xy}}{\gamma_x\gamma_y} & \frac{\sigma_y^2}{\gamma_y^3} & \frac{2\sigma_{xy}}{\gamma_y} & \frac{\sigma_y^2}{2\gamma_y^2} \\ \frac{\sigma_x^2}{2\gamma_x^2} & \frac{2\sigma_{xy}}{\gamma_y} & \frac{\sigma_x^2}{\gamma_x} & \frac{2\sigma_{xy}}{\gamma_x+\gamma_y} \\ \frac{2\sigma_{xy}}{\gamma_x} & \frac{\sigma_y^2}{2\gamma_y^2} & \frac{2\sigma_{xy}}{\gamma_x+\gamma_y} & \frac{\sigma_y^2}{\gamma_y} \end{bmatrix} \tag{2.18}$$

being the noise matrix $\boldsymbol{\Sigma}_1$ as a function of the Wiener noise and revertion strengths, with the symbol $\circ$ in (2.17) denoting the Hadamard product. The parameters σ_x, σ_{xy} and σ_y are the Wiener process noise values in the OU process. The matrix $\boldsymbol{\Sigma}_2$ is a function of the prediction time step and revertion strengths, and given as

$$\boldsymbol{\Sigma}_{\mathbf{2}}(t) = \begin{bmatrix} f(t\gamma_x) & h(t,\boldsymbol{\gamma}) & k(t\gamma_x) & \frac{\frac{g(\frac{t}{2}\gamma_y)}{\gamma_y} - \frac{t}{2}(\gamma_x+\gamma_y)}{\gamma_x+\gamma_y} \\ h(t,\boldsymbol{\gamma}) & f(t\gamma_y) & \frac{\frac{g(\frac{t}{2}\gamma_x)}{\gamma_x} - \frac{t}{2}(\gamma_x+\gamma_y)}{\gamma_x+\gamma_y} & k(t\gamma_y) \\ k(t\gamma_x) & \frac{\frac{g(\frac{t}{2}\gamma_x)}{\gamma_x} - \frac{t}{2}(\gamma_x+\gamma_y)}{\gamma_x+\gamma_y} & g(t\gamma_x) & g(\frac{t}{2}(\gamma_x+\gamma_y)) \\ \frac{\frac{g(\frac{t}{2}\gamma_y)}{\gamma_y} - \frac{t}{2}(\gamma_x+\gamma_y)}{\gamma_x+\gamma_y} & k(t\gamma_y) & g(\frac{t}{2}(\gamma_x+\gamma_y)) & g(t\gamma_y) \end{bmatrix} \tag{2.19}$$

with the functions in (2.19) defined as

$$f(t) := \frac{1}{2}(2t + 4e^{-t} - e^{-2t} - 3) \tag{2.20}$$

$$g(t) := \frac{1}{2}(1 - e^{-2t}) \tag{2.21}$$

$$h(t,\boldsymbol{\gamma}) := t - \frac{1 - e^{-t\gamma_x}}{\gamma_x} - \frac{1 - e^{-t\gamma_y}}{\gamma_y} + \frac{1 - e^{-t(\gamma_x+\gamma_y)}}{\gamma_x+\gamma_y} \tag{2.22}$$

$$k(t) := e^{-2t}(1 - e^{t})^2 \tag{2.23}$$

The model parameters in general need to be estimated depending on the ship type and the local traffic area. See [116] for more information on the OU process model derivation from the stochastic differential equation (SDE) framework.

2.4.3 Kinematic Vessel Model

A kinematic model based on the speed over ground (SOG) and course over ground (COG) for a vessel can be used to model nearby dynamic obstacles, and also the own-ship in an MPC prediction. It is formulated in discrete time as

$$\begin{aligned} x_{k+1} &= U_k cos(\chi_k) \\ y_{k+1} &= U_k sin(\chi_k) \\ \chi_{k+1} &= \frac{1}{T_\chi}(\chi_{d,k} - \chi_k) \\ U_{k+1} &= \frac{1}{T_U}(U_{d,k} - U_k) \end{aligned} \tag{2.24}$$

with χ, χ_d and U, U_d as the actual and desired vessel SOG and COG pairs, respectively. Perfect vessel motion control is assumed through the model, as no lower level controllers are considered. Furthermore, zero crab and slip angles are also assumed. One can also extend the model to include linear acceleration and course rate terms.

2.4.4 Kinetic Vessel Model

A three Degrees of Freedom (DOF) kinetic model for the own-ship vessel model is typically used in lower level motion control of maritime vessels, where the need to compensate for modelling errors and environmental disturbances acting on all DOFs is important.

The model is used to describe the horizontal motion of the own-ship in surge, sway and yaw [110]. The vessel position in the North-East-Down (NED) coordinate system is given by $\boldsymbol{\eta} = [x, y, \psi]^T$. The variables x, y and ψ are the ship north and east coordinates and the heading, respectively. The ship velocity in the BODY-fixed coordinate system is given as $\boldsymbol{\nu} = [u, v, r]^T$. Here, u and v are the surge and sway velocity, respectively, while r is the yaw rate. The vector $\boldsymbol{\tau} = [X, Y, N]^T$ describes the generalized forces and moments affecting the ship in surge, sway and yaw. The equations of motion for the vessel can then be represented in vectorial form as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.25}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{w} \tag{2.26}$$

where $\boldsymbol{R}(\cdot)$ is the rotation matrix from the NED frame $\{n\}$ to the BODY frame $\{b\}$, $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}(\cdot)$ the coriolis and centripetal matrix and $\boldsymbol{D}(\cdot)$ the nonlinear damping matrix. For more information on the different terms in (2.25)-(2.26), see [110].

2.5 Guidance, Navigation and Control

A system for guidance, navigation and control enables the automatic control of vehicles that move under water, on the surface or in space [110]. The guidance part enables the vehicle to follow trajectories and paths without direct human control. The navigation part is responsible for determining the vehicle position or attitude, velocity and acceleration. Lastly, the control system makes sure that position, attitude, velocity and acceleration are automatically controlled. In other words, the GNC system is the autopilot for the vessel. The remainder of this Section is mostly based on [110].

As the topic of navigation and motion control for autonomous ships have not been focused in this PhD work, only the *guidance* part is relevant here. In the remaining text of this Chapter, the Line of Sight (LOS) guidance method, which is a simple and intuitive method for steering a ship towards a path, is detailed.

2.5.1 Guidance

LOS guidance is a common and simple way to steer the own-ship towards a desired path, often parameterized as waypoints, and produce a reference course for the autopilot control system to track. The intuition behind the method is to direct the vessel course towards a point a certain lookahead distance along the path. A small lookahead distance will then lead to aggressive steering towards the path, and vice versa for a larger value.

When considering piece-wise straight line paths, the closest active straight line segment between two waypoints is considered in the LOS guidance. Given two waypoints in the NED frame, consisting of their Cartesian coordinates: $\boldsymbol{p}^n = [x^n, y^n]^T$ and $\boldsymbol{p}^{n+1} = [x^{n+1}, y^{n+1}]^T$, the first step in the LOS law is to find the path-tangential angle α^n, given as

$$\alpha^n = \text{atan2}(y^{n+1} - y^n, x^{n+1} - x^n) \tag{2.27}$$

where n is defined as the index corresponding to the current waypoint, with $n+1$ then being the next waypoint to be reached by the vessel. The operator atan2$(\cdot)$ is the four-quadrant inverse tangent function. The path-tangential angle is then used to compute the along-track error $s(t)$ and cross-track error $e(t)$, defined relative to the path-fixed reference frame as

$$\boldsymbol{\epsilon}(t) = \begin{bmatrix} s(t) \\ e(t) \end{bmatrix} = \boldsymbol{R}_p(\alpha^n)^T(\boldsymbol{p}^n(t) - \boldsymbol{p}^n) \tag{2.28}$$

where

$$\boldsymbol{R}_p = \begin{bmatrix} cos(\alpha^n) & -sin(\alpha^n) \\ sin(\alpha^n) & cos(\alpha^n) \end{bmatrix} \tag{2.29}$$

The goal of the path-following is thus to reduce the cross track error $e(t)$ towards zero, as this means that the vessel has converged onto the path. The desired course, which ensures vessel convergence onto the path, is computed as

$$\chi_d(e) = \alpha^n + \chi_r(e) \tag{2.30}$$

with

$$\chi_r(e) = \text{atan}\left(-K_p^{LOS} e\right) \tag{2.31}$$

being the velocity-path relative angle. The inverse tangent function atan returns values in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$. The parameter $K_p^{LOS} = \dfrac{1}{\Delta^{LOS}} > 0$ is a function of the lookahead distance Δ^{LOS}, and determines the steering aggressiveness towards the path. To switch between straight line path segments and increment $\boldsymbol{p}^n$ to $\boldsymbol{p}^{n+1}$, the condition

$$(x^{n+1} - x(t))^2 + (y^{n+1} - y(t))^2 \leq R_a^2 \tag{2.32}$$

must be fullfilled, with R_a being the radius of acceptance around waypoint with index $n + 1$.

In case of disturbances, integral action can be added to the LOS steering law in (2.31) to compensate for non-zero sideslip and crab angles. Extensions to curved path following can be done by parameterizing the path as a geometric curve as a function of a path variable, but will not be detailed here.

The assignment of the desired speed for the vessel control system to follow can be done in various manners. A simple solution is to set piece-wise constant values to be followed for each waypoint segment. On the other hand, specifying a speed profile as function of a path variable, with time as a constraint, leads to a *time-varying trajectory* to be tracked. The choice depends on the application, where e.g. shipping transport normally specify constant speed along piece-wise straight line segments.

If not stated otherwise, the COLAV planning algorithms presented in the upcoming chapters use one of the vessel models in sections 2.4.1 - 2.4.4 for the own-ship, and LOS-guidance to steer the vessel itself. Navigational estimates of the own-ship states are most often assumed known, as usage of GNSS and an INS with an estimator such as an Extended Kalman filter (EKF) make the corresponding uncertainty negligible.

2.6 The Scenario-based Model Predictive Control

The SB-MPC is a sampling based optimization algorithm originally published in [70]. It is based on MPC, which is an optimal control technique involving the use of a prediction model for solving a finite horizon open loop control problem, using the current system state or measurement as initial condition. The output is an optimal input sequence, where the first control input in the sequence is used applied to the controlled system, as illustrated in Fig. 2.3. The optimization is then performed at regular intervals.

What separates the SB-MPC from traditional MPC is the consideration of a finite control set (FCS). The method is therefore typically named direct MPC or Finite Control Set MPC (FCS-MPC) in the literature [118], [119], commonly used in power electronics applications due to its straight forward implementation and usage with inputs such as power converter switching states. Its formulation removes the need for using a complex numerical solver to find the optimal solution, as one

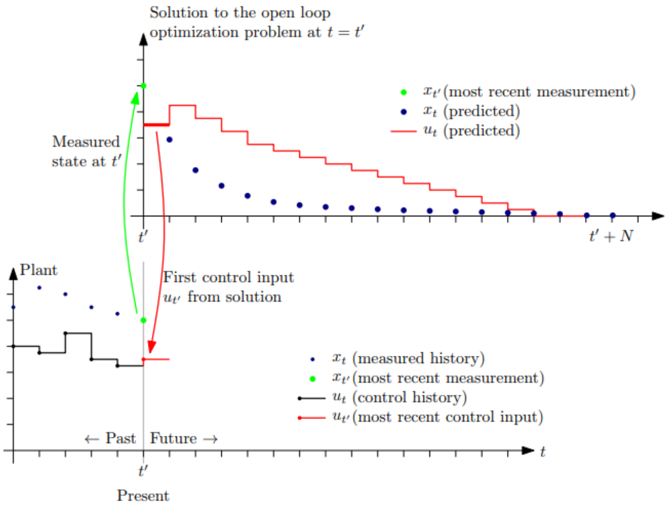


Figure 2.3: Illustration of the MPC principle [117].

only need to evaluate the cost for all considered control variables, and extract the optimal one giving minimal cost.

2.6.1 Control Behaviours

Specifically designed for autonomous ship COLAV, the SB-MPC considers a set of own-ship *control behaviors* or so-called scenarios. A control behaviour represent a certain own-ship trajectory, typically parameterized as a surge speed and course modification. The set of control behaviors typically used in the algorithm are the following finite sets

- Course offsets $\chi_m \in \{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$ degrees.
- Nominal propulsion (keep current speed), slow forward and full stop in forward speed, i.e $U_m \in \{1.0, 0.5, 0.0\}$.

which sums up to $13 \cdot 3 = 39$ behaviours. Ideally, one should investigate all control behaviors at each sample time in the SB-MPC prediction, as in a traditional MPC. However, this will quickly make the real-time implementation infeasible, as the computational demand increases substantially. For instance, with five planned changes in speed and course modifications over the prediction horizon, the amount of scenarios to evaluate grows to $39^5 = 90224199$. This is why the control behavior in the SB-MPC is normally assumed fixed over the horizon, which is a trade-off between computational cost and performance from a safety perspective.

The combination of surge and course modifications are used to predict multiple different own-ship trajectories. To predict the own-ship trajectories with the surge and course modifications, LOS guidance was used in [70]. Alternatively, a transitional cost could be included in the SB-MPC to fully decouple the guidance strategy from the COLAV system, as in [71]. Moreover, effects from wind, waves and ocean current can also be taken into account through using a more complex own-ship motion model such as the 3 DOF surface vessel model in Section 2.4.4. A trajectory tree with one sequential maneuver, similarly to the predictions in the Branching-Course MPC [86] will then result from predicting with every control behaviour in the FCS.

The models of the own-ship with LOS-guidance, and obstacle kinematics are used as equality constraints in the optimization problem. A straight line prediction model is used for dynamic obstacles, using the current time estimated velocity vector of the obstacle. This equates to using a deterministic CV model with no process noise. An outline of the SB-MPC is summarized in Fig. 2.4.

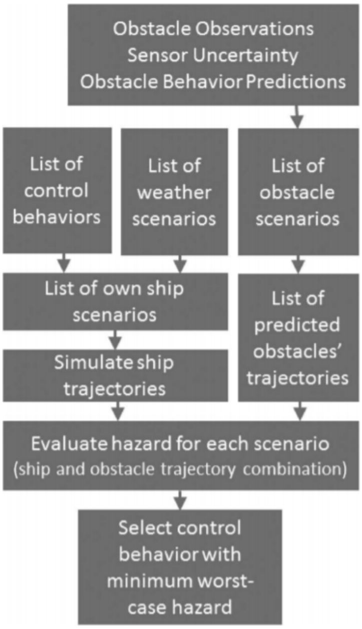

Obstacle Observations
Sensor Uncertainty
Obstacle Behavior Predictions
List of control behaviors
List of weather scenarios
List of obstacle scenarios
List of own ship scenarios
Simulate ship trajectories
List of predicted obstacles' trajectories
Evaluate hazard for each scenario
(ship and obstacle trajectory combination)
Select control behavior with minimum worst-case hazard

Figure 2.4: Summary of the collision avoidance control algorithm [70].

2.6.2 The Optimization Problem

The optimal control behaviour l^* in the SB-MPC is found by solving the optimization problem

$$l^*(t_0) = \arg\min_{l} \mathcal{H}^l(t_0) \tag{2.33}$$

at the current time t_0, where the candidate control behavior l consists of the modification tuple (χ_m^l, U_m^l) to the nominal guidance references χ_d and U_d in course and forward speed, respectively. Using the optimal control behaviour l^*, the SB-MPC modifies the autopilot references as $\chi_c = \chi_d + \chi_m^{l^*}$ and $U_c = U_d \cdot U_m^{l^*}$. Alternatively, the SB-MPC can output a reference trajectory for the autopilot to track.

The cost or hazard function $\mathcal{H}^l(t_0)$ is given as

$$\mathcal{H}^l(t_0) = \max_i \max_{t \in D(t_0)} \left(\mathcal{C}_i^l(t)\mathcal{R}_i^l(t) + \kappa_i \mu_i^l(t)\right) + f(\cdot) + g(\cdot) \tag{2.34}$$

balancing the penalization of high collision risk, COLREGS violation, path deviation and grounding, respectively. Here, $D(t_0) = \{t_0, t_0 + T_s, ..., t_0 + T\}$ contains the discrete sample times in the prediction from the current time t_0, with T_s as the sample time and T as the prediction horizon. The following text will elaborate on the cost terms with the variables involved.

Risk of Collision

The collision risk parameterization in the SB-MPC utilizes an ad-hoc risk factor $\mathcal{R}_i^l(t)$, which is multiplied by the cost of collision $\mathcal{C}_i^l(t)$ associated with obstacle i in scenario l at time t. The risk factor is defined as

$$\mathcal{R}_i^l(t) = \begin{cases} \frac{1}{|t-t_0|^p}\left(\frac{d_{safe}^i}{d_{0,i}^l(t)}\right)^q, & \text{if } d_{0,i}^l(t) \leq d_i^{safe}. \\ 0, & \text{otherwise} \end{cases} \tag{2.35}$$

where $t > t_0$ is the current prediction time. The exponent $q \geq 1$ and safety distance d_{safe}^i are chosen large enough such that COLREGS rule 16 is satisfied. The value of $p \geq \frac{1}{2}$ weights the importance of time until the event of collision occurs. Further, the cost of collision $\mathcal{C}_i^l(t)$ is given by

$$\mathcal{C}_i^l(t) = K_i^{coll} \left\|\boldsymbol{v}_0^l(t) - \boldsymbol{v}_i^l(t)\right\|^2 \tag{2.36}$$

which essentially is a scaling of the kinetic energy given by the relative velocity between the own-ship and obstacle i. Here, K_i^{coll} is the cost scaling factor, dependent on the obstacle type and size. $\boldsymbol{v}_0^l(t)$ and $\boldsymbol{v}_i^l(t)$ are the velocity vectors of the own-ship and obstacle i in the horizontal plane, respectively.

COLREGS Compliance

In order to penalize COLREGS violations, the SB-MPC employ a set of inequality tests to calculate a binary indicator $\mu_i^l \in \{0, 1\}$ representing whether or not the own-ship breached COLREGS in control behavior l with respect to obstacle i. For this penalization to be active, the dynamic obstacle must be inside a radius d_{close} about the own-ship at predicted time t. Then, a set of inequality tests are used to determine the COLREGS situation for the own-ship and obstacle i pair at the predicted time t for control behaviour l, which are given by the boolean variables CLOSE, OVERTAKEN, STARBOARD, HEAD-ON and CROSSED. These are described below.

- An obstacle i is said to be CLOSE if

$$d_{0i}^k(t) \leq d_{close} \tag{2.37}$$

- The own-ship is said to be OVERTAKEN by obstacle i if

$$\boldsymbol{v}_0^l(t)^T \boldsymbol{v}_i^l(t) > cos(68.5^\circ) \left\|\boldsymbol{v}_0^l(t)\right\| \cdot \left\|\boldsymbol{v}_i^l(t)\right\| \tag{2.38}$$

  and in addition the obstacle is CLOSE and has higher velocity than the own-ship, i.e. $\left\|\boldsymbol{v}_0^l(t)\right\| < \left\|\boldsymbol{v}_i^l(t)\right\|$.

- Obstacle i is said to be STARBOARD to the own-ship, if

$$\angle \boldsymbol{L}_i^l(t) \geq \psi^l(t) \tag{2.39}$$

  where $\boldsymbol{L}_i^l(t)$ is the unit LOS-vector from the own-ship to obstacle i, and $\psi^l(t)$ is the own-ship heading.

- Obstacle i is said to be HEAD-ON if

$$\boldsymbol{v}_0^l(t)^T \boldsymbol{v}_i^l(t) < -cos(22.5^\circ) \left\|\boldsymbol{v}_0^l(t)\right\| \cdot \left\|\boldsymbol{v}_i^l(t)\right\| \tag{2.40a}$$

$$\boldsymbol{v}_0^l(t)^T \boldsymbol{L}_i^l(t) > -cos(15^\circ) \left\|\boldsymbol{v}_0^l(t)\right\| \tag{2.40b}$$

  holds, and if it is CLOSE and the obstacle speed $\left\|\boldsymbol{v}_i^l(t)\right\|$ is not close to zero.

- Obstacle i is said to be CROSSED if it is CLOSE and

$$\boldsymbol{v}_0^l(t)^T \boldsymbol{v}_i^l(t) < cos(68.5^\circ) \left\|\boldsymbol{v}_0^l(t)\right\| \cdot \left\|\boldsymbol{v}_i^l(t)\right\| \tag{2.41}$$

The angles not associated with overtaking can in general be adjusted based on the type of obstacle, velocity and so on. Then, using the above defined boolean variables, a binary indicator $\mu_i^l \in \{0, 1\}$ determines whether rule 14 or 15 in the COLREGS are violated between the own-ship and obstacle i in scenario l, given as

$$\mu_i^l = \text{RULE14} \quad || \quad \text{RULE15} \tag{2.42}$$

where

$$\text{RULE14} = \text{CLOSE \& STARBOARD \& HEAD-ON} \tag{2.43a}$$

$$\begin{aligned}\text{RULE15} = {} & \text{CLOSE \& STARBOARD \& CROSSED} \\ & \text{\& NOT OVERTAKEN}\end{aligned} \tag{2.43b}$$

Rule 13 is also included here, as it states that the overtaking vessel shall keep out of the way. The COLREGS violation indicator is then multiplied by the penalty parameter κ_i in the total COLREGS penalization cost.

A weakness with the COLREGS penalization is that one only determines a violation at an instantaneous prediction time t, instead of considering the entire trajectory of the own-ship and obstacle. This can lead to premature violations. Furthermore, COLREGS rule 8 on readily apparent actions and 17 on stand-on responsibilities are not strictly enforced. Another weakness with the SB-MPC formulation is that it does not scale properly to multi-ship situations as the dynamic obstacle cost and COLREGS cost should then be weighted with respect to each ship.

Trajectory Deviation Cost and Grounding Cost

As the modifications in a control behaviour will lead to a deviation from the reference trajectory produced by the guidance part, the term

$$
\begin{aligned}
f(\chi_m^l, U_m^l) = & K_U(1 - U_m^l) + K_\chi \chi_m^2 + K_{\Delta U}|U_m^l - U_{m,last}| \\
& + K_{\Delta\chi}(\chi_m^l - \chi_{m,last})^2
\end{aligned}
\tag{2.44}
$$

penalizes this behavior, commonly referred to as the trajectory deviation cost. Here, K_U, K_χ, $K_{\Delta U}$ and $K_{\Delta\chi}$ are penalty parameters. K_χ and $K_{\Delta\chi}$ are generally asymmetric about the origin, chosen such that starboard course changes are preferred to port side changes. Penalization for deviation from the previous optimal modifications $U_{m,last}$ and $\chi_{m,last}$ prevents large decision variations from the MPC.

2.6.3 Location in the Control Hierarchy

The position of the SB-MPC method in the GNC-system for an autonomous ship proposed in [70] is shown in Figure 2.5. The authors used an apparent decoupling between the COLAV system and the mission planning and steering parts of the vessel. This allowed for easy modification and reusability of the COLAV method across different GNC architectures.

The inputs to the SB-MPC are predicted obstacle positions and velocities with basis from tracking system information, grounding hazard data, and own-ship navigational information and references for course and speed from the mission planner.

The method can also be directly used as a mid-level and/or low-level COLAV which takes in a nominal trajectory as input and produces a modified trajectory for the ship autopilot to track, similarly to that shown in Fig. 1.4.

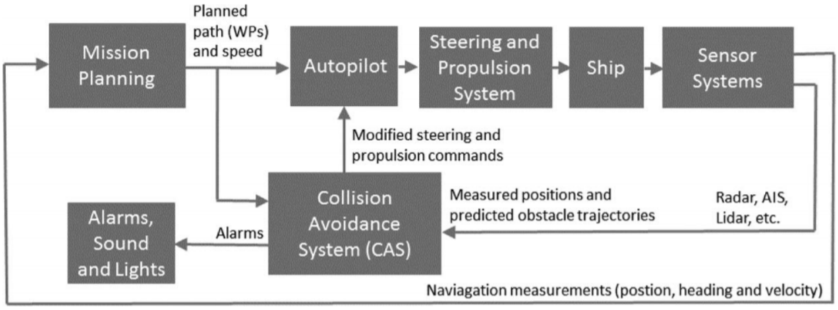

Mission Planning
Planned path (WPs) and speed
Autopilot
Steering and Propulsion System
Ship
Sensor Systems
Modified steering and propulsion commands
Collision Avoidance System (CAS)
Measured positions and predicted obstacle trajectories
Radar, AIS, Lidar, etc.
Alarms, Sound and Lights
Alarms
Naviagation measurements (postion, heading and velocity)

Figure 2.5: Block diagram illustrating the information flow between the main modules in the system [70].

Chapter 3

First Edition of Collision Risk Assessment in a Probabilistic Scenario-based MPC

The chapter is based on the following publication, which was the first step in the work of this PhD to address kinematic uncertainties of nearby objects in deliberate COLAV planning.

[97] **T. Tengesdal**, E. F. Brekke and T. A. Johansen, "On collision risk assessment for autonomous ships using scenario-based MPC", IFAC-PapersOnLine, vol. 53, no. 2, pp. 14509-14516, 2020, *21st IFAC World Congress, Berlin.*

3.1 Introduction

3.1.1 Motivation

To have adequate situational awareness in the autonomous ship, a system which detects and tracks nearby obstacles is important. This is commonly referred to as a tracking system in the maritime situational awareness domain, and is formally responsible for providing the ship COLAV system with tracks of nearby obstacles. An obstacle in this context here refers to a dynamic vessel or obstacle, but can include static grounding hazards, animals and other non-categorized objects being potential collision candidates for the ship. The track typically contains estimates of position and velocity for the generating target. These estimates will have an inherent kinematic uncertainty originating from sensor detection errors, sensor inaccuracies, modelling and configuration errors in the tracking system. Depending on if a tracker such as a KF, Probabilistic Data Association Filter (PDAF) or other stochastic Multi Target Tracking (MTT) methods are being used [120], the estimates are accompanied by a corresponding uncertainty estimate, typically an error covariance.

An autonomous ship COLAV system using these estimates needs to be informed about the level of uncertainty in the tracks received, in order to make a safe and efficient avoidance maneuver. Blindly taking the estimates in a track as ground truth can lead to a collision with fatal consequences, as is demonstrated in some of the work in this thesis. A way of taking a more informed decision is by using the accompanied kinematic uncertainty estimates to estimate the probability that a collision with the nearby dynamic obstacle will occur within some time interval into the future. This collision probability can not only be utilized by COLAV systems, but also for vessel operators as a decision support aid.

3.1.2 Previous Work

There are many existing COLAV algorithms which have COLREGS compliance at varying degree. However, only a few of these are performing probabilistic risk assessment in collision situations considering uncertainties present, as this has mostly been ignored for such systems [28]. As the COLAV problem will involve considering uncertainties present in the current situation, taking these into account, and then choosing the risk minimizing action. Deterministic approaches will therefore have limitations for efficient and robust COLAV systems. For a general treatment of different collision risk measures, see for instance [121] and [122]. Relevant maritime COLAV algorithms which incorporate some form of probabilistic risk measure are summarized below.

In [69], an A* search method is applied to collision-free trajectory planning which penalizes high collision risk, COLREGS breaches and trajectory deviation. The collision risk is estimated by calculating collision probabilities using sampling based techniques, considering the positional uncertainty Details on the method for calculating the probabilities are however not given. A* search is also used in [66], to plan a collision-free trajectory through an occupancy grid. Here, occupancy probabilities for obstacles in two-dimensional space are calculated using a numerical approximation, considering their kinematic uncertainty. The search then tries to find a trajectory which minimizes the cost due to non-zero occupancy probabilities, and the Euclidean distance to the goal. COLREGS is not considered here.

In [88], MCS is used to estimate the collision probability beween the own-ship and an obstacle, both with time varying uncertainty. This is done by first forming a Probability Density Function (PDF) at the tracked obstacle position, with a covariance that is the sum of the estimated vessel position covariances. The ratio of samples drawn from this PDF that are inside a collision risk zone, to the total number of samples, is used as a collision probability estimate. The collision probability estimate is then used to decide on replanning collision-free waypoints for the ship to follow, which also adhere to the COLREGS.

[123] plans a collision-free trajectory using Theta* search, based on the current locally estimated sea state, nearby static and dynamic obstacles, and own-ship motion uncertainty. MCS is used to sample dynamic obstacle positions and velocities based on their perception uncertainty, and used together with a precomputed

state transition table to index an estimated mean time between failure for the USV, which is then used to estimate the probability of failure in reaching a motion goal due to collision and local environmental disturbances. This failure probability is then penalized in the search cost function, together with trajectory execution time and COLREGS breaches.

Maneuvering intentions of an obstacle are estimated using a KF in [89]. The intentions are used to calculate the collision probability with obstacles by considering reachable sets. A COLAV system then makes evasive maneuvers when the collision probability exceeds a certain threshold, with no COLREGS consideration.

3.1.3 Contributions

The proposed method is the first probabilistic version of the Scenario-based Model Predictive Control by [70], i.e. Probabilistic SB-MPC (PSB-MPC). Here, the probability of collision with nearby obstacles is estimated, and used to minimize the collision risk on the prediction horizon. A novel contribution is how the collision probabilities are estimated through the use of MCS combined with a KF for the attenuation of statistical noise resulting from few MCS samples. The uncertainty in both position and velocity for the obstacles are considered, as obtained from a tracking system based on the KF. This gives increased situational awareness for the autonomous ship, as the kinematic uncertainty in both position and velocity is an information source not being used in most COLAV systems.

3.1.4 Chapter Overview

The Chapter is organized as follows: In Section 3.2, the own-ship model used for guidance, control and prediction, and the obstacle model used in the tracking system and MPC predictions, are presented. The collision probability framework used here is introduced in Sections 3.3-3.4, whereas the PSB-MPC is introduced in Section 3.5. Results comparing the PSB-MPC against the original SB-MPC are then given in Section 3.6, before conclusions are summarized in Section 3.7.

3.2 Models

3.2.1 Own-ship

A model with 3 degrees of freedom (DOF) is used to describe the horizontal motion of the own-ship in surge, sway and yaw [110], as was described in Section 2.4.4, and restated here as

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

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{w} \tag{3.2}$$

Environmental disturbances are not considered since they are assumed compensated for in the autopilot, and thus $\boldsymbol{w} = 0$. The position and velocity of the own-ship is assumed to be accurately measured, and thus its uncertainty in position and velocity is neglected. The own-ship is steered using LOS (see Section 2.5.1), with a feedback linearizing controller used for surge, and a Proportional-Derivative (PD) controller for the heading. The details on the control system can be found in [124].

3.2.2 Obstacles

The CV, as described in Section 2.4.1, is employed to describe the motion of nearby dynamic obstacles. For an obstacle i, it is restated as

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{F}\boldsymbol{x}_k^i + \boldsymbol{v}_k^i \tag{3.3}$$

$$\boldsymbol{z}_k^i = \boldsymbol{H}\boldsymbol{x}_k^i + \boldsymbol{w}_k^i \tag{3.4}$$

The CV is used with a KF for tracking the obstacles, which results in multivariate Gaussian PDFs $p^i(\boldsymbol{x}, t_k) = \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}_k^i, \boldsymbol{\Sigma}_k^i)$, where $\hat{\boldsymbol{x}}_k^i$ and $\boldsymbol{\Sigma}_k^i$ are the obstacle track estimate and associated covariance, respectively. A deterministic CV with the full state vector available, obtained by omitting the noise terms, is used in the MPC predictions. Note that the CV assumption has limitations in scenarios where maneuvers are expected, as for instance in ship encounters.

3.3 Collision Probability Definition

Probabilities are always relative to the domain of events considered, and a clear definition is therefore needed to avoid amibiquity and confusion. Here, the following events are used to define the collision probability between the own-ship and an obstacle.

$$\begin{aligned}\mathcal{A}_k^i = &\ \textit{A collision occurs between obstacle i and the} \\ &\ \textit{own-ship at some time } t_c \geq t_k.\end{aligned} \tag{3.5}$$

$$\begin{aligned}\mathcal{B}_k^i = &\ \textit{A collision between obstacle i and the own-ship} \\ &\ \textit{does not occur at any time } t_c \geq t_k.\end{aligned} \tag{3.6}$$

which are mutually exclusive. Collision is the breach of the safety zone, which is defined as a circular region with radius d_{safe} around the own-ship. The probability of collision with obstacle i at time t_k then becomes

$$\mathbb{P}_{c,k}^i = \Pr\{\mathcal{A}_k^i\} = 1 - \Pr\{\mathcal{B}_k^i\} \tag{3.7}$$

Note that this definition of collision probability is predictive, as it allows for the collision to happen at any time in the future. The collision probability $\mathbb{P}_{c,k}^i$ is found by integrating the obstacle tracked state PDF $p^i(\boldsymbol{x}, t_k)$:

$$\mathbb{P}_{c,k}^i = \int_{\mathcal{S}} p^i(\boldsymbol{x}; t_k) d\boldsymbol{x} \tag{3.8}$$

where $\mathcal{S} \subset \mathbb{R}^4$ is a region which include all straight line trajectories which make the obstacle cross and recide in the own-ship safety zone at the Closest Point of

Approach (CPA) [124]. This makes the formulation of a compact $\mathcal{S}$ difficult, due to time being an implicit constraint. More specifically, the integration limits on the obstacle velocities depend on both the starting position (x_k^i, y_k^i) of the obstacle, which is uncertain, and the time interval for which the given trajectory starting at (x_k^i, y_k^i) gives an obstacle position inside the own-ship safety zone at CPA.

An illustration of the issue is given in Figure 3.1, where a sample trajectory based on the obstacle uncertainty in position and velocity is shown. The own-ship is shown in blue at the current time following a straight line trajectory, and also at the CPA in dashed blue with the safety zone of radius d_{safe} enclosing it. The obstacle is shown in green at the current time with its 3σ position probability ellipse. If the time to CPA gives an obstacle position on the indicated red part of the trajectory, the trajectory is in $\mathcal{S}$ and may result in a collision.

3.4 Collision Probability Estimation

The calculated collision probability between the own-ship and obstacle i is here filtered recursively using a KF [125]. Probabilities calculated through MCS to approximate the integral (3.8) are used as measurements. The KF is used to attenuate the statistical noise inherent in the MCS with a finite number of samples, and to make use of knowledge about the collision probability from the previous time step. The simple model used in the KF is

$$\mathbb{P}^i_{c,k+1} = \mathbb{P}^i_{c,k} + \bar{v}^i_k \tag{3.9a}$$

$$y^i_k = \mathbb{P}^i_{c,k} + \bar{w}^i_k \tag{3.9b}$$

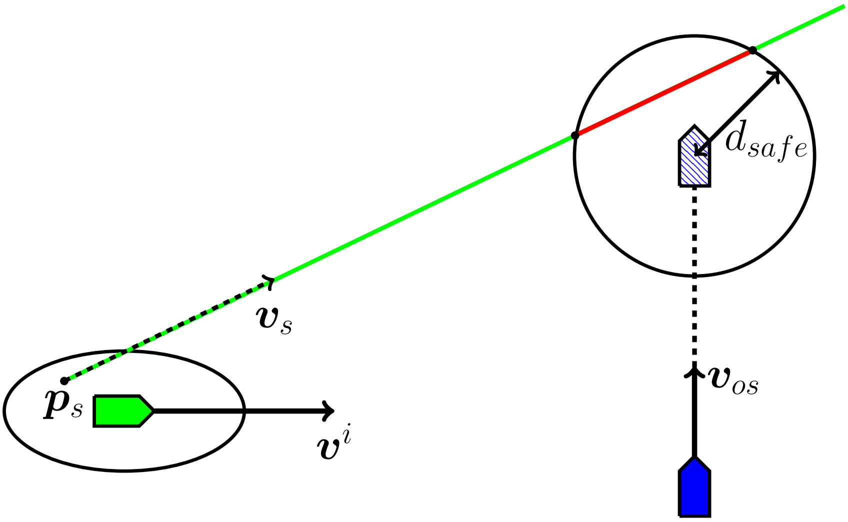


Figure 3.1: Illustration of the problem of determining if an obstacle trajectory is in $\mathcal{S}$. The sampled obstacle trajectory is given by the sampled starting position $\boldsymbol{p}_s$ and velocity $\boldsymbol{v}_s$. The expected obstacle velocity $\boldsymbol{v}^i$ and own-ship velocity $\boldsymbol{v}_{os}$ are also indicated.

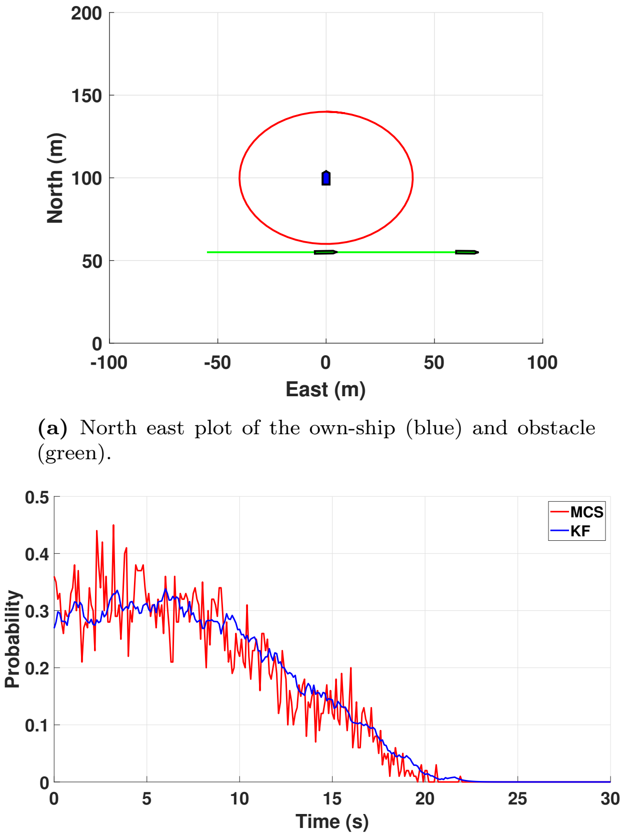
where y is the measurement, $\bar{v} \sim \mathcal{N}(\bar{v}; 0, q_P)$ and $\bar{w} \sim \mathcal{N}(\bar{w}; 0, r_P)$ are the process and measurement noise, respectively. The collision probability measurement y_k^i is obtained through MCS as

$$y_k^i = \frac{1}{N_{MC}} \sum_{s=1}^{N_{MC}} I\{\boldsymbol{x}_s \in \mathcal{S}\} p^i(\boldsymbol{x}_s, t_k) \tag{3.10}$$

where N_{MC} is the number of samples drawn from the obstacle tracked state PDF. This is done by sampling from a standard normal distribution, followed by a transformation through the obstacle state estimate $\hat{\boldsymbol{x}}_k^i$ and Cholesky factorization of the obstacle state covariance $\boldsymbol{\Sigma}_k^i$. The indicator variable $I\{\boldsymbol{x}_s \in \mathcal{S}\} \in \{0, 1\}$ determines if the straight line trajectory sample parameterized by $\boldsymbol{x}_s$ makes the obstacle cross and recide inside the own-ship safety zone at the CPA, assuming that the own-ship also follows a straight line trajectory at time t_k. These assumptions are made in order to have a tractable approach of calculating the collision probability.

In general, an integral estimate obtained through MCS is consistent by the law of large numbers, when the underlying probability model is accurate [126]. In this case, the consistency of the collision probability estimate produced by the MCS and KF are conditioned on the validity of the assumptions of obstacles being modelled as CVs with Gaussian distributed states, the validity of the model (3.9), and the own-ship being assumed to also follow a straight line trajectory at the time of probability calculation. Thus, it is typically a conservative estimate, as factors such as the own-ship and obstacles' intention of avoiding collision and adhering to COLREGS are not accounted for. The estimate will anyhow be used here as an indication of the collision risk. An increased situational awareness by the autonomous ship will be gained regardless, due to the tracking uncertainty being used to have a probabilistic risk picture.

Simulation results for a simple scenario with one obstacle are shown in Figure 3.2. Here, the own-ship is stationary at coordinates $(x, y) = (100, 0)$, whereas an obstacle starting at $(55, -55)$ with assumed known expected position is travelling east with speed 4 m/s. The safety zone is indicated as the red circle. The obstacle is shown at CPA, directly south the own-ship at time $t = 13.75$ s, and also at the end of the simulation. A number of $N_{MC} = 100$ samples are used. The noise covariances are tuned to be $q_P = 0.00005$ and $r_P = 0.001$, based on trial and error and Normalized Innovation Error (NIS) considerations [127]. The filter was initialized with prior probability and variance of 0 and 0.3, respectively. This was partially based on an initial guess, and the assumption that the own-ship starts relatively far from nearby obstacles. The obstacle tracked state has a constant covariance matrix of $\boldsymbol{\Sigma} = \text{diag}([25, 25, 4, 4])$, which causes the collision probability to have a maximum right above 0.3, and decreasing as it passes the own-ship, due to the number of possible straight line trajectories which can cross the safety zone decreases.



(a) North east plot of the own-ship (blue) and obstacle (green).

(b) Collision probability between the own-ship and obstacle estimated with a KF and the “measurements” obtained through MCS.

Figure 3.2: Test scenario.

Thus, from the results, the advantage in using MCS and KF for collision probability estimation is apparent in the reduction of statistical noise at lower computational cost than pure MCS, as fewer samples are needed in the MCS calculations.

3.5 The Probabilistic Scenario-based Model Predictive Control

A Probabilistic variant of the SB-MPC, which was detailed in Section 2.6, modifies the cost function (3.11) to

$$\begin{aligned}\mathcal{H}^l(t_0) = \max_i \Bigg[\sum_{t\in D(t_0)} \mathcal{C}_i^l(t)\hat{\mathbb{P}}_c^{l,i}(t)\text{exp}\left(-\frac{(t+t^l_{cpa}(t)-t_0)}{T_c}\right) + \\ \max_{t\in D(t_0)} \kappa_i \mu_i^l(t)\Bigg] + f(\cdot) + g(\cdot)\end{aligned} \tag{3.11}$$

involving the accumulated probabilistic collision cost over the horizon, exponentially discounted by the time until the potential collision. Thus, the ad hoc risk term $\mathcal{R}_i^l$ from the original SB-MPC is replaced by the collision probability estimate $\hat{\mathbb{P}}_c^{l,i}$. Here, grounding is not considered and thus $g(\cdot) = 0$. The discounting type was chosen mainly due to its simplicity and common use for devaluating events or data, as for instance in the recursive least squares method [128]. The time constant T_c is a tuning parameter. The variable t^l_{cpa} indicates the time until the CPA between the own-ship and obstacle i occurs, calculated at the time t using the corresponding predicted states. As the collision probability is predictive, the time until CPA is added to weight the collision cost by the time of occurrance. Note that this is under the assumption of straight line trajectories at the time of calculation. Moreover, because the collision probabilities are summed, it is assumed that they are independent from one time step to another. This is conservative, as there will be dependencies due to obstacle dynamics and the fact that at maximum one collision between the own-ship and an obstacle can occur in the horizon.

A reasonable alternative to this MPC formulation would be to use a collision risk constraint instead, to retain the risk to a certain limit. However, due to the consistency issues mentioned in the previous section, the constraint limit would be ad hoc. Further, as the collision probability calculations are done in the open loop MPC predictions, with no future feedback accounted for, the limit should not be set too low. Moreover, issues with constraint infeasibility would need to be solved with slack variables to allow practical use.

The advantage of penalizing the collision cost as in the original SB-MPC, is the guarantee of a feasible solution and the intuition of balancing the cost terms. With this approach one is able to choose the maneuver with minimum probabilistic collision cost, which would not be possible in a risk constrained PSB-MPC, as maneuvers are only deemed feasible or not. However, in situations with high collision risk for all control behaviors, the optimally chosen maneuver may still be infeasible in practice and possibly lead to collision, which would require extra handling with this MPC formulation.

3.6 Results

The PSB-MPC and SB-MPC were tested in a head-on scenario, and a congested traffic scenario with multiple obstacles. A sampling interval of $T_s = 0.1\,\mathrm{s}$ was used for the obstacle motion and as the step time in Euler's method to simulate the own-ship motion. A sample time of $T_{s,MPC} = 0.5\,\mathrm{s}$ was used for the MPC. The own-ship was in each scenario planned to follow the straight line going north, with forward speed $U_d = u_d = 9\,\mathrm{m/s}$, starting in the origin. The obstacles were randomly initialized inside a grid of 1000m × 800m, with obstacle velocities varying from $2\,\mathrm{m/s}$ to $v^i_{max} = 9\,\mathrm{m/s}$. The process noise parameter σ_a was uniformly randomly generated between $0\,\mathrm{m/s^2}$ and $0.03\,\mathrm{m/s^2}$ for each obstacle. This simulates scenarios with vessels of lengths around 30 m, driving around cruising speed and below. A measurement covariance of $\boldsymbol{R} = \mathrm{diag}([25, 25])\mathrm{m}^2$ was used to generate obstacle position measurements at 0.4 Hz, with values based on the results using a radar-based tracker in [129].

KFs were used to track each obstacle, with the track estimates fed into the PSB-MPC and SB-MPC. The corresponding covariance estimates were used in the collision probability calculation of the PSB-MPC for the entire time horizon. Thus, the tracked state covariance was not propagated using a deterministic CV in the MPC predictions, but kept constant. This simplification makes the collision probability estimate less conservative, as the covariance would increase when propagated in time. As nearby obstacles will in reality react to the own-ship maneuvers, and measurements of their positions will be used to reduce the uncertainty, this was deemed reasonable. Further, single-point track initiation was used for the obstacle estimates based on the results in [130], where the obstacle initial position and velocity are set to the first measurement and zero, respectively. The a priori obstacle state covariance is set using the KF measurement covariance matrix and maximum velocity U^i_{max}, where it is here assumed that the maximum speed is known a priori, and identical for all obstacles.

To illustrate the importance of considering kinematic uncertainty in a COLAV system, the KF in the tracking system was tuned conservatively in both scenarios with a measurement covariance matrix 25 times larger than the actual covariance $\boldsymbol{R}$, and process noise parameter $\sigma_a = 0.5\,\mathrm{m/s^2}$ for all obstacles. This can be the case when measurements from a radar device becomes unreliable due to extreme weather conditions, and where one in addition wants to account for fast obstacle maneuvers by having an increased process noise. This combined with the chosen track initialization, will cause more uncertain obstacle course estimates, which has been shown to cause problems in deterministic reactive COLAV systems [131].

The collision probability filter was again initialized with prior probability and variance of 0 and 0.3, respectively. The parameters used for each COLAV method are summarized in Table 3.1, and are partially based on [132] and trial and error. The COLAV methods are run every fifth second in the simulations, with 39 control behaviors with one planned evasive maneuver on the prediction horizon.

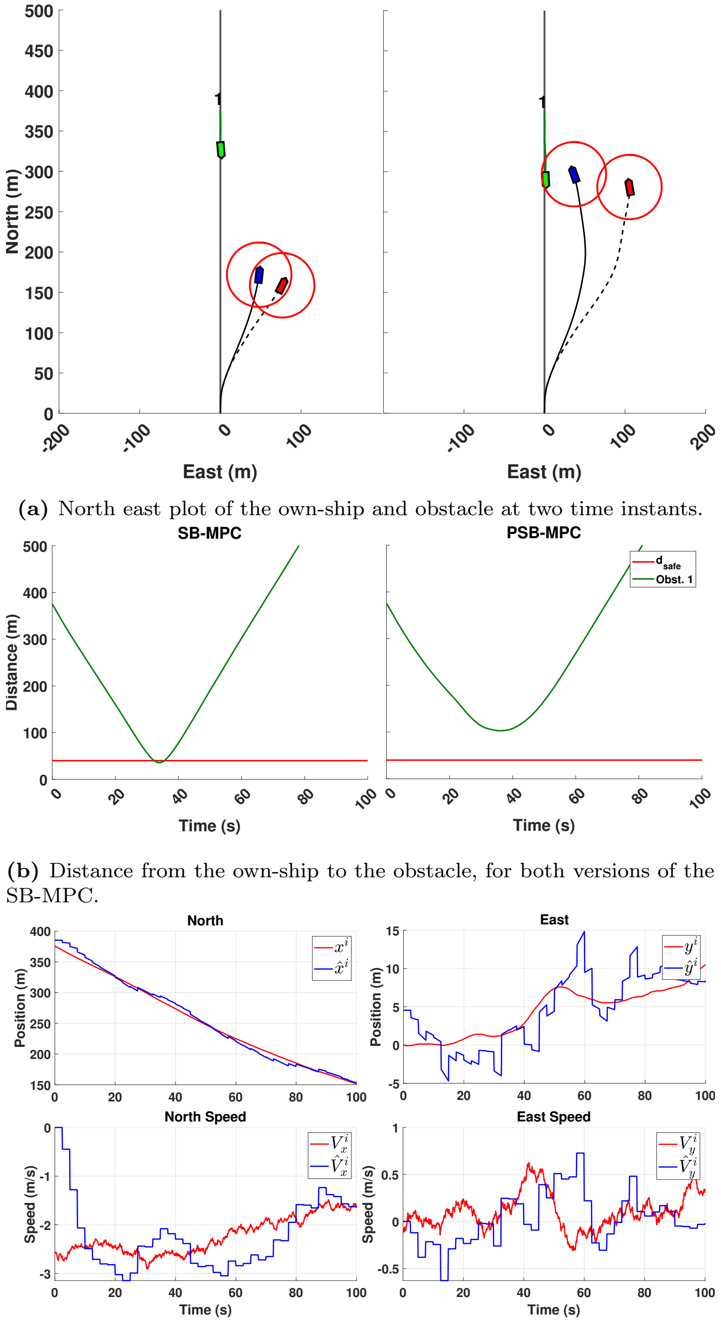
Results comparing the original SB-MPC and the PSB-MPC are shown in Figure 3.3 and 3.4. In each Figure, the first part shows a north east plot for the own-ship with the original SB-MPC (blue boat with black trajectory) and the PSB-MPC (red boat with dashed black trajectory), with the safety zone of radius d_{safe} enclosing them. Also, the obstacles are shown as green boats, with numbered trajectories of different colors. The second part shows the distance from the own-ship to each obstacle, for both COLAV versions, with the safe distance also indicated in red. The obstacle and own-ship sizes are enlarged for visualization purposes. The scenario in Figure 3.3 also include a track plot.

The scenario in Figure 3.3 shows that the PSB-MPC is more risk averse than the original SB-MPC by taking a maneuver compliant with COLREGS rule 14 and 16, with larger safety margins. In contrast, the SB-MPC trusts the obstacle track estimate blindly, which causes a small safety zone violation while performing the COLREGS-compliant maneuver. This is because the SB-MPC believes the obstacle is travelling south with a negative east speed for the first 37 seconds. This is caused by the high obstacle track uncertainty coming from a conservatively tuned KF and the Single-Point track initialization, which causes a velocity variance surpassing $20\,\mathrm{m/s}^2$ in each direction initially.

Figure 3.4 shows that the PSB-MPC is capable of making safe maneuvers in more complex scenarios. Here, the original SB-MPC initially violates rule 14 COLREGS slightly and makes a poor decision to turn port, due to its overconfidence in the obstacle track estimates. This is because obstacles are initialized to zero speed, which makes a port maneuver optimal in the SB-MPC as this gives the minimum ad hoc collision cost. The PSB-MPC again accounts for the estimated uncertainty and decide on a starboard maneuver, due to its minimum probabilistic collision

Table 3.1: Parameters for the COLAV methods.

	SB-MPC	**PSB-MPC**
Parameter	**Value**	**Value**
T	200 s	200 s
$T_{s,MPC}$	0.5 s	0.5 s
d_{safe}	40 m	40 m
κ_i	10.0	10.0
K_U	9.0	9.0
$K_{\Delta U}$	8.0	8.0
$K_{\chi,\text{port}}$	1.8	1.8
$K_{\chi,\text{starboard}}$	1.5	1.5
$K_{\Delta\chi,\text{port}}$	1.2	1.2
$K_{\Delta\chi,\text{starboard}}$	0.9	0.9
T_c	-	6 s
N_{MC}	-	100
q_P	-	0.00005
r_P	-	0.001

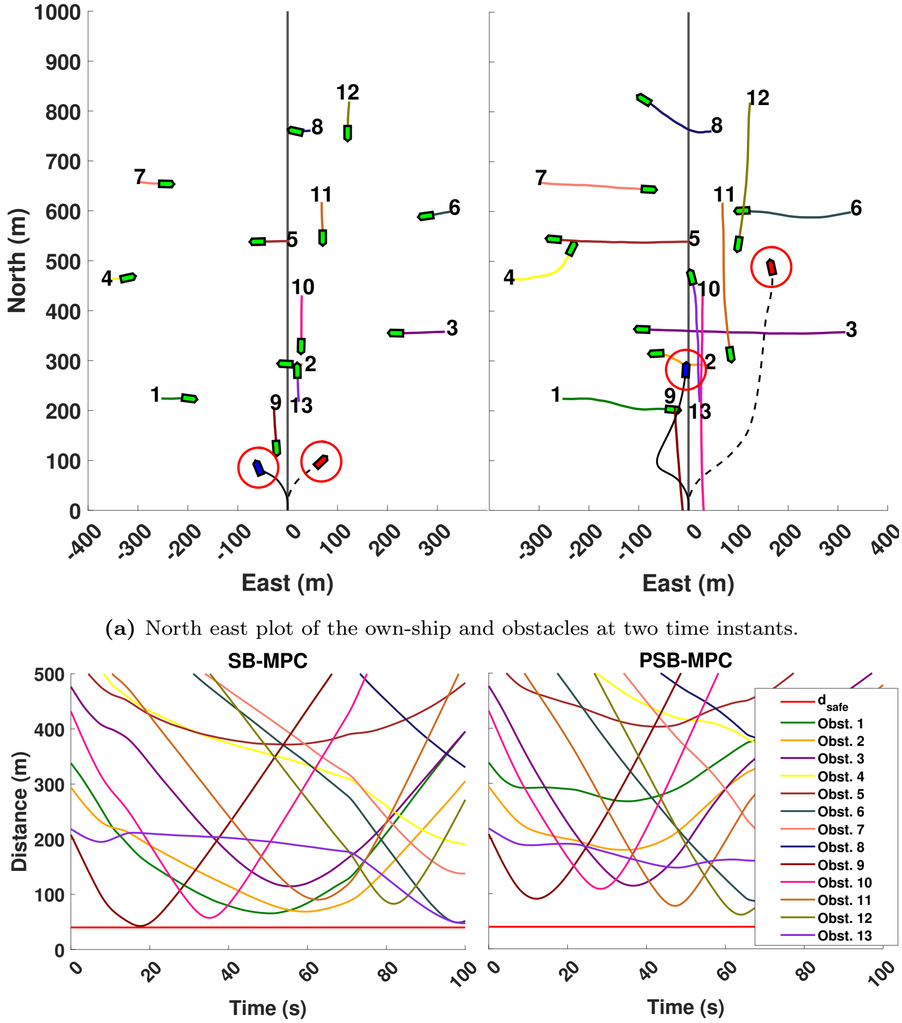


(a) North east plot of the own-ship and obstacle at two time instants.

(b) Distance from the own-ship to the obstacle, for both versions of the SB-MPC.

(c) True obstacle state (red) versus the tracked state (blue).

Figure 3.3: Head-on scenario with 1 obstacle.



(a) North east plot of the own-ship and obstacles at two time instants.

(b) Distance from the own-ship to each obstacle, for both versions of the SB-MPC.

Figure 3.4: Congested traffic scenario with 13 obstacles.

cost.

3.7 Conclusion

The first version of the PSB-MPC attempts to take dynamic obstacle collision probabilities into account, which considers uncertainty in both position and velocity, and is able to take safe decisions in complex scenarios due to increased situational awareness. The probabilistic collision cost gives larger safety margins, but this is conditioned on the quality of the track estimates and collision probability estimates.

Simulation results show that its performance with regards to trajectory following is also on par with the original SB-MPC. Note that the PSB-MPC formulated here is preliminary, and aims to illustrate the benefits of introducing probabilistic risk assessment in a COLAV system.

Moreover, as the method for calculating collision probabilities is simplistic, based on straight line trajectory assumptions, and relatively slow, work is needed in order to make the calculations more efficient and also more consistent and realistic by considering obstacle maneuvers and probabilistic COLREGS compliance. Lastly, the proposed planner should be compared with other methods on different metrics such as energy expenditure and distance to the nominal trajectory, in addition to the safety aspect.

Chapter 4

Second Edition of Collision Risk Assessment in a Probabilistic Scenario-based MPC with Obstacle Intent Consideration

This chapter is based on the following publication

[98] **T. Tengesdal**, T. A. Johansen and E. F. Brekke, "Risk-based autonomous maritime collision avoidance considering obstacle intentions", *23rd International Conference on Information Fusion*, 2020.

and represent one iterative improvement of the CPE introduced in the previous chapter to also account for dynamic obstacle maneuvering. Furthermore, the PSB-MPC is improved to also take obstacle intention uncertainty into account, where a general maritime vessel intent inference framework is introduced.

4.1 Introduction

4.1.1 Motivation and Previous Work

To enable robust and efficient COLAV planning for the autonomous ship, a high degree of situational awareness is needed in the system to allow for deliberate choices of avoidance maneuvers that achieve acceptable collision risk while not being overly conservative. Here, information about the intention of nearby obstacles has high value, as it will enable the COLAV system to take more informed and less conservative decisions by considering future trajectories which reflect the current obstacle intention.

Predicting obstacle trajectories and inferring their intentions will be a key part of robust deliberate COLAV planning algorithms. Different approaches for doing this have been introduced outside the COLAV setting. For COLAV, one is typically interested in the time scale of minutes, and there are different methods employing AIS data for long-term predictions, as in for instance [79], [133]–[135]. The intent of objects are predicted in [102] using a Bayesian approach, when assuming that a finite set of possible endpoints for their trajectories are known. The method constructs so-called bridge distributions for each possible endpoint, and uses a linear motion model conditioned on the endpoint to reduce the trajectory uncertainty from the current object position to its waypoint. The motion model parameters are learned using historic data.

Considering obstacle intentions in collision-free trajectory planning for air traffic and road vehicles has previously been studied [91][92]. For instance in [92], goal hypotheses of road driving obstacles are formed based on the current road topology, and a probabilistic motion model is used to predict their future trajectories conditioned on the hypotheses. However, for maritime applications, it is to the author's knowledge only [69] and [89] that considers nearby obstacles to be agents capable of different maneuvers or intentions. In [69], an A* search method is applied to collision-free trajectory planning which penalizes high collision risk, traffic rule violations and trajectory deviation. An intention motion model is used for nearby civilian vessels, where historical state observations and vessel characteristics are used to output predicted trajectories and classify the vessels as compliant to the COLREGS or not. Information on how this intention model is implemented is limited. The positional uncertainty of an obstacle's predicted trajectory is estimated offline using MCS for a given scene. The predicted trajectories also incorporate reactive obstacle avoidance for the obstacle to avoid the own-ship and other vessels. In [89], the maneuvering intent of obstacles are estimated using a KF. The intents are further used to calculate the collision probability with obstacles by considering reachable sets. A simple COLAV planner is then implemented by making evasive maneuvers when the collision probability is above a certain threshold.

4.1.2 Contributions

In this chapter, the effect of taking probabilistic information of obstacle intentions into account will be showcased. The novelty lies in introducing a generalized framework for obstacle intention inference and applying this in a COLAV planning algorithm. The probability of a finite set of obstacle intentions is considered, when the next waypoint of obstacles is assumed known from some source of information in addition to their degree of COLREGS compliance. This can be the case when vessel-to-vessel communication is employed to get waypoint information of nearby vessels or if local traffic pattern analysis is used. A modified version of the PSB-MPC [97] is proposed, which takes obstacle intentions into account through an enhanced prediction scheme using an OU process as in [79]. The scheme allows obstacles to take different alternative maneuvers at multiple time instants in the horizon. In addition, an updated method of estimating collision probabilities is used in the PSB-MPC, which calculates the collision probability estimates considering piecewise linear segments for the own-ship and obstacle trajectories.

4.1.3 Chapter Overview

The chapter is organized as follows. Section 4.2 outlines the proposed intention modelling framework and includes an example heuristic model. The updated collision probability estimation is detailed in Section 4.3. Section 4.4 gives an overview of the second PSB-MPC edition, with Section 4.5 showing simulation results. Lastly, Section 4.6 gives final remarks on the work in the chapter.

4.2 Intention Probability Framework

4.2.1 Generalized Framework

The index $a = 1, 2, ..., n_a$ is here defined as the obstacle intention. The probability of intention a for obstacle i is denoted $\mathbb{P}^i_a$. The probability is for a finite time interval from t_{k-1} to t_k, where k is the discrete time index, and is assumed to be known as an input to the COLAV planning algorithm in the form of a conditional probability

$$\mathbb{P}^i_a = \Pr\{a|i, \mathcal{I}\}, \quad \text{where} \quad \sum_{a=1}^{n_a} \mathbb{P}^i_a = 1 \tag{4.1}$$

In general, the variable $\mathcal{I}$ contains information on all factors that will affect the obstacle intention. Factors such as the type of obstacle, the grounding hazards, nearby static and dynamic obstacles, weather, the obstacle's current state of perception, and its planned route will all affect this probability. A Bayesian network can here be used to represent the dependence of obstacle intentions on these factors. This is illustrated for an example network in Fig. 4.1. The variable $\mathcal{I}$ is here represented by the seven factors. Note that the network is not unique, and a modelling choice has been made such that intentions are indirectly dependent on for instance the ship type and nearby obstacles through the situation type.

It is a non-trivial task to estimate $\mathbb{P}^i_a$, as the information in $\mathcal{I}$ must be inferred from a subset of all factors involved. Knowledge of this variable is assumed. The purpose of this chapter is not to show how to infer obstacle intentions, but to demonstrate the potential gain of using probabilistic information about this for decision making in a COLAV planning algorithm. However, the framework opens up the possibility to use machine learning methods to learn Bayesian nets for obstacle intention inference by for instance employing historic AIS data.

4.2.2 A Simple Intention Model

Model Assumptions

Three intentions for an obstacle are here considered ($n_a = 3$), corresponding to the obstacle keeping its current course and speed ($a = 1$), taking a starboard ($a = 2$) or port turn ($a = 3$), respectively. It is assumed that the next waypoint WP^i of an obstacle i is known, for instance through usage of historical AIS data. Also, probabilistic information about its tendency to adhere to the COLREGS is known.

Based on the above assumptions and the situation type (ST), the intention probability model used is determined. This is similar to the case in Fig. 4.1 when only the situation type (ST^i), COLREGS compliance (CC^i), nearby obstacles (including the own-ship) and the next waypoint for obstacle i are considered. Moreover, the situation type is assumed independent on the next obstacle waypoint here.

