

Trym Tengedal

Uncertainty Management in a Scenario-Based MPC for Collision Avoidance

Master's thesis in Cybernetics and Robotics

Supervisor: Tor Arne Johansen

June 2019

Trym Tengesdal

Uncertainty Management in a Scenario-Based MPC for Collision Avoidance

Master's thesis in Cybernetics and Robotics
Supervisor: Tor Arne Johansen
June 2019

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

 **NTNU**
Norwegian University of
Science and Technology

Preface

This work concludes my Master of Science in Cybernetics and Robotics at NTNU. I would like to thank my supervisor Tor Arne Johansen and co-supervisors Edmund Brekke and Giorgio D. Kwame Minde for valuable discussions, guidance and information during the course of the spring semester.

Furthermore, the work atmosphere at G-238 this spring was very appreciating, with interesting discussions and conversations. Lastly, a thanks are in order, to my parents for giving motivation and support when the thesis progression was slow.

Problem Description

A Collision Avoidance (COLAV) System is essential for autonomous systems which are to operate in complex situations involving dynamic and static obstacles. The autonomous system will have to employ a tracking method to obtain information about the obstacles. Information uncertainty plays a crucial factor in this system, and will affect the COLAV performance.

This thesis has the aim of investigating methods for incorporating uncertainty in a COLAV system using the Scenario-Based Model Predictive Control (SBMPC) approach[19], and see if this yields better performance for the system. In maritime navigation, uncertainty has many sources. Uncertainty in the association of data to tracks from nearby vessels, uncertainty in the intent of nearby vessel captains in collision situations and vessel kinematic uncertainty are some of the inherent challenges which have to be dealt with in a good manner in Autonomous Surface Vehicles (ASVs). The thesis will mainly deal with kinematic uncertainty, as a first step in managing uncertainty in a COLAV system. The main goals of the thesis are the following.

- Perform a literature survey on tracking methods, COLAV algorithms and collision risk and probability estimation between the own-ship (defined as the ASV) and obstacles.
- Implement the Scenario-Based MPC in a simulator together with a tracking system
- Implement and validate a method for quantifying collision probability between two objects
- Modify the SB-MPC to also account for collision probability, using the chosen method. Furthermore, perform Monte Carlo simulations to validate the COLAV performance with the implemented modifications

Abstract

A robust Collision Avoidance (COLAV) system is of paramount importance in order to ensure safe operation of Autonomous Surface Vessels (ASVs). The COLAV system is highly dependent on a reliable tracking system to infer and predict the trajectories of nearby obstacles. Here, uncertainty becomes an important factor to be dealt with, as sensor systems are never perfect, nor are all vessel driven equally.

This thesis used obstacle state estimates and the corresponding uncertainty in the form of error covariances produced by a Kalman Filter, to estimate the probability of collision of the ASV with obstacles, using Monte Carlo integration and importance sampling. Two strategies were tested, where only the first was made to be a working approach. This strategy samples possible straight line obstacle trajectories and uses the fraction of the number of them crossing a defined safety zone around the ASV at the Closest Point of Approach (CPA) as the probability estimate. Decent results were obtained, but not feasible for real-time due to long computational times per probability evaluation. This was mainly because uncertainty in both position and velocity were considered, causing the need for sampling in 4 dimensions. Here, a minimum number of $n_{mc,int} = 1000$ samples should be drawn to have acceptably low variance in the probability estimate.

The collision probability evaluation method was further used in a probabilistic version of the Scenario-Based Model Predictive Control (SBMPC), where three different modifications to the latter method's cost function to account for collision probability were tested. Simulation results with the suggested modifications did not give any significant performance gain, but serves as a first-step in finding better ways of modifying the SBMPC to account for uncertainty.

Samandrag

Eit robust antikollisjonssystem er viktig for å sørge for sikker styring av autonome overflatefartøy. Dette antikollisjonssystemet er avhengig av eit påliteleg målfølgingsystem for å halde styr på hindringar i nærleiken. Her er usikkerheit ein viktig faktor som må handterast, ettersom ingen sensorsystem er perfekte, ei heller blir alle fartøy køyrt på samme måte.

Denne masteroppgåva brukte tilstandsestimat for hindringar med tilhøyrande usikkerhet representert som kovariansar, til å estimere sannsynet for kollisjon mellom det autonome overflatefartøyet og hindringane. Eit Kalmanfilter blei brukt for å produsere estimata og kovariansane. Monte Carlo- integrasjon og viktighetssampling blei brukt til utrekning av kollisjonssannsyna. To strategiar blei her testa, der bare den første var ein fungerande metode. Den fungerande strategien punktprøver moglege rettlinja banar hindringar kan ta, og brukar andelen av dei banane som krysser ein definert sikkerheitssone rundt det autonome overflatefartøyet ved det nærmaste punktet mellom to objekt med rettlinja rørsle (CPA), som eit estimat for kollisjonssannsynet. Greie resultat blei oppnådd, men ikkje gode nok til bruk av metoden i sanntid, på grunn av lang reknetid per evaluering av kollisjonssannsynet. Her burde eit minimum av $n_{mc,int} = 1000$ samplepunkt bli brukt for å få ein akseptabelt låg varians i estimatet av sannsynet.

Evalueringemetoden for å rekne ut kollisjonssannsynet blei vidare brukt i ein probabilistisk versjon av den scenariobaserte modellprediktive regulatoren (SBMPC), der tre forskjellige endringar på kostfunksjonen i den sistnevnte metoden, for å ta høgde for kollisjonssannsyn, blei testa. Simuleringsresultat med dei foreslåtte endringane gav ingen betydelege forbetringar for antikollisjonssystemet, men utgjer eit første skritt på vegen for å finne betre måtar å endre SBMPC-en på for å ta høgde for usikkerheit.

Contents

Preface	i
Problem Description	ii
Abstract	iii
Samandrag	iv
List of Tables	viii
List of Figures	ix
Nomenclature	xviii
1 Introduction	1
1.1 Motivation	1
1.2 Previous Work	2
1.3 Thesis outline	4
2 Simulator and GNC System	5
2.1 Simulation Environment	5
2.1.1 Own-ship Model	5
2.1.2 Constant Velocity Model	7
2.2 GNC System	9
2.2.1 The Control System	9
2.2.2 The Guidance System	10
3 COLREGS and SBMPC	13
3.1 International Regulations for Preventing Collision at Sea	13

3.1.1	Description of Rules	13
3.2	Model Predictive Control	16
3.3	Scenario-Based Model Predictive Control	17
4	Tracking Methods	25
4.1	The Kalman Filter	26
4.2	The PDAF	28
4.3	The IPDAF	31
4.4	The JIPDAF	32
4.5	The MHT	37
4.6	Remark	38
5	Collision Probability Evaluation Methods	39
5.1	Thesis Collision Probability Definition	40
5.2	Monte Carlo Integration	41
5.3	Probability Flow	43
5.4	Remarks	47
6	Collision Probability Validation	49
6.1	Simulation Setup	50
6.2	Example 1: Static Obstacle	50
6.2.1	Setup	50
6.2.2	Importance Sampling Scheme	53
6.2.3	Results	56
6.3	Example 2: Dynamic Obstacle 1	58
6.3.1	Setup	59
6.3.2	Importance Sampling Scheme	61
6.3.3	Results	64
6.4	Example 3: Dynamic Obstacle 2	67
6.4.1	Setup	67
6.4.2	Importance Sampling Scheme	68
6.4.3	Results	68
6.5	Discussion	69
7	A Probabilistic SBMPC	71
7.1	SBMPC Cost Function Modifications	71
7.2	Simulation	72

7.2.1	Setup	72
7.2.2	Results Modification 1	76
7.2.3	Results Modification 2	86
7.2.4	Results Part 3	95
7.3	Discussion	104
8	Conclusion and Future Work	107
8.1	Conclusion	107
8.2	Future Work	108
	References	111

List of Tables

2.1	Own-ship controller gains.	10
2.2	Guidance law parameters.	11
3.1	SBMPC parameters used in the thesis.	23
6.1	Safety corridor parameters.	52
6.2	Average computation time \bar{t}_{π_3} per collision probability evaluation for the strategy $\pi_3(\mathbf{x}, t)$, for the different values of $n_{mc,int}$, with $N_{MC} = 10$	67
7.1	Parameters used in the thesis for the two versions of the SBMPC.	75

List of Figures

3.1	Overtaking situation, with green as the overtaking vessel. The correct behavior for the overtaking vessel is shown by the dashed arrows.	14
3.2	Head-on situation, with the correct behavior for the vessels shown through the dashed arrows.	15
3.3	Crossing situation, with the correct behavior for the give-way vessel shown with the dashed arrow.	15
3.4	Illustration of the MPC principle [11].	16
3.5	Summary of the collision avoidance control algorithm [19].	18
3.6	Block diagram illustrating the information flow between the main modules in the system [19].	19
4.1	JPDF state estimation update where the new position estimate of target 1 $\hat{z}_{k k}^1$ is weighted based on the distance from the validated measurements z_k^j to the predicted position $\hat{z}_{k k-1}^1$ and the possibility of them originating from another target. Notice that the closest measurement z_k^3 is weighted significantly less than the other measurements. This is because this measurement is much closer to another target estimate $\hat{z}_{k k-1}^2$ [17].	33
5.1	Illustration of the combined probability ellipse and the combined safe separation zone: The combined probability ellipse is obtained through coordinate transformation. The combined safe separation zone is denoted by a circle whose radius is the sum of the safe separation distances of the own and target ships[32].	45

6.1	Illustration of the safety corridor from the own-ship (blue) along a straight line, with the uncertainty ellipse of an obstacle (green) included (assumed to be covering the entire probability mass). The conflict zone is here shown as the filled grey area.	51
6.2	Illustration of the first example, with the own-ship in blue and obstacle drawn in green with a sample orientation of $\psi_1 = \frac{\pi}{2}$. Moreover, a sample 3σ (3 standard deviations) probability ellipse of the obstacle positional uncertainty is shown in black. The 1σ probability ellipse of the multivariate Gaussian $\pi_1(\mathbf{x}, t)$ used for random sampling in the first method is indicated in blue. The other sampling method $\pi_2(\mathbf{x}, t)$ will have constant density in the y_b direction, and equal variance as $\pi_1(\mathbf{x}, t)$ in the x_b direction.	55
6.3	Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the first importance sampling method $\pi_1(\mathbf{x}, t)$	56
6.4	Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the second importance sampling method $\pi_2(\mathbf{x}, t)$	57
6.5	Collision probability $\mathbb{P}_{c,k}^i$ comparison for the two importance strategies, when $N_{MC} = 1000$ and $n_{mc,int} = 50000$	57
6.6	Samples generated by the two importance strategies $\pi_1(\mathbf{x}, t)$ (MVG Sampling) and $\pi_2(\mathbf{x}, t)$ (U (Uniform) + Scalar G (Gaussian) Sampling) with $n_{mc,int} = 50000$, referenced to the body frame. The red cross marks $\boldsymbol{\mu}_{ci}^b$, and the blue cross marks the obstacle position. The blue ellipse is the 3σ probability ellipse for the obstacle.	58
6.7	North-East plot of the obstacle (green) path (also green), with the own-ship plotted in blue, with its safety zone of radius d_{safe} in purple. The obstacle is plotted at the CPA, directly below the own-ship, in addition to at its position at the end of the simulation.	59

- 6.8 Illustration of the attempted more effective sampling scheme for Example 2, with the own-ship in blue and obstacle drawn in green, with the circular safety zone of radius d_{safe} drawn around the own-ship. The obstacle velocity vector \mathbf{v}^i , importance function velocity expectation \mathbf{v}_{eff}^i and their difference \mathbf{v}_{diff} are also indicated, drawn from the tip of the obstacle, instead of the point mass center. The 1σ probability ellipse of the obstacle position uncertainty is shown, and a sampled trajectory starting at a position (x_s, y_s) with velocity vector \mathbf{v}_s is drawn. An example standard deviation in velocity for the particular importance sample is shown for the velocity vector \mathbf{v}_s , in \mathbf{v}_1 and \mathbf{v}_2 , which is calculated based on the difference velocity \mathbf{v}_{diff} 63
- 6.9 Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the third importance sampling method $\pi_3(\mathbf{x}, t)$ 65
- 6.10 Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the fourth importance sampling method $\pi_4(\mathbf{x}, t)$ 65
- 6.11 Sampling plot in position and velocity for the fourth importance strategy $\pi_4(\mathbf{x}, t)$ in the NED frame. The 3σ probability ellipse for the importance function is shown in red, and the same for the obstacle in blue. The red- and blue crosses mark their expectations, respectively. For the velocity sampling, an example integration region of where the samples are valid, i.e. nonzero, is shown in the orange ellipse. For the position sampling, the importance strategy uses the same covariance and expectation as the obstacle's, and the two probability ellipses thus overlap. 66
- 6.12 North-East plot of the obstacle path (both in green), with the own-ship (blue) path plotted in black, with its position and safety zone of radius d_{safe} in purple drawn at the CPA. The obstacle and own-ship are plotted at the CPA in addition to at their end positions. The red line marks the own-ship planned path. 68
- 6.13 Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the third importance sampling method $\pi_3(\mathbf{x}, t)$ 69

7.1	Overtaking situation with cost function alteration (M1). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.	77
7.2	Overtaking situation with cost function alteration (M1). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.	78
7.3	Overtaking situation with cost function alteration (M1). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	78
7.4	Overtaking situation with cost function alteration (M1). Track estimates for the obstacle, versus the true motion.	79
7.5	Overtaking situation with cost function alteration (M1). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.	79
7.6	Head-on situation with cost function alteration (M1). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.	80
7.7	Head-on situation with cost function alteration (M1). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.	81
7.8	Head-on situation with cost function alteration (M1). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	81

7.9	Head-on situation with cost function alteration (M1). Track estimates for the obstacle, versus the true motion.	82
7.10	Head-on situation with cost function alteration (M1). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m. . . .	82
7.11	Crossing situation with cost function alteration (M1). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship. . . .	83
7.12	Crossing situation with cost function alteration (M1). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown. . . .	84
7.13	Crossing situation with cost function alteration (M1). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	84
7.14	Crossing situation with cost function alteration (M1). Track estimates for the obstacle, versus the true motion.	85
7.15	Crossing situation with cost function alteration (M1). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m. . . .	85
7.16	Overtaking situation with cost function alteration (M2). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship. . . .	86
7.17	Overtaking situation with cost function alteration (M2). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown. . . .	87

7.18	Overtaking situation with cost function alteration (M2). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	87
7.19	Overtaking situation with cost function alteration (M2). Track estimates for the obstacle, versus the true motion.	88
7.20	Overtaking situation with cost function alteration (M2). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.	88
7.21	Head-on situation with cost function alteration (M2). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.	89
7.22	Head-on situation with cost function alteration (M2). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.	90
7.23	Head-on situation with cost function alteration (M2). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	90
7.24	Head-on situation with cost function alteration (M2). Track estimates for the obstacle, versus the true motion.	91
7.25	Head-on situation with cost function alteration (M2). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.	91
7.26	Crossing situation with cost function alteration (M2). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.	92

7.27	Crossing situation with cost function alteration (M2). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown. . . .	93
7.28	Crossing situation with cost function alteration (M2). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	93
7.29	Crossing situation with cost function alteration (M2). Track estimates for the obstacle, versus the true motion.	94
7.30	Crossing situation with cost function alteration (M2). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m. . . .	94
7.31	Overtaking situation with cost function alteration (M3). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship. . . .	96
7.32	Overtaking situation with cost function alteration (M3). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown. . . .	96
7.33	Overtaking situation with cost function alteration (M3). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	97
7.34	Overtaking situation with cost function alteration (M3). Track estimates for the obstacle, versus the true motion.	97
7.35	Overtaking situation with cost function alteration (M3). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.	98

7.36	Head-on situation with cost function alteration (M3). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path when the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.	99
7.37	Head-on situation with cost function alteration (M3). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown. . . .	99
7.38	Head-on situation with cost function alteration (M3). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	100
7.39	Head-on situation with cost function alteration (M3). Track estimates for the obstacle, versus the true motion.	100
7.40	Head-on situation with cost function alteration (M3). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m. . . .	101
7.41	Crossing situation with cost function alteration (M3). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path when the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.	102
7.42	Crossing situation with cost function alteration (M3). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown. . . .	102
7.43	Crossing situation with cost function alteration (M3). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.	103

7.44 Crossing situation with cost function alteration (M3). Track estimates for the obstacle, versus the true motion. 103

7.45 Crossing situation with cost function alteration (M3). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m. . . . 104

Nomenclature

Acronyms

AIS Automatic Identification System

ASV Autonomous Surface Vessel

COLREGS International Regulations for Preventing Collisions at Sea

CPA Closest Point of Approach

CVM Constant Velocity Model

DOF Degrees of Freedom

GNC Guidance, Navigation and Control

ICP Instantaneous Collision Probability

IPDAF Integrated Probability Data Association Filter

JIPDAF Joint Integrated Probability Data Association Filter

JPDAF Joint Probability Data Association Filter

KF Kalman Filter

LOS Line of Sight

MHE Multiple Hypothesis Tracker

MM Multiple Model

MVG Multivariate Gaussian

NED North-East-Down

PDAF Probabilistic Data Association Filter

PDF Probability Density Function

PSBMPC Probabilistic Scenario-Based Model Predictive Control

SBMPC Scenario-Based Model Predictive Control

VO Velocity Obstacle

COLAV Collision Avoidance

Chapter 1

Introduction

1.1 Motivation

Collisions between vehicles occur every year, and can cause significant human casualties, environmental damage and destruction of property. In the maritime sector, the European Maritime Safety Agency reported a total of 14002 casualty events with a ship from 2011 to 2017, in which collisions represented 23.2% (3248) of these [10].

In these maritime accidents, humans are estimated to be the main cause of collision 80% of the time [26]. According to [7] and [32], operational factors such as lack of experience, competency and situational awareness are detrimental here. Despite the emergence of advanced navigational aid systems such as radar, Automatic Identification System (AIS), GPS, radio and navigational sensors, there is not any evident decrease in maritime accidents [10][39][1].

This gives incentives for the employment of autonomous systems, which can operate on land, at sea and in the air. An autonomous system is a complex and intelligent self-governing agent, which through sensory input from its environment can execute various tasks as commanded. In recent years, low cost sensory equipment, big data and gradually more effective data processors have made it possible to realize these systems at a higher degree. By eliminating the human factor, there is a large room for improvements in the safety aspect. However, large challenges yet remain for these systems to be trusted and integrated into our daily lives [36][41].

For Autonomous Surface Vessels (ASVs) [37], the operation is dependent on

a highly reliant and fault tolerant system for Guidance, Navigation and Control (GNC). One of the most important aspects of the ASV is the Collision Avoidance (COLAV) system, which is meant to ensure that the vessel does not collide with surrounding obstacles in its path. Moreover, the avoidance maneuvers should comply with The International Regulations for Preventing Collisions at Sea (COLREGS) [18], which defines a set of rules to follow depending on the type of situation.

Collision avoidance for ships has been a research topic since the Second World War [42], and far from all challenges have been solved. One of the main challenges is related to quantifying the risk and uncertainty for collision between vessels. A tracking system using for instance radar equipment is typically employed for the ASV, to monitor the positions of nearby vessels. However, as there is no such thing as a perfect sensor, uncertainty will be inherent in the application. In addition, as a vessel can be driven in many ways, there will be uncertainty related to the intent of the ship commander in a collision situation, and whether or not the vessel follows COLREGS. This thesis attempts to incorporate uncertainty, typically from a tracking system, into a COLAV system, and test if this can improve the performance of the system.

1.2 Previous Work

Numerous approaches for solving the COLAV problem in maritime navigation have been suggested in recent years, for instance based on evolutionary algorithms [8], neural networks [40] and fuzzy logic [24]. However, these approaches suffer from bad scalability to situations with increasing number of vessels, and have only been tested in a simulation environment [21].

State of the art maritime COLAV algorithms today with COLREGS compliance, which have been tested in real-time, includes the Velocity Obstacle (VO) method [21], the Scenario-Based Model Predictive Control (SBMPC) [19][15] and grid based path planners using A-star (A^*) search [38]. The first method computes a cone (the VO) in the velocity space of the ASV, and selects an own-ship (the ASV) velocity outside of the cone in order to avoid collision. The second method evaluates a cost function which penalizes being too close to nearby vessels, breaching COLREGS, and large deviations from the planned path, for different control behaviours, selecting the one which yields the minimal cost. The third method incorporates collision avoidance in a lattice-based path planner

using A* search [16], which uses the collision probability with nearby vessels in its cost function.

ASVs with collision avoidance need to have a tracking system to get information about the trajectories of nearby vessels. When multiple measurements are received by the sensor system, false measurements becomes an apparent problem, in addition to the problem of associating the measurements obtained at any given time to different tracks/targets. For cases where tracks are well separated, Probabilistic Data Association Filters (PDAF) [4], which is a single-target tracking method, can be used for each track. This filter calculates the probabilities of associating any of the obtained measurements to a track at a given time, and uses these probabilities to weigh the correction term in a Kalman Filter [20]. This method is in general sub-optimal, because it assumes for instance that the track (target) exists and is initialized, and that all past information about the track can be summarized as a Gaussian posterior distribution.

Another caveat with the PDAF, is that it assumes that all non-target measurements origin from false measurements which are uniformly spatially distributed inside the surveillance region of the sensor system, i.e random inference. In cases where tracks coalesce, due to targets approaching each other, this assumption will not hold, and we have persistent interference. The Joint Probability Data Association Filter (JPDAF) solves this problem by calculating the measurement-to-target association probabilities jointly for multiple targets, when assuming a known number of targets [2].

Further, the Integrated Probability Data Association Filter (IPDAF) [28] and Joint Integrated PDAF(JIPDAF) [27] for single- and multi-target tracking, respectively, extends the PDAF/JPDAF to include target existence probabilities, using Markov chains. An alternative to the JPDAF/JIPDAF, is the Multiple Hypothesis Tracker (MHT) [35], which considers all possible sequences of measurements that could origin from a target. At each time sample, the MHT considers three possibilities for each measurement; that the measurement originated from a previous track, a new track, or is a false measurement. This method thus includes track initiation into the algorithm, which is not the case for the JPDAF.

The problem of quantifying risk and probability related to collisions has been an active research field for a long time, in both the maritime and air traffic sector. Some previous work on this topic in the marine sector have focused on using Quantitative Risk Assesment (QRA) models to estimate the frequency and consequence of collisions [13][25], which use historical data,

human error statistics and ship dynamics to perform the estimation. Other work have assessed collision risk by using the closest point-, time- or distance of approach between two vessels [22][9]. The above mentioned methods suffer from the fact that no explicit representation of vessel position uncertainty is included in the assessments.

Recent, more advanced methods for air and maritime traffic, consider the positional uncertainty of the own-ship and nearby vessels, in addition to a pre-set safe distance between vessels, and use this to quantify collision probability [32]. The probability is obtained by integrating the probability distribution over a conflict zone, which is the overlap between the combined multivariate position uncertainty ellipse (in practice the probability distribution) and combined safety zone of two vessels. The integration is usually approximated, due to its complexity. For air traffic, the integration is approximated numerically in [23] and analytically with different assumptions in for instance [33][31].

In this thesis, the collision probability will be calculated using Monte Carlo integration [6], which is an importance sampling method that uses the probabilistic definition of a variable or function's expectation to find an approximation of the integral. The probability evaluation will also consider uncertainty in the obstacle velocity, which is not accounted for in most of the above previous work [32][23][33][31]. The collision probability estimate will further be used to modify the SBMPC method in order to account for obstacle tracking uncertainty.

1.3 Thesis outline

The thesis is organized as follows. Chapter 2 describes the GNC-system and simple simulator used as a basis for implementing the tracking system and COLAV-algorithm. Chapter 3 will provide necessary information about the COLREGS and the COLAV method used in this thesis. Chapter 4 introduces different tracking methods to be used in conjunction with the COLAV system. Chapter 5 introduces two ways of quantifying collision probability, where Chapter 6 uses one of the introduced methods to validate the collision probability for different examples. Chapter 7 suggest several modifications to be made on the COLAV method in order to account for collision probabilities, and compares the performance for the COLAV system with these modifications to the original method in simulations. Lastly, Chapter 8 concludes the thesis and presents suggestions for future work.

Chapter 2

Simulator and GNC System

This chapter provides information about the models used for simulating collision avoidance scenarios (the simulation environment), in addition to a description of the Guidance, Navigation and Control system for the own-ship (the ASV), used as a foundation for implementing the COLAV-system in this thesis.

2.1 Simulation Environment

The simulator uses a 3 Degrees of Freedom (DOF) model for the own-ship vessel model and Constant Velocity Models (CVM) for nearby vessels (obstacles). These models are described in Section 2.1.1 and 2.1.2 below, respectively.

2.1.1 Own-ship Model

The 3DOF own-ship model describes the motion of a displacement vessel in surge, sway and yaw (horizontal motion), and is based on the work in [14]. The Society of Naval Architects and Marine Engineers (SNAME) notation is used for forces, moments, velocities and angles, with equations of motion in vectorial form as in [12]. The model derivation follows the same lines as in [14].

The vessel position in the North-East-Down (NED) coordinate system is given by $\boldsymbol{\eta} = [x \quad y \quad \psi]^T$, whereas its velocity in the BODY-fixed coordinate system is given as $\boldsymbol{v} = [u \quad v \quad r]^T$. The vector $\boldsymbol{\tau} = [X \quad Y \quad N]^T$ describes the generalized forces and moments affecting the ship in surge, sway and yaw. The equations of motion for the own-ship can then be represented in vectorial

form as [12]

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{v} \quad (2.1a)$$

$$\mathbf{M}\dot{\boldsymbol{v}} + \mathbf{C}(\boldsymbol{v})\boldsymbol{v} + \mathbf{D}(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau} + \boldsymbol{w} \quad (2.1b)$$

where $\mathbf{R}(\psi)$ is the rotation matrix from the NED frame $\{n\}$ to the BODY frame $\{b\}$, and is for the horizontal 3DOF model given as

$$\mathbf{R}(\psi) = \mathbf{R}_b^n = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

This matrix thus represents a rotation of ψ about the NED frame z-axis. The inertia matrix \mathbf{M} consists of the rigid-body mass of the vessel, in addition to the added virtual mass due to the vessel having to displace surrounding fluid as it moves through it [12]: $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$. The rigid body mass matrix will be given as

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (2.3)$$

where m and I_z are the vessel mass and moment of inertia about the body z-axis, respectively. Similarly, the coriolis and centripetal matrix $\mathbf{C}(\boldsymbol{v})$ can be decomposed into a rigid body part and an added mass part: $\mathbf{C}(\boldsymbol{v}) = \mathbf{C}_{RB}(\boldsymbol{v}) + \mathbf{C}_A(\boldsymbol{v})$. The rigid body coriolis matrix is given as

$$\mathbf{C}_{RB}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & -m\boldsymbol{v} \\ 0 & 0 & m\boldsymbol{u} \\ m\boldsymbol{v} & -m\boldsymbol{u} & 0 \end{bmatrix} \quad (2.4)$$

The added mass part for the inertia matrix \mathbf{M} and coriolis matrix $\mathbf{C}(\boldsymbol{v})$ were set to zero in [14], and is also done here.

Further, the damping term in the model (2.1) can be decomposed into a linear and a nonlinear part: $\mathbf{D}(\boldsymbol{v})\boldsymbol{v} = \mathbf{D}_L\boldsymbol{v} + \mathbf{D}_{NL}(\boldsymbol{v})\boldsymbol{v}$. The linear damping matrix \mathbf{D}_L is given as

$$\mathbf{D}_L = - \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \quad (2.5)$$

where the matrix element values are the linear damping coefficients in each degree of freedom, for each motion (surge, sway and yaw). The nonlinear damping part is given by

$$\mathbf{D}_{NL}(\mathbf{v})\mathbf{v} = - \begin{bmatrix} X_{|u|u}|u|u + X_{uuu}u^3 \\ Y_{|v|v}|v|v + X_{rrr}v^3 \\ N_{|r|r}|r|r + X_{rrr}r^3 \end{bmatrix} \quad (2.6)$$

The generalized force vector $\boldsymbol{\tau}$ can be written as a function of the forces in the body x- and y-direction, as

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ l_r F_y \end{bmatrix} \quad (2.7)$$

where F_x and F_y are the forces produced by the own-ship actuators in the x- and y-direction, respectively. l_r is the moment arm between the body z-axis and the y-direction force's point of action.