COLREGS compliance is considered when the distance from the obstacle to the own-ship d_{0i} is less than some threshold d_{close} [70]. The ST^i is formulated as a tuple determining if it is an overtaking (OT), head-on (HO) or crossing (CR) scenario, and whether or not the obstacle is the give-way (GW) or stand-on (SO) vessel. These situations can be determined using the position, heading and velocity of the own-ship and obstacle [70]. For the case when $d_{0i} > d_{close}$, i.e. when the ships are outside the defined COLREGS consideration limit, then $ST^i = \emptyset$. The conditional intention probabilities can then be calculated as

$$\mathbb{P}_a^i = \Pr\{a|WP^i, ST^i\}\Pr\{ST^i|CC^i\}\Pr\{CC^i\} \tag{4.2}$$

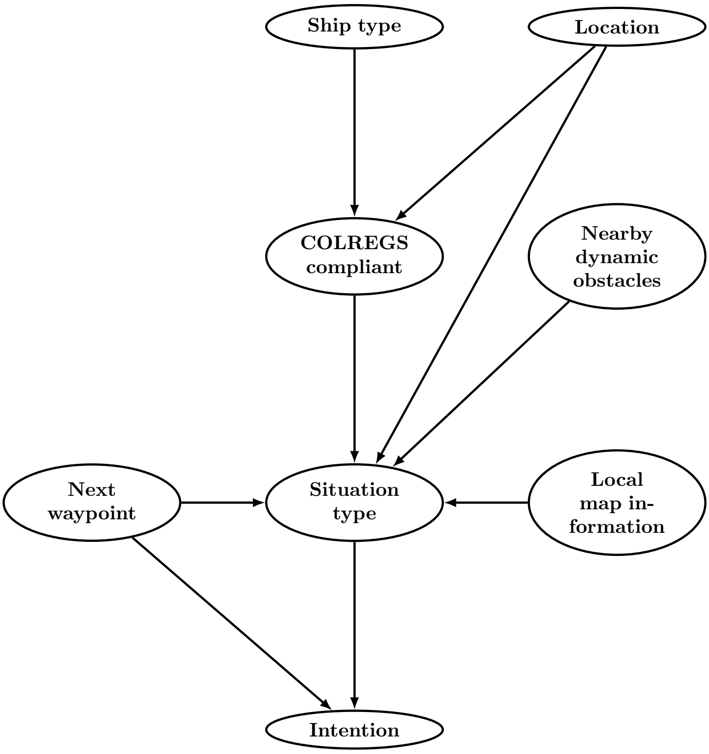


Figure 4.1: Example Bayesian net for intention inference for an obstacle ship, considering seven factors (topmost nodes). The situation type is here either overtaking, head-on or crossing, and whether or not the obstacle is stand-on or give-way vessel. Nearby dynamic obstacles can e.g. be represented by a list containing data structures carrying data on their states.

Table 4.1: Conditional intention probability given WP^i, ST^i and indirectly through CC^i. The situation type is given single letters from A to F to minimize notation space.

CC^i	ST^i	$\Pr\{a\| WP^i, ST^i\}$
$True$	$A = \emptyset$	$\Pr\{a\| WP^i, A\}$
	$B = (OT, SO)$	$\Pr\{a\|B\}$
	$C = (CR, SO)$	$\Pr\{a\|C\}$
	$D = (OT, GW)$	$\Pr\{a\|D\}$
	$E = (HO, GW)$	$\Pr\{a\|E\}$
	$F = (CR, GW)$	$\Pr\{a\|F\}$
$False$	$A = \emptyset$	$\Pr\{a\| WP^i, A\}$

An a priori COLREGS compliance probability $\Pr\{CC^i\}$ will be used here, but could in general be inferred through knowledge on for instance the obstacle ship type and its current location. The next obstacle waypoint is assumed known, but the route towards this waypoint is uncertain. Further, it is assumed that ST^i can be calculated deterministically given CC^i. Thus, the only unknown probability remaining is $\Pr\{a| WP^i, ST^i\}$, which is here specified by ad hoc intention models.

Table 4.1 summarizes the conditional intention cases given WP^i, ST^i and CC^i. In Table 4.1, the intention is indepedent of the next waypoint for $ST^i = B$ to F, as COLREGS compliance is assumed to have the highest priority. If the obstacle is not CC, then the waypoint dependent intention model will be used, i.e. as for $ST^i = A$. Note that this is weighted by the prior CC probability $\Pr\{CC^i\}$.

With the stated assumptions, the intention probability $\mathbb{P}_a^i$ simplifies to

$$\mathbb{P}_a^i = \Pr\{a| WP^i, ST^i\}\Pr\{CC^i\} \tag{4.3}$$

Waypoint Dependent Intention

For $ST = A$, the waypoint information is considered, where a simple ad hoc model for the obstacle intention is developed considering the obstacle course χ^i and the Line of Sight (LOS) vector $\boldsymbol{L}^i$ from the obstacle to its next waypoint. This is illustrated for an example head-on scenario in Fig. 4.2.

The angle θ is defined as the angular difference between the LOS-vector to the waypoint and the obstacle course, and used to define the intention probability of straight line motion, starboard or port maneuvers. The probability for the obstacle to keep its course $a = 1$ is then assumed on the form

$$\Pr\{a = 1| WP^i, A\} = \alpha_{1,WP} e^{-c_1|\theta|} + \alpha_{2,WP} \tag{4.4}$$

where $c_1 > 0$ is a parameter to decide the decrease/increase in probability as the obstacle turns towards the end point. The parameters $\alpha_{1,WP}$ and $\alpha_{2,WP}$ determines the maximum probability for the waypoint dependent intention a. For a starboard or port maneuver at some time $t_{turn} \geq t_0$, where t_0 is the current time, the intention probabilities are assumed to be

$$\Pr\{a = 2|\,WP^i, A\} = \begin{cases} \alpha_{1,\,WP}(1 - e^{-c_1\theta}) \\ \;+\alpha_{2,\,WP} & \text{if } \theta \geq 0. \\ \alpha_{3,\,WP}, & \text{otherwise.} \end{cases} \tag{4.5}$$

$$\Pr\{a = 3|\,WP^i, A\} = \begin{cases} \alpha_{1,\,WP}(1 - e^{c_1\theta}) \\ \;+\alpha_{2,\,WP} & \text{if } \theta \leq 0. \\ \alpha_{3,\,WP}, & \text{otherwise.} \end{cases} \tag{4.6}$$

respectively. The weigthing parameters satisfy

$$\alpha_{1,\,WP} + 2\alpha_{2,\,WP} + \alpha_{3,\,WP} = 1 \tag{4.7}$$

If the probability of a maneuver to a given side is high due to the angle θ being large, the probability for a maneuver to the other side is set to a small constant value $\alpha_{3,\,WP}$. The intention probabilities in (4.4) - (4.6) can be verified to sum to unity.

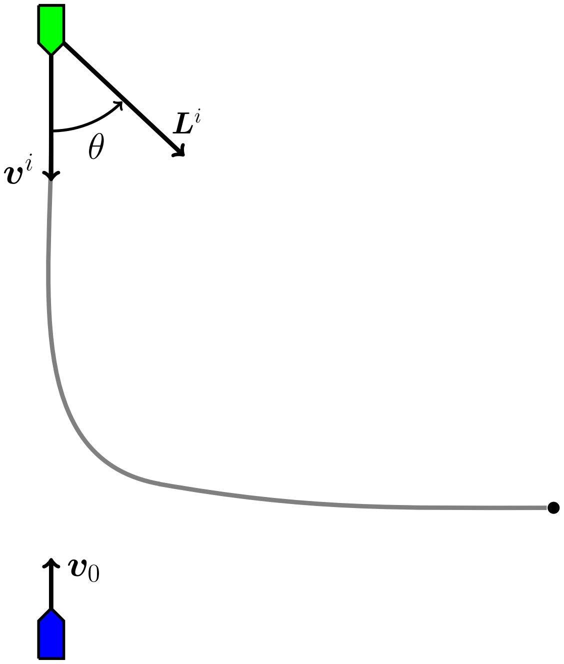


Figure 4.2: A head-on scenario with obstacle i in green and own-ship in blue. Their velocity vectors $\boldsymbol{v}^i$ and $\boldsymbol{v}_0$ are also shown. The unknown ground truth planned obstacle trajectory is shown in grey, with its next waypoint assumed known to the own-ship as the black dot.

Stand-on Dependent Intention

If the obstacle is stand-on vessel in either crossing or overtaking, it is assumed to be COLREGS-compliant and keep its current course and speed with a constant high probability, and thus the intention probabilities for this ST are

$$\Pr\{a|B\} = \Pr\{a|C\} = \{\alpha_{1,B}, \alpha_{2,B}, \alpha_{3,B}\} \tag{4.8}$$

where $\alpha_{a,ST}$ for $a = 1, 2, 3$ are parameters which sum to one for any ST, and $\alpha_{a,B} = \alpha_{a,C}$ by assumption. The parameter $\alpha_{1,B}$ is typically chosen higher than 0.9 due to the COLREGS compliance assumption.

Give-way Dependent Intention

When the obstacle is the give-way vessel, the intention probability is assumed to follow a model dependent on the distance d_{0i}. For the overtaking situation $ST = D$, the model is assumed of the form

$$\Pr\{a = 1|D\} \propto \alpha_{1,D} e^{c_2(d_{0i} - d_{close})} \tag{4.9}$$

$$\begin{aligned}\Pr\{a = 2|D\} \propto\ & (1 - \alpha_{2,D})(1 - e^{c_2(d_{0i} - d_{close})}) \\ & + \alpha_{2,D}\end{aligned} \tag{4.10}$$

$$\begin{aligned}\Pr\{a = 3|D\} \propto\ & (1 - \alpha_{3,D})(1 - e^{c_2(d_{0i} - d_{close})}) \\ & + \alpha_{3,D}\end{aligned} \tag{4.11}$$

where $c_2 > 0$ is a parameter to tune the the obstacle intention a decrease/increase. For the head-on situation $ST = E$, the probabilities are assumed of the form

$$\Pr\{a = 1|E\} \propto \alpha_{1,E} e^{c_2(d_{0i} - d_{close})} \tag{4.12}$$

$$\begin{aligned}\Pr\{a = 2|E\} \propto\ & (1 - \alpha_{2,E})(1 - e^{c_2(d_{0i} - d_{close})}) \\ & + \alpha_{2,E}\end{aligned} \tag{4.13}$$

$$\Pr\{a = 3|E\} \propto \alpha_{3,E} \tag{4.14}$$

Lastly, for the crossing-on situation $ST = E$, the probabilities are assumed of the form

$$\Pr\{a = 1|F\} \propto \alpha_{1,F} e^{c_2(d_{0i} - d_{close})} \tag{4.15}$$

$$\begin{aligned}\Pr\{a = 2|F\} \propto\ & (1 - \alpha_{2,F})(1 - e^{c_2(d_{0i} - d_{close})}) \\ & + \alpha_{2,F}\end{aligned} \tag{4.16}$$

$$\Pr\{a = 3|F\} \propto \alpha_{3,F} \tag{4.17}$$

In head-on and crossing, a maneuver towards starboard will be given the highest probability, and thus $\alpha_{2,E}$ and $\alpha_{2,F}$ will be chosen higher than $\alpha_{1,E}$ and $\alpha_{1,F}$, respectively. For the overtaking case, starboard and port maneuver intentions will be given equal weight, thus $\alpha_{2,D} = \alpha_{3,D}$, also chosen higher than the weight $\alpha_{1,D}$

on keeping the current course. To avoid overconfidence in the decision of the MPC, all parameters α_{a,ST^i} are chosen strictly higher than zero. Note that only the intention of a given maneuver is quantified through these models, not the time of occurance and how much change in course and speed the maneuver will have. The intention probabilities specified in the models (4.9) - (4.11), (4.12) - (4.14) and (4.15) - (4.17) sum to one after multiplication with the inverse of the normalization constant $\sum_{a=1}^{n_a} \Pr\{a = |ST^i\}$.

This intention probability model is ad hoc, makes many assumptions and neglects a significant amount of factors. However, the point is merely to display the effect on the performance of a COLAV planning algorithm when taking such information into account. When considering the example set of three maneuvering intentions, the model can be compared to the Interacting Multiple Model (IMM) approach commonly used in tracking [136], where probabilities of a target behaving as per a set of different modes is estimated based on kinematic measurements. The difference from the IMM approach is, however, that the DBN for intention inference proposed here can condition on a diverse set of information types, such as the current obstacle state, its ship type, its next waypoint or destination and inclination towards adhering to the COLREGS. The DBN could in general represent more complex obstacle intention states, e.g. the degree of COLREGS compliance, whether or not the obstacle has higher priority than other ships, what it considers as safe distance to nearby vessels and what it considers ample time.

4.3 Updated Collision Probability Estimation

The work in [97] introduced a method for estimating the collision probability between the own-ship and an obstacle using the obstacle uncertainty in position and velocity in a MCS and KF scheme. The method assumes straight line trajectories for both vessels for the entire time horizon at the time of evaluation, which will be overly conservative as the vessels will most likely make maneuvers in the future to reduce collision risk. This chapter further extends this method by calculating the collision probability on piece-wise linear segments along the vessel trajectories, to reduce the conservativeness of the estimate by exploiting the intention models in Section 4.2. A discretization time step of $T_{seg} = t_j - t_{j-1}$, for two time instants t_j and t_{j-1}, is used, typically larger than the prediction time step in the MPC. The concept is illustrated in Fig. 4.3 for a maneuvering own-ship and an obstacle.

Piece-wise linear segments from t_{j-1} to t_j along the vessel trajectories are created, with vessel velocities given by the average over the current segment and direction along the linear segment. These segments are then used as in the original method [97] to estimate the collision probability, with one alternation: If the time until CPA t_{cpa} is less than t_j for two linear segments in consideration, the vessel states at t_j instead of t_{cpa} is used in the MCS part. This is done to constrain the collision probability evaluation to only consider positions on the discretized vessel trajectories. The obstacle covariance at the time t_{j-1} is used in the MCS [97]. This alteration naturally requires retuning of the noise parameters r_P and q_P in the estimation.

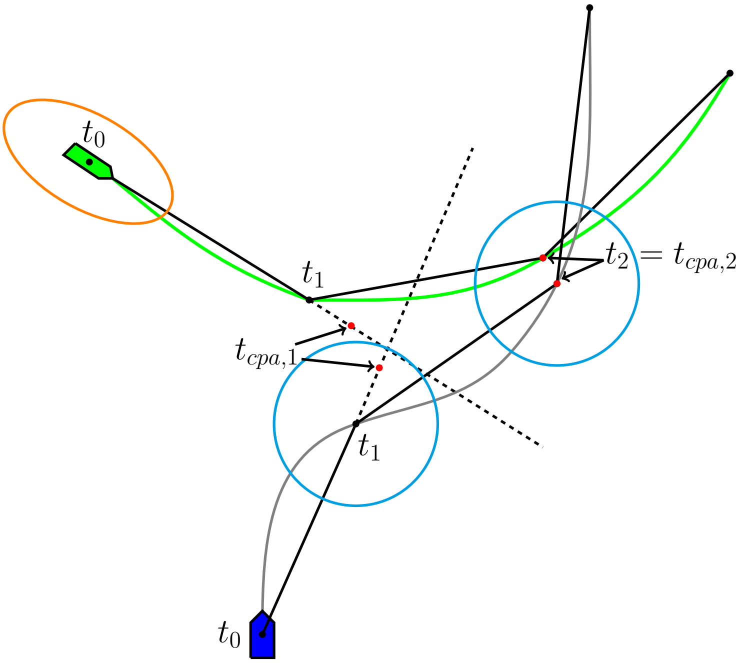


Figure 4.3: Illustration of the updated collision probability estimation method. Note that T_{seg} is chosen large to make the methodology more clear. Further note that the CPA drawn in the sketch are examplatory. The obstacle (green) and own-ship (blue) trajectories are shown in green and gray, respectively. The obstacle 3σ uncertainty ellipse at t_0 is shown in orange. The black dots indicate the vessel positions at times t_0, t_1 and t_2. The red dots indicate the vessel positions along the linear segments at CPA, which for the two segments in the own-ship trajectory gives two CPA times $t_{cpa,1}$ and $t_{cpa,2}$. Since $t_{cpa,1} > t_1$, the safety zone (cyan) around the own-ship is centered to the own-ship position at t_1, where the vessel positions at t_1 is used as basis for the collision probability estimation. For the second pair of segments, $t_{cpa,2} \leq t_2$ and the safety zone is thus centered over $t_{cpa,2}$.

4.4 Probabilistic Scenario-Based Model Predictive Control

The second version of the PSB-MPC algorithm [97] uses the updated CPE in its probabilistic collision risk assessment, and also facilitates the use of obstacle intention information through probabilities of starboard, constant course and port maneuvers. This is accomodated by using an OU model for predicting multiple alternative maneuvering scenarios for an obstacle in the enhanced MPC prediction scheme outlined below. See Section 3.5 in the previous chapter for information on the first PSB-MPC version.

4.4.1 Enhanced Prediction Scheme

The obstacle motion is predicted using a stochastic OU process. Details on the model was outlined in Section 2.4.2. A state vector $\boldsymbol{x}_k^i = [x_k^i, y_k^i, V_{x,k}^i, V_{y,k}^i]^T$ is used to describe the kinematics of obstacle i at time index k, and is given by the position and velocity components in a planar north and east reference system, respectively.

To make use of knowledge about different obstacle intentions in the MPC, the predictions must allow for port and starboard turns at different time instants in the horizon, in addition to the original straight line motion prediction. The number of turns and time of each turn for an obstacle is here determined by considering the time remaining until and distance at the estimated CPA, t_{cpa} and d_{cpa}, respectively.

If there is no predicted collision at t_{cpa}, i.e. the obstacle does not enter the safety zone of the own-ship with radius d_{safe} [97], then the alternative maneuvers are spaced evenly with t_{ts} apart throughout the horizon of length T. If there is a predicted collision, the time t_{cpa} determines how many alternative maneuvers are accounted for. Thus, the final turn time of the obstacle is given by

$$t_{ft} = \begin{cases} t_0 + T, & \text{if } d_{cpa} > d_{safe}. \\ t_0 + t_{cpa}, & \text{if } d_{cpa} \leq d_{safe} \ \& \ t_{cpa} > t_{ts}. \\ t_0, & \text{if } d_{cpa} \leq d_{safe} \ \& \ t_{cpa} \leq t_{ts}. \end{cases} \tag{4.18}$$

The case where $d_{cpa} \leq d_{safe}$ and $t_{cpa} > t_{ts}$ is illustrated for a head-on scenario in Figure 4.4. The two vessels are here predicted to collide at the blue cross (CPA).

The maneuvers are implemented by changing the velocity $\boldsymbol{v}^i_{OU,k}$ of the OU model at the turn time. In general, a finite number of different course changes can be used. In Fig. 4.4, three changes are used. As the explosion of obstacle uncertainty is avoided by using an OU process for prediction, its uncertainty can be divided into pieces given by the amount of different prediction scenarios specified. The spacing between turns and amount of different course changes should be set such that the union of the obstacle uncertainty in all prediction scenarios cover all possible paths. This is also a trade off with the computational effort required in the predictions.

4.4.2 Multiple Sequential Avoidance Maneuvers

To prevent conservative solutions, the second PSB-MPC edition is allowed to make n_M sequential avoidance maneuvers in the prediction horizon. The start time of each avoidance maneuver is selected as follows. The first avoidance maneuver is made at t_0. The subsequent maneuvers are made immediately after the closest obstacle in the current collision situation makes its maneuver at $t_0 + t_{ts}$, or immediately after t_{cpa} with the closest obstacle. When the closest obstacle is passed, the subsequent maneuvers are found in the same manner using the next relevant close obstacle. The spacing between the possible own-ship maneuvers will thus be of minimum t_{ts} seconds. Thus, the control behavior l now consist of the avoidance maneuvers $\left[(U^l_{m,1}, \chi^l_{m,1}), ..., (U^l_{m,n_M}, \chi^l_{m,n_M})\right]$. For the optimal control behavior l^*, the first maneuver $(U^{l^*}_{m,1}, \chi^{l^*}_{m,1})$ is the MPC output. The PSB-MPC will typically re-evaluate its optimal strategy at regular intervals, e.g. every 5 seconds.

4.4.3 Cost Function

To account for multiple obstacle intentions and own-ship maneuvers in the prediction horizon, the PSB-MPC cost function for the own-ship control behavior l is

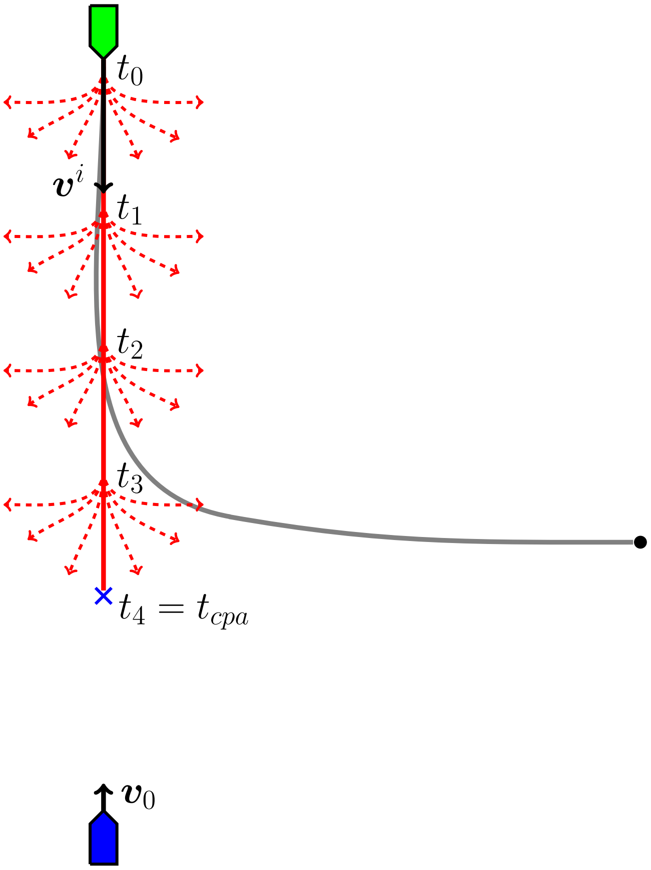


Figure 4.4: The previously shown head-on scenario with obstacle i in green and own-ship in blue. Their velocity vectors $\boldsymbol{v}^i$ and $\boldsymbol{v}_0$, respectively, are also shown. The prediction scheme allows for the obstacle to make different types of port and starboard maneuvers indicated at the discrete times t_0 to t_3 in this case, in addition to the original straight line prediction. The blue cross indicates the obstacle position at t_{cpa}.

modified to

$$\begin{aligned}\mathcal{H}^l(t_0) = & \max_i \sum_{a=1}^{n_a} \mathbb{P}_a^i(t_0) C_a^{l,i} + g(\cdot) + \\ & \frac{1}{n_M} \sum_{M=1}^{n_M} f(U_{m,M}^l, U_{m,M-1}^l, \chi_{m,M}^l, \chi_{m,M-1}^l) + \\ & \frac{1}{n_M - 1} \sum_{M=2}^{n_M} h(\chi_{m,M}^l, \chi_{m,M-1}^l, t_M - t_{M-1})\end{aligned} \tag{4.19}$$

where

$$C_a^{l,i} = \sum_{s=1}^{n_{ps}^i(a)} \frac{w^{i,s}}{n_{ps}^i(a)} \max_{t \in D(t_0)} \left[\mathcal{C}_i^{l,s}(t) \hat{\mathbb{P}}_c^{l,i,s}(t) + \kappa_i \mu_i^{l,s}(t) \right] \tag{4.20}$$

is the average cost for all predicion scenarios n_{ps}^i involving intention a for obstacle i. For the case in Fig. 4.4, $n_{ps}^i(1) = 1$ and $n_{ps}^i(2) = n_{ps}^i(3) = 12$. The weights $w^{i,s}$ are given as

$$w^{i,s} = \begin{cases} \Pr\{CC^i\}, & \text{if obstacle } i \text{ is CC in } s \\ 1 - \Pr\{CC^i\}, & \text{otherwise} \end{cases} \tag{4.21}$$

The check whether the obstacle is CC in a prediction scenario is done by determining whether it breaches COLREGS given that the own-ship keeps its course. The terms $\mathcal{C}_i^{l,s}$, $\hat{\mathbb{P}}_c^{l,i,s}(t)$ and $\kappa_i \mu_i^{l,s}(t)$ are the collision risk cost, collision probability and COLREGS penalization term, for control behavior sequence l, obstacle i in its prediction scenario s. Unlike [97], no discounting is made on the collision cost because this is done implicitly in the collision probability calculation when propagating the obstacle uncertainty in time. The set $D(t_0)$ contains all time samples in the prediction horizon.

The control reference cost $f(\cdot)$ is summed over all own-ship avoidance maneuvers in the control behavior l, where $U_{m,0}^l = U_{m,last}$ and $\chi_{m,0}^l = \chi_{m,last}$ are the offsets from the previous optimal MPC output. For $n_M > 1$, a new control reference cost has been introduced in $h(\cdot)$, which is given by

$$h(\chi_1, \chi_2, t) = \begin{cases} K_{sgn} \exp\left(-\frac{t}{T_{sgn}} \right), & \text{if } \operatorname{sign}(\chi_1) \neq \operatorname{sign}(\chi_2) \\ 0, & \text{otherwise} \end{cases} \tag{4.22}$$

and penalizes chattering behavior in course throughout the horizon, discounted by the time $t_M - t_{M-1}$ between maneuvers, with tuning parameters K_{sgn} and T_{sgn}. See [97] for more information on the cost terms and their parameters.

4.5 Simulation Results

The PSB-MPC is compared against the original SB-MPC described in Section 2.6 with one avoidance maneuver in two different scenarios. The performance of the

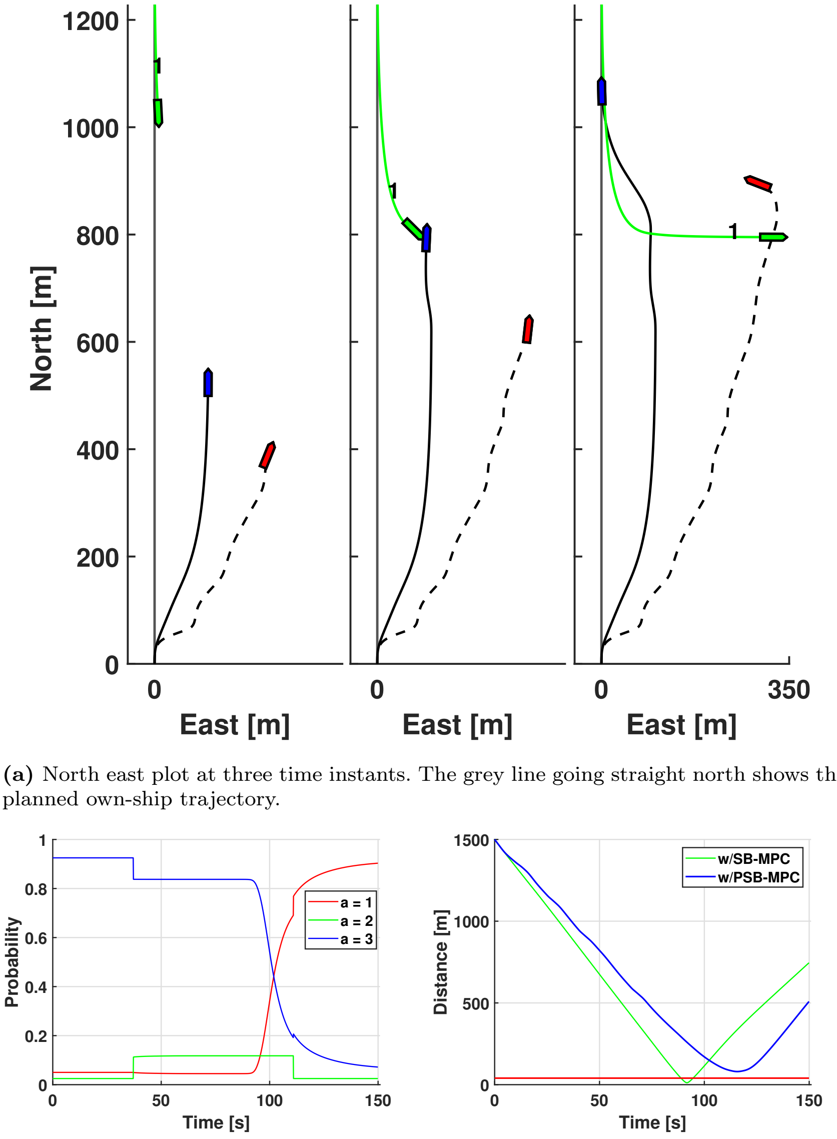
PSB-MPC will be gauged when different prior probabilities on the obstacle CC is used, when it knows the next waypoint for each obstacle.

Measurements for the obstacles are generated using a covariance $\boldsymbol{R} = \text{diag}(25, 25)\text{m}^2$. Obstacles of lengths $30 - 100\,\text{m}$ are considered. The obstacles are initialized to the ground truth but with a single-point initialized covariance, and otherwise tracked using Kalman-filters as in [97], where the filter measurement covariance and process covariance parameters are chosen as $\boldsymbol{R}^{KF} = 2\boldsymbol{R}$ and $\sigma_a^{KF} = 0.5\,\text{m/s}^2$, respectively. This represents a conservative KF which expect fast maneuvers for the obstacle, and therefore gives higher track uncertainty, as could be the case for a real time tracking system with model mismatch and/or degraded sensor performance.

Important parameters for the SB-MPC and PSB-MPC, segment-wise collision probability estimation and intention models are summarized in Table 4.2. The parameter d_{close} is chosen larger than in [97] because it here also determines the model switching in the ad hoc intent inference. Considering their common parameters, the two versions are tuned equally. For the PSB-MPC, course changes of 30, 60 and 90 degrees are considered for the obstacle predictions. Two sequential avoidance maneuvers are considered, where the first one samples 39 control behaviors as in [70], and the second maneuver samples a subset of those: $U_{m,2} \in \{1, 0.5\}$ and $\chi_{m,2} \in \{-90, -45, 0, 45, 90\}$ degrees, to limit the computational effort. For the MPC predictions, the initial typical velocity $\boldsymbol{v}^i_{OU,k}$ is set to the current velocity estimate of the obstacle.

Results for the two MPC versions are shown below. The own-ship using SB-MPC and PSB-MPC are shown in blue and red with continuous black and dashed black trajectories, respectively. The obstacles are shown in green with their corresponding index number. The obstacle intention probabilities and distance to the obstacles are also shown. In all scenarios, the PSB-MPC knows the next waypoint for the obstacles but not their route.

In the head-on scenario in Fig. 4.5, the a priori CC probability is set to $\Pr\{CC^i\} = 0.1$, indicating low trust in the obstacle to follow COLREGS. The scenario is also set up such that the obstacle breaches COLREGS through a port maneuver. Due to the intent inference being used as input to the PSB-MPC planning algorithm, it predicts a port maneuver from the waypoint information and therefore initially slows a bit down and makes a large CC starboard maneuver. The original SB-MPC does not forsee this, and risks collision with the obstacle. This is also contributed to the uncertainty in the track estimates, which is not considered in the original SB-MPC. The switching in intent probabilities in Fig. 4.5(b) occur due to the obstacle entering the COLREGS consideration limit and when it passes the own-ship. Some oscillation in the course offset for the PSB-MPC is also seen partially due to this. Because of a small a priori CC probability, the waypoint dependent intention will dominate. Note that alternative behaviors could also be feasible, such as a port turn to minimize collision risk at the cost of breaching COLREGS. A weighting between risk aversion and COLREGS compliance must therefore be made.



(a) North east plot at three time instants. The grey line going straight north shows the planned own-ship trajectory.

(b) Intention probabilities.

(c) Distance to obstacle with the own-ship safety zone marked as the red line.

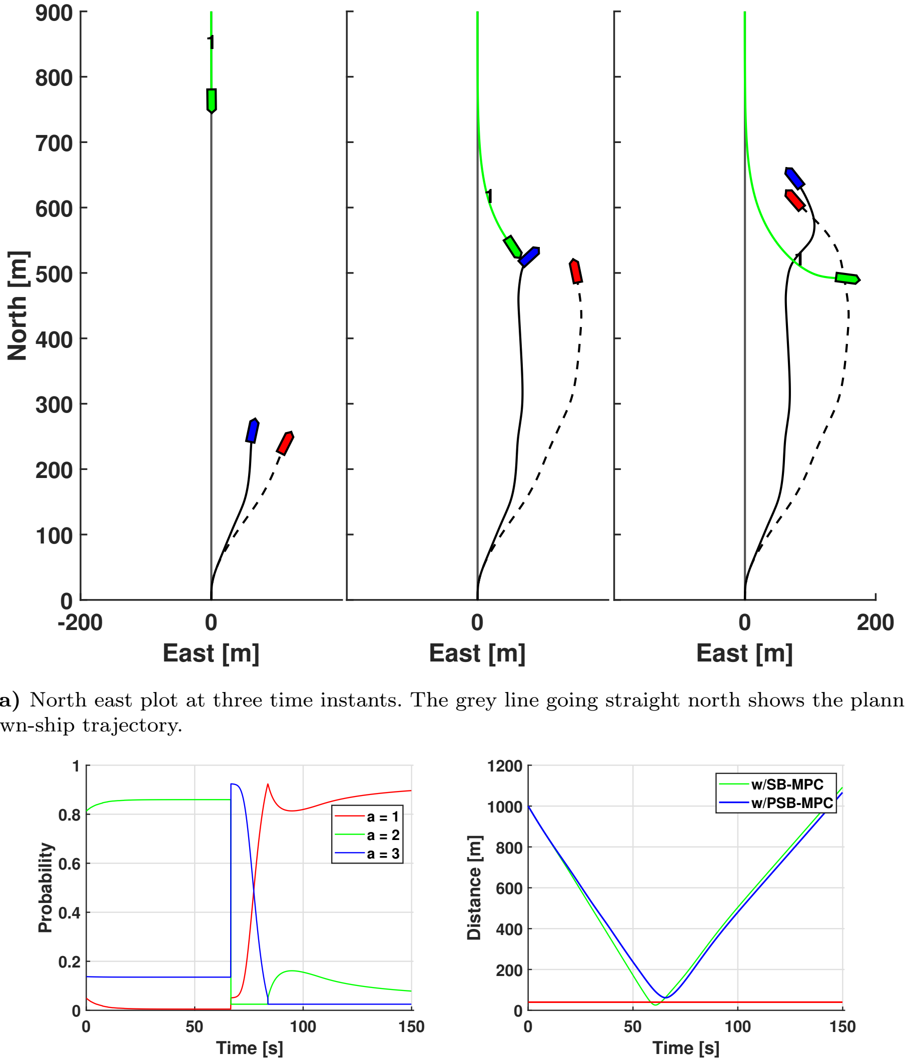
Figure 4.5: Non-CC obstacle in head-on scenario with more correct probabilistic information.

Results for the same head-on scenario when having a false high trust in CC for the obstacle, by setting $\Pr\{CC^i\} = 0.9$, are shown in Fig. 4.6. In this case the PSB-MPC makes a smaller CC starboard maneuver, as it expects the obstacle also will act accordingly. However, as the probability of non-CC is 0.1, the waypoint dependent intent causes a higher port intention probability than the probability of keeping the current course. Thus, a more conservative starboard maneuver is made for the PSB-MPC than the SB-MPC, which gives a higher safety margin when the obstacle makes the unforseen port turn. The intention probabilities switches one time during the simulation as ST^i changes from E to A when the own-ship is passed by. As no boundary conditions are enforced on the probabilities, discrete jumps can occur. The increase in starboard turn intention at the end is due to the angle θ going positive at some point during the port turn.

Lastly, results for a combined head-on and crossing scenario are shown in Fig. 4.7, where a non-CC obstacle head-on to the own-ship, basically identical to the case in Fig. 4.5, makes a port maneuver too late, and where a CC obstacle with assumed $\Pr\{CC^2\} = 0.9$ makes a late give-way maneuver in a crossing situation. The assumed a priori CC probability for each obstacle $i = 1, 2$ is set to 0.1 and 0.9, respectively. The performance of the SB-MPC again causes large hazard with obstacle 1, as only one deterministic straight line prediction based on the current track estimates of the obstacles are considered, while the PSB-MPC is able to utilize

Table 4.2: Parameters for the different methods and models.

	SB-MPC	**PSB-MPC**
Parameter	**Value**	**Value**
r_P	-	0.001
q_P	-	0.017
T_{seg}	-	1 s
d_{close}	1000 m	1000 m
n_a	-	3
n_M	1	2
t_{ts}	-	25 s
K_{sgn}	-	5
T_{sgn}	-	$4t_{ts}$
σ_x	-	$0.8\,\mathrm{m/s}^2$
σ_{xy}	-	$0\,\mathrm{m/s}^2$
σ_y	-	$0.8\,\mathrm{m/s}^2$
γ	-	$[0.1, 0.1]^T$
$\alpha_{:,WP}$	-	$\{0.875, 0.05, 0.025\}$
$\alpha_{:,B}$	-	$\{0.9, 0.05, 0.05\}$
$\alpha_{:,D}$	-	$\{0.05, 0.475, 0.475\}$
$\alpha_{:,E}$	-	$\{0.05, 0.9, 0.05\}$
$\alpha_{:,F}$	-	$\{0.05, 0.9, 0.05\}$



(a) North east plot at three time instants. The grey line going straight north shows the planned own-ship trajectory.

(b) Intention probabilities.

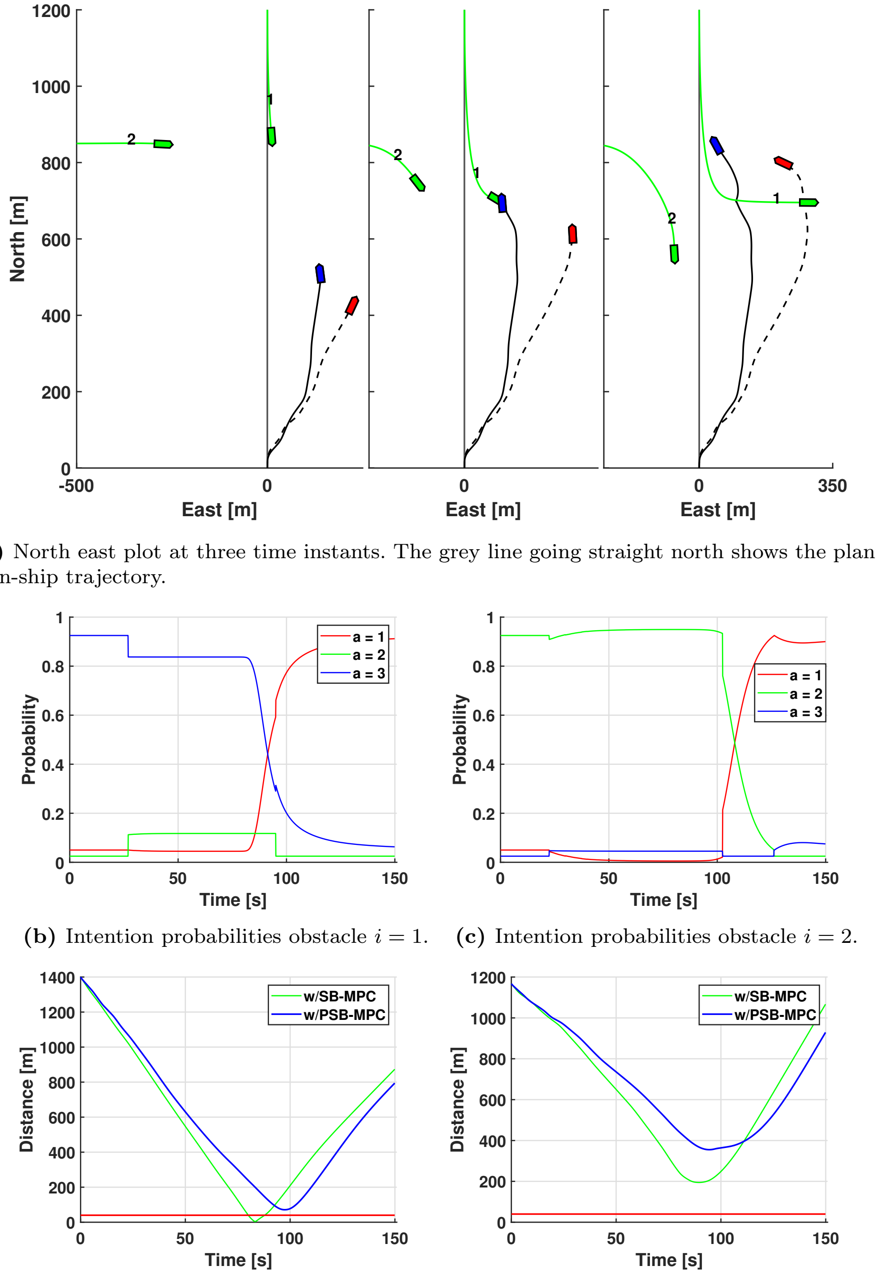
(c) Distance to obstacle with the own-ship safety zone marked as the red line.

Figure 4.6: Non-CC obstacle in head-on scenario with wrong probabilistic information.

the extra probabilistic information to make a larger proactive avoidance maneuver starboard. The algorithm could also be tuned to make the own-ship slow down and then take the turn, or make a port turn to minimize the collision risk with obstacle 1. As obstacle 2 is almost assumed fully CC, it could be safer to violate COLREGS and pass the obstacles with a port maneuver. The next waypoint for obstacle $i = 1$ again lies to the east, and the corresponding port intention dominates due to the low CC trust. Switches occur two times for the obstacles as they enter the COLREGS consideration limit and are passed by. The next waypoint for obstacle $i = 2$ lies in the south, and an increase in port intention is here seen due to the angle θ switching sign.

4.6 Conclusion

The PSB-MPC extended to also consider probabilistic obstacle intentions and multiple sequential avoidance maneuvers, gives increased situational awareness and thus improved decision making. This is here shown for an ad hoc intention inference model, which is only used for illustration. As the complexity of the MPC predictions are significantly increased, further simulation studies are needed to investigate optimal tuning parameters and prediction scenario configurations. Existing methods should be used on AIS data to estimate the parameters of the OU process used in predictions. Furthermore, the robustness of the PSB-MPC for various obstacle intention probability configurations should be studied in more detail.



(a) North east plot at three time instants. The grey line going straight north shows the planned own-ship trajectory.

(b) Intention probabilities obstacle $i = 1$.

(c) Intention probabilities obstacle $i = 2$.

(d) Distance to obstacle $i = 1$ with the own-ship safety zone marked as the red line.

(e) Distance to obstacle $i = 2$ with the own-ship safety zone marked as the red line.

Figure 4.7: Combined scenario.

Chapter 5

Third Edition of Collision Risk Assessment in a Probabilistic Scenario-based MPC Using The Cross-Entropy Method

This chapter is based on the publication

[94] **T. Tengesdal**, E. F. Brekke and T. A. Johansen, "Ship collision avoidance utilizing the cross-entropy method for collision risk assessment", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11148-11161, 2022.

which outlines a new approach to collision probability estimation. The CPE outlined in the previous chapters assumed constant velocity for the vessels involved over a time step or more. Furthermore, it suffers from the curse of dimensionality when low collision probabilities are considered, as a higher sampling number is required. In this chapter, the developed Cross-Entropy (CE) based method improves on CPEs developed in the previous two chapters by using adaptive importance sampling [137], which is able to produce accurate estimates at low computational cost. With accuracy it is here meant the property of low variance and unbiased estimates. Through its adaptivity, it can better handle rare collision events with low probabilities.

5.1 Introduction

5.1.1 Motivation

As COLAV planning algorithms typically use information about nearby obstacles from e.g tracking systems, it needs to account for the uncertainties present in these information sources in order to make robust and safe decisions. However, as shown in previous chapters, the associated uncertainty with a tracking system estimate can be difficult to assess and utilize in COLAV systems [97]. A collision probability estimator (CPE) can be used to take into account the kinematic uncertainty from the tracking system, to provide a higher level of situational awareness for the autonomous ship. The CPE should then use a robust method with a proper trade-off between overconfidence and conservativeness in the estimate.

Collision probabilities can furthermore be good indicators for determining hazardous situations. A CPE and other probabilistic methods for evaluating collision risk can therefore also be used in general systems for supervisory risk control of autonomous ships [138], to update online risk models.

5.1.2 Previous Work

Collision Probability Estimation

Many different approaches for the problem of assessing collision probabilities and collision risk for both the maritime and air traffic sectors exist. They can be partitioned into methods which do not explicitly consider the uncertainties present in vessel navigation and control, and methods which do. Methods in the first group applied in the maritime sector, have used quantitative risk assessment models, rule-based systems with fuzzy logic, ship-ship geometry such as the time and distance to the CPA, neural networks and ship domain considerations to estimate the collision probability and risk [122].

For the second group, which is considered here, there exist different methods for collision probability estimation, as the probability can in general not be found analytically. The methods differ by the choice of framework, vessel motion and uncertainty representations, and the estimation approach used. The framework is here defined as the formulation and definition of safe separation zones, conflict zones and collisions. The word conflict is used in some of the methods, which is defined as the loss of a minimum separation distance between the vehicles, and where collision is the actual vehicle contact. For this article, a collision will instead be defined as the breach of a defined safety zone around the own-ship.

For air traffic, Paielli and Erzberger propose an approximate analytical solution to the problem in [139], [140]. A conflict probability is here calculated by integrating the combined positional PDF of two aircrafts assumed to have normally distributed trajectory errors flying with constant speed, over an extended conflict zone. The zone is essentially a corridor along the direction of the relative velocity centered to one of the aircrafts. A coordinate transformation is applied in order to make the probability feasible for direct computation. Due to the constant velocity assumption, the method performs poorly for maneuvering cases, which is improved in [91] by considering a limited rectangular conflict zone. A numerical solution for the same problem is introduced in [141], but where decoupled along-track and cross-track position uncertainties for the aircrafts are assumed. The estimation is further improved by refining the approximate conflict zone in [142] and through evaluating the approximate Cumulative Density Function (CDF) of a quadratic form of Gaussian variables for the exact conflict zone in [143]. Random sampling methods with MCS to solve the problem are used in [88], [144], [145]. The concept of probability flow was introduced and proposed for the estimation problem in [146]. Here, the rate of change of probability through the conflict zone is integrated using adaptive numerical integration methods to estimate an upper bound to the conflict probability in real-time. A thorough review of methods for collision probability estimation in the air traffic domain is given in [147], where a unified mathematical framework for the collision probability estimation problem is proposed.

For maritime vessels, a semi-analytical solution of the problem based on probability flow is presented in [148], built on the method in [146], which improves on the results summarized in the previous paragraph with less conservative estimates and lower computational time. Again, a numerical method is used to integrate the rate of change of probability, now modelled as a drift and diffusion process, to obtain the collision probability. However, the method was not tested for maneuvering ships with uncertain velocities, as only vessels with assumed constant deterministic velocities were considered in the study. Neither was the method tested in low collision probability scenarios. Furthermore, the method is not easily reproducable from the paper. An approach considering maneuvering intentions and reachable sets for the collision probability estimation is proposed in [89]. The computational efficiency of the method was however not discussed.

A combination of MCS and a KF has also been proposed recently for the estimation problem, detailed in the previous two chapters. First in [97] assuming uncertain constant velocity for both encountering vessels, and updated in [98] to consider maneuvering vessels by discretizing their trajectories and estimating the probability along piece-wise linear segments. Statistical "measurements" of the collision probability are here generated using sampled straight line trajectories from the four-dimensional obstacle uncertainty in the MCS, and processed through the KF to optimally use previous probability estimates and attenuate statistical noise due to a limited amount of samples used. This allows for a predictive collision probability calculation, but at the cost of more conservative results as only straight line trajectories are considered.

Risk-based Collision Avoidance

Maritime COLAV has been an active research field for a long time[26]. There are still many challenges associated with tackling uncertainties related to the vessel kinematics and intent of nearby obstacles present in hazardous situations. Several studies exist that take some of the uncertainties into consideration, but dealing explicitly with these challenges in a probabilistically sound manner has most often been ignored for maritime COLAV [28] as previously mentioned.

A probabilistic version of velocity obstacles for COLAV is introduced in [61], where the kinematic uncertainty in velocity for the obstacle is taken into account in a one-step reflective navigation planning approach. The computational efficiency of the method used for calculating the probabilistic velocity obstacles is however not detailed nor discussed.

A lattice-based planner is proposed in [69], where A* search is used to find a collision-free trajectory, where COLREGS breaches, trajectory deviation and high collision risk are penalized. How the collision probability between the own-ship and nearby obstacles is explicitly calculated is however not mentioned nor discussed. In [66], A* search to find a collision-free trajectory through an occupancy grid. The search uses a cost-to-go function that penalizes the Euclidean distance to the goal, and non-zero obstacle occupancy probabilities. The occupancy probabilities for obstacles are calculated using a numerical approximation, but the accuracy of the method is not discussed.

The COLAV approach in [88] uses MCS for collision probability estimation to re-plan waypoints for a collision-free trajectory when the probability estimate exceeds a threshold, considering the case when both the own-ship and obstacle have time varying kinematic uncertainty, assuming constant speed in the vessel prediction models.

The probabilistic variant of the SB-MPC [70] was introduced in [97], where a probabilistic collision cost was used in the COLAV planner to take into account kinematic obstacle uncertainty. The method was further expanded in [98] to consider obstacle intent uncertainty. The method developed for estimating collision probabilities is based on MCS considering the uncertainty in both position and velocity for obstacles. Thus, it will suffer from the curse of dimensionality and sample wasting when considering small probabilities. Both these editions were also detailed in the previous two chapters.

5.1.3 Contributions

In this chapter, a novel adaptive importance sampling method based on the Cross-Entropy method [149] is presented, for use in estimating ship-ship collision probabilities. The CPE will search for a near-optimal PDF to sample from, in order to maximize the number of effective samples used in estimation, thereby attaining better accuracy in the sense of lower variance estimates than conventional sampling methods such as pure MCS [145] and Importance Sampling (IS). The method is benchmarked against these traditional sampling-based estimation methods, in cases of low to high collision probabilities for typical maritime traffic situations involving maneuvering ships. This is something that has not been properly done in previous work, e.g. [148], as constant speed and course are often assumed for the vessels. The proposed CPE is furthermore shown able to deliver accurate estimates for real-time use.

To demonstrate a possible use case of the proposed method, it is employed in a maritime COLAV system to show the benefits of more accurate risk assessment. A simulation study is presented to validate the PSB-MPC [98] COLAV algorithm using the CE-method for collision probability estimation. The COLAV planning algorithm robustness when the obstacles perform unexpected maneuvers is tested, which requires accurate predicted collision probabilities in order to have adequate reactive COLAV capabilities. The PSB-MPC using the CE-method for collision probability estimation is compared to the original SB-MPC as in [70]. The case when all vessels involved are intelligent and use the PSB-MPC is also shown.

5.1.4 Chapter Overview

This chapter is organized as follows. Section 5.2 briefly outlines the prediction models used in the COLAV planning algorithm. The collision probability definition is given in Section 5.3, whereas Section 5.4 introduces the CE-method applied to the estimation of this probability. Section 5.6 comes with a description of the PSB-MPC algorithm and contains a simulation study of the COLAV algorithm in different situations. Finally, Section 5.7 comes with concluding remarks to the work in this article.

5.2 Models

The own-ship is represented by a ship model with 3 degrees of freedom as in [97], described in Section 2.4.4. The COLAV algorithm will use an OU process to describe the predicted motion of obstacles, as described in Section 2.4.2.

As in the previous two chapters, it is assumed that the obstacle state is Gaussian, such that $\boldsymbol{x}$ will have a predicted PDF $p^i(\boldsymbol{x}, t_k) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_k^i, \boldsymbol{P}_k^i)$ at the current prediction time t_k, with mean $\boldsymbol{x}_k^i$ and covariance $\boldsymbol{P}_k^i$.

5.3 Collision Probability

5.3.1 Assumptions

Before defining the collision probability, the assumptions behind the CPE framework are stated:

1. The own-ship trajectory has negligible uncertainty.
2. The safety zone around the own-ship is circular with radius d_{safe}.
3. The obstacle has a Gaussian distributed state obtained from the OU process, which will have a predicted PDF $p^i(\boldsymbol{x}, t_k)$.

The first assumption is made because more accurate sensing often makes the own-ship navigation uncertainty small compared to the tracked or predicted obstacle uncertainty [113].

The second and third assumptions allow for a tractable approach for estimating the collision probability. The safety zone radius d_{safe} can be inflated by for instance half the length of the encountering vessels, to take into account the geometric extent of the vessels.

The Gaussianity assumption for the obstacle uncertainty is common in vehicle tracking systems [120]. Thus, since the proposed CPE here uses estimates from such a tracking system, this assumption is natural. The predicted obstacle PDF will be initialized with the state estimate $\hat{\boldsymbol{x}}^i(t_0)$ and the corresponding error covariance matrix $\hat{\boldsymbol{P}}^i(t_0)$ at the current time t_0, obtained from the tracking system.

5.3.2 Definition

The collision probability, denoted $\mathbb{P}^i_{c,k}$, is defined as the probability for an encountering obstacle with index i to breach the own-ship safety zone within the predicted time interval $[t_k, t_{k+1}]$. It can be defined using the probability of the event

$$\begin{aligned} \mathcal{C}^i_k = & \textit{ An obstacle } i \textit{ breaches the own-ship} \\ & \textit{safety zone within the time interval } [t_k, t_{k+1}]. \end{aligned} \tag{5.1}$$

which can be expressed in integral form using the obstacle positional PDF as

$$\mathbb{P}^i_{c,k} = \Pr\{\mathcal{C}^i_k\} = \iint_{\mathcal{R}} p^i_2(x, y; t_k) dxdy \tag{5.2}$$

where $\mathcal{R}$ is the two-dimensional area of integration, namely the own-ship safety zone, and $p^i_2(\cdot)$ the predicted positional part of the obstacle PDF, hence the subscript 2. Although the integral is over the own-ship safety zone, the main contribution to the collision probability will come from the conflict zone $\mathcal{S}_2$, which is the region of overlap between the own-ship safety zone and the obstacle PDF, illustrated in Fig. 5.1. The obstacle PDF is here represented as a probability ellipse, which has the equation

$$(\boldsymbol{x} - \boldsymbol{x}^i_{1:2,k})^T (\boldsymbol{P}^i_{2,k})^{-1} (\boldsymbol{x} - \boldsymbol{x}^i_{1:2,k}) = \chi^2_{1-\alpha_p,n} \tag{5.3}$$

where the index $1:2$ for $\boldsymbol{x}^i_{1:2,k}$ signifies components 1 to 2 of the state vector. $\boldsymbol{P}^i_{2,k}$ is the positional part of the predicted obstacle covariance at time t_k, and $\chi^2_{1-\alpha_p,n}$ is the inverse cumulative Chi-squared value for a probability $1 - \alpha_p$ with $n = 2$ degrees of freedom. For instance $\alpha_p = 0.003$ will give approximately the 3σ probability ellipse, i.e. enveloping approximately 99.7% of the total positional uncertainty.

Assumption 1 from the previous section allows the simplification of only considering the obstacle uncertainty. However, the obstacle PDF could be replaced by the combined uncertainty of both the own-ship and obstacle if the own-ship uncertainty was deemed significant compared to that of the obstacle. This is done in e.g. [140]. The uncertainty in the obstacle velocity is here indirectly taken into account through the propagation of its predicted state $\boldsymbol{x}^i_k$ and covariance $\boldsymbol{P}^i_k$.

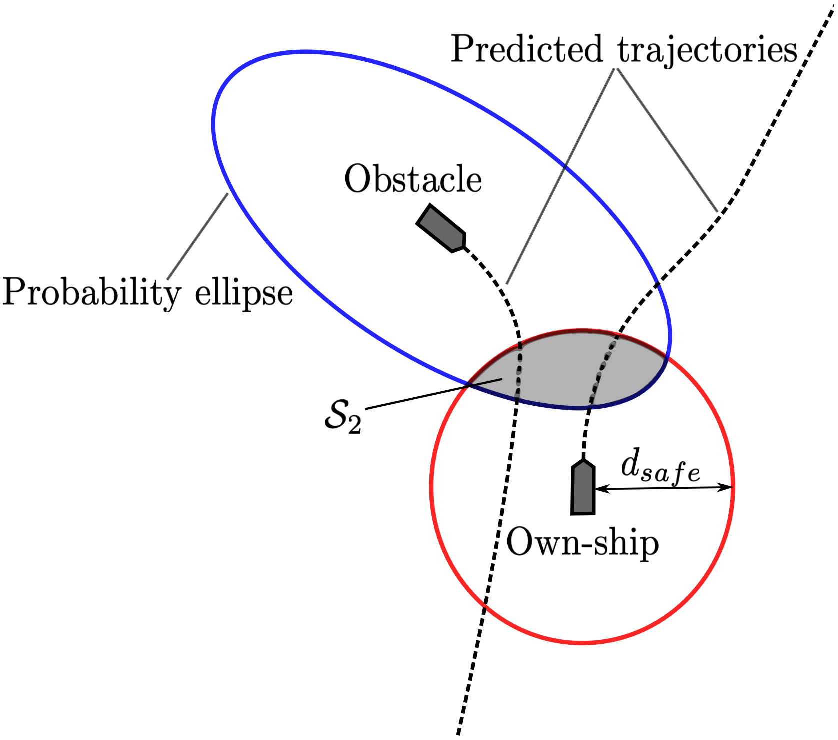


Figure 5.1: Collision probability estimation geometry illustration, with the own-ship safety zone in red, and the positional uncertainty for a dynamic obstacle represented by a 3σ probability ellipse. $\mathcal{S}_2$ is the conflict zone.

5.4 The Cross-Entropy Method Applied to Ship Collision Probability Estimation

5.4.1 Derivation

The collision probability (5.2) can be written as the integral

$$\mathbb{P}^{i}_{c,k} = \int_{\mathcal{R}} p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{x} \tag{5.4}$$

with $p(\boldsymbol{x}, \boldsymbol{\theta}) = p^{i}_{2}(x, y; t_k)$, where $\boldsymbol{\theta}$ is the parameter vector describing the obstacle positional PDF at time t_k. A conventional method for solving this using sampling based estimation is the Importance Sampling (IS) approach, described in Section

2.2.2, which estimates the integral using

$$\begin{aligned}\mathbb{P}^i_{c,k} &= \int_{\mathcal{R}} \frac{p(\boldsymbol{x},\boldsymbol{\theta})}{\lambda(\boldsymbol{x},\boldsymbol{\nu})}\lambda(\boldsymbol{x},\boldsymbol{\nu})d\boldsymbol{x} \\ &\approx \frac{1}{N}\sum_{z=1}^{N}\mathcal{I}\{\boldsymbol{x}^z \in \mathcal{R}\}\omega^z(\boldsymbol{\theta},\boldsymbol{\nu})\end{aligned} \tag{5.5}$$

where N is here the number of state samples $\boldsymbol{x}^z$, with z as the sample superscript.

The Cross-Entropy (CE) method can be used when finding a tractable importance density $\lambda(\boldsymbol{x},\boldsymbol{\nu})$ is hard [137]. A tractable density is easy to evaluate and will give a low variance estimator $\hat{\mathbb{P}}^i_{c,k}$ for (5.4). For estimation, the CE-method boils down to an adaptive importance sampling strategy which attempts to find the best possible importance density through iterative optimization. This is done by minimizing the Kullback-Leibler divergence between the optimal importance density $\lambda^*(\boldsymbol{x})$ and the current density $\lambda(\boldsymbol{x},\boldsymbol{\nu})$ in each step. The Kullback-Leibler divergence between two PDF´s Θ and Υ is given by

$$\begin{aligned}\mathcal{D}(\Theta,\Upsilon) = E_\Theta\left[\ln\frac{\Theta(\boldsymbol{x})}{\Upsilon(\boldsymbol{x})}\right] &= \int \Theta(\boldsymbol{x})\ln\frac{\Theta(\boldsymbol{x})}{\Upsilon(\boldsymbol{x})}d\boldsymbol{x} \\ &= \int \Theta(\boldsymbol{x})\ln\Theta(\boldsymbol{x})d\boldsymbol{x} - \int \Theta(\boldsymbol{x})\ln\Upsilon(\boldsymbol{x})d\boldsymbol{x}\end{aligned} \tag{5.6}$$

The CE method can then be formulated as the optimization problem

$$\min_{\boldsymbol{\nu}} \mathcal{D}(\lambda^*(\boldsymbol{x}),\lambda(\boldsymbol{x},\boldsymbol{\nu})) \tag{5.7}$$

where $\lambda^*(\boldsymbol{x})$ and $\lambda(\boldsymbol{x},\boldsymbol{\nu})$ are the optimal and current importance densities, respectively. The optimal importance density has the property of being able to estimate $\mathbb{P}^i_{c,k}$ with minimal variance, and can readily be found from (5.5) as

$$\lambda^*(\boldsymbol{x}) = \frac{\mathcal{I}\{\boldsymbol{x}\in\mathcal{R}\}p(\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{P}^i_{c,k}} \tag{5.8}$$

However, since $\mathbb{P}^i_{c,k}$ is unknown, (5.8) can not be evaluated and it can not be guaranteed that the current density $\lambda(\boldsymbol{x},\boldsymbol{\nu})$ will converge towards the optimal one. However, one can use (5.8) in reformulating the problem, noticing that only the latter integral in (5.6) will be dependent on $\boldsymbol{\nu}$, and thus (5.7) can be formulated as the maximization problem

$$\max_{\boldsymbol{\nu}} \int \lambda^*(\boldsymbol{x})\ln\lambda(\boldsymbol{x},\boldsymbol{\nu})d\boldsymbol{x} \tag{5.9}$$

Inserted for $\lambda^*(\boldsymbol{x})$, this gives

$$\max_{\boldsymbol{\nu}} \int \frac{\mathcal{I}\{\boldsymbol{x}\in\mathcal{R}\}p(\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{P}^i_{c,k}}\ln\lambda(\boldsymbol{x},\boldsymbol{\nu})d\boldsymbol{x} \tag{5.10}$$

which is equivalent to

$$\max_{\boldsymbol{\nu}} E_{\boldsymbol{\theta}} \left[\mathcal{I}\{\boldsymbol{x} \in \mathcal{R}\} \ln \lambda(\boldsymbol{x}, \boldsymbol{\nu})\right] \tag{5.11}$$

where the expectation is taken with respect to $p(\boldsymbol{x}, \boldsymbol{\theta})$. The solution $\boldsymbol{\nu}^*$ to (5.11) can be estimated through its stochastic counterpart

$$\max_{\boldsymbol{\nu}} \sum_{z=1}^{N} \mathcal{I}\{\boldsymbol{x}^z \in \mathcal{R}\}\omega^z(\boldsymbol{\theta}, \boldsymbol{\phi}) \ln \lambda(\boldsymbol{x}, \boldsymbol{\nu}) \tag{5.12}$$

where $\boldsymbol{x}^1, ..., \boldsymbol{x}^N$ are samples from $\lambda(\boldsymbol{x}, \boldsymbol{\phi})$, with $\boldsymbol{\phi}$ as the reference parameter used in the importance sampling of (5.11), typically taken as the prior importance density parameters. When (5.12) is convex and differentiable with respect to $\boldsymbol{\nu}$, the estimate $\hat{\boldsymbol{\nu}}^*$ for the new parameters can be found by solving the resulting system of optimality equations [137]

$$\frac{1}{N} \sum_{z=1}^{N} \mathcal{I}\{\boldsymbol{x}^z \in \mathcal{R}\}\omega^z(\boldsymbol{\theta}, \boldsymbol{\phi}) \nabla_{\boldsymbol{\nu}} \ln \lambda(\boldsymbol{x}^z, \boldsymbol{\nu}) = \mathbf{0} \tag{5.13}$$

The importance parameters can then be iteratively improved by some performance criteria by generating samples from the current importance density and solving (5.13). For random variables in the exponential family, analytical solutions to (5.13) can be found. As Gaussian distributions are considered, the mean $\boldsymbol{\mu}^{CE}$ and covariance $\boldsymbol{P}^{CE}$ are the relevant parameters. Then, iteratively solving (5.13) boils down to an iterative maximum likelihood estimation update

$$\boldsymbol{\mu}_j^{CE} = \frac{1}{N_{elite}} \sum_{e=1}^{N_{elite}} \boldsymbol{x}^e \tag{5.14}$$

$$\boldsymbol{P}_j^{CE} = \frac{1}{N_{elite}} \sum_{e=1}^{N_{elite}} (\boldsymbol{x}^e - \boldsymbol{\mu}_j^{CE})(\boldsymbol{x}^e - \boldsymbol{\mu}_j^{CE})^T \tag{5.15}$$

with N_{elite} being the number of best performing or elite samples $\boldsymbol{x}^e$, with superscript e, drawn from $\lambda(\boldsymbol{x}, \boldsymbol{\nu}_{j-1})$. The index j denotes the current importance density parameters. The superscript CE is used to separate the CE-method importance parameters from other variables with the same letter. See [149] and [137] for more information about the method derivation in general.

Note that although the Gaussian distribution is assumed here, the CE-method is general, and solving (5.12) using other sampling distributions can be done. This can however lead to non-convex programs, and might require a solver, which can lead to more computational effort needed in using the method.

5.4.2 Importance Density Update Criteria

The CE-method will iteratively improve the importance density by sampling from the current density, determine valid samples for update, and use these in (5.14) -

(5.15). The criterion for $\boldsymbol{x}^e$ being an elite sample is here formulated as the condition

$$\begin{aligned} \mathcal{V} = (\boldsymbol{x}^e - \boldsymbol{\mu}_{2,k}^{OS})^T(\boldsymbol{x}^e - \boldsymbol{\mu}_{2,k}^{OS}) \leq d_{safe}^2 \quad \& \\ (\boldsymbol{x}^e - \boldsymbol{\mu}_{2,k}^{i})^T(\boldsymbol{P}_{2,k}^{i})^{-1}(\boldsymbol{x}^e - \boldsymbol{\mu}_{2,k}^{i}) \leq \chi_{1-\alpha_p,n}^2 \end{aligned} \tag{5.16}$$

where $\boldsymbol{\mu}_2^{OS}$ and $\boldsymbol{\mu}_2^i$ are the predicted own-ship and obstacle positions, respectively. Thus $\mathcal{V} = \mathcal{I}\{\boldsymbol{x} \in \mathcal{S}_2\}$, i.e. the sample must be both inside the own-ship safety zone and the obstacle probability ellipse bounded by the inverse cumulative Chi-squared value $\chi_{1-\alpha_p,n}^2$ with $n = 2$. The parameter α_p should be chosen low enough to ensure that all samples giving significant weight from the entire conflict zone are used in updating the importance density, to ensure that $\lambda(\boldsymbol{x}, \boldsymbol{\nu})$ has proper support when estimating (5.2) after the final density update.

The reason for using the condition (5.16) stems from the desire to maximize the amount of samples used in estimation. Elite samples after this criterion will have non-zero importance weights, which means samples inside the safety zone, with integrand PDF values larger than zero. In cases of low collision probabilities, where the conflict zone is a region in the tail of the integrand PDF in (5.2), MCS and poorly chosen IS densities will typically give less accurate estimates with higher variance due to sample wasting. The CE-method can perform better by searching for the optimal area to sample from, which will be a density centered in and covering the conflict zone.

5.4.3 Smoothing

The updated density can be smoothed [149] using

$$\boldsymbol{\mu}_j^{CE} = \alpha \boldsymbol{\mu}_j^{CE} + (1-\alpha)\boldsymbol{\mu}_{j-1}^{CE} \tag{5.17}$$

$$\boldsymbol{P}_j^{CE} = \alpha \boldsymbol{P}_j^{CE} + (1-\alpha)\boldsymbol{P}_{j-1}^{CE} \tag{5.18}$$

with parameter $0 < \alpha \leq 1$. This is done in order to prevent degeneration in the importance density, which can occur when having a low number N_e of elite samples or an extremely small conflict zone.

5.4.4 Initialization

A heuristic initialization of the starting importance density parameters $\boldsymbol{\nu}_0$ as

$$\boldsymbol{\mu}_0^{CE} = \frac{1}{2}(\boldsymbol{\mu}_{2,k}^{OS} + \boldsymbol{\mu}_{2,k}^{i}) \tag{5.19}$$

$$\boldsymbol{P}_0^{CE} = \frac{d_{safe}^2}{3}\boldsymbol{I}_{n \times n} \tag{5.20}$$

is used, based on trial and error, where $\boldsymbol{\mu}_2^{OS}$ and $\boldsymbol{\mu}_2^i$ are the predicted own-ship and obstacle positions, respectively.

A hot start using the previous optimal parameters is applied if the algorithm satisfied the termination criteria with collecting enough samples in the previous iteration

$$\boldsymbol{\mu}_0^{CE} = \boldsymbol{\mu}_{opt,last}^{CE} + (\boldsymbol{v}_{k-1}^{OS} + \boldsymbol{v}_{k-1}^{i})T_s \tag{5.21}$$

$$\boldsymbol{P}_0^{CE} = \boldsymbol{P}_{opt,last}^{CE} + \sigma_{inj}^2 \boldsymbol{I}_{n\times n} \tag{5.22}$$

where $\boldsymbol{\mu}_{opt,last}^{CE}$ and $\boldsymbol{P}_{opt,last}^{CE}$ were the previous optimal parameters, T_s is the prediction time step and $\boldsymbol{v}_{k-1}^{OS}$ and $\boldsymbol{v}_{k-1}^{i}$ are the predicted north-east velocity vector of the own-ship and the obstacle, respectively, at the previous time t_{k-1}. As the output importance sampling PDF can get very small when the conflict zone is small, a change in the vessel positions between two time instants can cause the density to not have sufficient support considering the new conflict zone. Using the previously predicted vessel velocities in the hot start will help in mitigating this problem, in addition to injecting the covariance matrix with σ_{inj}. The variance injection also makes the algorithm robust against the particle deprivation problem [150].

5.4.5 Termination Criterion

The main termination criterion for the CE-method is that $N_{elite} \geq \rho N$ elite samples have been collected, for a parameter $\rho > 0$. The secondary termination criterion is if the maximum allowed number of iterations $N_{max,iter}$ has been reached. After terminating, a final sample set of size N is generated from the current density, and the collision probability is estimated using (5.5) on the problem (5.2). The main criterion attempts to make sure that a high percentage of the samples are used in the calulation of the estimate. The parameter ρ should be chosen sufficiently high in order to obtain accurate estimates, but not too high as this can cause convergence issues. The adaptive importance sampling method for collision probability estimation based on the CE-method can be summarized in Algorithm 1.

5.5 Evaluation of the CE-method

5.5.1 Comparison of Sampling Schemes

The CE-method is compared with the method in [98], hereby abbreviated as M1. M1 considers predicted trajectories of the own-ship and obstacle, and discretizes them using piecewise linear segments. For each of these segments, straight line trajectories from the four-dimensional obstacle PDF at the relevant time are sampled. The ratio of the trajectories causing an obstacle position inside the own-ship safety zone at CPA to the total number of trajectories, is used as a collision probability estimate. A Kalman filter is then used to reduce the statistical noise in this estimate, which comes from using a finite number of samples. The collision probability is in this method defined as a collision at some time $t_c \geq t_k$ in the future. Thus, to allow for comparison, a linear segment discretization time step of $T_{seg} = t_{k+1} - t_k$ is used, i.e. equal to the prediction step T_s. This limits the possible future collisions to be inside the time interval $[t_k, t_{k+1}]$. See [98] for more details.

Algorithm 1 The CE-method based CPE algorithm.

```
$\boldsymbol{\nu}_0 \leftarrow \{\boldsymbol{\mu}_0^{CE}, \boldsymbol{P}_0^{CE}\}$ calculated using (5.19) - (5.20) or (5.21) - (5.22)
for $j = 1$ to $N_{max,it}$ do
    $N_{elite} \leftarrow 0$
    for $z = 1$ to $N$ do
        Sample $\boldsymbol{x}^z$ from $\lambda(\boldsymbol{x}, \boldsymbol{\nu}_{j-1}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{j-1}^{CE}, \boldsymbol{P}_{j-1}^{CE})$.
        if $\boldsymbol{x}^z$ satisfies (5.16) then
            Tag $\boldsymbol{x}^z$ as an elite sample
            $N_{elite} \leftarrow N_{elite} + 1$
        end if
    end for
    if $N_{elite} > \rho N$ then
        Break;
    else
        Update and smooth $\boldsymbol{\nu}_j$ using (5.14) - (5.15) and (5.17) - (5.18) with the $N_{elite}$ elite samples
    end if
    $j \leftarrow j + 1$
end for
Sample $\boldsymbol{x}^1, \boldsymbol{x}^2 ..., \boldsymbol{x}^N$ from $\lambda(\boldsymbol{x}, \boldsymbol{\nu}_j)$ and use in estimating (5.2) with (5.5).
```

The proposed method is also benchmarked against the following other traditional sampling based strategies M2 and M3.

M2: MCS

Estimating (5.2) using MCS with samples directly from the obstacle positional PDF, i.e. using $\lambda(\boldsymbol{x}, \boldsymbol{\nu}) = p_2^i(x, y; t_k)$ in the approximation (5.5). This is essentially the same approach as in [145].

M3: Naive IS

Estimate $\mathbb{P}_{c,k}^i$ from (5.5) by sampling from a Gaussian PDF centered at the own-ship position at time t_k with variance $\frac{d_{safe}^2}{3}$ in both x and y. The variance value was chosen in order to give a good spread of samples on the entire safety zone, with a minimum amount outside the zone.

5.5.2 Simulation Results

The different methods were compared in three prediction scenarios with one obstacle, as described below. Vessels of lengths around 40 m were considered. An approximate ground truth (AGT) was obtained by using an MCS strategy with 10^6 samples for estimating the probability, to provide a benchmark for the other methods in terms of accuracy. The root mean square error (RMSE) of the collision probability estimates for the other methods *relative* to the AGT were used as a performance metric. Results using these metrics were averaged over 50 MCS for each scenario and for different sample numbers N. The simulations were performed in MATLAB R2019a running on a laptop with an Intel(R) Core(TM) i9-8950HK 2.90 GHz processor and 32GB RAM.

1. Head-on scenario with the own-ship starting at $(x, y) = (0, 0)$ heading north with forward speed $u = 9\,\text{m/s}$, while the obstacle is starting at $(x^i, y^i) = (150, 50)$ with initial covariance matrix $\boldsymbol{P}^i = \text{diag}\left([10^2, 5^2, 0.5^2, 0.5^2]\right)$ following a nearly constant speed of 9 m/s, but immediately starting a port maneuver to keep clear of the own-ship. There is no real threat of collision in this scenario, but the probability of collision will be non-zero due to the obstacle uncertainty causing a non-empty conflict zone.
2. Crossing scenario with the own-ship again starting at $(x, y) = (0, 0)$ heading north with forward speed $u = 9\,\text{m/s}$. The obstacle starts at $(x^i, y^i) = (400, -400)$ with initial covariance $\boldsymbol{P}^i = \text{diag}\left([5^2, 10^2, 0.5^2, 0.5^2]\right)$ travelling eastward with constant speed 9 m/s. No avoidance maneuvers are taken, so a direct collision will occur at $(400, 0)$ in the nominal scenario.
3. Overtaking scenario with the own-ship starting at $(x, y) = (100, 0)$ heading north with forward speed $u = 15\,\text{m/s}$, while the obstacle is starting at $(x^i, y^i) = (200, 0)$ with covariance $\boldsymbol{P}^i = \text{diag}\left([10^2, 5^2, 0.5^2, 0.5^2]\right)$ also going north with constant speed 3 m/s. The own-ship takes a starboard maneuver immediately to avoid the obstacle, before converging back to the trajectory going straight north.

Scenarios with different proximity between the own-ship and obstacle have been chosen to illustrate the performance of the methods in dealing with different collision situations which can be encountered in the predictions of a COLAV planner. The starting covariance values are partially based on the results in [129].

Important parameters are given in Table 5.1. For the OU process, parameters are chosen partially based on values estimated in [79], but with higher values due to a time scale of seconds and not hours being used. They were also chosen to assume the obstacles are smaller vessels with high maneuverability by having larger revertion strength. Increased maneuverability will increase the predicted kinematic uncertainty of a vessel, and the process noise was therefore also increased to reflect this. Equal revertion in both x and y leads to an equal increase of obstacle uncertainty in these directions, which will coincide well with a higher tendency for maneuvering. The CE parameters are chosen based on trial and error.

For the sample collision probabilities shown in the results below, the number of samples N used for each method M1 - M3 and CE have been chosen individually to account for the difference in computational effort required, to make their computation time approximately equal on average.

Note that the vessel sizes indicated in the figures below have been scaled for visualization, and does not necessarily reflect real vessel dimensions. In the north east

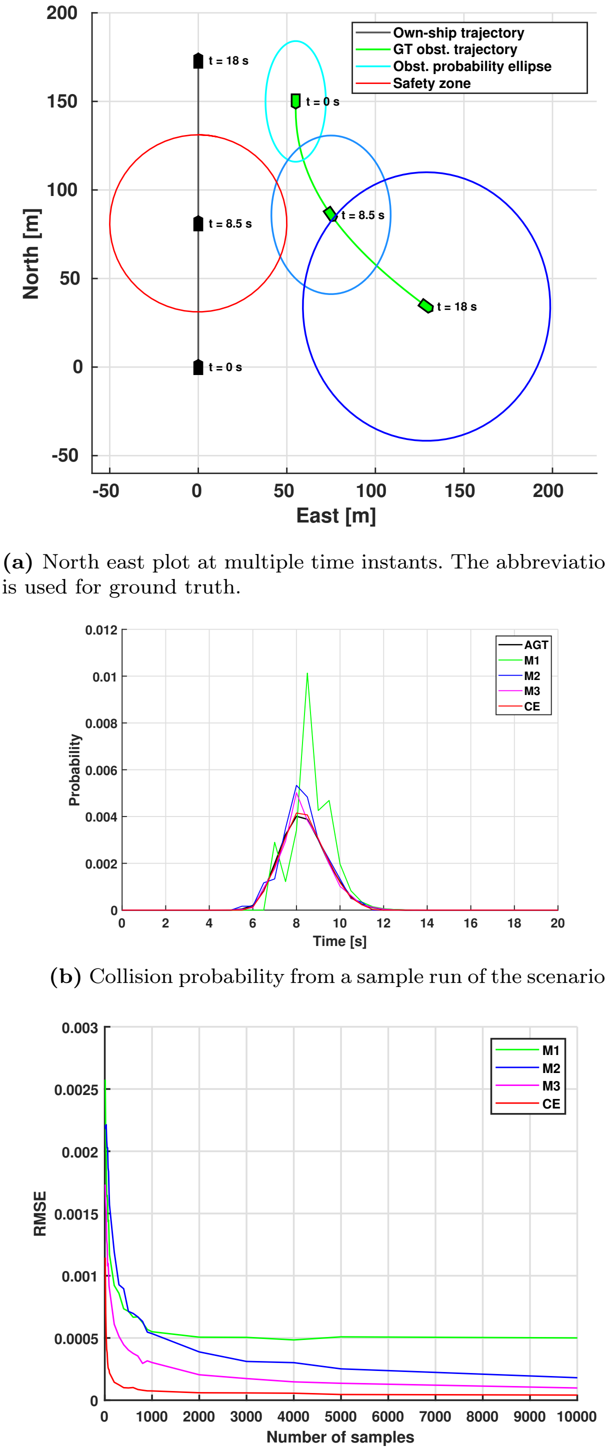
Table 5.1: Important CE-parameters.

Parameter	**Value**	Comment
T_s	$0.5\,\mathrm{s}$	Prediction time step
d_{safe}	$50\,\mathrm{m}$	Safety zone radius
σ_x	$0.9\,\mathrm{m/s}^2$	Process noise parameter
σ_{xy}	$0\,\mathrm{m/s}^2$	-
σ_y	$0.9\,\mathrm{m/s}^2$	-
γ	$[0.1, 0.1]^T\,1/\mathrm{s}$	Revertion parameter
ρ	0.9	Sample fraction for parameter update
α	0.9	Smoothing parameter
α_p	0.001	Elite sample boundary parameter
σ_{inj}	$d_{safe}/2$	Injection parameter
$N_{max,iter}$	10	Max number of iterations
N_{CE}	1000	# of samples used for CE
N_{M1}	200	# of samples used for M1
N_{M2}	4000	# of samples used for M2
N_{M3}	4000	# of samples used for M3

plots, the obstacle is shown in green and the own-ship in black. The 3σ obstacle positional probability ellipse is shown at time steps with varying color, in addition to the own-ship safety zone in red.

Results for the head-on scenario are shown in Figs. 5.2(a), 5.2(b) and 5.2(c), where the collision probability is small due to a small conflict zone. Fig. 5.2(b) shows that the CE method is able to estimate the probability well compared to the AGT. The other methods have higher variance, especially M1 due to a limited sample number in the run, and because it samples directly from the four-dimensional obstacle PDF. This will lead to many wasted samples not placing the obstacle in the conflict zone. The same can be said for M2 and M3, although with less variance due to more samples and doing it from positional PDFs. Fig. 5.2(c) shows that the CE method has notably lower RMSE compared to the other methods, which will naturally even out as the number of samples increase. A higher RMSE for M1 is partially caused by the fact that only straight line trajectories are sampled in the algorithm [98], in addition to the curse of dimensionality.

Run-time results for the head-on scenario are shown in Table 5.2. An approximate linear relationship was found between the number of samples and the mean and max evaluation time, when performing the MCS. The computational effort required in the CE method is not excessive, with approximately 1 ms per run for $N = 1000$, making it feasible for real-time use. The methods M2 and M3 are naturally faster due to only one batch of samples drawn from a two-dimensional distribution being used in the estimation. For M1 there is a larger increase due to sampling trajectories from a 4-dimensional distribution [98]. Similar results can be obtained for the other scenarios. Note that the run-time results are highly dependent on hardware, implementation details and the programming language used.



(a) North east plot at multiple time instants. The abbreviation GT is used for ground truth.

(b) Collision probability from a sample run of the scenario.

(c) Root mean square error relative to the AGT averaged over 50 MCS.

Figure 5.2: Head-on scenario.

Table 5.2: Mean evaluation time per sample for the head-on scenario, averaged over 50 MCS, in addition to the max evaluation time per sample and the standard deviation on the results.

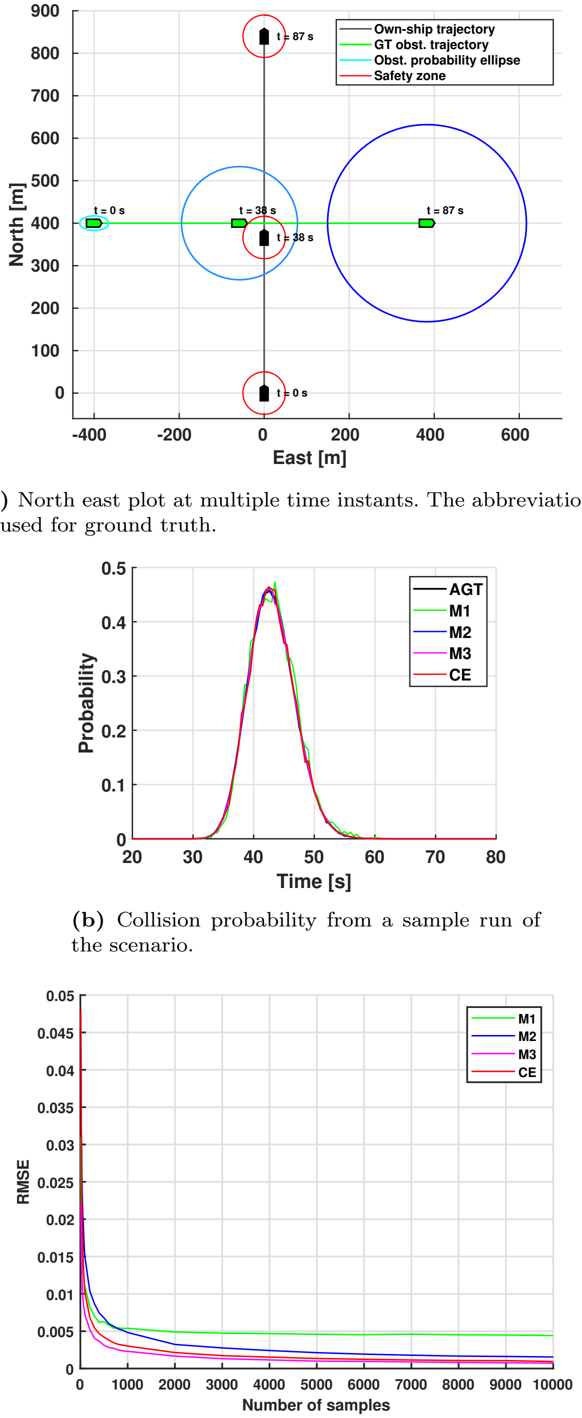
Method	Mean	Max	Std. dev.	Unit
CE	8.84×10^{-7}	3.08×10^{-6}	7.46×10^{-7}	s/sample
M1	3.99×10^{-6}	7.52×10^{-6}	3.76×10^{-6}	
M2	2.46×10^{-7}	5.96×10^{-7}	2.57×10^{-7}	
M3	2.52×10^{-7}	5.51×10^{-7}	2.71×10^{-7}	

Results for the crossing scenario are shown in Figs. 5.3(a), 5.3(b) and 5.3(c). The performance of M1-M3 is better than for the first scenario due to a larger conflict zone. Looking at the RMSE, the CE method performs on par with M3. For this scenario, the method M3 will waste few samples as the obstacle uncertainty quickly overlaps the entire safety zone over a longer time period, making the conflict zone equal to the safety zone. Thus, for high collision probability cases where the conflict zone overlaps almost fully with the own-ship safety zone, the CE-method will form an importance density centered and covering the safety zone.

This is, however, most often not the case in realistic scenarios, as ships will perform maneuvers in order to minimize collision risk. Moreover, the simulation considers a conservative safety zone of 50 m, which together with no avoidance maneuvers lead to high collision probability. Choosing another safety zone parameterization such as a polygon shape better fitted to the own-ship would decrease the probability. In terms of run-times, the computational demand of the adaptive importance sampler can be reduced by pre-checking whether or not there is any significant conflict zone, in which case one can approximate $\mathbb{P}^{i}_{c,k} \approx 0$. The case of full overlap with the safety zone can also be detected, such that the estimator can terminate its optimization earlier in order to reduce run-time.

Note also that because the estimator considers the probability of collision defined within the interval $[t_k, t_{k+1}]$, the probability only reach 1 when the obstacle uncertainty is completely overlapped by the own-ship safety zone within one time step. However, if the quantity was defined as the probability of a collision at anytime over the entire own-ship trajectory, the predicted collision probability should come close to or equal to 1 in this scenario.

Figs. 5.4(a), 5.4(b) and 5.4(c) show results for the overtaking scenario. The CE-method achieves a low variance estimate for the low probabilities in the beginning and the end, as can be seen from Fig. 5.4(b), and reflected through Fig. 5.4(c).

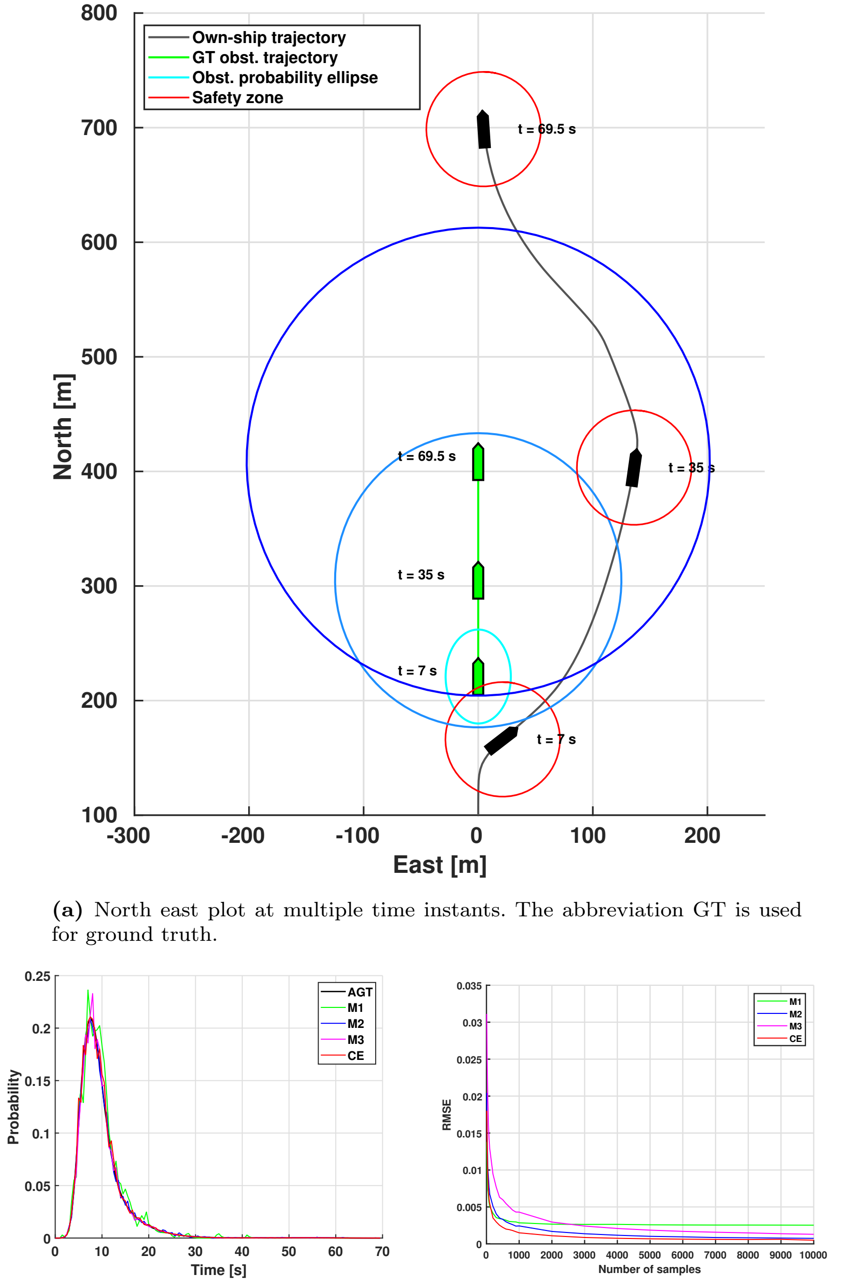


(a) North east plot at multiple time instants. The abbreviation GT is used for ground truth.

(b) Collision probability from a sample run of the scenario.

(c) Root mean square error relative to the AGT averaged over 50 MCS.

Figure 5.3: Crossing scenario.



(a) North east plot at multiple time instants. The abbreviation GT is used for ground truth.

(b) Collision probability from a sample run of the scenario with $N = 1000$.

(c) Root mean square error relative to the AGT averaged over 50 MCS.

Figure 5.4: Overtaking scenario.

5.6 The Probabilistic Scenario-based Model Predictive Control

The third edition of the PSB-MPC utilizes the CE-based CPE for collision risk assessment. Here, a more complete cost function, including previously developed extensions to the SB-MPC [71], [72], is described. The extensions include a transitional cost meant to reduce chattering decision making behavior in the MPC [71], and to take into account that track loss can occur when using the method with e.g. a radar-based tracking system [72]. The prediction scheme outlined in Section 4.4 in the previous chapter is again used here.

Again, the control behaviours are parametrized by a sequence $\left[(U^l_{m,1}, \chi^l_{m,1}), ..., (U^l_{m,n_M}, \chi^l_{m,n_M})\right]$ consisting of speed multiplicative factors U_m and additive course angle offsets χ_m to the autopilot references U_d and χ_d in forward speed and course angle, which represent n_M sequential avoidance maneuvers taken by the own-ship. The optimal control behaviour

$$l^*(t_0) = \arg\min_{l} \mathcal{H}^l(t_0) \tag{5.23}$$

at the current time t_0 is then followed, with the first maneuver $(U^{l^*}_{m,1}, \chi^{l^*}_{m,1})$ as the MPC output. The extended cost function $\mathcal{H}^l(\cdot)$ is now given as

$$\mathcal{H}^l(t_0) = \max_i \sum_{a=1}^{n_a} \mathbb{P}^i_a(t_0) C^{l,i}_a + g(\cdot) + \frac{1}{n_M} \sum_{M=1}^{n_M} f(\cdot) + \frac{1}{n_M - 1} \sum_{M=2}^{n_M} h(\cdot) \tag{5.24}$$

with

$$C^{l,i}_a = \sum_{s=1}^{n^i_{ps}(a)} \frac{w^{i,s}}{n^i_{ps}(a)} \max_{t \in D(t_0)} \left[\zeta_i \mathcal{C}^{l,s}_i(t) \hat{\mathbb{P}}^{l,i,s}_c(t) + \kappa_i \mu^{l,s}_i(t) + \tau^{l,s}_i(t)\right] \tag{5.25}$$

as the average cost for all predicion scenarios n^i_{ps} involving intention a for obstacle i. Again, the terms inside the brackets in (5.25) are the probabilistic collision cost $\mathcal{C}^{l,s}_i(t)\hat{\mathbb{P}}^{l,i,s}_c(t)$ (now weighted by the track loss factor ζ_i for the obstacle [72]), the COLREGS penalization cost $\kappa_i \mu^{l,s}_i(t)$ and transitional cost $\tau^{l,s}_i(t)$ [71], respectively. The estimated collision probability $\hat{\mathbb{P}}^{l,i,s}_c$ has an expanded notation compared to the probability defined in (5.2), which involves the own-ship trajectory when following control behavior l and the obstacle prediction scenario s. Expressions for the control deviation cost $f(\cdot)$, chattering cost $h(\cdot)$, the weights $w^{i,s}$ and other cost function terms are found in the previous two chapters, and Section 2.6 on the SB-MPC.

5.6.1 Simulation Results

The simulation results are split in two. In the first part, the PSB-MPC is compared to the original SB-MPC from Section 2.6 in the three situations described below. The own-ship starts in the origin $(x, y) = (0, 0)$ in all scenarios, heading north with constant forward speed $U = 9\,\text{m/s}$ in situation 1 and 2, and $U = 6\,\text{m/s}$ in situation 3. The obstacle configuration in all situations are given below.

1. Head-on scenario with an obstacle travelling straight south towards the own-ship with a speed of 9 m/s, and makes a hazardous port maneuver at a certain time instant, breaking COLREGS.
2. Crossing scenario where an obstacle travels westwards with a speed of 10 m/s, and breaks its stand-on duty [21] in the COLREGS situation with the own-ship by making a port maneuver towards the south.
3. Combined scenario where a fast obstacle $i = 1$ overtakes the ownship with a speed of 12 m/s in a COLREGS-compliant maneuver. Obstacle $i = 2$ travels straight eastward with a speed of 6 m/s, with no regard for adhering to COLREGS, simulating the case for an unaware vessel or with steering problems.

The scenarios in part one put emphasis on testing the COLAV planning method´s robustness where obstacles make unexpected maneuvers which require good reactive capabilities with accurate risk assessment.

Part two of the simulations uses the same set of situations and initial conditions with regular head-on, crossing and combined crossing and overtaking, but where all vessels involved are assumed to be intelligent and uses the PSB-MPC, to illustrate that multiple vessels running the same COLAV system with the CE-method as a CPE can work together to avoid collision and adhere to the COLREGS.

For both parts, a number of 50 MC simulations were used for each situation. The percentage of own-ship safety zone violations was calculated for each situation, for both MPC versions in part one and only the PSB-MPC in part two. The percentage is calculated considering if there is a safety zone violation with respect to any obstacle at minimum 1 point in time during a simulation. A safety zone violation is defined as a collision here, as per (5.1). Note that the own-ship safety zone $d_{safe} = 50\,\text{m}$ is inflated to be larger than the geometric vessel extent, and thus the metric should be taken as a crude performance measurement.

The MPC versions are tuned similar to that in [98], and the CE-based CPE uses similar parameters to the ones in Table 5.1. The COLAV system is assumed to use a KF-based tracking system implemented as in [98] for obtaining dynamic obstacle state estimates $\hat{\boldsymbol{x}}^i(t_0)$ and error covariances $\hat{\boldsymbol{P}}^i(t_0)$ at the current time t_0.

Furthermore, the PSB-MPC considers intentions $a \in \{1, 2, 3\}$ as described in the previous chapter, and assumes constant intention probabilities of $\mathbb{P}^i_a = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$ for the obstacles. This can represent the case when no information on obstacle intents is known.

Part One

The figures below illustrate a particular realization from the Monte Carlo simulation for each situation. The own-ship running the PSB-MPC and the original SB-MPC are shown in red and blue, with black and grey dashed trajectories, respectively. The obstacles are shown in green, with their ground truth and tracked trajectories in green and purple, respectively. The associated measurements are shown as black circles. Further, the 3σ obstacle positional probability ellipse using the tracking system information is shown at the different time steps in teal. Figures displaying the distance from the own-ship to the obstacles are also shown.

Results from the head-on scenario are shown in Figs. 5.5(a) and 5.5(b). Both versions of the MPC algorithm adhere to the COLREGS, but only the PSB-MPC avoids a breach of its safety zone, due to taking into account both kinematic and intention uncertainty for the obstacle. Again, the tracking system has a higher state error covariance matrix due to being tuned with higher process noise to take into account fast obstacle maneuvers. This becomes an issue for the SB-MPC, where its decision made just before t_2 does not take into account the kinematic uncertainty, nor does it anticipate an obstacle maneuver, which leads to the obstacle violating the own-ship safety zone. In the next SB-MPC iteration, this then leads to a panic maneuver directed away from the obstacle after t_2, because the maximum collision cost in the algorithm is attained already at the start of the prediction.

In the crossing situation in Figs. 5.6(a) and 5.6(b) the own-ship has give-way duty, and both MPC versions make the COLREGS-compliant starboard maneuvers. However, due to another unexpected obstacle maneuver, the SB-MPC struggles with staying well clear of the obstacle. The PSB-MPC utilizes the accurate CPE and probabilistic risk assessment in the algorithm to make a safer maneuver in order to avoid collision.

Results from the combined crossing and overtaking situation are shown in Figs. 5.7(a), 5.7(b) and 5.7(c), where the own-ship has stand-on duty with respect to both obstacles. This situation is challenging due to having an overtaking obstacle $i = 1$ on the own-ship´s starboard side with no regard for keeping a minimum safe distance. On the other hand, a port maneuver to avoid the overtaking obstacle violating the own-ship safety zone would possibly cause a collision with the obstacle $i = 2$ to the northwest. Both MPC versions tackle the situation fairly well, taking small avoidance maneuvers to avoid safety zone violations.

Part Two

For the intelligent simulation, the same set of figures are given in Figs. 5.8(a) and 5.8(b) for the head-on situation, in Figs. 5.9(a) and 5.9(b) for the crossing situation, and Figs. 5.10(a), 5.10(b) and 5.10(c) for the multi-ship situation. Statistics on safety zone violations for both parts are shown in Table 5.3. For the first part, the results indicate that the PSB-MPC is better by a large margin in avoiding collisions in the first two situations. The SB-MPC performs best in the last situation because no sudden obstacle maneuvers are taken as in the two other situations. When all vessels utilize the PSB-MPC with the CE-method for CPE in the COLAV system as in part two, the situations are resolved in a safe and COLREGS-compliant manner.

Table 5.3: Percentage of safety zone violations for the own-ship using both MPC-versions in part one of the simulations, and for the PSB-MPC in part two, calculated over the 50 MCS.

Part	Method	Situation 1	Situation 2	Situation 3
One	SB-MPC	88%	95%	30%
	PSB-MPC	0%	0%	4%
Two	PSB-MPC	0%	0%	0%

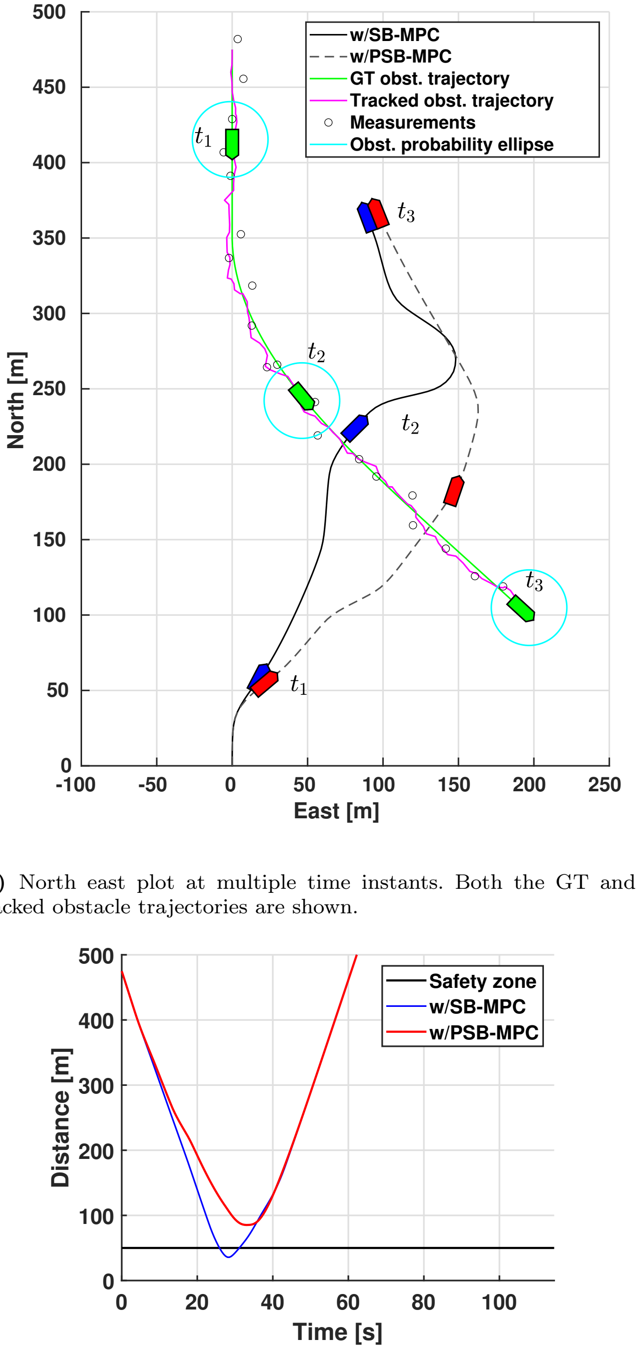
5.7 Conclusion

Because the predicted kinematic uncertainties of vessels can get large when considering time horizons over the minute, predicted collision probabilities will often be small. The new CE-based CPE presented here and used in the PSB-MPC for calculating probabilistic collision costs is more accurate than other sampling based methods in the sense of lower estimator variance, when used in estimating small collision probabilities. For high collision probabilities, it performs on par with other sampling based methods. Furthermore, the method is capable of delivering the estimates in real-time. This results in a more robust COLAV planning algorithm by making it less susceptile to statistical noise in sampling-based collision probability estimation. Alternatively, the computational efficiency improvements can be used to evaluate more intention scenarios for the obstacles, also contributing to improved robustness.

The resulting closed-loop COLAV system is shown to be able to perform adequate reactive collision avoidance in simulation, when nearby obstacles make unexpected maneuvers. It is also demonstrated that safety is attained in a manner compliant with rules 13 - 17 of COLREGS, when all vessels involved in a situation use the same closed-loop COLAV system.

The proposed CPE is beneficial to use in general risk management systems for ships to determine the current or predicted collision risk, where accuracy is important, which can further be used to determine or trigger feasible avoidance actions. The low variance property causes the decision making relying on collision risk estimates to be more robust. However, for crude precaution measures or detecting hazardous situations, simpler methods based on CPA parameters can be more reasonable, as the added computational effort from the optimization in the CE-method might then not be necessary.

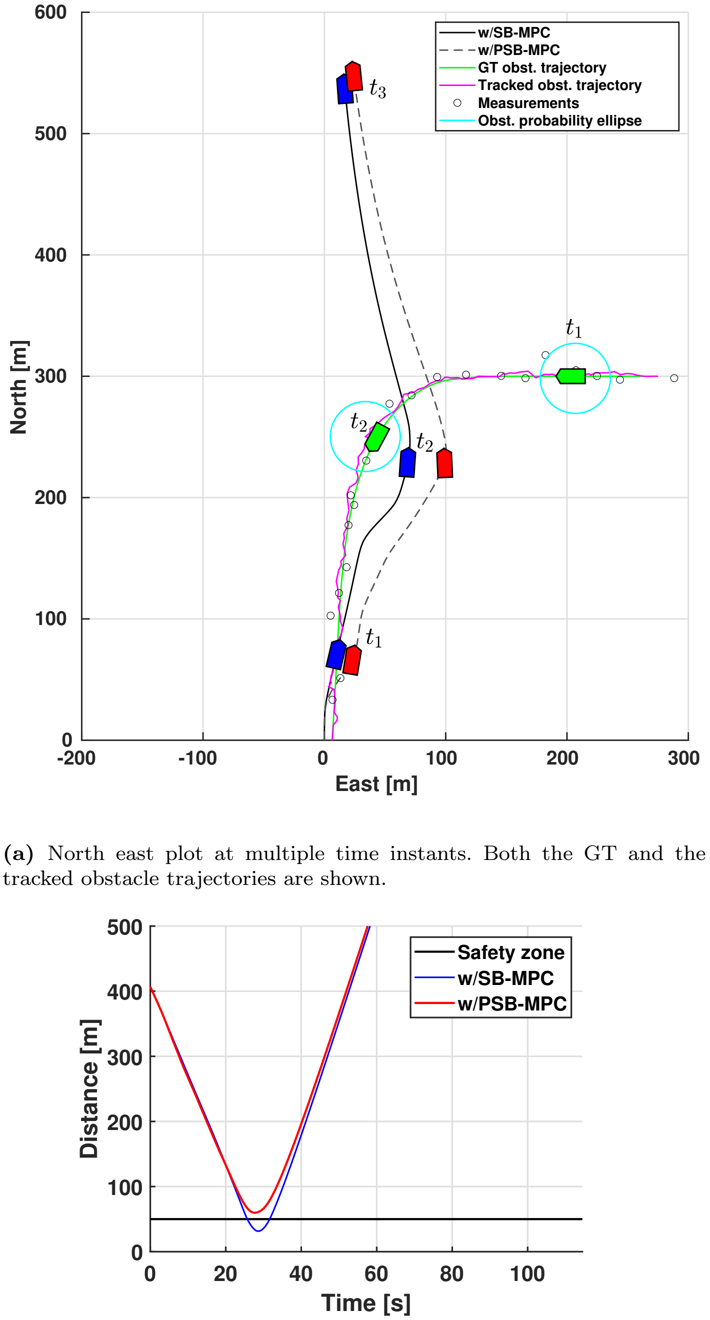
Since only kinematic uncertainty is considered in the CPE, the collision probabilities estimated considering predicted trajectories will be high compared to the case in a real-time situation, due to other factors not being considered. Thus, future work will involve improving the CPE by taking into account information such as the obstacle ship type and intention uncertainty in addition to its kinematic uncertainty, to make the collision probability estimates less conservative.



(a) North east plot at multiple time instants. Both the GT and the tracked obstacle trajectories are shown.

(b) Distance to the obstacle.

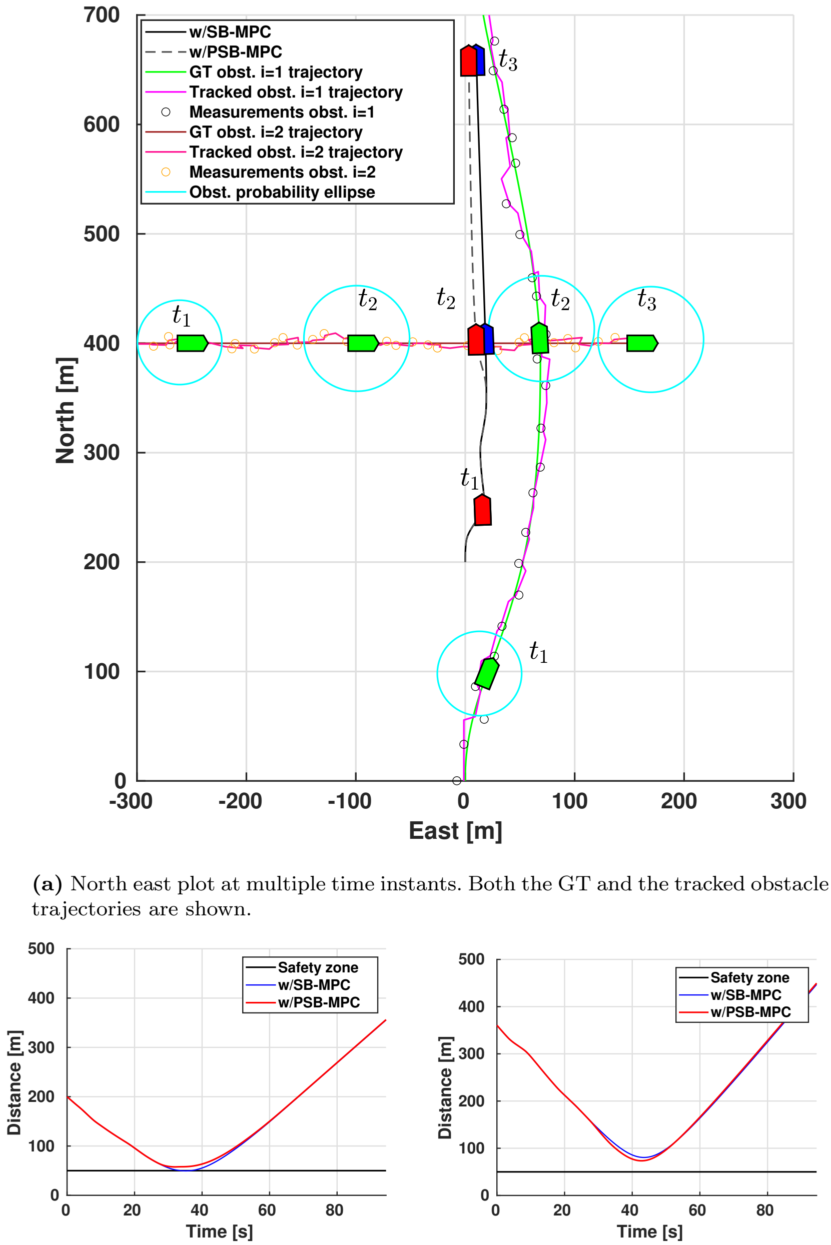
Figure 5.5: Head-on situation part one.



(a) North east plot at multiple time instants. Both the GT and the tracked obstacle trajectories are shown.

(b) Distance to the obstacle.

Figure 5.6: Crossing situation part one.

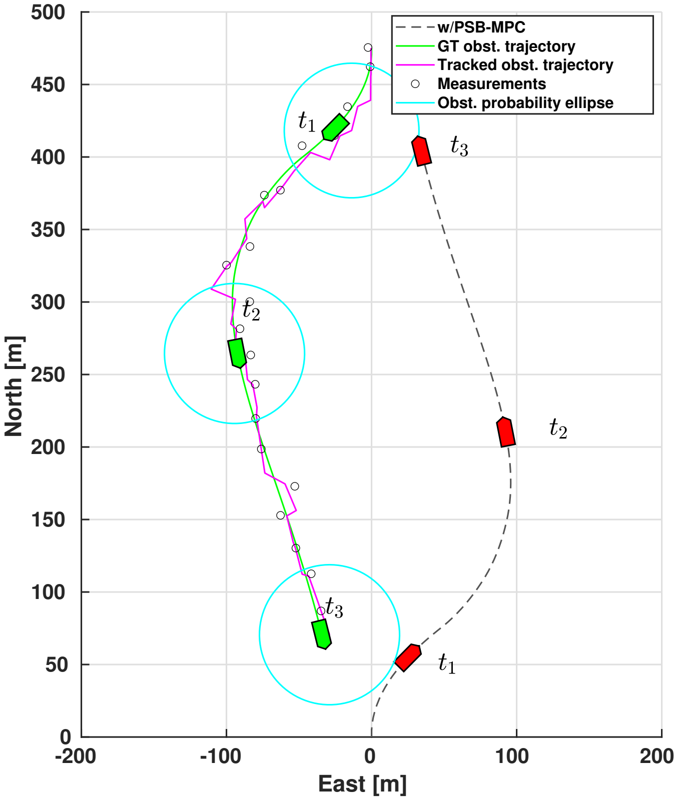


(a) North east plot at multiple time instants. Both the GT and the tracked obstacle trajectories are shown.

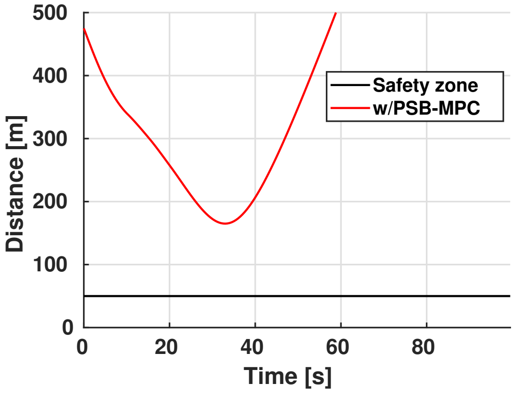
(b) Distance to the obstacle $i = 1$.

(c) Distance to the obstacle $i = 2$.

Figure 5.7: Combined crossing and overtaking situation part one.

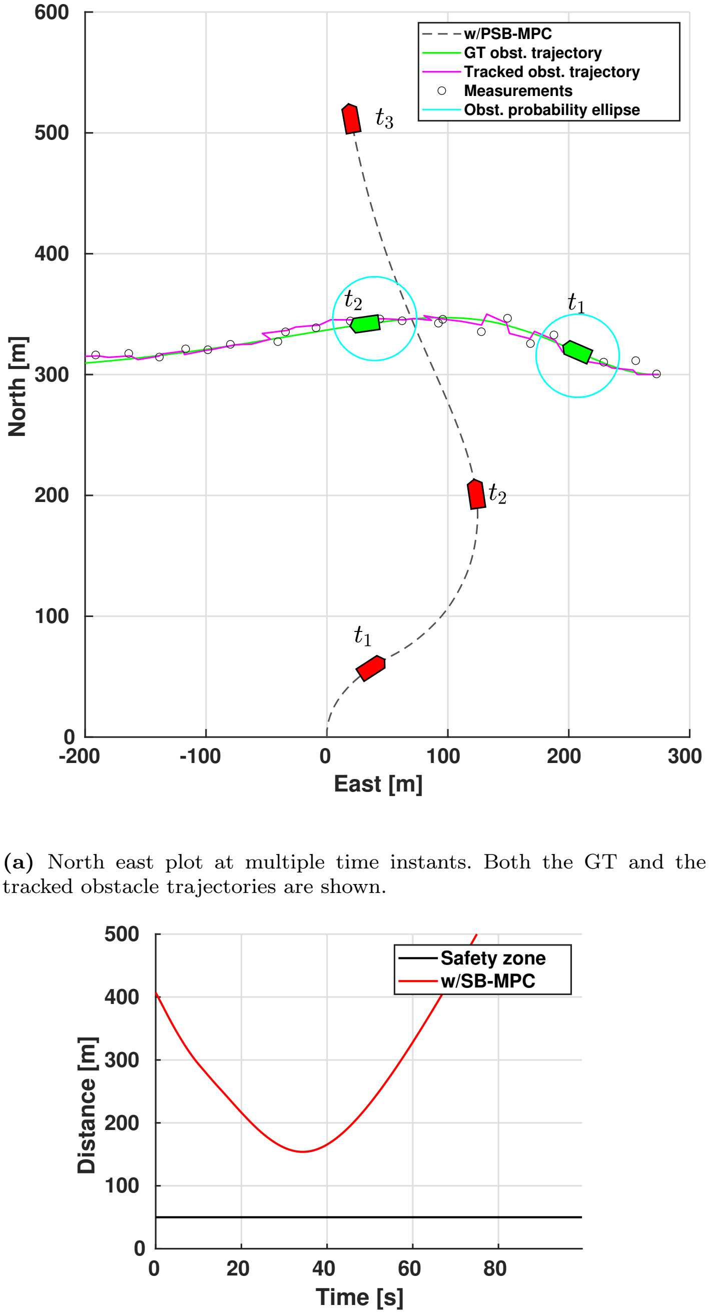


(a) North east plot at multiple time instants. Both the GT and the tracked obstacle trajectories are shown.



(b) Distance to the obstacle.

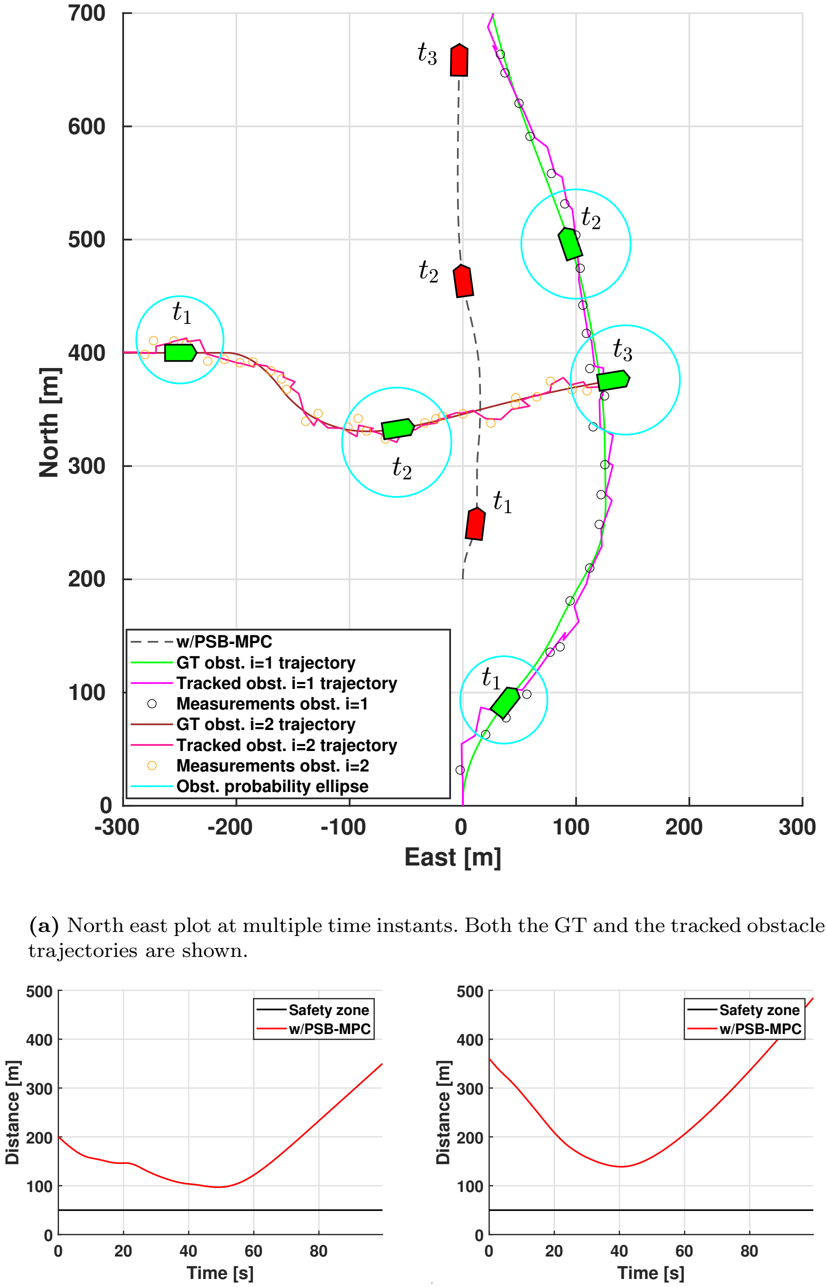
Figure 5.8: Head-on situation part two.



(a) North east plot at multiple time instants. Both the GT and the tracked obstacle trajectories are shown.

(b) Distance to the obstacle.

Figure 5.9: Crossing situation part two.



(a) North east plot at multiple time instants. Both the GT and the tracked obstacle trajectories are shown.

(b) Distance to the obstacle $i = 1$.

(c) Distance to the obstacle $i = 2$.

Figure 5.10: Combined crossing and overtaking situation part two.

Chapter 6

Real-time Feasible Probabilistic Scenario-based MPC

The following publication forms the basis of this chapter, which outlines a real-time feasible PSB-MPC algorithm that employs parallel computation. By doing this, the MPC can consider increasing amounts of possible own-ship maneuvering decisions and situational information at lower computational cost.

[99] **T. Tengesdal**, T. A. Johansen, T. D. Grande and S. Blindheim, "Ship Collision Avoidance and Anti Grounding Using Parallelized Cost Evaluation in Probabilistic Scenario-based Model Predictive Control", *IEEE Access*, 2022, Submitted.

6.1 Introduction

6.1.1 Motivation

Autonomous ships will require a high level of data processing in order to have adequate situational awareness and to make deliberate decisions. This requires efficient and robust algorithms, and well chosen platforms to enable fast computation. When facing a hazardous situation in e.g. confined space with multiple static and dynamic obstacles, the need to evaluate a larger set of future control behaviours or trajectories for the autonomous ship and other obstacles will be necessary, such that a risk minimizing or collision-free trajectory is possible to find. This can also make it easier to comply to the COLREGS [21]. However, evaluating the risk associated in any of these control behaviours can be computationally expensive. Thus, to meet run-time requirements, a COLAV planning algorithm which scales well in the evaluation of different control behaviours will be both beneficial and necessary in such cases. Increased robustness can then also result as a consequence of being able to evaluate more vessel behaviour scenarios and situational information in the system at run-time.

An SB-MPC approach is here a viable option that can incorporate most of the elements needed in a robust COLAV planning algorithm, such as anti-grounding, dynamic obstacle avoidance and multi-ship adherence to COLREGS. This is because of its flexibility in the formulation of its optimization problem, with different control objectives and possible integration of constraints, and which has a rich theoretical foundation. The sampling-based method is also flexible in the prediction models used to generate own-ship and dynamic obstacle prediction scenarios. The problem with this approach however, and especially for the probabilistic version (PSB-MPC), is that the optimization problem in the COLAV planning algorithm scales poorly with an increasing set of considered own-ship avoidance maneuvers, static obstacles and dynamic obstacles with their own alternative prediction scenarios. The prediction of the collision risk with respect to all dynamic and uncertain obstacles involved, and calculating distances to all static obstacles for anti-grounding purposes, has exponentially increasing computational cost as the optimization problem increases.

6.1.2 Literature Review

Many studies on maritime collision avoidance exists today, and are mainly summarized in review papers such as [26], [28]–[30], whereas we here focus on deliberative COLAV planning methods having dynamic obstacle avoidance and COLREGS adherence in addition to anti-grounding in their algorithms. For a general overview on planning algorithms, see [151]. A lattice-based trajectory planner using A* search for finding collision-free trajectories is introduced in [69], where non-adherence to the COLREGS, trajectory deviation and collision risk with respect to static and dynamic obstacles is penalized in the cost function. An intention based motion model is used for dynamic obstacles, which relies on learning the positional prediction uncertainty for a given scene when used in calculating collision probabilities. The details on this model is not given, and results on how the planner scales in run-time with increasing lattice grid density, dynamic and static obstacles are however not given.

The work in [152] introduces a hierarchical system with three levels. The top level trajectory planner uses lattice-based A* search combined with an Optimal Control Problem (OCP) method for generating collision-free trajectories with respect to static obstacles. A mid level MPC-based COLAV planning algorithm modifies this trajectory to adhere to the COLREGS and avoid collisions with respect to dynamic obstacles. Lastly, a low-level reactive COLAV sampling-based planning algorithm acts as a fail-safe in case the levels above can not handle the situation. The system does however assume straight line trajectories for dynamic obstacle predictions without uncertainty, which does not coincide with real-time vessel behaviour in hazardous situations. Furthermore, scalability and run-time properties with an increasingly complex situation is not discussed.

In [68], a field-test verified A-star search trajectory planner is developed, which attempts to find a COLREGS-compliant and collision-free trajectory with respect to dynamic and static obstacles in a lattice. To predict nearby dynamic obstacle trajectories, the planner employs Monte-Carlo (MC) simulation using fuzzy logic and the trajectory history of the obstacle to find a set of probable trajectories, where the most probable one is considered for collision avoidance. As in [152], the method does not consider the prediction uncertainty associated with dynamic obstacles. Furthermore, the computational efficiency of the planner only tested for a set of 500 possible own-ship trajectories and one expected dynamic obstacle trajectory, which can be inadequate in highly congested scenarios.