Lastly, the environmental disturbance vector \mathbf{w} is neglected in this thesis, as is done in [14]. The parameters for this 3DOF model are the same as those used in [14].

2.1.2 Constant Velocity Model

The trajectories of nearby obstacles (vessels) are often predicted by straight line motion with constant velocity [19][4][35][2]. The model also may include some process noise in order to take into account the uncertainty of the prediction. This is typically done in a tracking system, but control methods such as the SB-MPC will typically ignore this and use a deterministic model. In the following, the general CVM with process noise and a measurement model will be presented.

The state vector for the CVM used to model the kinematics of obstacle i is here given as $\mathbf{x}^i = [x^i \quad V_x^i \quad y^i \quad V_y^i]^T$, where x^i and y^i are the north -and east

coordinates, and V_x^i and V_y^i are the north -and east velocities, respectively. The position of the vessel is assumed measured, such that the model in discrete time for obstacle i is given as

$$\mathbf{x}_{k+1}^i = \mathbf{F}\mathbf{x}_k^i + \mathbf{v}_k^i \quad (2.8a)$$

$$\mathbf{z}_k^i = \mathbf{H}\mathbf{x}_k^i + \mathbf{w}_k^i \quad (2.8b)$$

where \mathbf{F} and \mathbf{H} are the transition and measurement matrix, respectively. The vectors \mathbf{v}_k^i is the process noise affecting obstacle i , and \mathbf{w}_k^i is the measurement noise affecting the measurement at discrete time instant t_k . Lastly, the vector \mathbf{z}_k^i contains the noise corrupted position measurement of obstacle i at time t_k for instance obtained through a radar system or GPS. The transition matrix \mathbf{F} is given as

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

where $T = t_{k+1} - t_k$ is the sampling time for the linear prediction, and could be time varying. The measurement matrix \mathbf{H} is given as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.10)$$

The process noise and measurement noise are assumed to be zero mean, white, mutually independent and Gaussian with known covariance matrices \mathbf{Q} and \mathbf{R} , respectively. For this model, the process noise covariance is given as

$$\mathbf{Q} = \sigma_a^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \quad (2.11)$$

where the process noise strength σ_a is chosen based on the expected maneuverability of the vessel [6]. Note that other prediction- and measurement models are also possible, for instance the constant turn rate model [5] or the constant

acceleration model [3] for prediction. In this thesis, the CVM with process noise and a measurement model as in (2.8) is used in the tracking system, whereas a deterministic version with full state knowledge is used in the SB-MPC prediction.

2.2 GNC System

A system for guidance, navigation and control enables the automatic control of devices that move under water, on the surface or in space [12]. 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.

In this thesis, the own-ship position and velocity are assumed known through the vessel model from the previous Section. Thus, the navigation part of the GNC system entails using the vessel model (2.1) to get the current state of the own-ship. Therefore, only the control system and guidance system are described in the following two subsections.

2.2.1 The Control System

The own-ship is assumed to have thrusters which produce forces F_x and F_y in the body x -and y-direction. The controllers implemented in this thesis are the same as in [14]. A feedback linearizing controller

$$F_x = -mvr - (X_u + X_{|u|u}|u| + X_{uuu}u^2)u + K_{p,u}m(u_d - u) \quad (2.12)$$

is used to control the surge speed u of the own-ship, where u_d is the desired surge speed. The heading ψ is the other controlled variable, using the PD controller

$$F_y = I_z \frac{K_{p,\psi}}{I_r} ((\psi_d - \psi) - K_{d,\psi}r) \quad (2.13)$$

The controller gains are given in Table 2.1, and differ somewhat from [14] due to different tuning. The tuning is mostly based on trial and error.

Table 2.1: Own-ship controller gains.

Parameter	Value	Unit
$K_{p,u}$	5.0	s^{-1}
$K_{p,\psi}$	5.0	s^{-2}
$K_{d,\psi}$	2.0	s

2.2.2 The Guidance System

Line of Sight (LOS) guidance[12] is used to keep the own-ship on the desired path, taking in the desired path in the form of waypoints, and outputting the reference course for the autopilot control system. The LOS law tries to minimize the cross-track error, which is the distance from the vessel to the path, perpendicular to the path.

For piecewise straight line paths, which are considered here, the active straight line segment between two waypoints is considered in the LOS law. Given two waypoints in the NED frame, consisting of their Cartesian coordinates: $\mathbf{p}_k^n = [x_k \ y_k]^T$ and $\mathbf{p}_{k+1}^n = [x_{k+1} \ y_{k+1}]^T$, the first step in the LOS law is to find the path-tangential angle α_k between north and the path segment given by the two waypoints:

$$\alpha_k = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \quad (2.14)$$

where k here is an index corresponding to the current waypoint, such that $k + 1$ is the next waypoint to be reached by the vessel. atan2 is the four-quadrant inverse tangent function. This 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

$$\boldsymbol{\epsilon}(t) = \begin{bmatrix} s(t) \\ e(t) \end{bmatrix} = \mathbf{R}_p(\alpha_k)^T (\mathbf{p}^n(t) - \mathbf{p}_k^n) \quad (2.15)$$

where

$$\mathbf{R}_p = \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix} \quad (2.16)$$

The goal of the path-following is thus to achieve $e(t) = 0$, as this means that

the vessel has converged onto the straight line segment. For the LOS-law, the desired course, which ensures vessel convergence onto the path, is computed as

$$\chi_d(e) = \alpha_k + \chi_r(e) \quad (2.17)$$

where

$$\chi_r(e) = \text{atan} \left(\frac{-e}{\Delta} \right) \quad (2.18)$$

is the velocity-path relative angle. atan is the inverse tangent returning values in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$. The lookahead distance $\Delta > 0$ determines how aggressive the steering towards the path becomes. Further, as the vessel approaches the next waypoint \mathbf{p}_{k+1}^n , the inequality

$$(x_{k+1} - x(t))^2 + (y_{k+1} - y(t))^2 \leq R_a^2 \quad (2.19)$$

is used to switch between straight line segments, i.e. incrementing the current waypoint \mathbf{p}_k^n to \mathbf{p}_{k+1}^n . Here, R_a is the radius of acceptance, whose circle the vessel has to be within for the switching to activate. The LOS law parameters were here chosen to the values given in Table 2.2, again mostly based on trial and error.

Table 2.2: Guidance law parameters.

Parameter	Value	Unit
Δ	100.0	m
R_a	20.0	m

As the environmental disturbance vector \mathbf{w} were assumed neglected in the vessel model from Section 2.1.1, the desired heading ψ_d for the heading controller in Section 2.2.1 is taken directly as χ_d here. If this assumption did not hold in the simulator, crab-angle compensation or a LOS law with integral action could be implemented to account for non-zero environmental disturbances [12].

Chapter 3

COLREGS and SBMPC

This chapter provides the necessary background regarding the relevant COLREGS rules and the COLAV algorithm SBMPC, used in this thesis. A small section about Model Predictive Control is also given, for the readers unfamiliar with optimization based control.

3.1 International Regulations for Preventing Collision at Sea

The International Regulations for Preventing Collision at Sea (COLREGS) define a set of "rules for the road" for vessels to follow, and was published by the International Maritime Organization in 1972 [18]. The rules in the COLREGS from Section B about steering and sailing have most relevance for this thesis, and these will be briefly described here. Only rules 8(b and d), 13, 14, 15, 16 and 17 will be introduced. These rules specify actions for a vessel to take in different collision situations, depending on its classification as a give-way or stand-on vessel.

3.1.1 Description of Rules

The following descriptions are mostly taken from [18].

Rule 8 - Action to Avoid Collision

Actions taken by a vessel in order to avoid collision should be made in good time and be clearly visible (8b), and the action should result in passing at a safe distance from the other vessel (8d).

Rule 13 - Overtaking

A vessel A is said to be overtaking if it is approaching another vessel B from a direction more than 22.5 degrees abaft of vessel B. Moreover, the overtaking vessel shall keep out of the way of the vessel being overtaken. The situation is depicted in Figure 3.1.

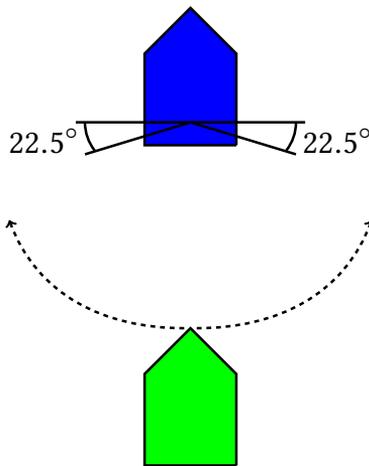


Figure 3.1: Overtaking situation, with green as the overtaking vessel. The correct behavior for the overtaking vessel is shown by the dashed arrows.

Rule 14 - Head-on Situation

If two vessels see each other ahead or nearly ahead and are approaching each other, the vessels shall alter their course to starboard such that the other vessel is passed on the port side. This is illustrated in Figure 3.2.

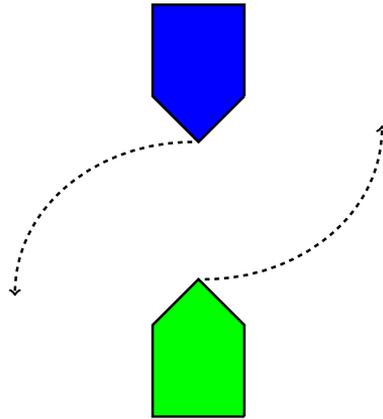


Figure 3.2: Head-on situation, with the correct behavior for the vessels shown through the dashed arrows.

Rule 15 - Crossing Situation

When two vessels are in a crossing situation, the vessel which has the other on her starboard side shall keep out of the way, and avoid crossing ahead of the other vessel if the circumstances of the situation allow it. The situation is illustrated in Figure 3.3.

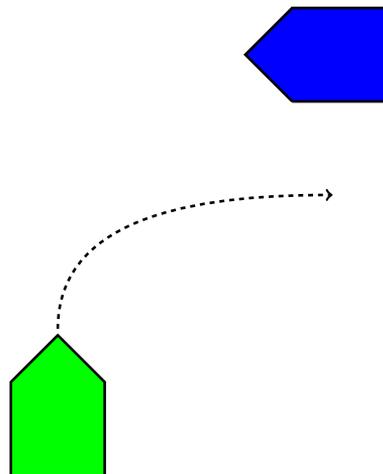


Figure 3.3: Crossing situation, with the correct behavior for the give-way vessel shown with the dashed arrow.

Rule 16 - Actions by Give-way Vessel

Vessels which are directed to keep out of the way of other vessels shall take, if possible, early and substantial actions to keep well clear of the other vessels.

Rule 17 - Actions by Stand-on Vessel

One of the two vessels in a possible collision situation shall keep out of the way of the other (give-way vessel), whereas the other shall keep her course and speed (stand-on vessel). The stand-on vessel may take action to avoid collision if it becomes apparent that the give-way vessel does not do so according to the Rules.

3.2 Model Predictive Control

Model Predictive Control (MPC) is an optimization-based technique which employs a plant model to solve a finite horizon open loop control problem using the current plant state or measurement as initial condition. This results in an optimal input sequence for the horizon from t' to $t' + N$, where t' is the current time and N is the prediction horizon. Here, the first control input in this sequence is used at the current time as input to the system. This optimization is then performed at each sampling instant. The method is displayed in Figure 3.4.

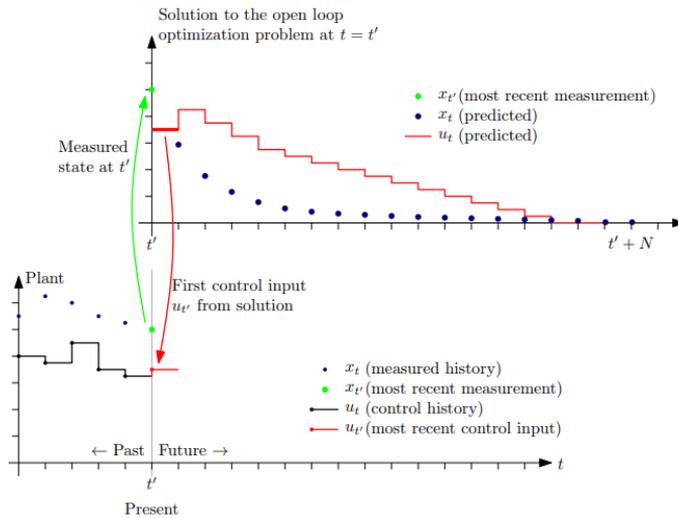


Figure 3.4: Illustration of the MPC principle [11].

3.3 Scenario-Based Model Predictive Control

The SBMPC [19][15] is a COLAV algorithm based on MPC which evaluates a cost function for a finite set of own-ship control behaviors or so-called scenarios, and chooses the scenario which yields the minimum cost over all nearby obstacles and over the entire prediction horizon. Models of the own-ship and obstacle kinematics are used as equality constraints for the prediction part, typically the ones described in the previous Chapter, from Section 2.1. The deterministic version of the Constant Velocity Model (CVM) is used for predicting the obstacle motion, i.e. the measurement model and process noise are disregarded, and the state is propagated using only the transition matrix F :

$$\mathbf{x}_{k+1}^i = F\mathbf{x}_k^i \quad (3.1)$$

The LOS-law from Section 2.2.2 is also used in the prediction. Alternatively, a transitional cost could be included in the SBMPC to fully decouple the guidance strategy from the COLAV system, as in [15]. Moreover, effects from wind, waves and ocean current can also be taken into account, but will not be done here.

The control behaviors specify offsets in desired course and surge speed from the nominal references as produced by the guidance system, and the optimal offsets are chosen in order to minimize the risk of collision, path deviation and in addition comply to the COLREGS. The SBMPC thus evaluates the hazard for each scenario and obstacle over the prediction horizon. The algorithm is summarized in Figure 3.5.

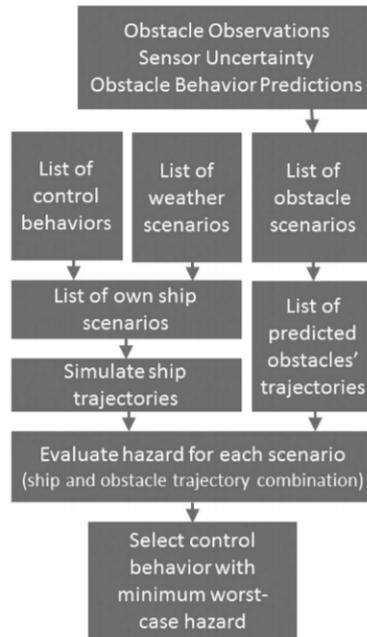


Figure 3.5: Summary of the collision avoidance control algorithm [19].

Architecture

The position of the SBMPC COLAV method in the GNC-system for the ASV is shown in Figure 3.6, which shows an apparent decoupling between the COLAV system and the mission planning and steering parts of the ASV. This allows 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 possibly with associated error covariances from the tracking system (see Chapter 4), own-ship navigational information and references for course and speed from the mission planner. The outputs consists of offsets χ_m and u_m to course and speed, respectively. The modified references $\chi_c = \chi_d + \chi_m$ and $u_c = u_d + u_m$ are fed into the ASV autopilot, where χ_d and u_d are the original course and surge speed references from the mission planner, respectively.

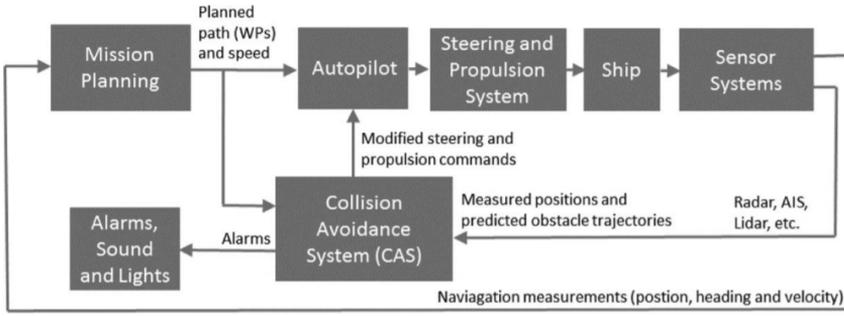


Figure 3.6: Block diagram illustrating the information flow between the main modules in the system [19].

Control Behaviors

The set of control behaviors typically used, are given below [19]

- Course offsets $\chi_m \in \{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\}$ degrees.
- Full stop in forward speed, slow forward and nominal propulsion (keep current speed), i.e $u_m \in \{0, 0.5, 1\}$.

which sums up to $13 \cdot 3 = 39$ scenarios. Ideally, one should investigate all control behaviors at each sample time in the SBMPC 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 5 changes in control behavior over the prediction horizon, the amount of scenarios to evaluate grows to $39^5 = 90224199$. Thus, the control behavior is therefore assumed fixed over the horizon.

Risk of Collision

The hazard of collision is included in the cost function, and is given by the collision risk factor $\mathcal{R}_i^k(t)$ multiplied by the cost of collision $C_i^k(t)$ associated with obstacle i in scenario k at time t . The risk factor is defined as

$$\mathcal{R}_i^k(t) = \begin{cases} \frac{1}{|t-t_0|^p} \left(\frac{d_i^{safe}}{d_{0,i}^k(t)} \right)^q, & \text{if } d_{0,i}^k(t) \leq d_i^{safe}. \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

where t_0 is the current time and $t > t_0$ is the current time in the prediction. The exponent $q \geq 1$ and distance d_i^{safe} are chosen large enough such that COLREGS Rule 16 from 3.1.1 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 $C_i^k(t)$ is given by

$$C_i^k(t) = K_i^{coll} \left\| \mathbf{v}_0^k(t) - \mathbf{v}_i^k(t) \right\|^2 \quad (3.3)$$

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, possibly dependent on the type of obstacle and its size.

COLREGS Satisfiability

In order to penalize breaching the COLREGS, one must first determine the type of collision situation, given that the nearby obstacle is within a predefined distance d_{close} . To comply to the COLREGS, the rules from Section 3.1.1 must be satisfied. The following inequality tests are used to determine the situation at time t in scenario k , which is given by the boolean variables CLOSE, OVERTAKEN, STARBOARD, HEAD-ON and CROSSED:

- An obstacle i is said to be CLOSE if

$$d_{0i}^k(t) \leq d_{close} \quad (3.4)$$

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

$$\mathbf{v}_0^k(t)^T \mathbf{v}_i^k(t) > \cos(68.5^\circ) \left\| \mathbf{v}_0^k(t) \right\| \cdot \left\| \mathbf{v}_i^k(t) \right\| \quad (3.5)$$

and in addition the obstacle is CLOSE and has higher velocity than the own-ship, i.e. $\left\| \mathbf{v}_0^k(t) \right\| < \left\| \mathbf{v}_i^k(t) \right\|$. Here, $\mathbf{v}_0^k(t)$ and $\mathbf{v}_i^k(t)$ are the velocity vectors of the own-ship and obstacle i in the horizontal plane, respectively.

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

$$\angle L_i^k(t) \geq \psi^k(t) \quad (3.6)$$

where $L_i^k(t)$ is the unit LOS-vector from the own-ship to obstacle i , and $\psi^k(t)$ is the own-ship heading.

- Obstacle i is said to be HEAD-ON if

$$\mathbf{v}_0^k(t)^T \mathbf{v}_i^k(t) < -\cos(22.5^\circ) \left\| \mathbf{v}_0^k(t) \right\| \cdot \left\| \mathbf{v}_i^k(t) \right\| \quad (3.7a)$$

$$\mathbf{v}_0^k(t)^T \mathbf{L}_i^k(t) > -\cos(15^\circ) \left\| \mathbf{v}_0^k(t) \right\| \quad (3.7b)$$

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

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

$$\mathbf{v}_0^k(t)^T \mathbf{v}_i^k(t) < \cos(68.5^\circ) \left\| \mathbf{v}_0^k(t) \right\| \cdot \left\| \mathbf{v}_i^k(t) \right\| \quad (3.8)$$

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 $\mu_i^k \in \{0, 1\}$ determines whether Rule 14 or 15 in the COLREGS are violated between the own-ship and obstacle i in scenario k , given as

$$\mu_i^k = \text{RULE14 or RULE15} \quad (3.9)$$

where

$$\text{RULE14} = \text{CLOSE \& STARBOARD \& HEAD-ON} \quad (3.10a)$$

$$\begin{aligned} \text{RULE15} = \text{CLOSE \& STARBOARD \& CROSSED} \\ \text{\& NOT OVERTAKEN} \end{aligned} \quad (3.10b)$$

Rule 13 is also included here, as it states that the overtaking vessel shall keep out of the way.

Path Deviation Cost and Grounding Cost

As the offsets will lead to a deviation from the references set by the LOS-law, the term

$$\begin{aligned} f(\chi_m, u_m) = K_{u_m}(1 - u_m) + K_\chi \chi_m^2 + K_{\Delta u_m}(u_m - u_{m,last}) \\ + K_{\Delta \chi}(\chi_m - \chi_{m,last}) \end{aligned} \quad (3.11)$$

penalizes this behavior. Here, K_{u_m} , K_χ , $K_{\Delta u_m}$ and $K_{\Delta\chi}$ are penalty parameters. In [14] it was suggested to use different cost on the terms in (3.11) depending on whether a port or starboard turn is made, in order to help assuring compliance with the COLREGS. This gives the modified path deviation cost

$$f(\chi_m, u_m) = K_{u_m}(1 - u_m) + K_\chi(\chi_m) + \Delta_{u_m}(u_m, u_{m,last}) + \Delta_\chi(\chi_m, \chi_{m,last}) \quad (3.12)$$

where

$$K_\chi(\chi_m) = \begin{cases} K_{\chi,port}\chi_m^2, & \text{if } \chi_m < 0. \\ K_{\chi,starboard}\chi_m^2, & \text{otherwise} \end{cases} \quad (3.13)$$

and

$$\Delta_{u_m}(u_m, u_{m,last}) = K_{\Delta u_m}|u_m - u_{m,last}| \quad (3.14)$$

,

$$\Delta_\chi(\chi_m, \chi_{m,last}) = \begin{cases} K_{\Delta\chi,port}\chi_m^2, & \text{if } \chi_m < 0. \\ K_{\Delta\chi,starboard}\chi_m^2, & \text{otherwise} \end{cases} \quad (3.15)$$

are the new penalty parameters. This path deviation cost is used here. The cost for port turn is here larger than for starboard, which ensures the correct behavior in the SBMPC. A grounding cost $g(\cdot)$ can also be included in the cost function, which quantifies penalty based on information about the nearby environment, from an electronic map or possibly own-ship sensor data.

The Cost Function

The hazard or cost $\mathcal{H}^k(t_0)$ associated with scenario k at current time t_0 is given as

$$\mathcal{H}^k(t_0) = \max_i \max_{t \in D(t_0)} (C_i^k(t)\mathcal{R}_i^k(t) + \kappa_i\mu_i^k(t)) + f(\chi_m, \chi_{m,last}, u_m, u_{m,last}) + g(\cdot) \quad (3.16)$$

where $D(t_0) = \{t_0, t_0 + T_s, \dots, t_0 + T\}$ contains the discrete sample times in the prediction from the current time t_0 , where T_s is the sample time and T is the prediction horizon. Lastly, κ_i is a tuning parameter for the COLREGS cost term.

The optimal scenario which yields minimal hazard is then found on 5 – 10

minute intervals as

$$k^*(t_0) = \arg \min_k \mathcal{H}^k(t_0) \quad (3.17)$$

after the hazard of all scenarios have been evaluated. The parameter values for the SBMPC used in this thesis are given in Table 3.1.

Table 3.1: SBMPC parameters used in the thesis.

Parameter	Value	Unit
T	150	s
T_s	0.05	s
d_i^{safe}	40.0	m
d_{close}	200.0	m
p	1.0	
q	4.0	
K_i^{coll}	0.5	
κ_i	3.0	
K_{um}	2.5	
$K_{\Delta_{um}}$	2.0	
$K_{\chi, port}$	1.8	
$K_{\chi, starboard}$	1.5	
$K_{\Delta_{\chi, port}}$	1.2	
$K_{\Delta_{\chi, starboard}}$	0.9	

The parameters differ slightly from those used in [14], due to different tuning, again mostly based on trial and error.

Chapter 4

Tracking Methods

In this chapter, a selection of single- and multi-target tracking methods are introduced. A tracking method utilize the information obtained by the own-ship sensor system in order to produce an estimate of nearby obstacle/target positions and velocities, often in addition to some measure of the estimate uncertainty. These estimates produced by a tracking system are vital for use in a COLAV system, where it is paramount to keep track of the obstacles, be it static or dynamic ones.

The Kalman Filter (KF) is introduced as the first single target tracking method, with the Probability Data Association Filter (PDAF) and its Integrated version (IPDAF) derived afterwards, mainly based on [2],[28] and [6], with some derivations from [43].

The multi-target tracking methods introduced are the Joint Integrated Probability Data Association Filter (JIPDAF) and Multiple Hypothesis Tracker (MHT), with derivations mainly based on [2], [27] and [6]. A slightly different notation from the sources mentioned above are used in the derivation of the tracking methods, with the index j used for the measurement number, which is preserved for i in the literature. However, the index i is preserved for the obstacle number in this thesis.

Radar systems are often used by vessels to detect nearby obstacles, which sends out beams of radio waves in 360 degrees covering its surveillance region. When nearby obstacles reflect these waves, the time of return and direction of the reflection can be used to determine the obstacle position. However, as the radio waves can be reflected of irrelevant entities such as water and clouds, so-called

clutter or false measurements will be a challenge here. In addition, because the reflected signals need to be above a certain threshold in order to pass on as valid measurements, there will be false alarms originating from for example noisy measurements. Thus, at each sampling instant or scan k , multiple measurements $\mathbf{Z}(k) = \{\mathbf{z}_k^j\}_{j=1}^{m'_k}$ can be obtained, which can originate from multiple targets or from clutter. Here, m'_k is the *total* number of measurements obtained at sampling instant k . This give rise to the problem of associating measurements to each target, which require more complex tracking methods than the Kalman Filter in order to have good performance, one of them being the PDAF. The extra uncertainty from data association is also a challenge that need to be dealt with in a COLAV system for an ASV.

4.1 The Kalman Filter

The Kalman Filter was introduced in 1960 by Rudolf E. Kálmán [20], and is an optimal recursive estimator in the sense of minimal variance. The estimator can be derived by solving a least squares optimization problem, or by maximizing the conditional probability density $p(\mathbf{x}_k | \mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k)$, using the measurements \mathbf{z} from an initial time t_0 up to the current time t_k [34]. The Kalman filter is said to be optimal in the Bayesian sense if the following criteria are satisfied

- The system is linear
- The system is observable
- The initial state \mathbf{x}_0 is normally distributed with a known mean and covariance
- The process noise \mathbf{v} and measurement noise \mathbf{w} are independent zero mean white noise processes with covariance matrices \mathbf{R} and \mathbf{Q} , respectively.

The KF uses one measurement at each sampling instant for its estimate update, and thus $m'_k = 1$. In general, the covariance matrices \mathbf{R} and \mathbf{Q} can vary with time. As a result of the given optimality criteria, the conditional probability density $p(\mathbf{x}_k | \mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k)$ is a multivariate Gaussian, where the expectation and covariance are updated through the Kalman Filter. A multivariate Gaussian can be described by

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{|2\pi|^{\frac{n_x}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})} \quad (4.1)$$

where $\boldsymbol{\mu}$ and Σ are the mean and covariance of the distribution, respectively, and n_x is the dimension of \mathbf{x} . Further, the noise covariance matrices should also have proper values for the KF to be optimal.

When $k = 0$, the KF is initialized with

$$\hat{\mathbf{x}}_{k|k-1} = \bar{\mathbf{x}}_0 \quad (4.2a)$$

$$\mathbf{P}_{k|k-1} = \bar{\mathbf{P}}_0 \quad (4.2b)$$

where $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are the a priori state estimate and estimation error covariance, respectively. $\bar{\mathbf{x}}_0$ and $\bar{\mathbf{P}}_0$ are the a priori initial state and error covariance, respectively. These variables are obtained through process knowledge, data, or just by guessing. A highly uncertain initial guess for $\hat{\mathbf{x}}_{k|k-1}$ corresponds to a large $\mathbf{P}_{k|k-1}$. Upon arrival of the first measurement, the estimate and error covariance are updated respectively through the Kalman gain and Ricatti equation

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R})^{-1} \quad (4.3a)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \quad (4.3b)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (4.3c)$$

\mathbf{K}_k is here the Kalman gain at time instant t_k , which blends the measurement and process noise covariances to update the a posteriori estimate $\hat{\mathbf{x}}_{k|k}$ using the new measurement \mathbf{z}_k . $\mathbf{P}_{k|k}$ is the a posteriori error covariance, updated through the recursive Ricatti equation. Moreover, $\hat{\mathbf{z}}_{k|k-1} = \mathbf{H} \hat{\mathbf{x}}_{k|k-1}$ is the predicted measurement at time t_k . Before the next time step t_{k+1} , the new a priori estimate and error covariance are projected ahead using the estimation model

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F} \hat{\mathbf{x}}_{k|k} \quad (4.4)$$

$$\hat{\mathbf{z}}_{k|k} = \mathbf{H} \hat{\mathbf{x}}_{k|k}$$

which yields the prediction part

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F} \hat{\mathbf{x}}_{k|k} \quad (4.5a)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{Q} \quad (4.5b)$$

4.2 The PDAF

When the number of measurements $m'(k)$ obtained at sampling instant k is greater than 1, the problem of determining which measurements originated from which target arises, i.e. the so-called data association problem. In this situation, assuming the target track has been initialized, the KF needs some extra logic in order to determine which measurement to use in its update, for instance by choosing the measurement closest to its predicted measurement $\hat{\mathbf{z}}_{k|k-1}$, in a nearest-neighbour fashion. However, whenever the closest measurement is clutter, the performance of the KF can quickly degrade. The PDAF solves this problem by calculating the probability that measurement j is associated with the target, using the following assumptions [2]

- Only one target of interest exists.
- The track of the target has been initialized.
- The past information about the target can be summarized by the statistic

$$p(\mathbf{x}_k | \mathbf{Z}^{k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (4.6)$$

where $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$ is the multivariate Gaussian described in (4.1), with mean and covariance $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, respectively. Moreover, $\mathbf{Z}^{k-1} = \{\mathbf{Z}(l)\}_{l=1}^{k-1}$ is the cumulative set of measurements from sampling instant 1 up to $k-1$.

- At each sampling time, a validation region (gate)

$$\mathbf{v}_k^{jT} \mathbf{S}_k^{-1} \mathbf{v}_k^j \leq \gamma_G \quad (4.7)$$

is set up around the current measurement prediction $\hat{\mathbf{z}}_{k|k-1}$, where γ_G is the gate threshold and $\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}$ is the predicted innovation covariance of the single obstacle/target. Moreover, $\mathbf{v}_k^j = \mathbf{z}_k^j - \hat{\mathbf{z}}_{k|k-1}$ is measurement innovation j . All measurements \mathbf{z}_k^j inside the region are validated, and considered as possibly target-originated. A total of $0 \leq m_k \leq m'_k$ measurements are validated at each sampling instant k .

- Among the validated measurements, at most one of them can be target originated. The rest are assumed to be caused by clutter or false alarms, and are modelled as independently identically distributed (i.i.d), uniformly distributed inside the validation space.
- The target is detected with known probability P_D , independent of time. Thus, the target generates a measurement \mathbf{z}_k with probability P_D .

The value of the gate threshold γ_G is determined through the inverse cumulative χ^2 distribution with n_z degrees of freedom, where n_z is the measurement dimension. It is often chosen such that measurements are gated with a high probability $P_G \geq 0.9$.

The PDAF updates the state estimate and covariance in the same fashion as the KF, with the difference in that the innovation used in the update part is blended with the association probability β_k^j of each validated measurement being target originated, or that none was target originated for $j = 0$. To find the association probabilities, the association hypothesis

$$\theta_k^j = \begin{cases} \mathbf{z}_k^j \text{ is the target originated measurement,} & j = 1, 2, \dots, m_k \\ \text{no measurement originated from the target,} & j = 0 \end{cases} \quad (4.8)$$

is defined. Then, the posterior mean $\hat{\mathbf{x}}_{k|k}$ can be written as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= E[\mathbf{x}_k | \mathbf{Z}^k] \\ &= \sum_{j=0}^{m_k} E[\mathbf{x}_k | \theta_k^j, \mathbf{Z}^k] P(\theta_k^j | \mathbf{Z}^k) \\ &= \sum_{j=0}^{m_k} \hat{\mathbf{x}}_{k|k}^j \beta_k^j \end{aligned} \quad (4.9)$$

using the total probability theorem. The association probabilities $\beta_k^j = P(\theta_k^j | \mathbf{Z}^k)$ are given below in (4.15). Then, the update part of the filter is given by [2]

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T \mathbf{S}_k^{-1} \quad (4.10a)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k \quad (4.10b)$$

$$\mathbf{v}_k = \sum_{j=1}^{m_k} \beta_k^j \mathbf{v}_k^j \quad (4.10c)$$

$$\mathbf{P}_{k|k} = \beta_k^0 \mathbf{P}_{k|k-1} + (1 - \beta_k^0) \mathbf{P}_{k|k}^c + \tilde{\mathbf{P}}_k \quad (4.10d)$$

where the covariance $\mathbf{P}_{k|k}^c$ in (4.10d) is associated with the state update using the correct measurement:

$$\mathbf{P}_{k|k}^c = \mathbf{P}_{k|k-1} + \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \quad (4.11)$$

and the last term in (4.10d) is the spread of the innovations, given as

$$\tilde{\mathbf{P}}_k = \mathbf{K}_k \sum_{j=1}^{m_k} (\beta_k^j \mathbf{v}_k^j \mathbf{v}_k^{jT} - \mathbf{v}_k \mathbf{v}_k^T) \mathbf{K}_k^T \quad (4.12)$$

One can see from (4.10d) that if the association probability of no measurement being target originated is $\beta_k^0 = 1$, the state covariance is not updated, and $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1}$. Otherwise, the effect of having multiple validated measurements and data origin uncertainty is seen through an increase of $\tilde{\mathbf{P}}_k$ in the posterior state covariance $\mathbf{P}_{k|k}$.

To find the association probabilities, a model of the clutter measurements is needed. The clutter measurement distribution can be modelled as a Poisson model with clutter density λ :

$$\mu_F(m_k) = e^{-V_k \lambda} \frac{(V_k \lambda)^{m_k}}{m_k!} \quad (4.13)$$

where V_k is the volume of the validation region, given as

$$V_k = c_{n_x} |y_G \mathbf{S}_k|^{1/2}, \quad c_1 = 2, c_2 = \pi, c_3 = \frac{4\pi}{3}, \dots \quad (4.14)$$

where n_x is the dimension of the target (obstacle) state vector. The a priori clutter density λ is replaced with $\frac{m_k}{V_k}$ when the non-parametric clutter model is used. For the parametric case, the density λ is either estimated or determined a priori. The resulting association probabilities β_k^j when using the diffuse prior model are shown here, and can be found as [43]

$$\beta_k^j = \begin{cases} \frac{1}{c_1} e^{-\frac{1}{2} \mathbf{v}_k^{jT} \mathbf{S}_k^{-1} \mathbf{v}_k^j}, & j = 1, 2, \dots, m_k. \\ \frac{1}{c_1} \frac{2(1-P_D P_G)}{y_G} m_k, & j = 0 \end{cases} \quad (4.15)$$

where c_1 is a normalization constant such that $\sum_{j=0}^{m_k} \beta_k^j = 1$.

This filter can be used reliably also in multi-target scenarios, if the targets are far enough from each other such that their PDAF validation gates do not overlap. Due to its recursive form, and it being an extension to the KF, the filter is fast and computationally efficient, although in general sub-optimal.

4.3 The IPDAF

The Integrated Probability Data Association filter proposed by Darko Musicki in 1994 [28] extends the PDAF to include target existence probability, which relaxes the PDAF assumption that the target must exist. For the IPDAF, the target exist at sampling instant k with posterior probability ϵ_k , and does not with probability $1 - \epsilon_k$ [6]. To include the target existence into the filter, the state vector is extended with a discrete variable indicating whether the target exists or not. This variable can then be used in the tracking system as a binary decision to choose whether or not to terminate the track. The existence probability is propagated using Markov chains, where the type of chain to be used can vary based on if the track is in its initiation phase, or in the maintenance phase [28]. The first Markov chain is stated here, which only includes whether or not the target exists at time t_k or not. The other Markov chain also includes the possibility that the target may or may not be observable at the current time instant.

The IPDA-algorithm can be decomposed in 5 steps [6]: *existence probability prediction, state prediction, calculate association weights, state update and existence probability update*. Existence probability prediction using the first Markov chain is given by

$$\begin{bmatrix} \epsilon_{k+1|k} \\ 1 - \epsilon_{k+1|k} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \epsilon_k \\ 1 - \epsilon_k \end{bmatrix} \quad (4.16)$$

where $\epsilon_{k+1|k}$ is the a priori existence probability for the next time instant, and ϵ_k is the a posteriori existence probability at time t_k . The coefficients in the prediction matrix satisfy $p_{11} + p_{12} = p_{21} + p_{22} = 1$.

Further, using the following quantity (see [28] or [6] for the derivation)

$$\delta_k^\epsilon = P_D P_G + P_D P_G \frac{1}{\lambda} \sum_{j=1}^{m_k} p(z_k^j | Z^{k-1}) \quad (4.17)$$

where $p(z_k^j | Z^{k-1}) = \mathcal{N}(z_k^j; \hat{z}_{k|k-1}, S_k)$ is the a priori likelihood function for measurement j and λ is the clutter density, the existence probability update becomes

$$\epsilon_k = \frac{1 - \delta_k^\epsilon}{1 - \delta_k^\epsilon \epsilon_{k|k-1}} \epsilon_{k|k-1} \quad (4.18)$$

With the target existence included in the filter framework, the association probabilities β_k^j then becomes [28]

$$\beta_k^j = \begin{cases} \frac{1 - P_D P_G}{1 - \delta_k^\epsilon}, & j = 0 \\ \frac{P_D P_G \frac{1}{\lambda} p(z_k^j | Z^{k-1})}{1 - \delta_k^\epsilon}, & j = 1, 2, \dots, m_k \end{cases} \quad (4.19)$$

4.4 The JIPDAF

In situations with multiple targets relatively close to each other such that their validation gates overlap [6], the performance of the PDAF/IPDAF will degrade, as all except at most one validated measurement are modelled as random inference (clutter). In this case, with other targets being present in close vicinity of each other, persistent interference will occur, and not only random inference. The multi-target extension of the PDAF; the Joint Probability Data Association Filter, solves this problem by considering hypotheses or joint association events which corresponds to all possible feasible combinations of which validated measurements that originated from which target. Moreover, the Integrated version (JIPDAF) incorporates track existence probabilities into the filtering framework in a similar manner as for the IPDAF.

Figure 4.1 gives an illustration of the problem when the tracks from two targets $t = 1$ and $t = 2$ coalesce, causing the validation gates around the predicted measurements to overlap. Using a PDAF in this situation for the two targets shown, can cause problems, due to the filter assumption that all measurements not target originated are clutter. The probability that the measurement present in the validation region of both targets is classified as either clutter or as originated from target 1 or 2 must now be calculated.

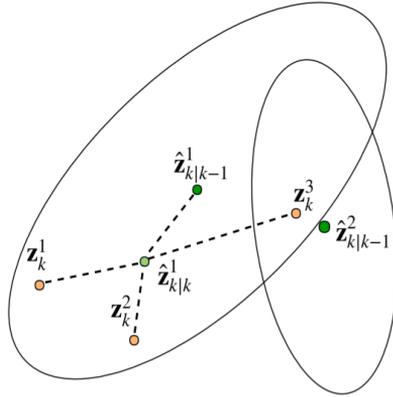


Figure 4.1: JIPDAF state estimation update where the new position estimate of target 1 $\hat{z}_{k|k}^1$ is weighted based on the distance from the validated measurements z_k^j to the predicted position $\hat{z}_{k|k-1}^1$ and the possibility of them originating from another target. Notice that the closest measurement z_k^3 is weighted significantly less than the other measurements. This is because this measurement is much closer to another target estimate $\hat{z}_{k|k-1}^2$ [17].

The assumptions underlying the JIPDAF are given below

- Measurements can occur inside of the validation region of multiple neighbouring targets over many sampling instants, causing the persistent interference.
- Past information about each target state is (as in PDAF) summarized by an approximate statistic, which gives the mean and covariance of each target/track state.
- The past information is normally distributed with known mean and covariance
- Each target has a model as in (2.8), where the same model need not be used for all targets.
- Each target t is detected with known probability P_D^t , independent of time. Thus, the target generates a measurement z_k with probability P_D^t .
- Each track validates measurements inside its validation region with probability P_G^t , in the same manner as for the PDAF. Validated measurements

inside overlapping validation regions from neighbouring tracks are clustered (see third paragraph below).

As one can see, many of the points are similar or equal to that in the PDAF/IPDAF. The main improvement from the IPDAF to the JIPDAF is the evaluation of the measurement-to-target joint association probabilities based on joint events, which are computed jointly for all targets. As for the IPDAF, the probabilities are computed only based on the latest set of measurements, and the state estimation is done separately for each target t . A coupled estimation approach is described in [2], for the interested reader. The a posteriori state estimate and covariance for target t are here denoted by $\hat{\mathbf{x}}_{k|k}^t$ and $\mathbf{P}_{k|k}^t$, respectively, and similarly for the a priori entities.

In a similar manner, the existence probabilities ϵ_k^t for each target $t = 1, 2, \dots, n_t$ are maintained in the JIPDAF, with the existence prediction again following (4.16). The number of targets n_t varies depending on whether on not a track is terminated or initialized, where the termination can be made on a basis of the target existence probabilities. Initialization of tracks can be done using for instance the 2/2&m/n logic [2].

To prevent computational overflow in evaluating association probabilities, overlapping validation regions from multiple tracks (as in Figure 4.1) or tracks with separated validation regions can form clusters in the JIPDAF. For each cluster, there is a set of m_k validated measurements, and a joint event defines a possible assignment of these measurements to the cluster targets, with the assumption of maximum one target-originated measurement. Thus, the association hypothesis is here redefined as a vector $\boldsymbol{\theta}_k = [\theta_k(1) \ \theta_k(2) \ \dots \ \theta_k(n_t)]^T$ [6] where

$$\theta_k(t) = \begin{cases} j, & \text{if measurement } j \text{ is claimed by target } t \\ 0, & \text{if no measurement is claimed by target } t \end{cases} \quad (4.20)$$

for $j = 1, 2, \dots, m_k$. The set of all association hypotheses for each cluster are mutually exclusive and exhaustive. The posterior joint association probabilities are found via Bayes' formula as

$$\begin{aligned} P(\boldsymbol{\theta}_k | \mathbf{Z}^k) &= P(\boldsymbol{\theta}_k | \mathbf{Z}_k, m_k, \mathbf{Z}^{k-1}) \\ &= \frac{1}{c_2} p(\mathbf{Z}_k | \boldsymbol{\theta}_k, m_k, \mathbf{Z}^{k-1}) P(\boldsymbol{\theta}_k | m_k) \end{aligned} \quad (4.21)$$

where c_2 is a normalization constant, and where one has used that the association hypothesis θ_k is only conditionally dependent on the current number of measurements m_k . Denoting φ as the number of clutter measurements hypothesized by the event θ_k , the first term in the product of (4.21), which is the likelihood of the measurement function, is found using the total probability theorem, conditioned on every target state $\mathbf{x}_k^{t, \theta_k(t)}$ under the hypothesis $\theta_k(t)$, as [6]

$$p(\mathbf{Z}_k | \theta_k, m_k, \mathbf{Z}^{k-1}) = \frac{1}{V_k^\varphi} \prod_{t: \theta_k(t) > 0} \mathcal{N}(\mathbf{z}_k^{\theta_k(t)}; \mathbf{H}\hat{\mathbf{x}}_k^{t, \theta_k(t)}, \mathbf{H}\mathbf{P}_k^{t, \theta_k(t)}\mathbf{H}^T + \mathbf{R}) \quad (4.22)$$

where

$$\hat{\mathbf{x}}_k^{t, \theta_k(t)} = \begin{cases} \hat{\mathbf{x}}_{k|k-1}^t & \text{if } \theta_k(t) = 0 \\ \hat{\mathbf{x}}_{k|k-1}^t + \mathbf{K}_k^t(\mathbf{z}_k^{\theta_k(t)} - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}^t) & \text{if } \theta_k(t) > 0 \end{cases} \quad (4.23)$$

and

$$\mathbf{P}_k^{t, \theta_k(t)} = \begin{cases} \mathbf{P}_{k|k-1}^t & \text{if } \theta_k(t) = 0 \\ (\mathbf{I} - \mathbf{K}_k^t\mathbf{H})\mathbf{P}_{k|k-1}^t(\mathbf{I} - \mathbf{K}_k^t\mathbf{H})^T + \mathbf{K}_k^t\mathbf{R}\mathbf{K}_k^{t,T} & \text{if } \theta_k(t) > 0 \end{cases} \quad (4.24)$$

are the state estimate and error covariance of target t conditioned under the hypothesis $\theta_k(t)$. The likelihood (4.22) says that the clutter measurements are uniformly distributed in the validation region volume V_k , whereas the target-originated measurements come from Gaussian distributions.

Further, the following track-oriented configuration vector is now defined [6], as a step in finding the last term in the product of (4.21): $\boldsymbol{\tau}_c = [\tau_c(1) \ \dots \ \tau_c(n_t)]^T$, which is not to be mistaken as the generalized force vector from Section ???. This vector of binary elements encode information about which targets that are detected under the association hypothesis θ_k , although which measurements these detections correspond to are not given. The element $\tau_c(t) = 1$ if target t is detected, and 0 otherwise. Thus, the following relation is established between $\boldsymbol{\tau}$, φ and m_k :

$$m_k = \varphi + \sum_{t=1}^{n_t} \tau_c(t) \quad (4.25)$$

The unconditional probability of the event τ_c , when including the target existence framework of the JIPDAF, is given as

$$p(\tau_c) = \prod_{t:\theta_k(t)=0} (1 - P_D^t P_G^t \epsilon_k^t |_{k-1}) \prod_{t:\theta_k(t)>0} P_D^t P_G^t \epsilon_k^t |_{k-1} \quad (4.26)$$

Further, the following probabilities conditioned on τ_c

$$P(m_k | \tau_c) = e^{-V_k \lambda} \frac{(V_k \lambda)^\varphi}{\varphi!} \quad (4.27)$$

$$P(\theta_k | \tau_c, m_k) = \frac{\varphi!}{m_k!} \quad (4.28)$$

are used. The first probability comes from the fact that, among the remaining φ measurements not assigned in τ_c , the rest must be from clutter, which here are distributed through a Poisson model. The latter probability (4.28) is due to the existence of $m_k! / (m_k - \sum_t \tau_c(t))! = m_k! / \varphi!$ equally likely permutations of the detected measurements, given τ_c . Thus, the latter term in the product of (4.21) is found using Bayes' formula as

$$\begin{aligned} p(\theta_k | m_k) &= p(\theta_k, \tau_c | m_k) \\ &= \frac{1}{c_3} p(\theta_k | \tau_c, m_k) p(m_k | \tau_c) p(\tau_c) \\ &= \frac{1}{c_3} e^{-V_k \lambda} \frac{(V_k \lambda)^\varphi}{m_k!} \prod_{t:\theta_k(t)=0} (1 - P_D^t P_G^t \epsilon_k^t |_{k-1}) \prod_{t:\theta_k(t)>0} P_D^t P_G^t \epsilon_k^t |_{k-1} \\ &\propto \prod_{t:\theta_k(t)=0} (1 - P_D^t P_G^t \epsilon_k^t |_{k-1}) \prod_{t:\theta_k(t)>0} P_D^t P_G^t \epsilon_k^t |_{k-1} \end{aligned} \quad (4.29)$$

again where c_3 is a normalization constant. This gives the posterior joint association probabilities [2][27][6]

$$\begin{aligned}
p(\theta_k | \mathbf{Z}^k) &= \frac{1}{c_4} \mu_F(\varphi) \frac{\varphi!}{m_k!} V_k^{-\varphi} \prod_{t:\theta_k=0} (1 - P_D^t P_G^t \epsilon_k^t |_{k-1}) \\
&\quad \prod_{t:\theta_k>0} P_D^t P_G^t \epsilon_k^t |_{k-1} \mathcal{N}(\mathbf{z}_k^{\theta_k(t)}; \mathbf{H} \hat{\mathbf{x}}_k^{t, \theta_k(t)}, \mathbf{H} \mathbf{P}_k^{t, \theta_k(t)} \mathbf{H}^T + \mathbf{R}) \\
&\propto \prod_{t:\theta_k=0} (1 - P_D^t P_G^t \epsilon_k^t |_{k-1}) \\
&\quad \prod_{t:\theta_k>0} P_D^t P_G^t \epsilon_k^t |_{k-1} \mathcal{N}(\mathbf{z}_k^{\theta_k(t)}; \mathbf{H} \hat{\mathbf{x}}_k^{t, \theta_k(t)}, \mathbf{H} \mathbf{P}_k^{t, \theta_k(t)} \mathbf{H}^T + \mathbf{R})
\end{aligned} \tag{4.30}$$

with normalization constant c_4 . Finally, the probabilities for all joint events where a measurement \mathbf{z}_k^j is associated with each target t , $t = 1, 2, \dots, n_t$, are summed, giving the marginal association probabilities

$$\beta_k^{t,j} = \sum_{\theta_k: \theta_k(t)=j} p(\theta_k | \mathbf{Z}^k) \tag{4.31}$$

These are applied in the decoupled target state estimation (4.10) (replaces β_k^j) for each target t , in exactly the same manner as for the single-target PDAF.

4.5 The MHT

The Multiple Hypothesis Tracker proposed by Donald B. Reid in 1979[35] develops association hypotheses by considering all possible sequences of measurements (i.e all hypothesized tracks), and calculates all the corresponding association probabilities. This causes an exponential increase in complexity with time, and several methods for mitigating this problem are used in practice. Methods used for complexity reduction are pruning of low probability hypotheses, merging of similar hypotheses and clustering of tracks. Each sequence of measurements or hypothesized track is updated and propagated in time using a Kalman Filter as described in Section 4.1.

A thing that separates MHT from the approaches described in the previous sections, is that it is measurement-oriented. The MHT evaluates the probability that a certain measurement sequence originated from either an established target or a new target. On the other hand, the PDAF/JPDAF are target-oriented approaches, which evaluate the probability that measurements origin from a specific target. Because of this feature in the MHT, track initiation can easily

be included in its algorithm. The derivation of this tracker can for instance be found in [2] or [6].

4.6 Remark

Several methods were here introduced for tracking obstacles, with different levels of complexity. As mentioned in the Problem Description, this thesis only deals with kinematic uncertainty and sensor precision uncertainty, as a first step of incorporating uncertainty into a COLAV method. Thus, the simplest of the methods introduced here will be used as a tracking method in the rest of this thesis, namely the Kalman Filter.

The reason why the other methods were described, is to give an introduction to the multi-target tracking problem and the data association problem, which come with extra challenges to be solved in future work.

Chapter 5

Collision Probability Evaluation Methods

In Chapter 3, it was shown how the SBMPC defines a safety zone around the own-ship in order to determine its cost penalization. Moreover, in Chapter 4, it was shown how the uncertainty related to the nearby obstacle position and its velocity can be represented using the mean and covariance produced by the tracking system, which essentially describes an ellipse in 4 dimensions (position and velocity). This uncertainty will be used in determining collision probabilities. In this chapter, two methods for quantifying the probability of collision with other obstacles are presented, using the uncertainty produced by the tracking system.

Previous work that attempts to determine the collision probability is mainly based on integrating only the PDF describing the obstacle (and own-ship) positional uncertainty, over the so-called conflict zone [32][30][31]. The conflict zone [32] is defined as the overlapping region between the safety zone and probability ellipse. Thus, the probability of collision can be obtained by integrating the probability density of the obstacle position over the conflict zone. If the kinematic uncertainty of the own-ship also were taken into account, which is not done here, the combined positional uncertainty of both vessels would have to be used instead. To do this, the own-ship uncertainty ellipse must then be transformed in order to be combined with the obstacle uncertainty ellipse.

Now one may ask if it is sufficient to only consider positional uncertainty in evaluating the collision probability? The answer will depend on how the

probability is defined and what the application is. If the goal is to calculate the instantaneous collision probability at a particular time instant, only the positional uncertainty is required. However, if the goal is to predict the probability of collision at some point in the future, the uncertainty in velocity also need to be accounted for. This introduces challenges related to efficient evaluation methods, as 4-dimensional integrals involving horizontal position and velocity need to be solved, or 6-dimensional if 3D-motion is considered. This thesis will primarily look at the last case for horizontal motion, considering also velocity uncertainty, as this can be important knowledge to utilize in a COLAV method for more efficient avoidance maneuvers.

The first method presented here which attempts to quantify the collision probability and which is used in the rest of this thesis, is based on importance sampling, namely Monte Carlo integration. The other method presented in this thesis uses the concept of probability flow to determine this probability [32], which is at the front in the literature with respect to efficiency and quality of the collision probability calculated, although it does not consider uncertainty in velocity in its calculations. Thus, this method can not be used to predict collision probabilities, using the full PDF of the obstacle.

Note that the notation used in these two chapters (5 and 6) have the obstacle index i as a superscript instead of a subscript, as is used in Chapter 3 about COLREGS and SBMPC.

5.1 Thesis Collision Probability Definition

The collision probability between an obstacle i and the own-ship, is in this thesis defined using the events

$$\begin{aligned} \mathcal{A}_k^i = & \text{A collision occurs between obstacle } i \text{ and the own-ship} \\ & \text{at some time } t_c \geq t_k \text{ inside the own-ship safety zone.} \end{aligned} \quad (5.1)$$

and

$$\begin{aligned} \mathcal{B}_k^i = & \text{A collision between obstacle } i \text{ and the own-ship does not occur} \\ & \text{at any time } t_c \geq t_k \text{ inside the safety zone.} \end{aligned} \quad (5.2)$$

which are mutually exhaustive. The safety zone is in general the circular region with radius $d_{safe} = d_i^{safe}$ around the own-ship (same for all obstacles in this

thesis, see Section 3.3), as defined in the SBMPC (Section 3.3). The probability of collision with obstacle i at time t_k then becomes

$$\mathbb{P}_{c,k}^i = P\{\mathcal{A}_k^i\} = 1 - P\{\mathcal{B}_k^i\} \quad (5.3)$$

Thus, the definition of the collision probability in this thesis allows for predicting this value, instead of just calculating it for the current time instant. Note that the probability flow method which will be described in Section 5.3 defines the probability of collision to be at a particular time instant, i.e. the probability that $t_c = t_k$ in the above defined events.