Candeloro et. al. 2017 [77] propose a global and local lattice-based trajectory planner which uses Voronoi Diagrams to generate a set of static obstacle collision-free waypoints, from where a continuous trajectory is generated using Fermat‘s Spiral. The method considers local replanning windows for taking detected dynamic and static obstacles into account, and predicts dynamic obstacle motion with the CV model [78]. A convex hull representing the dynamic obstacle uncertainty up until time to CPA is created from using the position estimates and error covariances from a Kalman filter, which is then regarded as an area to avoid in the planner. This may however be overly conservative, due to the unrealistic uncertainty growth in the CV model [79]. How the local replanning run-time scales with increasing windows size, dynamic and static obstacles is not considered.

NMPC for static and dynamic obstacle collision avoidance with environmental disturbance rejection was proposed in [76]. A deterministic CV model was used for the dynamic obstacle prediction, which will not be the case in real-time hazardous maritime situations where ships will maneuver. Furthermore, how the MPC scales with static and dynamic obstacles was not considered.

[83] introduces a sampling-based static and dynamic obstacle considerate trajectory planner with COLREGS-compliant COLAV planning algorithm based on Rapidly exploring Random Trees (RRTs), where a joint simulator is used to predict both the own-ship and dynamic obstacle motion. Potential fields are used in the prediction to ensure that all the vessels have collision-free trajectories with respect to each other and static obstacles. The method is shown to have beneficial run-times feasible for real-time. However, the underlying assumption in the prediction is however that ships will always perform deterministic COLREGS-compliant maneuvers if possible, which is not necessarily true in practice.

Collision avoidance within a distributed flocking control strategy based on MPC was considered in [153], with respect to nearby dynamic vehicles in the flock and static obstacles. The computational efficiency or scalability of the method was however not discussed, and the states of all vehicles involved are assumed deterministic.

6.1.3 Contributions

In this chapter, an implementation of the sampling-based PSB-MPC algorithm on a GPU platform which facilitates efficient anti-grounding and dynamic obstacle avoidance is introduced. The main contribution of the chapter compared to current state-of-the-art deliberate static and dynamic obstacle COLAV planning algorithms is the description of a parallelization algorithm for efficient cost evaluation of possible own-ship trajectories in the PSB-MPC, taking into account dynamic obstacle uncertainties and complex static obstacles in maritime hazardous situations. The algorithm is feasible for real-time, as the MPC cost function evaluation scales linearly with increasing numbers of dynamic obstacles with their own prediction scenarios and also static obstacles, due to the parallelization. Static obstacles are read in from Electronic Navigational Chart (ENC) data and processed into simplified polygons using the Ramer-Douglas-Peucker (RDP) algorithm [154]. The efficiency of the parallelized implementation makes it possible for the COLAV planning algorithm to consider more dynamic obstacle prediction scenarios and own-ship trajectories, and more complex static obstacle maps for elevated situational awareness and better trajectory planning. Furthermore, a side contribution of the chapter is that the dynamic obstacle prediction scheme in [94] is updated to use a kinematic model with incorporated LOS guidance for more realistic trajectories.

6.1.4 Chapter Overview

The chapter is organized as follows. Section 6.2 gives new background information about the fourth edition of the PSB-MPC, with prediction models, cost function structure and grounding hazard extraction and representation. An outline of a sequential implementation of the algorithm is also given. A parallelized implementation of the PSB-MPC is given in Section 6.3. Finally, Section 6.4 show simulation results with the PSB-MPC, and Section 6.5 concludes the work.

6.2 The Probabilistic Scenario-based Model Predictive Control

The third edition of the PSB-MPC described in the previous chapter is here considered, but extended to include grounding hazard consideration in addition to the classic dynamic obstacle avoidance. Furthermore, the prediction model used for dynamic obstacles and the own-ship is changed. An illustration of the COLAV problem involving both static and dynamic obstacle avoidance, is given in Fig. 6.1, where it is noted that the control behaviours selected are arbitrary.

Each control behaviour is evaluated by a cost function $\mathcal{H}^l(\cdot)$ which penalizes probabilistic collision risk, grounding risk, COLREGS violation and nominal trajectory deviation. The optimal one is selected as

$$l^*(t_0) = \arg\min_{l} \mathcal{H}^l(t_0) \tag{6.1}$$

In this chapter, a way to parallelize the evaluation of the cost in (6.1) is shown, and also for considering anti-grounding in the PSB-MPC.

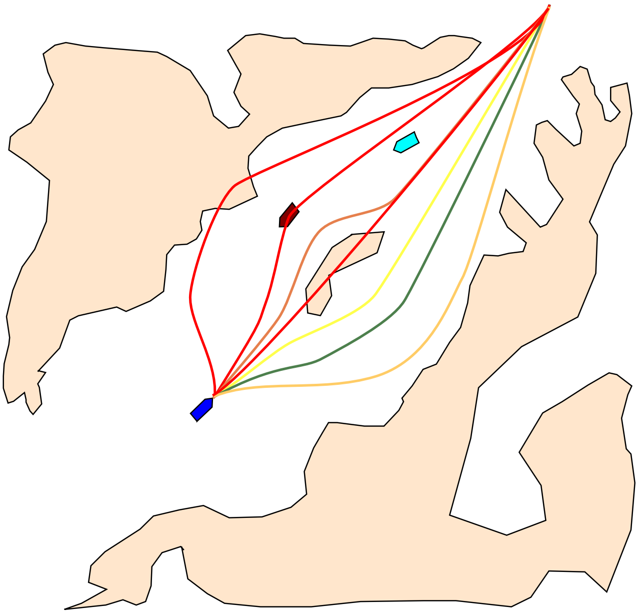


Figure 6.1: PSB-MPC illustration, with the own-ship running the algorithm in blue. Nearby dynamic obstacles are shown in cyan and brown. Grounding hazards are shown in beige. Candidate control behaviours predicted in the MPC are also shown, where the color from red to green represents their cost, with green being the lowest. Thus, the green candidate trajectory is the optimal one. The nominal trajectory goes straight north-east through the confined environment.

6.2.1 Prediction Models

Own-ship

For deliberate COLAV planning algorithms, the own-ship prediction model can be selected to be complex or simple depending on how much vessel information one has. The PSB-MPC can easily handle complex ship motion models in its framework. However, as the kinematic uncertainty associated with the ship motion prediction increases substantially with time, having a simple model to capture qualitatively the approximate own-ship behaviour is often adequate for deliberative COLAV planning algorithms, where the low-level vessel control systems (autopilot) can compensate for model inaccuracies and disturbances. As predictions of vessel motions over longer time horizons are inherently uncertain, due to environmental disturbances, future maneuvering decisions and unforeseen events, especially in hazardous situations, it is argued that there is limited gain in using an overly complex model. On the other hand, for reactive collision avoidance methods and lower level motion control with shorter prediction horizons, it will be more important to consider the ship dynamics accurately. Thus, in this chapter a discrete kinematic model as outlined in Section 2.4.3 is used

$$\begin{aligned} x_{k+1} &= x_k + U_k cos(\chi_k) \\ y_{k+1} &= y_k + U_k sin(\chi_k) \\ \chi_{k+1} &= \chi_k + \frac{1}{T_\chi}(\chi_{d,k} - \chi_k) \\ U_{k+1} &= U_k + \frac{1}{T_U}(U_{d,k} - U_k) \end{aligned} \tag{6.2}$$

is used, which describes the own-ship state $\boldsymbol{x}_k = [x_k, y_k, \chi_k, U_k]^T$ motion during the MPC prediction, using LOS guidance to predict all candidate trajectories, as described in Section 2.5.1.

This combination of a kinematic model used with LOS guidance allows for a lightweight prediction of candidate own-ship trajectory candidates. For each own-ship control behaviour, the speed modifications $u^l_{m,M}$ and course modifications $\chi^l_{m,M}$ for all n_M sequential maneuvers are applied to the LOS guidance references for speed and course at maneuvering times $t_M, M = 1, 2, ..., n_M$, now evenly spaced througout the horizon with a time spacing parameter t_{ts} for simplicity.

To create trajectories simulating the ship motion for the own-ship and dynamic obstacles forward in time, Euler‘s method is used for numerical integration. Specifically, the integration is done over the prediction horizon with discrete predicted times $t_k \in D(t_0) = \{t_0, ..., t_0 + k\Delta_{mpc}, ..., T_{mpc}\}$, with Δ_{mpc} as the time step and T_{mpc} as the prediction horizon.

Dynamic Obstacles

As one often do not have information on the underlying dynamic obstacle vessel or object, their motion models should be simple. The second and third edition of the PSB-MPC used the OU process [116] in order to predict the motion of dynamic obstacles, and allows for alternative obstacle prediction scenarios [98]. However, the trajectories only specify a single change in course, and are thus not necessarily realistic. A more realistic approach as shown in Fig. 6.2 is now employed, where more avoidance like maneuvers are used.

The predicted obstacle motion is implemented using the same kinematic model as for the own-ship in (6.2):

$$\begin{aligned} x^i_{k+1} &= x^i_k + U^i_k cos(\chi^i_k) \\ y^i_{k+1} &= y^i_k + U^i_k sin(\chi^i_k) \\ \chi^i_{k+1} &= \chi^i_k + \frac{1}{T_\chi}(\chi^i_{d,k} - \chi^i_k) \\ U^i_{k+1} &= U^i_k + \frac{1}{T_U}(U^i_{d,k} - U^i_k) \end{aligned} \tag{6.3}$$

where the superscript i is used for dynamic obstacles as usual. LOS guidance [155] is again used to specify $\chi^i_{d,k}$, when assuming the obstacle has a nominal trajectory along a straight line which can be parametrized by waypoints. Its current speed estimate from the tracking system is used as $U^i_{d,k}$. To branch out on different scenarios as shown in Fig. 6.2, multiples of an artificial cross-track error r_{ct} is added to the actual cross track error in the LOS guidance.

The obstacle kinematic uncertainty is again predicted forward in time *heuristically* using an OU-process as used in the previous two chapters:

$$\boldsymbol{P}^i_{k+1} = \boldsymbol{P}^i_0 + \boldsymbol{\Sigma}_1 \circ \boldsymbol{\Sigma}_2(t_{k+1} - t_0) \tag{6.4}$$

The 3σ positional uncertainty is heuristically bounded by r_{ct} in the prediction, such that each obstacle trajectory has a tube uncertainty with approximate radius r_{ct}. As for the own-ship, the parameters for the obstacle prediction are all dependent on the type of ship, ship control system, ship captain etc., and should be estimated using available data about the obstacle.

As mentioned, it is assumed that the obstacle nominal trajectory is a straight line from its current course, where waypoints on this line. Vessel to vessel communication or e.g. road map methods [156] may be used to predict the nominal obstacle trajectory in more confined spaces where the straight line trajectory assumption is restrictive. An illustration of the uncertainty prediction together with the dynamic obstacle trajectories is shown for a case with a non-straight line obstacle trajectory in Fig. 6.3. One should here choose a sufficiently large number of prediction scenarios such that the obstacle maneuvering uncertainty is covered.

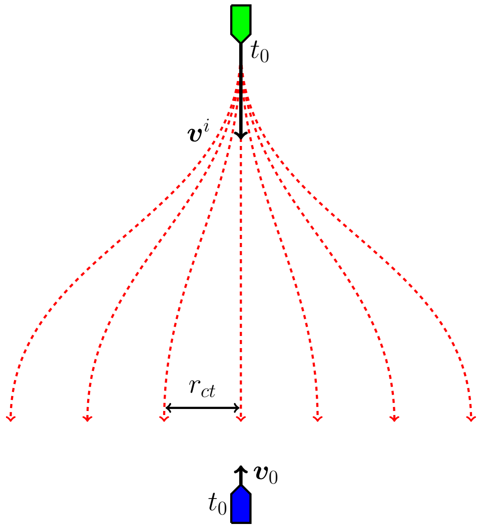


Figure 6.2: Head-on scenario with obstacle i in green and own-ship in blue. Their velocity vectors $\boldsymbol{v}^i$ and $\boldsymbol{v}_0$, respectively, are also shown. The updated prediction scheme using LOS-guidance allows for the obstacle to make realistic alternative maneuvers to port and starboard. The stationary time spacing between trajectories is determined by r_{ct}.

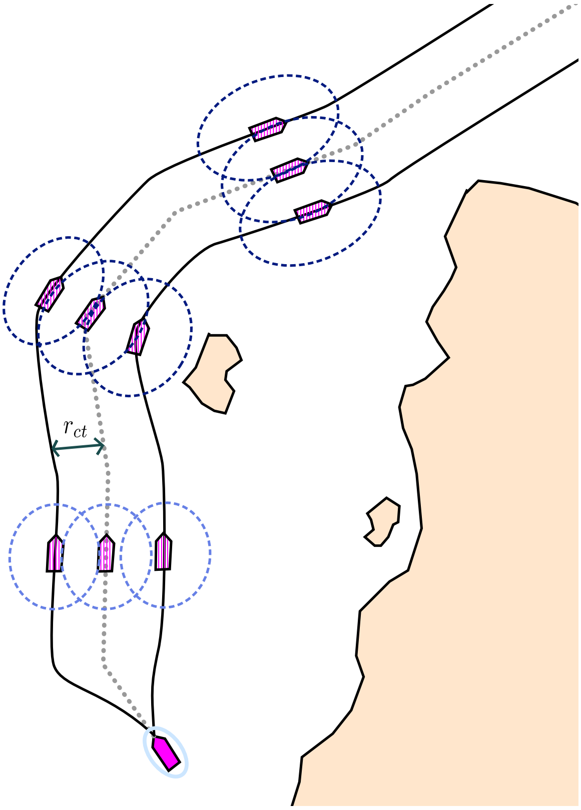


Figure 6.3: Dynamic obstacle prediction illustration with an obstacle in purple. Three prediction scenarios are shown, all starting at t_0, where the vessel is depicted in full purple with its tracked estimation error covariance represented around it as a 3σ probability ellipse in light blue. The nominal predicted obstacle trajectory is shown with the grey dotted line, whereas the alternative scenarios are spaced r_{ct} apart.

6.2.2 Grounding Hazards

The static obstacles or grounding hazards considered in the PSB-MPC are parameterized as two-dimensional polygons. Here, polygons are read in from shapefiles using the C based library Shapefile C Library, which are generated using the Electronic Navigational Chart processing module in seacharts corresponding to the relevant map region considered [157]. If real-time sensor data is available, this can also be used to update the polygons used in the MPC.

Because electronic map data can have high accuracy, larger polygons extracted can have tens of thousands of vertices. However, for collision avoidance, this level of detail is not necessary, and the polygons should thus be simplified in order to save computation time in the algorithm. One of the earliest and most common curve simplification methods that can be used for polygon simplification is the RDP algorithm [154]. The method recursively simplifies a curve of points by consecutively

Algorithm 2 The Ramer-Douglas-Peucker curve simplification algorithm.

```
function RDP($Points$, $\epsilon_{rdp}$)
    $d_{max} \leftarrow 0$, $j \leftarrow 1$, $end \leftarrow length(Points)$.
    for $i = 2, .., end$ do
        $d \leftarrow perpendicularDistance(Points[i],$
        $Points[1], Points[end])$
        if $d > d_{max}$ then
            $j \leftarrow i$, $d_{max} \leftarrow d$
        end if
    end for
    if $d_{max} > \epsilon_{rdp}$ then
        $rResults1 = $RDP$(Points[1, .., j], \epsilon)$
        $rResults2 = $RDP$(Points[j, .., end], \epsilon)$
        $newPoints = \{rResults1[1, ..,$
        $length(rResults1) - 1], rResults2\}$
    else
        $newPoints = \{Points[1], Points[end]\}$
    end if
    return $newPoints$
end function
```

considering its line segments, pruning away points which are further away from the considered line segment than a specified threshold ϵ_{rdp}. The distance tolerance parameter ϵ should be chosen as not to overly simplify the polygons, preserving as much structure as possible. The method is summarized in Algorithm 2. A graphical illustration of the algorithm can be found in https://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm. The distance from the own-ship center to nearby polygons is used in the PSB-MPC grounding cost. It is obtained by using a point to polygon calculation method [158], using the ray intersection method for determining if the own-ship center point is inside the polygon, which is suitable for both convex and concave polygons.

6.2.3 Cost function reformulation

In this chapter, the following restructuring of the PSB-MPC cost function is considered

$$\mathcal{H}^l(t_0) = \mathcal{H}^l_{do} + \mathcal{H}^l_{colregs} + \mathcal{H}^l_{so} + \mathcal{H}^l_{p}, \tag{6.5}$$

for a control behaviour l, where the four terms are the cost associated with dynamic obstacles, COLREGS violation, static obstacles or grounding hazards and trajectory tracking, respectively. The dynamic obstacle related cost is reformulated to

$$\mathcal{H}^l_{do} = \sum_{i=1}^{n_{do}} w^i \mathcal{H}^{l,i}_{do} \tag{6.6}$$

where w^i represent the weight of the cost from obstacle i, in general influenced by factors such as distance, bearing, nearby grounding hazards and vessel-vessel communication. If no prior information is used, it is set to $w^i = 1$. The dynamic obstacle i cost is given by

$$\mathcal{H}_{do}^{l,i} = \sum_{s=1}^{n_{ps}^i} \hat{\mathbb{P}}_s^i C_s^{l,i} \tag{6.7}$$

where n_{ps}^i is the number of prediction scenarios for the obstacle, $\hat{\mathbb{P}}_s^i$ represent the associated prediction scenario probabilities from an intention inference module [98][96], and $C_s^{l,i}$ is the cost involving prediction scenario s for obstacle i, given as

$$C_s^{l,i} = \max_k \zeta_i \mathcal{C}_{i,k}^{l,s} \hat{\mathbb{P}}_{c,k}^{l,i,s} exp(-t_k/T_{coll}) \tag{6.8}$$

which is taken as the maximum of the probabilistic collision risk, involving the relative kinetic energy term $\mathcal{C}_i^{l,s}(t)$ as defined in (2.36) from Section 2.6. The variable $\hat{\mathbb{P}}_{c,k}^{l,i,s}$ is the collision probability estimate calculated using the CE method, introduced in the previous chapter [94]. An exponential discounting term with time constant T_{coll} gives lower weighting of collision events far ahead in the future.

The intention uncertainty of a dynamic obstacle is represented through the scenario probabilities $\hat{\mathbb{P}}_s^i$ for each considered obstacle prediction scenario. Given a representable set of obstacle prediction scenarios, we are able to cover most anticipated obstacle maneuvering cases because we predict the uncertainty for each of the scenarios. These probabilities of different target ship plans or trajectories are typically inferred by an intention model as outlined in Chapter 4 and embodied in [96], and can be used for having elevated situational awareness in the planner. Furthermore, the probabilities are an adequate way of taking into account intention information, as they are easy to interpret, can be used to define risk and leads to a natural way of weighting the collision risk associated with different decision candidates for an obstacle ship. The downside is that one needs a validated intention inference model, and a sufficient set of dynamic obstacle prediction scenarios in order to have meaningful estimates.

To favor COLREGS compliance in multi-ship situations, the COLREGS related cost is now separated into its own term in the PSB-MPC, and given as

$$\mathcal{H}_{colregs}^{l} = \kappa \sum_{i=1}^{n_{do}} w^i \mu^{l,i} \tag{6.9}$$

with the tuning parameter κ and

$$\mu^{l,i} = \sum_{s=1}^{n_{ps}^i} \hat{\mathbb{P}}_s^i \mu_s^{l,i}, \tag{6.10}$$

with $\mu_s^{l,i} \in \{0, 1\}$ as the indicator of the own-ship following control behaviour l violating COLREGS with respect to obstacle i in prediction scenario s, calculated as in Section 2.6. The parameters w^i and dynamic obstacle scenario probabilities are again used for weighting purposes. The new formulation now penalizes COLREGS violations of rules 13-17, with respect to all dynamic obstacles, and allows for better handling of compliance in multi-ship situations.

The static obstacle related cost or grounding cost is parameterized as

$$\mathcal{H}^{l}_{so} = \max_{j} \mathcal{H}^{l,j}_{so} \tag{6.11}$$

where

$$\mathcal{H}^{l,j}_{so} = \max_{k} \left[(G_1 + G_2 \phi^{l}_{j,k} V_w^2) exp(-(G_3 |d^{l}_{0j,k} - d_{safe}| + G_4 t_k)) \right] \tag{6.12}$$

inspired by [159], where G_1 to G_4 are tuning parameters, V_w the estimated wind speed, $\phi^{l}_{j,k} = \max(0, \boldsymbol{\omega} \cdot \boldsymbol{L}^{l}_{0j,k})$ with $\boldsymbol{\omega}$ as the wind direction unit vector. The unit vector $\boldsymbol{L}^{l}_{0j,k}$ points from the own-ship to the static obstacle j, with $d^{l}_{0j,k}$ as the corresponding distance between them.

The trajectory deviation cost is given as

$$\mathcal{H}^{l}_{p} = \frac{1}{n_M} \sum_{M=1}^{n_M} f(\cdot) + \frac{1}{n_M - 1} \sum_{M=2}^{n_M} h(\cdot) \tag{6.13}$$

with $f(\cdot)$ and $h(\cdot)$ as the control deviation and change cost, described in Section 4.4 of Chapter 4.

6.2.4 Standard PSB-MPC Implementation

As the PSB-MPC is a FCS-MPC, the solution to the non-convex Mixed Integer Nonlinear Programming (MINLP) problem in (6.1) is parameterized by the chosen discrete set of own-ship control behaviours or controls. The benefit of the finite-set MPC formulation is that by brute force iterating over the set of control behaviours one is able to find a global solution, which would be hard in the case if numerical optimization was used.

Implementing the cost evaluation in the PSB-MPC on a sequential computing platform will involve loops over the own-ship control behaviours, where loops over static and dynamic obstacles in their set of prediction scenarios are found within. This would look something like the method outlined in Algorithm 3. One can see that this implementation involves several nested for loops, especially the one over dynamic obstacles and their prediction scenarios. In addition, one must also loop over the number of discrete samples $t_k \in D(t_0)$ in the predicted trajectories. Thus, the MPC problem will scale poorly with increasing number of control behaviours, static and dynamic obstacles.

6.3 Parallelized PSB-MPC Implementation

The nature of the FCS-MPC described in the above section makes it possible to independently evaluate the cost associated with the control behaviours, and thus

Algorithm 3 Standard PSB-MPC cost evaluation on a sequential processing platform, assuming all obstacle prediction scenarios are generated beforehand.

```
Initialize optimal control behaviour to $l^* = 1$.
for $l = 1, .., n_{cbs}$ do
    Predict the own-ship trajectory following control behaviour $l$.
    Calculate the trajectory related cost $\mathcal{H}_p^l$ using (6.13).
    for $j = 1, .., n_{so}$ do
        Calculate the static obstacle $j$ grounding cost $\mathcal{H}_{so}^{l,j}$ using (6.12).
    end for
    Calculate total grounding cost $\mathcal{H}_{so}^l$ using (6.11).
    for $i = 1, .., n_{do}$ do
        for $s = 1, .., n_{ps}^i$ do
            Calculate probabilistic collision cost $C_s^{l,i}$ from (6.8) and COLREGS indicator $\mu_s^{l,i}$ in (6.10).
        end for
        Calculate dynamic obstacle $i$ cost $\mathcal{H}_{do}^{l,i}$ using (6.7).
    end for
    Calculate total dynamic obstacle cost $\mathcal{H}_{do}^l$ using (6.6).
    Calculate control behaviour cost $\mathcal{H}^l(t_0) = \mathcal{H}_{do}^l + \mathcal{H}_{colregs}^l + \mathcal{H}_{so}^l + \mathcal{H}_p^l$.
    if $\mathcal{H}^l(t_0) < \mathcal{H}^{l^*}(t_0)$ then
        Set $l^* = l$.
    end if
end for
```

apply parallelism in the main part of the algorithm. Furthermore, all obstacle prediction scenarios are assumed to be independent of the own-ship control behaviour and can be generated beforehand. This is deemed reasonable as maneuvering uncertainty is taken into account in the obstacle prediction.

When considering large amounts of situational information and a dense set of possible own-ship trajectories, evaluating the cost of an own-ship control behaviour sequentially will not make the COLAV real-time feasible. Parallelizing the cost evaluation will allow for more refined own-ship decision making, as more own-ship trajectories can be considered. Also, more static obstacles and prediction scenarios for dynamic obstacles can then be considered, resulting in increased situational awareness for the own-ship. The limiting factor here will then be how many threads that can be scheduled on the parallel computation platform.

A naive way of cost function evaluation parallelization would be to schedule GPU threads to evaluate the cost (6.5). However, this is a big task for a single thread, as it among others involves going through all static and dynamic obstacles in all their prediction scenarios to find the total cost. This equates to a nested for loop over obstacles, prediction scenarios and discrete time samples in the code that implements the MPC as in Algorithm 3, and will scale poorly with an increase in the number of obstacles and number of prediction scenarios n_{ps}^i for dynamic obstacles. As GPU cores have limited processing power compared to CPU cores, their tasks should be as lightweight as possible.

Two of the main bottlenecks in the cost evaluation is calculating the distance to static obstacles and the estimation of collision probabilities. The first bottleneck is readily apparent when considering large polygons with tens of thousands of vertices. However, the RDP algorithm will reduce the number of vertices in a polygon and thus alleviate computational effort. Reducing the number of time steps to evaluate the grounding cost can also aid in fixing this problem.

For the second bottleneck, giving each thread the job of estimating collision probabilities associated with only a pair of trajectories will give higher throughput, at the cost of scheduling more threads on the GPU and therefore having higher memory demands. However, as GPU technology continue to improve with respect to single core processing power and device memory, this is deemed a worthy trade-off. Furthermore, the calculation efficiency using the CE method for collision probability estimation [94] is increased by estimating $\hat{\mathbb{P}}_c^{l,i,s} \approx 0$ when the predicted distance between the own-ship and an obstacle is larger than $d_{safe} + 4\sigma^i_{largest}$, where $\sigma^i_{largest}$ is the standard deviation along the axis where obstacle i has the largest predicted positional uncertainty.

Thus, a way to solve the bottlenecks in (6.1) utilizing parallel processing can be done in two steps: First schedule n_{cbs} threads to predict the own-ship trajectory and calculate the trajectory related cost (6.13) for each control behaviour $l = 1, 2, .., n_{cbs}$. Then, schedule

$$n_{ct} = n_{cbs} \cdot (n_{so} + \sum_{i=1}^{n_{obst}} n^i_{ps}) \tag{6.14}$$

threads that evaluates the cost (6.12), (6.8) and the COLREGS violation indicator in (6.10). The total cost (6.5) is finally stitched together afterwards on the CPU. This way, no GPU thread has run-times dependent on large nested for-loops, and the MPC-problem scales better with increasing number of obstacles and dynamic obstacle prediction scenarios. This approach of using parallelization to solving (6.1) can be summarized in Algorithm 4.

Note that how the PSB-MPC algorithm is implemented both on the CPU and GPU will have big impacts on the obtained run-time results. Hardware, programming language and software libraries used will be significant factors here. An alternative to the structure in algorithm 4 would be to have separate kernels to evaluate the static and dynamic obstacle partial costs. This could be better suiting for a setup with multiple GPUs, as the two kernels could then be run concurrently. Lastly, because of the extra latency overhead due to porting data from the host (CPU) to the device (GPU), as much memory as possible for the relevant data needed on the GPU should be pre-allocated.

Algorithm 4 Parallelized PSB-MPC cost evaluation, assuming all obstacle prediction scenarios are generated beforehand.

```
Schedule $n_{cbs}$ GPU threads, transferring all the required data for own-ship trajectory prediction and calculating (6.13).
parfor $l = 1, .., n_{cbs}$ do
    Predict the own-ship trajectory following control behaviour $l$, save trajectory in GPU memory for use by the subsequent processing.
    Calculate the trajectory related cost $\mathcal{H}_p^l$ using (6.13).
    Return the results to CPU memory.
end parfor
Schedule $n_{ct}$ GPU threads, transferring all the required data needed for partial static and dynamic obstacle cost evaluation.
parfor $ct = 1, .., n_{ct}$ do
    Extract control behaviour $l$, static obstacle $j$ or dynamic obstacle $i$ and prediction scenario $s$ to consider.
    Calculate the grounding cost $\mathcal{H}_{so}^{l,j}$ using (6.12) or $C_s^{l,i}$ using (6.8) and the indicator $\mu_s^{l,i}$, depending on if a static or dynamic obstacle is considered in the thread.
    Return the results to CPU memory.
end parfor
Use all the calculated $\mathcal{H}_{so}^{l,j}$ to calculate $\mathcal{H}_{so}^l$ using (6.11).
Use all the calculated $C_s^{l,i}$ and $\mu_s^{l,i}$ plus other relevant data to calculate $\mathcal{H}_{do}^l$ using (6.6) and $\mathcal{H}_{colregs}^l$ using (6.9).
Finally, calculate (6.5) for all control behaviours using the previously calculated terms $\mathcal{H}_{do}^l$, $\mathcal{H}_{colregs}^l$, $\mathcal{H}_{so}^l$ and $\mathcal{H}_p^l$ of the cost function, and extract the optimal one $l^*$ giving minimal cost.
```

6.4 Simulation Study

The GPU-based PSB-MPC is tested in two situations to illustrate that the COLAV planning algorithm can tackle dynamic obstacles with uncertainties in addition to grounding hazards. The first river scenario is chosen to test how the COLAV planning method handles avoidance in confined spaces, whereas the second scenario aims to test the algorithm performance in a longer time horizon with multiple dynamic obstacles in a mix of an open sea area and a narrow channel. The setup with tracking system and parameters are similar to that in [94], where the obstacle tracker is deliberately tuned conservatively to test the MPC robustness against kinematic uncertainty. The situations are described below, with a number of $N_{MC} = 50$ MC simulations used for each situation. A run-time analysis considering the first situation is performed, comparing the CPU and GPU implementations of the PSB-MPC, Algorithm 3 and 4, respectively. The CPU version evaluates the PSB-MPC cost for all own-ship control behaviours sequentially on CPU cores. The simulations are performed on a work station with an Intel(R) Core(TM) i9-10900K 3.70 GHz processor, with 32 GB RAM and an NVIDIA GeForce RTX 3090 GPU. C++ is used to implement the CPU version of the PSB-MPC, whereas C++ and

CUDA is used for the GPU version.

1. Head-on scenario in Nidelva in Trondheim, Norway. The own-ship travels upstream with constant speed 2 m/s, whereas two dynamic obstacles travels downstream with constant speed 2 m/s. Vessels of lengths 5 m are here considered, and an own-ship safety zone of $d_{safe} = 5$ m is used.
2. Multi-ship situation with grounding hazards near Sakshaug, Trøndelag in Norway. Dynamic obstacle $i = 1$ is traveling from the south through Straumen with constant speed 5 m/s and ends up in an overtaking situation with respect to the own-ship, whereas dynamic obstacle $i = 2$ travels east-west through Straumen with constant speed 6 m/s and ends up in head-on situations with respect to the other vessels. Obstacle $i = 3$ travels with speed 7 m/s east-west from Straumen towards the own-ship in a head-on situation, and obstacle $i = 4$ just north-east of the own-ship travels south with speed 8 m/s. The own-ship travels with constant speed 7 m/s. Vessels of lengths 10 m are considered, and an own-ship safety zone of $d_{safe} = 10$ m is used. In addition to COLREGS adherence with respect to multiple ships, the challenge here is voyage through the narrow passage in Straumen, beneath the bridge which has two pylons that the vessels have to avoid.

For simplicity, a uniform set of scenario probabilities $\hat{\mathbb{P}}_s^i$ are defined for the dynamic obstacles, which resembles a conservative case when no prior information from intent inference is available. For the grounding hazards, only polygons within a range d_{so} are considered, to reduce computation time. Waypoints for the own-ship are set in a way that a top level planner could generate, but with small margins to static obstacles, such that the anti-grounding part of the PSB-MPC becomes important. Furthermore, the waypoints are set such that a nominal collision-free trajectory does not exist for all vessels involved.

The MPC is tuned such that anti-grounding and collision avoidance is prioritized over adhering to COLREGS and following the nominal trajectory. Naturally, because river voyage is different from sea voyage, the PSB-MPC has a different tuning for the two situations. Important parameters for the first situation tuning are given in Table 6.1.

6.4.1 Nidelva Situation

Results for the first situation are given in Fig. 6.4. The dynamic obstacles are here assumed to be intelligent running their own PSB-MPC algorithm to simulate human behaviour. The conservative tracking system tuning will create an extra challenge for the COLAV planning algorithm, with higher kinematic obstacle uncertainty. Despite this and nearby grounding hazards, all vessels involved are able to avoid collision and grounding in addition to adhering to the COLREGS rules 8, 14 and 16 related to clear actions, head-on situation and actions for give-way vessels, respectively. The near constant minimum distance to the closest static obstacle in the statistics is because the own-ship is closest to a grounding hazard

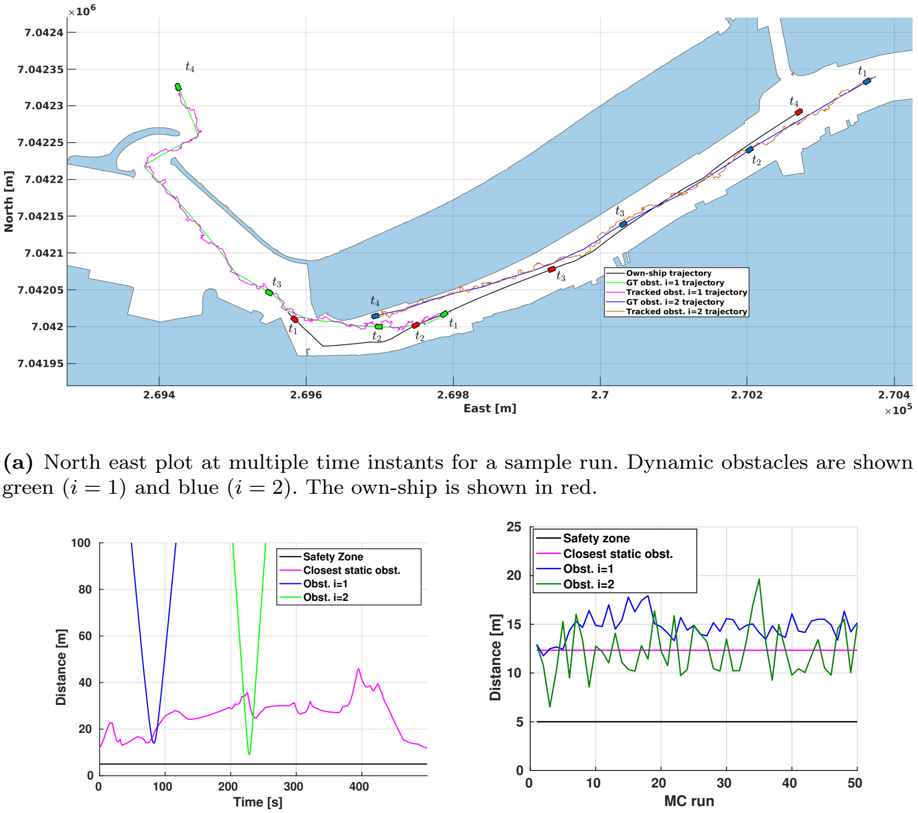
Table 6.1: Important PSB-MPC parameters for the Nidelva situation.

Parameter	**Value**	Comment
ϵ	$2\,\mathrm{m}$	RDP distance threshold
T_{MPC}	$150\,\mathrm{s}$	Prediction horizon
T_s	$0.5\,\mathrm{s}$	Prediction time step
T_{coll}	$100\,\mathrm{s}$	Collision cost time discounting parameter
n_{ps}^{LOS}	5	Number of LOS prediction scenarios
r_{ct}	$10.0\,\mathrm{m}$	Prediction scenario spacing
d_{so}	$200.0\,\mathrm{m}$	Static obstacle consideration range
n_M	2	Number of sequential avoidance maneuvers
$U_{offsets,1}$	$\{1.0, 0.5, 0.0\}$	Speed offsets first maneuver
$U_{offsets,2}$	$\{1.0, 0.5\}$	Speed offsets second maneuver
$\chi_{offsets,1}$	$\{-60, -45, -30, -15, -10, -5, 0, 5, 10, 15, 30, 45, 60\}$	Course offsets first maneuver
$\chi_{offsets,2}$	$\{-60, -45, -30, -15, -10, -5, 0, 5, 10, 15, 30, 45, 60\}$	Course offsets second maneuver

initially. Note that the map data for the river area do not include the piers at which boats are docked, which would be taken into account through usage of e.g. LIDAR data in a real-time application.

For the situation in Nidelva, a run-time analysis was performed with respect to the number of control behaviours n_{cbs} for the MPC, and the number of dynamic obstacle prediction scenarios n_{ps}^i considered. The number of control behaviours is increased by increasing the number of sequential maneuvers n_M in the horizon, and by expanding the finite set of course and surge modifications. Both the CPU and GPU implementations were run for N_{MC} simulations for each parameter setting. Figs. 6.5, 6.6 and 6.7 show a box-plot representation of the results. The GPU-implementation of the PSB-MPC performs better than the CPU-version when the number of control behaviours increase beyond a thousand. With $n_{cbs} < 1000$ and a scheduled number of threads $n_{ct} < 5000$, the overhead of launching the GPU kernels becomes too large compared to the gain of parallelized cost evaluation. This makes the CPU-implementation feasible for cases where typically $n_M = 1$ and a small number of possible course and speed changes is enough, and only a small number of static and dynamic obstacles are considered.

Furthermore, one can see that the CPU implementation performs better than the GPU implementation when considering increasing numbers of prediction scenarios up until $n_{ps}^i = 101$ for dynamic obstacles, when using a low number of control



(a) North east plot at multiple time instants for a sample run. Dynamic obstacles are shown in green ($i = 1$) and blue ($i = 2$). The own-ship is shown in red.

(b) Distance to static and dynamic obstacles for the sample run.

(c) Statistics on the minimum distance to the obstacles over all N_{MC} simulation runs.

Figure 6.4: Results for the situation in Nidelva with multiple obstacles.

behaviours $n_{cbs} = 39$. In this case, the GPU run-time is mainly caused by the overhead of porting data back and forth between the host and device side. The contrary result is the case when considering $n_{cbs} > 1000$. This is again because a CPU is optimized for fast sequential execution on fewer but more complex tasks, whereas a GPU is optimized for execution of many simple tasks in parallel. A similar result is obtained by increasing n_{obst} while keeping n_{ps}^{i} constant, but will not be reported here.

From Figs. 6.5 - 6.7, an approximate linear scaling of the MPC run-time complexity with increasing own-ship control behaviours, dynamic obstacle scenarios and static obstacles can be found. For static obstacles represented as polygons, one also have to take into account the added run-time complexity due to the number of vertices in the polygons.

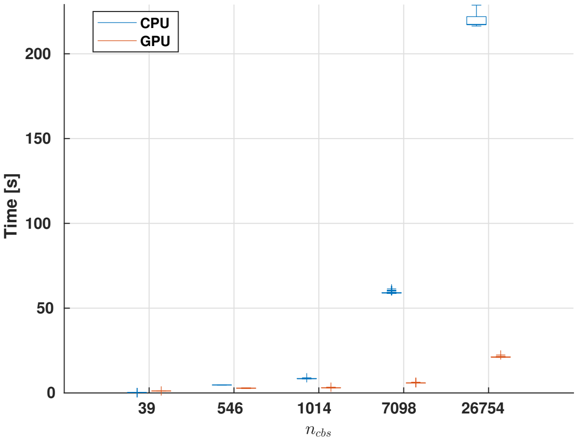


Figure 6.5: Box-plot representation of the runtime results with respect to increasing numbers of control behaviours n_{cbs}, when keeping the number of dynamic obstacle prediction scenarios constant at $n^i_{ps} = 1$.

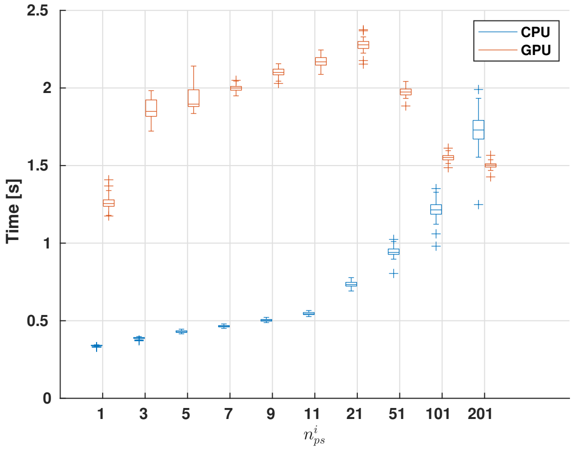


Figure 6.6: Box-plot representation of the runtime results with respect to increasing dynamic obstacle prediction scenarios n^i_{ps}, when keeping the number of own-ship avoidance maneuvers constant at $n_M = 1$ and a total number of control behaviours $n_{cbs} = 39$.

Also, tests to compare the run-time related to calculating predominantly the grounding cost in the MPC on a CPU and GPU platform was performed, when the own-ship is located in Nidelva standing still. No dynamic obstacles are considered, and thus the calculation of the distance to static obstacles will be the bottleneck. The largest static obstacle in the region is a polygon with 21962 vertices originally, and has 1734 vertices after application of the RDP algorithm. The map environment around Nidelva in Trondheim is illustrated in Fig. 6.8, where the static obstacle

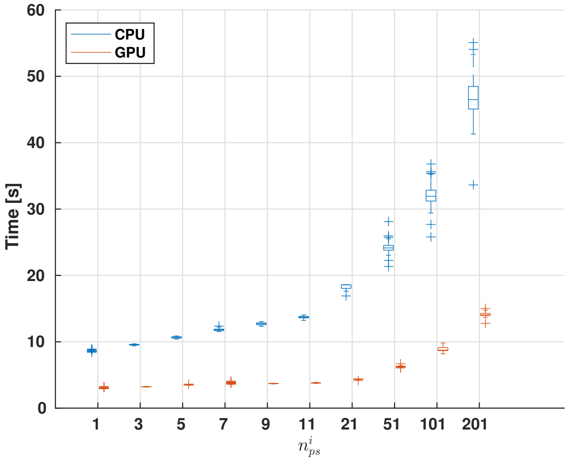


Figure 6.7: Box-plot representation of the runtime results with respect to increasing dynamic obstacle prediction scenarios n_{ps}^{i}, when keeping the number of own-ship avoidance maneuvers constant at $n_M = 2$ and a total number of control behaviours $n_{cbs} = 1014$.

$j = 13$ is the largest one with 21962 vertices. Information about the number of vertices for each polygon is given in Table 6.2.

Table 6.2: Polygon vertices before and after applying RDP on the Nidelva environment.

Polygon	Vertices before	Vertices after
1	8	3
2	207	27
3	649	95
4	322	33
5	140	18
6	890	53
7	207	48
8	8	3
9	8	5
10	1633	187
11	2110	162
12	2483	143
13	21961	1734

The first test compares the run-time when only considering the largest polygon, with and without usage of the RDP algorithm. This is a worst case scenario, as a real-time anti-grounding system should preprocess large polygons such that only the relevant local part is considered. The test is included for completeness, as it shows the importance of polygon preprocessing. Results are here given in Fig. 6.9.

The results in Fig. 6.10 show a run-time analysis for increasing numbers of static

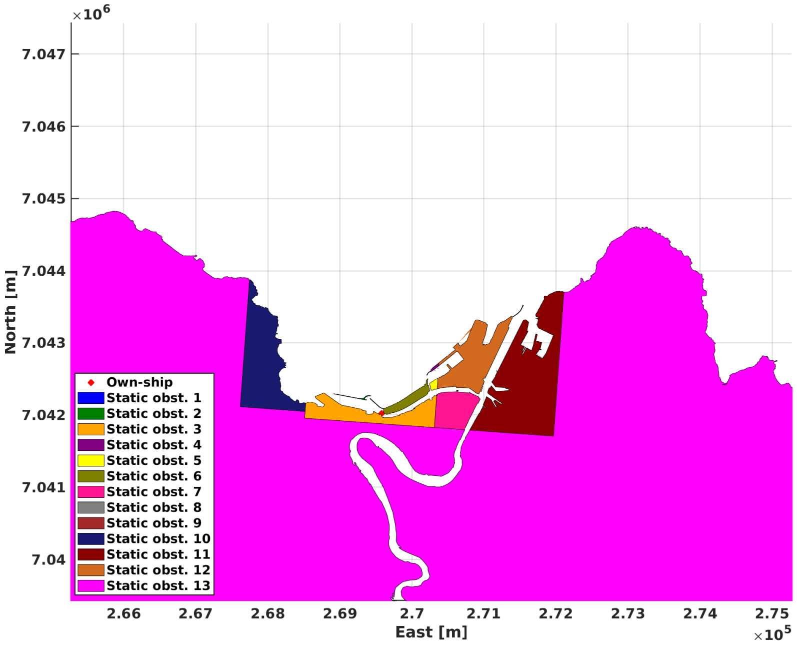


Figure 6.8: Map of the Trondheim region with Nidelva in the middle, with all relevant polygons labelled with different colors. The own-ship position is the small red dot in Nidelva in the middle.

obstacles, after using the RDP for polygon simplification. Note that the results considering an increasing number of static obstacles are strongly dependent on the number of vertices for each obstacle, which varies from 3 to 1734 vertices as seen from Table 6.2 after using RDP on this environment. This is why there is a sharp increase in average run-time when $n_{so} = 13$, because the largest polygon is then included in the consideration. An approximate linear run-time increase can however be found when considering polygons of fairly the same complexity. The trend from these results is that the GPU implementation becomes more feasible than the CPU one when the number of scheduled parallel threads n_{ct} surpasses around 5000.

6.4.2 Sakshaug Situation

Important parameters for the tuning are given in Table 6.3, with results shown in Fig. 6.11 and 6.12. The first case show results when only the own-ship has a COLAV planning algorithm, whereas the second case show results when all vessels involved use the PSB-MPC. For the first case, waypoints for the obstacles are set such that they will not collide with each other, but would collide with the own-ship if no COLAV planning algorithm was used.

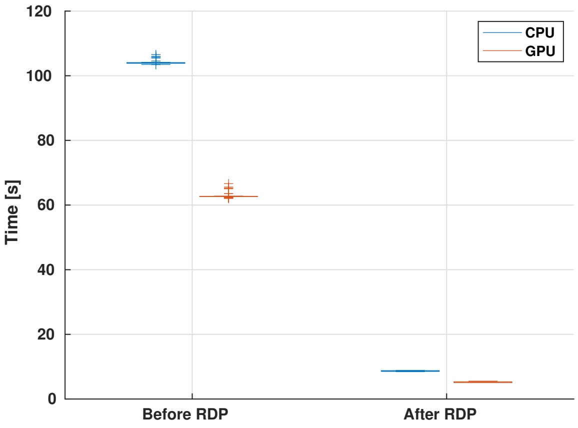


Figure 6.9: Box-plot representation of the runtime results with respect to the worst case polygon scenario before and after applying RDP, when keeping the number of own-ship avoidance maneuvers constant at $n_M = 2$ and a total number of control behaviours $n_{cbs} = 1014$.

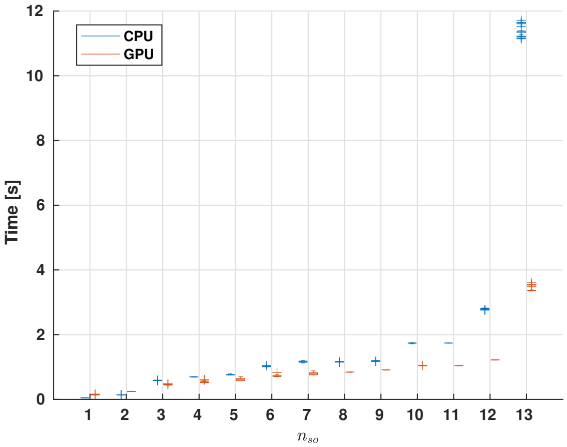


Figure 6.10: Box-plot representation of the runtime results with respect to increasing numbers of static obstacles n_{so}, when keeping the number of own-ship avoidance maneuvers constant at $n_M = 2$ and a total number of control behaviours $n_{cbs} = 1014$.

For both the first and second case, the own-ship has difficulties with overtaking purple obstacle $i = 1$ while simultaneously avoid grounding and avoiding blue obstacle $i = 2$ head-on, that adheres to both COLREGS rules 13 and 14 regarding overtaking and head-on. Especially in the time period between t_2 and t_3, the own-ship struggles with figuring out the side to overtake obstacle $i = 1$ on when entering Straumen, hence the oscillations in the trajectory in this period. The black obstacle

$i = 3$ and green obstacle $i = 4$ are easier to avoid as the vessels are here less constrained by land.

Thus, the own-ship is in general able to avoid collision with all obstacles in both cases, but COLREGS adherence in the narrow passage is difficult to accomplish with respect to all ships. This is mainly due to constant conservative intent information being used, with uniform prediction scenario probabilities for dynamic obstacle trajectories, essentially assuming that no dynamic obstacle will have specific inclinations towards adhering to the COLREGS. Also needing to avoid grounding in the narrow passage further restricts the PSB-MPC‘s ability to adhere to COLREGS in a safe manner. The algorithm is however able to keep safe distance to all obstacles in all Monte Carlo simulation runs. The diversity of the environment makes algorithm tuning challenging, as one can argue that the COLAV planning algorithm parameters should be adaptive based on changes in the situation.

When the dynamic obstacles do not explicitly follow COLREGS in the first case, the own-ship can be more excused for not doing the same with respect to all vessels. For the second case, one see the potential for vessel-vessel communication to explicitly reduce trajectory uncertainties and adhere to COLREGS, during the passage through Straumen. Addressing these issues is the topic of future research more focused on multi-ship COLREGS compliance in confined waters.

Regarding run-time complexity for this example, it will be similar as for the first situation when considering increasing dynamic obstacles and their prediction scenarios. There will be a small increase in the run-time due to the Sakshaug situation has larger and more complex static obstacles, although a smaller set than for the Nidelva situation is considered in the proximity of the own-ship. In total, run-time results generated for this example would be fairly similar to the first simulation, albeit with a bias on the static obstacle run-time complexity due to larger obstacles considered.

6.5 Conclusion

The PSB-MPC COLAV planning algorithm presented in this chapter facilitates both dynamic and static obstacle avoidance, with the most performance-critical part of its algorithm implemented on the GPU. What separates it from current state-of-the-art is the computational speed of the algorithm, where the cost evaluation is parallelized such that the MPC problem scales approximately linearly with increasing control behaviours, static and dynamic obstacles and prediction scenarios, as shown in the run-time results presented. This makes the COLAV planner able to consider more control behaviours and dynamic obstacle prediction scenarios efficiently, which results in real-time capabilities and performance gains in cases where large amounts of situational information and possible own-ship decisions have to be considered.

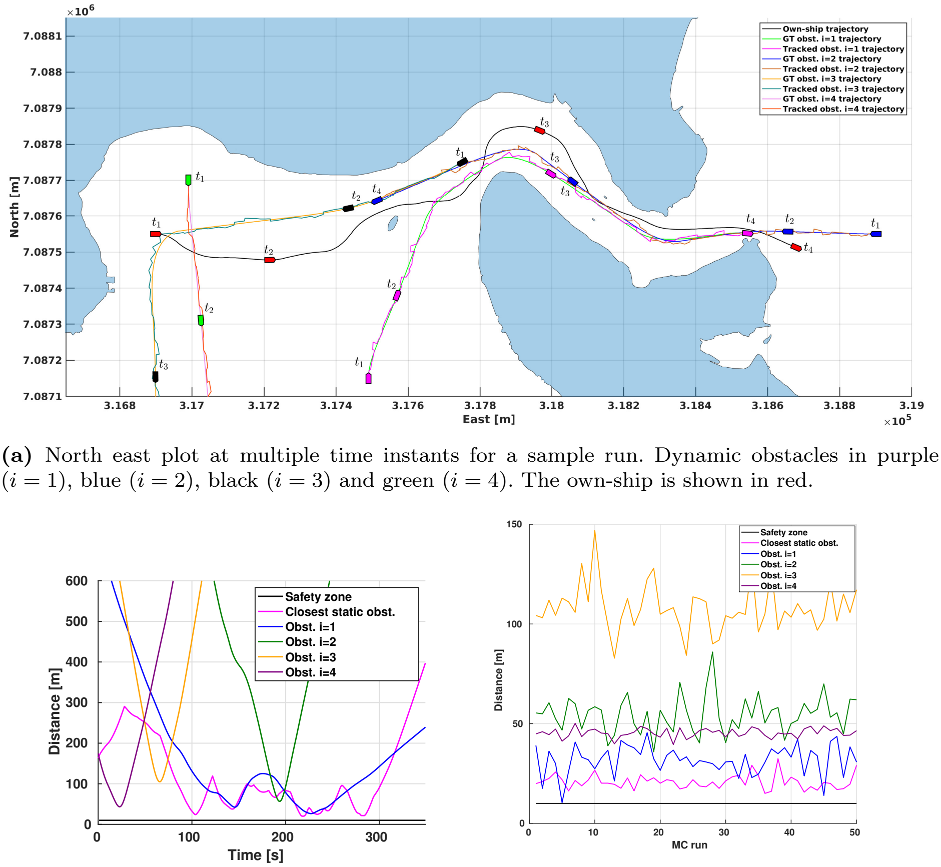
In simulation, the COLAV algorithm is shown to handle both grounding hazards

Table 6.3: Important PSB-MPC parameters for the Sakshaug situation.

Parameter	**Value**	Comment
ϵ	$2\,\mathrm{m}$	RDP distance threshold
T_{MPC}	$150\,\mathrm{s}$	Prediction horizon
T_s	$1.0\,\mathrm{s}$	Prediction time step
T_{coll}	$100\,\mathrm{s}$	Collision cost time discounting parameter
n_{ps}^{LOS}	5	Number of LOS prediction scenarios
r_{ct}	$20.0\,\mathrm{m}$	Prediction scenario spacing
d_{so}	$800.0\,\mathrm{m}$	Static obstacle consideration range
n_M	2	Number of sequential avoidance maneuvers
$U_{offsets,1}$	$\{1.0, 0.5, 0.0\}$	Speed offsets first maneuver
$U_{offsets,2}$	$\{1.0, 0.5\}$	Speed offsets second maneuver
$\chi_{offsets,1}$	$\{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$	Course offsets first maneuver
$\chi_{offsets,2}$	$\{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$	Course offsets second maneuver

and multiple dynamic obstacles in a safe manner, both in a narrow river environment, and also in a mix of more open sea and narrow waters. However, there is an inherent challenge in finding parameters that will make the algorithm work robustly and adhere to COLREGS for multiple types of situations, especially when the environmental constraints vary a lot.

Future work will involve making the PSB-MPC adaptive to the environment faced, and utilize historical data for tuning the algorithm. Also, the dynamic obstacle prediction and COLREGS penalization cost evaluation should be extended to consider static obstacles, for better applicability in confined spaces.

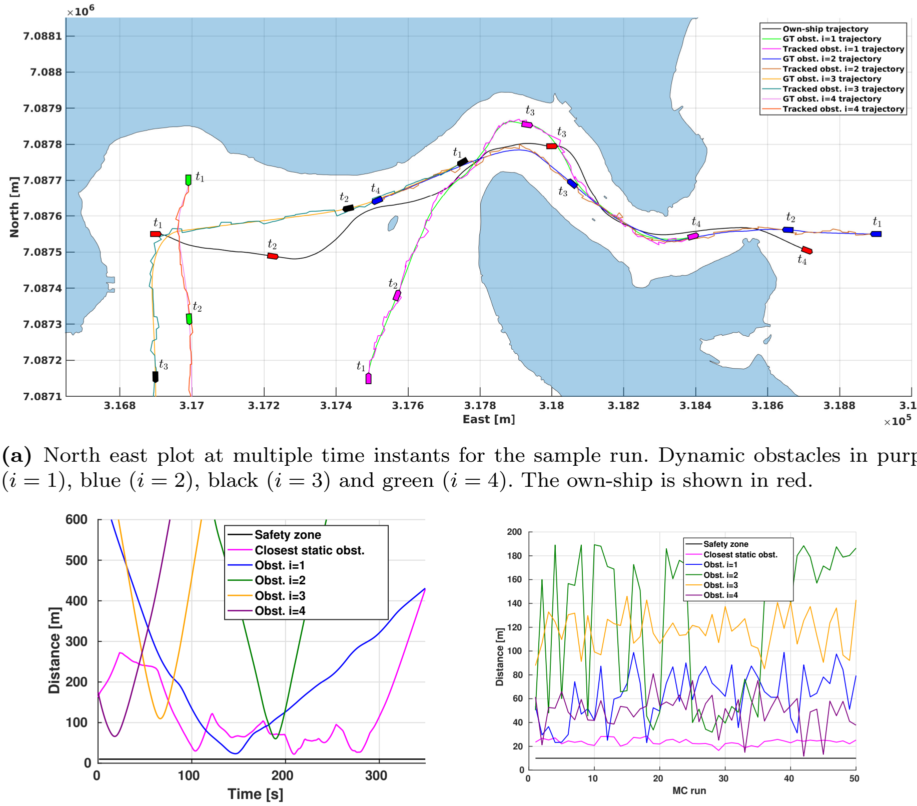


(a) North east plot at multiple time instants for a sample run. Dynamic obstacles in purple ($i = 1$), blue ($i = 2$), black ($i = 3$) and green ($i = 4$). The own-ship is shown in red.

(b) Distance to static and dynamic obstacles for the sample run.

(c) Statistics on the minimum distance to the obstacles over all N_{MC} simulation runs.

Figure 6.11: Results for the situation in Sakshaug with multiple obstacles in the first case.



(a) North east plot at multiple time instants for the sample run. Dynamic obstacles in purple ($i = 1$), blue ($i = 2$), black ($i = 3$) and green ($i = 4$). The own-ship is shown in red.

(b) Distance to static and dynamic obstacles for the sample run.

(c) Statistics on the minimum distance to the obstacles over all N_{MC} simulation runs.

Figure 6.12: Results for the situation in Sakshaug with multiple obstacles in the second case.

Chapter 7

Full-scale Experiments With an Obstacle Intention-Aware Probabilistic Scenario-based MPC

The chapter is based on the following publication

[100] **T. Tengesdal**, Sverre V. Rothmund, Erlend A. Basso, T. A. Johansen, and H. Schmidt-Didlaukies, "Obstacle Intention Awareness in Automatic Collision Avoidance: Full Scale Experiments in Confined Waters", *Field Robotics*, 2022. Submitted.

Here, a DBN for intention inference is combined with the latest version of the GPU-based PSB-MPC planner and verified in closed-loop full scale experiments using the Milliampere 2 ferry. Probabilistic information on obstacle intention states is used to make more informed decisions in the deliberate COLAV planner, such as the choice of disregarding one‘s stand-on role in a COLREGS situation where the other obstacle ship is inferred to be non-compliant. The work presented in this chapter combines all the major developments on intention inference and risk-based COLAV done throughout this thesis.

7.1 Introduction

7.1.1 Motivation

COLAV can be split into a two-part process. The first part is having adequate situational awareness. Situational awareness can be divided into three levels, perception, comprehension, and projection [160]. For COLAV at sea, perception represents data from sensors, such as radar, Lidar (light detection and ranging), or data from the Automatic Identification System (AIS), that lets the autonomous ship estimate the states of obstacles in the environment. Comprehension represents the ability to track the state of the obstacles based on the combined information from the different sensors. The last piece, projection, is the ability to predict the future motion of other ships based on tracking results. The work in this chapter focuses on the projection part of situational awareness. The projection is achieved by inferring the intentions of other ships based on past observations from the tracking system. The intentions are then used to evaluate the probability that other ships will follow different alternative maneuvering scenarios [96].

The second part of collision avoidance considers the decision-making based on the situational awareness. A challenge when developing decision-making algorithms for collision avoidance is the need to plan COLREGS-compliant and optimal trajectories while being able to react quickly when changes are observed. A hybrid approach for handling this challenge is to divide the COLAV planning system into multiple levels, as done in e.g. [55], [56], [85]. In a three-layer structure, as in [85], the highest level is a planner, which runs at low frequency and is responsible for finding a globally optimal trajectory to the ship's goal location while taking static obstacles into account. The mid-level planner is responsible for handling the collision avoidance and compliance with COLREGS in the local area, thus needing to take both static and dynamic obstacles into account. Lastly, the low-level planner is designed to operate at a high frequency in order to handle reactive collision avoidance when new situational information renders the trajectories planned by the higher levels unsafe. In this chapter, the focus is placed on the mid-level COLAV planning, which based on the understanding of the intentions of other ships, finds a collision-free trajectory that complies with the COLREGS regulations [97].

To fully enable both situational awareness and decision-making for collision avoidance, there is also a need for efficient computational platforms, which can both handle and take advantage of the increasing amounts of situational information made available today through modern sensor technology and AIS data. As certain situations can require the COLAV system to consider thousands of possible evasive own-ship maneuvering decisions, static obstacles, and dynamic obstacle intention scenarios with inherent uncertainty, an important part of the decision-making will be to process situational awareness information and possible decision candidates efficiently.

7.1.2 Previous Work

Maritime COLAV has been an active research field since the 1950s [26], and many algorithms have been proposed for solving this problem. This literature review will consider the subset of proposed methods that tackle static and dynamic obstacles, that explicitly consider uncertainties present in dynamic obstacle kinematics, and uncertainty in their intentions. For a general comprehensive literature review on maritime COLAV, please refer to [26], [27], [29], [30], [122].

The work in [68] developed an A-star search based trajectory planner for finding COLREGS-compliant and collision-free trajectories considering dynamic and static obstacles in a lattice. MC simulation with fuzzy logic and trajectory history data was used to find the set of most probable dynamic obstacle trajectories. From the set, the most probable trajectory is considered in the collision avoidance module. No prediction uncertainty was considered for the dynamic obstacles, and the computational efficiency of the planner was only tested in a specific setting.

A* search is applied to collision-free lattice-based trajectory planning [69], where trajectory deviation, collision risk, and non-compliance of COLREGS are penalized in the cost function. An intention-based motion model is used for dynamic obstacles, which uses historical data in order to classify a vessel as COLREGS-compliant or not, and which incorporates reactive COLAV. The details on this model are however not given.

A two-layered COLAV planning system with a global and local lattice-based trajectory planner was developed in 2017 [77]. Here, Voronoi Diagrams were used to generate a set of static obstacle collision-free waypoints, which again is used to generate a continuous path using Fermat‘s Spiral. Local re-planning windows are used for taking detected dynamic and static obstacles into account, where the dynamic obstacle motion is predicted using the CV model [78]. Using the predicted dynamic obstacle uncertainty up until time to CPA, a convex hull is created with the predicted positions and covariances from a KF, which is then regarded as an area to avoid in the planner. However, this can be overly conservative, as the CV model often has unrealistic uncertainty growth in real-world cases [79]. Furthermore, the run-time properties of the local planner were not studied.