The goal is thus in general to find the collision probability by integrating the obstacle PDF $p^i(\mathbf{x}, t_k)$ over some region in \mathbb{R}^4 . If only uncertainty in position is considered, as is the case in much of the previous work [32][30][31], the obstacle position PDF is integrated over the conflict zone $\mathcal{D}_{xy} \subset \mathbb{R}^2$:

$$\mathbb{P}_{c,k}^i = \iint_{\mathcal{D}_{xy}} p_{xy}^i(x, y; t_k) dx dy \quad (5.4)$$

where $p_{xy}^i(x, y; t_k)$ is the obstacle position PDF at time t_k . The conflict zone represent the overlapping area between the in general combined positional uncertainty of the obstacle and own-ship, and the safety zone around the own-ship (see Figure 5.1). The own-ship position (and velocity) are in this thesis assumed known perfectly, thus only the obstacle uncertainty is considered in the integral (5.4). If both position and velocity uncertainty is considered, the integral

$$\mathbb{P}_{c,k}^i = \int_{\mathcal{D}_{full}} p^i(\mathbf{x}; t_k) d\mathbf{x} \quad (5.5)$$

must be evaluated, where $\mathcal{D}_{full} \subset \mathbb{R}^4$ is a region in 4 dimensions which include all obstacle trajectories that cross the own-ship safety zone at some time.

5.2 Monte Carlo Integration

Given a PDF $\pi(\mathbf{x})$ where \mathbf{x} is for instance an obstacle position, one would often like to extract properties such as the mean and covariance of \mathbf{x} , through an integral

$$I_1 = \int_{\mathbf{x}} g(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \quad (5.6)$$

where $g(\mathbf{x}) = \mathbf{x}$ if for instance the mean is to be found. A closed form expression of the integral in (5.6) can in general not be expected to exist, for an arbitrary $\pi(\mathbf{x})$. In these cases, an estimator can be used instead. For the calculation of the mean, one can samples values \mathbf{x}_l from the PDF $\pi(\mathbf{x})$, and use the average

$$\hat{\mathbf{x}} = \frac{1}{N} \sum_{l=1}^N \mathbf{x}_l \quad (5.7)$$

as an estimator for the mean $\bar{\mathbf{x}}$, where N is the amount of samples drawn. This can be extended to the general case where one can generate sample values $g(\mathbf{x}^i)$ using $\pi(\mathbf{x})$, and approximate the integral (5.6) using the average of these sample values.

Now, lets say one wants to determine an integral

$$I_2 = \int_{\mathbf{x}} f(\mathbf{x})d\mathbf{x} \quad (5.8)$$

If one can decompose the integrand as $f(\mathbf{x}) = g(\mathbf{x})\pi(\mathbf{x})$, then the above method can be applied also here. In the Monte Carlo Integration method[6], an integral of the form

$$\begin{aligned} I &= \int_{\mathbf{x}} f(\mathbf{x})d\mathbf{x} = \int_{\mathbf{x}} g(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \\ &= \int_{\mathbf{x}} g(\mathbf{x}) \frac{\pi(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})d\mathbf{x} \end{aligned} \quad (5.9)$$

which one wish to evaluate, is approximated as

$$I_N = \frac{1}{N} \sum_{l=1}^N g(\mathbf{x}_l)\tilde{w}(\mathbf{x}_l) \quad (5.10)$$

where $\tilde{w}(\mathbf{x}_l) = \frac{\pi(\mathbf{x}_l)}{q(\mathbf{x}_l)}$ is known as an importance weight. As it may be difficult to sample from $\pi(\mathbf{x})$, the approximation $q(\mathbf{x})$ can be used instead. Otherwise, $q(\mathbf{x}) = \pi(\mathbf{x})$.

Following the above described Monte Carlo integration method to find an approximation to the integral (5.5), the integrand (also a function of time) $f(\mathbf{x}, t)$ is here equal to $p^i(\mathbf{x}, t)$, where \mathbf{x} is here the state vector of the obstacle, consisting

of both position and velocity, thus,

$$g(\mathbf{x}) = \frac{p^i(\mathbf{x}, t_k)}{\pi(\mathbf{x}, t_k)} \quad (5.11)$$

where $\pi(\mathbf{x}, t_k)$ is the PDF to sample from. For the double integral (5.4), the density $p^i(\mathbf{x}, t_k)$ in (5.11) is merely replaced by $p_{xy}^i(x, y, t_k)$.

5.3 Probability Flow

The conflict zone probability or collision probability has in previous studies been calculated through the use of for instance Monte Carlo Integration, analytical approximations and neural networks. The more recent method developed by Jeonghong Park and Jinwhan Kim[32] uses the flow of probability across the conflict zone boundary to find the collision probability, where the flow is decomposed into drift- and diffusion components. The method employs a kinematic model similar to the one described in Section 2.1.2 for trajectory prediction. The difference is that the state vector for the own-ship and target only consist the respective positions. The target- and obstacle model follows the Wiener process (using the notation in [32])

$$\begin{aligned} \dot{\mathbf{x}}_O &= [\dot{x}_O \quad \dot{y}_O]^T + \mathbf{w}_O \\ &= [v_O \cos\psi \quad v_O \sin\psi]^T + \mathbf{w}_O \end{aligned} \quad (5.12a)$$

$$\begin{aligned} \dot{\mathbf{x}}_T &= [\dot{x}_T \quad \dot{y}_T]^T + \mathbf{w}_T \\ &= [v_T \cos\psi \quad v_T \sin\psi]^T + \mathbf{w}_T \end{aligned} \quad (5.12b)$$

where \mathbf{x}_O and \mathbf{x}_T are the own-ship and target (obstacle) x-y position, respectively. Similarly, v_O and v_T are the speeds of the own-ship and obstacle, and ψ_O and ψ_T are the respective heading angles. The process noise \mathbf{w}_O and \mathbf{w}_T are assumed zero-mean and Gaussian distributed with covariance matrices \mathbf{Q}_O and \mathbf{Q}_T , respectively. As the kinematic uncertainty of the own-ship is not included in the simulation environment of this thesis, the process noise of the own-ship is neglected. Thus, only the obstacle uncertainty is of interest here, when determining the collision probability.

When the two objects approach each other, their respective position uncertainty ellipses (probability densities) will eventually overlap. The Instantaneous Collision Probability (ICP) at time t between the own-ship and obstacle is then found by integrating the joint probability density of the objects over the overlapping region

$$\mathbb{P}_c^i = \iint_{\mathcal{D}} p_O(\mathbf{x}; t) p_T(\mathbf{x}; t) dx dy \quad (5.13)$$

where $\mathcal{D} \subset \mathcal{R}^2$ is the overlapping region, and $p_O(\mathbf{x}; t)$ and $p_T(\mathbf{x}; t)$ are the ownship and obstacle probability densities, respectively.

The method further defines a safety zone around the obstacle and own-ship in a similar manner to the SB-MPC, which is the minimum distance to be kept in order to avoid collision. Further, the safety zones for both ships are combined and centered around the own-ship. In a similar way, the combined uncertainty of the own-ship and obstacle are calculated using a transformation of the own-ship probability ellipse to the obstacle, centered to the obstacle. This is shown in Figure 5.1.

As the integral in (5.13) in general has no analytical solution, the proposed method attempts to calculate it using the flow of probability across the boundary of the conflict zone. Then, the ICP at time t is calculated as the integral of the probability flow from a time t_0 to t :

$$\mathbb{P}_c^i = \mathbb{P}_c^i(t_0) + \int_{t_0}^t \dot{\mathbb{P}}_c^i(\tau) d\tau \quad (5.14)$$

where $\dot{\mathbb{P}}_c^i(t)$ is the rate of change of collision probability through the combined safety zone boundary. Further, as the rate of change of probability must be equal to the net flow of probability into the safety boundary, and as no probability can disappear or be created inside the safety zone, the continuity equation

$$\frac{\partial}{\partial t} f(\mathbf{x}, t) = -\nabla \mathcal{J} \quad (5.15)$$

must be satisfied. Here, $f(\mathbf{x}, t) = p(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ is the PDF of the combined uncertainty ellipse, Gaussian and described by (4.1) with mean $\boldsymbol{\mu}$ and covariance $\Sigma = \mathbf{Q}_c \mathbf{Q}_c^T$, where \mathbf{Q}_c is the 2×2 combined positional covariance of the own-ship and target. \mathcal{J} is the flux of probability over the safety zone boundary. This flux is here decomposed into a drift and diffusion component

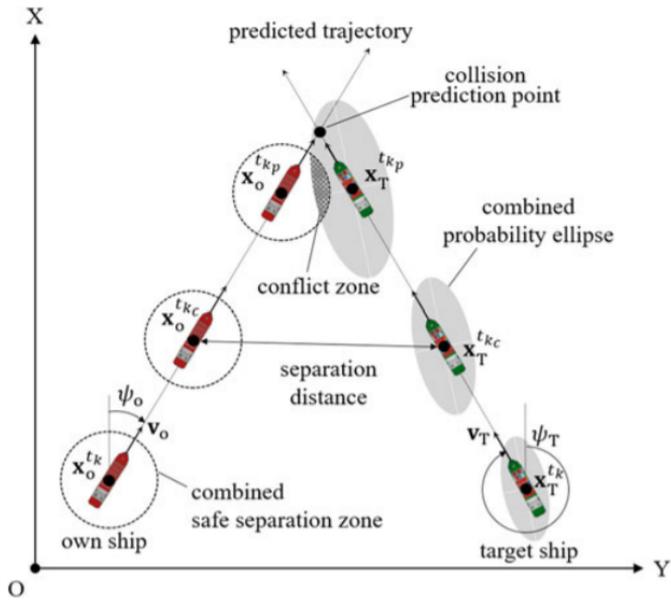


Figure 5.1: Illustration of the combined probability ellipse and the combined safe separation zone: The combined probability ellipse is obtained through coordinate transformation. The combined safe separation zone is denoted by a circle whose radius is the sum of the safe separation distances of the own and target ships[32].

$$\mathcal{J} = \mathcal{J}_{drift} + \mathcal{J}_{diff} \quad (5.16)$$

which describes the flux originating from the translation and expansion of the probability ellipse, respectively. The drift component is defined as[32]

$$\mathcal{J}_{drift} = f(\mathbf{x}, t)\mathbf{v} \quad (5.17)$$

where \mathbf{v} is the relative velocity between the own-ship and the obstacle. Further, the diffusion component is defined as

$$\mathcal{J}_{diff} = -D(t)(\nabla f(\mathbf{x}, t)) \quad (5.18)$$

where $D(t)$ is the diffusion coefficient and ∇ denotes the differential operator with respect to the obstacle position. Then, the rate of change of ICP can be found as the flux integral along the safety zone boundary

$$\begin{aligned} \dot{\mathbb{P}}_c^i &= - \iint_S (\nabla \cdot \mathcal{J}) dS \\ &= - \iint_{\mathcal{L}} ((-D(\nabla p(\mathbf{x}; \boldsymbol{\mu}, \Sigma)) + \mathbf{v}p(\mathbf{x}; \boldsymbol{\mu}, \Sigma)) \cdot \mathbf{n}) d\mathcal{L} \end{aligned} \quad (5.19)$$

where S is the conflict zone area, \mathcal{L} is the combined safety zone boundary and \mathbf{n} is the normal vector pointing outwards from the safety boundary. The ICP along the own-ship trajectory can then be estimated using (5.14).

The method also proposes to use the ICP in a collision risk index, using also the relative velocity \mathbf{v} between the own-ship and obstacle. This index is defined as

$$\Gamma_c = \eta \int_{t_0}^{t_f} \mathbb{P}_c^i(\tau) \|\mathbf{v}\|^2 d\tau \quad (5.20)$$

where η is determined by the combined mass of the two objects, and $t \in [t_0, t_f]$ is the time interval considered. Thus, the index gives the collision risk based on the kinetic energy of the system with the two objects.

5.4 Remarks

Of the two above described methods, the probability flow method has the strongest assumptions, namely that of a Wiener process describing the obstacle motion, with Gaussian distributed process noise. As the own-ship and obstacle speeds are assumed constant with no uncertainty, only the position uncertainty are considered. This assumption will clearly not hold in all cases, especially not with small boats with high speed and maneuverability showing varying behavior, and is a strong simplification. However, for large tankers with little maneuverability, the assumption can be acceptable.

Thus, depending on the type of obstacle, different models can be used and possibly combined to predict its motion. A Multiple Model (MM) approach [2] can for instance be used for obstacles with varying behavior, which may switch from one model to another based on the probability of a mode being in effect. There is however a trade off with complexity here, as the model framework increases, which must be taken into account.

The Monte Carlo integration method makes no assumptions on the underlying PDF for the obstacle uncertainty, and thus no assumptions about the obstacle motion are made. This makes the method applicable for both instantaneous and predictive collision probability calculation. However, a modelling choice for the obstacle motion must also be made here, and the same trade-off with complexity must be considered. The Monte Carlo integration method will be used in the rest of this thesis, due to it being able to handle uncertainty also in velocity, and because it is not directly clear from [32] how the probability flow method was implemented.

Chapter 6

Collision Probability Validation

The previous chapter presented two different methods for calculating the collision probability. In this chapter, the collision probability is calculated for three different examples, using Monte Carlo simulation. This is done to check whether the results make sense or not, as this is important to verify before using the probabilities in an SBMPC for collision avoidance. The first example calculates the probability when there is a static obstacle directly in the path of the own-ship, with only positional uncertainty considered. The other two examples deal with a dynamically moving obstacle, which requires that its velocity uncertainty is also considered when calculating the collision probability. The own-ship does not have an active COLAV system in these examples, and will thus not attempt to avoid collision.

The Monte Carlo Integration method as described in Section 5.2 is used to calculate the collision probability in order to check if reasonable results are obtained. The Monte Carlo simulation involves simulating the same example for different random number generator seeds, in order to sample different realizations of the random number sequences, in order to obtain an average result. The number of Monte Carlo simulations and number of random samples drawn in the Monte Carlo integration will be varied, in order to see how this affects the results.

6.1 Simulation Setup

A number of N_{MC} Monte Carlo simulations are performed on each example, where the seed of the random number generator, which is used to generate samples for the Monte Carlo Integration, is varied. The examples assume that a deterministic CVM is available, thus with the full state of the obstacle available, with no process noise. Further, the obstacle and own-ship are treated as point masses. There is in general a circular safety zone around the own-ship, with radius $d_{safe} = d_i^{safe}$, i.e. in the same way as for the SBMPC described in Section 3.3 and as was described in the previous chapter.

Further, a number of $n_{mc,int}$ samples are generated for the Monte Carlo integration. These two parameters are varied in order to see the effects on the collision probabilities. Here, the following values for N_{MC} and $n_{mc,int}$ are used:

- $N_{MC} \in \{1, 10, 100, 1000\}$
- $n_{mc,int} \in \{100, 1000, 10000, 50000\}$

A number of $N_{end} = 300$ samples are simulated with a sample time of $dt_{int} = 0.1$ s, i.e. each simulation/example lasts 30 seconds. Moreover, as there is only one obstacle in consideration, $i = 1 = n_{obst}$. The resulting collision probabilities $\mathbb{P}_{c,k}^i$ for each sampling instant $k = 0, 1, \dots, N_{end}$ (not to be mistaken as the scenario number in the SBMPC) are averaged over the amount of Monte Carlo simulations N_{MC} . Matlab R2018b is used as the programming software for all the simulations in this thesis.

6.2 Example 1: Static Obstacle

This example was considered as a first step in finding a general method for finding the collision probability (5.5) which considers uncertainty in both position and velocity. Thus, in this example, only the positional uncertainty is considered, and it is the integral (5.4) that is estimated here.

6.2.1 Setup

When evaluating the conflict probability or probability of collision in this example, the safety zone around the own-ship is not represented as a circle, as was shown and used by the SBMPC in Section 3.3, but instead as a corridor (corr) of

width $2d_{corr}$ stretching from the own-ship and ahead along a straight line up to a predefined distance l_{corr} , where its slope is given by the own-ship heading ψ . This is illustrated in Figure 6.1, and is similar to the approach in [30] and [31].

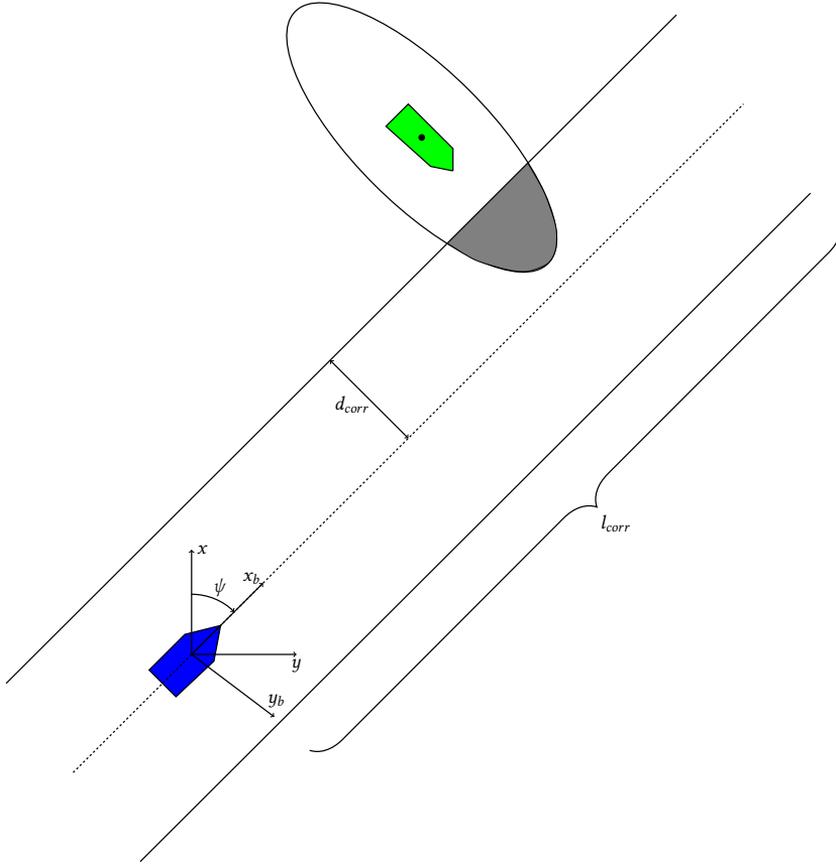


Figure 6.1: Illustration of the safety corridor from the own-ship (blue) along a straight line, with the uncertainty ellipse of an obstacle (green) included (assumed to be covering the entire probability mass). The conflict zone is here shown as the filled grey area.

The parameter values for d_{corr} and l_{corr} used here are given in Table 6.1, and are essentially equal to the SBMPC parameters d_{safe} and d_{close} described in Section 3.3.

In this example, the own-ship is travelling due north with constant speed of $u = 5 \text{ m s}^{-1}$ starting in the origin $(x, y) = (0, 0)$, and a static obstacle is placed at coordinates $(x^i, y^i) = (100, -46.5)$, where x^i and y^i are the xy -coordinates of the

Table 6.1: Safety corridor parameters.

Parameter	Value	Unit
d_{corr}	40.0	m
l_{corr}	200.0	m

obstacle (with index 1). This is done for the purpose of validating the collision probability for a simple starting case. The obstacle position is assumed to be constant and known by the own-ship through a constant Gaussian distribution with expectation

$$\boldsymbol{\mu}^i = \begin{bmatrix} x^i \\ 0 \\ y^i \\ 0 \end{bmatrix} \quad (6.1)$$

and with a covariance

$$\boldsymbol{P}^i = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.2)$$

The variance in x^i and y^i of 25 m^2 for the obstacle is deemed reasonable based on the results in [43]. The own-ship will here collide with the obstacle at around $t_c = 20 \text{ s}$ with probability $\mathbb{P}_{c,exact}^i = 0.0985$, and pass through it with constant speed, as the collision dynamics are not accounted for in the simulator. The "exact" probability $\mathbb{P}_{c,exact}^i$, as given by (5.4) with integration limits given by the safety corridor

$$\mathbb{P}_c^i(t) = \int_0^{l_{corr}} \int_{-d_{corr}}^{d_{corr}} p_{i,xy}(\mathbf{x}; t) dx dy \quad (6.3)$$

was calculated by discretizing the safety corridor in the body x- and y-direction with discretization step length $\Delta x = \Delta y = 0.1 \text{ m}$, and summing the integrand (obstacle PDF) for the xy values. After passing/colliding with the obstacle, the collision probability will quickly decrease to zero. The example is illustrated in

Figure 6.2, which is found in the next subsection.

6.2.2 Importance Sampling Scheme

Two strategies $\pi_1(\mathbf{x}, t)$ and $\pi_2(\mathbf{x}, t)$ (importance functions) for generating random position samples for the Monte Carlo integration method are used here. The first one samples from a 2 dimensional multivariate Gaussian (MVG) for the obstacle position

$$\pi_1(\mathbf{x}, t) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ci}^b, \mathbf{P}_{ci}^b) \quad (6.4)$$

referenced to the body fixed frame, which is essentially the obstacle position uncertainty projected (rotated and translated) into the safety corridor, with variances aligned with the corridor. The variance along the y_b axis is designed to be equal to d_{corr} , and the variance along the x_b axis will be equal to the aligned x or y variance of the obstacle uncertainty, depending on its orientation.

The importance center $\boldsymbol{\mu}_{ci}^b$ is found by calculating the closest point of approach (CPA) between the own-ship (vessel A) and obstacle (vessel B) [17]

$$t_{CPA} = \begin{cases} 0, & \text{if } \|\mathbf{v}_A - \mathbf{v}_B\| < \epsilon_{CPA} \\ -\frac{(\mathbf{p}_A - \mathbf{p}_B) \cdot (\mathbf{v}_A - \mathbf{v}_B)}{\|\mathbf{v}_A - \mathbf{v}_B\|^2}, & \text{otherwise} \end{cases} \quad (6.5)$$

which gives that

$$\boldsymbol{\mu}_{ci}^b = \mathbf{p}_A + \mathbf{v}_A t_{CPA} \quad (6.6)$$

It is assumed here that the positions and velocities are represented in the NED frame. The parameter ϵ_{CPA} was here chosen to be 0.5 m s^{-1} . The y_b component will naturally be 0 as the own-ship and obstacle are assumed to follow straight lines in the instant when the probability is calculated. The first step in finding the 2×2 importance Gaussian covariance matrix \mathbf{P}_{ci}^b in body is by aligning the obstacle uncertainty using a rotation matrix \mathbf{R}_{align} , such that

$$\bar{\mathbf{P}}_{ci}^b = \mathbf{R}_{align} \mathbf{P}_{xy}^i \mathbf{R}_{align}^T \quad (6.7)$$

where \mathbf{P}_{xy}^i is the position only covariance of obstacle i represented in NED, extracted from \mathbf{P}^i . The angle α of the rotation is found by first finding the orientation θ_i of the obstacle uncertainty ellipse with respect to the NED frame,

which is given by the direction of the largest eigenvector \mathbf{e}_l of the covariance matrix \mathbf{P}_{xy}^i :

$$\theta_i = \text{atan}\left(\frac{e_{l,y}}{e_{l,x}}\right) \quad (6.8)$$

Depending on the value of $\psi - \theta_i$, the angle α is then given as

$$\alpha = \begin{cases} \psi - \theta_i, & |\psi - \theta_i| \leq \frac{\pi}{4} \\ \psi + \frac{\pi}{2} - \theta_i, & \frac{\pi}{4} < \psi - \theta_i \leq \frac{3\pi}{4} \\ \psi + \pi - \theta_i, & \frac{3\pi}{4} < \psi - \theta_i \leq \frac{5\pi}{4} \\ \psi + \frac{3\pi}{2} - \theta_i, & \frac{5\pi}{4} < \psi - \theta_i \leq \frac{7\pi}{4} \end{cases} \quad (6.9)$$

where it is assumed that $\psi - \theta_i \geq 0$ for the 3 last cases in (6.9). The rotation of α aligns the obstacle uncertainty ellipse such that the ellipse eigenvector which is closest to being parallel to the corridor, is rotated to being parallel. The variance along this rotated eigenvector is used for the Gaussian importance variance along the x_b axis. Hence, the importance function covariance \mathbf{P}_{ci}^b is found as

$$\mathbf{P}_{ci}^b = \begin{bmatrix} \sigma_{ci,x}^2 & 0 \\ 0 & \sigma_{ci,y}^2 \end{bmatrix} = \begin{cases} \text{diag}\left([\bar{P}_{ci,11}^b \quad d_{corr}^2]\right), & \text{if } \alpha = \psi - \theta_i \text{ or } \alpha = \psi + \pi - \theta_i \\ \text{diag}\left([\bar{P}_{ci,22}^b \quad d_{corr}^2]\right) & \text{otherwise} \end{cases} \quad (6.10)$$

where $\bar{P}_{ci,11}^b$ and $\bar{P}_{ci,22}^b$ are the x_b and y_b variance of $\bar{\mathbf{P}}_{ci}^b$, respectively.

The other random sampling method uses a uniform distribution $\mathcal{U}(y; a, b)$ with $a = -b = d_{corr}$ along the y_b direction, and a scalar normal distribution $\mathcal{N}(\mathbf{x}; \mu_{ci,x}^b, \sigma_{ci,x}^2)$ along the x_b axis with mean $\mu_{ci,x}^b$ and variance $\sigma_{ci,x}^2$. Thus, the other sampling density becomes

$$\pi_2(\mathbf{x}, t) = \mathcal{U}(y; -d_{corr}, d_{corr})\mathcal{N}(\mathbf{x}; \mu_{ci,x}^b, \sigma_{ci,x}^2) \quad (6.11)$$

Figure 6.2 illustrates the sampling scheme for $\pi_1(\mathbf{x}, t)$.

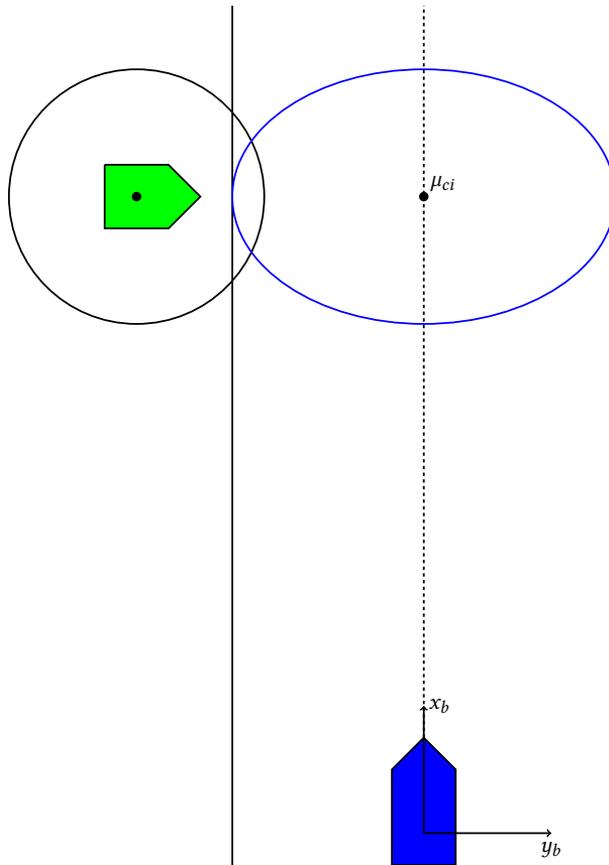


Figure 6.2: Illustration of the first example, with the own-ship in blue and obstacle drawn in green with a sample orientation of $\psi_1 = \frac{\pi}{2}$. Moreover, a sample 3σ (3 standard deviations) probability ellipse of the obstacle positional uncertainty is shown in black. The 1σ probability ellipse of the multivariate Gaussian $\pi_1(\mathbf{x}, t)$ used for random sampling in the first method is indicated in blue. The other sampling method $\pi_2(\mathbf{x}, t)$ will have constant density in the y_b direction, and equal variance as $\pi_1(\mathbf{x}, t)$ in the x_b direction.

6.2.3 Results

The resulting collision probability $\mathbb{P}_{c,k}^i$ from simulation using the first and second importance strategy $\pi_1(\mathbf{x}, t)$ and $\pi_2(\mathbf{x}, t)$ with the different values of N_{MC} and $n_{mc,int}$, are shown in Figures 6.3 and 6.4, respectively. A comparison of the two strategies when $N_{MC} = 1000$ and $n_{mc,int} = 50000$ is shown in Figure 6.5. Time is shown on the x-axis instead of sample number k in the plots.

From the Figures one can see that the collision probability calculated using Monte Carlo integration, for both sampling strategies, varies around the "exact" value of 0.0985 in the first 17 seconds, before it reduces to zero as the own-ship passes the obstacle. From for instance Figure 6.3, it is seen that the variation in the collision probability decreases with the amount of samples $n_{mc,int}$ used in the Monte Carlo integration. Furthermore, it is seen that the amount of Monte Carlo simulations N_{MC} does not affect the results. This is as expected, because the amount of Monte Carlo simulations will in this case only give a more averaged result.

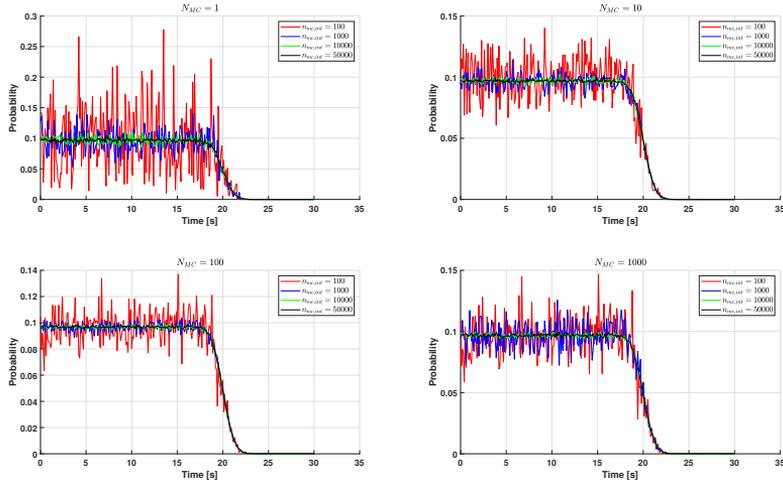


Figure 6.3: Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the first importance sampling method $\pi_1(\mathbf{x}, t)$.

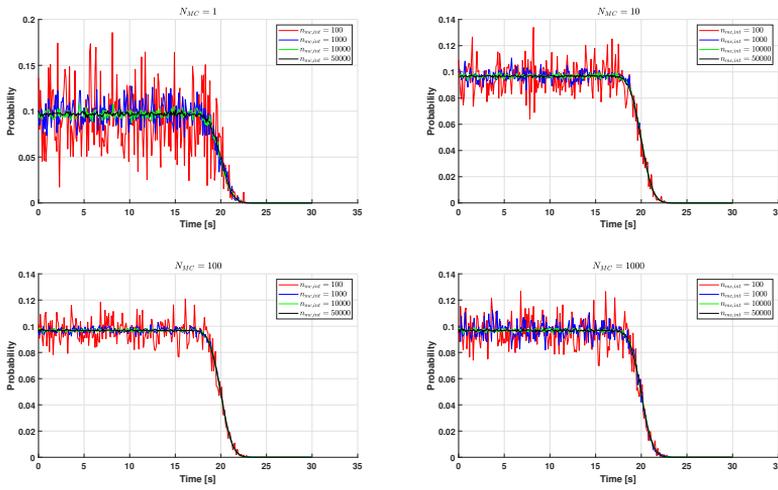


Figure 6.4: Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the second importance sampling method $\pi_2(\mathbf{x}, t)$.

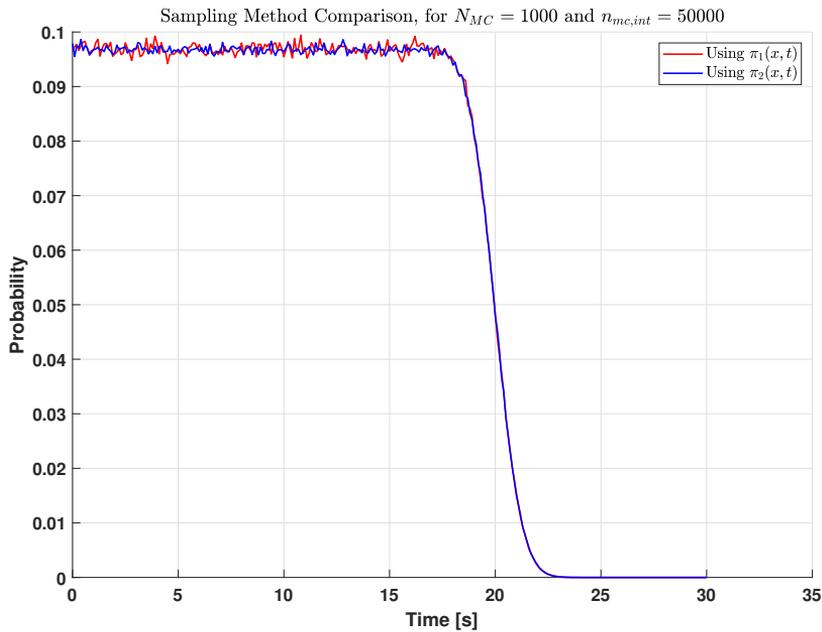


Figure 6.5: Collision probability $\mathbb{P}_{c,k}^i$ comparison for the two importance strategies, when $N_{MC} = 1000$ and $n_{mc,int} = 50000$.

Comparing the two strategies, there is no apparent difference in the resulting probabilities. However, by plotting the samples used in the MC integration, one can see that $\pi_2(\mathbf{x}, t)$ is clearly more effective, as the uniform sampling in the y_b direction is only done inside the safety corridor. This is shown in Figure 6.6. The multivariate Gaussian sampling wastes many samples here, as many of them are outside the integration domain inside the corridor.

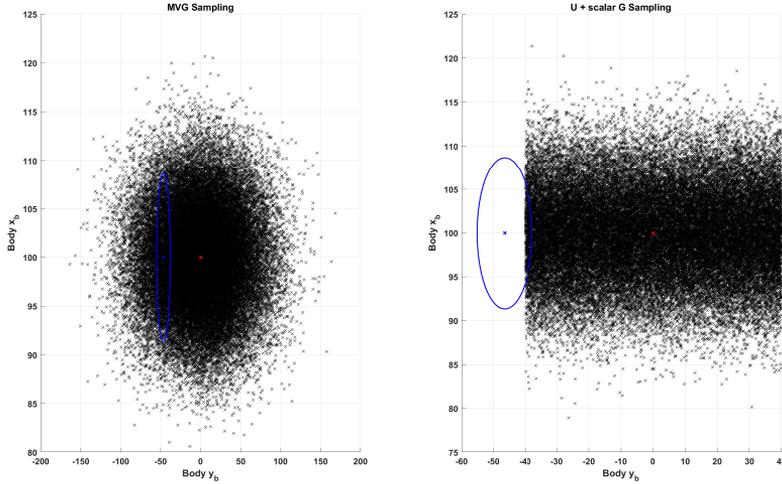


Figure 6.6: Samples generated by the two importance strategies $\pi_1(\mathbf{x}, t)$ (MVG Sampling) and $\pi_2(\mathbf{x}, t)$ (U (Uniform) + Scalar G (Gaussian) Sampling) with $n_{mc,int} = 50000$, referenced to the body frame. The red cross marks μ_{ci}^b , and the blue cross marks the obstacle position. The blue ellipse is the 3σ probability ellipse for the obstacle.

6.3 Example 2: Dynamic Obstacle 1

In this example and all the following, the safety zone for the own-ship is again chosen to be a circle, as is used in the SB-MPC algorithm, with radius of d_{safe} . Further, as uncertainty in the obstacle velocity is also included in the collision probability calculation of these two dynamic examples, the integral (5.5) is estimated here.

6.3.1 Setup

In the first example with a moving obstacle, the own-ship is standing still at $(x, y) = (100, 0)$, and the obstacle now starts at $(x^i, y^i) = (55, -55)$, and is travelling east with constant speed $V_y^i = 4 \text{ m s}^{-1}$. Thus, the obstacle will pass the own-ship 5 m south of its circular safety zone. The example is shown in Figure 6.7.

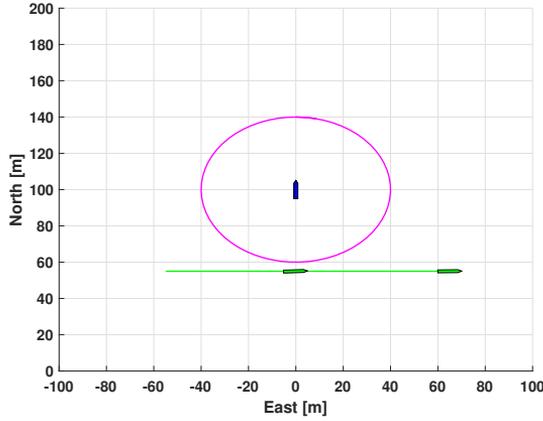


Figure 6.7: North-East plot of the obstacle (green) path (also green), with the own-ship plotted in blue, with its safety zone of radius d_{safe} in purple. The obstacle is plotted at the CPA, directly below the own-ship, in addition to its position at the end of the simulation.

The obstacle covariance received by the collision probability evaluation module is chosen to be constant and equal to

$$P^i = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (6.12)$$

and its state vector is updated and assumed known directly (the full state) through a deterministic CVM with parameter $T = dt_{int} = 0.1 \text{ s}$, i.e. no process noise included. The same variance in position as for Example 1 is chosen. The variance in north and east velocities are chosen primarily based on a guess for a vessel of length in the range 20 – 40 meters, similar to the NTNU Gunnerus

vessel [29].

As the velocity and its corresponding uncertainty is also considered in evaluating the collision probability, it calls for a method to determine the integration limits in the integral (5.5). Here, this is done by creating a ray from each sample of position and velocity from the importance functions in the MC integration, and checking whether this ray intersects the safety zone at some time $t_c \geq t_k$, with t_k being the current time. This is repeated for each sample, and all non-intersecting rays give integrand value of zero. The ray can be described by

$$\mathbf{r} = \mathbf{p}_l + \mathbf{v}_l t \quad (6.13)$$

where $\mathbf{p}_l = [x_l \ y_l]^T$ and $\mathbf{v}_l = [V_{x,l} \ V_{y,l}]^T$ are the sampled obstacle position and velocity, and t is the time parameter for the ray \mathbf{r} .

The safety zone boundary around the own-ship can be described by the circle

$$(x_c - x)^2 + (y_c - y)^2 = (d_{safe})^2 \quad (6.14)$$

where x and y are the own-ship north- and east coordinates. This can be represented in vector form as

$$\mathbf{p}_O^T \mathbf{p}_O - 2\mathbf{p}_O^T \mathbf{p}_c + \mathbf{p}_c^T \mathbf{p}_c = (d_{safe})^2 \quad (6.15)$$

where $\mathbf{p}_O = [x \ y]^T$ and $\mathbf{p}_c = [x_c \ y_c]$. Inserting \mathbf{p}_c for the ray equation (6.13), the second order polynomial

$$At^2 + Bt + C = 0 \quad (6.16)$$

with

$$A = \mathbf{v}_l^T \mathbf{v}_l \quad (6.17a)$$

$$B = 2(\mathbf{p}_l^T - \mathbf{p}_O^T) \mathbf{v}_l \quad (6.17b)$$

$$C = \mathbf{p}_O^T \mathbf{p}_O - 2\mathbf{p}_O^T \mathbf{p}_l + \mathbf{p}_l^T \mathbf{p}_l - (d_{safe})^2 \quad (6.17c)$$

can be found. If the roots of this polynomial are real, the ray intersects the safety

zone at two time instants (with distinct roots) or one time instant (with repeated roots). This can then be used to count the sampled trajectories which cross the safety zone. However, a time constraint on what interval of time the obstacle has to be within in order for a collision to happen is not added here, except from the criteria that the collision time satisfies $t_c \geq t_k$.

6.3.2 Importance Sampling Scheme

Again, two strategies $\pi_3(\mathbf{x}, t)$ and $\pi_4(\mathbf{x}, t)$ were here tested. The first one samples from the obstacle Gaussian distribution directly, i.e

$$\pi_3(\mathbf{x}, t) = \mathcal{N}(\mathbf{x}; \mathbf{x}^i, \mathbf{P}^i) \quad (6.18)$$

This strategy can be considered as a brute force method for calculating the collision probability, where all possible obstacle trajectories are sampled, and the fraction of the number of intersecting trajectories with the safety zone to the total number of trajectories, are used as the collision probability estimate.

The strategy $\pi_4(\mathbf{x}, t)$ attempts to make the sampling more effective, by ideally sampling only the trajectories that may intersect with the safety zone at some future point in time, Thus, the importance function is a MVG

$$\pi_4(\mathbf{x}, t) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{eff}^i, \mathbf{P}_{eff}^i) \quad (6.19)$$

with expectation $\boldsymbol{\mu}_{eff}^i$ and covariance \mathbf{P}_{eff}^i . The position part of the expectation is just taken as the current obstacle expected position, i.e. $\mathbf{p}_{eff}^i = [x^i \quad y^i]$. The velocity part of the expectation was found by assuming that the obstacle position at CPA is at the own-ship position at CPA, i.e. a direct collision with full overlap between the two objects:

$$\mathbf{v}_{eff}^i = \begin{bmatrix} v_{eff,x}^i \\ v_{eff,y}^i \end{bmatrix} = \frac{1}{t_{CPA}} (\mathbf{p}_{CPA} - \mathbf{p}^i) \quad (6.20)$$

Thus, this velocity vector points towards the own-ship position at CPA. As the obstacle (i) has a max speed U_{max}^i , the velocity vector \mathbf{v}_{eff}^i is saturated to

$$\mathbf{v}_{eff}^i = \begin{bmatrix} U_{max}^i \cos(\hat{\psi}_{eff}^i) \\ U_{max}^i \sin(\hat{\psi}_{eff}^i) \end{bmatrix} \quad (6.21)$$

when $\|\mathbf{v}_{eff}^i\| > U_{max}^i$, where $\hat{\psi}_{eff}^i$ is a crude heading estimate of the obstacle, when \mathbf{v}_{eff}^i is taken as its velocity. The estimate is here calculated as

$$\hat{\psi}_{eff}^i = \text{atan}\left(\frac{v_{eff,y}^i}{v_{eff,x}^i}\right) \quad (6.22)$$

Then, the importance expectation is

$$\boldsymbol{\mu}_{eff} = \begin{bmatrix} x^i \\ v_{eff,x}^i \\ y^i \\ v_{eff,y}^i \end{bmatrix} \quad (6.23)$$

The covariance of the importance function is chosen to be on the form

$$\mathbf{P}_{eff}^i = \text{diag}\left([\mathbf{P}_{xy}^i \quad \mathbf{P}_{eff,\mathbf{v}}^i]\right) \quad (6.24)$$

where \mathbf{P}_{xy}^i is the 2×2 position covariance part of \mathbf{P}^i , and $\mathbf{P}_{eff,\mathbf{v}}^i$ is the 2×2 velocity covariance of the importance PDF, ideally designed with variance such that all sampled trajectories intersect the own-ship safety zone. The position-velocity covariance part is designed to be zero for simplicity.

The velocity covariance is heuristically based on the difference velocity $\mathbf{v}_{diff} = \mathbf{v}_{eff}^i - \mathbf{v}^i$ using the variance expression

$$\text{var}_{u,a,b} = \frac{(a-b)^2}{12} \quad (6.25)$$

for a uniform distribution, with a and b as the interval limits of the distribution, here taken as $a = \mathbf{v}_{eff}$ and $b = \mathbf{v}^i$. The velocity covariance for the importance PDF was then chosen as

$$\mathbf{P}_{eff,\mathbf{v}}^i = \begin{bmatrix} \frac{\mathbf{v}_{diff,x}^2}{12} & 0 \\ 0 & \frac{\mathbf{v}_{diff,y}^2}{12} \end{bmatrix} \quad (6.26)$$

as a first trial in finding a more effective importance sampling, than the one used in $\pi_3(\mathbf{x}, t)$. The importance strategy is attempted illustrated in Figure 6.8.

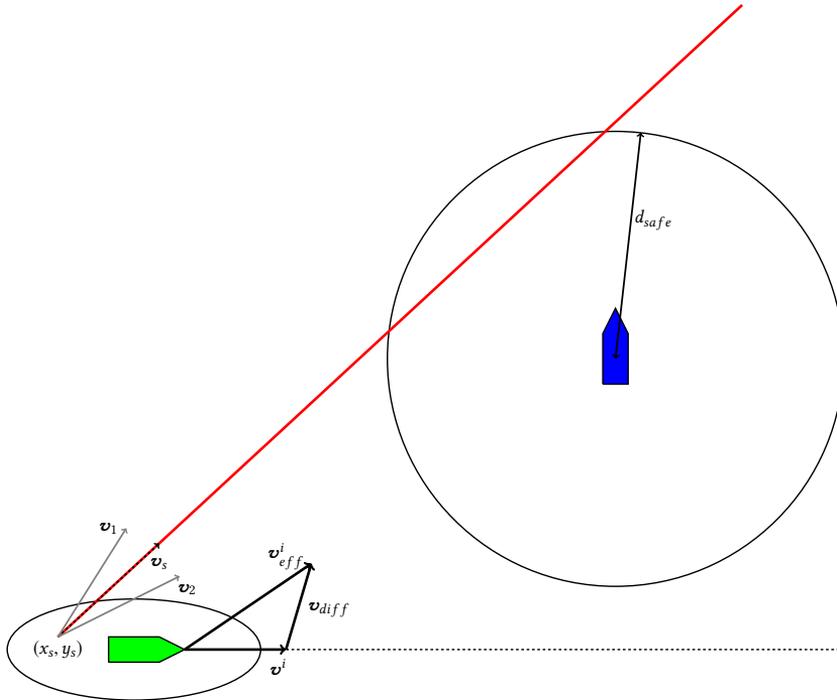


Figure 6.8: Illustration of the attempted more effective sampling scheme for Example 2, with the own-ship in blue and obstacle drawn in green, with the circular safety zone of radius d_{safe} drawn around the own-ship. The obstacle velocity vector \mathbf{v}^i , importance function velocity expectation \mathbf{v}^i_{eff} and their difference \mathbf{v}_{diff} are also indicated, drawn from the tip of the obstacle, instead of the point mass center. The 1σ probability ellipse of the obstacle position uncertainty is shown, and a sampled trajectory starting at a position (x_s, y_s) with velocity vector \mathbf{v}_s is drawn. An example standard deviation in velocity for the particular importance sample is shown for the velocity vector \mathbf{v}_s , in \mathbf{v}_1 and \mathbf{v}_2 , which is calculated based on the difference velocity \mathbf{v}_{diff} .

6.3.3 Results

The resulting collision probability $\mathbb{P}_{c,k}^i$ from simulation using the third and fourth importance strategy, $\pi_3(\mathbf{x}, t)$ and $\pi_4(\mathbf{x}, t)$, for the different values of N_{MC} and $n_{mc,int}$, are shown in Figures 6.9 and 6.10. Results when $N_{MC} = 1000$ have been dropped due to excessive simulation time required. A comparison plot of the two strategies have also been dropped due to the less effective strategy not being a working approach. Time is shown on the x-axis instead of sample number k in the plots.

From Figure 6.9 one can see that the collision probability starts at about 0.4, before it decreases to zero as the obstacle approaches and passes by the own-ship. This makes sense, as the probability will be higher in the start due to a larger amount of trajectories which can cross the safety zone are possible, and this amount decreases as the obstacle gets closer.

The less effective approach shows relatively stable collision probabilities when the number of samples drawn $n_{mc,int}$ is sufficiently high, as seen from Figure 6.9. This is as expected, as all possible trajectories from the obstacle PDF are sampled, and one merely need to count the ones intersecting with the safety zone to obtain an estimate of the probability. The variance in this estimate naturally decreases with the number of samples used.

One can see large fluctuations in the collision probability when using the attempted more effective importance sampling, which can be seen from Figure 6.10. This can be explained using Figure 6.11, which shows the samples in position and velocity from strategy $\pi_4(\mathbf{x}, t)$ when $n_{mc,int} = 10000$. Here, one can see that the obstacle velocity expectation is relatively far from the importance function velocity expectation, which results in relatively few samples inside the illustrated integration region (only an example region, not the actual one) where the samples are valid (non-zero and used in the MC integration). Variations in these few samples give rise to large variations in the MC integration. This is a problem with the attempted effective sampling, as one uses the velocity that the obstacle would have if it reached the same position at CPA as the own-ship, which may differ quite from the current expected obstacle velocity.

Another issue is the velocity variance used in this importance sampling, which here was taken as a uniform variance expression in the difference velocity \mathbf{v}_{diff} . This variance is heuristically based, and will not be a true representation of the variance which give samples only inside the region where sampled trajectories intersect with the safety zone. More work is here needed to refine this

approach.

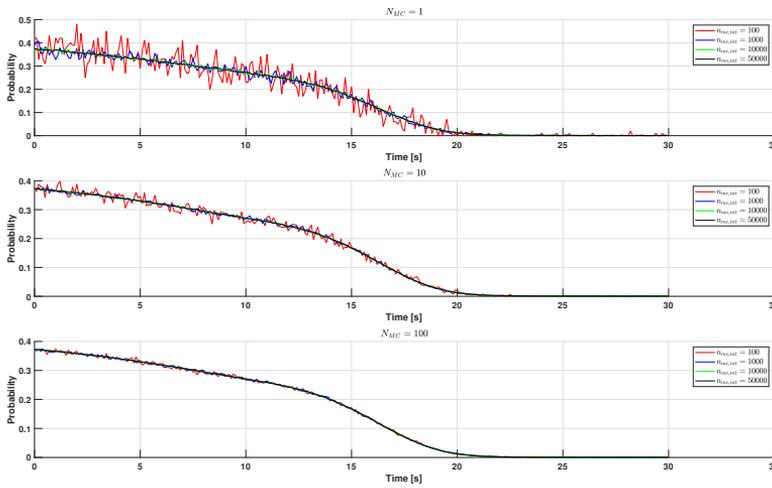


Figure 6.9: Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the third importance sampling method $\pi_3(\mathbf{x}, t)$.

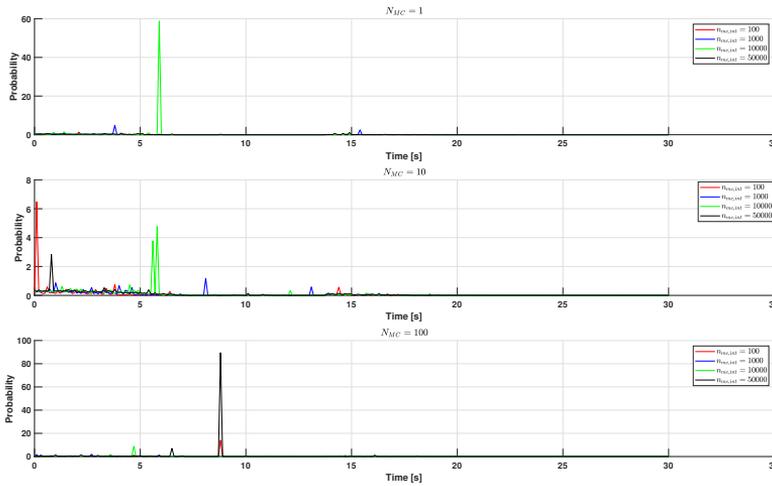


Figure 6.10: Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the fourth importance sampling method $\pi_4(\mathbf{x}, t)$.

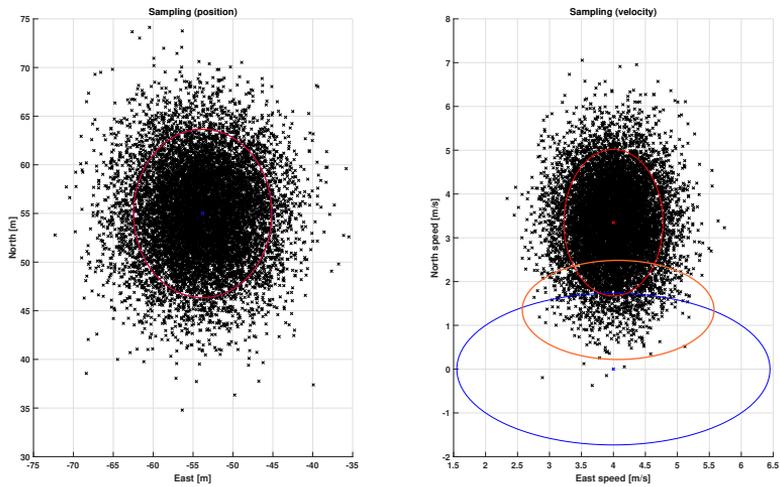


Figure 6.11: Sampling plot in position and velocity for the fourth importance strategy $\pi_4(\mathbf{x}, t)$ in the NED frame. The 3σ probability ellipse for the importance function is shown in red, and the same for the obstacle in blue. The red- and blue crosses mark their expectations, respectively. For the velocity sampling, an example integration region of where the samples are valid, i.e. nonzero, is shown in the orange ellipse. For the position sampling, the importance strategy uses the same covariance and expectation as the obstacle's, and the two probability ellipses thus overlap.

A crude estimate of the average computation time per collision probability evaluation for the less effective third importance strategy $\pi_3(\mathbf{x}, t)$ was also calculated. Running on an Intel i7 - 7700 CPU with 32 GB installed RAM on a 64-bit operating system with almost no running background programs, the results are summarized for the different random sample numbers $n_{mc,int}$ in Table 6.2.

Table 6.2: Average computation time \bar{t}_{π_3} per collision probability evaluation for the strategy $\pi_3(\mathbf{x}, t)$, for the different values of $n_{mc,int}$, with $N_{MC} = 10$.

$n_{mc,int}$	\bar{t}_{π_3}
100	0.0013 s
1000	0.011 s
10000	0.11 s
50000	0.50 s

One can see here that relatively high computation times are needed per evaluation of the collision probability, with around 1 ms being the lowest average time.

6.4 Example 3: Dynamic Obstacle 2

6.4.1 Setup

In this example, both the obstacle and own-ship are moving. The obstacle starts in $(x^i, y^i) = (100, 100)$, and is travelling south-west with velocity $\mathbf{v}^i = [-2.5 \quad -4]^T$. The own-ship starts in the origin, and travels straight north with forward speed $u = 5 \text{ m s}^{-1}$. A deterministic CVM is again employed for the obstacle, and its covariance is again set to be constant, and the same as in (6.12). The example is shown in Figure 6.12, which shows the obstacle and own-ship trajectories in the NE-plane.

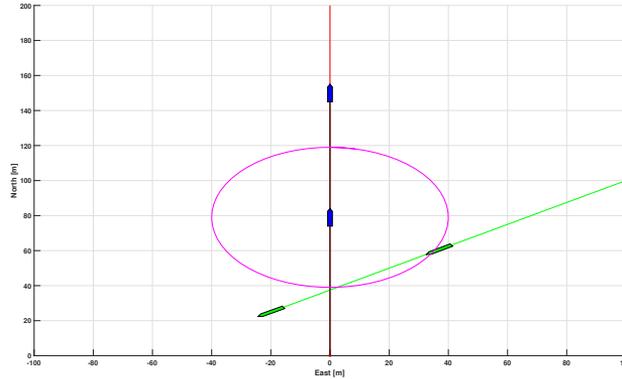


Figure 6.12: North-East plot of the obstacle path (both in green), with the own-ship (blue) path plotted in black, with its position and safety zone of radius d_{safe} in purple drawn at the CPA. The obstacle and own-ship are plotted at the CPA in addition to at their end positions. The red line marks the own-ship planned path.

6.4.2 Importance Sampling Scheme

The scheme is the same as for Example 2, only that the attempt of a more effective sampling $\pi_4(\mathbf{x}, t)$ has been left out, due to it not being a working approach at this moment. Thus, only the less effective strategy $\pi_3(\mathbf{x}, t)$ is used from now on.

6.4.3 Results

The resulting collision probability $\mathbb{P}_{c,k}^i$ from simulation using the third importance strategy, $\pi_3(\mathbf{x}, t)$ for the different values of N_{MC} and $n_{mc,int}$, are shown in Figure 6.13. Results when $N_{MC} = 1000$ have again been dropped due to excessive simulation time required. Time is shown on the x-axis instead of sample number k in the plots.