RRTs was used in a sampling-based COLAV planning algorithm for COLREGS-compliant dynamic and static collision avoidance in [83]. The planner used a joint simulator for predicting both the own-ship and dynamic obstacle motion, with potential fields being used in the joint prediction to ensure collision-free trajectories. The planning algorithm was shown to be real-time feasible through multiple simulations. However, the joint simulator does assume that nearby ships will always perform deterministic COLREGS-compliant maneuvers if possible, which is not necessarily the case in practice.

Static and dynamic obstacle collision avoidance with consideration of environmental disturbances was proposed in [76] using NMPC. The dynamic obstacle motion was predicted using a deterministic CV model, which has limited quality in hazardous situations where ships will maneuver. How the MPC scaled with static and dynamic obstacles was not considered.

The work in [85] proposed a hierarchical COLAV system with three layers: A top-level trajectory planner responsible for generating collision-free trajectories with respect to static obstacles; a mid-level MPC-based COLAV planning algorithm for modifying this top-level trajectory in order to avoid collision with dynamic obstacles and adhere to COLREGS; and a reactive COLAV algorithm in the bottom layer used for planning emergency avoidance maneuvers in case the layers above do not plan a sufficiently safe trajectory.

Different types of VO based methods have been proposed for more reactive COLAV planning in [41], [61], [161]–[163], where the core idea is to compute a set of reachable velocities for the own-ship which do not cause collision with nearby dynamic obstacles. COLREGS adherence and taking dynamic obstacle kinematic uncertainties into account have been considered by specifying additional constraints when computing the VO, in probabilistic versions of the algorithm [61], [163]. These methods do however assume constant velocity for dynamic obstacles, unless their trajectories are known beforehand, and can be classified as reactive approaches since the sense-act methodology is used.

In [164], nonlinear VO (NL-VO) is used for the own-ship, together with a method for estimating the intent of vessels having a give-way role. The trajectories of dynamic obstacles with give-way roles are processed using the Douglas-Peucker algorithm, to find their action parameterized as turning points. The reachable velocity of the obstacle at these action points are then checked for intersection with the corresponding own-ship generated NL-VO set, and is used for determining whether the own-ship having stand-on role should perform emergency evasive maneuvers.

Fast probabilistic velocity obstacle (fPVO) for multi-ship COLAV is introduced in [162], where the own-ship roles with respect to all nearby vessels are calculated based on a symmetric own-ship - target ship COLREGS role classification method, and used to determine whether or not an evasive maneuver should be taken. The evasive maneuver taken minimizes a cost function that penalizes high collision risk, COLREGS role violation, and trajectory deviation. However, the study assumes that all involved vessels adhere to the COLREGS and try to avoid collision, which is not always valid in practice. The method is extended into a reciprocal (R-fPVO) version in [90], using a DBN for intent inference as in [165]. The Bayesian network estimates the probability of an obstacle ship being COLREGS-compliant and maps this into a rule-compliance factor which scales the COLREGS role violation cost. In contrast, the work presented in this chapter uses an intention inference method able to infer multiple intention states which in total models the behavior of meeting traffic.

The SB-MPC for maritime COLAV was first presented in [70] and was extended to the PSB-MPC in [97] to explicitly handle dynamic obstacle kinematic uncertainty through the estimation of collision probabilities associated with pairs of predicted own-ship and obstacle trajectories. The method was extended in [98] to incorporate intent information from e.g. DBNs in the form of a discrete set of intention probabilities which scaled the dynamic obstacle cost and COLREGS violation cost. The intention probabilities consider only if an obstacle was likely to keep its course, or take a starboard or port maneuver, which made it difficult to assess whether an intention was COLREGS-compliant or not. However, the MPC was shown able to take into account alternative obstacle trajectory scenarios, where most COLAV algorithms only assume the obstacle will keep its course and speed in the prediction. The MPC was further refined in [94] to instead consider intent information through using a set of estimated probabilities for a set of predicted obstacle trajectories or scenarios and implemented on a GPU with embedded anti-grounding in [99]. Prior to the present research, the PSB-MPC COLAV planning algorithm had not been tested with an intention inference module, only with usage of a priori intention information.

7.1.3 Contributions

Previous work on automatic maritime COLAV has most often neglected or simplified the situational awareness part of COLAV, assuming constant behavior in speed and course for nearby dynamic obstacle ships without uncertainty. This assumption of constant course and speed for vessels involved in encounters will not hold in practice, and limits the own-ship decision-making, possibly leading to more reactive avoidance maneuvers being made. Except for a few articles as in [68], [69], [77], [89], [90], the majority of studies in the literature have used this assumption without taking into account the dynamic obstacle kinematic uncertainty.

Thus, in this chapter, we propose a COLAV system with an intention inference module as in [96] for providing added situational information to the PSB-MPC planning algorithm as in [94], [99], and validate the system in experimental trials. All obstacle ships involved in the encounters broadcast their GNSS information for easier tracking system handling, which lets the present chapter focus on the projection part of situational awareness. This chapter combines the work on intent inference in [96] for evaluating the probability of a ship following different predicted trajectories, with the work in [94], [97], [99] for taking into account intention information under uncertainty in deliberate COLAV planning. The probability information, predicted trajectories, and other dynamic obstacle information is combined with static obstacle data from navigational charts and fed into the PSB-MPC COLAV planning algorithm. The PSB-MPC utilizes parallel processing to handle larger amounts of situational information and possible own-ship avoidance decisions than what would be possible on a sequential computation platform.

The resulting system has increased situational awareness in hazardous encounters and an improved ability to make decisions based on uncertain information on the kinematics and intents of nearby dynamic obstacles. The system is compliant with all the behavioral rules specified in COLREGS for power-driven vessels operating outside of traffic separation schemes, more specifically rules 8 and 13-17. Also, the system has better adherence to rule 7 on collision risk assessment as more situational information is considered.

The contributions of this chapter are thus as follows.

- Exploit intention models for obstacle ships in a risk-based COLAV for increased situational awareness.
- Validate if the intention-aware COLAV system with parallel processing capabilities has benefits through sea experiments.

7.1.4 Chapter Overview

The chapter is structured as follows. Section 7.2 gives an overview of the COLAV system architecture and software framework, in addition to the experimental platform used. Section 7.4 and 7.3 details the intention inference module and the PSB-MPC COLAV planning algorithm. Experimental results are given in Section 7.5 and conclusions are lastly given in Section 7.6.

7.2 System Architecture and Experimental Setup

7.2.1 COLAV System

An overview of the system architecture when using the PSB-MPC COLAV planning algorithm with a situational awareness system including intention inference is given in Fig. 7.1. Relevant system components are described below:

GHM Grounding Hazard Manager. Responsible for processing all data related to static obstacles, in a parameterization suitable for the PSB-MPC. A list of relevant hazards, typically inside a radius of $d_{so,relevant}$ around the current own-ship position, is sent to the PSB-MPC. Electronic Navigational Chart (ENC) data is used to get grounding hazard information.

DOM Dynamic Obstacle Manager. Responsible for processing all data related to dynamic obstacles, and for generating prediction scenarios. A prediction scenario here refers to an alternative maneuver or trajectory for the obstacle. The dynamic obstacle data includes tracking system information and estimated intention information for confirmed tracked obstacles. A list of data from relevant dynamic obstacles, inside a radius of $d_{do,relevant}$, is sent to the PSB-MPC. The dynamic obstacle data specifically includes, among others, the current time estimates and error covariances, predicted trajectories, and estimated probabilities for each alternative maneuvering scenario.

DOII Dynamic obstacle intention inference. Updates the intention model based on observed behavior. Receives dynamic obstacle data with predicted trajectories for each obstacle, and evaluates the probability that the obstacle will follow different candidate trajectories. See section 7.4.

PSB-MPC Finds the optimal own-ship trajectory based on static and dynamic obstacle data. See section 7.3.

7.2.2 The Own-ship Platform

The field experiments used the fully actuated milliAmpere 2 vessel as the own-ship platform, shown in Figure 7.2. The vessel is owned by NTNU and is used for research and technology development purposes in the field of autonomous maritime

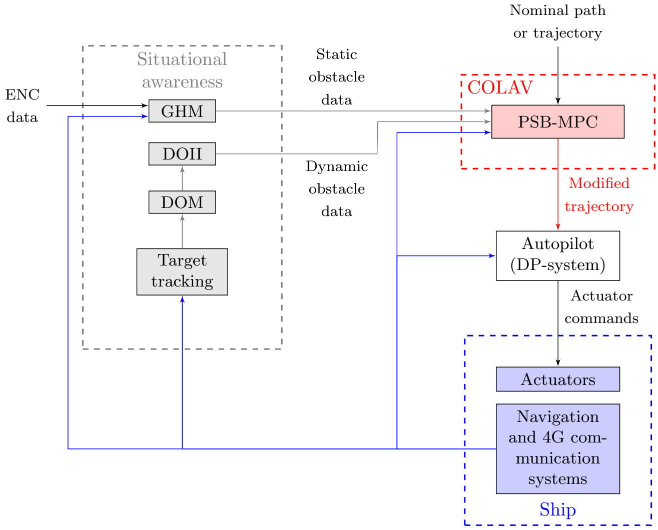


Figure 7.1: COLAV system overview.

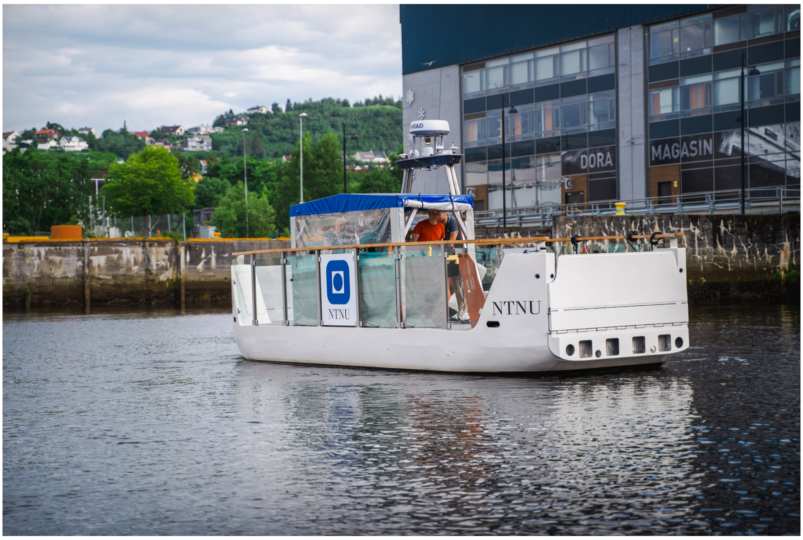


Figure 7.2: The Milliampere 2 ferry. Courtesy of Mikael Sætereid.

transport in urban areas. It is 8.6 m long, 3.5 m wide, fully electric, and equipped with four Fischer Panda 10 kW azimuth thrusters. To obtain accurate own-ship navigation data, the ferry uses a moving base Real-Time Kinematic (RTK) solution with two Global Navigation Satelite Systems (GNSS) receivers and two RTK antennas. At the time of the experiments, a Virtual Reference Station (VRS) was used with a Network Transport of RTCM data over IP (NTRIP) client for corrections, due to an RTK base station being down.

The Robotic Operating System (ROS) is used as middleware for the COLAV system, with software packages for each system component. The Milliampere 2 ferry uses a commercial DP system from Marine Technologies, which takes in trajectory references in planar pose, velocity, and acceleration. It is configured to perform small heading changes, mostly relying on speed changes, since back-and-forth ferry transportation is the goal. This created some challenges for trajectory tracking and COLREGS adherence in this work, as it is important to make apparent maneuvers to comply with COLREGS rule 8.

The PSB-MPC COLAV planning algorithm calculates the desired trajectory for the ferry to follow, and converts the output to the correct format of input for the DP system. The PSB-MPC and the situational awareness module runs on a separate computer, which is connected to an onboard Milliampere 2 computer via ethernet. The onboard computer is responsible for enabling autonomy on the ferry and connects to the navigation, sensor, and DP systems. The separate computer is a workstation running with Ubuntu 20.04.3 LTS as its operating system with an Intel(R) Core(TM) i9-10900K 3.70 GHz processor, 32 GB RAM, and an NVIDIA GeForce RTX 3090 GPU.

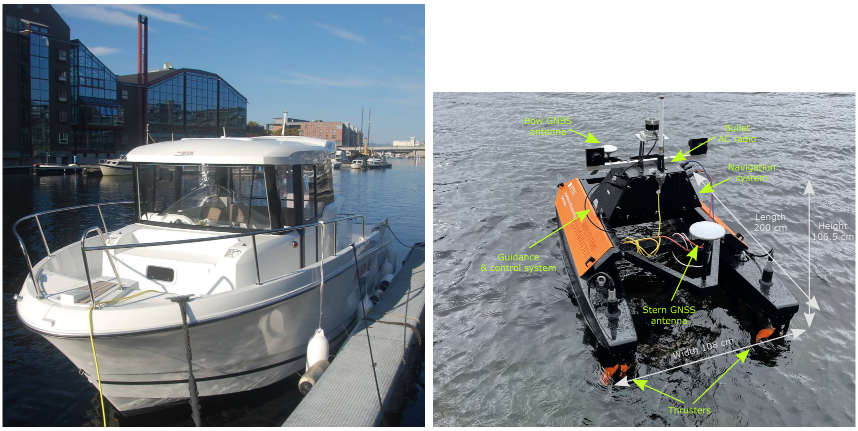
7.2.3 Target Tracking

As the main research objective is to showcase how an intention inference module can be used with a COLAV planning algorithm for safer and more efficient ship guidance, we use a simple communication setup where dynamic obstacles send GPS information to the own-ship tracking system. The tracking system then filters these measurements using a linear KF [125] to produce tracks for each vessel, using a CV model [95] for the estimation, as done in simulation in [97]. We then use 4G routers on each vessel to establish communication.

The target ships (dynamic obstacles) considered in the experiments is a Jeanneau Marlin 65 vessel called Havfruen depicted in Fig. 7.3(a), and the Cyberotter [1] depicted in Fig. 7.3(b). To get GNSS information from Havfruen, the vessel-driver runs a laptop with a Ublox ZED-F9P receiver that sends position measurements over ROS to the own-ship using ROS Ublox driver software. The Cyberotter has an onboard SBG GNSS system, which in the same fashion sends its GPS information over ROS. The own-ship is here configured as the ROS master, running on the workstation on milliAmpere 2, which is connected via ethernet and the 4G communication link to Havfruen and the Cyberotter.

Due to the Milliampere 2 ferry being restricted to operate within channels and harbor areas without waves, all experiments were performed in the easternmost basin of Nyhavna, Trondheim, Norway.

[1] More information at https://otter.itk.ntnu.no/doku.php?id=cyberotter.



(a) Havfruen. Courtesy of Mannhullet at NTNU.

(b) Cyberotter. Courtesy of Øystein Volden.

Figure 7.3: The target ships or dynamic obstacles Havfruen and Cyberotter used in the experiments.

7.3 The PSB-MPC COLAV Planning Algorithm

The PSB-MPC in this chapter is extended from the GPU-based version presented in the previous chapter, to more explicitly consider COLREGS rules 8 and 13-17 with usage of intention information, and with an update to the trajectory deviation cost.

As stated in previous chapters, the objective of the PSB-MPC is to solve the following optimization problem

$$l^*(t_0) = \arg\min_{l} \mathcal{H}^l(t_0) \tag{7.1}$$

at the current time t_0, to find the optimal control behavior l which minimizes $\mathcal{H}^l(t_0)$. The cost function penalizes dynamic obstacle collision risk cost $\mathcal{H}^l_{do}$, breaching the COLREGS with respect to all dynamic obstacles $\mathcal{H}^l_{colregs}$, grounding on static obstacles $\mathcal{H}^l_{so}$, in addition to penalizing deviations from the nominal trajectory $\mathcal{H}^l_p$, respectively. The solution gives us an optimal avoidance trajectory for the own-ship, which is regarded as a reference trajectory to be tracked by the DP-system. A sampling time of 5 seconds is used for the MPC. Algorithm 4 described in the previous chapter is used to evaluate the cost of all control behaviors, and to find the optimal solution.

The following sub-sections go through the prediction models used in the MPC and the different elements in the cost function, and how the PSB-MPC is used with the Milliampere 2 ferry.

7.3.1 Prediction Models

Similarly to the previous chapter, a discrete kinematic model as outlined in Section 2.4.3 was used in the PSB-MPC during the experiments:

$$\begin{aligned} x_{k+1} &= x_k + U_k cos(\chi_k) \\ y_{k+1} &= y_k + U_k sin(\chi_k) \\ \chi_{k+1} &= \chi_k + \frac{1}{T_\chi}(\chi_{d,k} - \chi_k) \\ U_{k+1} &= U_k + \frac{1}{T_U}(U_{d,k} - U_k) \end{aligned} \tag{7.2}$$

which describes the own-ship state $\boldsymbol{x}_k = [x_k, y_k, \chi_k, U_k]^T$ motion during the MPC prediction, and where LOS guidance is used to predict all candidate trajectories, as described in Section 2.5.1. This is also the case for dynamic obstacles, where the same prediction scheme as outlined in Section 6.2.1 was used, with a kinematic model and LOS guidance for generating alternative obstacle maneuvering scenarios.

Considering a prediction horizon of length T_{mpc} with a time step Δ_{mpc}, Euler‘s method for numerical integration is used to generate predicted own-ship and dynamic obstacle trajectories with samples at the discrete predicted times $t_k \in D(t_0) = \{t_0, t_0 + \Delta_{mpc}, ..., t_0 + k\Delta_{mpc}, ..., T_{mpc}\}$.

7.3.2 Cost Function

As in the previous chapter, the PSB-MPC cost function for a specific control behavior l can be decomposed as

$$\mathcal{H}^l(t_0) = \mathcal{H}^l_{do} + \mathcal{H}^l_{colregs} + \mathcal{H}^l_{so} + \mathcal{H}^l_{p}, \tag{7.3}$$

where the penalization terms are associated with dynamic obstacles, COLREGS violation, static obstacles or grounding hazards and trajectory tracking, respectively. The following text will go through their definitions, where the COLREGS cost and trajectory deviation cost have been subjects to modification in this chapter.

Dynamic Obstacle Cost

The dynamic obstacle cost captures the probabilistic collision risk and is here chosen as a sum over all nearby obstacles

$$\mathcal{H}^l_{do} = \sum_{i=1}^{n_{do}} \mathcal{H}^{l,i}_{do} \tag{7.4}$$

where the individual dynamic obstacle cost is given by a weighted sum over the n^i_{ps} prediction scenarios

$$\mathcal{H}^{l,i}_{do} = \sum_{s=1}^{n^i_{ps}} \hat{\mathbb{P}}^i_s \mathcal{H}^{l,i,s}_{do} \tag{7.5}$$

with the prediction scenario probabilities $\{\hat{\mathbb{P}}^i_s\}_{s=1}^{n^i_{ps}}$ as weights, estimated from the dynamic obstacle intention inference module. The cost associated with the pair of an own-ship control behavior l and a dynamic obstacle i behaving as in prediction scenario s is, as in the previous chapter, calculated as

$$\mathcal{H}^{l,i,s}_{do} = \max_k \zeta_i \mathcal{C}^{l,s}_{i,k} \hat{\mathbb{P}}^{l,i,s}_{c,k} exp(-t_k/T_{coll}) \tag{7.6}$$

The CE method is used to estimate the collision probabilities $\hat{\mathbb{P}}^{l,i,s}_{c,k}$ as in Chapter 5, and multiplied by the collision consequence function

$$\mathcal{C}^{l,s}_{i,k} = K_{coll}||\boldsymbol{v}^{i,s}_k - \boldsymbol{v}^l_k||^2 \tag{7.7}$$

to obtain the collision risk, where K_{coll} is again the collision cost penalty parameter, $\boldsymbol{v}^{i,s}_k$ is the predicted obstacle velocity in prediction scenario s and $\boldsymbol{v}^l_k$ the predicted own-ship velocity, both at the predicted time t_k.

By specifying a sufficiently rich set of obstacle prediction scenarios, which in total spans the maneuvering possibilities of a dynamic obstacle, the scenario probabilities $\hat{\mathbb{P}}^i_s$ can capture its intention uncertainty and thus give the PSB-MPC elevated situational awareness.

COLREGS Violation Cost

The COLREGS violation cost has in this chapter been updated to consider the entire candidate trajectory of the own-ship when evaluating whether or not the rules were breached and is meant to make adherence to the COLREGS rules $8, 13, 14, 15, 16$ and 17 easier. The total COLREGS cost, $\mathcal{H}^l_{colregs}$ is evaluated by summing the COLREGS violation costs over all dynamic obstacles

$$\mathcal{H}^l_{colregs} = \sum_{i=1}^{n_{do}} \mathcal{H}^{l,i}_{colregs} \tag{7.8}$$

which enables multi-ship COLREGS adherence. The cost towards a specific dynamic obstacle i for a control behavior l is evaluated by summing the violation costs for all different prediction scenarios s weighted by the prediction scenario probabilities as

$$\mathcal{H}^{l,i}_{colregs} = \sum_{s=1}^{n^i_{ps}} \hat{\mathbb{P}}^i_s \mathcal{H}^{l,i,s}_{colregs} \tag{7.9}$$

where the scenario-specific cost is

$$\mathcal{H}^{l,i,s}_{colregs} = \kappa_{SO}\mu^{l,i}_{SO}\hat{\mathbb{P}}^i_{WGW} + \kappa_{GW}\mu^{l,i}_{GW}\hat{\mathbb{P}}^i_{CCEM} + \kappa_{RA}\mu^l_{RA} \tag{7.10}$$

and $\mu^{l,i,s}_{SO}$ and $\mu^{l,i,s}_{GW}$ are binary indicators of whether or not the own-ship following control behavior l violated its stand-on or give-way role with respect to dynamic obstacle i behaving as in scenario s. The binary indicator μ^l_{RA} is equal to 1 if the control behavior l induces an initial avoidance maneuver that does not lead to a readily apparent action. The cost (7.10) considers the entire own-ship and dynamic obstacle trajectories in the violation evaluation, which is different from the original SB-MPC where only instantaneous states were considered. Distinct penalty parameters κ_{SO}, κ_{GW} and κ_{RA} are used to weight the stand-on, give-way, and readily apparent violation costs separately. The stand-on violation cost is weighted by the estimated probability $\hat{\mathbb{P}}^i_{WGW}$ that the obstacle will fulfill its give-way obligations when specified by the COLREGS, obtained from the intention inference module. The give-way violation cost is similarly weighted by the estimated probability $\hat{\mathbb{P}}^i_{CCEM}$ that the obstacle will perform a COLREGS-compliant evasive maneuver when specified by COLREGS, also obtained from the intention inference module. See Section 7.4 for more information.

The COLREGS situation CS is defined when the distance d_{0i} from the own-ship to obstacle ship i at the current time is less than a given parameter $d_{colregs}$. This ensures that subsequent changes in course or bearing do not change the situation, as stated in COLREGS rule 13(d). The COLREGS situation is determined as follows

- If the bearing from the own-ship to the obstacle is more than 22.5° abaft the beam of the own-ship, then the own-ship is being overtaken and $CS = OT\text{-}en$ (COLREGS rule 13(b)).
- If the bearing from the obstacle to the own-ship is more than 22.5° abaft the obstacle beam then the own-ship is overtaking and $CS = OT\text{-}ing$ (COLREGS rule 13(b)).
- If the relative heading between the ships is within $180° \pm 10°$, then there is a head-on situation and $CS = HO$ (COLREGS rule 14).
- If there is no head-on or overtaking situation then there is a crossing situation (COLREGS rule 15), and either a port side or starboard side crossing based on the following:
  - If the bearing to the obstacle ship relative to the own-ship heading is negative, the obstacle ship is on the port side and $CS = CR\text{-}PS$.
  - If the bearing to the obstacle ship relative to the own-ship heading is positive, the obstacle ship is on the starboard side, and $CS = CR\text{-}SS$.

For a control behavior l, no penalty is given if the time until closest point of approach (CPA) with respect to obstacle i, $t^{l,i}_{cpa}$, is longer than a time threshold $T_{start} =$ s. Also, no penalty is given if the ships will pass at a distance at CPA, $d^{l,i}_{cpa}$, that is larger than a distance threshold $d_{colregs}$, i.e. $d^i_{cpa} > d_{colregs}$. Note that all distance calculations take the extent of the own-ship and an obstacle into account.

A control behavior l gets a stand-on violation penalty if the own-ship has a stand-on role and the control behavior leads to a change in course (CIC^l) of more than $CIC^{SO}_{max} = 15°$, or change in speed (CIS^l) of more than $CIS^{SO}_{max} = 0.5\,\text{m/s}$, as per COLREGS rule 17(a). No stand-on penalty is given if the current distance d_{0i} to the obstacle is below a critical value $d_{critical} = 15\,\text{m}$, as all ships should then act to avoid collision, as stated in COLREGS rule 17 (a)(ii) and (b). The stand-on penalty indicator is thus given as

$$\begin{aligned}\mu_{SO}^{l,i} =& t_{cpa}^{l,i} < T_{start} \land d_{cpa}^{l,i} < d_{colregs} \land d_{0i} > d_{critical} \quad \land \\ & (CS == OT\text{-}en \lor CS == CR\text{-}PS) \land (CIC^{l} \neq none \lor CIS^{l} \neq none)\end{aligned} \tag{7.11}$$

A control behavior l gets a give-way violation penalty if it does not act as specified in COLREGS rules 14, 15, and 17. COLREGS rule 14 specifies that ships in a head-on situation should pass port-to-port ($P2P$). According to COLREGS rule 15 a ship in a crossing situation with the other on its starboard side $CR\text{-}SS$, should cross aft of that ship (CA). If a ship is forced to take action in a crossing situation with the other on its port side, $CR\text{-}PS$, then it should avoid changing the course to port (COLREGS rule 17(c)).

$$\begin{aligned}\mu_{GW}^{l,i} =& t_{cpa}^{l,i} < T_{start} \land d_{cpa}^{l,i} < d_{colregs} \land (CS == HO \land \neg P2P^{l} \quad \lor \\ & CS == CR\text{-}SS \land \neg CA^{l} \lor CS == CR\text{-}PS \land CIC^{l} == port)\end{aligned} \tag{7.12}$$

Lastly, a control behavior l gets a readily apparent violation penalty if it does not adhere to COLREGS rule 8, which will be the case if the behavior induces any non-zero initial speed or course modification that is not sufficiently high to make the resulting avoidance maneuver apparent. Here, any non-zero course modification which is less than $CIC_{min}^{GW} = 45°$ or any non-zero speed modification less than $CIS_{min}^{GW} = 0.5\,\text{m/s}$ gains a violation:

$$\mu_{RA}^{l,i} = t_{cpa}^{l,i} < T_{start} \land d_{cpa}^{l,i} < d_{colregs} \quad \land \tag{7.13}$$

$$((|\chi_{m,1}^{l}| > 0 \land |\chi_{m,1}^{l}| < CIC_{min}^{GW}) \lor (U_{m,1}^{l} > 0 \land U_{m,1}^{l} < CIS_{min}^{GW})) \tag{7.14}$$

The large course change threshold was chosen due to the Milliampere 2 ferry not being tuned to make large heading changes when tracking trajectories.

Static Obstacle Cost

Static obstacles or grounding hazards are parameterized as polygons and read in from map data from the considered area to navigate in, as described in Section 6.2.2 in the previous chapter. The cost of grounding on static obstacles is given as

$$\mathcal{H}_{so}^{l} = \max_{j} \mathcal{H}_{so}^{l,j} \tag{7.15}$$

where

$$\mathcal{H}_{so}^{l,j} = \max_{k}(G_1 + G_2 \phi_{j,k}^{l} V_w^2) exp\left(-(G_3|d_{0j,k}^{l} - d_{safe}| + G_4 t_k)\right) \tag{7.16}$$

Here, V_w is the estimated wind speed and $\phi_{j,k} = \max(0, \boldsymbol{\omega}^T \boldsymbol{L}_{0j,k}^{l})$ with $\boldsymbol{\omega}$ denoted as the unit wind direction and $\boldsymbol{\omega}^T$ being its transpose vector. The unit vector $\boldsymbol{L}_{0j,k}^{l}$

points from the own-ship to the static obstacle j, with $d^l_{0j,k}$ as the corresponding distance. The parameter d_{safe} is the own-ship safety zone radius, and G_1 to G_4 are tuning parameters for the grounding cost. If no wind information is available, the wind speed is set as $V_w = 0$, which was the case in these experiments. However, the grounding cost in general forces the MPC to take action if the own-ship is drifting towards land due to non-zero wind speed.

Trajectory Deviation Cost

A change in the trajectory deviation cost from previous chapters is made to take into account large cross-track errors from the nominal trajectory. The cost of deviating from the nominal trajectory is parameterized as

$$\begin{aligned}\mathcal{H}^l_p = \max_k K_e|e^l_k| + \frac{1}{n_M}\sum_{M=1}^{n_M} \big[K_U|1 - U^l_{m,M}|^2 + K_{\Delta U}|U^l_{m,M} - U^l_{m,M-1}|+ \\ K_\chi(\chi^l_{m,M})^2 + K_{\Delta\chi}|\chi^l_{m,M} - \chi^l_{m,M-1}|^2\big]\end{aligned} \tag{7.17}$$

where e^l_k is the cross-track error at predicted time t_k when following control behavior l, penalized with parameter K_e. The parameters K_U and $K_{\Delta U}$ determine the severity of the penalties of speed change from the nominal reference, speed change from one maneuver to the next, and similarly for K_χ and $K_{\Delta\chi}$, where course modifications are penalized.

7.3.3 Creating Autopilot References

The PSB-MPC outputs a desired trajectory which consist of N_s predicted desired own-ship state samples $\boldsymbol{x}_{d,k} = [x_{d,k}, y_{d,k}, \chi_{d,k}, U_{d,k}]^T, \quad k = 1, 2, ..., N_s$. The inputs to the Milliampere 2 DP-system consist of planar pose, velocity, and acceleration references. These are obtained from the PSB-MPC reference trajectory by constructing a continuously differentiable spline using methods as in [166], where a geometric path is composed using $x_{d,k}$, $y_{d,k}$, $\chi_{d,k}$ and with a speed profile using $U_{d,k}$. Every time the PSB-MPC outputs a new reference trajectory, interpolation is used to generate a new spline that uses the references at the current time from the old reference trajectory as boundary conditions, which ensures continuity of the velocity references.

7.4 Dynamic Obstacle Intention Inference

The Dynamic Obstacle Intention Inference (DOII) module gets a list of dynamic obstacle estimates and prediction scenarios from the DOM. The goal of the DOII is then to evaluate the probabilities that the dynamic obstacle will follow the different scenarios ($\hat{\mathbb{P}}^i_s$). Additionally, the module gives the probability that the obstacle will fulfill its give-way obligations ($\hat{\mathbb{P}}^i_{WGW}$) and the probability that it will perform a COLREGS-compliant evasive maneuver ($\hat{\mathbb{P}}^i_{CCEM}$) when supposed to.

The relevant probabilities are evaluated using a DBN. This model is a modified version of the one presented in [96], which is included at the end of this thesis.

Compared to [96] this model removes nodes related to whether or not the situation has started and whether or not there is a risk of collision. As the experiments in the present chapter will be done in confined environments, there is little to no uncertainty in whether there is a risk of collision and when the situation starts. Additionally, some small alterations were made to improve the model based on the experience gathered from extensively testing the method.

The DBN takes the perspective of one of the dynamic obstacles in an encounter. This dynamic obstacle is referred to as the reference ship in this section. Multiple ships are modeled by running an instance of the model for each dynamic obstacle in the encounter.

The DBN contains different intention nodes that define how the ship will act in different situations. The different intention nodes are given in Table 7.2. These nodes are stochastic and represent the own-ship's belief regarding the intentions of the reference ship. The distributions of the intention nodes are updated based on observed behavior and are used to evaluate the probability of the different obstacle prediction scenarios.

The DBN has an output node that represents whether or not different measured properties are compatible with the state of the intention nodes. The measured properties are given in Table 7.3 and are evaluated assuming all obstacle ships keep their current course and speed. The intention nodes are updated by inserting evidence based on an observation in a new time-step of the DBN together with evidence stating that the output node is in the state "true".

Whether or not the measured properties are compatible with the intentions is based on the behavioral rules specified in COLREGS rules 8, and 13 to 18. Section 7.4.1 transcribes these rules as logic statements, which are used to build the conditional probability tables used by the DBN. The topography of the resulting network is shown in Fig. 7.4.

The probabilities that the reference ship will follow the different prediction scenarios are evaluated in a similar manner. The different measured properties are first evaluated based on the specific prediction scenario. These are then inserted on a new temporary time-step in the DBN, which is removed after the evaluation. The probability that the output node is in the state "true" gives the probability that the scenario is compatible with the obstacle intentions ($Pr(C[t])$). As a large number of prediction scenarios can be compatible with the intentions, the resulting distribution over the different scenarios must be normalized such that it sums to 1.

$$\hat{\mathbb{P}}_s^i = \eta \Pr\{C[t]\}(1 + \cos(\delta_{s0})) \tag{7.18}$$

A prior of $1 + \cos(\delta_{s0})$, stating that the ship is more likely to keep its current course than to change it, is used in the probability evaluation. The variable η is a normalization factor, and δ_{s0} the difference between the course held 10 s into a scenario s and the current course of the reference ship. The time interval of 10 s is

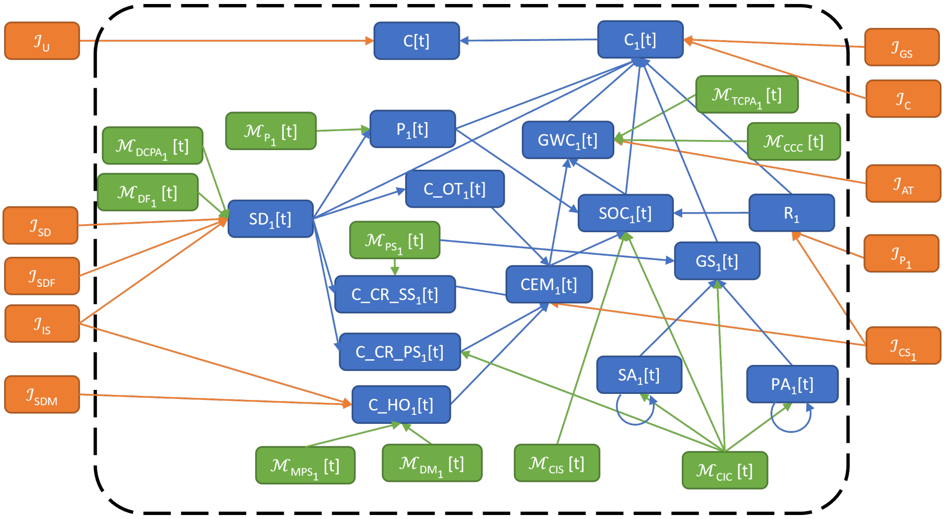


Figure 7.4: Topography of the DBN used for intention modeling. The figure shows the case of an encounter with two ships. Everything inside the dotted line is time-varying and repeats for each time step in the DBN. The intention nodes, shown in orange, are time-invariant. The abbreviations are explained in tables 7.2, 7.3, 7.4.

Table 7.1: Abbreviations

Abbreviation	Description
SO	Stand-on
GW	Give-way
HO	Head-on
OT-ing	Overtaking
OT-en	Overtaken
CR-SS	Crossing with other ship on starboard side
CR-PS	Crossing with other ship on port side

chosen in our experiments as it gives enough time for the reference ship to change its course.

7.4.1 Course applicability logic

The measured properties are compatible with the intentions of the ship (C) if they are compatible towards all ships in the encounter (C_ρ) or the ship behaved in an unmodelled fashion $(\mathcal{I}_U)$. $\mathcal{I}_U$ works as a catch-all for all unmodelled behavior, indicating that we have no knowledge of how the reference ship will act in the future. Subscript ρ represents the ID of another ship in the encounter. P represents the set of all ships in the encounter other than the reference ship.

$$C[t] = (\forall_{\rho \in P} C_\rho[t]) \vee \mathcal{I}_U \tag{7.19}$$

Table 7.2: Intention variables. $\mathcal{N}(\mu, \sigma)_{[a,b]}$ indicates a normal distribution with expected value μ, standard deviation σ, truncated to be between a and b, and discretized into 30 evenly spaced intervals. The probability of “true" is given for binary states.

Symbol	Description	States	Prior
$\mathcal{I}_{AT}$	What time until CPA the reference ship considers as ample time to act	real-valued	$\mathcal{N}(60\,\mathrm{s}, 4\,\mathrm{s})_{[0,100]}$
$\mathcal{I}_{C}$	Whether the reference ship intends to follow COLREGS	binary	0.98
$\mathcal{I}_{CS_\rho}$	What COLREGS situation the reference ship thinks it has towards ship ρ	$\{OT\text{-}ing, OT\text{-}ing, HO, CR\text{-}PS, CR\text{-}SS\}$	See section 7.4.3
$\mathcal{I}_{GS}$	Whether the reference ship intends to act according to good seamanship	binary	0.99
$\mathcal{I}_{P_\rho}$	Relative priority the reference ship has to ship ρ	$\{higher, similar, lower\}$	[0.05, 0.90, 0.05]
$\mathcal{I}_{SD}$	What the reference ship considers a safe distance at CPA	real-valued	$\mathcal{N}(25\,\mathrm{m}, 2.5\,\mathrm{m})_{[0,30]}$
$\mathcal{I}_{SDF}$	How far in front of ship ρ the reference ship considers a crossing as safe	real-valued	$\mathcal{N}(20\,\mathrm{m}, 4\,\mathrm{m})_{[0,50]}$
$\mathcal{I}_{SDM}$	What the reference ship considers a safe distance at CPA to the current midpoint between the reference ship and ship ρ. (Relevant for HO)	real-valued	$\mathcal{N}(15\,\mathrm{m}, 2.5\,\mathrm{m})_{[0,30]}$
$\mathcal{I}_{U}$	Whether the reference ship acts in an unmodelled way	binary	0.00001

Table 7.3: Measurement variables

Symbol	Description	States
$\mathcal{M}_{DCPA_\rho}[t]$	Distance between reference ship and ship ρ at CPA	real-valued
$\mathcal{M}_{DF_\rho}[t]$	How far the reference ship crosses in front on ship ρ. This value is set to ∞ if the ship does not cross in front of ship ρ	real-valued
$\mathcal{M}_{DM_\rho}[t]$	The reference ship's distance at CPA to the current midpoint between the reference ship and ship ρ	real-valued
$\mathcal{M}_{P_\rho}[t]$	Whether the reference ship has passed ship ρ	binary
$\mathcal{M}_{TCPA_\rho}[t]$	Time until CPA between reference ship and ship ρ	real valued
$\mathcal{M}_{PS_\rho}[t]$	Whether the reference ship will pass with ship ρ on its port or starboard side	$\{starboard, port\}$
$\mathcal{M}_{MPS_\rho}[t]$	Whether the reference ship will pass the current midpoint between itself and ship ρ on its port or starboard side	$\{starboard, port\}$
$\mathcal{M}_{CIC}[t]$	Whether the reference ship has changed course more than X since the start of the encounter	$\{starboard, straight, port\}$
$\mathcal{M}_{CIS}[t]$	Whether the reference ship has changed speed more than Y since the start of the encounter	$\{higher, none, lower\}$
$\mathcal{M}_{CCC}[t]$	Whether the reference ship is currently changing course or speed	binary

The measured properties are compatible with the intentions of the reference ship towards ship ρ (C_ρ) if one of the listed conditions is met

- If the ships have passed each other safely (P_ρ).
- If the role (R_ρ) of the reference ship is give-way and it gives way correctly (GWC_ρ), while adhering to god seamanship (GS_ρ) or if it does not intend to act with good seamanship $(\mathcal{I}_{GS})$.
- If the role (R_ρ) of the reference ship is stand-on and stands on correctly (SOC).
- If the reference ship does not intend to comply with the COLREGS regulations $(\mathcal{I}_{CC})$ and passes at what it considers to be a safe distance (SD_ρ), while adhering to god seamanship (GS_ρ) or if it does not intend to act with good seamanship $(\mathcal{I}_{GS})$

$$\begin{aligned} C_\rho[t] = & P_\rho[t] \lor (R_\rho == SO \land SOC_\rho[t]) \quad \lor \\ & \Big(\big((R_\rho == GW \land GWC_\rho[t]) \lor (\neg\mathcal{I}_{CC} \land SD_\rho) \big) \land (GS_\rho \lor \neg\mathcal{I}_{GS}) \Big) \end{aligned} \tag{7.20}$$

Table 7.4: Model variables

Symbol	Description	States
$C[t]$	Observation compatible with the intentions of the reference ship	binary
$C_\rho[t]$	Observations and intentions compatible towards ship ρ	binary
$CEM_\rho[t]$	Correct evasive maneuver towards ship ρ	binary
$C\_CR\_SS_\rho[t]$	Correct crossing evasive maneuver with ship ρ on the starboard side	binary
$C\_CR\_PS_\rho[t]$	Correct crossing evasive maneuver with ship ρ on the port side	binary
$C\_HO_\rho[t]$	Correct head-on evasive maneuver towards ship ρ	binary
$C\_OT_\rho[t]$	Correct overtaking evasive maneuver towards ship ρ	binary
$GS_\rho[t]$	Good seamanship towards ship ρ	binary
$GWC_\rho[t]$	Gives way correctly towards ship ρ	binary
$P_\rho[t]$	Safely passed ship ρ	binary
R_ρ	Role towards ship ρ	$\{GW, SO\}$
$PA_\rho[t]$	There has been a port action towards ship ρ	binary
R_ρ	Role towards ship ρ	$\{GW, SO\}$
$SD_\rho[t]$	The reference ship will cross at a safe distance towards ship ρ	binary
$SA_\rho[t]$	There has been a starboard action towards ship ρ	binary

The reference ship stands on correctly towards ship ρ (SOC_ρ) if it does not change its course ($\mathcal{M}_{CIC}$) or speed ($\mathcal{M}_{CIS}$). A change in course or speed is accepted if it is performing a correct evasive maneuver (CEM) towards another ship in the encounter (λ) that it has a give-way role (R) and has not already passed (P) (COLREGS rule 17). The ship is considered as keeping its course if the difference between the current course and the initial course in the encounter is less than 10°. As none of the dynamic obstacles were to intentionally change their speed, the change in speed measurement was overridden to always be in the state *none*. This simplified the experiments as it was difficult to get the ships up to speed before the situation started in the confined area.

$$\begin{aligned} SOC_\rho[t] = &(\mathcal{M}_{CIC}[t] == \mathit{straight} \land \mathcal{M}_{CIS}[t] == \mathit{none}) \\ &\lor \left(\exists_{\lambda \in P \setminus \{\rho\}} R_\lambda == GW \land CEM_\lambda[t] \land \neg P_\lambda[t]\right) \end{aligned} \tag{7.21}$$

The reference ship gives way correctly towards ship ρ (GWC_ρ) if one of the listed conditions is met

- If the reference ship is executing a correct evasive maneuver (CEM_ρ).

- If the time until CPA ($\mathcal{M}_{TCPA_\rho}$) is longer than what the reference ship considers "ample time", ($\mathcal{I}_{AT}$) (COLREGS rule 8).
- If the reference ship is currently in the middle of a course or speed change $\mathcal{M}_{CCC}$.

Thus, before ample time the ship is allowed to stand on correctly (SOC_ρ). In the middle of a course change, the reference ship may have changed course enough to not stand on correctly while it has not changed enough to pass the other ship at a safe distance.

$$\begin{aligned} GWC_\rho[t] =& CEM_\rho[t] \vee \mathcal{M}_{CCC} \\ & \vee \big(\mathcal{M}_{TCPA_\rho}[t] < \mathcal{I}_{AT} \wedge SOC_\rho[t]\big) \end{aligned} \tag{7.22}$$

COLREGS rules 13-17 define different encounter situations and how to act in them. $\mathcal{I}_{CS_\rho}$ represents which situation the reference ship thinks it is. An evasive maneuver is correct (CEM_ρ) if it complies with the rules.

$$\begin{aligned} CEM_\rho[t] =& (\mathcal{I}_{CS_\rho} == HO \wedge C\_HO_\rho[t]) \\ & \vee (\mathcal{I}_{CS_\rho} == OT\text{-}ing \wedge C\_OT_\rho[t]) \\ & \vee (\mathcal{I}_{CS_\rho} == OT\text{-}en \wedge C\_OT_\rho[t]) \\ & \vee (\mathcal{I}_{CS_\rho} == CR\text{-}SS \wedge C\_CR\_SS_\rho[t]) \\ & \vee (\mathcal{I}_{CS_\rho} == CR\text{-}PS \wedge C\_CR\_PS_\rho[t]) \end{aligned} \tag{7.23}$$

A correct overtaking evasive maneuver (C_OT) is performed if the ships pass at a safe distance (SD_ρ) (COLREGS rule 13).

$$C\_OT_\rho[t] = SD_\rho[t] \tag{7.24}$$

A correct head-on evasive maneuver (C_HO_ρ) is performed if the ships pass each other at a safe distance on the port side (CORLEGS rule 14). To ensure that both ships must give way, the distance to the current midpoint between the ships ($\mathcal{M}_{DM_\rho}$) is considered.

$$C\_HO_\rho[t] = \big(\mathcal{M}_{MPS_\rho}[t] == port\big) \wedge (\mathcal{M}_{DM_\rho}[t] > \mathcal{I}_{SDM}) \tag{7.25}$$

A correct crossing starboard-side evasive maneuver ($C\_CR\_SS_\rho[t]$) is performed if the reference ship passes at a safe distance (SD_ρ) aft of ship ρ (COLREGS rule 15). If the reference ship is passing aft of ship ρ in a starboard-side crossing situation, then it will pass with ship ρ on its port side.

$$C\_CR\_SS_\rho[t] = (\mathcal{M}_{PS_\rho} == port) \wedge SD_\rho[t] \tag{7.26}$$

A correct crossing port-side evasive maneuver ($C\_CR\_PS_{\rho}[t]$) is performed if the reference ship passes at a safe distance and does not change the course ($\mathcal{M}_{CIC}$) to port (COLREGS rule 17(c)).

$$C\_CR\_PS_{\rho} = (\mathcal{M}_{CIC} \neq \mathit{port}) \wedge SD_{\rho}[t] \tag{7.27}$$

COLREGS rules 13 to 15 specifies which ship that has a give-way and stand-on role (R_{ρ}) in an encounter. Additionally, a ship may stand on if it has higher priority ($\mathcal{I}_{P_{\rho}}$), either as specified in COLREGS rule 18 or due to unwritten rules [22]. The other ship will then have lower priority and must give way.

$$R_{\rho} = \begin{cases} GW & \begin{aligned} &\mathit{if}\, \mathcal{I}_{P_{\rho}} == \mathit{lower} \\ &\quad \vee \big(\mathcal{I}_{P_{\rho}} == \mathit{similar} \\ &\quad \wedge (\mathcal{I}_{CS_{\rho}} == \mathit{HO} \\ &\quad \vee \mathcal{I}_{CS_{\rho}} == \mathit{CR\text{-}SS} \\ &\quad \vee \mathcal{I}_{CS_{\rho}} == \mathit{OT\text{-}ing})\big) \end{aligned} \\ SO & \mathit{otherwise} \end{cases} \tag{7.28}$$

The ship can have different definitions of what it considers a safe distance in front of a ship ($\mathcal{I}_{SDF}$) and how close it can be at CPA ($\mathcal{I}_{SD}$). $\mathcal{M}_{DCPA_{\rho}}$ marks the distance at CPA while $\mathcal{M}_{DF_{\rho}}$ marks how far in front of ship ρ the reference ship crosses.

$$SD_{\rho}[t] = (\mathcal{M}_{DCPA_{\rho}}[t] > \mathcal{I}_{SD}) \wedge (\mathcal{M}_{DF_{\rho}}[t] > \mathcal{I}_{SDF}) \tag{7.29}$$

A ship is considered to act according to good seamanship (GS) if it does not perform both a starboard action (SA_{ρ}) and port action (PA_{ρ}) and if a change in course ($\mathcal{M}_{CIC}$) indicates whether it plans to pass with the other ship on its port or starboard side ($\mathcal{M}_{PS}$).

$$GS_{\rho}[t] = \neg(SA_{\rho}[t] \wedge PA_{\rho}[t]) \wedge \mathcal{M}_{CIC}[t] \neq \mathcal{M}_{PS}[t] \tag{7.30}$$

$$SA_{\rho}[t] = \mathcal{M}_{CIC}[t] == \mathit{starboard} \vee SA_{\rho}[t-1] \tag{7.31}$$

$$PA_{\rho}[t] = \mathcal{M}_{CIC}[t] == \mathit{port} \vee PA_{\rho}[t-1] \tag{7.32}$$

$$\tag{7.33}$$

The reference ship has safely passed ship ρ (P_{ρ}) if they have passed each other ($\mathcal{M}_{P_{\rho}}$) and are at a safe distance (SD_{ρ}).

$$P_{\rho}[t] = \mathcal{M}_{P_{\rho}}[t] \wedge SD_{\rho}[t] \tag{7.34}$$

Lastly, the probability information used by the COLREGS cost evaluation in the PSB-MPC for more informed traffic rule compliance is derived. The first probability $\hat{\mathbb{P}}^{i}_{WGW}$ considers if the reference ship or obstacle i will fulfill its give-way obligations, which is dependent on if it does not have higher priority, it intends to adhere to the COLREGS and does not display unmodelled behavior:

$$\hat{\mathbb{P}}^i_{WGW} = \Pr\{(\mathcal{I}_{P_\rho} \neq \mathit{higher}) \wedge \mathcal{I}_C \wedge \neg\mathcal{I}_U\} \tag{7.35}$$

The second probability considers if the reference ship or obstacle i will perform a COLREGS-compliant evasive maneuver, and is given by

$$\hat{\mathbb{P}}^i_{CCEM} = \Pr\{\mathcal{I}_C\} \tag{7.36}$$

7.4.2 Evaluating Prediction Scenarios

When evaluating a dynamic obstacle prediction scenario or candidate trajectory, it is assumed that the reference ship will follow that trajectory while all other ships keep their course and speed. Time until CPA, $\mathcal{M}_{TCPA_\rho}$, is only used to evaluate whether the reference ship must act now or can wait and act later. This is not relevant when evaluating a trajectory as the entire future motion of the reference ship is assumed known. Time until CPA is therefore instead set as the lowest acceptable definition of ample-time, $\mathcal{I}_{AT}$. If the reference ship intends to act after this point then a trajectory that keeps the course or speed for the entire encounter will get a high probability. This ensures that the own-ship will act to avoid collision if the reference ship plans to avoid collision at an unacceptable short time before collision.

7.4.3 Prior distribution

The prior distribution of what the reference ship thinks the COLREGS situation is, $\mathcal{I}_{CS_\rho}$, is defined at the start of the encounter. The probability that it is an overtaking situation is based on the classifier shown in figure 7.5. The classifier shown in figure 7.6 is used to evaluate the probability that it is a head-on situation. The uncertainty in the classifier represents the uncertainty that arises due to measurement errors and different definitions of the situation borders. The probability of there being a crossing situation is equal to the probability that it is not an overtaking nor a head-on situation. Whether the reference ship is aft or front of ship ρ defines whether the reference ship is overtaking, $OT\text{-}ing$, or being overtaken, $OT\text{-}en$. Whether ship ρ is on the starboard or port side of the reference ship at the start of the encounter defines whether it is an *CR-PS* or *CR-SS* situation.

7.4.4 Limiting computational burden

As the computational burden of evaluating the DBN increases with each new time-step, the number of time steps has to be limited. This was achieved by in most cases inserting a new measurement on the current time-step thereby overriding the previous measurement. A new time-step is made if the previous time a new time-step was made is more than $\Delta_{ts,max} = 20\,\mathrm{s}$, or if the previous time-step was made no less than $\Delta_{ts,min} = 10\,\mathrm{s}$ and either ship in the encounter has changed their course more than $\Theta = 15°$ or speed more than $\Upsilon = 15\,\mathrm{m/s}$.

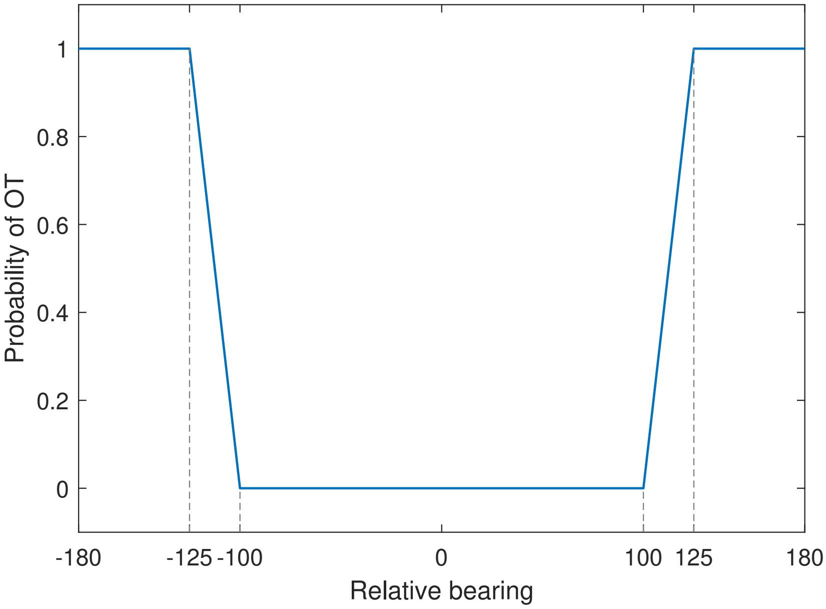


Figure 7.5: Classifier giving the probability that it is an overtaking situation. Relative bearing is defined as the bearing from the ship being overtaken to the overtaking ship relative to the heading of the ship being overtaken. 22.5° abaft the beam as specified in COLREGS rule 13 is the same as ±112.5° relative to the heading. This classifier considers a 15° uncertainty in the situation.

7.5 Experimental Results

The COLAV system described in Sections 7.2 - 7.3 was tested in the following nine different scenarios, with the relevant COLREGS rules indicated in parentheses:

1. Head-on scenario with correct dynamic obstacle behavior, where the obstacle makes a starboard turn (COLREGS rule 14).
2. Head-on scenario with wrong dynamic obstacle behavior, where the obstacle makes a port turn (COLREGS rule 14).
3. Crossing with a dynamic obstacle as stand-on and own-ship as the give-way vessel (COLREGS rule 15 and 16).
4. Crossing with the own-ship as stand-on, and a dynamic obstacle which does not adhere to COLREGS and does not give way (COLREGS rule 15 and 17).
5. Crossing with the own-ship as stand-on, only with a COLREGS-compliant dynamic obstacle taking a starboard turn (COLREGS rule 15 and 17).
6. Overtaking scenario with the obstacle being overtaken (COLREGS rule 13 and 16).

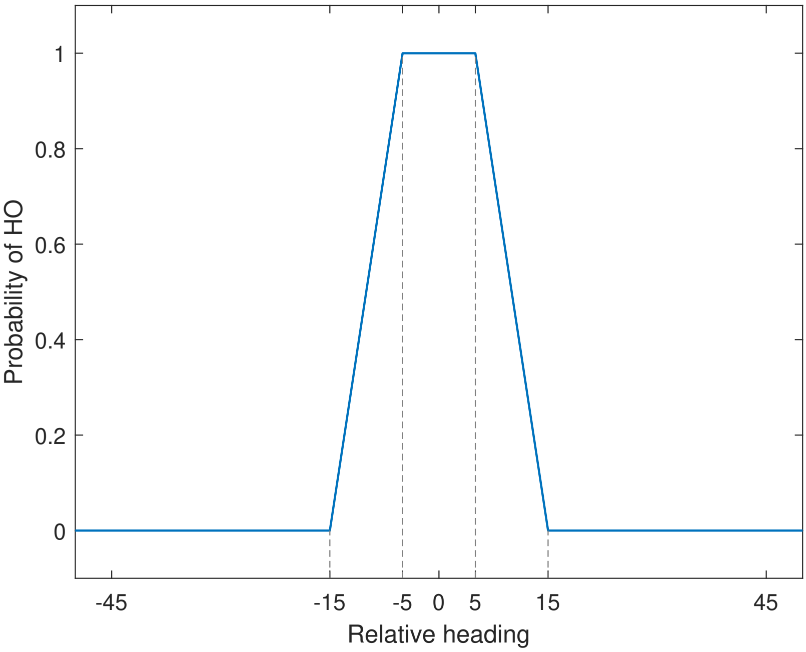


Figure 7.6: Classifier giving the probability that it is a head-on situation. The relative heading between the two ships defines the probability. This classifier considers a 10° uncertainty in the situation.

7. Overtaking scenario with the own-ship being overtaken (COLREGS rule 13 and 17).
8. Combined overtaking and crossing starboard side scenario, with the own-ship overtaking an obstacle and being the give-way vessel for another obstacle (COLREGS rules 13, 15 and 16-17).
9. Combined crossing starboard side and port side scenario (COLREGS rules 15 and 16-17).

Furthermore, COLREGS rule 7 on adequate collision risk assessment and rule 8 on performing apparent actions in ample time are also relevant for all the scenarios.

For each scenario, the chosen trajectory of the Milliampere 2 ferry together with the different prediction scenarios is shown at three different time instants. The thickness of the dashed lines, representing the different prediction scenarios, are scaled based on their likelihood as evaluated by the DOII. Additionally, how the intention states develop through the scenarios are shown together with the course and speed of all vessels involved. The priority state indicates whether the vessel will act as if it has a higher priority (ignoring its obligations to give way) or lower priority (ignoring its obligation to stand on). The COLREGS-compliant state indicates whether the vessel will ignore all the specifications in COLREGS, but still try to avoid collision. Good seamanship indicates whether or not the ship shows how it is going to act. Unmodelled behavior indicates all other non-compliant behavior.

In the two-ship scenarios, Havfruen was used as the dynamic obstacle $i = 1$. In the three-ship scenarios, the Cyberotter is the second dynamic obstacle $i = 2$. Milliampere 2 was set to track a desired speed of 1 m/s in all scenarios, except when it was overtaking another vessel, and in the combined overtaking and port side crossing scenario, where reference speeds of 2.0 m/s and 1.5 m/s were used, respectively. Low speeds were used due to the experiments being performed in confined waters, and because Milliampere 2 has a max speed limitation of 2.5 m/s. The dynamic obstacles speeds vary from 0.5 m/s to 3.0 m/s.

The PSB-MPC COLAV planning algorithm was tuned to reflect the confined space Milliampere 2 was to operate in. Two sequential maneuvers ($n_m = 2$) was considered in the horizon of the MPC, where the second maneuver is taken after $t_{ts} = 60$ s. Thus, the own-ship is planned to perform an initial maneuver at t_0, and a corrective one at $t_0 + t_{ts}$, which does not necessarily return the ship to its nominal trajectory. The own-ship safety zone was set to $d_{safe} = 8.6$ m. The algorithm parameters for grounding cost were chosen to allow the own-ship to maneuver within a margin of approximately 5 m to nearby grounding hazards.

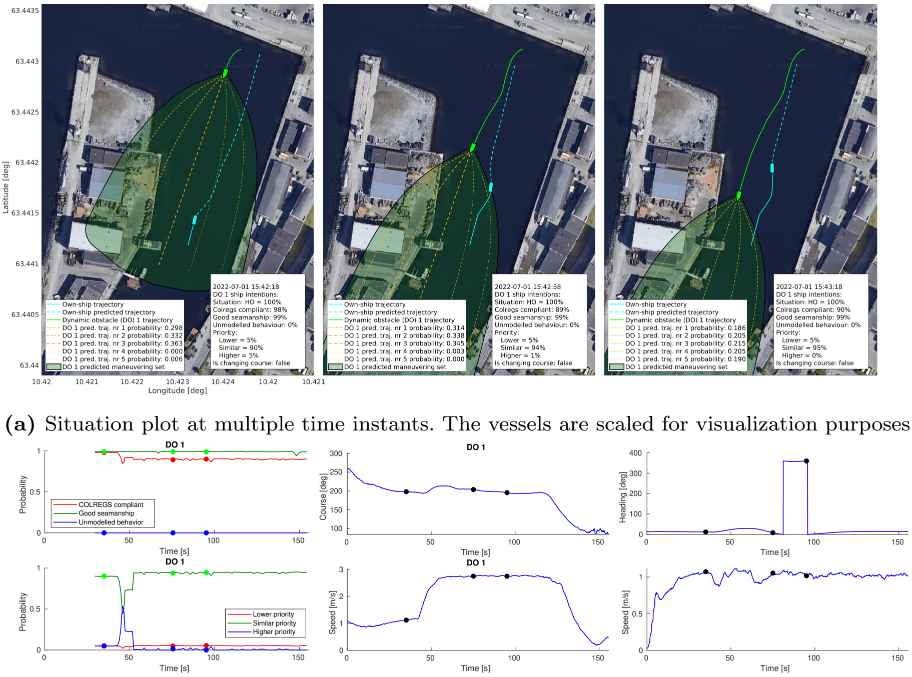
7.5.1 Scenarios

Head-on With a Compliant Dynamic Obstacle

Results from the compliant head-on scenario are given in Fig. 7.7 and show that the own-ship performs a COLREGS-compliant evasive maneuver. Shortly after the 40 s mark, the intention model starts to infer that the obstacle ship acts as if it has a higher priority than the own-ship. This is due to the ships getting quite close without the dynamic obstacle taking action. Once the dynamic obstacle takes an evasive action, the probability of it having higher priority quickly drops to 0.

Head-on With a Non-compliant Dynamic Obstacle

Results from the non-compliant head-on scenario is given in Fig. 7.8. The scenario shows that the own-ship starts to perform a COLREGS-compliant evasive maneuver. Similar to the previous scenario, the probability of the dynamic obstacle acting as if it has higher priority increases for a short time as the dynamic obstacle acts quite late. Once the dynamic obstacle changes course to port, the intention model switches between the dynamic obstacle either not being COLREGS-compliant or showing unmodelled behavior. At $t = 50$ s it has concluded on the dynamic obstacle not being COLREGS-compliant. The knowledge of the obstacle being non-compliant enables the own ship to disregard COLREGS as well and avoid collision through a port avoidance maneuver.



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

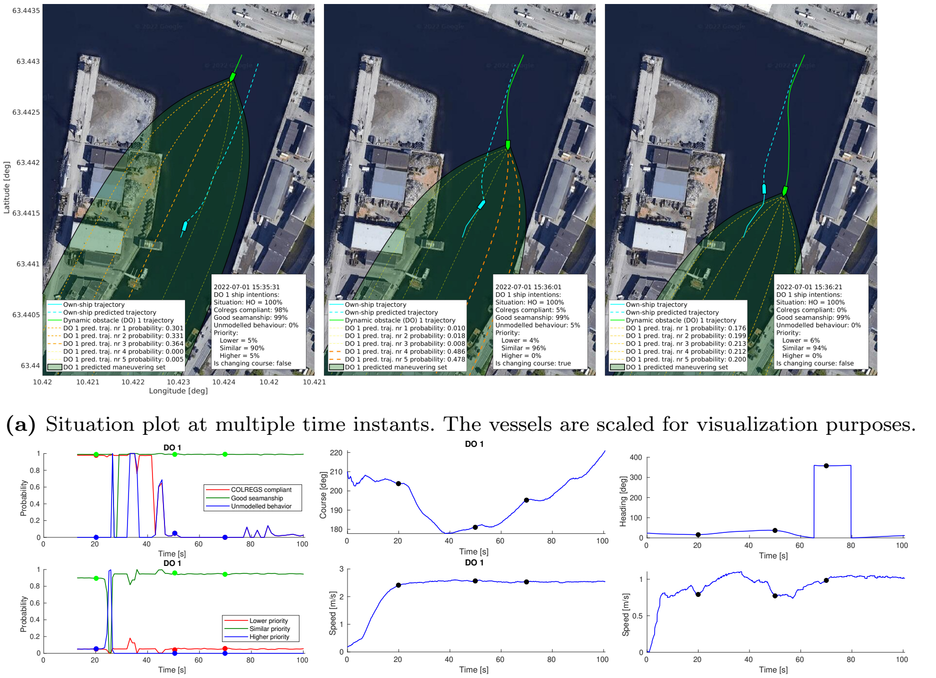
Figure 7.7: Head-on scenario with a compliant dynamic obstacle (DO). The dots on the lower subfigures show the values at the three time instants in (a).

Crossing With the Own-ship as Give-way Vessel, and a Compliant Stand-on Dynamic Obstacle

Fig. 7.9 shows results for the starboard side crossing. The own-ship performs a COLREGS-compliant evasive maneuver by changing its course to starboard and reducing its speed to avoid collision. Reducing the speed is the most effective action in this case as the confined spaces make larger course changes more susceptible to grounding. Furthermore, the Milliampere 2 thruster configuration and DP-system are configured for small heading changes and slow movements, which makes it easier to change speed than to alter the course. The collision avoidance algorithm is tuned such that Milliampere 2 should still try to change its course as this can be easier to see from the other vessel‘s point of view.

Crossing With the Own-ship as Stand-on Vessel, and a Compliant Dynamic Obstacle

Results from this scenario are given in Fig. 7.10. The intention model correctly predicts that the obstacle ship will make a starboard maneuver to avoid collision. The own-ship can therefore keep its course and speed without increasing the risk of collision. This fulfills the requirements that stand-on vessels shall keep their course



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

Figure 7.8: Head-on scenario with a non-compliant dynamic obstacle. The dots on the lower subfigures show the values at the three time instants in (a).

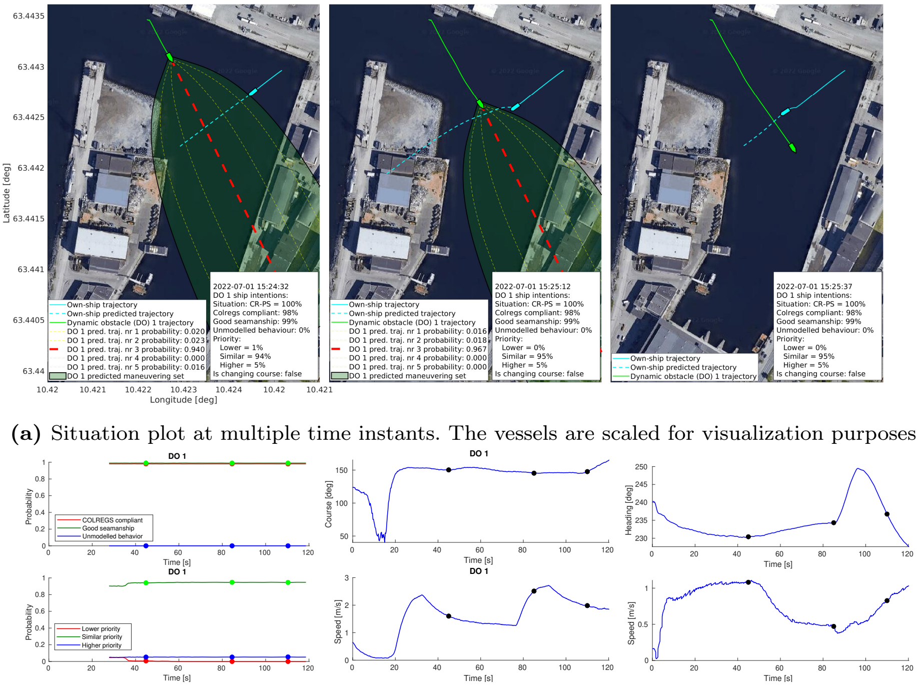
and speed (unless forced to give way), as specified in COLREGS rule 17.

Crossing With the Own-ship as Stand-on and a Non-compliant Dynamic Obstacle

Results from the port side crossing scenario with a non-compliant dynamic obstacle are given in Fig. 7.11. The probability of the dynamic obstacle acting as if it has a higher priority gradually increases as the obstacle ship comes closer without significantly changing its course or speed. Once it is quite likely that the dynamic obstacle will not give way, the own-ship decides to half its speed to avoid a potential collision. This shows that the resulting algorithm is able to deviate from the stand-on requirements when needed, as specified in COLREGS rule 17(b). The algorithm also adheres to rule 17(c) by not changing its course to port.

Overtaking

Results from the overtaking scenario are given in Fig. 7.12. The scenario shows that the own ship is able to avoid collision while overtaking, but the intention predictions are not ideal. This is due to the intention module interpreting all course changes larger than some threshold as being done with an intention. This does not work well



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

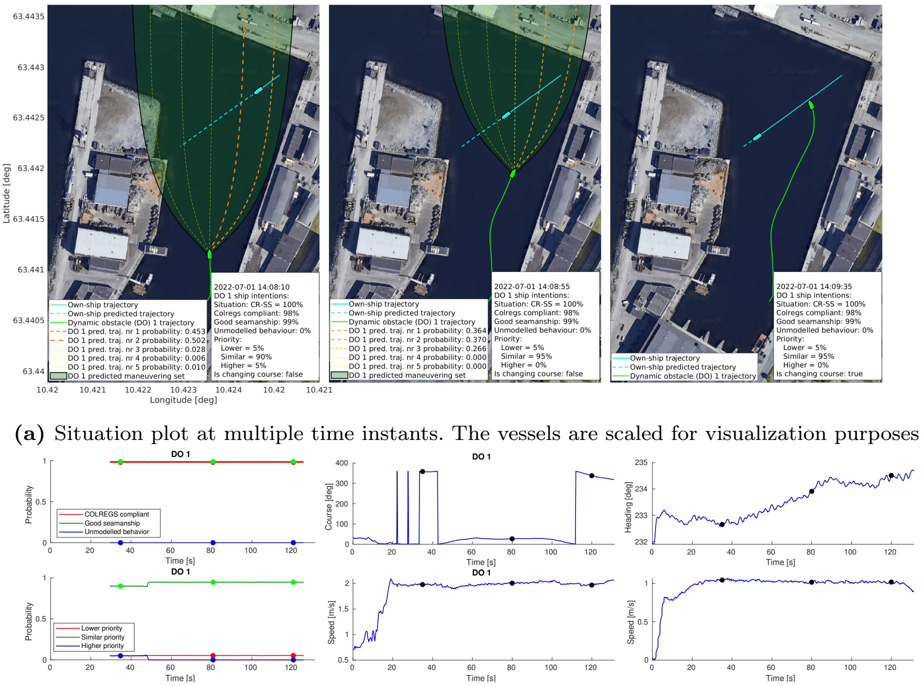
Figure 7.9: Crossing starboard side scenario with a compliant stand-on dynamic obstacle. The dots on the lower subfigures show the values at the three time instants in (a).

in this scenario as the dynamic obstacle is holding a too low velocity for keeping its course steady enough, as can be seen in Fig. 7.12(c). This experiment could not be performed at a higher speed due to the speed limitations of Milliampere 2.

The intention module concludes at the 40 s mark that the dynamic obstacle is displaying unmodelled behavior. This state can explain all possible behaviors and thus gives all future scenarios equal likelihood. The variations in the rest of the states after the 40 s mark can therefore not be caused by new observations as the observations are already explained by the unmodelled behavior state, nor do they affect the likelihood of different scenarios. We believe that these variations are computational quirks caused by the states being unobservable.

Overtaken

The COLAV system is shown to also handle its stand-on role when being overtaken, with results given in Fig. 7.13. A bit after the $t = 60\,\text{s}$ mark the dynamic obstacle changes course toward the own-ship. This is to avoid collision with floating platforms not shown in the figure. As the intention model observed that the dynamic obstacle changes course towards what seems like a collision course with



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

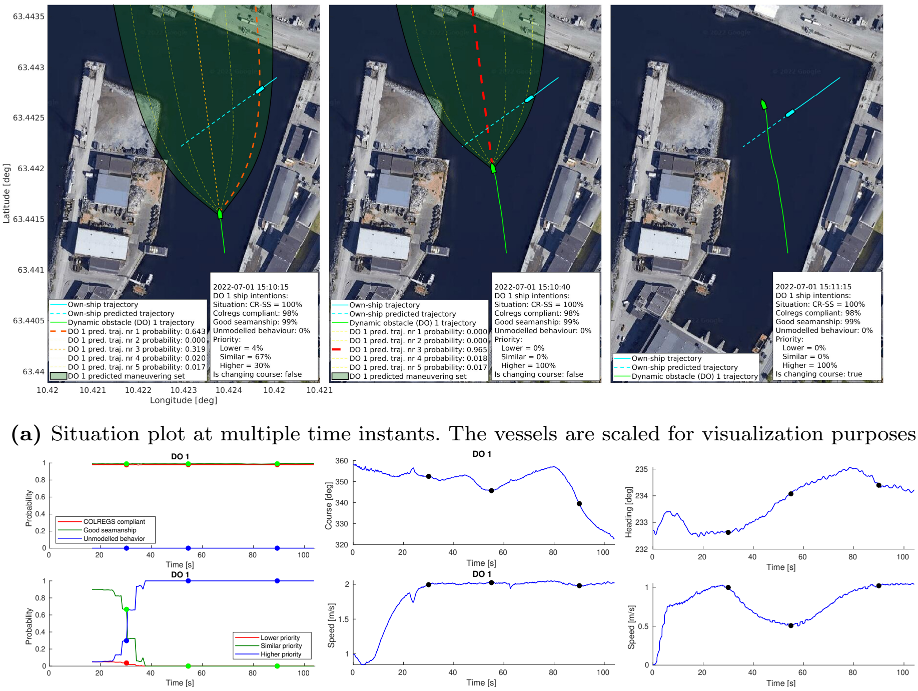
(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

Figure 7.10: Crossing port-side scenario with a compliant dynamic obstacle. The dots on the lower subfigures show the values at the three time instants in (a).

the own-ship, it marks the dynamic obstacle as showing unmodelled behavior. The good seamanship decreases right before it is marked as unmodelled behavior as the dynamic obstacle changes course to port while still planning to cross with the own-ship on its port side. Similar variations in intention states as discussed in section 7.5.1 are observed once unmodelled behavior becomes equal to 1.

The scenario also shows a weakness in the current PSB-MPC dynamic obstacle prediction setup described in Section 7.3.1, where obstacles are predicted to follow alternative trajectories about a nominal straight line from their current course and speed when the COLREGS situation starts. The waypoints set for the nominal straight line path of DO 1 did here not properly reflect the ground truth planned trajectory of the obstacle, as it changed course northwards when starting the overtaking maneuver. An improved approach would be to update the dynamic obstacle nominal straight line trajectory at regular intervals, especially after a COLREGS situation has ended. This would make the alternative prediction scenarios better reflect the possible maneuvering areas in the vicinity of the obstacle.



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

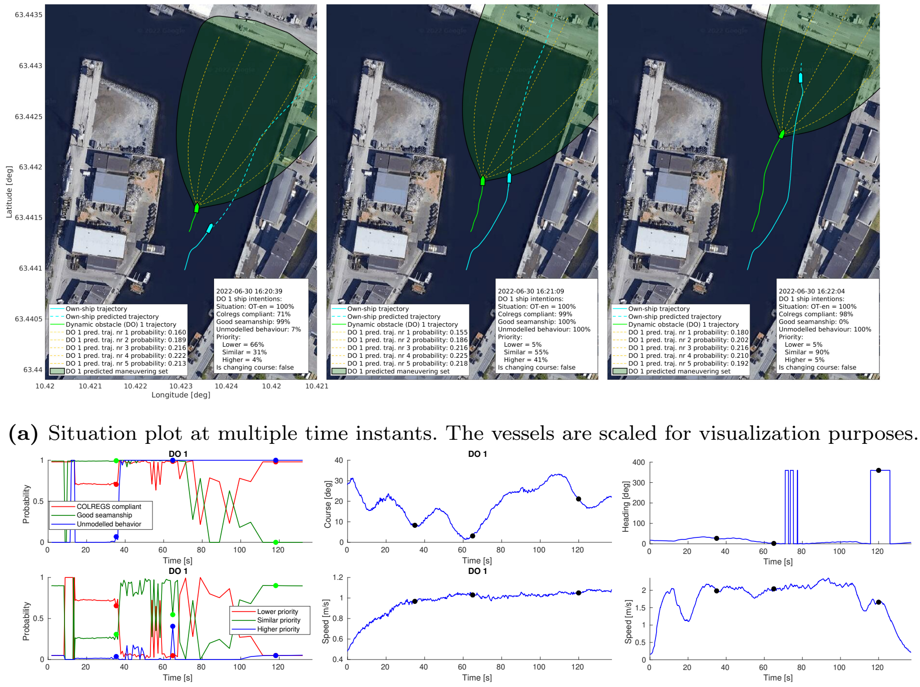
Figure 7.11: Crossing port-side scenario with a non-compliant dynamic obstacle. The dots on the lower subfigures show the values at the three time instants in (a).

Combined Overtaking and Crossing Port Side

Results from the first three-ship scenario is given in Fig. 7.14. Milliampere 2 has a stand-on role towards Havfruen (DO 1), and a give-way role towards the Cyberotter (DO 2). The PSB-MPC shows compliance with COLREGS rule 17(d) by ignoring its stand-on role and performing an evasive starboard maneuver. The intention inference module is able to estimate that Havfruen will make a give-way maneuver by passing behind both ships. For the Cyberotter, the intention model starts by correctly predicting that it will keep its course and speed. At the time $t = 50\,\text{s}$ the intention module notices that the Cyberotter changes its course towards starboard. As this is a change in course towards a collision, the intention module concludes with unmodelled behavior.

Combined Crossing Starboard Side and Port Side

Results from the second three-ship scenario are given in Fig. 7.15. The own-ship running the COLAV system initially slows down and then changes its course to pass behind the Cyberotter. The intention model initially predicts correctly that the Cyberotter will keep its course and speed, while Havfruen will cross behind the own-ship. A bit before the 50 s mark the Cyberotter is observed changing its



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

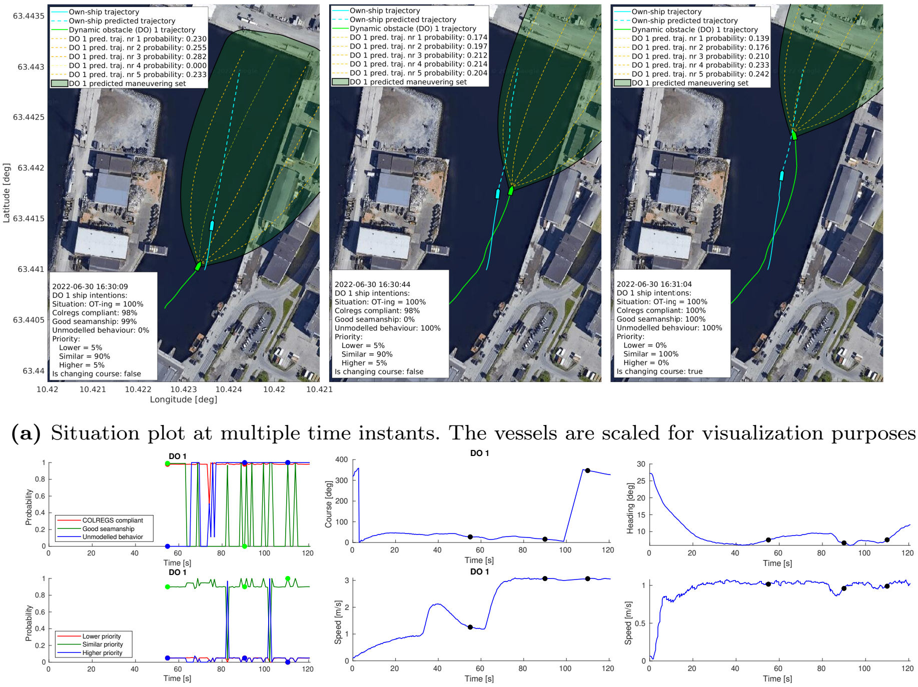
Figure 7.12: Scenario where the own-ship overtakes a dynamic obstacle. The dots on the lower subfigures show the values at the three time instants in (a).

course towards a collision, which makes the intention model mark it as unmodelled behavior.

7.5.2 Discussion

Experiment Outcome

From gauging the experimental results, the COLAV system using the PSB-MPC as a deliberate planning algorithm with a DBN for intent information shows promise and results in increased situational awareness for the own-ship. In the two crossing situations where the own-ship has a stand-on role, the intention-aware COLAV system enables the own-ship to keep its course and speed, as it predicts that the obstacle will perform an evasive maneuver. When it is apparent that the obstacle will not perform an evasive maneuver, the COLAV system makes use of its intention model to better avoid collision. In the head-on situation with a non-compliant dynamic obstacle, the intention module estimated that the dynamic obstacle was not adhering to the COLREGS, which enabled the PSB-MPC to ignore COLREGS as well to plan a collision-free trajectory. In the trials, the own-ship running the COLAV system is shown to comply with the COLREGS rules 8 and 13-17. Furthermore, better adherence with COLREGS rule 7 is also achieved, as the intention



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

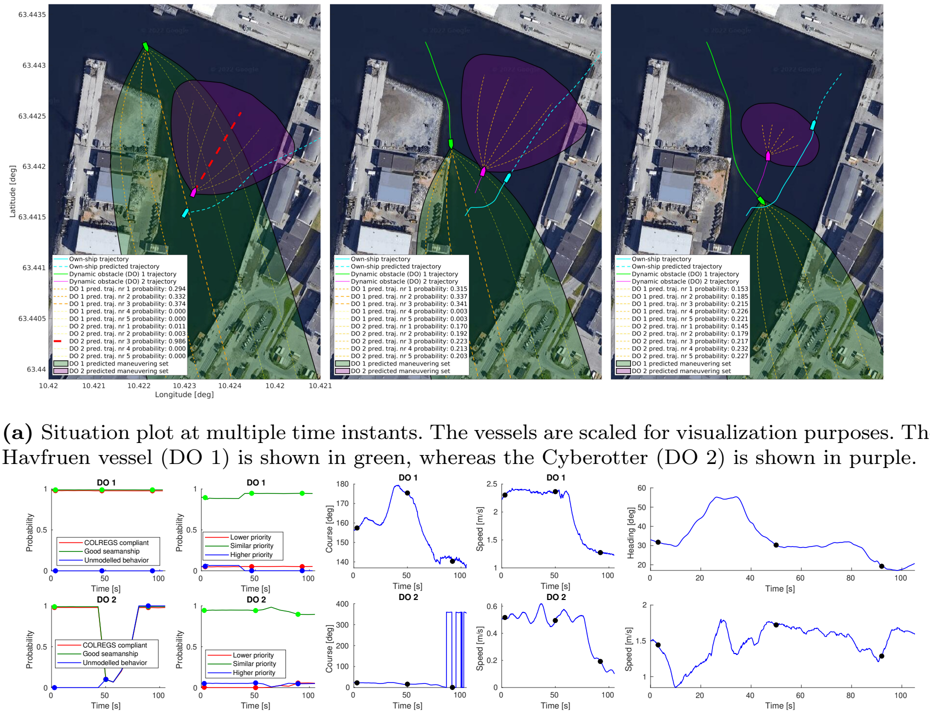
Figure 7.13: Scenario where the own-ship is overtaken by a semi-compliant dynamic obstacle. The dots on the lower subfigures show the values at the three time instants in (a).

module enables better collision risk assessment and COLREGS situation evaluation in the PSB-MPC planning algorithm.

Uncertainty Management

As the own-ship uses GNSS with a VRS for real-time corrections, the navigation data for the ferry has negligible uncertainty compared to the safety margin (d_{safe}) set for the ferry. On the other hand, no corrections were applied to the GNSS data from the two dynamic obstacles, and we thus relied on the Kalman-filter performing adequately. The filter was tuned such that positional estimates with standard deviations around 0.7 m were obtained, which was verified to be correct before the experiments started. Furthermore, the kinematic uncertainty associated with dynamic obstacle estimates was handled through the collision probability estimation in the PSB-MPC, through usage of the CE-method.

Regarding the map data used for avoiding grounding hazards in the PSB-MPC, a manual drive-through of the Nyhavna basin boundary was done to verify the data accuracy. As the map data did not include newer static obstacles such as the



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes. The Havfruen vessel (DO 1) is shown in green, whereas the Cyberotter (DO 2) is shown in purple.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

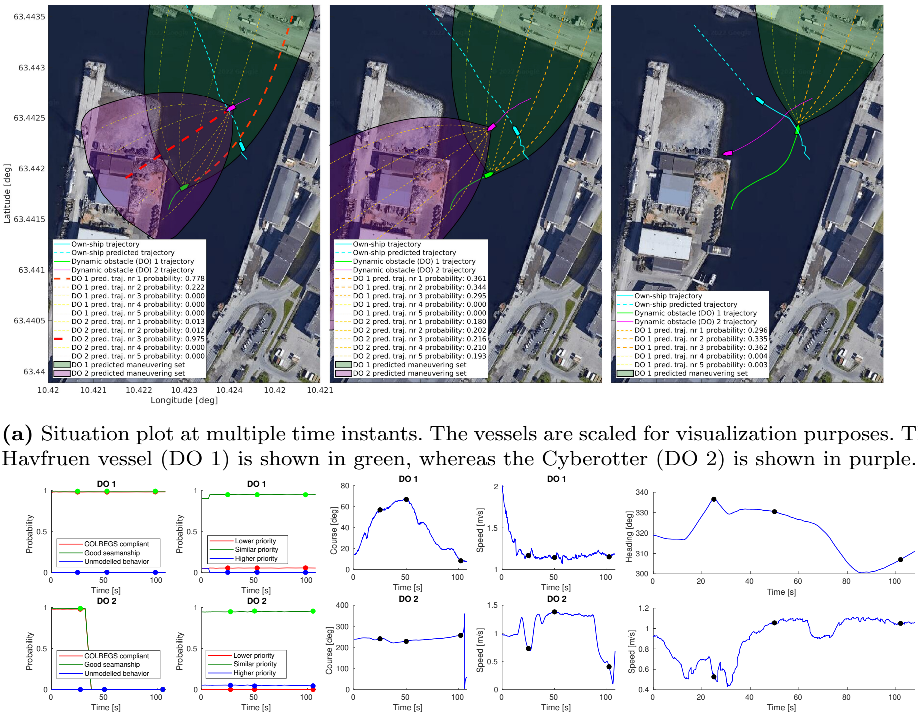
(d) Own-ship heading and speed.

Figure 7.14: Overtaking and crossing port-side scenario with compliant dynamic obstacles. The dots on the lower subfigures show the values at the three time instants in (a).

Havet sauna platform, a new land filling on the western side, and a few docked vessels along the basin boundary, un-mapped hazards were added manually before the experiment.

Limitations

There are a few limitations to the experiments in this work. Firstly the collision avoidance algorithm was designed for ships meeting on open seas, where COLREGS is normally considered. Due to the limitations of the Miliampere 2 ferry, the experiments had to be done in inland areas where the sea was sufficiently calm. The largest available area in Trondheim was quite small making it difficult to realistically test the algorithm. Secondly, the Milliampere 2 platform with its commercial DP-system was not designed for agile ship maneuvering, and this put a limitation on the performance one could extract from the vessel. The DP-system was overdamped, tuned for passenger comfort and small, slow movements as a ferry should perform. The vessel slows down substantially when performing heading maneuvers, which made the trajectory tracking challenging. For the overtaking scenarios, the own-ship had problems keeping to its speed due to the platform being designed



(a) Situation plot at multiple time instants. The vessels are scaled for visualization purposes. The Havfruen vessel (DO 1) is shown in green, whereas the Cyberotter (DO 2) is shown in purple.

(b) Dynamic obstacle intention states.

(c) Dynamic obstacle course and speed.

(d) Own-ship heading and speed.

Figure 7.15: Crossing on both starboard and port side, with compliant dynamic obstacles. The dots on the lower subfigures show the values at the three time instants in (a).

and tuned for keeping speeds of nominally 1.0 m/s.

These concerns made the limitations of the intention inference module affect the experiments more than we would expect it to do in open waters. With low velocities, the dynamic obstacles had problems keeping a constant course. As the intention inference module interprets all course changes larger than a threshold as deliberate actions taken by the dynamic obstacle, it often had to default to the unmodelled behavior state to explain its observations. This was especially apparent when using the Cyberotter (Sections 7.5.1 and 7.5.1) and for Havfruen in the overtaking scenario (Section 7.5.1) when its speed was reduced to 1 m/s. One could increase the course change threshold to filter away more of the random course changes. However, this has the drawback of making the module potentially miss actual course changes that it should consider. A smarter method for ignoring random motions is therefore warranted. Furthermore, the intention model does not consider land and static obstacles. This was especially noticeable in the overtaken scenario (Section 7.5.1) when the dynamic obstacle changes its course to avoid collision with a floating platform.

The intention inference module had little knowledge on neither its own nor the dynamic obstacle ship-type. Thus, the estimation of the dynamic obstacle acting as if it has higher priority, essentially means the module believes that the encountered obstacle is a ferry or other utility transport, which other leisure or smaller sized vessels have give-way obligations for in Norway. These cases would therefore still be COLREGS-compliant, despite the obstacle standing on in situations where it nominally has give-way obligations. However, future work would be to take ship-type information into account in the intention inference, as the obstacle would then be classified as a non-compliant ship because it in reality will have similar priority to the own-ship in these experiments. This will also make adherence to COLREGS rule 18 regarding vessel priorities easier.

While doing the experiments it became apparent that being able to use the heading of the other ships would be better than using the course. While supervising the experiments on board Milliampere 2 we could see that the heading of the dynamic obstacle had changed long before it became apparent when looking at the course alone. Enabling an autonomous ship to track the heading of the other ship, by for example using extended object tracking [167], could enable the inference module to more quickly realize what the other ship is doing and thereby enable the collision avoidance module to respond quicker.

7.6 Conclusion

In this chapter, a dynamic obstacle intention-aware PSB-MPC-based COLAV system has been presented, using a DBN for intention inference online. The resulting system is verified in experimental trials to show compliance with COLREGS rules 7-8, 13-17. The results presented show that incorporating a way of inferring the intentions of nearby dynamic obstacles proves to give COLAV planners such as the PSB-MPC improved situational awareness. This results in more efficient trajectory planning, where COLREGS-compliant maneuvers can be aborted if the other vessel is shown or estimated to be non-compliant. It also better enables stand-on compliance for autonomous agents, as it is necessary to infer to which degree a give-way vessel will perform proper maneuvering.

Future work involves also incorporating ship-type information in the intention inference and testing the intention-aware COLAV system in a more open sea environment, as the intention inference algorithm was originally not designed for situations in confined space with limited maneuvering possibilities. The presented COLAV system should also be tested in a less pre-planned setting, with target tracking in real maritime traffic. Further developments are needed to make the intention inference more robust to natural variations in the measured course of other ships. Furthermore, work is needed to improve the prediction of alternative obstacle maneuvering scenarios to also consider nearby grounding hazards and the current COLREGS situation.

Chapter 8

Vessel Destination and Kinematics Prediction Using a Maritime Traffic Graph

This chapter is based on the publication

[101] T. Tengesdal, L. Millefiori, P. Braca, and E. F. Brekke, "Joint Stochastic Prediction of Vessel Kinematics and Destination based on a Maritime Traffic Graph", in *2nd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives*, 2022. In press.

and details a method for long-term vessel prediction and inference of its destination. More accurate vessel predictions for longer time horizons, and infering their intended destination is useful in a COLAV setting, as the own-ship can better pro-actively plan an informed and efficient avoidance trajectory to follow.

Note that the notation is changed somewhat in this chapter, with the subscript i here denoting a destination, and *not* a dynamic obstacle as in the previous chapters.

8.1 Introduction

8.1.1 Motivation

Predicting the states of maritime vessels in long time horizons is important for ensuring safety at sea and efficient voyaging. This is especially the case for autonomous ship technology, where the situational awareness of the autonomous ship is dependent on the prediction quality of nearby vessel states. As the amount of ships at sea increases, it increases also the risk of collisions, which necessitates proper surveillance systems for vessels and also for onshore stations. However, given the ubiquity and volume of (live and historical) AIS data available on commercial traffic, data-driven methods have the potential to improve vessel predictions considering traffic patterns, thus alleviating the risk of collisions.

8.1.2 Previous Work

The problem of long-term prediction of vessel kinematic states from AIS data has been receiving much attention in the scientific literature recently. In [133], a vessel state prediction algorithm considering Single Point Neighbor Search (SPNS) around the current predicted state is used to find likely next-step state candidates. The main idea is that the next predicted state is calculated from the mean speed and course of the nearest neighbor set; If no data is available, the method defaults to a CV model. The major drawbacks of this method are that there is no clear uncertainty representation accompanying the prediction, and that it requires sufficient data to perform satisfactorily. The Neighbor Course Distribution Method (NCDM) is presented in [134], where a Gaussian Mixture Model (GMM) for prediction, with each component of the mixture representing a branch in a prediction tree. Nearest neighbor search is again applied to find likely prediction candidates, and Expectation Maximization (EM) is used to fit the GMM for all the branches in the tree, at each time step in the prediction. As in [133], the method performs poorly in areas with little historical data, but was an improvement in terms of having an uncertainty description.

With a purely model-based approach, in [79], [116] it has been proposed to represent a vessel's velocity with an OU stochastic process. This has been shown to work quite well with approximately non-maneuvering vessels, but the approach was also extended in [168] to account some vessel maneuvering. The advantage of using an OU process to model the ship‘s velocity is the limited growth in prediction uncertainty when compared with that of the CV model, thanks to the presence of a feedback loop on the target‘s velocity. Thanks to the presence of an additional parameter, the long-run mean velocity, which represents the desired (cruise) velocity of a vessel, the model can be used [168] to create a graph representation of maritime traffic, where major sea lane patterns are captured. More recently, the problem of vessel trajectory prediction has also been approached with recurrent neural networks (RNNs) in a Bayesian learning fashion; more specifically, an encoder-decoder structured Long Short-Term Memory (LSTM) architecture was introduced in [169]. The method, which is fully data-driven, is dependent upon an initial training phase, which requires adequate volume of training data in order to learn the predictive distribution of maritime trajectories. One interesting advantage of such neural architectures in this context is that additional information, such as the destination of the vessel, can be easily used to increase the prediction accuracy, for example discarding sea lanes that are not compatible with the destination.

Concerning the problem of destination inference, a Bayesian solution is proposed in [102]. A classical Bayesian framework is used to infer the most likely vessel endpoint out of a set of N_d a priori known destinations by creating bridged models connecting predicted states to the set of possible endpoints, eventually reducing the state prediction uncertainty. This method implicitly assumes that the initial state, the destination and the time of arrival are all independent. Even if this can be approximately true in many scenarios, it is not compatible with a Markov state process. This limitation has been overcome in [103], where the destination information was introduced in the form of pseudo-measurements, and the method was here shown to be more computationally efficient as the state space dimension is reduced compared to [102]. Recent work [170] has however showed that the two mentioned methods have practical challenges in more complex situations.

8.1.3 Contributions

The main contribution of this Chapter is the development of a Bayesian framework for destination inference utilizing a piecewise OU processes for capturing the destination information. The method relies on a directed maritime graph describing the major sea lanes or possible change points for vessels in a region, and uses it to construct sets of stochastic processes toward each possible destination $\boldsymbol{d}_i$. Each process is then described by a combination of an OU-process with piecewise long-run mean velocity values guiding the vessel along common sea-lanes, and a destination-reverting model towards the endpoint. The long-run mean velocities are taken from the maritime graph, which in turn is created from historical traffic data. This makes the Bayesian inference indirectly being able to account for ship maneuvering and better discount destinations that are passed by.

8.1.4 Chapter Overview

The Chapter is structured as follows. Section 8.2 contains information on the models used in this work, including details on the general bridging model format, the piecewise OU process and the ERV bridging model. In Section 8.3, background on maritime graphs are presented, and the graph used in in this Chapter is detailed. Utilizing the maritime graph information, the presented method for destination conditioned state prediction in Section 8.4 is detailed, which is used in the destination inference presented in Section 8.5. Results are then presented in Section 8.6 before conclusions are given in Section 8.7.

8.2 Models

8.2.1 Bridging Model

For the bridging processes to given destinations, the following Gaussian Linear Time Invariant (LTI) model is considered

$$\boldsymbol{x}_k = \boldsymbol{F}(h, \boldsymbol{d})\boldsymbol{x}_{k-1} + \boldsymbol{M}(h, \boldsymbol{d}) + \boldsymbol{w}_k \tag{8.1}$$

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{\epsilon}_k \tag{8.2}$$

to represent the Markovian transition density $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{d})$ and measurement likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$. The destination $\boldsymbol{d}$ is a variable vector in the model, and does in general contain the destination location position with zero velocity. The vessel kinematics are described by the state vector $\boldsymbol{x}_k = \boldsymbol{x}(t_k) = [\boldsymbol{s}_k^T \quad \dot{\boldsymbol{s}}_k^T]^T$ consisting of the

$r = 2$ two-dimensional position s_k and velocity $\dot{\boldsymbol{s}}_k$ of the maritime vessel in a suitable planar coordinate system. The term $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k; \boldsymbol{0}_{r\times 1}, \boldsymbol{Q}(h, \boldsymbol{d}))$ in (8.1) is the Wiener process noise, and $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{\epsilon}_k; \boldsymbol{0}_{r\times 1}, \boldsymbol{R}(h, \boldsymbol{d}))$ in (8.2) is the measurement noise. The subscript in $\boldsymbol{0}_{r\times 1}$ denotes the r-dimensional null vector. The matrices $\boldsymbol{F}$, $\boldsymbol{M}$ and $\boldsymbol{Q}$ describe the bridging model, and are in general functions of the time step $h = t_k - t_{k-1}$ and the destination d. For the measurement model, the matrix $\boldsymbol{H}$ depends on available information, but typically position plus SOG and COG can be used if AIS data is considered. A projection is then needed to transform the reported position from AIS in latitude and longitude to the planar reference system used. Here, $\boldsymbol{R}$ describe the measurement noise covariance.

8.2.2 The Equilibrium Reverting Velocity Model

The ERV model was introduced in [102] for destination inference applied to pointing tasks on in-car displays. The model is given through the following SDE

$$d\boldsymbol{x}(t) = \boldsymbol{A}_{ERV}(\boldsymbol{\mu}_d - \boldsymbol{x}(t))dt + \boldsymbol{B}_{ERV}d\boldsymbol{w}(t) \tag{8.3}$$

where $\boldsymbol{w}(t)$ is the Wiener process noise, $\boldsymbol{\mu}_d = [\boldsymbol{p}_d^T \quad \boldsymbol{0}_{r\times 1}^T]^T$ is the mean of the destination d consisting the position $\boldsymbol{p}_d$ and zero speed, and where $\boldsymbol{B} = [\boldsymbol{0}_{2r\times r} \quad \boldsymbol{\sigma}]$ is a matrix that governs the noise on the state velocity, with $\boldsymbol{\sigma}$ as the r-dimensional velocity noise covariance, often taken as diagonal. The system matrix is given through

$$\boldsymbol{A}_{ERV} = \begin{bmatrix} \boldsymbol{0}_{r\times r} & -\boldsymbol{I}_{r\times r} \\ \boldsymbol{\eta} & \boldsymbol{\rho} \end{bmatrix} \tag{8.4}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\rho}$ are the r-dimensional matrices of mean revertion strengths and drag coefficients in each spatial dimension, often taken as diagonal. In the form of (8.1), the discrete time system matrices can be written as

$$\boldsymbol{F}_{ERV}(h, \boldsymbol{d}) = e^{-\boldsymbol{A}h} \tag{8.5}$$

$$\boldsymbol{M}_{ERV}(h, \boldsymbol{d}) = (\boldsymbol{I}_{2r\times 2r} - e^{-\boldsymbol{A}h})\boldsymbol{\mu}_d \tag{8.6}$$

$$\boldsymbol{Q}_{ERV}(h, \boldsymbol{d}) = \int_t^{t+h} e^{-\boldsymbol{A}(t+h-l)}\boldsymbol{\sigma}\boldsymbol{\sigma}^T e^{-\boldsymbol{A}^T(t+h-l)}d\boldsymbol{l} \tag{8.7}$$

where the ERV covariance matrix $\boldsymbol{Q}_{ERV}(h, \boldsymbol{d})$ can be calculated using matrix exponentials as in [171]. The model can be viewed as if the state $\boldsymbol{x}(t)$ is being pulled by a virtual spring towards the destination by the $\boldsymbol{\eta}$ factor, and with damping given by the $\boldsymbol{\rho}$ factor constraining the velocity.

8.2.3 The Piecewise Ornstein-Uhlenbeck Process

As described in Section 2.4.2, the OU process has beneficial uncertainty propagation properties and validity when used for modelling non-maneuvering vessels [115]. Since commercial traffic vessels normally follow patterns where they keep their current speed, with changes in course at selected waypoints, the case when the OU mean velocity can change is considered. Here, piecewise mean velocities $\boldsymbol{v}_j, \forall j \in \{1, 2, 3, ..., N_w\}$ for each segment from waypoint with index j to $j + 1$ are considered, which will coincide with the edges in the maritime graph described in the next section. The detection of mean velocity changes can be done using Page's test as in [168].

Then, considering an OU process as in (2.14) with piecewise constant mean velocities, the expected predicted state arriving at the final waypoint in the path $\boldsymbol{x}_{N_w}$ from the current time t_k is given by [172]

$$\begin{aligned}\boldsymbol{x}_{N_w} &= \boldsymbol{\Phi}(\Delta_{N_w})\boldsymbol{x}_k + \boldsymbol{\Psi}(\Delta_{N_w})\boldsymbol{v}_{N_w} + \sum_{n=k}^{N_w-1}\left[\prod_{m=k+1}^{N_w}\boldsymbol{\Phi}(\Delta_m)\right]\boldsymbol{\Psi}(\Delta_n)\boldsymbol{v}_n \qquad (8.8)\\ &= \boldsymbol{\Phi}(\Delta_{N_w})\boldsymbol{x}_k + \boldsymbol{c}_{N_w}, \qquad (8.9)\end{aligned}$$

with

$$\boldsymbol{c}_{N_w} = \boldsymbol{\Psi}(\Delta_{N_w})\boldsymbol{v}_{N_w} + \sum_{n=k}^{N_w-1}\left[\prod_{m=k+1}^{N_w}\boldsymbol{\Phi}(\Delta_m)\right]\boldsymbol{\Psi}(\Delta_n)\boldsymbol{v}_n, \qquad (8.10)$$

where $\Delta_{N_w} = t_{N_w} - t_k$, $\Delta_n = t_n - t_{n-1}$ and $\boldsymbol{v}_n$ is the mean velocity relevant on the time span Δ_n from t_{n-1} to t_n.

8.3 Maritime Graph Representation

8.3.1 Description and Creation

As the majority of traffic at sea is regular, one can utilize historical AIS data from multiple vessels to construct a graph $\mathcal{G}$, which represents recurrent traffic patterns in a region of choice. The creation of the graph can be done using unsupervised learning and piecewise OU processes [168], but will not be elaborated here. From such a maritime graph, one can extract the required components used in the proposed destination inference method:

- The set of $N_\mathcal{C}$ waypoints where vessels change their velocity direction.
- An adjacency matrix $\boldsymbol{\Lambda}_{ADJ} = (\tau_{nm}) \in \mathbb{N}^{N_\mathcal{C} \times N_\mathcal{C}}$ which determines if there is a connection or not between waypoints n and m.
- Distances $\mathcal{D}_{nm}$ between connected waypoints n and m.
- The OU mean velocity angle or direction $\angle \boldsymbol{v}_{nm}$ on an edge between connected waypoints n and m.

It is assumed that the vessels considered in the inference are associated with an edge in the maritime graph, i.e., that the vessel velocity is compatible with the OU mean velocity along its associated waypoint segment. If that is not the case, one can perform track-to-graph association as in [173].

For the purpose of testing the destination inference method in this chapter, a maritime graph manually based on an AIS dataset from 2015 in the Trondheimsfjord was constructed. The result can be seen in Fig. 8.1, where all the destinations with

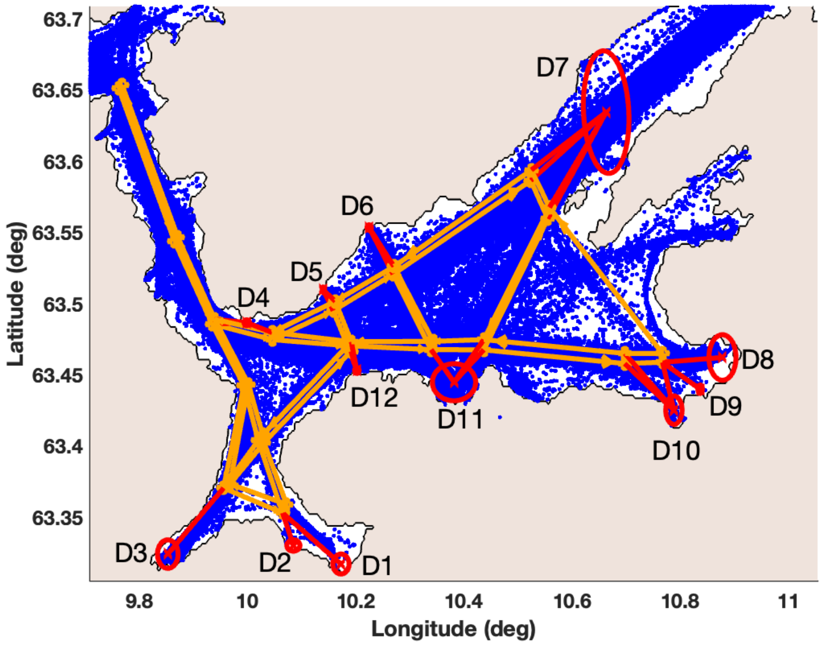


Figure 8.1: Overview of a manually created maritime graph based on AIS data from the Trondheimsfjord, with the data points shown as blue dots. The orange circles and lines denote graph nodes and the directional edges between them. Red ellipses describe the prior on all the destinations $D1$ to $D12$ shown in the fjord, where the red lines mark the bidirectional edge from a close graph node to a destination.

their priors are also shown, for the destination set described in Table 8.1. The graph nodes are put at common change points for vessels, with directional edges in between which shows the major maritime traffic lanes.

8.3.2 Destination-conditioned Paths

To predict a vessel trajectory long-term from an initial state $\boldsymbol{x}_0$, the maritime graph is used to find the set of N_d shortest paths from $\boldsymbol{x}_0$ to each $\boldsymbol{d}_i$, using Dijkstra's algorithm [174]. Each path $\mathcal{P}_i = \{\mathcal{W}^i, \angle\nu^i, \mathcal{T}^i_{OU}\}$ consists of

- Its waypoints $\mathcal{W}^i = \{\boldsymbol{W}^i_1, ..., \boldsymbol{W}^i_j, ..., \boldsymbol{W}^i_{N^i_w}\}$.
- The angles of the mean velocity $\angle\nu^i = \{\angle v^i_1, ..., \angle v^i_j, ..., \angle v^i_{N^i_w - 1}\}$ along each waypoint segment.
- The predicted times $\mathcal{T}^i_{OU} = \{T^i_{OU,1}, ..., T^i_{OU,j}, ..., T^i_{OU,N^i_w}\}$ for the vessel to arrive at each waypoint.

The number of waypoints in the path towards the i-th destination, excluding the destination itself, is denoted with N_w^i. The angles of the mean velocity for each leg are found through $\angle v_j^i = \operatorname{atan2}(y_{j+1}^i - y_j^i, x_{j+1}^i - x_j^i)$, where $\operatorname{atan2}(y, x)$ is the four-quadrant inverse tangent function, and x_j^i and y_j^i are the north and east coordinates of waypoint W_j^i. The predicted waypoint arrival times for the vessel are found heuristically through $T_{OU,j}^i = dist_{j,j+1} / \left\| \boldsymbol{v}_j^i \right\|$, where $dist_{j,j+1}$ is the distance between waypoint $\boldsymbol{W}_j^i$ and $\boldsymbol{W}_{j+1}^i$ and $\left\| \boldsymbol{v}_j^i \right\|$ is the mean velocity modulus. For the initial segment, the time $T_{OU,j}^i$ is calculated using the distance from the vessel to the waypoint. Note that these arrival times are a priori and uses the current OU mean velocity estimate for the vessel. To remedy this, the shortest paths are recomputed on regular intervals, to take into account changes in vessel speed and position.

For this reason, the long-run mean velocity is estimated as follows: the long-run mean speed is estimated from AIS with a running average of the reported speed, while the course is given by the angle of the long-run mean velocity at the closest waypoint segment.

8.4 Path-conditioned Prediction

What separates the approach presented here from [102] and [103] is that, given a destination $\boldsymbol{d}_i$, a bridging process such as ERV is only considered for destination inference from the last waypoint node $\boldsymbol{W}_{N_w^i}^i$ in the graph $\mathcal{G}$, which is closest to the considered destination node. Before that stage, when the vessel is not on the navigational leg from waypoint $\boldsymbol{W}_{N_w^i}^i$ to $\boldsymbol{d}_i$, piecewise OU processes are used for prediction.

Thus, using the bridging model in (8.1) together with the piecewise OU predicted state (8.9) at the final waypoint in the graph $\mathcal{G}$ closest to a destination $\boldsymbol{d}_i$, the mean state of the prediction from the current state $\boldsymbol{x}_k$ to an arrival time state $\boldsymbol{x}_k$ at destination i can be written as

$$\begin{aligned} \boldsymbol{\mu}_K^i &= \boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i) \left[\boldsymbol{\Phi}(\Delta_{N_w^i}) \boldsymbol{x}_k + \boldsymbol{c}_{N_w^i} \right] + \boldsymbol{M}(\Delta_{K^i}, \boldsymbol{d}_i) \\ &= \boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i) \boldsymbol{\Phi}(\Delta_{N_w^i}) \boldsymbol{x}_k + \boldsymbol{\zeta}^i \end{aligned} \tag{8.11}$$

with $\Delta_{K^i} = t_K - t_{N_w^i}$, $\boldsymbol{\zeta}^i = \boldsymbol{M}(\Delta_{K^i}, \boldsymbol{d}_i) + \boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i) \boldsymbol{c}_{N_w^i}$. The term $\boldsymbol{\zeta}^i$ will here be a factor reflecting convergence effect the piecewise OU process has, combined with the pull tendency from the ERV bridge model.

$$\boldsymbol{\Sigma}_K^i = \boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i) \boldsymbol{P}_{N_w^i}^i \boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i)^T + \boldsymbol{Q}(\Delta_{K^i}, \boldsymbol{d}_i) \tag{8.12}$$

where $\boldsymbol{P}_{N_w^i}$ is the predicted covariance at the last waypoint using the piecewise OU process, initialized at the current time t_k to a destination conditioned Kalman-filtered estimate $\boldsymbol{\Sigma}_k^i$. The KF used to track the vessel then uses the combined piecewise OU model and bridge model as basis.

When $t_k > T_{OU,N_w^i}^i$, meaning the vessel is on the bridge process from the last waypoint in the path to destination $\boldsymbol{d}_i$, the prediction in (8.11) reduces to the bridge model prediction in (8.1). For the uncertainty prediction initialization, the covariance $\boldsymbol{\Sigma}_k$ obtained from the KF would then be used instead. Prediction with the proposed method for the case when $t_k \leq T^i_{ou,N^i_w}$ is illustrated in Fig. 8.2 for two destinations.

Thus, the predictive distribution $p(\boldsymbol{x}_K|\boldsymbol{x}_k)$ from the current state $\boldsymbol{x}_k$ to an arrival time state $\boldsymbol{x}_K$ at destination i is

$$p(\boldsymbol{x}_K|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{x}_K; \boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i)\boldsymbol{\Phi}(\Delta_{N^i_w})\boldsymbol{x}_k + \boldsymbol{\zeta}^i, \boldsymbol{\Sigma}_{K^i}), \tag{8.13}$$

with moments found in (8.11) and (8.12).

8.5 Destination Inference

8.5.1 Bayesian Framework

The goal is to infer the posterior destination weights. Using Bayes' rule, one can write the posterior destination weights as

$$p(\boldsymbol{d} = \boldsymbol{d}_i|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{d} = \boldsymbol{d}_i)p(\boldsymbol{y}_{1:k}|\boldsymbol{d} = \boldsymbol{d}_i) \tag{8.14}$$

with $p(\boldsymbol{d} = \boldsymbol{d}_i)$ as the destination prior. For simplicity and in the interest of light notation, $p(\boldsymbol{d}_i)$ is used to denote $p(\boldsymbol{d} = \boldsymbol{d}_i)$.

A similar approach as in [103] is considered, where the a-priori information on destinations is incorporated as pseudo-observations $\widetilde{\boldsymbol{y}}^i_K$ at unknown random arrival time $t_K = T$, with prior distributions

$$p(\widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i|\boldsymbol{x}_K, \boldsymbol{d} = \boldsymbol{d}_i) = \mathcal{N}(\widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i|\widetilde{\boldsymbol{G}}\boldsymbol{x}_K, \boldsymbol{\Sigma}_i), \tag{8.15}$$

where the observation matrix $\widetilde{\boldsymbol{G}}$ depends on available information about the position and speed of the vessel at the destinations. This entails that each destination $\boldsymbol{d}_i$ has prior $\boldsymbol{d}_i \sim \mathcal{N}(\boldsymbol{a}_i, \boldsymbol{\Sigma}_i)$.

As the arrival time is included in the conditioning and considered a random variable, one must marginalize out T through

$$p(\boldsymbol{y}_{1:k}|\boldsymbol{d}_i) = \int p(\boldsymbol{y}_{1:k}|\boldsymbol{d}_i, T)p(T|\boldsymbol{d}_i)d_T \tag{8.16}$$

in order to calculate the destination weights in (8.14). This one-dimensional integral is in general non-trivial, and one typically consider a quadrature approximation such as Simpsons' method with a fixed set of quadrature points $\{T_1, T_2, ..., T_{N_q}\}$ in order to approximate (8.16). More information can be found in [102]. Again, in order to simplify notation, the conditioning on the arrival time T is omitted in the remaining text.

The incorporation of pseudo-observations for each destination is done by using that $p(\boldsymbol{y}_{1:k}|\boldsymbol{d}_i) = p(\boldsymbol{y}_{1:k}|, \widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i)$ [103]. Then, the Predicted Error Decomposition

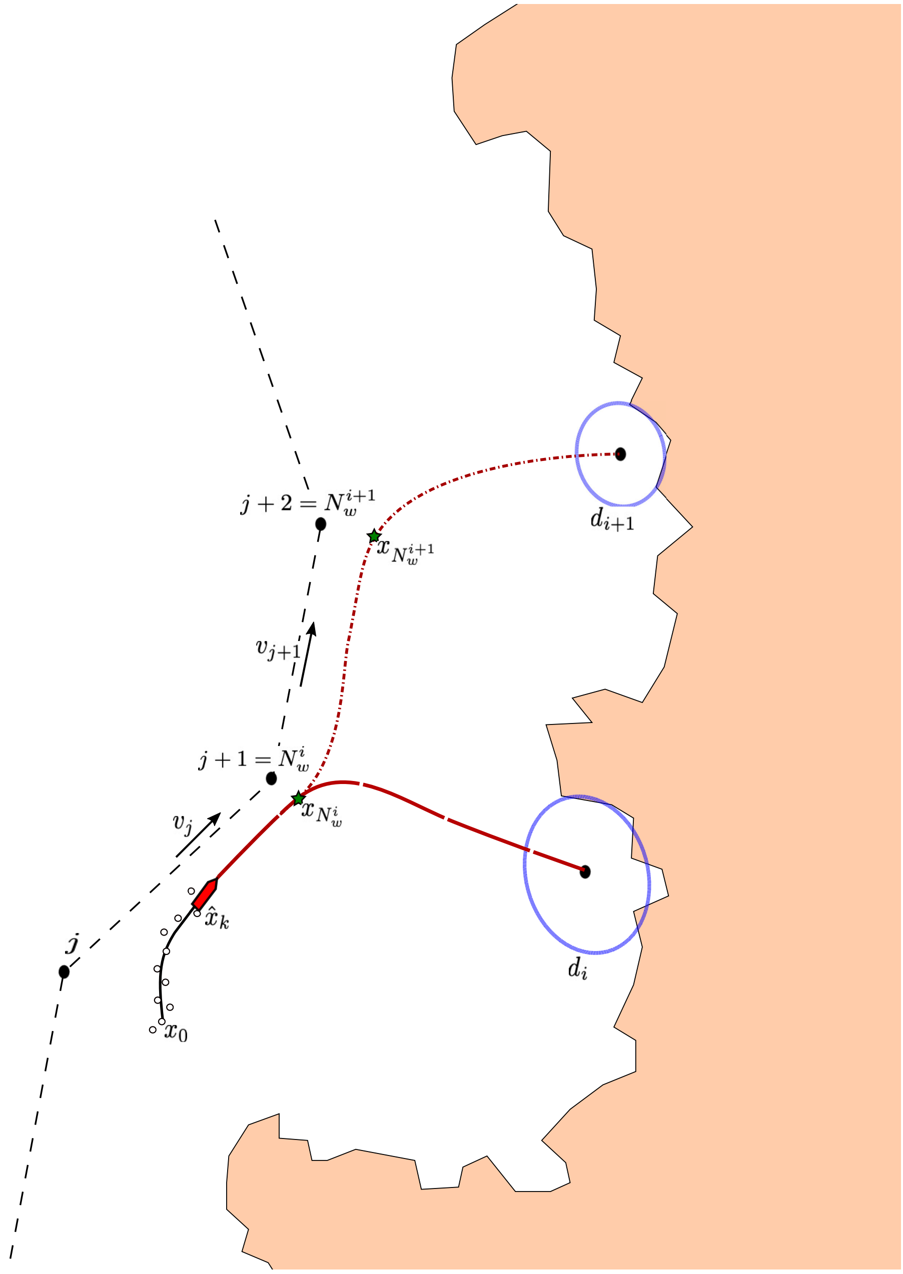


Figure 8.2: Illustration of the proposed method with two arbitrary destinations $\boldsymbol{d}_i$ and d_{i+1}. An example maritime graph with waypoints of indices j, $j+1$ and $j+2$, and the OU mean velocities $\boldsymbol{v}_j$ and $\boldsymbol{v}_{j+1}$ along each leg are also shown. The sample vessel track starts in $\boldsymbol{x}_0$, and is estimated using the measurements marked as circles up until the illustrated current time red vessel position. From there, sample predictions for each destination are shown. As the leg from j to $j+1$ is closest to the initial position at $\boldsymbol{x}_0$, previous graph legs are not considered.

(PED) $p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{d}_i)$ is calculated as

$$\begin{aligned}
p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{d}_i) &= \int p(\boldsymbol{y}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}, \widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i) d\boldsymbol{x}_k \\
&= \int p(\boldsymbol{y}_k|\boldsymbol{x}_k) \frac{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) p(\widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i|\boldsymbol{x}_k)}{p(\widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i|\boldsymbol{y}_{1:k-1})} d\boldsymbol{x}_k \\
&= \int \mathcal{N}(\boldsymbol{y}_k; \boldsymbol{H}\boldsymbol{x}, \boldsymbol{R}) \cdot \mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}^i_*, \boldsymbol{\Sigma}^i_*) d\boldsymbol{x}_k \\
&= \mathcal{N}(\boldsymbol{y}_k; \boldsymbol{\mu}^i_y, \boldsymbol{\Sigma}^i_y),
\end{aligned} \tag{8.17}$$

where $\boldsymbol{\mu}^i_y = \boldsymbol{H}\boldsymbol{\mu}^i_*$ and $\boldsymbol{\Sigma}^i_y = \boldsymbol{H}\boldsymbol{\Sigma}^i_*\boldsymbol{H}^T + \boldsymbol{R}$. Similarly to [103], by using the path-conditioned prediction in (8.13), the moments $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}_*$ can be found to be

$$\begin{aligned}
\boldsymbol{\mu}^i_* &= \boldsymbol{\mu}^i_{k|k-1} + \boldsymbol{L}^i_* \left[\boldsymbol{a}_i - \boldsymbol{B}_*\boldsymbol{\mu}^i_{k|k-1} - \widetilde{\boldsymbol{G}}\boldsymbol{\zeta}^i \right], \\
\boldsymbol{L}^i_* &= \boldsymbol{\Sigma}^i_{k|k-1}\boldsymbol{B}^T_* \left[\boldsymbol{B}_*\boldsymbol{\Sigma}^i_{k|k-1}\boldsymbol{B}^T_* + \widetilde{\boldsymbol{G}}\boldsymbol{\Sigma}_{K^i}\widetilde{\boldsymbol{G}}^T + \boldsymbol{\Sigma}_i \right], \\
\boldsymbol{\Sigma}^i_* &= (\boldsymbol{I}_{r\times r} - \boldsymbol{L}^i_*\boldsymbol{B}_*)\boldsymbol{\Sigma}^i_{k|k-1}, \\
\boldsymbol{B}_* &= \widetilde{\boldsymbol{G}}\boldsymbol{F}(\Delta_{K^i}, \boldsymbol{d}_i)\boldsymbol{\Phi}(\Delta_{N^i_w})
\end{aligned} \tag{8.18}$$

The terms $\boldsymbol{\mu}^i_{k|k-1}$ and $\boldsymbol{\Sigma}^i_{k|k-1}$ are the predicted mean and covariance in the distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{d}_i)$, which is given by

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{d}_i) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{d}_i) p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}, \boldsymbol{d}_i) d\boldsymbol{x}_{k-1} \tag{8.19}$$

with moments calculated through N_d KFs [125] using the estimates $\boldsymbol{\mu}^i_k$ and $\boldsymbol{\Sigma}^i_k$ at the previous time step. Note that the moments $\boldsymbol{\mu}^i_{k|k-1}$ and $\boldsymbol{\Sigma}^i_{k|k-1}$ are different from the moments $\boldsymbol{\mu}^i_*$ and $\boldsymbol{\Sigma}^i_*$ in that the latter also incorporate conditioning on the arrival time T, which is not explicitly reported for simplicity in notation.

8.5.2 Method

The destination inference method is split into three stages: The first stage concerns the initialization of the method; then, for each destination, the two next stages deal with the case of the estimated vessel state at the current time being on a waypoint leg in the maritime graph, or on a bridge process from a graph waypoint $\boldsymbol{W}^i_{N^i_w}$ to $\boldsymbol{d}_i$. The stages are further elaborated below, with the entire approach summarized in Algorithm 5.

Initialization

The method requires a maritime graph, which is used to calculate the set of shortest paths from the considered initial vessel state to each destination. Each path $\mathcal{P}_i = \{\mathcal{W}^i, \angle\nu^i, \mathcal{T}^i_{OU}\}$ is then used to predict the vessel state conditioned on each destination, and in the PED calculation. A KF with the piecewise OU model and ERV bridging model as basis is used to track the vessel trajectory and calculate the predicted moments $\boldsymbol{\mu}^i_{k|k-1}$, $\boldsymbol{\Sigma}^i_{k|k-1}$ and corrected moments $\boldsymbol{\mu}^i_{k|k}$, $\boldsymbol{\Sigma}^i_{k|k}$.

On a Waypoint Segment

If $t_k \leq T^i_{OU,N^i_w}$, where T^i_{OU,N^i_w} is the a priori predicted time that the vessel will arrive at the start of the bridging leg from waypoint $\boldsymbol{W}^i_{N^i_w}$ to $\boldsymbol{d}_i$, the predictive distribution $p(\boldsymbol{x}_K|\boldsymbol{x}_k)$ from (8.13) is calculated. Then, the PED is calculated using the moments in (8.18).

On a Bridge Segment

If the vessel at the current time $t_k \geq t_{N^i_w}$ is on a bridging process between waypoint $W^i_{N^i_w}$ and $\boldsymbol{d}_i$, one can use destination inference for $\boldsymbol{d}_i$ as in [103], calculating the PED using only the bridging model.

Algorithm 5 Destination inference with the piece-wise OU, ERV bridging model and pseudo-observations

```
Require: {Graph $\mathcal{G}$; Pseudo-obs.: $\{\widetilde{\boldsymbol{y}}^i_K = \boldsymbol{a}_i, \boldsymbol{\Sigma}_i\}_{1 \leq i \leq N_d}$; Obs.: $\{\boldsymbol{y}_{1:k}\}$ }
Compute shortest path from the initial vessel state $\boldsymbol{x}_0$ to $\boldsymbol{d}_i$, $\forall \quad i$ along graph $\mathcal{G}$
Set likelihoods $L^{i,q}_0 = 1, \quad \forall \quad i, q$.
for $k = 2 : K$ do
    for $i = 1 : N_d$ do
        Predict and correct $\boldsymbol{\mu}^i_{k|k-1}$, $\boldsymbol{\Sigma}^i_{k|k-1}$ and $\boldsymbol{\mu}^i_{k|k}$,
            $\boldsymbol{\Sigma}^i_{k|k}$ using a KF
        for $q = 1 : N_q$ do
            if $t_k < T^i_{ou,N^i_w}$ then
                Compute $\boldsymbol{\mu}^i_*$ and $\boldsymbol{\Sigma}^i_*$ from (8.18)
            else
                Compute $\boldsymbol{\mu}^i_*$ and $\boldsymbol{\Sigma}^i_*$ as in [103]
            end if
            Compute $l^{i,q}_k = p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{d}_i)$ from (8.17)
            Compute $L^{i,q}_k = l^{i,q}_k \cdot L^{i,q}_{k-1}$
        end for
        Compute likelihood $p(\boldsymbol{y}_{1:k}|\boldsymbol{d}_i)$ from (8.16) as in [102]
    end for
    Compute $p(\boldsymbol{d}_i|\boldsymbol{y}_{1:k}) \approx \dfrac{p(\boldsymbol{y}_{1:k}|\boldsymbol{d}_i)p(\boldsymbol{d}_i)}{\sum_{i=1}^{N_d} p(\boldsymbol{y}_{1:k}|\boldsymbol{d}_i)p(\boldsymbol{d}_i)}$
end for
```

8.6 Experimental Results

The proposed method for destination inference is tested on an AIS dataset from the Trondheimsfjord. The testing is based on the maritime graph created in Fig. 8.1 with the chosen set of 12 total destinations. For the geographically interested reader, the description of the destination set considered are given in Table 8.1. In the results below, the method is compared to the methods in [102], [103] in

Table 8.1: Full set of considered destinations in the Trondheimsfjord.