In this example, the collision probability is predicted to be relatively high at just below 0.6, due to a high amount of possible trajectories which may cause collision. The probability reaches peak value at CPA around $t_{CPA} \approx 16$ s, as the obstacle is at this time almost inside the own-ship safety zone, and almost 80% of all possible trajectories will intersect with the zone. This is reasonable, considering the standard deviation of 5 m in the obstacle x - and y-position, and the velocity standard deviation of 1 m s^{-1} and 1.41 m s^{-1} in the x- and y-direction,

which amounts to a sector with relatively small spread, in which the obstacle future trajectory lies.

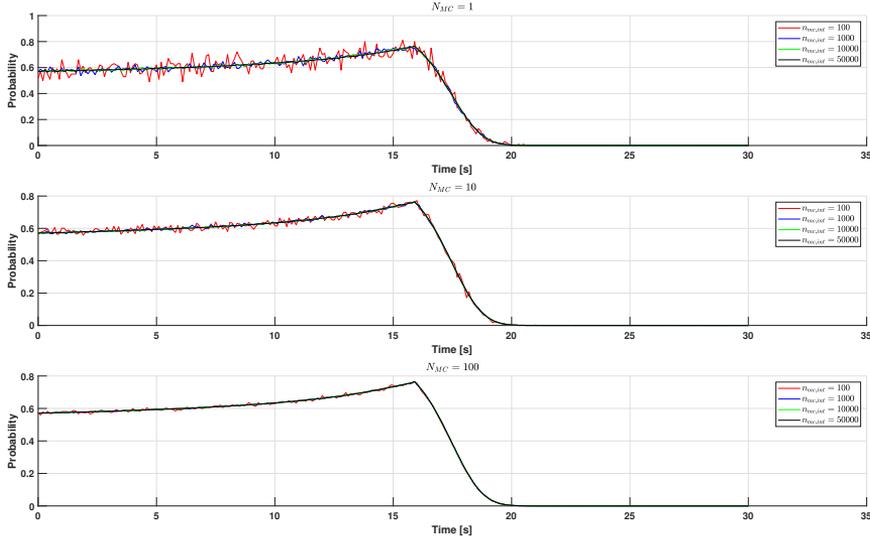


Figure 6.13: Collision probability $\mathbb{P}_{c,k}^i$ for different values of N_{MC} and $n_{mc,int}$, using the third importance sampling method $\pi_3(\mathbf{x}, t)$.

6.5 Discussion

The probability of collision were calculated for three examples, where the obstacle was static in the first, and dynamic in the two last. In the example with a static obstacle, only positional uncertainty was used in order to calculate the probability, which gave decent results, matching the "exact" collision probability quite well. From the results for this example, one need a minimum of $n_{mc,int} = 1000$ in order to have an acceptably low variance in the probability estimate.

When including velocity uncertainty in calculation of the probability as in the two other examples, a collision probability prediction method is obtained. The attempted more efficient method $\pi_4(\mathbf{x}, t)$ for evaluating the probability requires refining, and is a subject of future work. Results using MC integration with the less effective importance sampling gave reasonable estimates of collision

probability, but the computation time used per probability calculation is not acceptable for real-time use in an SBMPC. This can be illustrated with the following example calculation. With 39 possible scenarios, a prediction horizon of $T = 150$ s and sample time of $T_s = 0.1$ s in the SBMPC, the required total computation time for all collision probability evaluations in one run of the algorithm amounts to around $0.0013 \text{ s} \times 39 \times \frac{T}{T_s} = 76$ s, this when using $n_{mc,int} = 100$, which is not feasible for a real-time system.

It is therefore need for more effective evaluation methods, possibly similar to that of the probability flow method described in Section 5.3, but extended to be able to deal with velocity uncertainty. However, the less effective probability evaluation method developed here, can be used to benchmark more efficient methods in the future.

Note that only straight line motion were considered here. In practice, a dynamic obstacle can take a large variety of paths depending on for instance its maneuverability and the type of driver "behind the wheel". A suggestion here would be to use machine learning in order to infer the maneuvering behavior of a vessel based on for instance lots of AIS data of vessels travelling in a region. The resulting model would then be representative of that region only, and the type of vessels from which data are collected. The model could then be used to modify the predicted straight line path for the obstacle, or to make a switch between multiple models in a MM framework, as was discussed briefly in the previous chapter. This would then require the development of another type of importance sampling strategy if Monte Carlo integration is to be used, as more parameters than the current position and velocity would then be needed to predict the obstacle trajectory.

Chapter 7

A Probabilistic SBMPC

The previous chapters have described the SBMPC COLAV method and several methods for tracking obstacles. The tracking method yields a corresponding state estimate and error covariance. Further, a method for calculating collision probability has been presented, which will use the tracking method outputs, and work yet remains here in order to come up with a more effective and correct approach, as was discussed in the previous chapter.

In this chapter, the author aims at bringing all these parts together in one system, by altering the SBMPC to account for the probability of collision with obstacle i , which indirectly makes use of its kinematic uncertainty. The resulting probabilistic COLAV method is then tested and compared to the original deterministic approach in the end. This is done as an attempt of increasing the performance of the COLAV method.

7.1 SBMPC Cost Function Modifications

The SBMPC cost function or hazard corresponding to scenario k at the current time t_0 , is, as described in Section 3.3, given as (3.16)

$$\begin{aligned} \mathcal{H}^k(t_0) = & \max_i \max_{t \in D(t_0)} (C_i^k(t) \mathcal{R}_i^k(t) + \kappa_i \mu_i^k(t)) \\ & + f(\chi_m, \chi_{m,last}, u_m, u_{m,last}) + g(\cdot) \end{aligned} \quad (7.1)$$

The proposed modifications made by the author here in order to incorporate uncertainty into this framework, is to change the collision cost and risk term from $C_i^k(t)\mathcal{R}_i^k(t)$ to the following suggestions

$$aC_i^k(t)\mathbb{P}_c^i(t) \quad (\text{M1})$$

$$C_i^k(t)\mathcal{R}_i^k(t)\mathbb{P}_c^i(t) \quad (\text{M2})$$

$$aC_i^k(t)\mathbb{P}_c^i e^{-(t-t_0)} \quad (\text{M3})$$

where a is a constant parameter. The SBMPC with modifications to account for kinematic uncertainty (collision probability) will from now on be called the Probabilistic SBMPC (PSBMPC).

7.2 Simulation

7.2.1 Setup

The SBMPC and PSBMPC were simulated for an overtaking situation, a head-on situation and a crossing situation, with $n_{obs} = i = 1$ obstacle for simplicity. The number of samples N_{end} simulated for each situation is chosen to be $N_{end} = 800$, and a simulation step length of $dt_{int} = 0.1$ s was again used. A number of $N_{MC} = 1$ Monte Carlo simulations were performed, due to long simulation times required, and little time left until deadline.

In all the three situations, the own-ship starts in the origin at rest with $\mathbf{x} = \mathbf{0}$ (where \mathbf{x} here is the 6×1 own-ship state vector), and is set to travel north with desired surge speed $u = 9$ m s⁻¹. Waypoints are placed at (0, 0) and (800, 0) to achieve this through the Guidance system (see Section 2.2.2). Obstacle motion description:

- For the overtaking situation, the obstacle starts in $(x^i, y^i) = (100, 0)$ and initially travels north with speed $V_x^i = 3$ m s⁻¹.
- For the head-on situation, the obstacle starts in $(x^i, y^i) = (600, 0)$ and initially travels south with speed $V_y^i = -4$ m s⁻¹.
- For the crossing situation, the obstacle starts in $(x^i, y^i) = (400, -200)$ and initially travels towards east with speed $V_y^i = 4$ m s⁻¹.

Process noise will be added to the obstacle motion, making its path non - deterministic, and will simulate a sort of randomness in its maneuvering intent. This is achieved by varying the random number generator seed in Matlab before each simulation. A regular Kalman Filter as described in Section 4.1 is used to track the obstacle, using the stochastic CVM from Section 2.1.2, where it is assumed that position measurements are available. These measurements are here designed to have a covariance

$$\mathbf{R} = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \quad (7.2)$$

which the KF matches. The noise covariance values are deemed reasonable based on the results in [43]. The parameter T was set to $T = 0.1$ s for the tracking system, as this is the sample time in simulation. Then, with a chosen noise strength of $\sigma_a = 0.05$, the process noise covariance is then given by (2.11), which the KF also matches. This choice of σ_a was based on simulating a vessel with relatively low maneuverability [6][43]. The KF uses the same noise covariance matrices as the true model for simplicity. The initial a priori covariance in the KF-tracking of obstacle i for all three situations was here set to

$$\bar{\mathbf{P}}_0^i = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (7.3)$$

i.e. the same as chosen for the dynamic examples in the previous chapter. The initial a priori position estimate is set to be equal to the first measurement arriving (i.e. the recursive equations (4.3) and (4.5) are used for $k \geq 1$), and the initial a priori velocity estimate is set to be zero. Thus, the initial obstacle state estimate is

$$\bar{\mathbf{x}}_0^i = \begin{bmatrix} z_{0,x} \\ 0 \\ z_{0,y} \\ 0 \end{bmatrix} \quad (7.4)$$

For both versions of the SBMPC, the deterministic CVM is used to predict

the obstacle motion, initialized with the KF state estimate $\hat{\mathbf{x}}^i$ of the obstacle. The PSBMPC in addition uses the corresponding error covariance \mathbf{P}^i to evaluate the collision probability at all time steps in the prediction.

For the collision probability evaluation in the PSBMPC, the Monte Carlo integration method is used, considering both position and velocity in the calculation, as was used in the last two examples of the previous chapter. The less effective importance sampling method (6.18) is used

$$\pi_3(\mathbf{x}, t) = \mathcal{N}(\mathbf{x}; \mathbf{x}_p^i, \mathbf{P}^i) \quad (7.5)$$

with a number of $n_{mc,int} = 100$ samples drawn and used in the estimation of the collision probability (5.5), and where \mathbf{x}_p^i is the predicted obstacle state using the deterministic CVM, initialized to $\hat{\mathbf{x}}^i$ at the current simulation time step. The probability evaluation is done for each scenario and time step in the prediction of the PSBMPC.

Further, the guidance parameters from Table 2.2 and controller parameters from Table 2.1 are used for both SBMPC versions. The parameters for both versions are given in Table 7.1 below. The value of $n_{mc,int} = 100$ for the PSBMPC was chosen to be able to simulate faster, due to the relatively long computation time per probability evaluation, as was discussed in the previous chapter.

Table 7.1: Parameters used in the thesis for the two versions of the SBMPC.

	SBMPC	PSBMPC	
Parameter	Value		Unit
T	150	150	s
T_s	0.05	0.1	s
d_i^{safe}	40.0	40.0	m
d_{close}	200.0	200.0	m
p	1.0	1.0	
q	4.0	4.0	
K_i^{coll}	0.5	0.5	
κ_i	3.0	3.0	
K_{u_m}	2.5	3.0	
$K_{\Delta_{u_m}}$	2.0	2.5	
$K_{\chi, port}$	1.8	1.8	
$K_{\chi, starboard}$	1.5	1.5	
$K_{\Delta_{\chi, port}}$	1.2	1.2	
$K_{\Delta_{\chi, starboard}}$	0.9	0.9	
a		1	
$n_{mc, int}$		100	

Note that because the number of samples drawn in the Monte Carlo integration was chosen to be $n_{mc,int} = 100$, to reduce the computational time needed per probability evaluation, there will be a variance in the probability estimate which may affect the resulting PSBMPC behavior in the simulations. Ideally, this number should be chosen larger than 1000 in order to have a sufficiently low variance for use in the PSBMPC, based on the results from the previous chapter.

7.2.2 Results Modification 1

Here, results for the SBMPC and PSBMPC with the cost function modification (M1) are given for the three situations.

Overtaking

Figures 7.1 - 7.5 show the overtaking simulation results. A North-East plot of the vessel motion is shown in Figure 7.1, and a comparison of the heading response and surge speed response for the two versions in Figure 7.2 and 7.3, respectively. Lastly, the track estimates versus the true obstacle motion and the distance from the own-ship to the obstacle are shown in Figure 7.4 and 7.5, respectively.

From the results, one can see that the original SB-MPC performs decently, following COLREGS Rule 13 as the overtaking vessel (Figure 7.1), and keeping its distance to the obstacle (Figure 7.5). The PSBMPC on the other hand waits a good 15 s before it decides to drive forward and avoid the obstacle. This can be explained from the cost function alteration (M1). Here, the risk of collision \mathcal{R}_i^k in (3.2), which gives a non-zero value when the obstacle is inside the own-ship safety zone and based on the time it happens, is not considered anymore. Since only the collision cost and the corresponding collision probability are considered, the COLAV system becomes overly conservative and gives high hazard for all scenarios with $u_m \neq 0$.

Moreover, Figure 7.4 shows that the KF manages to track the obstacle, which is as expected, as the optimality criteria described in Section 4.1 are satisfied, and the process noise and measurement noise covariances are identical for the true model and the estimation model. The process noise added on the obstacle motion results in a Wiener process for its velocity, which in return results in different trajectories for each realization of the noise.

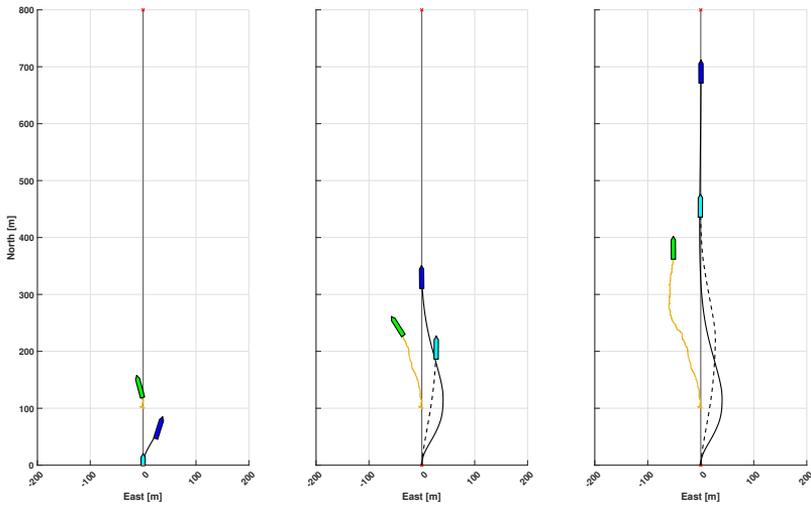


Figure 7.1: Overtaking situation with cost function alteration (M1). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

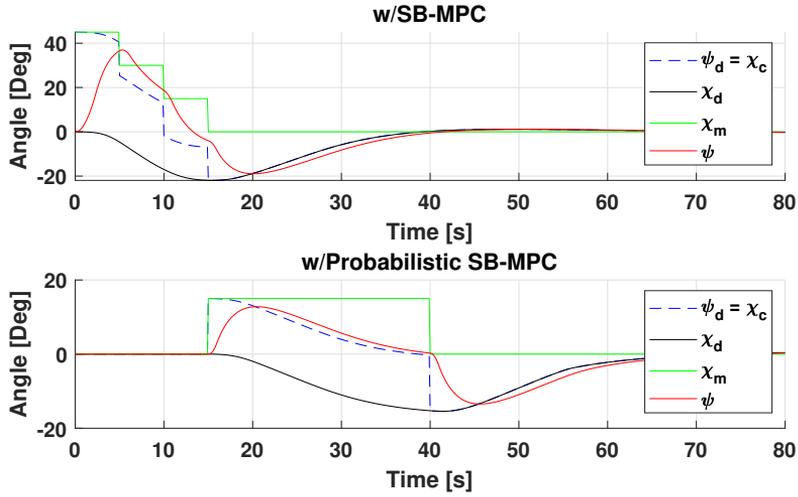


Figure 7.2: Overtaking situation with cost function alteration (M1). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

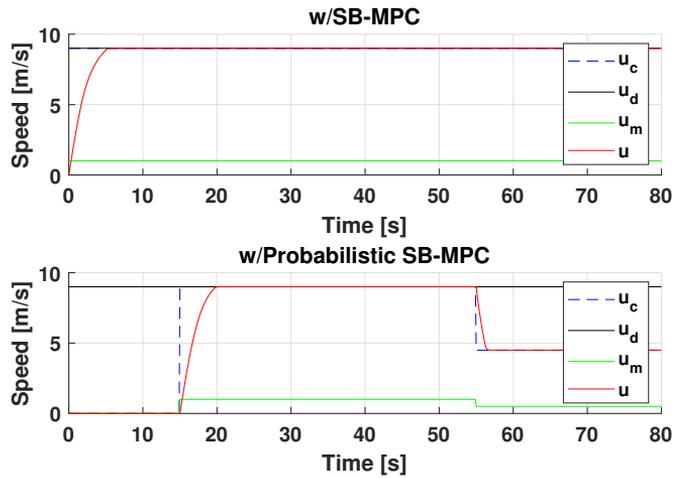


Figure 7.3: Overtaking situation with cost function alteration (M1). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

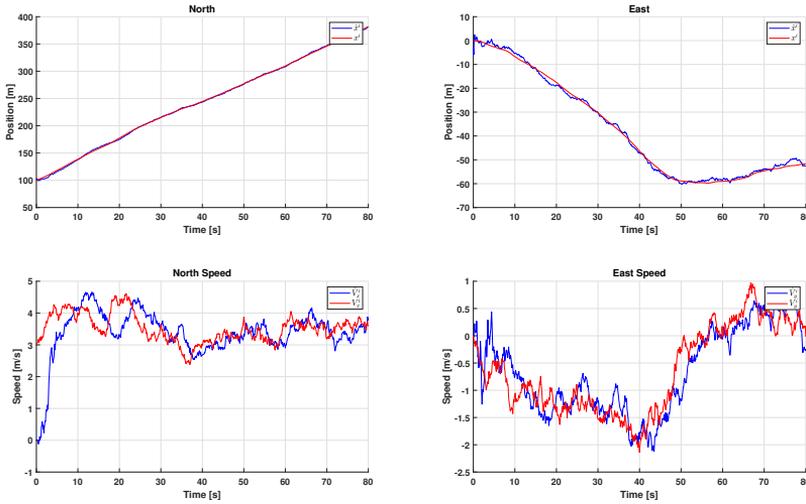


Figure 7.4: Overtaking situation with cost function alteration (M1). Track estimates for the obstacle, versus the true motion.

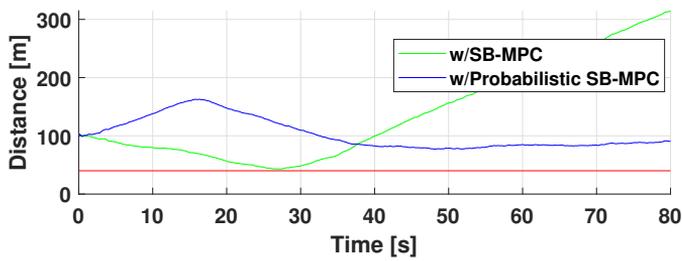


Figure 7.5: Overtaking situation with cost function alteration (M1). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

Head-On

Figures 7.6 - 7.10 show the head-on simulation results, with the same set of figures attached as for the overtaking situation. The PSBMPC again behaves overly cautious, by standing still and waiting out the situation for the first 15 s, before it this time also breaches COLREGS Rule 14 for a head-on situation. However, an argument can be made in that, since the obstacle turns to port, the own-ship with PSBMPC must also turn to port in order to avoid the possible head-on collision. The conservative behaviour in the PSBMPC is again caused by the risk of collision being omitted in the cost function. The original SBMPC on the other hand again performs correctly, following COLREGS Rule 14 by turning to starboard in order to avoid the obstacle. Some oscillations are seen in the heading response from both SBMPC versions in Figure 7.7, which are caused by a non-optimal tuning. A decrease in the path deviation penalty parameters $K_{\chi, \text{starboard}}$ and $K_{\chi, \text{port}}$ for the course offset should here be made to obtain smoother trajectories.

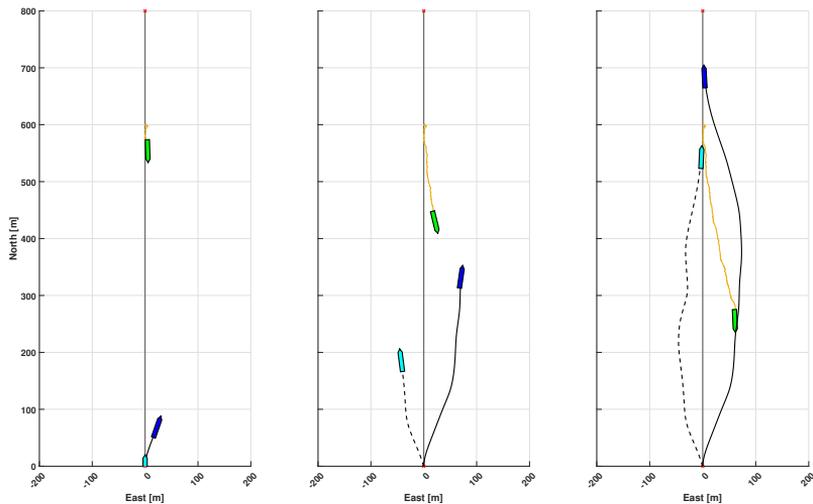


Figure 7.6: Head-on situation with cost function alteration (M1). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

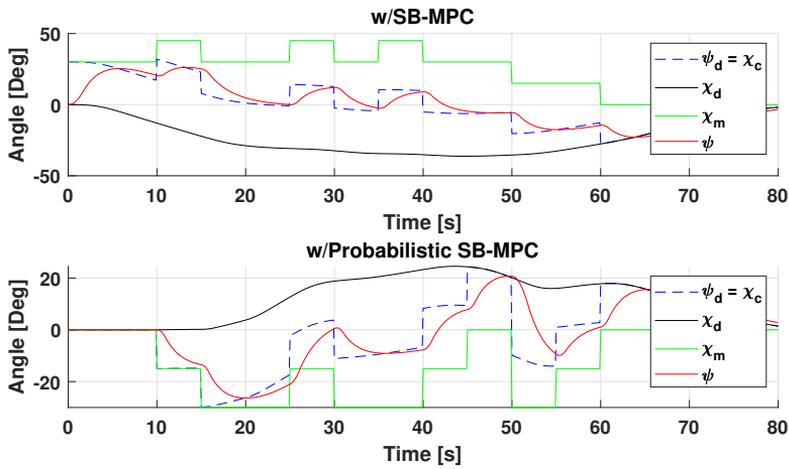


Figure 7.7: Head-on situation with cost function alteration (M1). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

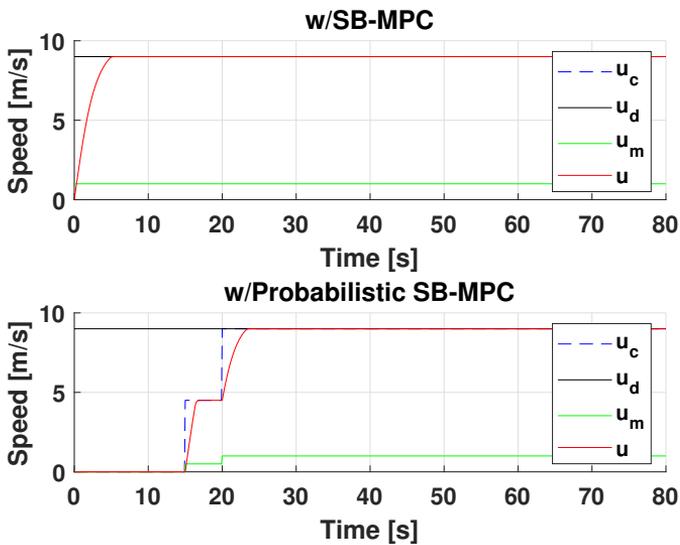


Figure 7.8: Head-on situation with cost function alteration (M1). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

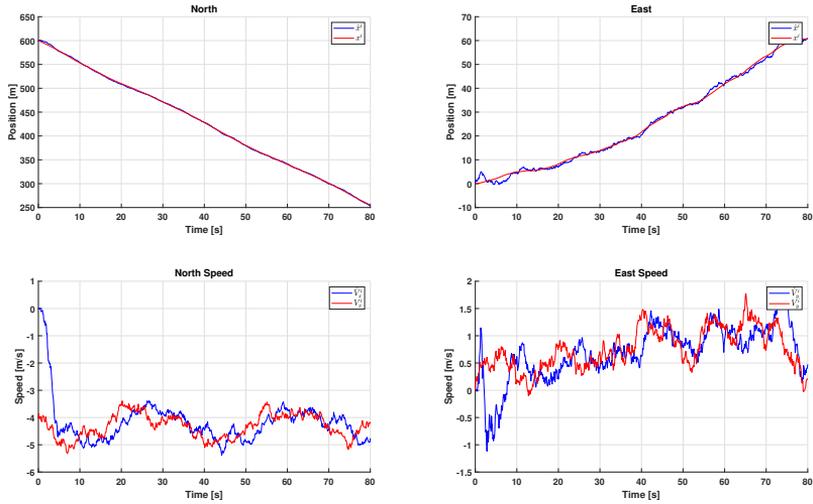


Figure 7.9: Head-on situation with cost function alteration (M1). Track estimates for the obstacle, versus the true motion.

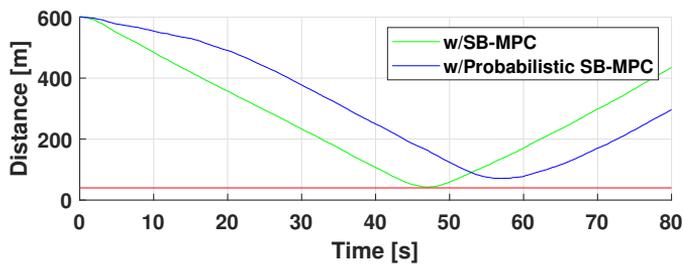


Figure 7.10: Head-on situation with cost function alteration (M1). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

Crossing

Figures 7.11 - 7.15 show the crossing simulation results. In the situation, the obstacle is meant to give-way for the own-ship, following COLREGS Rule 15, but this does not happen, such that the own-ship need to perform evasive maneuvers. One can again see that the PSBMPC behaves conservatively with the first cost function modification, due to high hazard for all $u_m \neq 0$, and waits until the obstacle has crossed its path before it starts increasing its forward speed, which can be seen from Figure 7.11 and 7.13. The original SBMPC performs decently, taking early preemptive action in order to avoid collision.

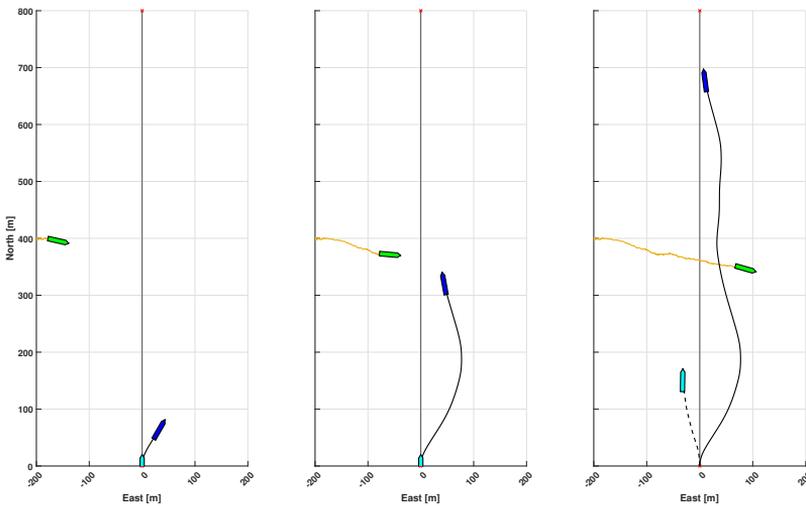


Figure 7.11: Crossing situation with cost function alteration (M1). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

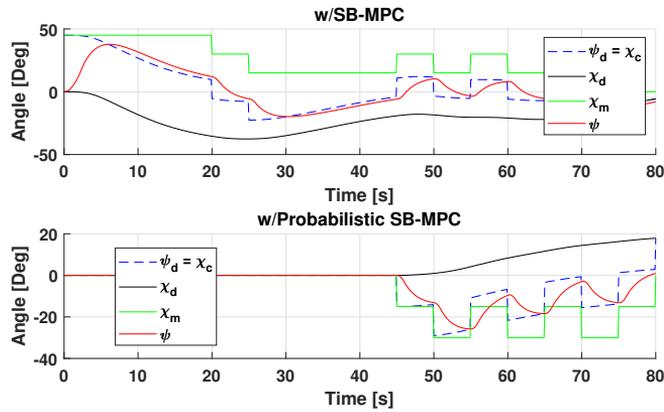


Figure 7.12: Crossing situation with cost function alteration (M1). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

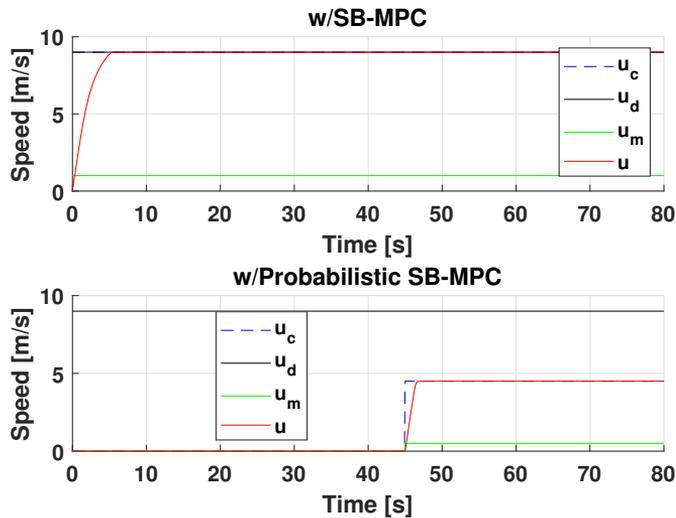


Figure 7.13: Crossing situation with cost function alteration (M1). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

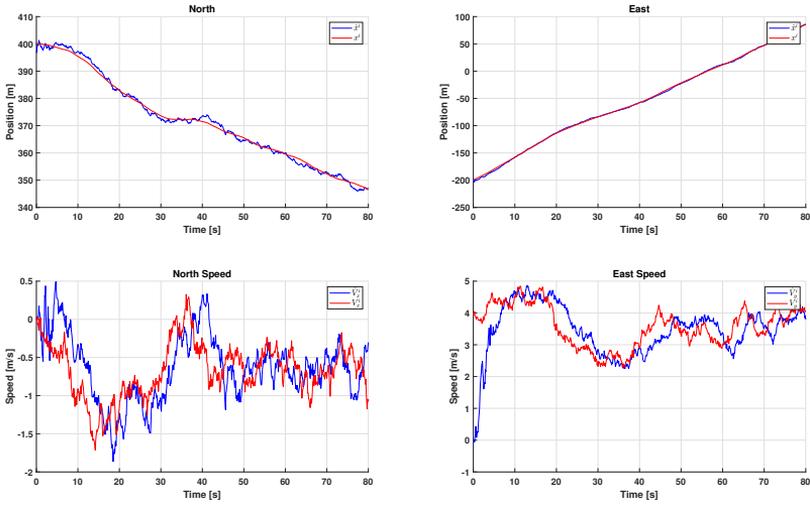


Figure 7.14: Crossing situation with cost function alteration (M1). Track estimates for the obstacle, versus the true motion.

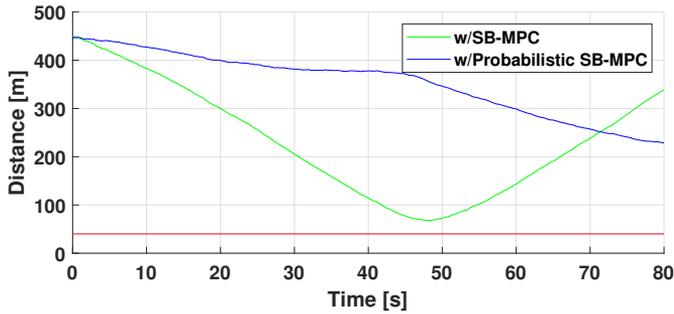


Figure 7.15: Crossing situation with cost function alteration (M1). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

7.2.3 Results Modification 2

Here, results for the SBMPC and PSBMPC with the cost function modification (M2) are given for the three situations.

Overtaking

Figures 7.16 - 7.20 show the overtaking simulation results. One can see from the figures that the SBMPC and PSBMPC perform identically for this case, with no added benefit of the cost function alteration (M2) here. This can be explained from the fact that the collision probability $\mathbb{P}_c^i(t)$ becomes close to or equal to 1 before t_{CPA} for the majority of the SBMPC scenarios, and thus the max value of the modified collision risk and cost term (M2) will be nearly identical to the max value of $C_i^k \mathcal{R}_i^k$ in the original cost function, when $a = 1$ in the PSBMPC.

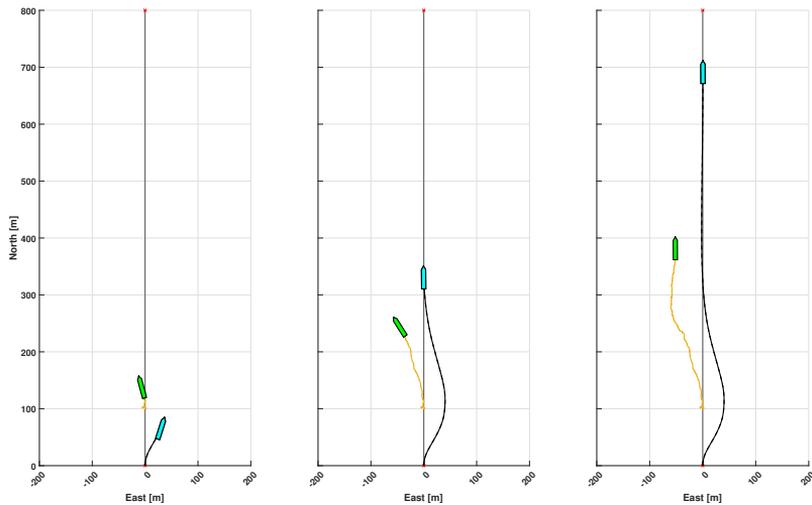


Figure 7.16: Overtaking situation with cost function alteration (M2). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

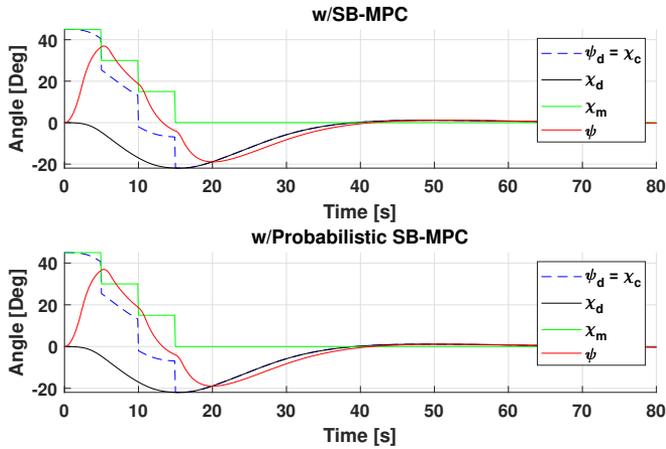


Figure 7.17: Overtaking situation with cost function alteration (M2). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

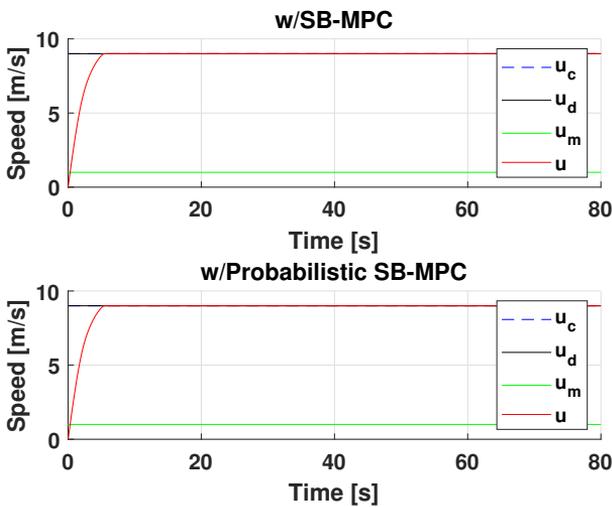


Figure 7.18: Overtaking situation with cost function alteration (M2). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

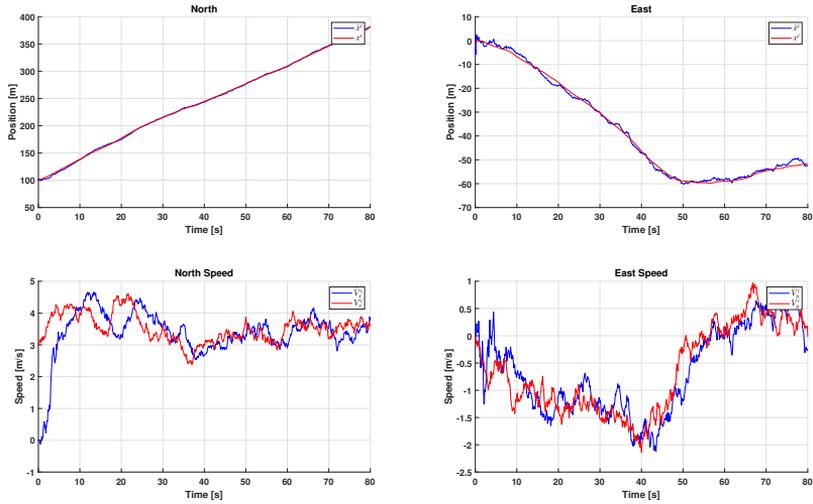


Figure 7.19: Overtaking situation with cost function alteration (M2). Track estimates for the obstacle, versus the true motion.

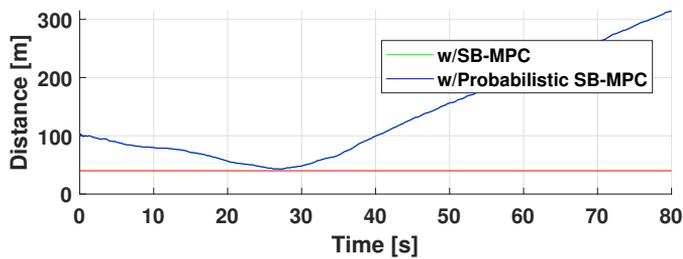


Figure 7.20: Overtaking situation with cost function alteration (M2). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

Head-On

Figures 7.21 - 7.25 show the head-on simulation results. Here, the COLAV methods behave correctly according to COLREGS Rule 8 and 14, taking early action to starboard in order to avoid collision. Similarly to the overtaking situation with the alteration (M2), the collision probability will be near or equal to 1 at some point before CPA, and thus the max value of the modified term in the cost function becomes approximately equal to that in the original cost function. This makes the PSBMPC and SBMPC performance similar.

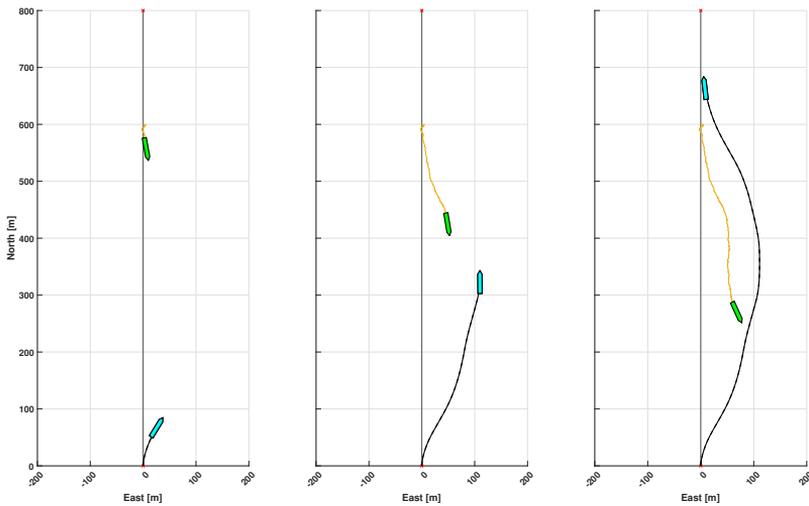


Figure 7.21: Head-on situation with cost function alteration (M2). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

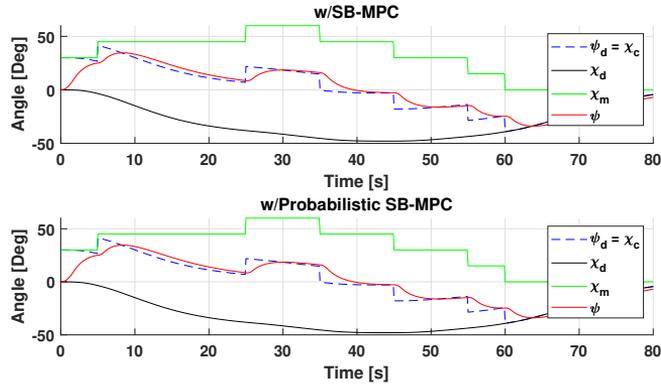


Figure 7.22: Head-on situation with cost function alteration (M2). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

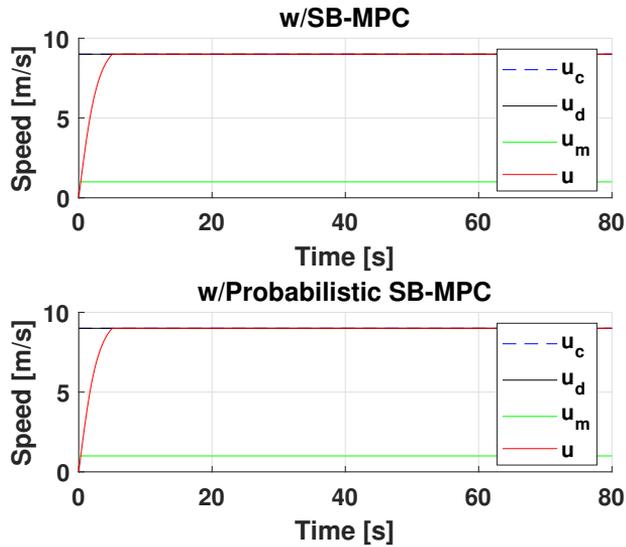


Figure 7.23: Head-on situation with cost function alteration (M2). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

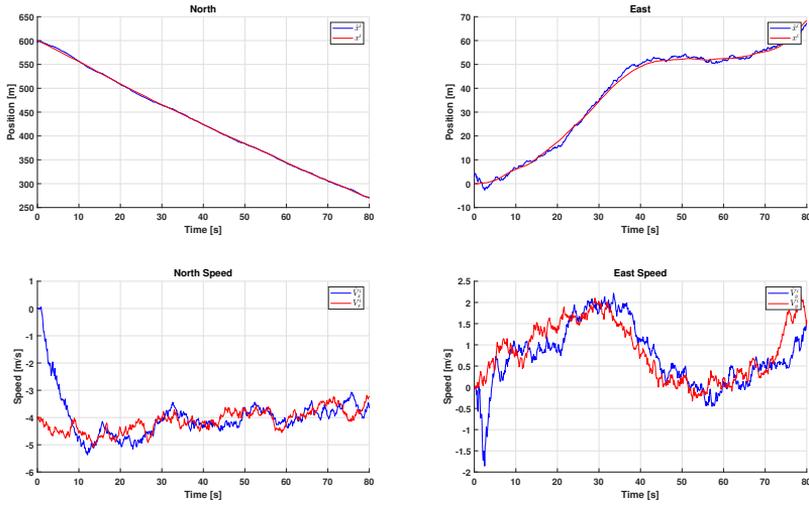


Figure 7.24: Head-on situation with cost function alteration (M2). Track estimates for the obstacle, versus the true motion.

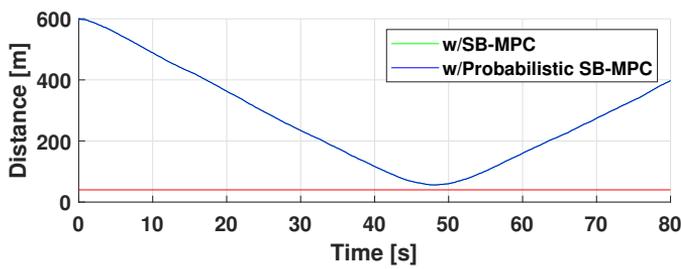


Figure 7.25: Head-on situation with cost function alteration (M2). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

Crossing

Figures 7.26 - 7.30 show the crossing simulation results. Again the behaviour with the two SBMPC versions are identical, due to the same argumentation as for the two previous situations. Moreover, both have trouble with the crossing situations, due to the obstacle not following COLREGS, and due to a non-optimal parameter configuration. Again, a decrease in the course offset penalty parameters $K_{\chi, \text{starboard}}$ and $K_{\chi, \text{port}}$ could improve on the performance here. However, the north-east direction the obstacle takes with time makes the hazard of a starboard maneuver higher, and thus the own-ship turns over to port to avoid collision with certainty.

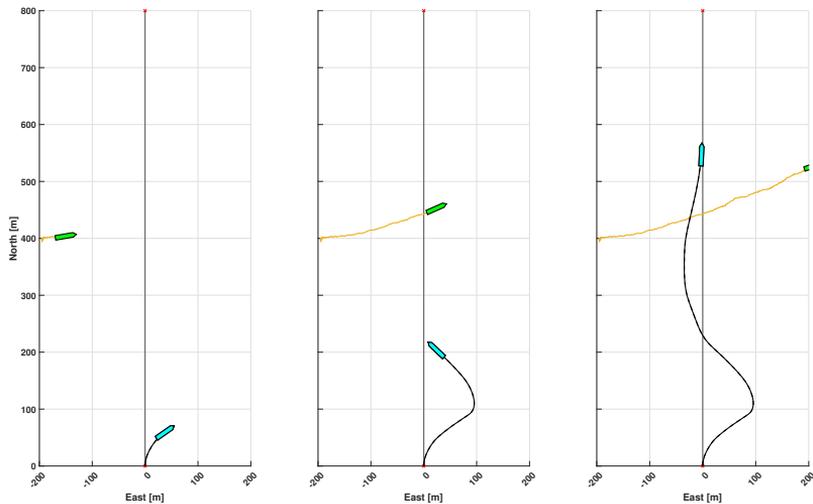


Figure 7.26: Crossing situation with cost function alteration (M2). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

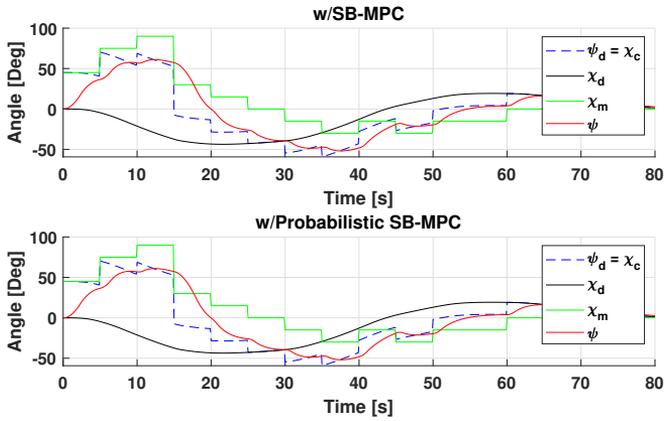


Figure 7.27: Crossing situation with cost function alteration (M2). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

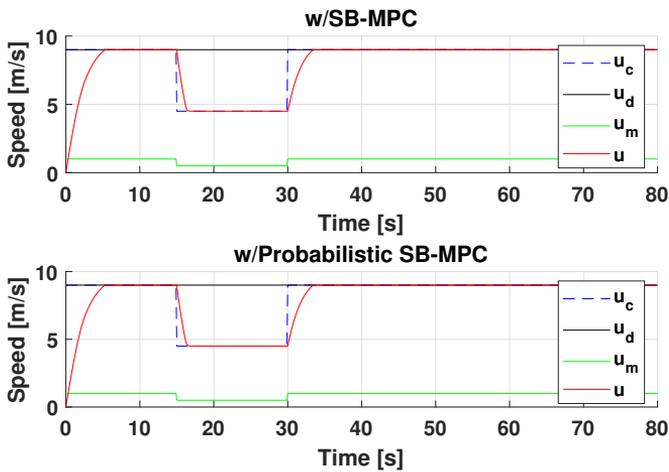


Figure 7.28: Crossing situation with cost function alteration (M2). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

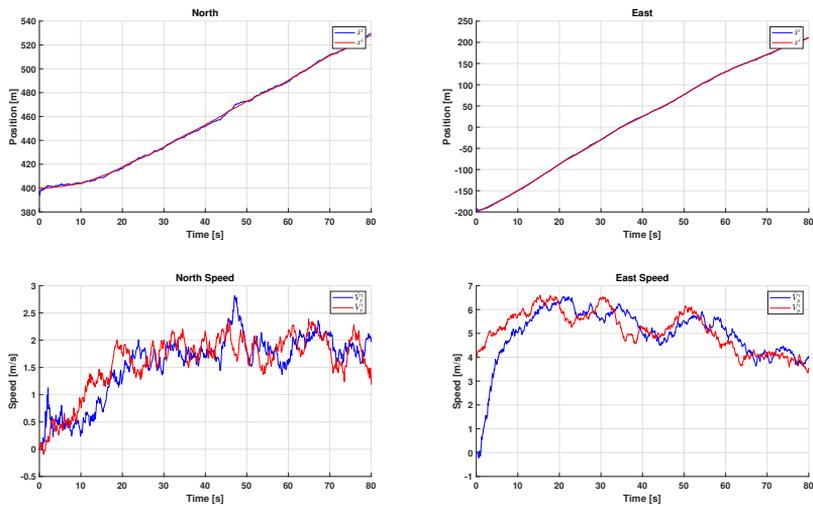


Figure 7.29: Crossing situation with cost function alteration (M2). Track estimates for the obstacle, versus the true motion.

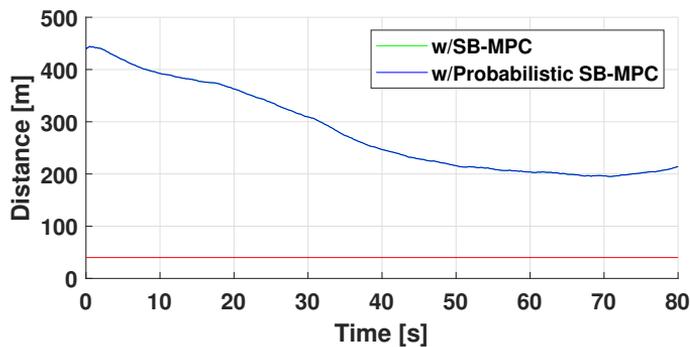


Figure 7.30: Crossing situation with cost function alteration (M2). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

7.2.4 Results Part 3

Here, results for the SBMPC and PSBMPC with the cost function alteration (M3) are given for the three situations.

Overtaking

Figures 7.31 - 7.35 show the overtaking simulation results. The SBMPC again performs correctly, obeying to the COLREGS Rule 13 being the overtaking vessel and also converging smoothly to the planned path. However, the PSBMPC just drives directly ahead and almost collides with the obstacle, which enters the own-ship safety zone around $t = 16$ s. This comes from the use of the exponential term $e^{-(t-t_0)}$ in (M3), which values collision cost longer forward in time lower than the cost closer to the current time. A parameter $0 < b < 1$ should here in hindsight be set in front of the exponent in the exponential ($e^{-b(t-t_0)}$) in order to adjust the discounting of future collision costs. As seen in some of the previous situations, some oscillations in the heading reference for the own-ship with PSBMPC occurs, mostly because of too high values of the penalty parameters $K_{\chi, \text{starboard}}$ and $K_{\chi, \text{port}}$.

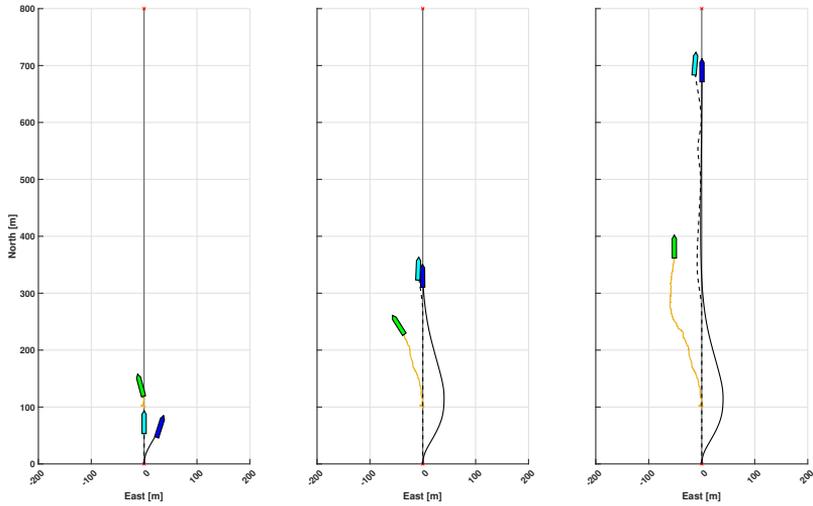


Figure 7.31: Overtaking situation with cost function alteration (M3). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path with the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

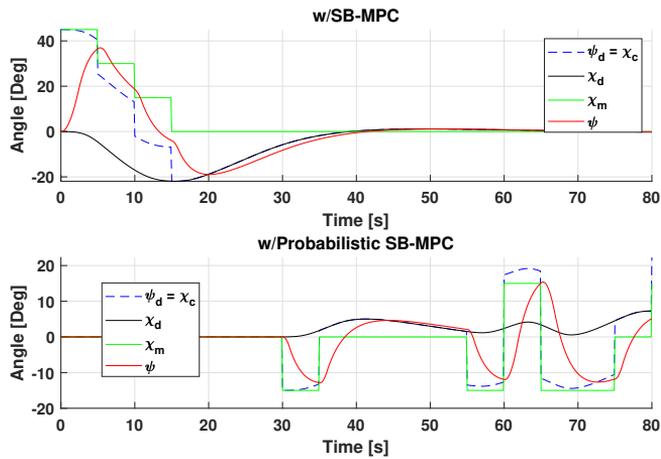


Figure 7.32: Overtaking situation with cost function alteration (M3). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

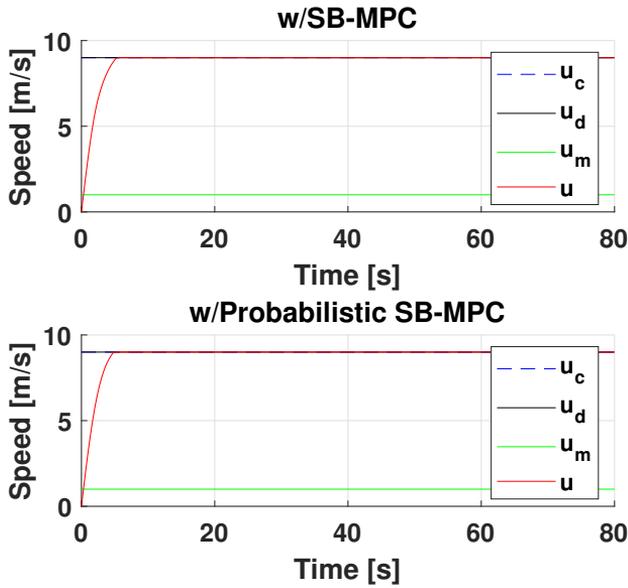


Figure 7.33: Overtaking situation with cost function alteration (M3). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

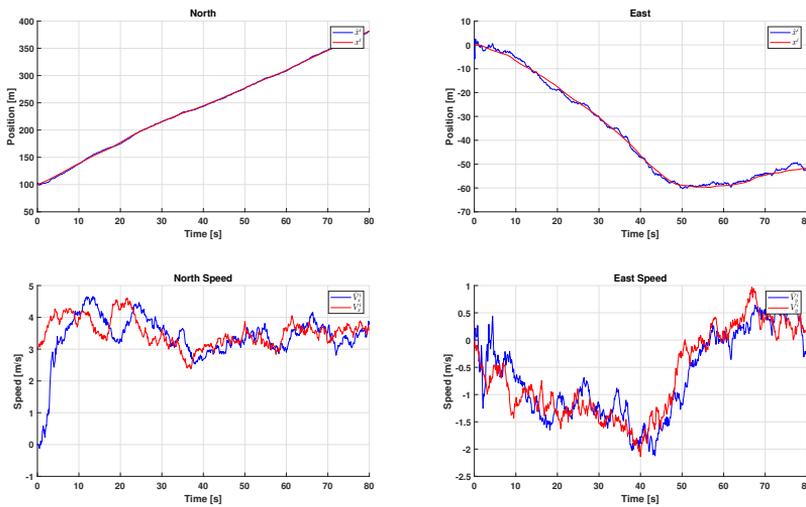


Figure 7.34: Overtaking situation with cost function alteration (M3). Track estimates for the obstacle, versus the true motion.

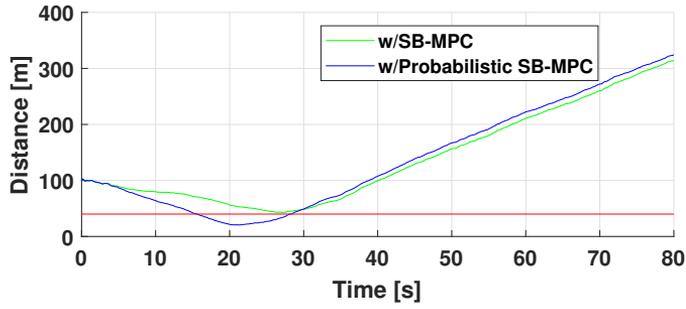


Figure 7.35: Overtaking situation with cost function alteration (M3). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

Head-On

Figures 7.36 - 7.40 show the head-on simulation results. The SBMPC here performs correctly according to the COLREGS Rule 8 and 14. The PSBMPC acts more poorly, turning to port first and then to starboard in order to avoid the obstacle. However, as the COLREGS violation is only detected when the obstacle is within $d_{close} = 200$ m in the SBMPC, the PSBMPC acts correctly from the point in time this threshold is crossed.

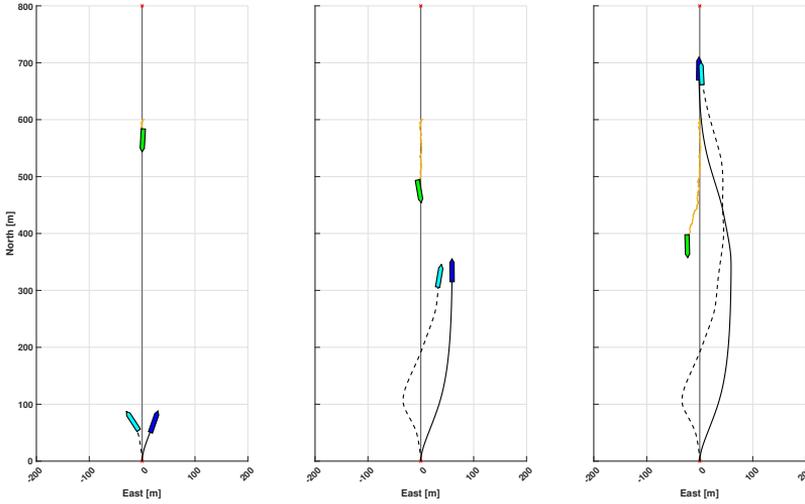


Figure 7.36: Head-on situation with cost function alteration (M3). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path when the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

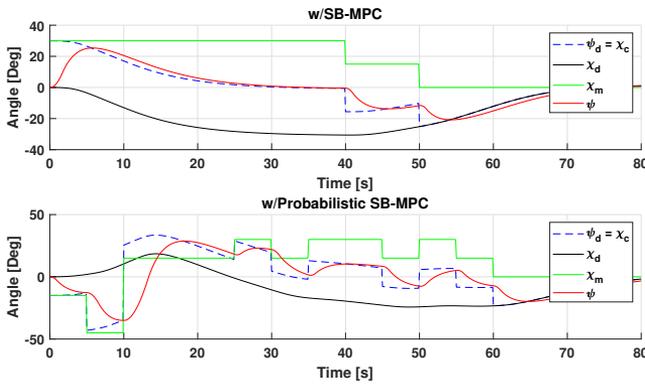


Figure 7.37: Head-on situation with cost function alteration (M3). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

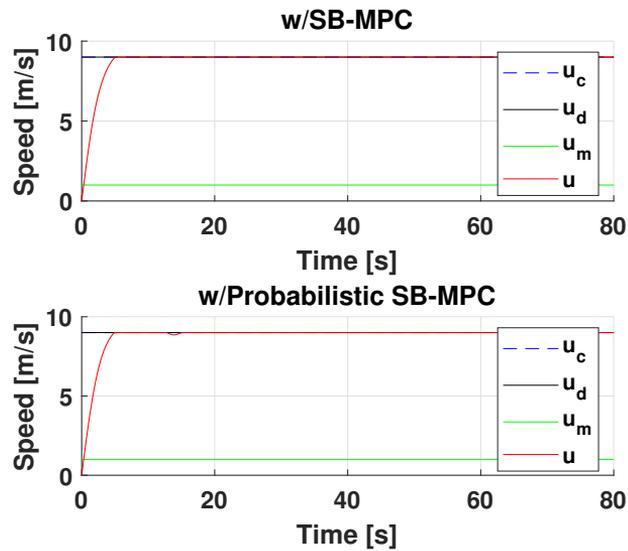


Figure 7.38: Head-on situation with cost function alteration (M3). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

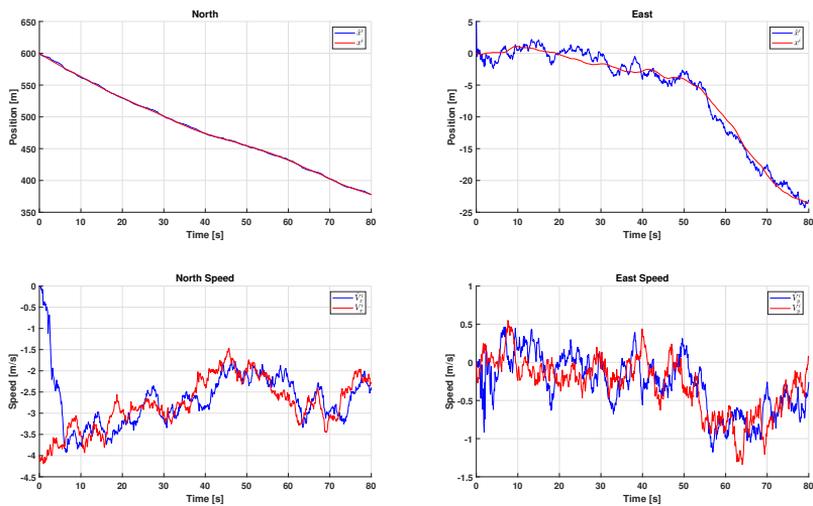


Figure 7.39: Head-on situation with cost function alteration (M3). Track estimates for the obstacle, versus the true motion.

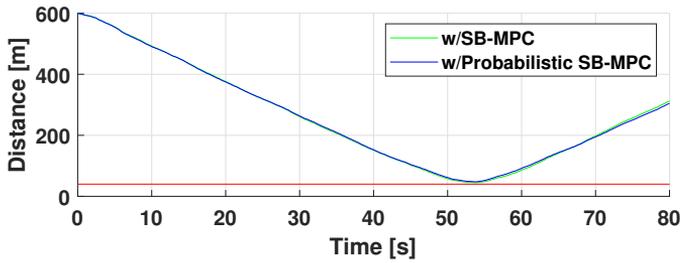


Figure 7.40: Head-on situation with cost function alteration (M3). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

Crossing

Figures 7.41 - 7.45 show the crossing simulation results. The PSBMPC here takes a more risky approach where it drives straight forward for a long time before taking a starboard turn in order to avoid collision, still keeping the obstacle at safe distance. The SBMPC takes a more cautious approach with an early evasive maneuver. The oscillations in the heading for the PSBMPC may be a result of the KF velocity estimate variation affecting the collision probability, which again affects the COLAV system behaviour. The variance in the collision probability evaluation with only $n_{mc,int} = 100$ samples may also affect this, in addition to non-optimal tuning of the COLAV method, as was mentioned for some of the previous situations.

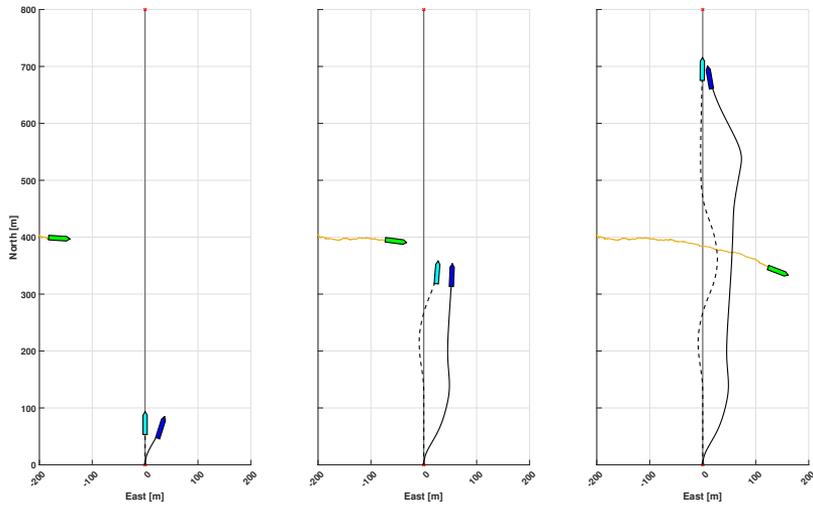


Figure 7.41: Crossing situation with cost function alteration (M3). North-East plot of the vessel motions at $t = 10$ s, $t = 40$ s and $t = 80$ s. The obstacle is plotted in green, with its KF tracked path shown in yellow/gold. The own-ship is plotted in blue with a black path when the original SBMPC. For the PSBMPC, the own-ship is plotted in cyan with a dashed black path. The grey line with two red crosses marks the planned path for the own-ship.

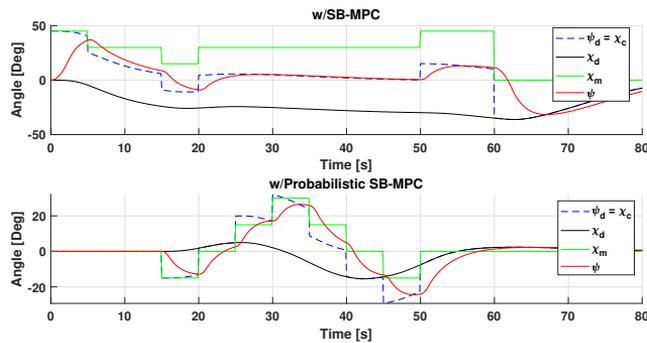


Figure 7.42: Crossing situation with cost function alteration (M3). Heading response for the two SBMPC versions. The nominal course reference χ_d , course offset χ_m , actual course (equal to heading) reference $\chi_c = \psi_d$ and own-ship heading ψ are here shown.

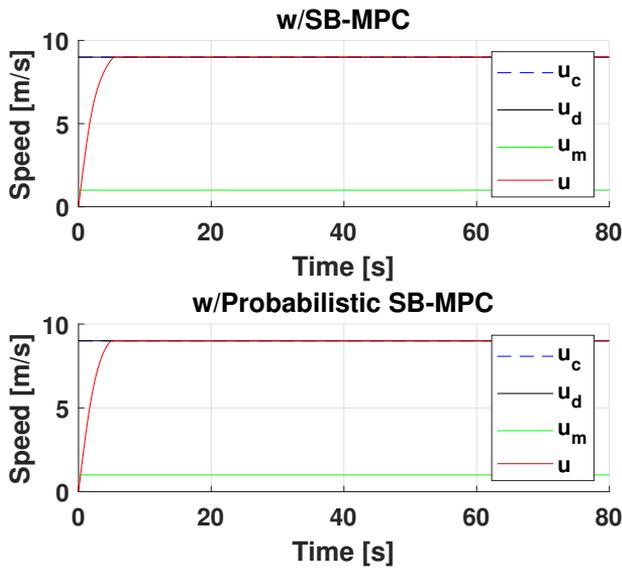


Figure 7.43: Crossing situation with cost function alteration (M3). Surge response for the two SBMPC versions. The nominal surge reference u_d , the surge (propulsion) offset u_m , the actual surge reference u_c and own-ship surge u are here shown.

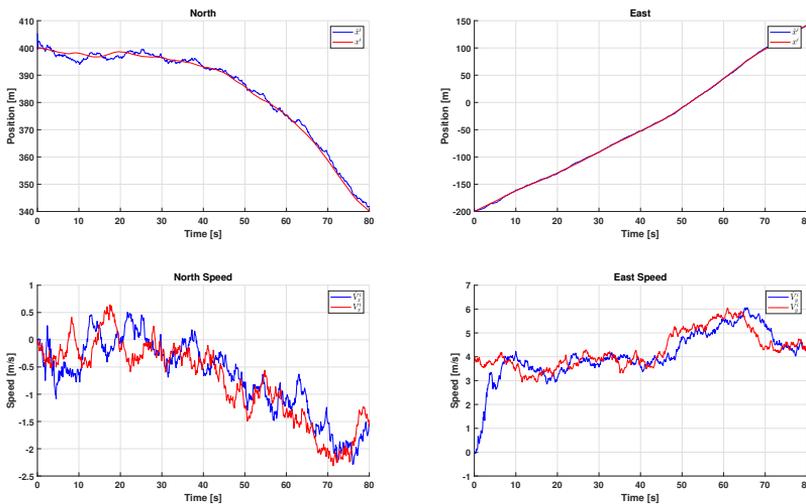


Figure 7.44: Crossing situation with cost function alteration (M3). Track estimates for the obstacle, versus the true motion.

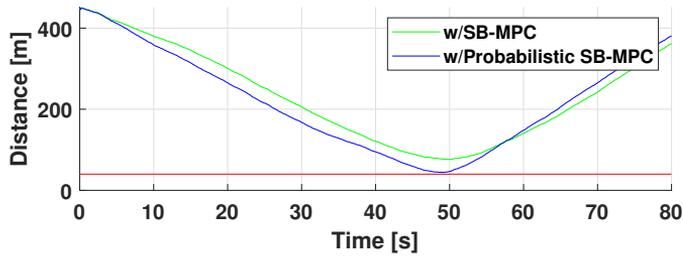


Figure 7.45: Crossing situation with cost function alteration (M3). Distance from the own-ship to the obstacle, for both SBMPC versions. The red line marks the safe distance $d_{safe} = d_i^{safe} = 40$ m.

7.3 Discussion

Note that the results obtained here can not be said to be representative for the average case, as only single realizations of the stochastic process describing the obstacle motion were sampled. The number of Monte Carlo simulations N_{MC} should be increased in order to get a more robust result. Moreover, different values of the noise strength σ_a in the CVM should be tested, in order to test the probabilistic COLAV performance for obstacles with varying maneuverability. Tests in a larger variety of situations, with multiple obstacles in congested areas, for instance near land in Trondheimsfjorden, should also be performed. However, the results gained here can serve as indicators for the probabilistic COLAV performance with the chosen modifications.

The Probabilistic SBMPC, which involved three suggested modifications to the SBMPC cost function, was tested against the original version for three situations. Results using the first cost function modification (M1) gave overly conservative behavior in all situations for the PSBMPC, due to the risk of collision term being omitted. Thus, it shows that a term accounting for the safety zone formalism, i.e. when an obstacle is inside the safety zone or not, and a weighting of the time when a collision might occur, need to be included in the cost function in order to obtain good performance.

For the second cost function modification (M2), the results with both SBMPC version were identical, and thus no added benefit with scaling the collision cost and risk term in the cost function (7.1) by the collision probability was gained here. This was due to the max value of the modified collision cost and risk term becoming approximately equal to the max value of the original collision cost

and risk term.

The third cost function modification (M3) gave a more risk-taking COLAV system, because possible collisions occurring later in the prediction horizon are discounted through the use of the exponential term $e^{-(t-t_0)}$, and thus given lower cost. In addition to the scaling by the collision probability, this allows the PSBMPC to make more daring maneuvers. A factor b should here have been used in front of the exponent in the exponential in order to tune the discounting of future possible collisions. The choice of b can adjust the PSBMPC behavior between being conservative and risk-averse, to being more risk-taking.

In total, the optimal modification to the SBMPC to account for uncertainty or collision probability with obstacles can not be said to be found yet. More modifications should here be done to the SBMPC and tested in multiple scenarios. The results with the three modifications (M1) - (M3) can however serve as a first step towards developing a better PSBMPC version.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this thesis, a probabilistic version of the Collision Avoidance System based on the Scenario-Based Model Predictive Control have been implemented and compared to the original version in three different situations; an overtaking situation, a head-on situation and a crossing situation. The simulation environment with the COLAV systems were implemented in Matlab, with a Kalman Filter used as the tracking method, which produces the obstacle state estimates and the corresponding error covariance.

The probabilistic version PSBMPC utilizes the probability of collision \mathbb{P}_c^i with an obstacle i in its framework. The collision probability was here calculated using Monte Carlo integration with a less effective importance sampling strategy. This strategy samples a number of $n_{mc,int}$ possible trajectories that the obstacle may take based on the current obstacle expected position and velocity and the corresponding uncertainty. Further, it counts the number of these trajectories which cross with the own-ship safety zone at the Closest Point of Approach, at some point in the future, and uses the ratio of crossing trajectories to the total number of trajectories as a collision probability estimate. An attempted more effective method for sampling the obstacle trajectories was also implemented. This was based on the expected velocity the obstacle need to have in order to

collide directly with the own-ship at CPA, and a heuristic estimate of the variance in velocity and position needed to cross the own-ship safety zone. However, work yet remains for this approach to function correctly, and it was thus not used in the PSBMPC testing. A limitation of the current collision probability evaluation method is that only straight line paths are assumed for the future obstacle trajectory. This will in practice be, among other factors, determined by the type of vessel, its goal destination and the driver type "behind the wheel".

Three modifications to the cost function in the SBMPC to account for the collision probability were suggested for the PSBMPC. The first modification gave poor results, because the risk of collision \mathcal{R}_i^k with scenario k , which involves the time of collision and whether obstacle i resides inside the own-ship safety zone, was omitted from the cost. This did however prove the need for having a term in the SBMPC cost function which determines whether or not the obstacle crosses the safety zone, and at what point of time in the horizon this occurs. The second modification gave identical results as for the original SBMPC, because of the max value of the modified- and original term being approximately equal, not causing different optimal course or propulsion offsets. For the third modification, inclusion of an exponential term involving the difference between the prediction time and current time allowed for more risk-taking behavior in the PSBMPC, as future possible collision events were discounted. A discounting factor b should here be introduced, to tune the COLAV method behavior between being overly cautious, and being too risk-taking.

8.2 Future Work

Several suggestions are made for future work, stated in the following.

Development of Efficient and Realistic Collision Probability Evaluation Methods

The first suggestion involves research on developing more efficient methods for evaluating the collision probability between the own-ship and an obstacle, which uses both the uncertainty in both position and velocity. The less effective method used here, based on Monte Carlo integration, uses a computational time per probability evaluation which is not feasible for real-time, as was discussed in Chapter 6. The attempted more effective sampling strategy developed could here

be refined to become a working approach, or a different method for instance based on an extension of the probability flow method described in Chapter 5 could be developed.

Development of Vessel Intent Models

The assumption used in this thesis that an obstacle travels along a straight line with constant velocity will not hold for all vessel types in practice. A possibility is here to use machine learning or a similar technique using AIS data for a large amount of vessels travelling in a region, to infer the maneuvering intent of an obstacle. This maneuvering intent can then be used to change the obstacle trajectory in the SBMPC prediction, and give a possibly more robust collision probability estimate which does not only consider straight line trajectories.

Testing Out More Modifications To The SBMPC

The second suggestion involves testing out more modifications to for instance the SBMPC cost function beyond the proposals made here. A stand-alone term involving the collision probability with possibly other factors such as the time of collision, could be added to the cost function. As a non-zero collision probability estimate higher than 1 percent is a strong hint that an evasive maneuver should be taken, this factor might need a term on its own to signify the importance of avoiding collision.

Testing the PSBMPC With More Complex Multi-Target Tracking Methods

The single target tracking method Kalman Filter used in this thesis for tracking an obstacle is simplistic and does not account for whether the target (obstacle) exists, nor data association uncertainty (when using for instance radar measurements) when multiple targets are present. Testing with a multi-target tracking method in more congested situations near land is therefore advised and will aid in proving the performance of the modified SBMPC.

References

- [1] Banda, O. A. V., Goerlandt, F., Montewka, J., and Kujala, P., “A risk analysis of winter navigation in finnish sea areas,” *Accident Analysis & Prevention*, vol. 79, pp. 100–116, 2015.
- [2] Bar-Shalom, Y. and Li, X.-R., *Multitarget-multisensor tracking: principles and techniques*. YBs Storrs, CT, 1995, vol. 19.
- [3] Bar-Shalom, Y., Li, X., and Kirubarajan, T., *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [4] Bar-Shalom, Y. and Tse, E., “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
- [5] Blaich, M., Köhler, S., Reuter, J., and Hahn, A., “Probabilistic collision avoidance for vessels,” *IFAC-PapersOnLine*, vol. 48, no. 16, pp. 69–74, 2015.
- [6] Brekke, E. F., *Fundamentals of Sensor Fusion: Target tracking, Navigation and SLAM*. 2019, Lecture notes for future course TTK4250 Sensor Fusion. In writing.
- [7] Chauvin, C., Lardjane, S., Morel, G., Clostermann, J.-P., and Langard, B., “Human and organisational factors in maritime accidents: Analysis of collisions at sea using the hfacs,” *Accident Analysis & Prevention*, vol. 59, pp. 26–37, 2013.

- [8] Colito, J., “Autonomous mission planning and execution for unmanned surface vehicles in compliance with the marine rules of the road,” PhD thesis, University of Washington Washington, 2007.
- [9] Debnath, A. K. and Chin, H. C., “Navigational traffic conflict technique: A proactive approach to quantitative measurement of collision risks in port waters,” *The Journal of Navigation*, vol. 63, no. 1, pp. 137–152, 2010.
- [10] EMSA, “The european maritime safety agency: Annual overview of marine casualties and incidents,” EMSA, Annual Report, Nov. 14, 2018.
- [11] Foss, B. and Heirung, T. A. N., *Merging Optimization and Control*. 2016.
- [12] Fossen, T. I., *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- [13] Goerlandt, F. and Montewka, J., “Maritime transportation risk analysis: Review and analysis in light of some foundational issues,” *Reliability Engineering & System Safety*, vol. 138, pp. 115–134, 2015.
- [14] Hagen, I. B., “Collision avoidance for asvs using model predictive control,” Master’s thesis, NTNU, 2017.
- [15] Hagen, I. B., Kufoalor, D., Brekke, E. F., and Johansen, T. A., “Mpc-based collision avoidance strategy for existing marine vessel guidance systems,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 7618–7623.
- [16] Hart, P. E., Nilsson, N. J., and Raphael, B., “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [17] Henriksen, E. S., “Automatic testing of maritime collision avoidance methods with sensor fusion,” Master’s thesis, NTNU, 2018.
- [18] IMO, “Colregs - international regulations for preventing collisions at sea,” *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, 2012.

- [19] Johansen, T. A., Perez, T., and Cristofaro, A., "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, 2016.
- [20] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [21] Kuwata, Y., Wolf, M. T., Zarzhitsky, D., and Huntsberger, T. L., "Safe maritime autonomous navigation with colregs, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2014.
- [22] Kwik, K., "Calculation of ship collision avoidance manoeuvres: A simplified approach," *Ocean engineering*, vol. 16, no. 5-6, pp. 475–491, 1989.
- [23] Lauderdale, T., "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.
- [24] Lee, S.-M., Kwon, K.-Y., and Joh, J., "A fuzzy logic for autonomous navigation of marine vehicles satisfying colreg guidelines," *International Journal of Control, Automation, and Systems*, vol. 2, no. 2, pp. 171–181, 2004.
- [25] Li, S., Meng, Q., and Qu, X., "An overview of maritime waterway quantitative risk assessment models," *Risk Analysis: An International Journal*, vol. 32, no. 3, pp. 496–512, 2012.
- [26] Macrae, C., "Human factors at sea: Common patterns of error in groundings and collisions," *Maritime Policy & Management*, vol. 36, no. 1, pp. 21–38, 2009.
- [27] Musicki, D. and Evans, R., "Joint integrated probabilistic data association: Jipda," *IEEE transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 1093–1099, 2004.
- [28] Musicki, D., Evans, R., and Stankovic, S., "Integrated probabilistic data association," *IEEE Transactions on automatic control*, vol. 39, no. 6, pp. 1237–1241, 1994.

- [29] NTNU. (May 29, 2019). Research vessel r/v gunnerus, [Online]. Available: <https://www.ntnu.edu/oceans/gunnerus>.
- [30] Paielli, R. A. and Erzberger, H., "Conflict probability estimation for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.
- [31] —, "Conflict probability estimation generalized to non-level flight," *Air Traffic Control Quarterly*, vol. 7, no. 3, pp. 195–222, 1999.
- [32] Park, J. and Kim, J., "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.
- [33] Patera, R. P., "Space vehicle conflict-avoidance analysis," *Journal of guidance, control, and dynamics*, vol. 30, no. 2, pp. 492–498, 2007.
- [34] Rawlings, J. B. and Mayne, D. Q., *Model predictive control: Theory and design*. Nob Hill Pub. Madison, Wisconsin, 2009.
- [35] Reid, D., "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [36] Rolls-Royce. (2016). Autonomous ships - the next step, [Online]. Available: <https://www.rolls-royce.com/~media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/rr-ship-intel-aawa-8pg.pdf>.
- [37] Schiaretti, M., Chen, L., and Negenborn, R. R., "Survey on autonomous surface vessels: Part 1-a new detailed definition of autonomy levels," in *International Conference on Computational Logistics*, Springer, 2017, pp. 219–233.
- [38] Shah, B. C., Švec, P., Bertaska, I. R., Sinisterra, A. J., Klinger, W., Ellenrieder, K. von, Dhanak, M., and Gupta, S. K., "Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic," *Autonomous Robots*, vol. 40, no. 7, pp. 1139–1163, 2016.

- [39] Soares, C. G. and Teixeira, A., "Risk assessment in maritime transportation," *Reliability Engineering & System Safety*, vol. 74, no. 3, pp. 299–309, 2001.
- [40] Statheros, T., Howells, G., and Maier, K. M., "Autonomous ship collision avoidance navigation concepts, technologies and techniques," *The Journal of Navigation*, vol. 61, no. 1, pp. 129–142, 2008.
- [41] Stormont, D. P., "Analyzing human trust of autonomous systems in hazardous environments," in *Proc. of the Human Implications of Human-Robot Interaction workshop at AAAI*, 2008, pp. 27–32.
- [42] Tam, C., Bucknall, R., and Greig, A., "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.
- [43] Wilthil, E. F., Flåten, A. L., and Brekke, E. F., "A target tracking system for asv collision avoidance based on the pdaf," in *Sensing and Control for Autonomous Vehicles*, Springer, 2017, pp. 269–288.