Destination	Description
D1	Buvika
D2	Børsa
D3	Orkanger
D4	Stadsbygd
D5	Rørvik ferry dock
D6	Vanvikan
D7	Towards Levanger
D8	Stjørdal
D9	Muruvik
D10	Hommelvik
D11	Trondheim
D12	Flakk

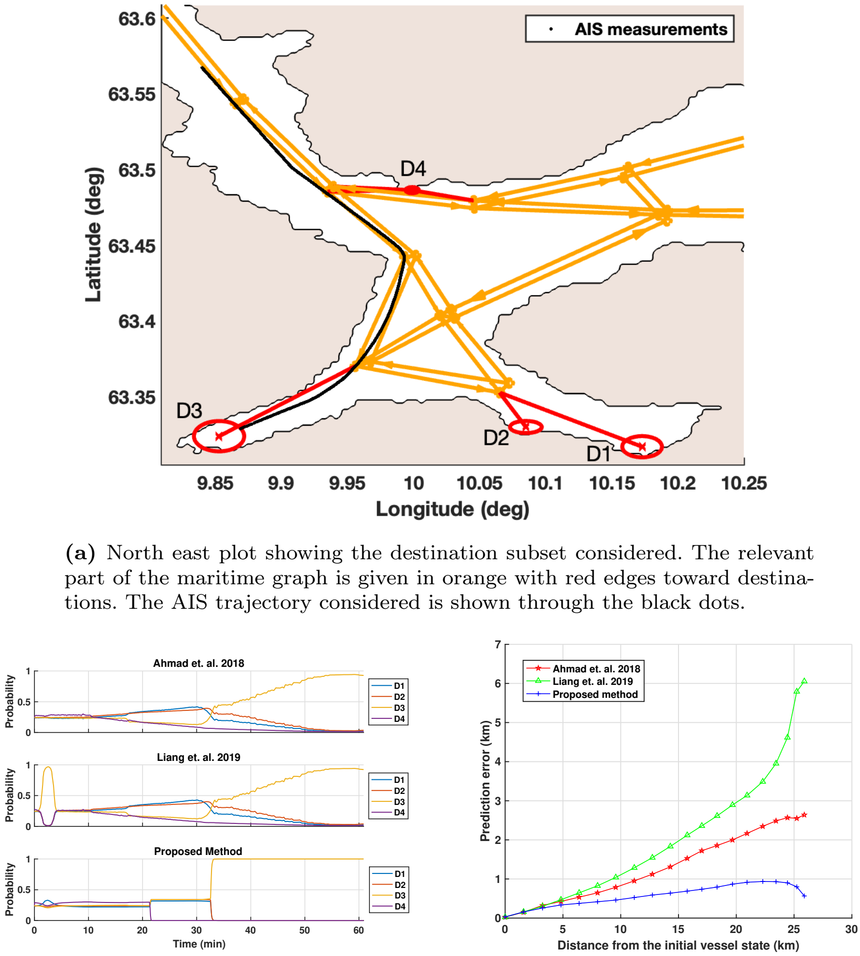
all scenarios. A linear KF using the piecewise OU model and bridging model is employed for tracking the vessel using the AIS measurements with position, SOG and COG available, whereas a CV model is used for the methods in [102], [103].

Fig. 8.3 shows the results on destination inference for a trajectory going from North to South towards Orkanger (D3), considering a subset of $N_d = 4$ destinations. Because of the consideration of common vessel traffic patterns along the created maritime graph, one is able to exclude passed destinations faster, as these destinations will have low likelihood under the piecewise OU model assumption. Likewise, one can more accurately predict vessel trajectories towards $\boldsymbol{d}_i$ that take into account the common turning points and thus also indirectly land. The current state of the art in [102], [103] either predicts straight lines or uses a bridging model directly from the current state, which will be inaccurate. This is partially reflected in the shown figure displaying the mean prediction error given the true destination.

8.7 Conclusion

In this chapter, a destination inference method for maritime vessels, which takes into account historical traffic patterns through the usage of a maritime graph, whose nodes represent common navigational waypoints and whose edges contain information on the direction and possible connections between waypoints. By constructing shortest paths through the graph to each considered endpoint, described by an OU-process and an ERV-process for the final segment, one can take typical ship maneuvering into account. The method is shown to perform better than current state of the art methods for destination inference on an experimental AIS dataset.

The main limitation of the prediction approach proposed here is that it only applies to typical trajectories (i.e., trajectories that are coherent with historical patterns



(a) North east plot showing the destination subset considered. The relevant part of the maritime graph is given in orange with red edges toward destinations. The AIS trajectory considered is shown through the black dots.

(b) Posterior destination weight evolution over time for all methods.

(c) Prediction error for all methods, based on their maximum a posteriori estimates conditioned on the true destination.

Figure 8.3: Destination inference results for a north-south going vessel towards destination D3 in Orkanger.

represented in the maritime traffic graph). Future research directions on this topic include the extension of the method to consider off-pattern trajectories whose behavior is not perfectly compatible with the maritime graph, nor conforms to having a constant speed profile.

Chapter 9

Conclusions

9.1 Summary and Discussion

This thesis has shown original contributions on topics within risk-based maritime COLAV for autonomous ships, with the specific aim of improving the collision risk assessment, situational awareness aspects and computational feasibility of such systems.

Results demonstrated in this work show the importance of performing adequate collision risk assessment in adherence with COLREGS rule 7, in order for the COLAV system to make safe and efficient maneuvering decisions. To enable this, uncertainty related to any hazardous situation at sea must be managed.

The first way towards uncertainty management for COLAV presented in this thesis, was through the development of two sampling-based CPEs for collision risk assessment in the PSB-MPC. By utilizing the uncertainty estimates accompanied with the kinematic state estimates from a tracking system, one essentially achieves a dynamical adjustment of the safety margins in the COLAV system. This is a step towards properly addressing research question **RQ-1** on adequate collision risk assessment, which in turn can result in better adherence to COLREGS rule 7. Using collision probability estimates in risk-based COLAV is advantageous as it is an intuitive way of considering uncertainty, which can be more easily interpreted than ad hoc risk functions that produce values with less root in reality. The disadvantage is on the other hand the COLAV's dependence on the modelling assumptions and definition of the probability underlying the estimator, which can often give conservative estimates. Furthermore, for sampling-based estimators, an adequate trade-off between high accuracy and low computational cost can be hard to find. The CPE will be another module to tune in the COLAV, the latter being a system of systems, which increases complexity.

Improving dynamic obstacle predictions and utilizing information on their intents has also been a large part of this work, which links to having adequate situational awareness in the COLAV system. A general Bayesian framework for obstacle intent inference was proposed, and lead to a DBN being developed in collaboration with Rothmund in [96]. The framework was used to provide the PSB-MPC with intent information, through e.g. the use of obstacle maneuvering scenario probabilities. To allow for using this information, an enhanced prediction scheme capable of predicting multiple alternative maneuvering scenarios for an obstacle was developed for the PSB-MPC, using the OU process, and then also as a combination with LOS-based guidance. The OU model made it possible to predict the obstacle uncertainty over longer time horizons with limited growth compared to if the classic CV model was used, which made it easier to use the CPE for probability estimation, and also to select an adequate number of obstacle maneuvering scenarios to consider. The resulting COLAV system was then capable of making more informed avoidance maneuvers, but at the cost of increased system complexity. The improvement of dynamic obstacle predictions and intent considerations in the PSB-MPC algorithm addresses many of the limitations with current state of the art in risk-based COLAV planning, and thus answers many of the concerns raised by research question **RQ-2**. Furthermore, improved dynamic obstacle prediction quality and knowledge on their intents will also help in achieving better collision risk assessment, thus also addressing research question **RQ-1**.

Integrating a CPE in the deliberate ship COLAV system adds a computational cost, as it is used to estimate the probability of collision between predicted trajectory pairs. This cost further increases with larger sets of obstacle maneuvering scenarios being considered. Using it bona-fide in the PSB-MPC on a sequential computation platform, the resulting run-time can be prohibitive for deploying the COLAV on a vessel. This is also the case when considering anti-grounding in MPC, as one typically has to calculate distances from the own-ship to possibly large numbers of static obstacles, for all considered avoidance trajectories. Depending on the grounding hazard representation and prediction horizon considered, this can also increase the COLAV run-time drastically. Thus, by developing a version of the PSB-MPC which utilizes a parallelized computing environment for its cost evaluation, the method can not only achieve reduced run-times, but also consider increasing amounts of information and explicitly account for the effect of uncertainty in hazardous situations at sea. By showing a detailed run-time analysis with respect to increasing situational information and own-ship decision candidates, the real-time computational feasibility of the algorithm is also demonstrated, which addresses research question **RQ-3**. Finally, experimental results using the PSB-MPC with a DBN for intention inference demonstrates that the deliberate COLAV planner also can be used in practice.

When travelling with a larger ship along a coast, there is a limited number of likely ports that can be considered for docking, with specific sea lanes typically being followed toward these ports. Knowing these sea lanes and the intended port of the ship is valuable, as one can better predict its future route, which again can be used in COLAV systems to increase their situational awareness. For this topic, a Bayesian inference approach informed by a maritime traffic graph was developed, in order to both predict the vessel kinematics in a longer time horizon, and also its intended destination. Historical traffic patterns was captured by the graph, making it possible to predict the vessel trajectory along common sea lanes towards considered destinations. This work on long-term prediction and destination inference also addresses the research question **RQ-2**, by improving on the vessel prediction quality compared to current state-of-the-art, and by capturing the intended ship destination.

9.2 Reflections and Future Work

The developed CPEs and intent information for use in a real-time feasible risk-based COLAV show promise when used in simulation for the assessment of difficult situations at sea, involving grounding hazards and maneuvering dynamic vessels, and where the target tracking performance of these vessels is limited. The experimental results with the PSB-MPC and DBN showed improved situational awareness and better risk assessment in a more pre-planned setting. However, work is needed to also demonstrate the the proposed intention-aware COLAV system can function in a less rehearsed setting at open sea, when faced with maritime traffic, environmental disturbances and variable target tracking performance. Furthermore, work is needed to integrate the Bayesian joint destination inference and long-term prediction method from Chapter 7 into the PSB-MPC. The work in this thesis further illuminates that topics within uncertainty management for COLREGS-compliant COLAV need to be more addressed in future research, to fully enable autonomous ship technology.

As the research on deliberate and risk-based COLAV progressed, the non-triviality of finding a proper tuning for such systems was quickly realized, which is a two-fold problem: Firstly, the tuning of the PSB-MPC algorithm itself is challenging, as a good trade-off between low dynamic collision risk, multi ship COLREGS compliance and minimal path deviation is hard to find for any given situation, when subject to a large number of adjustable parameters. One tuning for the PSB-MPC in an open sea situation with few dynamic and static obstacles can be inadequate when the autonomous ship approaches a confined environment near a harbour, with multiple dynamic obstacles. Furthermore, the tuning is made more difficult by considering multiple trajectory scenarios for nearby dynamic obstacles, which should in theory have associated uncertainties covering the entire maneuvering space for each obstacle. As the formulation of the PSB-MPC have non-trivial cost function terms and complex predictions to setup, future work is needed to investigate whether or not there exist easier and more comprehensible formulations to easier incorporate probabilistic risk assessment, traffic rules compliance and intent information. The development of an adaptive version of the PSB-MPC is also needed, which can adjust its behaviour based the changing environment. This is a future research direction which can also alleviate some tuning issues for COLAV. Another approach is to develop a supervisory control layer in the COLAV, with different tunings or versions applied in different settings.

Secondly, the ship collision avoidance system depends on multiple subsystems, including the dynamic obstacle management system, the autopilot GNC system and the grounding hazard management system. Ensuring a harmoneous interaction between these subsystems and the COLAV system is important to enable a fully functional and safe autonomous platform. The COLAV planning algorithm such as the PSB-MPC should be robust towards varying performance in these subsystems, and the CPE is here a way to enable this with respect to dynamic obstacles. However, work is needed to ensure this is also enabled with respect to also grounding hazards, where adequate sensors such as lidar and radar should be used to detect and estimate the location of unmapped or inaccurately mapped static hazards.

In terms of COLREGS adherence, most of the results in this thesis have demonstrated explicit compliance with rules 7-8 and 13-17. However, consideration of the ship types involved in collision situations is also needed, to comply with rule 18 dealing with responsibilities between vessels. This rule has mostly been ignored in research on maritime COLAV planning [31]. Also, consideration of rule 9 when voyaging in more narrow channels needs to be better addressed in future improvements on the PSB-MPC and other COLAV algorithms. Another point is that the developed method for joint vessel destination and kinematics prediction can be used to better enable COLAV systems to explicitly adhere to COLREGS rule 10 when following traffic separation schemes, as nearby dynamic vessels will then be more accurately predicted to follow these existing sea lanes.

Appendices

Appendix A

Attached Articles

Intention modelling and inference for autonomous collision avoidance at sea*

Sverre Velten Rothmund[a,b,*], Trym Tengesdal[a,b], Edmund Førland Brekke[a,b] and Tor Arne Johansen[a,b]

[a] *Center for Autonomous Marine Operations and Systems (NTNU AMOS), NTNU Norwegian University of Science and Technology, Trondheim, Norway*

[b] *Department of Engineering Cybernetics, NTNU Norwegian University of Science and Technology, O. S. Bragstads Plass 2D, Trondheim 7034, Norway*

ARTICLE INFO

Keywords:
Autonomous ship
COLREGs
Dynamic Bayesian Network (DBN)
Intention inference
Collision avoidance
Situational awareness

ABSTRACT

The open wording of the traffic rules of the sea, COLREGS, and the existence of unwritten rules, make it essential for an autonomous ship to understand the intentions of other ships. This article uses a dynamic Bayesian network (DBN) to model and infer the intentions of other ships in open waters based on their observed real-time behavior. Multiple intention nodes are included to describe the different ways a ship can interpret and conflict with the behavioral rules outlined in COLREGS. The prior probability distributions of the intention nodes are adapted to the current situation based on observable characteristics such as location and relative ship size. The resulting model is able to identify situations that are prone to cause misunderstandings and infer the state of multiple intention variables that describe how the ship is likely to behave. Different collision avoidance algorithms can use the resulting intention information to better know if, when, and how to act.

1. Introduction

When navigating at sea, understanding the intentions of other ships can be crucial for avoiding accidents (Chauvin, 2011). Blindly assuming that the other ship will follow the traffic rules put forth by the Convention on the International Regulations for Preventing Collisions at Sea (COLREGS)(IMO, 1972) is insufficient as shown in Chauvin and Lardjane (2008). They demonstrated the existence of local unwritten rules and agreements between captains that went contrary to the rules specified by COLREGS. Furthermore, COLREGS is open to disagreements making it unsafe to act only based on your own interpretation of the situation (Clawson Jr, 2013; Woerner, Benjamin, Novitzky and Leonard, 2019). For an autonomous ship to safely operate in these conditions, it is essential that the ship can pick up on the intentions of other ships.

A large variety of ship collision avoidance algorithms exists in the literature (Huang, Chen, Chen, Negenborn and van Gelder, 2020; Vagale, Oucheikh, Bye, Osen and Fossen, 2021). Most algorithms that consider COLREGS handle ships that do not fulfill the traffic rules by executing reactive evasive actions when the ships get close enough. In Eriksen, Bitar, Breivik and Lekkas (2020) this is handled by having a separate short-term controller, in addition to their COLREGS compliant controller, which disregards COLREGS when the ships are close enough. A different approach is taken in Johansen, Perez and Cristofaro (2016) where they have a separate collision risk and COLREGS compliance penalties. The collision risk penalty increases when the ships get closer, ensuring that an evasive action will be taken even if it conflicts with the main COLREGS rules.

A different approach is taken in Tengesdal, Johansen and Brekke (2020) where they instead simulate multiple possible future trajectories the other ships can follow. The probabilities of the different trajectories are based upon the likelihood of the other ships having different intentions, such as being COLREGS compliant. This enables the collision avoidance algorithm to take early and substantial actions if the intentions are uncertain or if it becomes apparent that the

* The work was sponsored by the Research Council of Norway [223254, 274441, 295033]

* Corresponding author

sverre.v.rothmund@ntnu.no (S.V. Rothmund); trym.tengesdal@ntnu.no (T. Tengesdal); edmund.brekke@ntnu.no (E.F. Brekke); tor.arne.johansen@ntnu.no (T.A. Johansen)

ORCID(s): 0000-0002-7659-7881 (S.V. Rothmund); 0000-0001-8182-3292 (T. Tengesdal); 0000-0001-8735-1687 (E.F. Brekke); 0000-0001-9440-5989 (T.A. Johansen)

other ship does not act according to the rules. However, Tengesdal et al. (2020) does not consider how these intentions can be identified.

Different methods exist for identifying the intentions of other ships (Du, Goerlandt, Valdez Banda, Huang, Wen and Kujala, 2020; Woerner et al., 2019; Cho, Kim and Kim, 2021). Du et al. (2020) presents a method to identify whether the give-way ship is doing an evasive action or not. This enables the ship to comply with CORLEGS rule 17, which states that stand-on ships should act if the give-way ship is not taking appropriate action. Cho et al. (2021) presents a Bayesian model that evaluates the probability that the other ship follows its obligations as specified by COLREGS rules 14 to 17 based on its observed motion. Woerner et al. (2019) develop a scoring system to evaluate to what degree ships follow COLREGS rules 7, 8, and 13-17. This method is designed to evaluate different collision avoidance algorithms but can also be used online to evaluate how well other ships are acting in accordance with the rules.

These articles (Du et al., 2020; Woerner et al., 2019; Cho et al., 2021) evaluate whether the other ship is acting as expected based on the own-ships interpretation of the situation. They do not model the underlying causes making the ship not act as expected. These underlying causes could, for example, be a disagreement of the situation or one of the ships having priority over the other.

Works on intention modeling exist for air traffic (Krozel and Andrisani, 2006; Yepes, Hwang and Rotea, 2007), road traffic (Hardy and Campbell, 2013), and for robot pedestrian interactions (Chen, Zhao and Lou, 2021; Hashimoto, Gu, Hsu and Kamijo, 2015). These works show different ways of inferring the goal, behavior, or trajectories of the other agents in the encounter. Only Hashimoto et al. (2015) consider underlying causes that affect how an agent acts. They use information on whether a pedestrian is alone or in a group to affect the prior probability that it will hurry over at a flashing green light.

The present article uses a dynamic Bayesian network (DBN) to model and infer the intentions of other ships in open waters. Different intention variables are defined based on the different ways ships can interpret and conflict with the behavioral rules specified by COLREGS. The DBN combines these intention variables with a model based on COLREGS Rule 7, 8, 11, and 13 to 18 to define the possible ways the ship can act. A ship's intentions are gradually inferred by ruling out all possible combinations of intention states that contradict the observed course and speed. This way of modeling ensures that the intention probabilities are independent of how often the model is updated with new observations.

The contribution of this article and the novelty compared to earlier literature is a modeling framework that considers how underlying causes affect a ship's behavior and which can infer the state of multiple different intention variables based on measured properties. Modeling the underlying causes enables the model to identify situations that can cause misunderstandings, making it possible to take early actions to avoid a potentially dangerous situation. Furthermore, it enables the model to adapt to the current situation by letting additional information, such as relative ship size and location, affect the intentions. Being able to infer the state of multiple intention nodes enables the model to describe the future motion of other ships with higher fidelity than simply being COLREGS compliant or not. The resulting intention probabilities can be used for collision avoidance with algorithms that explicitly consider the intentions (Tengesdal et al., 2020) or as decision criteria replacing the static distance used to decide when to always act to avoid collision (Eriksen et al., 2020).

The rest of the article is structured as follows. Section 2 give background information on Bayesian networks. Section 3 presents the proposed DBN which is demonstrated in section 4. The results are discussed in section 5 and a conclusion is given in section 6.

2. Background

Bayesian belief networks (BBN) are directed acyclic graphs (DAG) that model probabilistic relations. These networks consist of nodes that can be in a discrete set of states and arcs that define dependencies between nodes. An arc points from a parent node to a child node. Conditional probability tables (CPT) are supplied for all nodes and define the probability of the node being in a particular state as a function of the states of its parent nodes. If the nodes do not have a parent node, then the CPT defines the prior distribution of that node.

The Bayesian probability law is used to evaluate a node's probability distribution, given some evidence. Evidence is the set of information about the state of one or more nodes. If this information is uncertain, then virtual evidence can be used. Virtual evidence specifies the probability of observing this particular observation, given the state of the node. A state unlikely to result in the observation will be given a low probability, while one likely to result in the

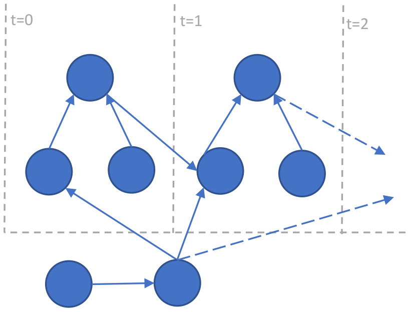


Figure 1: Example of dynamic Bayesian network (DBN) consisting of three time-dependent nodes and two time-independent nodes. Two time steps are shown.

observation will be given a high probability. A thorough explanation of virtual evidence can be found in Ben Mrad, Delcroix, Maalej, Piechowiak and Abid (2012).

BBNs can be made dynamic by repeating some or all of the nodes for each time step. Fig. 1 shows an example of the resulting DBN. DBNs make it possible to model how a system develops over time. The DBN can consist of time-independent nodes as well as time-dependent nodes.

Software libraries such as BAYESFUSION LLC (b) include different general solvers for evaluating DBNs and natively support the use of virtual evidence. More information on BBNs and DBNs can be found in Fenton and Neil (2018) and Russell and Norvig (2014).

3. Method

This section presents a DBN used to model and infer the intention of meeting ships. The term intentions will be used for a ship's internal states that we wish to infer such that we can understand how the ship will act. Examples of different intention variables are what the ship considers to be a safe distance, what priority it thinks it has relative to the other ships, and what it thinks that the COLREGS situation is.

The DBN model takes the perspective of a single ship, which will be called the reference ship, and models its relation to all other ships in the area. The index *i* will be used to identify the other ships in the area. To model multiple ships, the model must be repeated for each ship. How to make inference using the model is described in section 3.1.

Each of the intention variables are modeled as nodes in the DBN. These nodes are stochastic variables as the intention is unknown. The intention nodes are modeled as time-independent nodes as it is assumed that the intentions do not change within one encounter. The prior distribution of the intention nodes describes how often the different intentions are encountered in situations similar to this one. How these priors are designed is described in more detail in section 3.4.

The intentions are updated based on different measured properties that can be evaluated based on the relative position between the ships, theirs course, and their speed. The different measured properties are given in Table 3. A tracking system is assumed to be used to evaluate the ships course, speed, and position. The tracking system is assumed to give high quality tracks, such that the intention module does not need to account for measurement uncertainty.

The DBN evaluates the probability that a particular combination of measurements and intention node states are compatible. Which combinations that are compatible are defined by COLREGS and are described in section 3.2 using logic statements. How these can be translated into CPTs is described in section 3.3. The resulting DBN is shown in Fig. 5.

When a new observation is made, the different measured properties are inserted as evidence on the measurement nodes in a new time-step of the DBN. These measurement nodes are time-dependent, thereby enabling the system to combine information over time. The network can be used to evaluate the probability that the observation is compatible

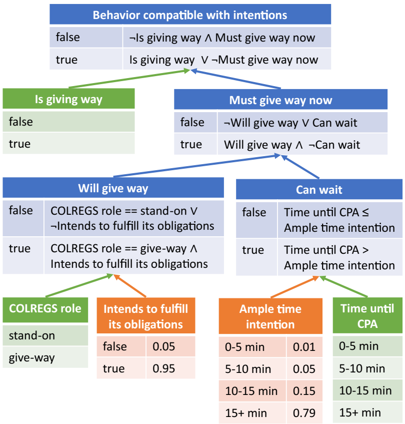

Behavior compatible with intentions
false
¬Is giving way ∧ Must give way now
true
Is giving way ∨ ¬Must give way now
Is giving way
false
true
Must give way now
false
¬Will give way ∨ Can wait
true
Will give way ∧ ¬Can wait
Will give way
false
COLREGS role == stand-on ∨ ¬Intends to fulfill its obligations
true
COLREGS role == give-way ∧ Intends to fulfill its obligations
Can wait
false
Time until CPA ≤ Ample time intention
true
Time until CPA > Ample time intention
COLREGS role
stand-on
give-way
Intends to fulfill its obligations
false 0.05
true 0.95
Ample time intention
0-5 min 0.01
5-10 min 0.05
10-15 min 0.15
15+ min 0.79
Time until CPA
0-5 min
5-10 min
10-15 min
15+ min

Figure 2: A simplified example network used to illustrate the proposed inference method. Measurement nodes are shown in green, intention nodes in orange, and modelling nodes in blue. The initial probability distribution is shown for the intention nodes.

with the prior distribution of the intention nodes. The distribution of the intention nodes can be updated by eliminating all combinations of intentions that contradict the observation. This is achieved by inserting evidence in the network stating that intentions and observed measurements are, in fact, compatible. The updated posterior distribution of the intention nodes can be used to give an updated prediction on how the reference ship will act. Two different ways of using the updated intention probability distributions for collision avoidance are outlined in section 3.5.

Modeling whether a particular combination of observations and intention node states are compatible enables the system to gradually infer the reference ship's intentions without considering how often observations are given to the system. Giving the exact same observation multiple times to the system will not affect the probability distribution of the intention nodes, as the first observation has already eliminated all combinations of intentions that would be eliminated by the second observation.

A simplified example can illustrate this procedure. Fig. 2 shows a simplified network that only considers when the ship will act. COLREGS rule 8(a) states that a ship should act in "ample time". Two intention nodes are then needed, one modeling the reference ship's definition of ample time and the other modeling whether the reference ship intends to follow this rule. When an observation is made, the following evidence is inserted: time until closest point of approach (CPA), which role the reference ship has according to COLREGS, and whether the reference ship is giving way. In this example, the observation is only compatible with the intention of the reference ship if either of the following is true: it is giving way, it has a stand-on role, if it does not intend to follow the rules, or if the time until CPA is longer than the reference ships definition of ample time.

The intention probabilities can be updated to reflect the observation by inserting evidence on the "Behavior compatible with intentions" node stating that it must be in the "true" state. If it, for example, is observed that the time until CPA is 10 minutes, the reference ship has a give-way role, and it is not giving way, then the model can exclude the possibility that the ship intends to follow its obligation to give way while at the same time considers ample time to be more than 10 minutes. It is left with the possibility that it will not follow its obligations at all or that it considers ample time to be shorter than 10 minutes. For this example, the updated probability that the reference ship does not intend to fulfill its obligations evaluates to 47%. This is due to the prior likelihood that the reference ship will give way at a short distance $(0.01 + 0.05 = 0.06)$ is similar to the prior likelihood that it will not fulfill its obligations

Table 1
Abbreviations

Abbreviation	Description
CPA	Closest point of approach
SO	Stand-on
GW	Give-way
HO	Head-on
OT_ing	Overtaking
OT_en	Overtaken
CR_SS	Crossing with other ship on starboard side
CR_PS	Crossing with other ship on port side

Table 2
Intention variables

Symbol	Description	States
$\mathcal{I}_{AT}$	What time until CPA the reference ship considers ample time	real valued
$\mathcal{I}_{C}$	Whether the reference ship acts according to COLREGS	binary
$\mathcal{I}_{CS_i}$	What COLREGS situation the reference ship thinks it has towards ship i	“OT_ing"/“OT_en"/ “HO"/“CR_PS"/ “CR_SS"
$\mathcal{I}_{GS}$	Whether the reference ship acts according to good seamanship	binary
$\mathcal{I}_{P_i}$	Relative priority the reference ship has to ship i	“higher"/“similar"/ “lower"
$\mathcal{I}_{RC}$	What distance at CPA the reference ship considers a risk of collision	real valued
$\mathcal{I}_{RCF}$	How far in front of a ship the reference ship considers a crossing as risky	real valued
$\mathcal{I}_{SD}$	What the reference ship considers a safe distance at CPA	real valued
$\mathcal{I}_{SDF}$	How far in front of a ship the reference ship considers a crossing as safe	real valued
$\mathcal{I}_{SDM}$	What the reference ship considers a safe distance at CPA to the current midpoint (See 3.2.12).	real valued
$\mathcal{I}_{SS}$	At what distance the reference ship consider that the situation starts	real valued
$\mathcal{I}_{U}$	Whether the reference ship acts in an unmodelled way	binary

(0.05). This simplified example is unable to model the underlying causes that influence how a ship will act. The rest of this section handles this by considering many more of the COLREGS rules.

3.1. Basic procedure

For every new observation:

1. Insert information from observed position, course, and speed as evidence on the measurement nodes
2. Insert evidence stating that the compatible to all node (C) is in the state true.
3. Evaluate the updated probabilities for the different intention states
4. Expand the network with a new time-step

3.2. Intention model logic

This section presents a series of logic statements that define which combinations of intentions and observations that are compatible. These statements are based on the behavioral rules specified by COLREGS Rule 7, 8, 11, and 13

Table 3
Measurement variables. The values are evaluated based on the measured position, speed, and course of the different ships in an encounter. Measurements that cannot be directly evaluated based on the position, speed, or course are described when the measurement is first used in section 3.2.

Symbol	Description	States
$\mathcal{M}_C[t]$	Current course of the reference ship	real valued
$\mathcal{M}_S[t]$	Current speed of the reference ship	real valued
$\mathcal{M}_{CS_i}[t]$	Current COLREGS situation reference ship has towards ship i (See 3.2.15)	"OT_ing"/"OT_en"/ "HO"/"CR_PS"/ "CR_SS"
$\mathcal{M}_{D_i}[t]$	Current distance between the reference ship and ship i	real valued
$\mathcal{M}_{DCPA_i}[t]$	Distance between reference ship and ship i at CPA assuming both will keep their current course and speed	real valued
$\mathcal{M}_{DF_i}[t]$	How far the reference ship crosses in front on ship i assuming both keep their current course and speed. This value is set to ∞ if the ship does not cross in front of ship i	real valued
$\mathcal{M}_{DM_i}[t]$	Distance at CPA to the current midpoint between the reference ship and ship i, assuming constant course and speed for the reference ship. (See 3.2.12)	real valued
$\mathcal{M}_{P_i}[t]$	Whether reference ship has passed ship i. (See 3.2.6)	binary
$\mathcal{M}_{TCPA_i}[t]$	Time until CPA between reference ships and ship i assuming both will keep their current course and speed	real valued
$\mathcal{M}_{AF_i}[t]$	Whether the reference ship will pass aft or in front of ship i assuming both keep their current course and speed. (See 3.2.13)	"Aft"/"Front"

to 18. Rules regarding traffic separation schemes (Rule 10), narrow channels (Rule 9), and sailing vessels (Rule 12) are not considered in this article.

The section is structured following a top-down approach where the statement describing the most general model variable is presented first. Model variables that are used in more general statements are then gradually introduced. The different model variables are given in Table 4, intention variables in Table 2, measurement variables in Table 3, and parameters in Table 5. See Table 1 for abbreviations used in this model.

3.2.1. $C[t]$ - Compatible to all

An observation is compatible with the intention states of the reference ship at time step t if it is compatible towards all ships in the area at that time step. The area considered must be large enough to encompass all ships that potentially affect how the reference ship acts. All observations are also considered compatible if the ship intends to act in an unmodelled manner ($\mathcal{I}_U$). This state works as a catch-all for behavior that does not fit the behavioral rules described in this section. Mathematically, this is expressed through the following logical clause:

$$C[t] = \left(\wedge_{i=1}^{n} C_i[t]\right) \vee \mathcal{I}_U \tag{1}$$

3.2.2. $C_i[t]$ - Compatible towards ship i

An observation is compatible with the intention states of the reference ship towards ship i if either of the following is true:

- The collision avoidance situation has not started yet (SS_i).

Table 4
Model variables

Symbol	Description	States
$C[t]$	Observation compatible with the intentions of the reference ship	binary
$C_i[t]$	Observations and intentions compatible towards ship i	binary
$CEM_i[t]$	Correct evasive maneuver towards ship i	binary
$C\_CR\_SS_i[t]$	Correct crossing evasive maneuver with ship i on the starboard side	binary
$C\_CR\_PS_i[t]$	Correct crossing evasive maneuver with ship i on the port side	binary
$C\_HO_i[t]$	Correct head-on evasive maneuver towards ship i	binary
$C\_OT_i[t]$	Correct overtaking evasive maneuver towards ship i	binary
$CIC_i[t]$	Change in course towards ship i	“starboard"/ “straight"/“port"
$CIS_i[t]$	Change in speed towards ship i	“higher"/ “none"/“lower"
$GS_i[t]$	Good seamanship towards ship i	binary
$GWC_i[t]$	Gives way correctly towards ship i	binary
$IC_i[t]$	Initial course when the situation started towards ship i. Course is given in the NED frame.	real valued
$IS_i[t]$	Initial speed when the situation started towards ship i	real valued
$P_i[t]$	Has passed ship i safely	binary
$PA_i[t]$	There has been a port action towards ship i	binary
R_i	Role towards ship i	“GW"/“SO"
$RC_i[t]$	There is currently a risk of collision with ship i	binary
$RS_i[t]$	It is a risky situation towards ship i	binary
$SOC_i[t]$	Stands on correctly towards ship i	binary
$SD_i[t]$	The reference ship will cross at a safe distance towards ship i	binary
$SS_i[t]$	Situation has started towards ship i	binary
$SA_i[t]$	There has been a starboard action towards ship i	binary
WGW_i	Will give way towards ship i	binary

- The ships have passed each other safely (P_i)
- The ships will pass each other in such a manner that it is not a risky situation (RS_i).
- If the reference ship has a give-way role (R_i) and gives way correctly (GWC_i) towards ship i.
- If the reference ship has a stand-on role (R_i) and stands on correctly (SOC_i) towards ship i.

$$
\begin{aligned}
C_i[t] =& \neg SS_i[t] \lor P_i[t] \lor \neg RS_i[t] \\
& \lor \Big(\big(R_i == \text{“GW"}\big) \land GWC_i[t] \Big) \\
& \lor \Big(\big(R_i == \text{“SO"}\big) \land SOC_i[t] \Big)
\end{aligned} \tag{2}
$$

Table 5
Example parameters chosen for demonstrative purposes. The parameters can be modified based on properties of the current situation, such as ship size, speed, and weather. The minimal acceptable definition of ample time (AT_{min}), safe distance at CPA (SD_{min}), safe distance front (SDF_{min}) and safe distance to the current midpoint (SDM_{min}), should be based on the maneuverability of the own-ship and how risk averse the operation should be.

Symbol	Description	Value
$\mathcal{P}_{CIC}$	Max change in course that is considered as keeping the course	$10°$
$\mathcal{P}_{CIS}$	Max change in speed that is considered as keeping the speed	2 m/s
AT_{min}	Ownships minimal accepted definition of ample time	60 s
SD_{min}	Ownships minimal accepted definition of safe distance at CPA	75 m
SDF_{min}	Ownships Minimal accepted definition of safe distance to cross in front	100 m
SDM_{min}	Ownships minimal accepted definition of safe distance to midpoint	75 m

3.2.3. $SS_i[t]$ - *Situation started*

According to COLREGS Rule 11, the behavioral rules only apply for ships in sight of each other. COLREGS Rule 3 specifies that a ship is in sight if it can be seen visually. At what distance the reference ship sees ship *i* is unknown and modeled with the intention variable $\mathcal{I}_{SS}$. The situation starts whenever the distance between the ships ($\mathcal{M}_{D_i}$) is shorter than the situation start intention. Map data can be used to evaluate at which distance the ships are likely to see each other.

$$SS_i[t] = SS_i[t-1] \lor \left(\mathcal{M}_{D_i}[t] < \mathcal{I}_{SS}\right) \tag{3}$$

3.2.4. $RS_i[t]$ - *Risky situation*

If there is a risk of collision (RC_i) at one point of time after the situation starts (SS_i), then the situation should be considered as risky.

$$RS_i[t] = \begin{cases} \text{“false"} & if \neg SS_i[t] \\ RC_i[t] \lor RS_i[t-1] & otherwise \end{cases} \tag{4}$$

3.2.5. $RC_i[t]$ - *Risk of collision*

Actions to avoid collision are only needed if the reference ship considers that there is a risk of collision (COLREGS Rule 7, 12, and 14). According to COLREGS Rule 7(i), a risk of collision exists if the compass bearing from the reference ship to ship *i* “does not appreciably change" (IMO, 1972). How much change that is sufficient would depend on the distance between the ships, as one would experience a quicker bearing change once the ships get closer. To simplify this requirement, the expected crossing distance is used to evaluate whether there is a risk of collision. The acceptable distance when crossing in front can be larger than what is acceptable to the side of the ship. This is handled by defining two different intention variables, one specifying how far in front of a ship the reference ship considers it risky to cross ($\mathcal{I}_{RCF}$) and one specifying the distance at CPA that is considered risky ($\mathcal{I}_{RC}$). These are compared to the expected crossing distance in front ($\mathcal{M}_{DF_i}$) and at CPA ($\mathcal{M}_{DCPA_i}$) assuming both ships keep their current course and speed.

$$RC_i[t] = \left(\mathcal{M}_{DCPA_i}[t] < \mathcal{I}_{RC}\right) \lor \left(\mathcal{M}_{DF_i}[t] < \mathcal{I}_{RCF}\right) \tag{5}$$

3.2.6. $P_i[t]$ - *Safely passed*

If the reference ship has passed ship *i* ($\mathcal{M}_{P_i}$) and is at a safe distance (SD_i), then the reference ship does not need to consider the ship any longer. A ship is considered as passed if the time until closest point of approach, assuming constant course and speed for all ships, is negative.

$$P_i[t] = \mathcal{M}_{P_i}[t] \wedge SD_i[t] \tag{6}$$

3.2.7. $SOC_i[t]$ - Stands on correctly

The reference ship stands on correctly towards ship i if it does not change its course (CIC_i) or speed (CIS_i), or if it does a correct evasive maneuver (CEM_j) towards another ship (j) it has a give-way role (R_j) for (Rule 17).

$$\begin{aligned} SOC_i[t] = &\Big(\big(CIC_i[t] == \text{“straight"}\big) \wedge \big(CIS_i[t] == \text{“none"}\big) \Big) \\ &\vee_{j=1}^{n} \Big(\big(R_j == \text{“GW"}\big) \wedge CEM_j[t] \Big) \end{aligned} \tag{7}$$

3.2.8. $GWC_i[t]$ - Gives way correctly

The reference ship gives way correctly towards ship i if it is executing a correct evasive maneuver CEM_i. According to COLREGS Rule 8, the ship must take evasive actions in what it considers “ample time" ($\mathcal{I}_{AT}$). The “time" in ample time is measured as the time until CPA assuming both ships keep their current course and speed ($\mathcal{M}_{TCPA_i}$). How long before CPA the reference ship consider as “ample time" is modeled with the intention variable $\mathcal{I}_{AT}$. The ship is allowed to stand on correct (SOC_i) before what it considers “ample time".

$$GWC_i[t] = CEM_i[t] \vee \Big(\big(\mathcal{M}_{TCPA_i}[t] > \mathcal{I}_{AT}\big) \wedge SOC_i[t] \Big) \tag{8}$$

3.2.9. $CEM_i[t]$ - Correct evasive maneuver

For an evasive maneuver to be correct, it must comply with “good seamanship" (GS_i) (COLREGS Rule 8) if the reference ship has an intention to act with “good seamanship" ($\mathcal{I}_{GS}$). Additionally, the maneuver must fulfill the requirements specified by COLREGS if the reference ship has an intention to follow COLREGS ($\mathcal{I}_C$). COLREGS specify a set of situations and how to act in each scenario. These consist of overtaking (OT_ing, Rule 13) another vessel, being overtaken (OT_en, Rule 17), head-on (HO, Rule 14), crossing with the other ship on the starboard side (CR_SS, Rule 15), and crossing with the other ship on the port side (CR_PS, Rule 17). What COLREGS situation the reference ship believes it has towards ship i is denoted as $\mathcal{I}_{CS_i}$.

$$\begin{aligned} CEM_i[t] = &\big(\neg\mathcal{I}_{GS} \vee GS_i[t]\big) \wedge \Big(\neg\mathcal{I}_C \vee \Big(\big((\mathcal{I}_{CS_i} == \text{“OT\_ing"}) \\ &\vee (\mathcal{I}_{CS_i} == \text{“OT\_en"})\big) \wedge C\_OT_i[t] \Big) \\ &\vee \Big(\big(\mathcal{I}_{CS_i} == \text{“HO"}\big) \wedge C\_HO_i[t] \Big) \\ &\vee \Big(\big(\mathcal{I}_{CS_i} == \text{“CR\_SS"}\big) \wedge C\_CR\_SS_i[t] \Big) \\ &\vee \Big(\big(\mathcal{I}_{CS_i} == \text{“CR\_PS"}\big) \wedge C\_CR\_PS_i[t] \Big) \Big) \end{aligned} \tag{9}$$

3.2.10. $SD_i[t]$ - Safe distance

According to COLREGS Rule 8, actions to avoid collision shall result in the ships passing at a safe distance. Whether the reference ship and ship i will pass at a safe distance is evaluated by assuming that both ships will keep their current course and speed. This assumption holds for ship i if it has a stand-on role, as stand-on ships are required to keep their course and speed (COLREGS Rule 17). If the reference ship has a give-way role, then it is expected to mark its intent by substantially changing its course or speed (COLREGS Rule 8) before returning to the initial course. Assuming that it will keep its course and speed should result in passing at a safe distance if the ship has started to act to avoid a collision. As with risk of collision ($RC_i[t]$), different intention and measurement nodes are included for a safe crossing distance in front ($\mathcal{M}_{DF_i}$, $\mathcal{I}_{SDF}$) and at CPA ($\mathcal{M}_{DCPA_i}$, $\mathcal{I}_{SD}$).

$$SD_i[t] = \big(\mathcal{M}_{DCPA_i}[t] > \mathcal{I}_{SD}\big) \wedge \big(\mathcal{M}_{DF_i}[t] > \mathcal{I}_{SDF}\big) \tag{10}$$

3.2.11. $C\_OT_i[t]$ - Correct overtaking evasive maneuver

COLREGS Rule 13 specifies that the overtaking vessel shall keep out of the way of the vessel being overtaken. Checking that the ships are crossing at a safe distance (SD_i) is therefore sufficient.

$$C\_OT_i[t] = SD_i[t] \tag{11}$$

3.2.12. $C\_HO_i[t]$ - Correct head-on evasive maneuver

For head-on situations, COLREGS Rule 14 specifies that the ships must make a starboard turn such that they pass each other port to port. As both ships have to give way in this situation, assuming that ship *i* will keep its current course is unrealistic. Instead, a new measurement is used that considers the distance at CPA to the current midpoint between the ships ($\mathcal{M}_{DM_i}$). As the current midpoint does not change when the ships courses change, considering a safe distance to the current midpoint thereby requires that the reference ship has to do an evasive maneuver even though ship *i* has already changed its course. The distance at CPA to the current midpoint is evaluated assuming the reference ship will keep its current course and speed. The distance to the midpoint is set to 0 if the ship passes with the midpoint on the starboard side. This ensures that the ship has to pass on the correct side. Which distance to the midpoint the reference ship considers as safe is denoted as $\mathcal{I}_{SDM}$.

$$C\_HO_i[t] = \left(\mathcal{M}_{DM_i}[t] > \mathcal{I}_{SDM}\right) \tag{12}$$

3.2.13. $C\_CR\_SS_i[t]$ - Correct crossing starboard-side evasive maneuver

In a crossing situation, Rule 15 of COLREGS specifies that a ship should, in addition to cross at a safe distance (SD_i), avoid crossing in front of another ship it has on its starboard side. Whether the reference ship crosses aft or front of ship *i* ($\mathcal{M}_{AF_i}$) is evaluated by first finding the intersection point of the paths followed by the ships assuming that they keep their current course. Which ship that first arrives at this point crosses in front of the other.

$$C\_CR\_SS_i[t] = \left(\mathcal{M}_{AF_i}[t] == \text{“aft"}\right) \wedge SD_i[t] \tag{13}$$

3.2.14. $C\_CR\_PS_i[t]$ - Correct crossing port-side evasive maneuver

If a ship with the other on its port side is forced to take action, then COLREGS Rule 17(c) specifies that it, in addition to cross at a safe distance (SD_i), should avoid changing its course (CIC_i) towards port.

$$C\_CR\_PS_i[t] = \left(CIC_i[t] \neq \text{“port"}\right) \wedge SD_i[t] \tag{14}$$

3.2.15. $\mathcal{M}_{CS_i}[t]$ - COLREGS situation

According to COLREGS Rule 13(b), a ship is overtaking another when it is coming up on the ship “from a direction more than 22.5 degrees abaft her beam" (IMO, 1972). Uncertainty in the heading of the other ship can lead to different interpretations of the situation. Uncertainty in whether it is an overtaking situation is modeled by using the classifier as shown in Fig. 3. The size of the uncertainty region can be based on a combination of historical data and expert opinion. Different situations could be presented to different experienced captains where they could express their trust that other ships would identify this situation correctly. The values used in this article are chosen for demonstrative purposes.

A head-on situation is defined by COLREGS Rule 14(a) to be when two vessels are meeting on “nearly reciprocal courses", while Rule 14 (b) specifies when a head-on situation exists based on the visibility of different lights of the other ship. This opens up for disagreements from different definitions of “nearly reciprocal" and how the ships observe each other. With the presence of current and winds, a ship observing the course of the other by radar or AIS might come to a different conclusion than one observing the heading of the other ship based on the visibility of lights (Woerner et al., 2019). Furthermore, measurement uncertainties in the course of the other ship can lead to misunderstandings. The classifier shown in Fig. 4 is used to accommodate this uncertainty. Identifying the uncertainty and mean of which angle a head-on situation starts can be evaluated similarly to the overtaking case. In addition, the mean can be chosen based on case law and certifying agency requirements as proposed in Woerner et al. (2019).

The probability that the reference ship evaluates the current situation as an overtaking or head-on situation is based on the two classifiers given in Fig. 3 and Fig. 4. The remaining probability gives the probability that the reference ship

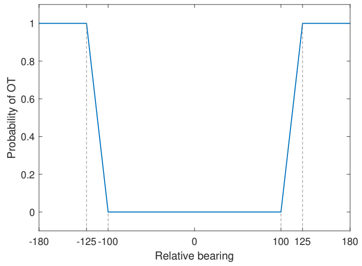


Figure 3: Classifier giving the probability that it is an overtaking situation. Relative bearing is defined as the bearing from the ship being overtaken to the overtaking ship relative to the heading of the ship being overtaken. 22.5° abaft the beam as specified in COLREGS Rule 13 is the same as ±112.5° relative to the heading. This classifier considers a 15° uncertainty in the situation.

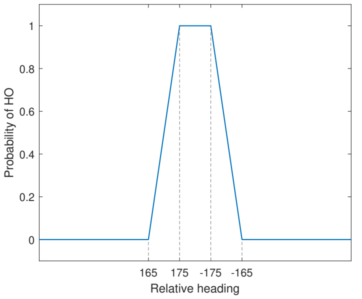


Figure 4: Classifier giving the probability that it is a head-on situation. The relative heading between the two ships defined the probability. This classifier considers a 10° uncertainty in the situation.

evaluates the situation to be a crossing situation. Whether the reference ship is in front or back of the other ship when the situation starts defines whether it is overtaking (“OT_ing) or being overtaken (“OT_en"). Whether the other ship is on the port or starboard side defines whether it is a crossing port side (“CR_PS") or crossing starboard side (“CR_SS") situation. This information is inserted as virtual evidence on the measured COLREGS situation node, $\mathcal{M}_{CS_i}$.

According to COLREGS Rule 13(d), subsequent alterations in bearing do not change the situation. The situation is therefore defined when the situation starts, which can lead to misunderstandings as the different ships may define that the situation starts at different time points (Clawson Jr, 2013). To model the uncertainty caused by when the reference ship thinks that the situation starts, a situation measurement node ($\mathcal{M}_{CS_i}$) is introduced. The state of this node is equal to the state of the situation intention node ($\mathcal{I}_{CS_i}$) only at the time-step where the reference ship thinks that the situation starts. At all other time-steps, the probability of measuring the different states of the measurement node is unaffected by the state of the intention node. Which time-step the reference ship thinks that the situation starts is uncertain, making it uncertain which measurement that defines the intention state. There should be an equal probability of measuring all states when the measurement node is independent of the intention node.

$$\mathcal{M}_{CS_i}[t] = \begin{cases} \mathcal{I}_{CS_i} & \text{if } SS_i[t] \wedge \neg SS_i[t-1] \\ [0.2, 0.2, 0.2, 0.2, 0.2] & \text{otherwise} \end{cases} \tag{15}$$

3.2.16. R_i - Role

A ship must give way if it has lower priority ($\mathcal{I}_{P_i}$), either as specified in COLREGS Rule 18 or due to unwritten rules (Chauvin and Lardjane, 2008). If the ship has higher priority, it must stand on. If the priority is similar, then the role is given by Rule 13 to 15. In a head-on situation, both ships must give way (Rule 14). In an overtaking situation, the overtaking vessel must give way (Rule 13). In a crossing situation, the one with the other ship on its starboard side must give way (Rule 15).

$$R_i = \begin{cases} \text{“GW"} & \begin{aligned} &\text{if } \left(\mathcal{I}_{P_i} == \text{“lower"}\right) \vee \Big(\left(\mathcal{I}_{P_i} == \text{“similar"}\right) \\ &\wedge \left((\mathcal{I}_{CS_i} == \text{“HO"}) \vee (\mathcal{I}_{CS_i} == \text{“CR\_SS"}) \right. \\ &\left. \vee (\mathcal{I}_{CS_i} == \text{“OT\_ing"}) \right) \Big) \end{aligned} \\ \text{“SO"} & \text{otherwise} \end{cases} \tag{16}$$

3.2.17. $GS_i[t]$ - Good seamanship

Good seamanship is difficult to define and can contain many different behaviors. In this article, good seamanship restricts the ship from changing which side it turns towards to avoid collision. The ship is not allowed to have made both a starboard action (SA) and a port action (PA) during a collision encounter.

$$GS_i[t] = \neg\left(SA_i[t] \wedge PA_i[t]\right) \tag{17}$$

$$SA_i[t] = \begin{cases} \text{“false"} & \text{if} \neg SS_i[t] \\ \left(CIC_i[t] == \text{“starboard"}\right) \vee SA_i[t-1] & \text{otherwise} \end{cases} \tag{18}$$

$$PA_i[t] = \begin{cases} \text{“false"} & \text{if} \neg SS_i[t] \\ \left(CIC_i[t] == \text{“port"}\right) \vee PA_i[t-1] & \text{otherwise} \end{cases} \tag{19}$$

3.2.18. $CIC_i[t]$ - Change in course

A change in course is evaluated by comparing the initial course (IC_i) with the measured course ($\mathcal{M}_C$). The initial course is saved when the situation starts (SS_i). If the change in course is less than $\mathcal{P}_{CIC}$ then it is considered as keeping the course. $\mathcal{P}_{CIC}$ should be chosen small enough to ensure that all intended course changes are marked as such, while being large enough to ensure that measurement uncertainty and small oscillations due to waves are not marked as a course change.

$$IC_i[t] = \begin{cases} \mathcal{M}_C[t] & \text{if} \neg SS_i[t] \\ IC_i[t-1] & \text{otherwise} \end{cases} \tag{20}$$

$$CIC_i[t] = \begin{cases} \text{“starboard"} & \text{if } \mathcal{M}_C[t] > \left(IC_i[t] + \mathcal{P}_{CIC}\right) \\ \text{“port"} & \text{if } \mathcal{M}_C[t] < \left(IC_i[t] - \mathcal{P}_{CIC}\right) \\ \text{“straight"} & \text{otherwise} \end{cases} \tag{21}$$

3.2.19. $CIS_i[t]$ - change in speed

The initial speed (IS_i) and change in speed are evaluated in the same manner as for the course. The same considerations should be made when choosing $\mathcal{P}_{CIS}$.

$$IS_i[t] = \begin{cases} \mathcal{M}_S[t] & \text{if} \neg SS_i[t] \\ IS_i[t-1] & \text{otherwise} \end{cases} \tag{22}$$

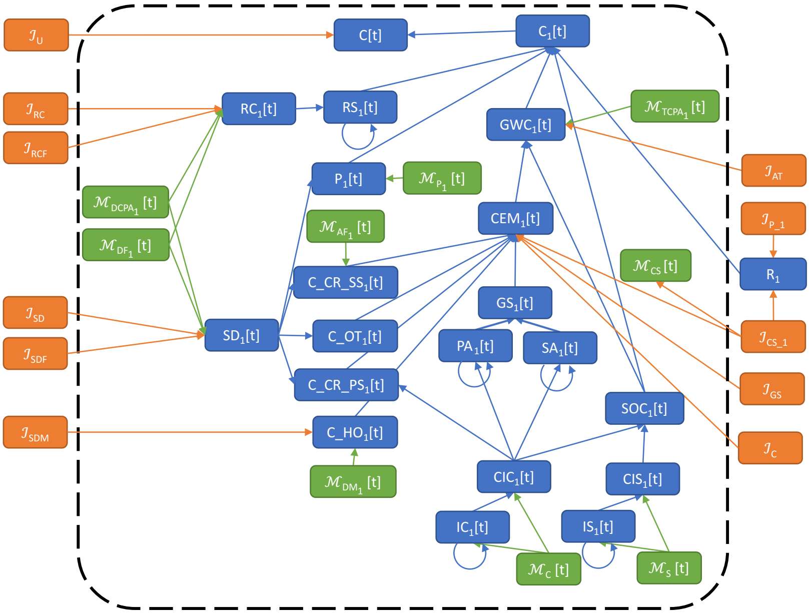


Figure 5: Figure showing the topography of the resulting DBN for a single ship encounter. Nodes related to situation start ($\mathcal{I}_{SS}$, $SS_i[t]$) are omitted to reduce complexity. See Table 2, Table 3, and Table 4 for abbreviations. Subscript 1 indicates that this model considers the relation between the reference ship and ship with index 1. In a multi-ship encounter, all nodes with index subscript would be repeated for any additional ship in the encounter. Green nodes represent measurements, orange node intentions, and blue nodes model variables. All nodes inside the box are time-dependent and are repeated for each time step. Circular arrows indicate connections between subsequent time steps.

$$CIS_i[t] = \begin{cases} \text{“higher"} & \textit{if}\ \mathcal{M}_S[t] > \left(IS_i[t] + \mathcal{P}_{CIS}\right) \\ \text{“lower"} & \textit{if}\ \mathcal{M}_S[t] < \left(IS_i[t] - \mathcal{P}_{CIS}\right) \\ \text{“none"} & \textit{otherwise} \end{cases} \tag{23}$$

3.3. Translation into DBN

A DBN is made from the logic statements given in section 3.2. A node is introduced for each intention variable, measurement variable, and model variable. Arcs are introduced based on the dependencies given by the equations in section 3.2. The resulting topology can be seen in Fig. 5.

The logical statements given in section 3.2 need to be translated into CPTs to be used by the DBN. This can be done by evaluating whether the output is “true" or “false" for all combinations of inputs. This results in CPTs consisting of 0/1 probabilities. Nodes that according to Table 2, Table 3, and Table 4 are real-valued must be discretized. A suitable range and discretization step must be defined. The software GeNIe (BAYESFUSION LLC, a) allows the user to specify equations and to use real-valued nodes. It can then automatically discretize and translate these equations into CPTs.

Table 6
Factors that can influence the intentions of the reference ship. Tab. 7 specifies and quantifies the dependencies.

Factor	Reason	States
Maneuverability	A poor maneuverability requires earlier actions and larger margins	low/medium/high
Location	Ships tend to act earlier and have larger margin in open seas than inland waterways	open sea/innland
Ship type	A leisure craft is less likely to know and follow rules and best practice	commercial/leisure
Relative ship size	Larger ships tends to have priority over smaller ships (Chauvin and Lardjane, 2008)	smaller/similar/larger
Speed	Ships require larger safety margins when going at a fast speed	slow/fast

3.4. Priors

Information from the current situation, such as ship types and the type of environment, can improve the prior distributions of the intention nodes. Examples of different factors that could be considered are shown in Table 6. These influencing factors can be included as time-independent nodes that affect the intention states.

Different approaches can be followed to identify factors that affect the intentions. One way is to have a workshop with experts in the field, such as experienced captains. This workshop can be similar to risk analysis workshops such as Hegde, Utne, Schjølberg and Thorkildsen (2018) and Rokseth, Utne and Vinnem (2017). Another option is to study captains during operation as done in Chauvin and Lardjane (2008). This has the advantage of being more correct than a workshop, but some factors might not show up during the study. A last option is to analyze historical data logged with the automatic identification system (AIS) that larger vessels are required to be equipped with (IMO). This method would be more general as much more data from different ships and situations could be analyzed. It will, however, be limited to the information that is logged with the AIS, which does not necessarily include all factors that could be of interest. A combination of the three approaches is preferable to maximize correctness and completeness.

The same methods can be used for quantifying how the intention nodes are affected by the identified factors. AIS data could be used to build prior distributions on, among others, how far before CPA different types of ships tend to give way and how close they tend to be at CPA. This information could be supplemented with data from operation studies and expert judgment to model how factors not included in the AIS affect the distribution. Different methods for building CPTs based on expert information are analyzed in Mkrtchyan, Podofillini and Dang (2016).

Performing a thorough identification and quantification is outside the scope of this article. Table 7 shows the quantification used to produce the results presented in section 4.

3.5. Using the intentions

This section presents two different ways of using the evaluated intention probabilities for collision avoidance.

3.5.1. Decision criteria

The first approach considers whether the own-ship should consider the reference-ship in the collision avoidance algorithm. Collision avoidance algorithms similar to Eriksen et al. (2020) do not need to consider the reference ship if the own-ship has a stand-on role, and the reference ship is planning to give way. A new node can be introduced into the network to evaluate whether the reference ship is planning to give way or not. A threshold can be proposed that defines how likely it must be that the reference ship will give way for it to be safely ignored by the collision avoidance algorithm.

The node representing whether the reference ship is planning to give way depends on whether the reference ship has a give-way role (R_i), and if its definitions of ample time ($\mathcal{I}_{AT}$), safe-distance at CPA ($\mathcal{I}_{SD}$), safe distance when crossing in front ($\mathcal{I}_{SDF}$), and safe distance to the current midpoint ($\mathcal{I}_{SDM}$) are acceptable. Additionally, the reference ship is assumed not to give way if it acts in an unmodelled manner ($\mathcal{I}_U$). Equation 24 shows the logic statement that defines whether the ship will give way towards ship i (WGW_i).

$$WGW_i = \left(R_i == \text{“GW"}\right) \land \left(\mathcal{I}_{AT} > AT_{min}\right)$$

Table 7
Prior probability distribution used in the simulation study for the different intention states as a function of the influencing factors. To keep the list short, factors are only included that were of relevance to the scenarios presented in section 4. States marked in bold are used unless otherwise specified. $\mathcal{N}(\mu, \sigma)$ indicates a truncated normal distribution with expected value μ, standard deviation σ, and limited to be larger than 0. For binary states the probability of "true" is given. Discrete states are given in the order specified in Table 2.

Intention	Influencing factor	Prior distribution
$\mathcal{I}_{AT}$	Maneuverability: low	$\mathcal{N}(480\,\mathrm{s}, 80\,\mathrm{s})$
	Maneuverability: medium	$\mathcal{N}(360\,\mathrm{s}, 75\,\mathrm{s})$
$\mathcal{I}_{C}$	Ship type: commercial	0.99
$\mathcal{I}_{CS_i}$	None	[0.2, 0.2, 0.2, 0.2, 0.2]
$\mathcal{I}_{GS}$	Ship type: commercial	0.995
$\mathcal{I}_{P_i}$	**Relative ship size: similar**	[0.05, 0.90, 0.05]
	Relative ship size: larger	[0.01, 0.59, 0.4]
$\mathcal{I}_{RC}$	Maneuverability: medium, Location: open sea	$\mathcal{N}(1\,\mathrm{km}, 175\,\mathrm{m})$
$\mathcal{I}_{RCF}$	Maneuverability: medium, Location: open sea	$\mathcal{N}(1.5\,\mathrm{km}, 250\,\mathrm{m})$
$\mathcal{I}_{SD}$	**Maneuverability: medium, Location: open sea, Speed: slow**	$\mathcal{N}(300\,\mathrm{m}, 75\,\mathrm{m})$
	Maneuverability: low, Location: open sea, Speed: slow	$\mathcal{N}(700\,\mathrm{m}, 100\,\mathrm{m})$
$\mathcal{I}_{SDF}$	Maneuverability: medium, Location: open sea, Speed: slow	$\mathcal{N}(500\,\mathrm{m}, 120\,\mathrm{m})$
$\mathcal{I}_{SDM}$	Maneuverability: medium, Location: open sea, Speed: slow	$\mathcal{N}(300\,\mathrm{m}, 75\,\mathrm{m})$
$\mathcal{I}_{SS}$	Maneuverability: medium, Location: open sea	$\mathcal{N}(7\,\mathrm{km}, 1.7\,\mathrm{km})$
$\mathcal{I}_{U}$	None	0.9999

$$\begin{aligned}&\wedge \left(\mathcal{I}_{SD} > SD_{min}\right) \wedge \left(\mathcal{I}_{SDF} > SDF_{min}\right)\\&\wedge \left(\mathcal{I}_{SDM} > SDM_{min}\right) \wedge \neg \mathcal{I}_{U}\end{aligned} \tag{24}$$

3.5.2. Candidate trajectories

The second approach evaluates whether a candidate trajectory for the reference ship is compatible with the estimated intentions. Measurements can be evaluated based on the candidate trajectory and inserted into the network. The network can then be used to evaluate the probability that this trajectory is compatible with the reference ship's intentions ($C[t]$). These candidate trajectories with corresponding probability can be used as scenarios in scenario-based collision avoidance algorithms similar to Tengesdal et al. (2020)

Minor alterations are needed to evaluate the measurements based on trajectories. All measurements that consider that the reference ship is keeping its course and speed are instead evaluated using the candidate trajectory of the reference ship while only assuming that all other ships in the encounter will keep their course and speed. The current course ($\mathcal{M}_C$) and speed ($\mathcal{M}_S$) must be evaluated a bit into the candidate trajectory so that the ship has time to execute the potential evasive action. If the situation has not started, then a trajectory keeping the course and speed will be wrongly given a high probability. This is avoided by setting the current distance ($\mathcal{M}_{D_i}[t]$) to zero. The time until CPA ($\mathcal{M}_{TCPA_i}$) is not relevant for the candidate trajectories as the entire future motion of the ship is considered as known. Instead, this measurement is set to the minimum acceptable time (AT_{min}). An intention to give way at a shorter time than acceptable will evaluate a high probability for trajectories that keep the course and speed. This makes the collision avoidance algorithm take evasive actions if it is likely that the reference ship will give way at an unacceptable short time before CPA. The rest of the measurements can be evaluated as usual.

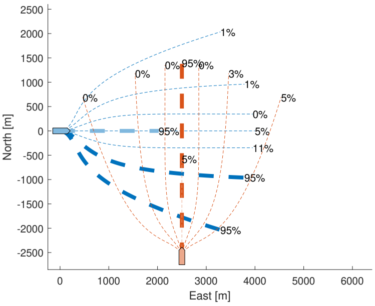


Figure 6: Scenario 1. Two ships are meeting on a collision course in a clear crossing situation. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

There are many different ways of generating candidate trajectories. This article generates trajectories based on line-of-sight guidance, as proposed in Johansen et al. (2016). These trajectories are generated by simulating a simple ship model that uses a line-of-sight guidance rule to evaluate a reference course that gradually converges towards the nominal path (Fossen, 2011). The nominal path is assumed to go in a straight line going through the position where the ship was first observed, pointing in the same direction as the ship's course at this point. Adding different constant offsets to the reference course generates different trajectories that quickly move away from and then align parallel to the original course. Fig. 6 to Fig. 16 shows the resulting trajectories with a constant offset in speed or course. All the trajectories assume that evasive actions are done at the current time-step and not at future time-steps. This assumption can be acceptable for collision avoidance, as it is enough to know if the other ship will give way in time and to what side it will give way.

4. Results

This section presents different simulation scenarios that demonstrate the capabilities of the intention model. For each scenario, the probabilities of the different candidate trajectories being compatible with the reference ship's intentions ($C[t]$) are presented, together with the probability that the ship will give way (WGW_i). The probabilities for all trajectory candidates do not need to sum to 1 as there can be multiple trajectory candidates that are compatible with the intentions of the reference ship. The DBN is in each scenario evaluated using the SMILE (BAYESFUSION LLC, b) library for C++. A separate instance of the model is run for all ships in the encounter.

Scenario 1 - Gradual inference

This scenario demonstrates an ability to identify the intentions based on observations. Fig. 6 shows two ships meeting on a collision course. The situation is a clear crossing situation where, according to COLREGS Rule 15, the blue ship is responsible for giving way while the red should stand on. The model evaluates a 93% chance that the blue ship will give way (WGW) and a 6% chance that the red ship will give way. The blue ship can give way either by reducing its speed or making a starboard turn.

Fig. 7 shows the same situation at a later time-point. As the blue ship has not yet done any action to avoid collision, it becomes more likely that it believes it has a higher priority making it not give way at all. The model, therefore, evaluates a 68% chance that the blue ship will give way. As the red ship has not changed its course or speed, it becomes less likely that it thinks it has lower priority, which results in a 1% chance that it will give way.

When the red ship starts to make an evasive maneuver, as shown in Fig. 8, it becomes more likely that the red ship acts to avoid collision. Note that the candidate trajectories are generated relative to the nominal path of the ship, which

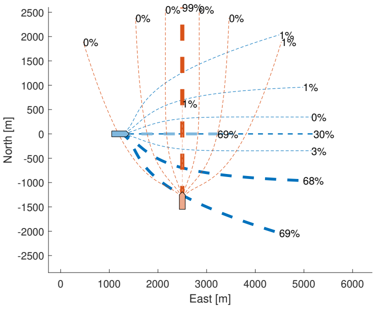


Figure 7: Scenario 1. Shows the same encounter as Fig. 6 at a later time-point. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

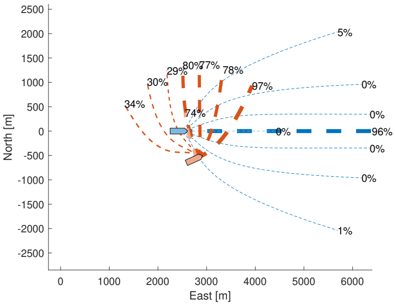


Figure 8: Scenario 1. Shows the same encounter as Fig. 6 and Fig. 7 at a later time-point. The red ship has changed its course 45°to starboard and halved its speed. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

is assumed to continue northwards. As the time until CPA is very short, it is unlikely that the red ship has such a short definition of ample time. The model, therefore, evaluates a 32% chance that the red ship acts in an unmodelled manner. The probability that the red ship will give way is evaluated to be 29%.

Scenario 2 - COLREGS incompliant action.

This scenario demonstrates the modeling of incompliant behavior. Fig. 9 shows two ships meeting on a collision course where the blue ship has turned its course to port to cross in front of the red ship. The model predicts that the blue ship will continue to cross in front even though this is COLREGS incompliant.

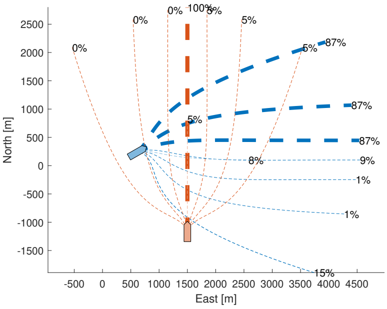


Figure 9: Scenario 2. The ships approached in the same manner as shown in Fig. 6. The blue ship performed a COLREGS incompliant maneuver by changing course to port to avoid collision. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

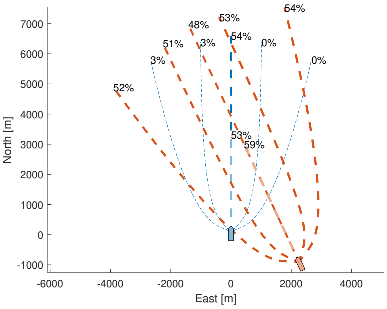


Figure 10: Scenario 3. The red ship is approaching the blue ship with a higher speed and a relative bearing of 113 degrees relative to the heading of the blue ship. The bearing is close to the limit between overtaking and crossing, which can cause uncertainty. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

Scenarios 3, 4, and 5 - Uncertain COLREGS situation

These scenarios demonstrate how uncertainty in the COLREGS situation affects the model. Fig. 10 shows a scenario where the red ship is approaching the blue ship from an angle that is close to the border between an overtaking and crossing situations. The situation metric evaluates a 54% chance of it being an overtaking situation, in which case the red ship should give way to either side. The remaining 46% is evaluated as a crossing situation, in which case the blue ship should give way behind the red ship. This results in a substantial probability for both keeping the course and speed and taking evasive actions. For the blue ship, none of the candidate trajectories where course alone was changed made the blue ship cross behind the red ship at a safe distance. The only option among the candidate trajectories that gave way behind the red ship was for the blue ship to reduce its speed.

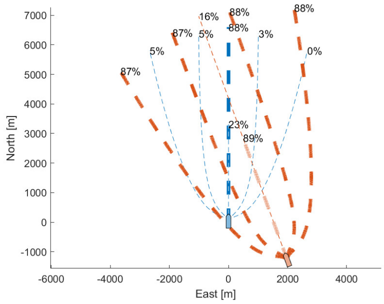


Figure 11: Scenario 4. The red ship is approaching the blue ship with a higher speed and a relative bearing of 123 degrees relative to the heading of the blue ship. The bearing is closer to an overtaking situation than for Fig. 10 making it more likely that the red ship will give way. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

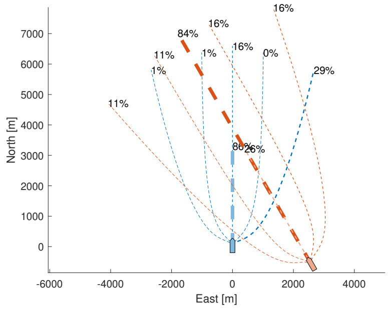


Figure 12: Scenario 5. The red ship is approaching the blue ship with a higher speed and a relative bearing of 103 degrees relative to the heading of the blue ship. The bearing is closer to a crossing situation than for Fig. 10 making it more likely that the blue ship will give way. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

Fig. 11 shows a similar situation but with the red ship approaching at a steeper angle, making it more likely to be an overtaking situation. Similarly, Fig. 12 shows a situation with the red ship approaching at a shallower angle, making it more likely to be a crossing situation.

Scenario 6 - Risk of collision

This scenario demonstrates uncertainties that arise from whether there is a risk of collision (RC_i). Fig. 13 shows two ships meeting in a head-on situation. The model evaluates a 61% chance that there is a risk of collision, and a 86% chance that either ship will give way. If there is no risk of collision, then all actions that keep the ships at a risk-free

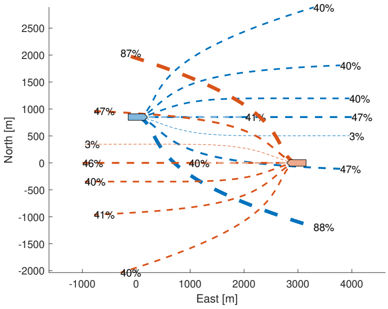


Figure 13: Scenario 6. Two ships are approaching in a head-on situation where it is uncertain whether there is a risk of collision (RC_i). The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

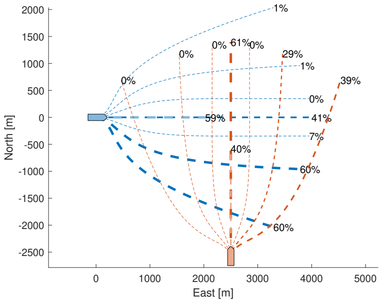


Figure 14: Scenario 7. Same situation as Scenario 1. Information that the blue ship is substantially larger than the red ship is inserted as prior information. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

distance are acceptable. Either way, making a large starboard turn is acceptable as it results in crossing as specified in COLREGS Rule 14.

Scenarios 7 and 8 - Effect of priors

These scenarios demonstrate how utilizing prior information to modify the prior probability distributions affects the model. Fig. 14 shows the same scenario as Scenario 1 but utilizes information that the blue ship is significantly larger than the red ship. The model, therefore, evaluates a substantially larger probability that the blue ship has priority over the red, which results in a 58% chance that the blue ship will give way and a 40% chance that the red ship will give way. Similarly, Fig. 15 shows the same scenario as Scenario 1 but with the maneuverability of both ships set to low. This makes it more likely that the blue ship will try to cross with a larger distance between the ships.

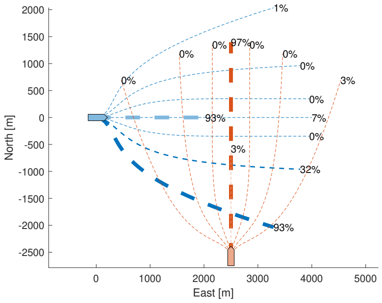


Figure 15: Scenario 8. Same situation as Scenario 1. Information that both ships have a low maneuverability is inserted as prior information. The figure shows the different candidate trajectories (dashed lines). The probability at the end of each trajectory and the thickness of the line shows the probability that the trajectory is compatible with the ship's intentions ($C[t]$). Trajectories with reduced speed are shown with a lighter color. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

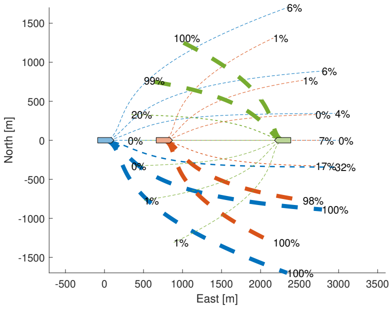


Figure 16: Scenario 9. A collision encounter consisting of three ships. The figure shows the different candidate trajectories (dashed lines) with their respective probability of being compatible with the ship's intentions ($C[t]$). The thickness of the line is proportional to the probability of that trajectory being compatible with the ship's intentions. The ship symbols are scaled for visualization purposes and do not represent the true ship size.

Scenario 9 - Multi-ship encounters

Fig. 16 shows an encounter with three ships, where the red and green ship have a head-on encounter, while the blue ship has an overtaking encounter with the red ship and a head-on encounter with the green ship. If the blue ship had only considered the red ship, then it would be allowed to cross on either side of the ship. As the evasive maneuver has to be correct towards both ships, it can only change its course towards starboard.

5. Discussion

Scenario 1 demonstrates that the model is able to infer the intentions of a ship based on its observed position, course, and speed. The blue ship did not change its course as it approached. This behavior could be explained by the blue ship having high priority or by having a short ample time. Once the ships came closer, the probability that the blue ship had a definition of ample time that was lower than the remaining time until CPA decreased. This increased the probability that the blue ship had higher priority. The red ship changed its course and speed shortly before CPA to avoid collision. Before this point in time, the model did not increase the chance that the red ship would give way as it did not give any indications of giving way. When the red ship finally changed course, the time until CPA was very short, making it quite unlikely that the red ship had such a short definition of ample time. As this behavior does not fit very well with the model, a high chance was evaluated that the red ship acts in an unmodelled way. A collision avoidance algorithm using this intention inference module should display conservative behavior when unmodelled behavior is observed. This will be the case when evaluating candidate trajectories, as all trajectories will have an increased probability of being compatible. When using the intentions as decision criteria, unmodelled behavior will count as not giving way, thereby making the own-ship give way.

Scenario 1 and Scenario 2 show that having multiple different intention variables that can explain a ship's behavior increases the fidelity of the model. In both scenarios, the blue ship acted in a COLREGS incompliant manner. Modeling how the ships are incompliant enables the model to distinguish between Scenario 1 where the blue ship will stand on and the red ship must give way, and Scenario 2 where the blue ship does an evasive maneuver, although to the wrong side.

Modelling of the underlying causes that can cause misunderstandings is demonstrated in Scenario 3, Scenario 4, and Scenario 5. Having a clear distinction between the different COLREGS situations is prone to cause misunderstandings, as it is unlikely that the ships will evaluate borderline situations exactly the same. By modeling this uncertainty, it becomes clear that its insufficient to blindly trust the own-ships interpretation of the situation.

In Scenario 6 the uncertainty stems from whether there is a risk of collision. This scenario gives an example where it is insufficient to consider a single parameter for collision avoidance, such as if the ship will give way. In most other situations, the own-ship must give way if the other ship does not fulfill its obligation. In this situation, the opposite is true; if the other ship fulfills its obligations, then both ships must give way. If the other ship keeps its course, then the own-ship can turn a safe situation into a potentially dangerous one by giving way with a significant starboard maneuver, which is required by COLREGS rule 14.

Scenario 7 and Scenario 8 show that additional information, such as the relative ship size or ship maneuverability, can be used to affect the intention probabilities. Having a collision avoidance algorithm that adapts to the current situation is crucial as ships act in very different manners in different situations, such as open waters and inland waterways. The proposed intention model presented in this article is a step towards this ability as it gives the collision avoidance algorithm an understanding of how the other ship will act in the current situation.

Scenario 9 demonstrates that the model can consider encounters with multiple ships. The model considers whether an observed position, course, and speed are compatible with the intention towards all vehicles. The model does not consider that the reference ship has an idea of what the other ships plan to do. This could, for example, be that the blue ship in Fig. 16 predicts that the red ship will make a starboard turn and therefore chooses to take an even larger starboard turn.

The different scenarios were chosen to illustrate how the model is sensitive to the choice of prior parameters. Scenario 3, Scenario 4, and Scenario 5 demonstrates the sensitivity to the situational classifier given in Fig. 3. The scenarios demonstrate how the gradual transition in the classifier causes a gradual transition in the predicted intention and future behavior. Scenario 7 and Scenario 8 show how modifying the prior distributions affected the behaviour. Scenario 7 showed how changing the probability that a ship had higher priority affected the probability that the ship would stand on. Scenario 8 showed how increasing the expected value of what was considered a safe distance made trajectories that crossed at a shorter distance less likely.

Evaluating different candidate trajectories has some advantages, such as being able to better portray situations such as the one shown in Scenario 6. For the trajectories to realistically portray how the reference ship will act, there must be a candidate trajectory that adequately describes the other ship's trajectory. The candidate trajectory and actual trajectory must be close enough to result in the correct collision avoidance behavior for a collision avoidance algorithm utilizing these intentions. Choosing suitable candidate trajectories is not a trivial task. The ones used in this article cannot handle more complicated situations, such as those where the ship is unable to act at the initial time-step but can act at a later one and where the reference ships make more drastic or sequential changes in course or speed.

The probabilities associated with each candidate trajectory do not represent the probability that the reference ship will follow this trajectory. Instead, it represents the probability that this trajectory is something the reference ship would consider acceptable when only considering properties related to COLREGS. If it is known that the ship will follow

COLREGS and how it defines the different ambiguities such as ample time and safe distance, then all trajectories that adhere to this definition of the rules will be given a 100% probability of being compatible with the intentions.

This article has not considered grounding risk or the COLREGS rules regarding traffic separation schemes (Rule 10), narrow channels (Rule 9), and sailing vessels (Rule 12). Regarding traffic separation schemes and grounding risk, generating candidate trajectories will be more challenging as the trajectories must cover the ship's different options, such as following and leaving the traffic separation scheme correctly. In these situations, it might be necessary to dynamically generate the trajectories based on the current circumstances. An additional challenge arises in narrow channels due to stand-on vessels being allowed to change their course to follow the channel (Woerner et al., 2019).

Furthermore, the model does not explicitly consider measurement uncertainties. This should not be a problem as long as the noise is less than $\mathcal{P}_{CIC}$ and $\mathcal{P}_{CIS}$. If the noise is substantial, then measurement uncertainty should be modeled as well. This can be achieved by having separate nodes representing the measured state and the measurement itself. The measurements themselves should be child nodes of the measured state, and their CPTs should describe the measurement uncertainty. This way of modeling is called the measurement idiom (Fenton and Neil, 2018).

The model assumes all initial changes in course are large enough to avoid collision without requiring additional course changes. This assumption does not hold if the model is fed an observation in the middle of a course change. The model can then evaluate that the ship is not standing on correct (as it changed its course), nor is it giving way correct (as the course change is too small to avoid collision). This can be handled by introducing a node indicating whether the other ship is currently changing its course.

To have acceptable computational time, the number of time-steps in the DBN must be limited. This can be achieved with a sliding window approach where only the last couple of observations are considered. The priors for the intention nodes must be updated to represent the information that is no longer inside the window. This is done by setting the intention priors equal to what the posterior was at the last time-step that is no longer in the window. With a limited window, the frequency of new observations inserted into the model must be considered. Feeding information more often makes the window consider a shorter time span which will contain more similar observations. This will reduce the inference capabilities of the model. Feeding information less often makes the model respond to changes slower. Not all measurements need to be saved as a time-step in the DBN. The newest time-step of the DBN could be updated at a quick frequency and then only saved as a new time-step if it contained substantial new information relative to the previously saved time-steps. This should make the DBN respond quickly and keep a high inference quality with a limited window.

6. Conclusion

This article presents a novel approach for modeling and inferring the intentions of other ships in a potential collision encounter at sea. The simulation study shows that the method is able to infer the state of different intention nodes, identify situations that are likely to lead to misunderstanding, and adapt the intention probabilities to the current situation. This opens up for new possibilities for collision avoidance algorithms. It could enable collision avoidance algorithms to act more safely and predictably as they will better understand the future motion of meeting traffic. They could become able to take early proactive actions to turn a situation prone to misunderstandings into a clear situation where all ships agree on how to act. Lastly, it opens up for collision avoidance algorithms to adapt to the current situation, such as relative ship size and locations. This is an essential feature for collision avoidance algorithms working in multiple different situations where different tuning parameters are needed.

The focus of this article is the enhanced modeling and inference capabilities achieved with the proposed framework. Future work is needed on expanding the model to include the parts of COLREGS that were not considered, to consider grounding, to consider factors outside of COLREGS that affect how ships behave, and to validate the model with historical data. Furthermore, work is needed on gathering the statistics that work as priors for the different intention states and on identifying how they are affected by available information on the current situation. Lastly, collision avoidance algorithms must be developed that can utilize the increased situational awareness provided by this model.

CRediT authorship contribution statement

Sverre Velten Rothmund: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft. **Trym Tengesdal:** Conceptualization, Software, Writing - Review & Editing. **Edmund Førland Brekke:** Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Tor Arne Johansen:** Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

BAYESFUSION LLC, a. GeNIe. URL: https://www.bayesfusion.com/smile/.

BAYESFUSION LLC, b. SMILE Engine. URL: https://www.bayesfusion.com/smile/.

Ben Mrad, A., Delcroix, V., Maalej, M.A., Piechowiak, S., Abid, M., 2012. Uncertain Evidence in Bayesian Networks: Presentation and Comparison on a Simple Example, in: Communications in Computer and Information Science, pp. 39–48. URL: http://link.springer.com/10.1007/978-3-642-31718-7{_}5, doi:10.1007/978-3-642-31718-7_5.

Chauvin, C., 2011. Human Factors and Maritime Safety. Journal of Navigation 64, 625–632. URL: https://www.cambridge.org/core/product/identifier/S0373463311000142/type/journal{_}article, doi:10.1017/S0373463311000142.

Chauvin, C., Lardjane, S., 2008. Decision making and strategies in an interaction situation: Collision avoidance at sea. Transportation Research Part F: Traffic Psychology and Behaviour 11, 259–269. URL: https://doi.org/10.1016/j.trf.2008.01.001, doi:10.1016/j.trf.2008.01.001.

Chen, Y., Zhao, F., Lou, Y., 2021. Interactive Model Predictive Control for Robot Navigation in Dense Crowds. IEEE Transactions on Systems, Man, and Cybernetics: Systems , 1–13URL: https://ieeexplore.ieee.org/document/9328889/, doi:10.1109/TSMC.2020.3048964.

Cho, Y., Kim, J., Kim, J., 2021. Intent Inference of Ship Collision Avoidance Behavior Under Maritime Traffic Rules. IEEE Access 9, 5598–5608. URL: https://ieeexplore.ieee.org/document/9312187/, doi:10.1109/ACCESS.2020.3048717.

Clawson Jr, S.R., 2013. Overtaking or crossing? Don't assume what other ship will do. URL: http://www.professionalmariner.com/August-2013/Overtaking-or-crossing-Dont-assume-what-other-ship-will-do/.

Du, L., Goerlandt, F., Valdez Banda, O.A., Huang, Y., Wen, Y., Kujala, P., 2020. Improving stand-on ship's situational awareness by estimating the intention of the give-way ship. Ocean Engineering 201, 107110. URL: https://doi.org/10.1016/j.oceaneng.2020.107110, doi:10.1016/j.oceaneng.2020.107110.

Eriksen, B.O.H., Bitar, G., Breivik, M., Lekkas, A.M., 2020. Hybrid Collision Avoidance for ASVs Compliant With COLREGs Rules 8 and 13–17. Frontiers in Robotics and AI 7, 1–18. URL: https://www.frontiersin.org/article/10.3389/frobt.2020.00011/full, doi:10.3389/frobt.2020.00011, arXiv:1907.00198.

Fenton, N., Neil, M., 2018. Risk Assessment and Decision Analysis with Bayesian Networks. 2nd editio ed., Chapman & Hall/CRC.

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. Wiley. doi:10.1002/9781119994138.

Hardy, J., Campbell, M., 2013. Contingency Planning Over Probabilistic Obstacle Predictions for Autonomous Road Vehicles. IEEE Transactions on Robotics 29, 913–929. URL: https://ieeexplore.ieee.org/document/6497657/, doi:10.1109/TRO.2013.2254033.

Hashimoto, Y., Gu, Y., Hsu, L.T., Kamijo, S., 2015. Probability estimation for pedestrian crossing intention at signalized crosswalks, in: 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), IEEE. pp. 114–119. URL: http://ieeexplore.ieee.org/document/7396904/, doi:10.1109/ICVES.2015.7396904.

Hegde, J., Utne, I.B., Schjølberg, I., Thorkildsen, B., 2018. A Bayesian approach to risk modeling of autonomous subsea intervention operations. Reliability Engineering and System Safety 175, 142–159. URL: https://doi.org/10.1016/j.ress.2018.03.019, doi:10.1016/j.ress.2018.03.019.

Huang, Y., Chen, L., Chen, P., Negenborn, R.R., van Gelder, P., 2020. Ship collision avoidance methods: State-of-the-art. Safety Science 121, 451–473. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925753519306356, doi:10.1016/j.ssci.2019.09.018.

IMO, . AIS transponders. URL: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx. accessed on: May 19, 2021.

IMO, 1972. COLREGs - Convention on the International Regulations for Preventing Collisions at Sea. URL: http://www.mar.ist.utl.pt/mventura/Projecto-Navios-I/IMO-Conventions(copies)/COLREG-1972.pdf.

Johansen, T.A., Perez, T., Cristofaro, A., 2016. Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment. IEEE Transactions on Intelligent Transportation Systems 17, 3407–3422. URL: http://ieeexplore.ieee.org/document/7464354/, doi:10.1109/TITS.2016.2551780.

Krozel, J., Andrisani, D., 2006. Intent Inference with Path Prediction. Journal of Guidance, Control, and Dynamics 29, 225–236. URL: https://arc.aiaa.org/doi/10.2514/1.14348, doi:10.2514/1.14348.

Mkrtchyan, L., Podofillini, L., Dang, V.N., 2016. Methods for building Conditional Probability Tables of Bayesian Belief Networks from limited judgment: An evaluation for Human Reliability Application. Reliability Engineering and System Safety 151, 93–112. doi:10.1016/j.ress.2016.01.004.

Rokseth, B., Utne, I.B., Vinnem, J.E., 2017. A systems approach to risk analysis of maritime operations. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 231, 53–68. doi:10.1177/1748006X16682606.

Russell, S.J., Norvig, P., 2014. Artificial Intelligence: A Modern Approach. 3 ed., Pearson Education.

Tengesdal, T., Johansen, T.A., Brekke, E., 2020. Risk-based Autonomous Maritime Collision Avoidance Considering Obstacle Intentions, in: 2020 IEEE 23rd International Conference on Information Fusion (FUSION), IEEE. pp. 1–8. URL: https://ieeexplore.ieee.org/document/9190212/, doi:10.23919/FUSION45008.2020.9190212.

Vagale, A., Oucheikh, R., Bye, R.T., Osen, O.L., Fossen, T.I., 2021. Path planning and collision avoidance for autonomous surface vehicles I: a review. Journal of Marine Science and Technology i, 2018–2028. URL: https://doi.org/10.1007/s00773-020-00787-6, doi:10.1007/s00773-020-00787-6.

Woerner, K., Benjamin, M.R., Novitzky, M., Leonard, J.J., 2019. Quantifying protocol evaluation for autonomous collision avoidance: Toward establishing COLREGS compliance metrics. Autonomous Robots 43, 967–991. URL: https://doi.org/10.1007/s10514-018-9765-y, doi:10.1007/s10514-018-9765-y.

Yepes, J.L., Hwang, I., Rotea, M., 2007. New Algorithms for Aircraft Intent Inference and Trajectory Prediction. Journal of Guidance, Control, and Dynamics 30, 370–382. URL: https://arc.aiaa.org/doi/10.2514/1.26750, doi:10.2514/1.26750.

References

[1] J. G. Landels, *Engineering in the ancient world.* University of California Press, 2000.

[2] J. C. Maxwell, "I. on governors," *Proceedings of the Royal Society of London*, vol. 16, pp. 270–283, 1868. DOI: 10.1098/rspl.1867.0055.

[3] P. N. Paraskevopoulos, *Modern control engineering.* CRC Press, 2017.

[4] J. Cusack, *Self-driving vehicles are steadily becoming a reality despite the many hurdles still to be overcome – and they could change our world in some unexpected ways.* Visited on 18.04.2022, 2021. [Online]. Available: https://www.bbc.com/future/article/20211126-how-driverless-cars-will-change-our-world.

[5] M. Tranzatto, F. Mascarich, L. Bernreiter, *et al.*, "CERBERUS: autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA subterranean challenge," *CoRR*, vol. abs/2201.07067, 2022. arXiv: 2201.07067. [Online]. Available: https://arxiv.org/abs/2201.07067.

[6] A. R. Graven, *Drones monitor the airport – developing autonomous sensor systems*, Visited on 18.04.2022, 2020. [Online]. Available: https://www.norceresearch.no/en/news/new-drone-inspection-solution-for-airports.

[7] S. Bennett, *A history of control engineering, 1800-1930. eng.* (IEEE control engineering series.). London: Peter Peregrinus on behalf of the Institution of Electrical Engineers, 1979, vol. 8.

[8] N. Minorsky, "Directional stability of automatically steered bodies," *Journal of the American Society for Naval Engineers*, vol. 34, no. 2, pp. 280–309, 1922. DOI: https://doi.org/10.1111/j.1559-3584.1922.tb04958.x.

[9] S. Kvaal and P. Østby, *The Jewel in the Crown.* Pax, 2015.

[10] EMSA, *The european maritime safety agency: Annual overview of marine casualties and incidents*, Annual Report, Nov. 2021. [Online]. Available: http://www.emsa.europa.eu/emsa-documents/latest/item/3406-annual-overview-of-marine-casualties-and-incidents-2018.html.

[11] C. Macrae, "Human factors at sea: Common patterns of error in groundings and collisions," *Maritime Policy & Management*, vol. 36, no. 1, pp. 21–38, 2009.

[12] C. Chauvin, "Human factors and maritime safety," *Journal of Navigation*, vol. 64, pp. 625–632, Oct. 2011.

[13] O. Levander, "Autonomous ships on the high seas," *IEEE Spectrum*, vol. 54, no. 2, pp. 26–31, Feb. 2017.

[14] R. Courtland, *DARPA's self-driving submarine hunter steers like a human*, Visited on 17.04.2022, 2016. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/military-robots/darpa-actuv-self-driving-submarine-hunter-steers-like-a-human.

[15] Kongsberg, *Automatic ferry enters regular service following worlds first crossing with passengers onboard*, Visited on 17.04.2022, 2020. [Online]. Available: https://www.kongsberg.com/maritime/about-us/news-and-media/news-archive/2020/first-adaptive-transit-on-bastofosen-vi/.

[16] A. Skredderberget. "Yara birkeland - the first ever zero emission, autonomous ship." (2018), [Online]. Available: https://www.yara.com/knowledge-grows/game-changer-for-the-environment/.

[17] U. Skoglund, *Førerløse ferger kan erstatte gangbruer*, Visited on 17.04.2022, 2018. [Online]. Available: https://gemini.no/2018/06/forerlose-ferger-kan-erstatte-gangbruer/.

[18] O. A. Cluster, *SFI autoship – eight years with autonomous ships*, Visited on 18.04.2022, 2021. [Online]. Available: https://oceanautonomy.no/sfi-autoship-eight-years-with-autonomous-ships/.

[19] T. Engineer, *Falco makes world's first autonomous ferry crossing*, Visited on 18.04.2022, 2018. [Online]. Available: https://www.theengineer.co.uk/falco-autonomous-ferry-rolls-royce/.

[20] NTNU, *Autoferry*, Visited on 18.04.2022, 2018. [Online]. Available: https://www.ntnu.edu/digital-transformation/autoferry.

[21] I. M. O. (IMO), "COLREGS - International Regulations for Preventing Collisions at Sea," *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, 1972.

[22] C. Chauvin and S. Lardjane, "Decision making and strategies in an interaction situation: Collision avoidance at sea," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 11, no. 4, pp. 259–269, 2008.

[23] S. R. Clawson Jr, *Overtaking or crossing? Don't assume what other ship will do*, 2013. [Online]. Available: http://www.professionalmariner.com/August-2013/Overtaking-or-crossing-Dont-assume-what-other-ship-will-do/.

[24] K. Woerner, M. R. Benjamin, M. Novitzky, and J. J. Leonard, "Quantifying protocol evaluation for autonomous collision avoidance," *Autonomous Robots*, vol. 43, no. 4, pp. 967–991, Apr. 2019, ISSN: 1573-7527. DOI: 10.1007/s10514-018-9765-y.

[25] T. Statheros, G. Howells, and K. M. Maier, “Autonomous ship collision avoidance navigation concepts, technologies and techniques,” *Journal of Navigation*, vol. 61, no. 1, pp. 129–142, 2008. DOI: 10.1017/S037346330700447X.

[26] C. Tam, R. Bucknall, and A. Greig, “Review of collision avoidance and path planning methods for ships in close range encounters,” *The Journal of Navigation*, vol. 62, no. 3, pp. 455–476, 2009.

[27] S. Campbell, W. Naeem, and G. Irwin, “A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres,” *Annual Reviews in Control*, vol. 36, no. 2, pp. 267–283, 2012, ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2012.09.008.

[28] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. van Gelder, “Ship collision avoidance methods: State-of-the-art,” *Safety Science*, vol. 121, pp. 451–473, 2020.

[29] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, “Path planning and collision avoidance for autonomous surface vehicles i: A review,” *Journal of Marine Science and Technology*, 2021, ISSN: 1437-8213. DOI: 10.1007/s00773-020-00787-6.

[30] A. Vagale, R. T. Bye, R. Oucheikh, O. L. Osen, and T. I. Fossen, “Path planning and collision avoidance for autonomous surface vehicles ii: A comparative study of algorithms,” *Journal of Marine Science and Technology*, 2021, ISSN: 1437-8213. DOI: 10.1007/s00773-020-00790-x.

[31] M. Akdağ, P. Solnør, and T. A. Johansen, “Collaborative collision avoidance for maritime autonomous surface ships: A review,” *Ocean Engineering*, vol. 250, p. 110 920, 2022.

[32] O. Mitrofanov, “An anti-collision indicator,” *Journal of Navigation*, vol. 21, no. 2, pp. 163–170, 1968. DOI: 10.1017/S0373463300030319.

[33] R. F. Riggs, “A modern collision avoidance display technique,” *Journal of Navigation*, vol. 28, no. 2, pp. 143–155, 1975. DOI: 10.1017/S0373463300037681.

[34] T. Miloh and S. D. Sharma, “Maritime collision avoidance as a differential game,” *Schiffstechnik*, vol. 24, no. 116, 1977.

[35] G. J. Olsder and J. L. Walter, “A differential game approach to collision avoidance of ships,” in *Optimization Techniques Part 1*, J. Stoer, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 264–271, ISBN: 978-3-540-35891-6.

[36] W. P. Cannell, “Collision avoidance as a game of co-ordination,” *Journal of Navigation*, vol. 34, no. 2, pp. 220–239, 1981.

[37] A. W. Merz, “Optimal evasive manoeuvers in maritime collision avoidance,” *Journal of Navigation*, vol. 20, 1973.

[38] A. W. Merz and J. S. Karmarkar, “Collision avoidance systems and optimal turn manœuvres,” *Journal of Navigation*, vol. 29, no. 2, pp. 160–174, 1976. DOI: 10.1017/S0373463300030150.

[39] C. de Wit and J. Oppe, "Optimal collision avoidance in unconfined waters," *Journal of Navigation*, vol. 26, no. 4, pp. 296–303, 1979. DOI: https://doi.org/10.1002/j.2161-4296.1979.tb01389.x.

[40] T. Degré and X. Lefèvre, "A collision avoidance system," *Journal of Navigation*, vol. 34, no. 2, pp. 294–302, 1981. DOI: 10.1017/S0373463300021408.

[41] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, Jan. 2014, ISSN: 0364-9059. DOI: 10.1109/JOE.2013.2254214.

[42] M. J. Dove, R. S. Burns, and C. T. Stockel, "An automatic collision avoidance and guidance system for marine vehicles in confined waters," *Journal of Navigation*, vol. 39, no. 2, pp. 180–190, 1986. DOI: 10.1017/S0373463300000059.

[43] J. Lisowski and M. Mohamed-Seghir, "Application of fuzzy set theory in the safe ship control process," *IFAC Proceedings Volumes*, vol. 31, pp. 275–279, 1998.

[44] Y.-i. Lee and Y.-G. Kim, "A collision avoidance system for autonomous ship using fuzzy relational products and colregs," in *Intelligent Data Engineering and Automated Learning – IDEAL 2004*, Z. R. Yang, H. Yin, and R. M. Everson, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 247–252, ISBN: 978-3-540-28651-6.

[45] Y. Yavin, C. Frangos, and T. Miloh, "Computation of feasible control trajectories for the navigation of a ship around an obstacle in the presence of a sea current," *Mathematical and Computer Modelling*, vol. 21, no. 3, pp. 99–117, 1995, ISSN: 0895-7177. DOI: https://doi.org/10.1016/0895-7177(94)00218-D.

[46] A. Miele, T. Wang, C. S. Chao, and J. B. Dabney, "Optimal control of a ship for collision avoidance maneuvers," *Journal of Optimization Theory and Applications*, vol. 103, no. 3, pp. 495–519, 1999, ISSN: 1573-2878. DOI: 10.1023/A:1021775722287.

[47] M. Ito, F. Zhnng, and N. Yoshida, "Collision avoidance control of ship with genetic algorithm," in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*, vol. 2, 1999, 1791–1796 vol. 2. DOI: 10.1109/CCA.1999.801243.

[48] R. Smierzchalski and Z. Michalewicz, "Modeling of ship trajectory in collision situations by an evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 227–241, 2000. DOI: 10.1109/4235.873234.

[49] C. Shi, M. Zhang, and J. Peng, "Harmonic potential field method for autonomous ship navigation," in *2007 7th International Conference on ITS Telecommunications*, 2007, pp. 1–6. DOI: 10.1109/ITST.2007.4295916.

[50] Y. Sato and H. Ishii, "Study of a collision-avoidance system for ships," *Control Engineering Practice*, vol. 6, no. 9, pp. 1141–1149, Sep. 1998. DOI: 10.1016/s0967-0661(98)00107-5.

[51] M. Benjamin and J. Curcio, "COLREGS-based navigation of autonomous marine vehicles," in *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, 2004, pp. 32–39. DOI: 10.1109/AUV.2004.1431190.

[52] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman, "A method for protocol-based collision avoidance between autonomous marine surface craft," *Journal of Field Robotics*, vol. 23, no. 5, pp. 333–346, 2006. DOI: https://doi.org/10.1002/rob.20121.

[53] J. Larson, M. Bruch, and J. Ebken, "Autonomous navigation and obstacle avoidance for unmanned surface vehicles," in *Unmanned Systems Technology VIII*, G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, Eds., International Society for Optics and Photonics, vol. 6230, SPIE, 2006, pp. 53–64. DOI: 10.1117/12.663798. [Online]. Available: https://doi.org/10.1117/12.663798.

[54] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," *Proc. of the 2007 Unmanned Systems North America Conference, Washington, D.C., USA*, Jan. 2007.

[55] Ø. A. G. Loe, "Collision avoidance for unmanned surface vehicles," M.S. thesis, NTNU, 2008.

[56] G. Casalino, A. Turetta, and E. Simetti, "A three-layered architecture for real time path planning and obstacle avoidance for surveillance usvs operating in harbour fields," in *OCEANS 2009-EUROPE*, 2009, pp. 1–8. DOI: 10.1109/OCEANSE.2009.5278104.

[57] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[58] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997. DOI: 10.1109/100.580977.

[59] T. Huntsberger, H. Aghazarian, A. Castano, *et al.*, "Intelligent autonomy for unmanned sea surface and underwater vehicles," *Proc of the 2008 AUVSI Unmanned Systems North America Conference, San Diego, California, USA*, 2008.

[60] L. Elkins, D. Sellers, and W. R. Monach, "The autonomous maritime navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles," *Journal of Field Robotics*, vol. 27, no. 6, pp. 790–818, 2010.

[61] B. Kluge and E. Prassler, "Recursive probabilistic velocity obstacles for reflective navigation," vol. 24, Jan. 2003, pp. 71–79, ISBN: 3-540-32801-7. DOI: 10.1007/10991459_8.

[62] P. Svec, B. C. Shah, I. R. Bertaska, *et al.*, "Dynamics-aware target following for an autonomous surface vehicle operating under COLREGs in civilian traffic," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nov. 2013, pp. 3871–3878. DOI: 10.1109/IROS.2013.6696910.

[63] M. Blaich, M. Rosenfelder, M. Schuster, O. Bittel, and J. Reuter, "Fast grid based collision avoidance for vessels using a* search algorithm," in *2012 17th International Conference on Methods Models in Automation Robotics (MMAR)*, 2012, pp. 385–390. DOI: 10.1109/MMAR.2012.6347858.

[64] M. Schuster, M. Blaich, and J. Reuter, "Collision avoidance for vessels using a low-cost radar sensor," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 9673–9678, 2014.

[65] E. M. Goodwin, "A statistical study of ship domains," *The Journal of navigation*, vol. 28, no. 3, pp. 328–344, 1975.

[66] M. Blaich, S. Köhler, J. Reuter, and A. Hahn, "Probabilistic collision avoidance for vessels," *IFAC-PapersOnLine*, vol. 48, no. 16, pp. 69–74, 2015.

[67] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz, "Stereo vision–based navigation for autonomous surface vessels," *Journal of Field Robotics*, vol. 28, no. 1, pp. 3–18, 2011.

[68] P. Agrawal and J. M. Dolan, "COLREGS-compliant target following for an unmanned surface vehicle in dynamic environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1065–1070. DOI: 10.1109/IROS.2015.7353502.

[69] B. C. Shah, P. Švec, I. R. Bertaska, *et al.*, "Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic," *Autonomous Robots*, vol. 40, no. 7, pp. 1139–1163, 2016.

[70] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016, ISSN: 1524-9050.

[71] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "MPC-based collision avoidance strategy for existing marine vessel guidance systems," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2018, pp. 7618–7623. DOI: 10.1109/ICRA.2018.8463182.

[72] D. K. M. Kufoalor, E. Wilthil, I. B. Hagen, E. F. Brekke, and T. A. Johansen, "Autonomous COLREGs-compliant decision making using maritime radar tracking and model predictive control," in *Proc. 18th European Control Conf. (ECC)*, Jun. 2019, pp. 2536–2542. DOI: 10.23919/ECC.2019.8796273.

[73] D. K. Kufoalor, T. Johansen, E. Brekke, A. Hepsø, and K. Trnka, "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios," *Journal of Field Robotics*, vol. 37, pp. 387–403, 2020.

[74] M. Abdelaal and A. Hahn, "NMPC-based trajectory tracking and collision avoidance of unmanned surface vessels with rule-based colregs confinement," in *Proc. Process and Control (ICSPC) 2016 IEEE Conf. Systems*, Dec. 2016, pp. 23–28. DOI: 10.1109/SPC.2016.7920697.

[75] M. Abdelaal, M. Fränzle, and A. Hahn, "NMPC-based trajectory tracking and collison avoidance of underactuated vessels with elliptical ship domain," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 22–27, 2016, 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.10.316.

[76] M. Abdelaal, M. Fränzle, and A. Hahn, "Nonlinear model predictive control for trajectory tracking and collision avoidance of underactuated vessels with disturbances," *Ocean Engineering*, vol. 160, pp. 168–180, 2018, ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2018.04.026.

[77] M. Candeloro, A. M. Lekkas, and A. J. Sørensen, "A voronoi-diagram-based dynamic path-planning system for underactuated marine vessels," *Control Engineering Practice*, vol. 61, pp. 41–54, 2017, ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2017.01.007.

[78] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003, ISSN: 2371-9877. DOI: 10.1109/TAES.2003.1261132.

[79] L. M. Millefiori, P. Braca, K. Bryan, and P. Willett, "Long-term vessel kinematics prediction exploiting mean-reverting processes," in *2016 19th International Conference on Information Fusion (FUSION)*, Jun. 2016, pp. 232–239.

[80] Y. Cheng and W. Zhang, "Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels," *Neurocomputing*, vol. 272, pp. 63–73, 2018, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.06.066.

[81] H. Shen, H. Hashimoto, A. Matsuda, Y. Taniguchi, D. Terada, and C. Guo, "Automatic collision avoidance of multiple ships based on deep q-learning," *Applied Ocean Research*, vol. 86, pp. 268–288, 2019, ISSN: 0141-1187. DOI: https://doi.org/10.1016/j.apor.2019.02.020.

[82] S. Xie, V. Garofano, X. Chu, and R. R. Negenborn, "Model predictive ship collision avoidance based on q-learning beetle swarm antenna search and neural networks," *Ocean Engineering*, vol. 193, p. 106 609, 2019, ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2019.106609.

[83] H. L. Chiang and L. Tapia, "COLREG-RRT: An RRT-based COLREGS-compliant motion planner for surface vehicle navigation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2024–2031, Jul. 2018, ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2801881.

[84] R. Zaccone, M. Martelli, and M. Figari, "A COLREG-compliant ship collision avoidance algorithm," in *Proc. 18th European Control Conf. (ECC)*, Jun. 2019, pp. 2530–2535. DOI: 10.23919/ECC.2019.8796207.

[85] B.-O. H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, "Hybrid collision avoidance for asvs compliant with colregs rules 8 and 13–17," *Frontiers in Robotics and AI*, vol. 7, 2020, ISSN: 2296-9144. DOI: 10.3389/frobt.2020.00011.

[86] B.-O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "The branching-course model predictive control algorithm for maritime collision avoidance," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 2019. DOI: https://doi.org/10.1002/rob.21900.

[87] E. F. Brekke, E. F. Wilthil, B.-O. H. Eriksen, *et al.*, "The autosea project: Developing closed-loop target tracking and collision avoidance systems," *Journal of Physics: Conference Series*, vol. 1357, p. 012 020, Oct. 2019. DOI: 10.1088/1742-6596/1357/1/012020.

[88] J. Park, J. Choi, and H. Choi, "COLREGS-compliant path planning considering time-varying trajectory uncertainty of autonomous surface vehicle," *Electronics Letters*, vol. 55, no. 4, pp. 222–224, 2019, ISSN: 0013-5194.

[89] Y. Cho and J. Kim, "Collision probability assessment between surface ships considering maneuver intentions," in *Proc. OCEANS 2017 - Aberdeen*, Jun. 2017, pp. 1–5. DOI: 10.1109/OCEANSE.2017.8084791.

[90] Y. Cho, J. Kim, and J. Kim, "Intent inference-based ship collision avoidance in encounters with rule-violating vessels," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 518–525, 2022. DOI: 10.1109/LRA.2021.3130386.

[91] I. Hwang and C. E. Seah, "Intent-based probabilistic conflict detection for the next generation air transportation system," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2040–2059, Dec. 2008, ISSN: 1558-2256.

[92] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, Aug. 2013, ISSN: 1941-0468. DOI: 10.1109/TRO.2013.2254033.

[93] B. H. Eriksen and M. Breivik, "MPC-based mid-level collision avoidance for asvs using nonlinear programming," in *Proc. IEEE Conf. Control Technology and Applications (CCTA)*, 2017, pp. 766–772.

[94] T. Tengesdal, T. A. Johansen, and E. F. Brekke, "Ship collision avoidance utilizing the cross-entropy method for collision risk assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 148–11 161, 2022.

[95] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*. YBs Storrs, CT, 1995, vol. 19.

[96] S. Rothmund, T. Tengesdal, E. Brekke, and T. Johansen, "Intention modelling and inference for autonomous collision avoidance at sea," 2022, Preprint submitted to the Journal of Ocean Engineering. DOI: 10.36227/techrxiv.16825870.

[97] T. Tengesdal, E. F. Brekke, and T. A. Johansen, "On collision risk assessment for autonomous ships using scenario-based MPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 509–14 516, 2020, 21st IFAC World Congress, ISSN: 2405-8963.

[98] T. Tengesdal, T. A. Johansen, and E. Brekke, "Risk-based autonomous maritime collision avoidance considering obstacle intentions," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020. DOI: 10.23919/FUSION45008.2020.9190212.

[99] T. Tengesdal, T. A. Johansen, T. A. Grande, and S. Blindheim, "Ship collision avoidance and anti grounding using parallelized cost evaluation in probabilistic scenario-based model predictive control," *IEEE Access*, 2022, Submitted.

[100] T. Tengesdal, S. V. Rothmund, E. A. Basso, T. A. Johansen, and H. Schmidt-Didlaukies, "Obstacle intention awareness in automatic collision avoidance: Full scale experiments in confined waters," *Field Robotics*, 2022, Submitted.

[101] T. Tengesdal, L. Millefiori, P. Braca, and E. Brekke, "Joint stochastic prediction of vessel kinematics and destination based on a maritime traffic graph," in *2nd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives*, In press, 2022.

[102] B. Ahmad, J. Murphy, P. Langdon, and S. Godsill, "Bayesian intent prediction in object tracking using bridging distributions," *IEEE Transactions on Cybernetics*, vol. 48, pp. 215–227, Jan. 2018. DOI: 10.1109/TCYB.2016.2629025.

[103] J. Liang, B. I. Ahmad, R. Gan, P. Langdon, R. Hardy, and S. Godsill, "On destination prediction based on Markov bridging distributions," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1663–1667, 2019. DOI: 10.1109/LSP.2019.2943081.

[104] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes.* Tata McGraw-Hill Education, 2002.

[105] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability and statistics for engineers and scientists.* Macmillan New York, 1993, vol. 5.

[106] J. Geweke, "Chapter 15 monte carlo simulation and numerical integration," in ser. Handbook of Computational Economics, vol. 1, Elsevier, 1996, pp. 731–800. DOI: https://doi.org/10.1016/S1574-0021(96)01017-9.

[107] H. Kahn and A. W. Marshall, "Methods of reducing sample size in monte carlo computations," *Journal of the Operations Research Society of America*, vol. 1, no. 5, pp. 263–278, 1953, ISSN: 00963984. [Online]. Available: www.jstor.org/stable/166789.

[108] J. A. O'Keefe, "The universal transverse mercator grid and projection," *The Professional Geographer*, vol. 4, no. 5, pp. 19–24, 1952.

[109] M. Kumar, "World geodetic system 1984: A modern and accurate global reference frame," *Marine Geodesy*, vol. 12, no. 2, pp. 117–126, 1988. DOI: 10.1080/15210608809379580.

[110] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons, 2011.

[111] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[112] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[113] E. F. Wilthil and E. F. Brekke, "Compensation of navigation uncertainty for target tracking on a moving platform," *19th International Conference on Information Fusion (FUSION)*, pp. 1616–1621, 2016.

[114] E. F. Brekke, *Fundamentals of Sensor Fusion: Target tracking, Navigation and SLAM*. -, 2019, Lecture notes for future course TTK4250 Sensor Fusion. In writing.

[115] L. M. Millefiori, G. Pallotta, P. Braca, S. Horn, and K. Bryan, "Validation of the Ornstein-Uhlenbeck route propagation model in the mediterranean sea," in *Proc. OCEANS 2015 - Genova*, May 2015, pp. 1–6. DOI: 10.1109/OCEANS-Genova.2015.7271565.

[116] L. M. Millefiori, P. Braca, K. Bryan, and P. Willett, "Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2313–2330, 2016. DOI: 10.1109/TAES.2016.150596.

[117] B. Foss and T. A. N. Heirung, *Merging Optimization and Control*. NTNU, 2016.

[118] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, *et al.*, "State of the art of finite control set model predictive control in power electronics," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 1003–1016, 2012.

[119] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, "Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 32–43, 2014.

[120] C. Qiu, Z. Zhang, H. Lu, and H. Luo, "A survey of motion-based multitarget tracking methods," *Progress In Electromagnetics Research B*, vol. 62, pp. 195–223, Jan. 2015.

[121] F. Goerlandt and J. Montewka, "Maritime transportation risk analysis: Review and analysis in light of some foundational issues," *Reliability Engineering & System Safety*, vol. 138, pp. 115–134, 2015.

[122] P. Chen, Y. Huang, J. Mou, and P. van Gelder, "Probabilistic risk analysis for ship-ship collision: State-of-the-art," *Safety Science*, vol. 117, pp. 108–122, 2019.

[123] P. Rajendran, T. Moscicki, J. Wampler, B. C. Shah, K. von Ellenrieder, and S. K. Gupta, "Wave-aware trajectory planning for unmanned surface vehicles operating in congested environments," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Aug. 2018, pp. 1–7.

[124] T. Tengesdal, "Uncertainty management in an SB-MPC for collision avoidance," M.S. thesis, NTNU, 2019.

[125] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[126] M. Evans and J. Rosenthal, *Probability and Statistics: The Science of Uncertainty*. W. H. Freeman, 2009.

[127] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. New York, NY, USA: John Wiley & Sons, Inc., 2001, ISBN: 047141655X.

[128] P. Ioannou and J. Sun, *Robust Adaptive Control*. Dover Publications Inc, 2012.

[129] E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "A target tracking system for ASV collision avoidance based on the PDAF," in *Sensing and Control for Autonomous Vehicles*, Springer, 2017, pp. 269–288.

[130] M. Mallick and B. L. Scala, "Comparison of single-point and two-point difference track initiation algorithms using position measurements," *Acta Automatica Sinica*, vol. 34, no. 3, pp. 258–265, 2008, ISSN: 1874-1029.

[131] B. H. Eriksen, E. F. Wilthil, A. L. Flåten, E. F. Brekke, and M. Breivik, "Radar-based maritime collision avoidance using dynamic window," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–9.

[132] I. B. Hagen, "Collision avoidance for asvs using model predictive control," M.S. thesis, NTNU, 2017.

[133] S. Hexeberg, A. L. Flaten, B.-O. H. Eriksen, and E. F. Brekke, "AIS-based vessel trajectory prediction," *20th International Conference on Information Fusion*, pp. 1–8, 2017.

[134] B. R. Dalsnes, S. Hexeberg, A. L. Flaten, B.-O. H. Eriksen, and E. F. Brekke, "The neighbor course distribution method with Gaussian mixture models for AIS-based vessel trajectory prediction," *21st International Conference on Information Fusion*, pp. 580–587, 2018.

[135] E. Tu, G. Zhang, S. Mao, L. Rachmawati, and G. Huang, "Modeling historical ais data for vessel path prediction: A comprehensive treatment," *ArXiv*, vol. abs/2001.01592, 2020.

[136] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998. DOI: 10.1109/7.640267.

[137] D. P. Kroese, R. Y. Rubinstein, and P. W. Glynn, "Chapter 2 - the cross-entropy method for estimation," in *Handbook of Statistics*, ser. Handbook of Statistics, C. Rao and V. Govindaraju, Eds., vol. 31, Elsevier, 2013, pp. 19–34.

[138] I. B. Utne, B. Rokseth, A. J. Sørensen, and J. E. Vinnem, "Towards supervisory risk control of autonomous ships," *Reliability Engineering & System Safety*, vol. 196, p. 106 757, 2020, ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.106757.

[139] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.

[140] R. A. Paielli and H. Erzberger, “Conflict probability estimation generalized to non-level flight,” *Air Traffic Control Quarterly*, vol. 7, no. 3, pp. 195–222, 1999.

[141] T. Lauderdale, “Probabilistic conflict detection for robust detection and resolution,” in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5643.

[142] K.-Y. Baek and H.-C. Bang, “ADS-B based trajectory prediction and conflict detection for air traffic management,” *International Journal of Aeronautical and Space Sciences*, vol. 13, pp. 377–385, Sep. 2012. DOI: 10.5139/IJASS.2012.13.3.377.

[143] W. Liu and I. Hwang, “Probabilistic trajectory prediction and conflict detection for air traffic control,” *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1779–1789, 2011. DOI: 10.2514/1.53645.

[144] L. Yang, J. Yang, J. Kuchar, and E. Feron, “A real-time Monte Carlo implementation for computing probability of conflict,” vol. 1, Aug. 2004, ISBN: 978-1-62410-073-4. DOI: 10.2514/6.2004-4876.

[145] A. Lambert, D. Gruyer, and G. Saint Pierre, “A fast Monte Carlo algorithm for collision probability estimation,” in *2008 10th International Conference on Control, Automation, Robotics and Vision*, Dec. 2008, pp. 406–411. DOI: 10.1109/ICARCV.2008.4795553.

[146] C. E. van Daalen and T. Jones, “Fast conflict detection using probability flow,” *Automatica*, vol. 45, no. 8, pp. 1903–1909, 2009.

[147] M. Mitici and H. A. P. Blom, “Mathematical models for air traffic conflict and collision probability estimation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, pp. 1052–1068, 3 2019, ISSN: 1558-0016. DOI: 10.1109/TITS.2018.2839344.

[148] J. Park and J. Kim, “Predictive evaluation of ship collision risk using the concept of probability flow,” *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 836–845, 2017.

[149] D. Kroese, S. Porotsky, and R. Rubinstein, “The cross-entropy method for continuous multi-extremal optimization,” *Methodology and Computing in Applied Probability*, vol. 8, pp. 383–407, Sep. 2006. DOI: 10.1007/s11009-006-9753-0.

[150] F. Daum and J. Huang, “Particle degeneracy: root cause and solution,” in *Signal Processing, Sensor Fusion, and Target Recognition XX*, I. Kadar, Ed., International Society for Optics and Photonics, vol. 8050, SPIE, 2011, pp. 367–377.

[151] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[152] B.-O. H. Eriksen, G. I. Bitar, M. Breivik, and A. M. Lekkas, “Hybrid collision avoidance for ASVs compliant with COLREGs rules 8 and 13-17,” *ArXiv*, vol. abs/1907.00198, 2019.

[153] Y. Lyu, J. Hu, B. M. Chen, C. Zhao, and Q. Pan, "Multivehicle flocking with collision avoidance via distributed model predictive control," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2651–2662, 2021. DOI: 10.1109/TCYB.2019.2944892.

[154] D. H. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973.

[155] M. Breivik and T. I. Fossen, "Guidance laws for planar motion control," in *Proc. 47th IEEE Conf. Decision and Control*, Dec. 2008, pp. 570–577. DOI: 10.1109/CDC.2008.4739465.

[156] B. Patle, G. Babu L, A. Pandey, D. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019, ISSN: 2214-9147. DOI: https://doi.org/10.1016/j.dt.2019.04.011.

[157] S. Blindheim and T. A. Johansen, "Electronic navigational charts for visualization, simulation, and autonomous ship control," *IEEE Access*, vol. 10, pp. 3716–3737, 2022. DOI: 10.1109/ACCESS.2021.3139767.

[158] C.-W. Huang and T.-Y. Shih, "On the complexity of point-in-polygon algorithms," *Computers and Geosciences*, vol. 23, no. 1, pp. 109–118, 1997, ISSN: 0098-3004. DOI: https://doi.org/10.1016/S0098-3004(96)00071-4.

[159] S. Blindheim, S. Gros, and T. Johansen, "Risk-based model predictive control for autonomous ship emergency management," *IFAC-PapersOnLine*, vol. 53, pp. 14 524–14 531, 2020.

[160] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, Mar. 1995, ISSN: 0018-7208. DOI: 10.1518/001872095779049543.

[161] D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Proactive collision avoidance for asvs using a dynamic reciprocal velocity obstacles method," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 2402–2409. DOI: 10.1109/IROS.2018.8594382.

[162] Y. Cho, J. Han, and J. Kim, "Efficient colreg-compliant collision avoidance in multi-ship encounter situations," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020. DOI: 10.1109/TITS.2020.3029279.

[163] Y. Cho, J. Kim, and J. Kim, "Intent inference of ship collision avoidance behavior under maritime traffic rules," *IEEE Access*, vol. 9, pp. 5598–5608, 2021. DOI: 10.1109/ACCESS.2020.3048717.

[164] L. Du, F. Goerlandt, O. A. Valdez Banda, Y. Huang, Y. Wen, and P. Kujala, "Improving stand-on ship's situational awareness by estimating the intention of the give-way ship," *Ocean Engineering*, vol. 201, p. 107 110, 2020, ISSN: 0029-8018.

[165] Y. Cho, J. Han, and J. Kim, “Intent inference of ship maneuvering for automatic ship collision avoidance,” *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 384–388, 2018, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.09.457.

[166] R. Skjetne, “The maneuvering problem,” Ph.D. dissertation, Mar. 2005.

[167] K. Granstrom, M. Baum, and S. Reuter, “Extended object tracking: Introduction, overview and applications,” *arXiv preprint arXiv:1604.00970*, 2016.

[168] P. Coscia, P. Braca, L. M. Millefiori, F. A. N. Palmieri, and P. Willett, “Multiple Ornstein–Uhlenbeck processes for maritime traffic graph representation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2158–2170, 2018. DOI: 10.1109/TAES.2018.2808098.

[169] S. Capobianco, N. Forti, L. M. Millefiori, P. Braca, and P. Willett, “Uncertainty-aware recurrent encoder-decoder networks for vessel trajectory prediction,” in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, 2021, pp. 1–5. DOI: 10.23919/FUSION49465.2021.9626839.

[170] F. H. Fridriksson, “Bayesian bridging distribution analysis for model based intent and long-term prediction of vessels,” M.S. thesis, NTNU, 2021.

[171] C. Van Loan, “Computing integrals involving the matrix exponential,” *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 395–404, 1978. DOI: 10.1109/TAC.1978.1101743.

[172] E. d’Afflisio, P. Braca, L. M. Millefiori, and P. Willett, “Detecting anomalous deviations from standard maritime routes using the Ornstein–Uhlenbeck process,” *IEEE Transactions on Signal Processing*, vol. 66, no. 24, pp. 6474–6487, 2018. DOI: 10.1109/TSP.2018.2875887.

[173] R. Grasso, L. M. Millefiori, and P. Braca, “Bayesian track-to-graph association for maritime traffic monitoring,” in *26th European Signal Processing Conference, EUSIPCO 2018, Roma, Italy, September 3-7, 2018*, IEEE, 2018, pp. 1042–1046. DOI: 10.23919/EUSIPCO.2018.8553443.

[174] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

ISBN 978-82-326-5751-3 (printed ver.)
ISBN 978-82-326-6431-3 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology