NTNU
Norwegian University of
Science and Technology

Automatic testing of maritime collision avoidance methods with sensor fusion

Eivind Sørum Henriksen

Master of Science in Cybernetics and Robotics
Submission date: June 2018
Supervisor: Edmund Førland Brekke, ITK
Co-supervisor: Tom Arne Pedersen, DNV GL

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Autonomous ships and autonomous surface vehicles in general need automatic collision avoidance (COLAV) for safe navigation. Such methods can use a variety of information sources, including communication-based solutions such as AIS and exterospective sensors such as radar and camera. The methods must be able to resolve situations of varying complexity and often with limitations both in available information and in maneuver possibilities.

This project will build on previous work on automatic testing of COLAV methods using DNV GL's CyberSea simulator. This work includes implementation and comparison of the velocity obstacle (VO) and scenario-based model-predictive control (SBMPC) methods for COLAV in this framework, and closing the loop with simulation of radar tracking in this framework. In this project, the candidate will further enhance the testing by studying more challenging tasks and circumstances. Relevant problems include:

1. Multi-target tracking with track initialization. False tracks and misdetections inevitably occur in a target tracking system, and this will affect the performance of the COLAV method. Multi-target tracking methods are needed whenever target ships move so close to each other that their tracks risk being confused.

2. Dealing with land. Constraints posed by the sea shore may severely limit the possible actions that the COLAV methods can choose between. Methods such as VO may need additional tweaks to function adequately at all when land is present. Also, areas near sea shore can often be troubling sources of false tracks in the tracking system.

3. Several subsequent COLREGS situations. Previous work on COLAV testing in CyberSea has only considered situations with up to 3 vessels approaching the same point through geometric iterative testing. While additional complexity due to more vessels approaching each other simultaneously is unlikely, situations where a double-digit number of vessels interact in rapid succession are common in the real world.

4. COLREGS violation and rogue behaviour in general. It is important that the COLAV methods are able to make sensible judgments when a ship for some reason violates COLREGS, or suddenly does something unexpected. Smaller boats with high maneuverability can be difficult to track, especially near the sea shore.

The goal of this project will be to study how at least two COLAV methods cope with at least two, and possibly more, of these challenges.

Preface

This thesis is the concluding part of the author's Masters degree in Cybernetics and Robotics, at the Norwegian University of Science and Technology (NTNU). I would like to thank my supervisors Edmund F. Brekke at NTNU and Tom Arne Pedersen at DNV GL, for their guidance, help and support during the work of this thesis.

This thesis is a continuation of the MSc thesis of Paal Kristian Minne (see section 1.3) and a preliminary project by the author in the fall of 2017 [31]. The simulator environment, where all the testing and implementations have been performed, is called CyberSea and has been provided by DNV GL (see section 5.1). The implemented tracking system is based on a MATLAB implementation of the JIPDAF, provided by Edmund Brekke. This MATLAB implementation is unpublished and have not been tested, optimized nor adapted to work in systems like the CyberSea. For the theoretic description of the tracking algorithms, the author of this thesis was provided with teaching-material drafts for use in a future course in sensor fusion. This compendium is referenced to as [10], and can be provided by contacting Edmund Brekke.

A special thanks goes to the guys at room G-234, for providing an open-minded, satisfying and highly appreciated discussion environment, which often resulted in the "whole-helmet" being the preferable choice compared to the "karate-suit".

Eivind Sørum Henriksen
Trondheim, June, 2018

Abstract

The development of Autonomous Surface Vessels (ASVs) has during the recent years seen a great progress. When being completely developed, these vessels may be used in a variety of scientific and commercial operations, eventually leading human based operations redundant.

An ASV needs a well-functioning Collision Avoidance (COLAV) system in order to operate at sea where other obstacles such as vessels and land are present. In addition of being able to handle collision situations, the COLAV system needs to comply with the rules for avoiding collision at sea (COLREGS). To retrieve the necessary information of the surroundings, the COLAV system utilizes a number of information sources. This may include exteroceptive sensors such as radar and cameras, or communication-based solutions such as the automatic identification system (AIS). In order to enable COLAV, the sensor information needs to be included into the state estimation of the surrounding obstacles. This is done by the use of a tracking system.

In this thesis, a COLAV system including a multi-target tracking system based on the Joint Integrated Probabilistic Data Association Filter (JIPDAF) and two COLAV algorithms, one based on the Velocity Obstacle (VO) and another method called Scenario-Based Model Predictive Control (SBMPC), has been tested in a wide variety of ASV scenarios. The scenarios are generated with a new method which challenges the COLAV system to a high number of succeeding vessel interactions. To evaluate the COLAV system's scenario performance, a number of evaluation metrics used to determine COLREGS compliance are applied.

The COLAV system have been implemented on a Platform Supply Vessel (PSV) and the testing has taken place in a simulated environment. The results from the testing show that when exposed to a small amount of noise, the tracking system delivers accurate obstacle estimates to the COLAV algorithm, resulting in good evaluation scores. In more challenging scenarios which includes considerably more noise, the tracking system delivers more fluctuating obstacle estimates and a high number of false tracks, resulting in poor performance of the SBMPC algorithm, while the VO algorithm performs remarkably well.

Sammendrag

Utviklingen av autonome overflateskip (ASV) har de siste årene sett stor fremgang. Når det en gang i fremtiden vil eksistere et utall variasjoner av ASVer til bruk for både forskningsmessige og kommersielle operasjoner, vil man til slutt se at menneskelig-baserte operasjoner blir overflødige.

En ASV trenger et velfungerende system for kollisjonsunngåelse (COLAV) for å kunne operere til sjøs der det finnes andre hindringer som f.eks. fartøyer og land. I tillegg til å kunne håndtere kollisjonssituasjoner, må COLAV-systemet overholde reglene for å unngå kollisjon til sjøs (COLREGS). For å innhente den nødvendige informasjonen om omgivelsene, bruker COLAV-systemet en rekke informasjonskilder. Dette kan omfatte eksteroceptive sensorer som radar og kameraer, eller kommunikasjonsbaserte løsninger som det automatiske identifikasjonssystemet (AIS). For å muliggjøre COLAV må sensorinformasjonen inngå i estimering av de omkringliggende fartøyers posisjon og hastighet. Dette gjøres ved bruk av et målfølgingssystem.

I denne oppgaven er et COLAV-system som inkluderer et multimålfølgingssystem basert på målfølgingsmetoden Joint Integrated Probabilistic Data Association Filter (JIPDAF) og to COLAV-algoritmer, en basert på hastighetshindringer (VO) og en annen metode som kalles Scenario-Based ModelPredictive Control (SBMPC), testet i en rekke ASV-scenarier. Scenarioene er generert med en ny metode som utfordrer COLAV-systemet til et høyt antall av etterfølgende skipsinteraksjoner. For å evaluere COLAV-systemets scenario-utførelse, brukes en rekke evalueringsmål for å bestemme i hvor stor grad reglene til sjøs (COLREGS) har blitt overholdt.

COLAV-systemet er implementert på et plattformforsyningsfartøy (PSV) og testingen har skjedd i et simulert miljø. Resultatet fra testingen viser at når COLAV-systemet blir utsatt for en liten mengde støy, gir målfølgingssystemet nøyaktige fartøyestimater til COLAV-algoritmen, som resulterer i gode evalueringsskår. I mer utfordrende scenarier som inneholder betydelig mer støy, leverer målfølgingssystemet mer fluktuerende fartøyestimater og et høyt antall falske mål, noe som resulterer i dårlig evalueringsskår for SBMPC-algoritmen, mens VO-algoritmen yter bemerkelsesverdig godt.

Table of Contents

Abbreviations

AI	Artificial Intelligence.
AIS	Automatic Identification System.
ASCs	Autonomous Surface Crafts.
ASV	Autonomous Surface Vessel.
CDF	Cumulative Distribution Function.
COLAV	Collision Avoidance.
COLREGS	The International Regulations for Preventing Collision at Sea.
CPA	Closest Point of Approach.
DOF	Degress of Freedom.
DW	Dynamic Window.
GNC	Guidance, Navigation and Control.
GNSS	Global Navigation Systems.
IGT	Iterative Geometric Testing.
ILOS	Integral Line of Sight.
IMO	International Maritime Organization.
IPDAF	Integrated Probabilistic Data Association Filter.
JIPDAF	Joint Integrated Probabilistic Data Association Filter.

JPDAF	Joint Probabilistic Data Association Filter.
LIDAR	Light Detection And Ranging.
LOS	Line of Sight.
MHT	Multi Hypothesis Tracker.
MIMO	Multi-Input Multi-Output.
MPC	Model Predictive Control.
NED	North-East-Down.
NIS	Normalized Innovation Squared.
PDAF	Probabilistic Data Association Filter.
PMM	Planar Motion Mechanism.
PSV	Platform Supply Vessel.
SBMPC	Scenario-Based Model Predictive Control.
SOI	Spread of Innovations.
SUT	System Under Test.
UAVs	Underwater Autonomous Vehicles.
USVs	Unmanned Surface Vessels.
VO	Velocity Obstacle.

Chapter 1

Introduction

1.1 Motivation and Background

The technological advance the world has seen the last century has opened the possibilities for a much higher level and wider range of automation. With the benefits of cost reduction and general efficiency enhancements, the concept of automation has infiltrated and influenced more or less the entire modern industrial society. Ranging from the clothes and food making industries to more complex production chains like oil refining and aircraft production, they all have taken part in this revolutionary progress. Based on the positive and much wanted aftereffects, it seems natural that automation will continue its journey to affect and improve even more areas.

In the latest years, more intelligent, complex systems called *autonomous* systems have started to appear. Instead of being a specialist in one single task (as an automated system), the autonomous system expands its field of responsibility - it is able to complete and execute more general defined problems with the use of intelligent reasoning. Everyday applications such as cutting the grass, cooking, cleaning etc. may now be performed by autonomous robots. More safety critical systems like autonomous cars has yet to be fully allowed, but the testing has yield promising results [2, 9].

Where the progress of automation for land-based transportation has been evident, the development in the maritime sector has also seen changes. A number of companies have started experimentation and development of autonomous vessels [43, 15, 49]. Together, these are to be used in both cargo and human transportation - a step in the direction of a fully autonomous sea environment.

As the maritime traffic density is estimated to increase in the future, the need for more efficient sea transportation is evident. When, in addition, there has been an increase in serious and very serious accidents at sea [14], the possibilities for Autonomous Surface Vessel (ASV)s clearly is present. In order to be commercialized, ASVs will need to prove to be as safe or safer than a humanly operated vessel [49] - a challenging [56], but reasonable requirement that has yet to be fulfilled.

One of the most challenging aspects when it comes to the development of ASVs, is to make a system that is able to not collide with any surrounding obstacles during travel. This is called Collision Avoidance (COLAV), and despite being investigated since the Second World War [57], still remains a challenge.

1.2 Previous work

Maritime COLAV has been studied and discussed in a variety of ways by numerous people. Tam et al. [57] gives a historical perspective of COLAV development and path planning, especially focusing on close range encounters. It includes a walk-through of previously published papers and tries to identify the better approaches and issues in close range marine navigation. Another historical review is given by Manley [42]. He presents the enabling technologies that have made Unmanned Surface Vessels (USVs) emerge to such a useful platform for marine operations, and gives a prediction of the future outlook of USVs.

Statheros et al. [54] follows along the same lines as [57], and presents a spherical understanding of autonomous ship navigation for collision avoidance together with some theoretical COLAV background. Both classical methods based on mathematical models and algorithms, and more soft-computing techniques based on Artificial Intelligence (AI) are discussed. The paper also explains the link between the development of COLAV algorithms and the human cognitive abilities.

A great number of direct methods on how to solve the COLAV problem exist. Kuwata et al. [38] presents a collision avoidance algorithm based on what is called the Velocity Obstacle (VO). It is an autonomous motion planning algorithm which calculates safe velocities based on the positions and velocities of other obstacles at a given time. Other COLAV methods rely on the concepts from Model Predictive Control (MPC) in order to solve the COLAV problem [37, 17, 28]. By utilizing a dynamic vessel model, weather information, grounding conditions etc., the Scenario-Based Model Predictive Control (SBMPC) by Johansen et. al [37] simulates future trajectories of all vessels in a situation, which is used to determine compliance with the The International Regulations for Preventing Collision at Sea (COLREGS).

One of the main challenges when developing a COLAV algorithm is the ability to evaluate its performance. A COLAV algorithm should behave as specified by the COLREGS, but

since these rules where composed during a time when one was to use ”common sense” when interpreting the rules, the formulation is somewhat vague and non-deterministic. Sparse research exists on the area of applying COLREGS evaluation to COLAV algorithms. The first extensive work on this topic was done by Kyle Woerner [63], where a set of metrics was developed to be used in the evaluation of COLAV algorithms.

A collision avoidance algorithm relies on information of the surroundings - information often provided by a system called Automatic Identification System (AIS). AIS delivers exact state information of the vessel carrying it to all other vessels in some vicinity which also carries an AIS system, but having such a system is not an international requirement, especially not for smaller vessels [35]. To be able to detect and retrieve state information of all vessels, independent of AIS, other types of sensors are necessary, e.g. radar, Light Detection And Ranging (LIDAR) and cameras. Since all sensor data contains some sort of noise, the data can not be considered to give perfect information. Hence, there is a need for some kind of pre-processing of the sensed data before it is used by the COLAV algorithm.

Target tracking aims to establish tracks based on a time series of measurements from sensor data. False measurements should be filtered out, and even though not being directly available from the sensed data, positions and velocities should be estimated. When having multiple targets in a surveillance region, the tracking problem can be divided into two subproblems: *filtering* and *data association* [6]. Filtering is the process of estimating the targets kinematic state, while data association is the process of determining which measurements originates from which targets and which measurements are false measurements (clutter).

A popular single-target tracking method is Bar-Shalom's Probabilistic Data Association Filter (PDAF) [4]. The method relies on a number of assumptions - two of them being: initialized targets and a Gaussian distribution of the past target state information. The PDAF is extended to include track initiation and track death in the the Integrated Probabilistic Data Association Filter (IPDAF) by Musicki [47]. Both the PDAF and the IPDAF can be used in multi-target tracking applications, but only as multiple single-target filters, resulting in poor performance when targets get in close vicinity of each other [6]. A multi-target extension of the PDAF, the Joint Probabilistic Data Association Filter (JPDAF) is given in [24], where the main difference compared to the PDAF is the calculation of the association probabilities. All the methods above are combined in the Joint Integrated Probabilistic Data Association Filter (JIPDAF) [46] - a multi-target tracking method with target existence integration.

The research regarding COLAV with a tracking system utilizing exteroceptive sensors is limited, but some publications exist. Shuster et al. [51] evaluates a COLAV system utilizing a low-cost radar and a laser range sensor for tracking surrounding vessels with the tracking method JPDAF. A high-speed ASV equipped with a maritime radar, is used to demonstrate the performance of the PDAF tracking method alongside a COLAV method based on the Dynamic Window (DW) approach in [18].

1.3 Objective and scope

This thesis is a continuation of the MSc thesis of Paal Kristian Minne [44]. He developed a framework for enabling automatic testing of collision avoidance algorithms. This was done by extending a vessel simulator (CyberSea) by DNV GL to include a Guidance, Navigation and Control (GNC) system together with a metric system used to evaluate maritime collision scenarios. Two COLAV algorithms were implemented, one based on the velocity obstacle [38] and the other based on a SBMPC algorithm [37]. The evaluation system was based on the work by Kyle Woerner [63].

The main intention of this thesis is to further improve the realistic aspects of the CyberSea simulator, where the final goal is to have a completely emulated COLAV system. Minne did not include a tracking system, thus relying on always getting perfect surrounding state information. In this thesis, a multi-target tracking system with track initialization is implemented. False tracks and misdetections inevitably occur in a target tracking system, and will affect the performance of the COLAV method.

The previous work on COLAV testing in CyberSea has only considered situations with up to 3 vessels approaching the same point at the same time. Further increasing the number of vessels in such situations will not be tested, as this type of scenarios is unlikely in the real world. Instead, a new testing scheme made to provide successive interactions between a larger amount of vessels, is presented. Inspired by real world situations, this is to enhance the testing of the COLAV system.

When being challenged to noisy measurements, and in general, difficult conditions for maritime COLAV, the COLAV algorithms should be able to make sensible judgments in order to follow the rules of the sea (COLREGS) and to avoid collision. With the addition of having smaller boats with high maneuverability on crossing courses, the tracking system should be able to give sufficient state information of the surrounding obstacles.

1.4 Contributions

The main contributions of this thesis are:

- Implementation of a tracking system based on the JIPDAF into an already existing COLAV testing framework. Although being given, the tracking system needed to be completely redefined in order to work in the new environment.
- A completely new testing scheme which enables testing of a COLAV system in a high number of successive obstacle scenarios, is presented.
- Testing of an entire COLAV system in a wide range of scenarios with varying tracking and COLAV conditions , and with the presence of other vessels with different

maneuverability.

In addition to the main contributions listed above, a number of smaller improvements and changes have been carried out:

- To be able to use the evaluation system implemented by Minne in the new testing scheme, the stand-on penalties had to be adjusted in order to not penalize movement towards the desired path. Additional bugs were fixed, and an number of other improvements were performed to make the evaluation metric coincides more with the COLREGS.
- The SBMPC algorithm was initially tuned by Minne, but because of an error in the simulator configuration, the extensive job of retuning the algorithms parameters had to be done. In addition, the algorithm has been modified to include the existence probability of the obstacle vessels.
- To improve the COLREGS compliance of the VO algorithm, the cost function in the algorithm has seen several modifications.
- A new smaller boat with higher maneuverability is implemented to make the testing scheme more realistic.

1.5 Thesis structure

Chapter 2 presents the mathematical concepts used to model and control the ASVs used in this thesis, in addition to a section describing the relevant COLREGS rules. Chapter 3 introduce maritime COLAV, where the focus is given to the COLAV algorithms VO and SBMPC. In chapter 4, the concepts and methods for handling uncertain measurement data are presented, including a number of different tracking methods. The simulation and testing environment is explained in chapter 5, while chapter 6 presents the results and a discussion from the COLAV scenario testing. The last chapter, chapter 7, conclude the work done in this thesis, with an additional section describing a number of possible extensions and problems that should be investigated in future work.

1.6 Terminology

To reduce the confusion when it comes to definitions and the use of specific names and terms, this section provides a description of some of the most used terms and their definitions.

- ”Own-ship” is the ship under consideration, and can be seen as ”our” ship, hence the term own-ship.
- ”Obstacles” or ”targets” represents the same. These are other vessels in some vicinity of the own-ship, which may or may not be visible.
- The terms ”COLAV algorithm” and ”COLAV method” are used to specify a specific collision avoidance method, e.g. SBMPC.
- ”COLAV system” is used to define an entire collision avoidance system. In this thesis, this includes a COLAV algorithm and a tracking system.

Chapter 2

Vessel modeling, navigation and control

This chapter introduces the concepts and methods used to control and model the ASVs used in this thesis. A section presenting the rules of the sea (COLREGS) is also included.

2.1 Guidance, Navigation and Control (GNC)

GNC covers the design of systems controlling unmanned vehicles that are moving under water, on the surface or in space [25]. In this thesis, we will cover all of these subjects, but the main focus will lie on the concepts of guidance and navigation.

Guidance is in this context the concept of finding the desired position, velocity and acceleration of a marine vessel. This process is done continuously by examining the current situation of the surroundings (which may include weather and wave information) and the vessels current state. Advanced optimization methods are often used to do the explicit calculation of the optimal trajectory or path for the ASV to follow. After determining the desired state of the vessel, this state information is handed to the motion control system.

Navigation is in general the science of directing a vehicle by determining its world state (i.e. position, attitude etc.). Global Navigation Systems (GNSS) together with sensors such as cameras, LIDAR and radar are used to determine the state of the surroundings. By combining the sensors with mathematical observers, the predicted state estimate will be improved and give the most accurate description to the guidance system.

For dynamically moving systems like ASVs, the Control part covers the determination of control forces and moments by the actuators such that the desired position and velocity are reached. Constructions of the control algorithm often involves the aspects of feedback and feedforward control.

2.2 Vessel modeling and control

This section will present the modeling scheme and the mathematical necessities used to model and control the marine vessels. The material is heavily based on [44], and the methods and notation are from [25].

2.2.1 Marine vessel modeling

Two coordinate frames are used to describe the matrices and vectors: the North-East-Down (NED) reference frame, dentoed $\{n\}$, and the body-fixed reference frame, denoted $\{b\}$. The body-fixed reference frame is a moving coordinate frame that is fixed to the vessel, while the NED reference frame is a constant coordinate frame fixed to the earth. To give the reader an example of the notation used, the position of the origo of the body-fixed frame $\{b\}$ with respect to the inertial (NED) frame $\{n\}$, expressed in the inertial frame, is given by $\boldsymbol{p}_{b/n}^{n} = [x, y]^T$.

3 Degress of Freedom (DOF) vessel model

The vessel model used in this thesis a 3 DOF model for displacement vessels, originally from [25]. When only interested in the horizontal motion, this model is simple and will give sufficient accuracy for the given application.

The dynamics associated with the motion in heave, roll and pitch are neglected, and it is assumed that the vessel has homogeneous mass distribution with xz-plane symmetry and that the origo of the $\{b\}$ frame is set in the center line of the vessel. The vectorial form of the equations of motion for a marine craft with this 3 DOF model can be written as

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{\varsigma} \\ \boldsymbol{M}\dot{\boldsymbol{\varsigma}} + \boldsymbol{C_{RB}}(\boldsymbol{\varsigma})\boldsymbol{\varsigma} + \boldsymbol{C_A}(\boldsymbol{\varsigma})\boldsymbol{\varsigma} + \boldsymbol{D}(\boldsymbol{\varsigma})\boldsymbol{\varsigma} &= \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \end{aligned} \tag{2.1}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{p}_{b/n}^{n} \\ \psi \end{bmatrix} \tag{2.2}$$

is the position and orientation of the vessel, and $\boldsymbol{p}_{b/n}^{n} = [x, y]^T$. $\boldsymbol{R}(\psi)$ is the transformation matrix from the body-fixed frame $\{b\}$ to the NED frame $\{n\}$, given by the Euler

angle ψ around the z-axis and

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

$\boldsymbol{\varsigma} = [\boldsymbol{v}^b_{b/n}, r]^T = [u, v, r]^T$ is the linear and angular body-fixed velocity vector. The generalized forces acting on the ship from the actuators are denoted, τ, while the environmental forces from wind and waves, are denoted τ_{wind} and τ_{wave}.

The total inertia matrix $\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A$, consisting of the rigid body inertia matrix ($\boldsymbol{M}_{RB}$) and the hydrodynamic added mass matrix ($\boldsymbol{M}_A$), is given by

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \quad \boldsymbol{M}_A = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix} \tag{2.4}$$

where m is the mass of the vessel and I_z is the moment of inertia about the z-axis. The hydrodynamic coefficients (the elements of the added mass matrix $\boldsymbol{M}_A$) together with m and I_z defines the behaviour and maneuverability of the vessel.

The rigid body and added mass coriolis-centripetal matrices, are given by

$$\boldsymbol{C}_{RB}(\boldsymbol{\varsigma}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix}, \tag{2.5}$$

$$\boldsymbol{C}_A(\boldsymbol{\varsigma}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v + Y_{\dot{r}} r \\ 0 & 0 & -X_{\dot{u}} u \\ -Y_{\dot{v}} v - Y_{\dot{r}} r & X_{\dot{u}} u & 0 \end{bmatrix} \tag{2.6}$$

where x_g is the x-component of the vector from the center of gravity of the vessel to the origin of the $\{b\}$ frame, $\vec{r}_g = [x_g, y_g, z_g]^T$.

The damping matrix $\boldsymbol{D}(\boldsymbol{\varsigma})$ usually consists of both non-linear and linear damping, but we will approximate the damping by only including the linear damping. By not including the non-linear damping, the complexity of the controller design is reduced and the directional stability is enhanced [44]. The damping matrix is given by

$$\boldsymbol{D}(\boldsymbol{\varsigma}) = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}. \tag{2.7}$$

For a more thorough description of this 3 DOF vessel model, the reader is referred to [25].

2.2.2 Low level controllers

To control the vessels (in surge, sway and yaw), [44] implemented a feedback linearizing Multi-Input Multi-Output (MIMO) controller, given by

$$\boldsymbol{\tau} = \begin{bmatrix} F_x \\ F_y \\ F_n \end{bmatrix} = \boldsymbol{M}(-\boldsymbol{K}_p\tilde{\boldsymbol{\varsigma}}) + \boldsymbol{n}(\boldsymbol{\varsigma}) \tag{2.8}$$

$$\boldsymbol{K}_p = \begin{bmatrix} K_{p,1} & 0 & 0 \\ 0 & K_{p,2} & 0 \\ 0 & 0 & K_{p,3} \end{bmatrix}, \quad K_{p,1}, K_{p,2}, K_{p,3} > 0 \tag{2.9}$$

$$\tilde{\boldsymbol{\varsigma}} = \boldsymbol{\varsigma} - \boldsymbol{\varsigma}_d = \begin{bmatrix} u \\ v \\ r \end{bmatrix} - \begin{bmatrix} u_d \\ v_d \\ r_d \end{bmatrix} \tag{2.10}$$

where $\boldsymbol{M}$ is the inertia matrix from the 3 DOF vessel model and $n(\boldsymbol{\varsigma})$ is the non-linear component

$$\boldsymbol{n}(\boldsymbol{\varsigma}) = \boldsymbol{C}_{RB}(\boldsymbol{\varsigma})\boldsymbol{\varsigma} + \boldsymbol{C}_A(\boldsymbol{\varsigma})\boldsymbol{\varsigma} + \boldsymbol{D}(\boldsymbol{\varsigma})\boldsymbol{\varsigma}. \tag{2.11}$$

The surge is assumed to equal the speed - a valid assumption, especially for large vessels. Thus, u_d is the desired speed given by the guidance system. v_d is set to 0 as the controller is used for path following, and r_d is the desired yaw given by the heading controller

$$r_d = K_{p,\psi}(\psi_d - \psi), \quad K_{p,\psi} > 0. \tag{2.12}$$

Controller gains

Both vessels used in this thesis (see subsection 5.1.3) are controlled by Equation 2.8 and Equation 2.12. The controller gains used for the Platform Supply Vessel (PSV) are

$$\boldsymbol{K}_p^{PSV} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad K_{p,\psi}^{PSV} = 0.08$$

and the controller gains for Gunnerus are

$$\boldsymbol{K}_p^{G} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \quad K_{p,\psi}^{G} = 0.12.$$

2.2.3 ILOS guidance

The Line of Sight (LOS) guidance method is frequently used for path-following. The method steers the vessel towards the path defined by the previous and the current way-

points. If a heading controller is implemented, the desired heading may be given as the angle between the path and the LOS vector (shown as ψ_d in Figure 2.1). This will lead the vessel to track the path given by the waypoints.

Minne [44] describes a LOS guidance law from [25] used for following straight line paths described by a sequence of waypoints, $[\boldsymbol{p}_1^n, \boldsymbol{p}_2^n, ...]^T$. By choosing the straight line between two waypoints ($\boldsymbol{p}_k^n = [x_k, y_k]^T$ and $\boldsymbol{p}_{k+1}^n = [x_{k+1}, y_{k+1}]^T$), the guidance scheme will use the properties shown in Figure 2.1 to steer the vessel towards the straight line.

The angle between the north axis and the path line, called the path-tangential angle α_k, is a positive angle used to rotate the coordinates of the vessel to a path-fixed reference frame

$$\alpha_k = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k). \tag{2.13}$$

The coordinates of the vessel can then be described by

$$\boldsymbol{\epsilon} = \begin{bmatrix} s(t) \\ e(t) \end{bmatrix} = \boldsymbol{R}_p(\alpha_k)^T(\boldsymbol{p}^n - \boldsymbol{p}_k^n) \tag{2.14}$$

where $\boldsymbol{p}^n = [x(t), y(t)]^T$ is the position of the vessel, and the rotation matrix is given by

$$\boldsymbol{R}_p(\alpha_k) = \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix}. \tag{2.15}$$

$s(t)$ is called the along-track distance and $e(t)$ is the cross-track error, both shown in Figure 2.1.

For path-following, the control objective is to make the cross-track error ($e(t)$) tend to zero

$$\lim_{t \to \infty} e(t) = 0. \tag{2.16}$$

One method used to insure both Equation 2.16 and that the vessel steers along the LOS vector, is called Lookahead-based steering. Minne [44] implemented an integral variation of the LOS, called Integral Line of Sight (ILOS), used for situations where there exists unknown disturbances such as ocean currents etc.

The desired course (or heading) is given by

$$\chi_d(e) = \alpha_k + \chi_r(r) \tag{2.17}$$

$$\chi_r(e) = -\text{arctan}\left(\frac{e + \sigma e_{int}}{\Delta}\right), \quad \Delta, \sigma > 0 \tag{2.18}$$

where

$$\dot{e}_{int} = \frac{\Delta e}{(e + \sigma e_{int})^2 + \Delta^2}. \tag{2.19}$$

Δ is the predefined lookahead distance and does the same as the proportional gain in a regular P-controller. The smaller the size of Δ, the more aggressive heading to accomplish $e(t) \to 0$ is given by the controller. A larger Δ will result in a much slower convergence of the track-loss error, but there will be a greater deal of movement in the LOS direction. The radius of the circular arc (in Figure 2.1) is found by the relation $e(t)^2 + \Delta^2 = R_{LOS}^2$.

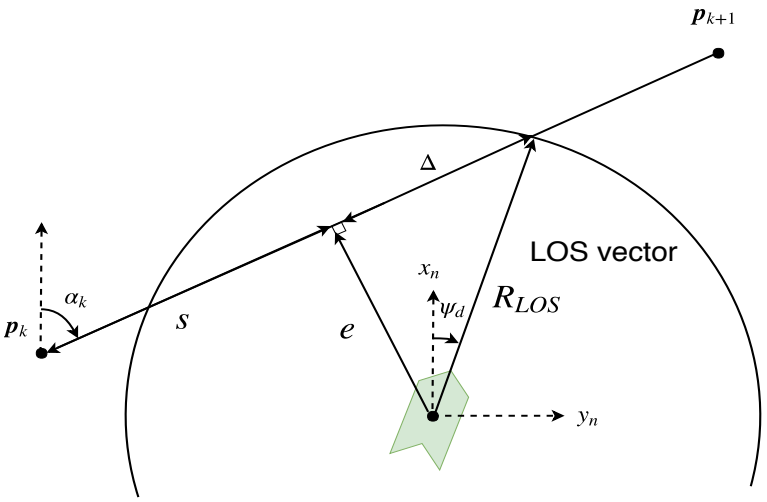


Figure 2.1: Line of sight guidance. Inspired by figure in [25].

ILOS guidance parameters

Both vessels used in this thesis (see subsection 5.1.3) use the same ILOS guidance parameters, given as

$$\Delta = 2000, \qquad \sigma = 2.$$

The parameters for Gunnerus could be tuned differently as it is a much smaller vessel than the PSV, however, the presented parameters resulted in a satisfying behaviour for both vessels.

2.3 International regulations for avoiding collisions at sea (COLREGS)

The current regulations used to determine correct behaviour for vessels at sea is called the COLREGS, officially published by the International Maritime Organization (IMO) in 1972. These rules have a much longer history - dating back to the industrial revolution

where the need for a set of standardized rules to be used in international waters became apparent. From this time until 1972, a set of rules gradually emerged, culminating into the COLREGS. After publishing the rules in 1972, they have been ratified by more than 140 countries - representing more than 97 percent of the world's shipping tonnage [1].

The COLREGS include 38 rules divided into five sections: Part A - General, Part B - Steering and Sailing, Part C - Lights and Shapes, Part D - Sound and Light signals, Part E - Exemptions. In this thesis, the focus has been on Part B - Steering and Sailing, where the rules of how a vessel should behave in collision situations are described. More specific, only the rules 8(b,d), 13, 14, 15, 16 and 17 are used. The next subsection will give the reader a more detailed description of these rules.

2.3.1 Rule description

This subsection provides an overview of the COLREGS rules used in this thesis. The presented material is from [34] and [44].

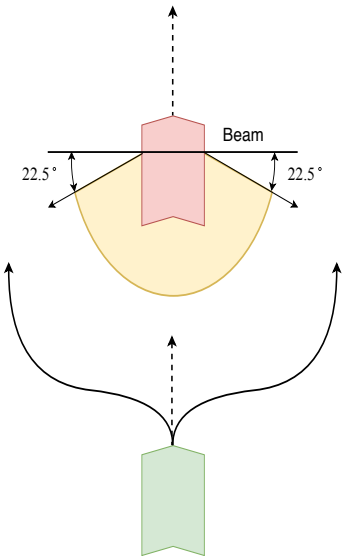


Figure 2.2: COLREGS overtaking situation. The dashed lines represent the velocity vectors of the vessels, while the continuous lines show the desired paths given by the COLREGS.

Rule 8 - Action to avoid collision

For vessels in situations with a risk of collision, the action taken should be made in ample time and the action itself should be readily apparent. In addition, the action taken should result in a passing at a safe distance, and it should be in accordance with the other rules in Part B.

Rule 13 - Overtaking situations

A vessel shall be deemed to be overtaking an other vessel when coming up from a direction more than 22.5 degrees abaft the other vessels beam. The overtaking vessel should keep out of the way of the vessel being overtaken. This situation is illustrated in Figure 2.2.

Rule 14 - Head-on situations

This occurs when two power-driven vessels are meeting on reciprocal or nearly reciprocal courses and there is a risk of collision. Each vessel should alter their course to starboard such that so that each shall pass on the port side of the other. Figure 2.3 shows a head-on situation.

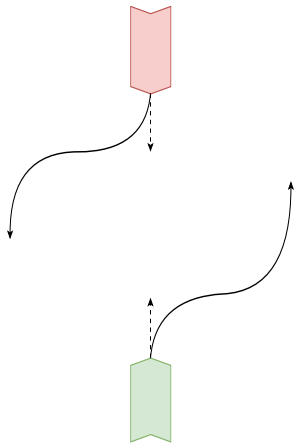


Figure 2.3: COLREGS head-on situation.

Rule 15 - Crossing situations

The vessel which has the other on her starboard side should keep out of the way of the other and avoid passing ahead of the other vessel. Two crossing situations are shown in

Figure 2.4.

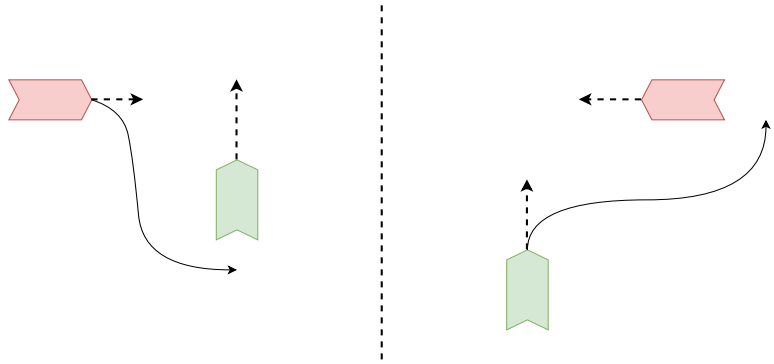


Figure 2.4: COLREGS crossing situations.

Rule 16 - Action by give-away vessel

Every vessel which is directed to keep out of the way of another vessel, shall so far as possible, take early and substantial action to keep well clear.

Rule 17 - Action by stand-on vessel

The vessel which is classified as the stand-on vessel, should keep her course and speed. However, if it becomes apparent that the other vessel is not taking appropriate action to keep out of the way, the stand-on vessel shall take action to avoid collision.

Chapter 3

Maritime collision avoidance

This chapter presents some of the theory and methods used in maritime collision avoidance systems. It starts by giving a background story of the development of COLAV for ASVs, continuing with two sections about the COLAV methods used in this thesis, namely the VO algorithm and the SBMPC algorithm. At last, there is a section describing some of the other existing maritime COLAV algorithms.

3.1 Background

Autonomous Surface Vessels (ASVs), Unmanned Surface Vessels (USVs) or Autonomous Surface Crafts (ASCs) are all different names of mostly the same category of vessels. As the names imply, the operator is removed from the direct control of the vehicle - leaving itself to decide what to do in a given situation. Mostly, the purpose of using a vessel is to transport something or someone from one place to another. Two main aspects can be derived from this: one is concerned with the problem of following a path connecting the start and end point - called path planning. The other aspect is about not crashing into other vessels or objects when following this path. This is called collision avoidance (COLAV).

Simple collision avoidance methods for maritime vessels started to appear in the early 1960s [57]. The focus of these early studies were situations where two ships were on a collision course. One of the proposed solutions was a starboard maneuver where the LOS vector should rotate counter clockwise during the maneuver. Since then, there have been numerous studies related to the same situation but also more complex situations have been studied.

Since the early 90s, research regarding path planning for marine vessels has seen a huge growth [57]. Path planning can be divided into two groups: the deterministic approach and the heuristic approach. The deterministic approach finds the best solution by following a set of predefined steps, while the heuristic method is only interested in finding an acceptable solution by searching through a subspace of the total search space. As a result, the heuristic approach requires less computation time and is preferred in simpler and faster systems.

A similar distinction is seen for COLAV algorithms, where they can be divided into reactive and deliberate algorithms [17]. Deliberate methods generate global paths based on available data of the complete environment [52], while reactive algorithms only consider the immediate available information from sensor data. Reactive algorithms are often preferred when the environment changes rapidly because of their usually low computational cost, but as the decisions are based on only local information, these algorithms might find suboptimal paths. All COLAV methods described in this thesis are reactive methods.

Today, the vessels are still controlled by humans, but assisted by advanced maritime control systems. The transition to a completely autonomous fleet of vessels may be a gradual process, ranging over a period of a few decades [49]. The first applications is expected to be on existing upgraded vessels, where a reduced number of crew members still is on-board, ready for specific tasks or in case of problems. However, as mentioned in the introduction, a number of high-level autonomous projects have started to appear [15, 43].

By introducing unmanned ships, a number of new possibilities regarding ship design and production opens up. When having no people on board, there is no need for a deckhouse. Additional human-based constraints may also be removed, resulting in a cheaper and lighter ship which is able to carry even more cargo. Being a company owning a fleet of autonomous vessels, operation optimization and profit maximization may also be easier; the best combination of route, cargo, maintenance schedule and fuel price for the fleet as a whole can be identified by looking at individual ships [49].

Although bringing a wide variety of benefits, and the potential of reducing human based errors, the introduction of autonomy to the maritime sector also induce a number of new risks that need to be addressed. The identification of these risks and being able to handle them in a satisfying manner is crucial.

3.2 Maritime COLAV based on model predictive control

Originally proposed by Johansen et al. [37], SBMPC is a collision avoidance method based on the more familiar concept called MPC. MPC is a model based controller design procedure which has proven to be effective in different types of applications, especially in the process industry [61]. MPC in general can be summarized as follows:

- A model of the plant is used to predict future states over finite prediction horizon based on a sequence of inputs.
- An optimal set of control inputs are found by minimizing an objective function over the prediction horizon.
- The first control input of the optimal set is applied to the plant. This process is repeated at the next time step when measurement updates are available.

A number of other maritime COLAV methods also relies on the concepts from MPC. In [28], a COLAV system with COLREGS compliance uses the SBMPC method to reduce the dependability of knowing the exact guidance scheme. The result is a COLAV system where it is possible to switch guidance strategies on the same vessel while running the same SBMPC COLAV algorithm. Eriksen et. al [17] presents a deliberate COLAV method based on the MPC principles and nonlinear programming. It is used as a middle level COLAV method in a three layer hybrid COLAV architecture.

3.2.1 SBMPC architecture

In the SBMPC method, the objective function and its constraints are defined by the plant model in addition to environmental forces, risk formalization, hazard and operational costs and objectives. The main challenge is to find the optimal solution of the optimization problem. Because of the complex nature of collision avoidance scenarios, non-convex optimization formulations may arise, leaving the task of finding the optimal solution harder and in some cases impossible. As a collision avoidance system, not finding a solution may not be an alternative.

In order to mitigate these problems, concepts from the literature on robust MPC can be used. A simple, but effective approach used in [37], is to select the control behavior based on a comparison between cost and feasibility among a finite set of possible control behaviors. This is used in the SBMPC implementation, and thus avoiding the numerical optimization present in conventional MPC. The drawbacks by doing this, is reduced degrees of freedom available for control, making the process of determining the set of possible control behaviours even more important.

As seen from Figure 3.1, the COLAV system requires information about the obstacles current state as well as their future trajectories. For the mission planning module, navigation information such as position, heading and velocity is needed to calculate a desired path. The path information is then handed to the autopilot, but if this nominal trajectory causes risk of collision, the COLAV system will adjust the trajectory such that collision is avoided and COLREGS is satisfied. This new trajectory is chosen to be as close to the nominal one as possible.

The modularity of the architecture proposed in Figure 3.1 shows that the COLAV system

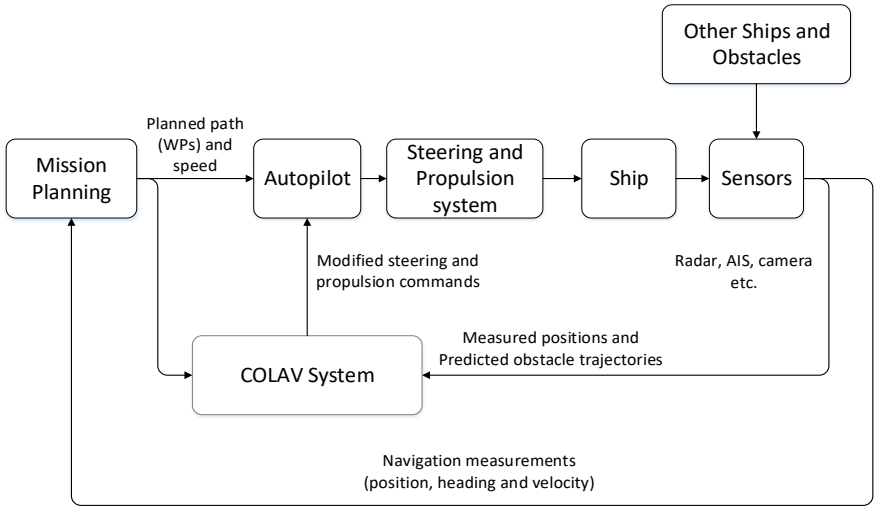


Figure 3.1: Overview of the modules and the information flow in the SBMPC architecture [37].

is separated from the steering system and mission planning module - making it easy to change COLAV system or add it to existing system where additional safety is required.

3.2.2 SBMPC COLAV

The collision avoidance functionality in the SBMPC architecture is based on a set of assumptions and choices. It is assumed that the COLAV system operates much faster than real time - making it able to calculate the ”best” control input in a real-time system. The optimization problem is solved for a given prediction horizon with re-optimization at regular intervals, often when new sensor data is available. The total set of scenarios is then given a cost based on a predicted possibility for grounding, collision hazards and COLREGS compliance. The control input associated with the scenario with lowest score is then passed on to the autopilot. The overall steps in the COLAV module can then be summarized by:

1. For each control behaviour, the future trajectories of the own-ship and the obstacles are predicted.
2. The trajectories (own-ship and obstacles) from one control behaviour, are evaluated by assigning a cost based on a set of criteria. This criteria set may include risk of

collision, COLREGS compliance etc.

3. Select the control behaviour that yields the lowest cost, and apply the first input to the system.
4. Repeat this loop when new sensor measurements are available.

Trajectory prediction

Uncertainty in both the own-ship trajectory and trajectories of obstacles is one of the general challenges for a COLAV method. To estimate the obstacles future states, a simple straight line trajectory estimate is often used. It is assumed that the obstacles will continue with the same velocity which they had at the last time step. When having little or no information on where the obstacles future path is going to lie, this simple estimation scheme will provide enough accuracy, but more complex estimation schemes can also be applied.

The estimation of the own-ship's trajectory is easier since ship dynamics information often is attainable in addition to path information (from the path planner module). This makes it possible to realize a manoeuvring model describing the ships motion. In this thesis, a 3 DOF horizontal plane motion model described in section 2.2.1 is used for this purpose.

Control behavior selection

The control behavior is decided by evaluating a finite number of different control behaviors. During the (future trajectory) estimation process, different control behaviour may be applied. One such set of control behaviours, is called a scenario. Higher performance can be expected if using different control behaviours, but at the cost of computational power.

Since a systems computational power is limited, not all control behaviors can be tested. According to [37], a minimum set of alternative control behaviors that should be tested is:

- Course offset at $\{$-90, -75, -60, -45, -30, -15, 0 , 15, 30, 45, 60, 75, 90$\}$ (in degrees).
- Keep speed (nominal propulsion), slow forward, stop and full reverse propulsion commands.

The control behaviours are given by offsets, i.e., if "keep speed" and a course offset of 0, the nominal velocity is chosen. The propulsion commands are given by $P_C \in \{-1, 0, 0.5, 1\}$.

Assuming that only one control behavior is used on the entire prediction horizon, there are $13 \cdot 4 = 52$ control behavior combinations. From a safety perspective it is desirable to test

more than just one control behaviour. Let's say the system is allowed to make p number of changes to the control behavior, then the total number of scenarios to be tested is 52^p. This number grows exponentially with respect to the number of control behavior changes, so from a computational point of view, the least amount of changes should be applied.

In this thesis, the evaluated scenarios a are taken from the set where only one control behaviour is used, i.e. the number of scenarios equals the number of control behaviours.

COLREGS compliance

In order to determine if a collision is at risk and to find out whether or not the COLREGS are met, there is a need for a situation identification procedure. From COLAV and COLREGS literature it is common to distinguish between four different situation types: crossing, head-on, overtaking, being overtaken. When reaching this stage in the COLAV process, the trajectories of the own-ship and the obstacles are predicted, making it able to calculate predicted situations for all the different trajectories.

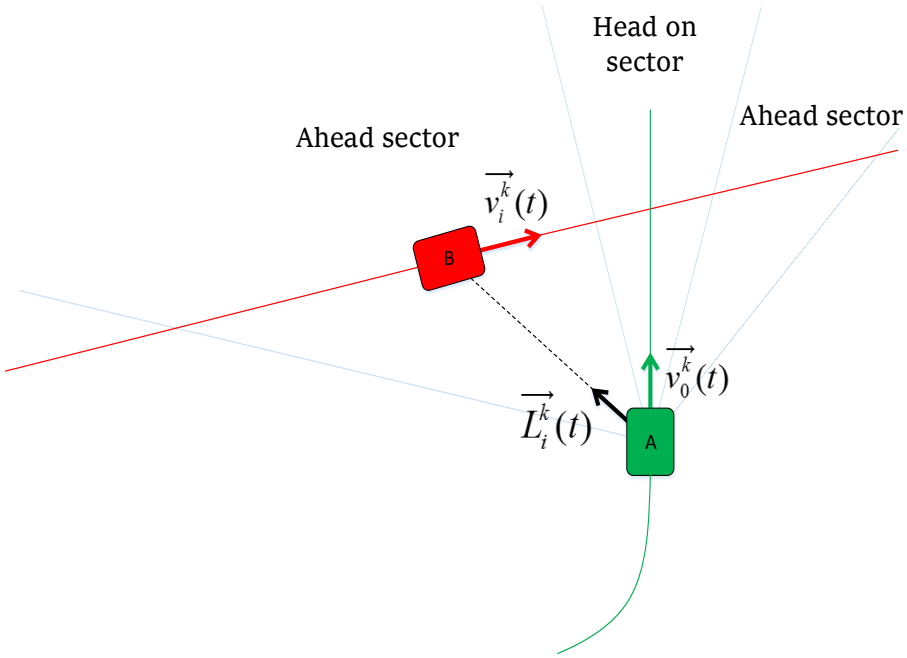


Figure 3.2: Sector information at a future time step used for hazard evaluation. Inspired by figure in [37].

Figure 3.2 shows the properties used to determine the situation type at a given time step in a

given scenario. The green dotted line is the predicted trajectory of the own-ship, calculated by considering the current state of own-ship in addition to control behaviors and nominal path given by way-points. The predicted trajectory of obstacle with index i is shown as the red dotted line. This straight-line prediction is based on the most recent estimates of this obstacle's state. The green and red rectangles are used to indicate the position at a future time step t for the own-ship and obstacle i, respectively. $\vec{L}_i^a(t)$ is a unit vector pointing in the LOS direction from the own-ship to obstacle i in scenario a, while $\vec{v}_0^a(t)$ and $\vec{v}_i^a(t)$ are the predicted velocities at time t in scenario a for the own-ship and obstacle with index i, respectively.

A set of Boolean variables are declared to be used in the scenario evaluation:

- To decide if an obstacle is so close that action has to be made in order to comply with the COLREGS, a variable called CLOSE is set to TRUE if

$$d_{0,i}^a(t) \leq d_i^{cl} \tag{3.1}$$

  where $d_{0,i}^a(t)$ is the distance from the own-ship to obstacle i and d_i^{cl} is a parameter determining the maximum range of where the COLREGS apply. This distance should be adjusted based on the own-ship and obstacles manoeuvrability and prediction uncertainty.

- The own-ship is said to be OVERTAKEN by obstacle i at time t in scenario a if the obstacle has higher speed, is close to the own-ship and

$$\vec{v}_0^a(t) \cdot \vec{v}_i^a(t) > \cos(68.5^\circ) \, |\vec{v}_0^a(t)| \, |\vec{v}_i^a(t)| \tag{3.2}$$

  where the angle 68.5° can be adjusted based on obstacle type and velocity.

- If the bearing angle of $\vec{L}_i^a(t)$ is larger than the heading angle of the own-ship, the obstacle with index i at time t in scenario a is said to be STARBOARD of the own-ship.

- A variable called HEAD-ON is set if obstacle i at time t in scenario a is close to the own-ship, the obstacle speed $\vec{v}_i^a(t)$ is not close to zero, and

$$\vec{v}_0^a(t) \cdot \vec{v}_i^a(t) < -\cos(22.5^\circ) \, |\vec{v}_0^a(t)| \, |\vec{v}_i^a(t)| \tag{3.3}$$

$$\vec{v}_0^a(t) \cdot \vec{L}_i^a(t) > \cos(\phi_{ahead}) \, |\vec{v}_0^a(t)| \tag{3.4}$$

  where ϕ_{ahead} is an angle that has to be found. The angle 22.5° can be adjusted and is based on the angle now defined to be 68.5°.

- Obstacle i is said to be CROSSED at time t in scenario a if it is close to the own-ship and

$$\vec{v}_0^a(t) \cdot \vec{v}_i^a(t) < \cos(68.5^\circ) \, |\vec{v}_0^a(t)| \, |\vec{v}_i^a(t)| \tag{3.5}$$

  where the angle 68.5° can be adjusted based on obstacle type and velocity.

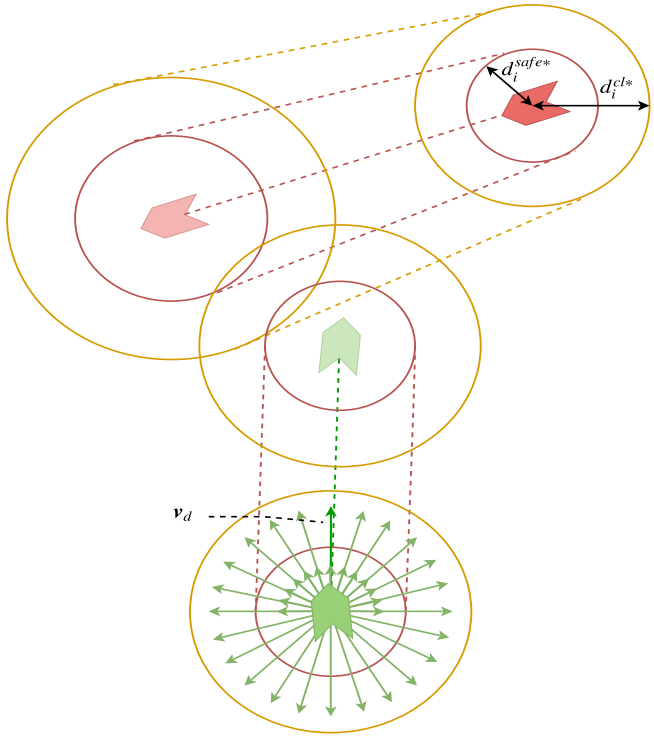


Figure 3.3: SBMPC trajectory prediction. The radius of the circles around the obstacle and the own-ship are based on the COLREGS distance (d_i^{cl}) and the safe distance (d_i^{safe}). Having an uncertain estimate of the obstacle's state will result in larger safety distances. This is illustrated by the larger circles around the obstacle vessel for the future time step. The green colored vectors around the own-ship is to visualize all the different search velocities.

Scenario evaluation

Based on the definitions and material presented earlier in this section, the scenario evaluation quantification can be carried out. This is done by assigning a cost to each scenario - a cost based on these three properties:

- Collision risk factor.
- COLREGS violation.
- Deviation from nominal trajectory.

The collision risk factor is given by

$$\mathcal{R}_i^a(t) = \begin{cases} \frac{1}{|t-t_0|^{p_r}} \left(\frac{d_i^{safe}}{d_{0,i}^a(t)} \right)^{q_r}, & \text{if } d_{0,i}^a(t) \leq d_i^{safe} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

where t_0 is the current time and t is a future time instant. From a COLREGS perspective, the parameters d_i^{safe} and $q_r \geq 1$ must be chosen such that appropriate distance are kept to other vessels. If the predicted state of an obstacle is uncertain, the uncertainty measure from the tracking system should be used to determine the distance d_i^{safe}. The parameter $p_r \geq \frac{1}{2}$ is used to weighting risks that are close in time higher than the ones that appear later. This is due to more accurate predictions for the short-time risks and because there is less time to take action. According to [37], typical choices for the exponents are $p_r = 1$ and $q_r = 4$.

To determine if the own-ship is violating any COLREGS rules at time t in scenario a, a binary term $\mu_i^a(t) \in \{0, 1\}$ is introduced. $\mu_i^a(t)$ equals 1 if the COLREGS are violated, 0 if not, and is given by

$$\mu_i^a(t) = \text{RULE14 or RULE15} \tag{3.7}$$

where

$$\text{RULE14} \;=\; \text{CLOSE \& HEAD-ON \& STARBOARD} \tag{3.8}$$

$$\text{RULE15} \;=\; \text{CLOSE \& STARBOARD \& CROSSED \& NOT-OVERTAKEN.} \tag{3.9}$$

By combining the presented material in this subsection, the total cost associated with scenario a based on available information at time t_0, is given by

$$\mathcal{H}^a(t_0) = \max_i \max_{t \in D(t_0)} \left(\mathcal{R}_i^a(t) + \kappa_i \mu_i^a(t) + f\left(P^a, \chi_{ca}^a\right) \right) \tag{3.10}$$

where $D(t_0) = \{t_0, t_0 + T_s, ..., t_0 + T_{pred}\}$ is the discrete sample times with T_s as sample interval and T_{pred} as prediction horizon. $f\left(P^a, \chi_{ca}^a\right)$ represents the cost related to path following and is given by

$$f(P, \delta) = k_P(1 - P) + k_\chi \chi_{ca}^2 + \Delta_P(P - P_{last}) + \Delta_\chi(\chi_{ca} - \chi_{ca,last}) \tag{3.11}$$

where k_P and k_χ are tuning parameters used to prioritize nominal speed and course, while Δ_P and Δ_χ are penalty functions used to influence the cost in order to prioritize readily apparent maneuvers. Both course parameters (k_χ and Δ_χ) include two different parameters based on if the course to be searched results in a turn to starboard or port.

The control behavior with the lowest cost is commanded to the autopilot module.

3.2.3 SBMPC parameter identification

The implementation of the SBMPC algorithm used in this thesis is set to update the control offset every 10 seconds. This is mainly done in order to reduce the simulation time, and

because of the slow dynamics of the simulated objects this will not have a significant affect on the algorithm's performance.

The minimum value of the safety parameters, d_i^{cl} and d_i^{safe}, are chosen based on the evaluation metrics described in 5.3.2. When tracking is performed, each obstacle's existence probability is used to adjust d_i^{cl} and d_i^{safe} according to

$$d_i^{cl} = \min\left(2d_i^{cl}, \frac{d_i^{cl}}{\varepsilon^i}\right), \tag{3.12}$$

$$d_i^{safe} = \min\left(2d_i^{cl}, \frac{d_i^{safe}}{\varepsilon^i}\right) \tag{3.13}$$

where ε^i is the existence probability for obstacle i given by the tracking system. The more certain the obstacle track is, the smaller the safety distances.

As for the remaining parameters, the values were found by tuning in a controlled environment until satisfactory behaviour was reached. As the tuning process for this algorithm is quite comprehensive, the parameters used in this thesis should not be considered to be perfect and there is a potential for improvement.

Table 3.1 lists all parameter values.

Parameter	**Value**	**Unit**
T_{pred}	600	s
d_i^{cl}	2000*	m
d_i^{safe}	1000*	m
p_r	2	
q_r	10	
κ_i	10	
k_P	0.1	
$k_{\chi,port}$	1	
$k_{\chi,starboard}$	0.05	
Δ_P	0.5	
$\Delta_{\chi,port}$	0.5	
$\Delta_{\chi,starboard}$	0.5	

Table 3.1: Parameters for the SBMPC algorithm.

3.3 Velocity Obstacle (VO) COLAV

This section presents another COLAV method based on what is called the Velocity Obstacle (VO). Originally developed by [21], the method was used to determine "safe" velocities

in dynamic environments for simple motion obstacles, e.g. 2 DOF robot manipulators. By using the current positions and velocities of the surrounding obstacles to calculate relative velocities (with the assumption of linear trajectories), safe velocities can be found. Ever since the method was first released, a number of extensions have been proposed, e.g. the "Reciprocal Velocity Obstacle" [59] used for navigation among actively moving obstacles and a crowd simulation in [60].

Kuwata et. al [38] presented an algorithm utilizing the concept of VO in addition of being able to handle the COLREGS. This system was implemented on a 12 meter long vessel and tested in situations where up to four obstacles were present. Minne describes a VO implementation based on the work by [38] and Stenersen [55], which is presented in the following subsections.

3.3.1 The Veloctiy Obstacle

As defined in [38], the VO of obstacle B for own-ship A is given by

$$VO_B^A(\boldsymbol{v}_B) = \{\boldsymbol{v}_A \mid \lambda(\boldsymbol{p}_A, \boldsymbol{v}_A - \boldsymbol{v}_B) \cap (\mathcal{B} \oplus -\mathcal{A}) \neq \emptyset\} \tag{3.14}$$

where $\boldsymbol{v}_B$ and $\boldsymbol{v}_A$ are the velocity of obstacle B and the own-ship, respectively. $\boldsymbol{p}_A$ is the own-ships position, while $\mathcal{A} \oplus \mathcal{B} = \{\boldsymbol{a} + \boldsymbol{b} \mid a \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B}\}$ is called the Minkowski sum. $-\mathcal{A} = \{-\boldsymbol{a} \mid \boldsymbol{a} \in \mathcal{A}\}$ defines the reflection of a set, and $\lambda(\boldsymbol{p}_A, \boldsymbol{v}_A - \boldsymbol{v}_B)$ is a vector starting in the own-ships position pointing in the direction of the relative velocity between the own-ship and the obstacle.

By taking the Mikowski sum of the geometry of obstacle B, $\mathcal{B}$, and the reflection of the set representation of the own-ships geometry, $\mathcal{A}$, we get the combined geometry. If the centre of the own-ship lies inside this set, a collision has occurred. Since the Mikowski sum is a quite expensive calculation, a common simplification is to use a circle approximation of the geometries of both the obstacle and the own-ship. This is seen in Figure 3.4. The own-ship is now a circle with radius r_A while the combined geometry is a circle with radius $r_A + r_B$.

The interpretation of equation 3.14 can now be carried out. If the relative velocity vector $(\boldsymbol{v}_A - \boldsymbol{v}_B))$ from the own-ship position $(\boldsymbol{p}_A)$ to the combined geometry intersects, the velocity $\boldsymbol{v}_A$ is said to belong to the velocity obstacle induced by obstacle B $(VO_B^A(\boldsymbol{v}_B))$ and will result in a collision. This is seen as the shape of a cone in Figure 3.4 (and Figure 3.5).

Even though the VO algorithm here is only outlined for one obstacle, it is easily transformed to cases where multiple obstacles are present. The only difference is that one will have an overall VO containing several VO sections (cones) for the individual obstacles.

The process of determining if a velocity lies inside the VO, boils down to find the left and right edges of the cone induced by the obstacle. This can be done in a variety of ways, but is not shown here, but we refer to [55] for one of these methods.

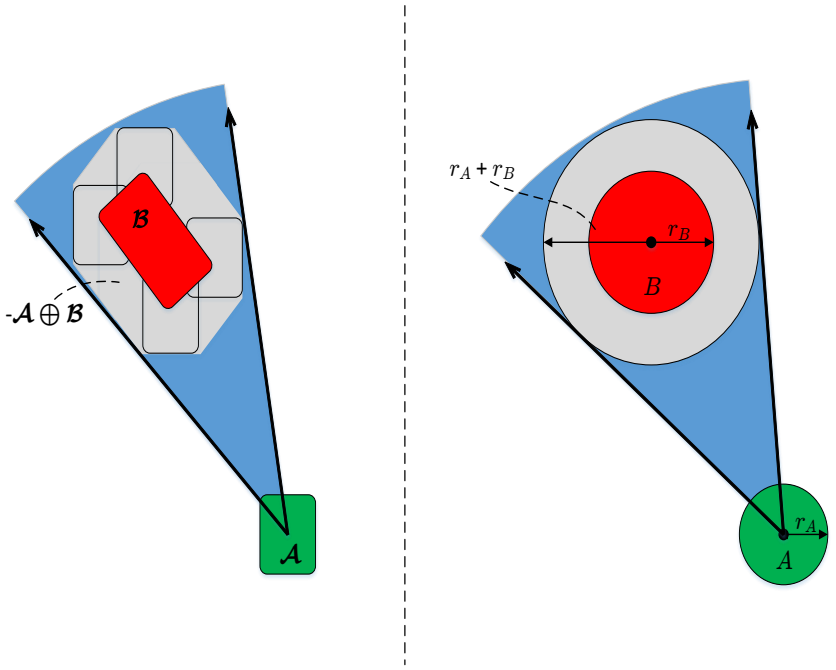


Figure 3.4: The left figure shows the true Mikowski sum (based on figure from [38]), while the circular approximation is shown on the right.

3.3.2 VO with COLREGS compliance

After the simplification process done in the previous section, and the addition that the cone of the VO from the obstacle has been found, we are left with a simple collision avoidance system which may be able to stay away of other obstacles. To be able to make more complex decisions on which velocity to choose, additional functionality has to be added.

When the VO has been found, it is a trivial task to add COLREGS compliance functionality. One method, used by [44] originally from [38], is to divide the area around the VO cone into three different regions (shown in Figure 3.5). Given a velocity, $\boldsymbol{v}_A$, the region separation is used to determine which side the own-ship will pass the obstacle on. It is then a straightforward task to check if the COLREGS are satisfied.

Mathematically, to find which region a specific velocity ($\boldsymbol{v}_A$) belongs to, [38] used the relations given below. If a velocity $\boldsymbol{v}_A$ lies in the $\mathcal{V}_3$ region - meaning that the relative velocity points in a direction away from the obstacle, this relation has to be satisfied

$$(\boldsymbol{p}_B - \boldsymbol{p}_A) \cdot (\boldsymbol{v}_A - \boldsymbol{v}_B) < 0. \tag{3.15}$$

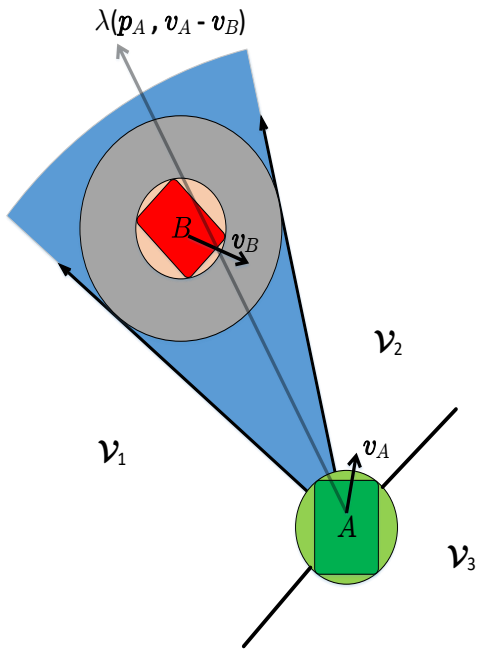


Figure 3.5: Region separation used to determine which side the ASV will pass the obstacle on. Inspired by figure in [44].

This velocity will result in larger distance between the own-ship and the obstacle. For cases when $\boldsymbol{v}_A$ is in region $\mathcal{V}_1$, we have

$$\mathcal{V}_1 = \{\boldsymbol{v}_A \mid \boldsymbol{v}_A \notin VO_B^A(\boldsymbol{v}_B),\ \ \boldsymbol{v}_A \notin \mathcal{V}_3,\ \ [(\boldsymbol{p}_B - \boldsymbol{p}_A) \times (\boldsymbol{v}_A - \boldsymbol{v}_B)]_z < 0\} \tag{3.16}$$

where $[\cdot]_z$ denotes the z component of the vector. Choosing a velocity with these properties will result in a situation where the ASV will pass the obstacle while having it on the starboard side. If complying to the COLREGS, this region should not be chosen if the active situation is either overtaking, head-on, or crossing from the right. Details regarding this is discussed later on. For the last region ($\mathcal{V}_2$) to be active, $\boldsymbol{v}_A$ has to satify this expression

$$\mathcal{V}_2 = \{\boldsymbol{v}_a \mid \boldsymbol{v}_a \notin VO_B^A(\boldsymbol{v}_B),\ \ \boldsymbol{v}_a \notin \mathcal{V}_1\}. \tag{3.17}$$

This will result in a situation where the obstacle is passed by having it on the port side.

It is important to mention that if the obstacle vessel is crossing from the left (and in overtaking scenarios), no COLREGS constraints are active because the responsible of avoiding a critical situation lies in the hands of the obstacle vessel. If the obstacle vessel is not complying with the COLREGS, and a dangerous situation is about to occur, the own-ship will take action as the hazard avoidance in the method is still active.

3.3.3 VO algorithm walk through

Precollision check

To increase the efficiency of the algorithm, a precollision check is performed before any other calculation is done. The check removes not-relevant obstacles from further calculation. To give an example: If an obstacle is visible by the sensor system, but is so far away in distance that collision is no way near in time and no COLREGS rules apply, then it is no point in calculating a desired velocity based on state information about this obstacle.

The precollision check starts by calculating the Closest Point of Approach (CPA) between the own-ship A and the obstacle vessel B. At first, the time to CPA, t_{CPA}, is found by using the positions and velocities of two vessels

$$t_{CPA} = \begin{cases} 0, & \text{if } ||\boldsymbol{v}_A - \boldsymbol{v}_B|| \leq \epsilon_{VO} \\ \frac{(\boldsymbol{p}_A - \boldsymbol{p}_B)\cdot(\boldsymbol{v}_A - \boldsymbol{v}_B)}{||\boldsymbol{v}_A - \boldsymbol{v}_B||^2}, & \text{otherwise.} \end{cases} \tag{3.18}$$

The distance at CPA, d_{CPA}, for the vessels is then calculated by

$$d_{CPA} = ||(\boldsymbol{p}_A + \boldsymbol{v}_A t_{CPA}) - (\boldsymbol{p}_B + \boldsymbol{v}_B t_{CPA})|| \,. \tag{3.19}$$

For each moving vessel given by the sensor system, the algorithm examines COLREGS compliance only if situations are likely to occur in near future, by checking if

$$0 \leq t_{CPA} \leq t_{max} \quad \text{and} \quad d_{CPA} \leq d_{max} \tag{3.20}$$

are satisfied.

Collision type determination

If the CPA of the obstacle in question satisfies the precollision check, the next step is to determine what kind of COLREGS situation the obstacle will induce. Minne [44] describes a method from [55] which uses the relative bearing to determine the type of the situation.

The relative bearing is the angle given by

$$\beta = \text{atan2}(y_A - y_B, x_A - x_B) - \psi_B \tag{3.21}$$

where $\boldsymbol{p}_A = [x_A, y_A]$, $\boldsymbol{p}_B = [x_B, y_B]$ and ψ_B is the yaw angle of the obstacle ship. atan2 is the four-quadrant version of the ordinary arctan function. The situation type is found by only seeing the relative bearing from the obstacle's point of view (see Figure 3.6). A new variable $\beta^{180} = \beta - 180°$, called the normalized relative bearing is defined. By applying this, the four situation types are defined according to:

- Head-on: $\beta^{180} \in [-15°, 15°)$

- Crossing from right: $\beta^{180} \in [15°, 112.5°)$
- Overtaking: $\beta^{180} \in [112.5°, 180°) \cup [-180°, -112.5°)$
- Crossing from left: $\beta^{180} \in [-112.5°, -15°)$

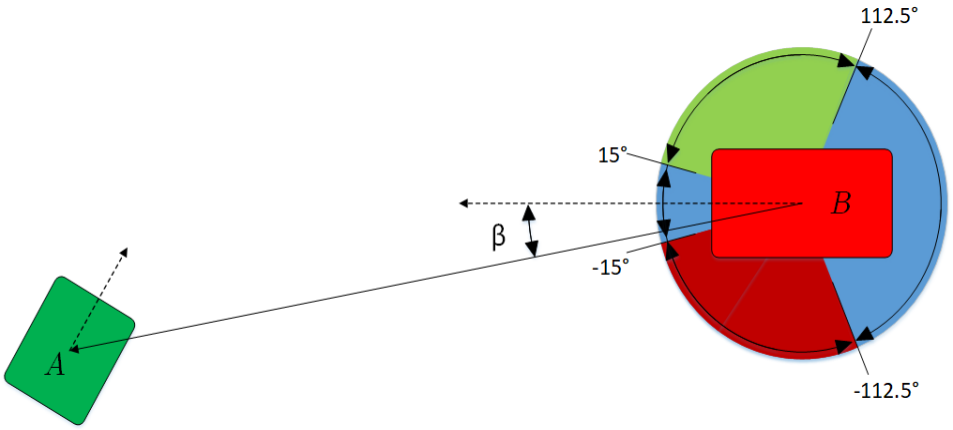


Figure 3.6: Determination of situation type. Relative bearing β is shown. Inspired by figure from [55].

Velocity determination

After determining the current situation type, there is a need to find the "best" velocity given these conditions. This is implemented as a search grid of possible headings and speeds where each cell represents a velocity. For each cell, a cost is assigned based on: if a collision will occur, if any COLREGS rules are active and how much it deviates from the nominal velocity given by the ILOS controller. The cost function, originally formulated by [55], is given by

$$C_{ij}^{'} = \alpha_c f_{collision}^{'}(u_i, \psi_j) + \beta_c f_{COLREGS}(u_i, \psi_j) + \tilde{\boldsymbol{v}}_{ij}^T \boldsymbol{Q} \tilde{\boldsymbol{v}}_{ij}. \tag{3.22}$$

$f_{collision}^{'}$ is a boolean function returning 1 if a collision will occur with the given speed (u_i) and heading (ψ_j). If the COLREGS are violated, $f_{COLREGS}$ returns 1, while α_c and β_c are parameters that is used to determine the cost related to collision and COLREGS violation, respectively. $\tilde{\boldsymbol{v}}_{ij}$ is given by

$$\tilde{\boldsymbol{v}}_{ij} = \begin{bmatrix} u_d \cos(\psi_d) - u_i \cos(\psi_j) \\ u_d \sin(\psi_d) - u_i \sin(\psi_j) \end{bmatrix} \tag{3.23}$$

and represents the deviation from the desired speed (u_d) and heading (ψ_d), while $\boldsymbol{Q} = q_c \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is a weighting matrix used to assign a cost to the velocity deviation. After

assigning a cost to every pair of speed and heading in the search grid, the pair with the lowest score is found and fed to the next module in the COLAV system.

3.3.4 Cost function modifications

During the testing phase of the VO algorithm, it was seen that some situations were not handled in accordance with the COLREGS. The algorithm was seen to handle crossing situations, especially when being the give-way vessel, in a poor way. The algorithm often chose to reduce the speed and not alter the course to starboard. This resulted in the obstacle vessel making avoiding maneuvers, which should not be necessary. There were also head-on situations where the maneuvers weren't apparent enough. So, to improve the performance of the VO algorithm, a new set of functionality is added to the cost function.

As the algorithm already includes which COLREGS situation is active for a given obstacle, this is used to influence the cost function in a more direct manner. First, we define a speed penalty function, given by

$$f_{speed} = \begin{cases} \min(1, |u_{d,last} - u_i|), & \text{if head-on or crossing from left} \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

where $u_{d,last}$ is the desired velocity from the previous time step. This function will make speed alteration more expensive, and thus, course alteration relatively cheaper. As this only will result in more course changes being taken, we need an additional penalty to make the algorithm prefer course alteration to starboard (when this is desired). If the current COLREGS situation is either head-on or crossing left, and

$$\psi_j - \psi_{d,colreg} < 0 \tag{3.25}$$

where $\psi_{d,colreg}$ was the desired heading the last time the COLREGS situation was updated, a new boolean function $f_{port} = 1$. This function is inspired by the evaluation penalty $\mathcal{P}_{PortTurn}$ described in 5.3.2.

The above cost function additions is made to improve the give-way vessels maneuvers, but we also need to adjust the stand-on vessel response in crossing situations. This is done by adjusting the boolean function, $f'_{collision}$. If the current COLREGS situation is crossing from left and the range to CPA is more than 2000 meters, the new function $f_{collision}$ returns 0. Based on the stand-on maneuver penalty in subsection 5.3.2, the stand-on vessel is allowed to make collision avoiding maneuvers if being close to the CPA. Using the range 2000 meters, the stand-on vessel is not penalized according to this penalty.

The old cost-function (3.22) is replaced with the cost function given in (3.26). ι_c and $\varkappa_c$ are used to weight the new speed and port penalties, respectively.

$$\begin{aligned} \mathcal{C}_{ij} = \alpha_c f_{collision}(u_i, \psi_j) + \beta_c f_{COLREGS}(u_i, \psi_j) + \iota_c f_{speed}(u_i, \psi_j) \\ + \varkappa_c f_{port}(u_i, \psi_j) + \tilde{\boldsymbol{v}}_{ij}^T \boldsymbol{Q} \tilde{\boldsymbol{v}}_{ij} \end{aligned} \tag{3.26}$$

3.3.5 VO parameter identification

All parameters used in the VO algorithm are found in Table 3.2. The values are chosen such that the safety constraints described in 5.3.2 are satisfied.

The own-ship radius (r_A) and obstacle-radius (r_b) are set to ensure a passing distance of 1000 meters. To satisfy rule 8, t_{max} is chosen to ensure that action is taken in ample time. The value of d_{max} is set to a significantly larger value than the combined radius of the combined geometries. This is to ensure that the algorithm doesn't assume that a collision situation is over when it is still taking place.

The cost function's adjustable parameters are chosen based on this simple logic: Collision is the most critical, and should be avoided for all cause. Second, the COLREGS should not be violated. Third, the velocity with the least deviation in speed and heading compared to the desired velocity given by the ILOS controller should be chosen.

The search grid is divided into evenly sized speed and heading sections. The resolution is 128 possible headings and 8 possible speeds, where the headings are centered around the desired heading (ψ_d). The speed values are in the range $[0\text{m/s}, 6\text{m/s}]$ and the heading alterations are in the range $[-135°, 135°]$.

Parameter	Value	Unit
r_A	500	m
r_B	500	m
t_{CPA}	1200	s
d_{CPA}	2000	m
ϵ_{VO}	0.1	
α_c	200	
β_c	100	
ι_c	50	
$\varkappa_c$	50	
q_c	1	

Table 3.2: Parameters for the VO algorithm.

3.4 Other COLAV methods

A numerous number of other COLAV methods for marine vessels exist. This section gives a short description of some of these methods.

3.4.1 Edge following

Edge following (also called the Bug algorithm) is a simple COLAV algorithm which only needs the relative angle and the distance to the object to calculate the new control input [40, 58]. It is used in the fields of robotic navigation and for Underwater Autonomous Vehicles (UAVs). The algorithm works by keeping a constant distance to the obstacle if the obstacle lies in the predefined path which the system is to follow. For UAVs, sonar is often used to estimate the obstacle boundary, but this can be hard or even impossible in some situations. The algorithms also assumes a holonomic vehicle, which is not true for an UAV [19]. So, in practise, a robust algorithm of this kind is not that easy to implement.

3.4.2 Dynamic window

Dynamic Window (DW) is another algorithm for collision avoidance used in several applications [53, 20]. It was originally developed for indoor mobile robots operating in hazardous and populated environments [26], but as a number of other methods used in robotics, this method has been applied and transformed for applications in other environments.

The main concept is to be able to deal with constraints imposed by limited velocities and accelerations [26]. When computing the next steering command, the method only considers a short time interval in order to reduce the general motion planning problem. During this time interval, the trajectories are approximated by circular curvatures, resulting in a two-dimensional search space of translational and rotational velocities. The search space is reduced such that the robot is able to stop safely, and by only considering velocities that can be reached in the next time interval. These velocities form the dynamic window, which is centered around the current velocity of the robot. A velocity is chosen by maximizing an objective function on the velocities from the dynamic window.

The DW methods used in marine COLAV often relies on a set of assumptions and approximations. The search space is found by neglecting sway motion and assuming constant surge speed and yaw rate. To prevent the algorithm from producing infeasible input commands, the dynamics of the vessel is taken into consideration. After the search space is defined, a trajectory is predicted by maximizing an objective function on the search space. The objective function is defined in a way which favors progression towards the end goal and that distances to obstacles are kept. This local reactive COLAV algorithm is more computationally heavy than for example the VO algorithm, but may perform better.

3.4.3 Potential field

At last we have a method called potential field [27]. Here, the obstacles are modeled as repulsive nodes while the target positions are modeled as attractive nodes. The system is then set up to follow the gradient of this potential field, but because of large and complex fields, the system may be trapped in local minima. Since the method does not include the systems dynamics, infeasible solution may also be found. The method has been implemented by numerous people, often in real-time systems, mainly because of its simplicity and low computational cost.

Chapter 4

Uncertainty and tracking

This chapter gives an introduction to how sensors and target tracking are used to handle the uncertainty induced in real-life COLAV scenarios. The first section describes the most used sensors for ASVs, which is followed by a general introduction to target tracking. The next sections presents some of the most common target tracking methods, finishing of with a section describing the implemented JIPDAF tracking system.

4.1 Uncertain world perception

All sensors include some kind of uncertainty. In autonomous systems where sensor information is a necessity for a well-functioning behavior, the need for sensor data filtering and processing is evident. No single sensor can provide sufficient performance under all possible conditions. Hence, in order to guarantee that the surrounding state information is accurate enough at all times, it is common to include multiple- and different types of sensors. The process of combining sensor data from multiple disparate sources to give more accurate information than the sensors would have given individually, is called *Sensor fusion* [16]. The sensor fusion process needs to be implemented such that subsequent path planning and COLAV methods always have a detailed description of the surrounding environment.

For ASVs in maritime environments, where all decision making is heavily based on sensor data of the surroundings, there is a need for an accurate sensor system. The most used sensors for ASVs are different types of radar, visual and thermal cameras, and LIDAR (Light Detection And Ranging). Figure 4.1 illustrates the sensor fusion concept with some of the possible maritime sensors available.

In the following, a description of several sensors - their usage, strengths and shortcomings are presented.

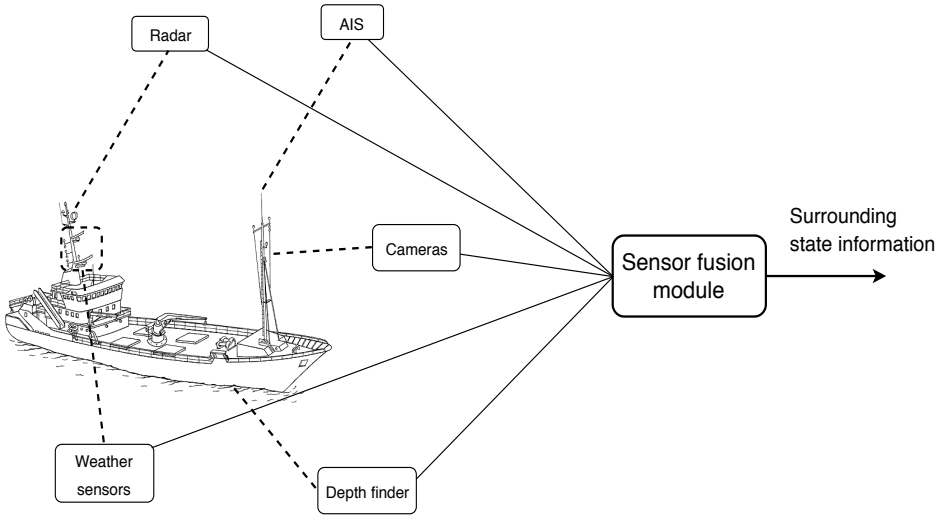


Figure 4.1: Sensor data are sent to the sensor fusion module for filtering and processing to make it understandable by the rest of the system. Vessel model is from [23].

4.1.1 Visual and infrared camera

A visual camera provides a digital image over the covered region in space. The image can be either monochrome or color, and is based on an array of passive light sensors, as opposed to radar and LIDAR where electromagnetic light is transmitted. Since the sensors are passive, the range to an object is not given, but can be found by using several cameras together in a more complex process (stereo vision). The practical use of stereo vision is limited because of poor accuracy - leaving the problem with range still present. Several other challenges exist; feature-extraction is non-trivial compared to LIDAR and radar. When there are large variations of light intensities in the image (very dark and very bright), the camera struggles to capture the whole image in a good way. Also, when having only very low light intensity, the visual camera does not perform well [33].

Some visual cameras cover parts of the infrared spectrum (near-IR), making it able to capture images in dark conditions. Near-IR is reflected from objects also at night, but the magnitude may be low. To strengthen the near-IR reflections from objects at night, one can use an external IR source to illuminate the objects. Thermal cameras are used when the radiation emitted has even higher wavelengths.

4.1.2 Radar

A radar is an object detection system which uses radio waves to determine the state of the surroundings. The radio waves transmitted from the radar antenna reflect off the objects and return to the receiver. The sensor provides relative bearing, range information over a wide range and possibly elevation. Radar is less affected by weather conditions compared to the cameras, but echoes from rain, snow and the water surface can be a problem [29]. Reflections of this kind, from elements which are not the objects of interest, and false measurements, are called clutter. This, together with issues with detecting close range objects (for traditional pulsed radars), are the most challenging problems when using radar.

4.1.3 Light Detection And Ranging (LIDAR)

LIDAR is a sensor method made to measure the distance to a target by emitting laser light and measure the reflected pulse with a sensor. Three-dimensional LIDARs can provide a 3D map of the surroundings with very high resolution. For marine applications, LIDARs can be used for close-range objects where the radar has limited value. One of the drawbacks when using LIDARs is the handling and processing of the large amount of data. The range is also heavily influenced by the reflectivity of the objects as well as weather conditions [33].

4.2 Target tracking

Target tracking is the procedure of estimating the state of a target (obstacle) based on measurement data given by a sensor system. The state of the target often includes position, velocity and heading, but additional state information as range, accelerations etc. may be included - all based on what kind of application the tracking is done. Since the measurement information often is corrupted (noisy), a method to calculate an estimate of the target's state with presence of uncertainty is needed. This is accomplished by, in addition to estimate the target's state, give an estimate of the target's uncertainty and the uncertainty of the measurements. In situations where there exist several measurements, the more uncertain measurements are given a lower weight than the measurements with less uncertainty.

For situations that only includes one measurement per target, methods like the Kalman Filter [11] or the particle filter [3] can be applied. A more challenging problem appears when there are several possible measurements per target, and you don't know which (if any) measurement is target originated (Figure 4.2). This is one of the main problems in all tracking systems and is called measurement origin uncertainty or data association uncertainty.

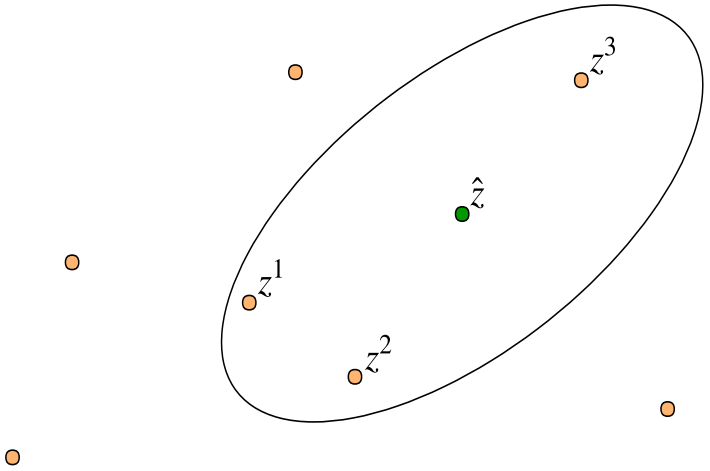


Figure 4.2: Situation with several measurements (z^i) inside the validation region around the predicted measurement ($\hat{z}$).

In target tracking applications, the measurements from the sensors need to be incorporated into the target state estimator. When using radar, the reflected signal from the target may be found within a time interval based on the anticipated range to the target [4]. Detections within this ”range gate” can then be associated with targets. If more measurements are available (for instance bearing, time, frequency, line-of-sight angels etc.), these can be combined to a multidimensional gate. This will make the search much more effective; instead of searching through the whole measurement space, one chooses the points where it is most likely that the target is. When a measurement is found within such a gate, it becomes a valid association candidate, even though it may not originate from the the target the gate pertains to. Because of this, this type of gate is called a validation gate or a validation region.

Based on the predicted target state vector, the validation region is formed such that the target measurement falls in it with high probability, called the gate probability [4], P_G. When one (or more) measurement is found within a validation region, we are faced with the aspect of data association uncertainty. Misdetections may be present, and if several measurements, we also need to figure out which of the measurements is target originated, and in addition calculate the sufficient statistics for the target. This includes the state estimate and its covariance matrix. Measurements outside the validation region is of no interest because of the low probability that these are target originated.

4.3 Motion and measurement model

A tracking method needs a suitable model of the target to be tracked. In tracking applications, observation data is often limited, making the target model even more important. The target motion model used in this thesis is a white noise acceleration model [39], known as a constant-velocity model. This is a very simple model, but it has proven to be suitable for these type of tasks.

The state vector of our model is defined as $\mathbf{x} = [\mathrm{N}, \mathrm{V_N}, \mathrm{E}, \mathrm{V_E}]$, where N and E are the north and east position of the target, and $\mathrm{V_N}$ and $\mathrm{V_E}$ are the north and east velocities of the target. All parameters are given in the NED reference frame. The complete model is defined as

$$\mathbf{x}_{k+1} = \mathbf{F}_T \mathbf{x}_k + \mathbf{v}_k, \qquad p(\mathbf{v}_k) = \mathcal{N}(\mathbf{v}_k; 0, \mathbf{Q}_T) \tag{4.1}$$

$$\mathbf{F}_T = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{Q}_T = \sigma_a^2 \begin{bmatrix} T^4/4 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/4 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix} \tag{4.2}$$

where $T = t_{k+1} - t_k$, which in general is time-varying. $\mathbf{v}_k$ is the process noise with probability density function $p(\mathbf{v}_k)$, where $\mathcal{N}$ denotes the normal distribution. The parameter σ_a can be seen as the process noise strength and is chosen in correspondence to the targets expected maneuverability [62].

We use a linear measurement model of the form

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, \qquad p(\mathbf{w}_k) = \mathcal{N}(\mathbf{w}_k; 0, \mathbf{R}) \tag{4.3}$$

where $\mathbf{w}_k$ is the measurement noise and $p(\mathbf{w}_k)$ is the normal probability density function. $\mathbf{H}$ and $\mathbf{R}$ are given by

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{R} = \sigma_r^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.4}$$

The measurement noise covariance matrix $\mathbf{R}$ used is constant and diagonal, reflecting an assumption that the size of the target is larger than the resolution cells [62].

4.4 PDAF

The probabilistic data associating filter is a single-target Bayesian tracking method able to handle cluttered environments. Originally presented by Bar-Shalom et. al [7], the method has been applied to a number of different tracking applications [48, 41, 5], mainly because of its simple and effective tracking scheme.

The PDAF method calculates the association probabilities to the target being tracked for each measurements inside the validation region. The calculation is done based on the distance from the measurement to the estimated target position, i.e. a measurement close in distance to the estimated target position is given a higher probability of being target originated than a measurement further away. This concept is shown in Figure 4.3.

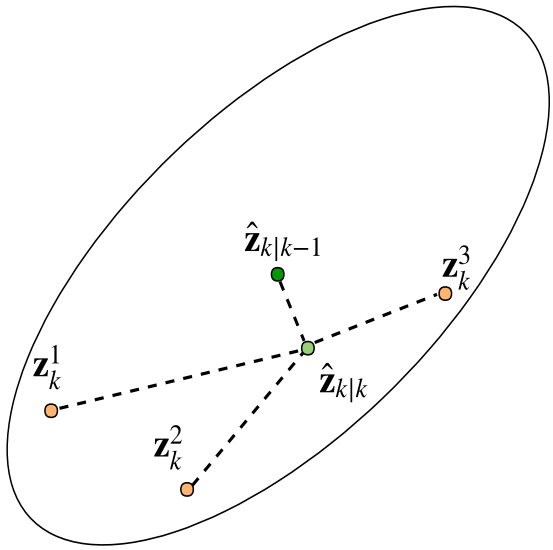


Figure 4.3: PDAF state estimation update where the updated target estimate $\hat{\mathbf{z}}_{k|k}$ is weighted based on the distance from the validated measurements $\mathbf{z}_k^i$ to the predicted $\hat{\mathbf{z}}_{k|k-1}^1$.

The PDAF relies on a set of assumptions:

- There is only one target of interest, whose kinematic model is given by equation 4.1 and 4.3.
- The track has been initialized.
- At each time step, a validation region defined by equation (4.10) is set up.
- Among possibly several validated measurements, at most one can be target originated.
- The remaining measurements are assumed to be clutter or false alarms and are modeled as independent and identically distributed with uniform spatial distribution.

- The target detections occur independently over time with known probability P_D.
- The past information about the target is summarized approximately by

$$p\left(\mathbf{x}_{k-1}|Z_{1:k-1}\right) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) \tag{4.5}$$

  which is the key assumption of the PDAF. $Z_{1:k-1}$ is the cumulative set of all measurements up to and including time step $k-1$.

4.4.1 PDAF Tracking

The propagation of tracks and covariance is assumed to evolve in time as the general Kalman filter prediction equations

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_T \hat{\mathbf{x}}_{k-1|k-1} \tag{4.6}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_T \mathbf{P}_{k-1|k-1} \mathbf{F}_T^T + \mathbf{Q}_T \tag{4.7}$$

$$\hat{\mathbf{z}}_k = \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \tag{4.8}$$

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \tag{4.9}$$

The Normalized Innovation Squared (NIS) can then be calculated by

$$\text{NIS} = (\mathbf{z}_k^i - \hat{\mathbf{z}}_k)^T \mathbf{S}_k^{-1} (\mathbf{z}_k^i - \hat{\mathbf{z}}_k) = \boldsymbol{\nu}_k^{iT} S_k^{-1} \boldsymbol{\nu}_k^i < \gamma_G \tag{4.10}$$

where $\boldsymbol{\nu}_k^i$ is called the measurement innovation. The NIS defines an elliptic region (the validation region) where the semiaxes of the ellipsoid are the square roots of the eigenvalues of $\gamma_G \mathbf{S}$. γ_G is the gate threshold which is obtained from the inverse Cumulative Distribution Function (CDF) of the χ^2-distribution with the degrees of freedom equal the dimension of the measurement vector $\mathbf{z}_k$. This threshold is used to determine if the measurement is to be associated with the target, and is usually chosen such that the target falls in the validation region with high probability (P_G).

The state update equation is given by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \boldsymbol{\nu}_k \tag{4.11}$$

$$\boldsymbol{\nu}_k = \sum_{i=1}^{m_k} \beta_k^i \boldsymbol{\nu}_k^i \tag{4.12}$$

$$\beta_k^i = \frac{1}{c} \exp\left(-\frac{1}{2} \boldsymbol{\nu}_k^{iT} \mathbf{S}_k^{-1} \boldsymbol{\nu}_k^i\right), \qquad i = 1...m_k \tag{4.13}$$

$$\beta_k^0 = \frac{1}{c} \frac{2(1 - P_D P_G)}{\gamma_G} m_k \tag{4.14}$$

with $\boldsymbol{\nu}_k$ as the combined innovation, m_k as the total number of measurements inside the validation region and c as a normalization constant. β_k^i is the probability of measurement

$\mathbf{z}^i$ being the true target measurement, and β_k^0 is probability of no measurement being target originated. The gain $\mathbf{K}_k$ and the updated covariance are calculated according to

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1} \tag{4.15}$$

$$\mathbf{P}_k = \beta_k^0\mathbf{P}_{k|k-1} + (1-\beta_k^0)\mathbf{P}_k^c + \tilde{\mathbf{P}}_k \tag{4.16}$$

where the covariance of the state updated with the correct measurement is

$$\mathbf{P}_k^c = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{4.17}$$

and the last term, called the Spread of Innovations (SOI), is given by

$$\tilde{\mathbf{P}}_k = \mathbf{K}_k\left(\sum_{i=1}^{m_k}\beta_k^i\boldsymbol{\nu}_k^i\boldsymbol{\nu}_k^{iT} - \boldsymbol{\nu}_k\boldsymbol{\nu}_k^T\right)\mathbf{K}_k^T \tag{4.18}$$

All three terms of equation (4.16) can be further analyzed. With probability β_k^0, there are no correct measurements, resulting in no state estimation update. This is shown in (4.16) as weighting the prediction covariance with β_k^0. When there exist a correct measurement, the second term accounts for this by weighting the updated covariance $\mathbf{P}_k^c$ with $1-\beta_k^0$. $\tilde{\mathbf{P}}_k$ is added to include the measurement origin uncertainty because we do not know which of the validated measurements is correct. $\tilde{\mathbf{P}}_k$ is positive semidefinite, therefore always adding more (or no) uncertainty to the system.

Because of the relatively simple state-estimation scheme, PDAFs computational cost is only about 50% higher than the ordinary Kalman Filter [4]. In multi-tracking scenarios, the PDAF can be used by running one filter for each target. This becomes problematic if some of the targets get in close vicinity of each other. Then the risk of track loss increases and track swapping may occur. A particularly dangerous situation where both tracks is between the two targets may also occur. These examples are called track coalescence and is a well-known problem for these kinds of methods.

4.5 JPDAF

The multi-target extension of the PDAF is introduced with the Joint Probabilistic Data Association Filter (JPDAF) [24]. The main difference compared to the PDAF lies in the calculation of the measurement-to-target association probabilities. Where the PDAF models all incorrect measurements as random interference with uniform spatial distribution, JPDAF allows a validated measurement to originate from another target. In tracking situations where a neighboring target's measurement falls in the validation region of another target over several time stamps, and therefore acts as a persistent interference, the performance of the PDAF significantly degrades [4].

In addition of assuming a known number of established tracks in the surveillance region, the JPDAF relies on some of the assumptions from the PDAF: The past system information

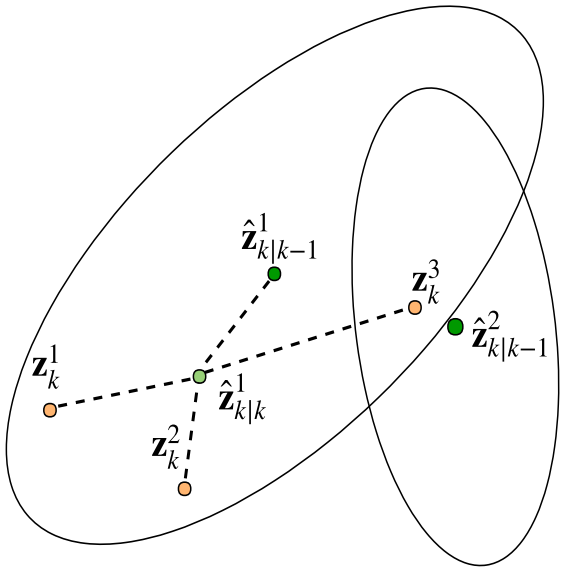


Figure 4.4: JPDAF state estimation update where the new position estimate of target 1 $\hat{\mathbf{z}}^1_{k|k}$ is weighted based on the distance from the validated measurements $\mathbf{z}^i_k$ to the predicted position $\hat{\mathbf{z}}^1_{k|k-1}$ and the possibility of them originating from another target. Notice that the closest measurement $\mathbf{z}^3_k$ is weighted significantly less than the other measurements. This is because this measurement is much closer to another target estimate $\hat{\mathbf{z}}^2_{k|k-1}$.

is summarized by an approximate sufficient statistics, consisting of a Gaussian distributed mean along with covariances for each target, and each target is modeled by (4.1) - (4.3).

The key element in the JPDAF is the evaluation of the conditional probabilities of the association hypothesis. An association hypothesis, $\boldsymbol{a}_k$, can be defined as a vector $\boldsymbol{a}_k = [\boldsymbol{a}_k(1), \boldsymbol{a}_k(2), ..., \boldsymbol{a}_k(n)]^T$ such that

$$\boldsymbol{a}_k(t) = \begin{cases} j & \text{if measurement } j \text{ is claimed by target } t \\ 0 & \text{if no measurement is claimed by target } t \end{cases} \tag{4.19}$$

where $j = 1, .., m_k$ and $t = 0, 1, ..., n$ is the number of measurements and targets, respectively.

Based on the modeling of the false measurements, the equation of the joint association probabilities ($P\{\boldsymbol{a}_k | Z_{1:k}\}$, where $Z_{1:k}$ is the the cumulative set of observations up until time k) can be found. (This calculation is not shown here as a more general version of the

same equation is presented later on for the JIPDAF.)

The state estimation is done in a similar manner as in the PDAF, where it is assumed that the states of the targets conditioned on the past observations are mutually independent. To calculate the targets' state estimates, the marginal association probabilities are needed. These are obtained from the joint association probabilities by summing over all joint events in which the marginal event of interest occurs. The remaining estimation is done as in the PDAF, with the equations (4.11)-(4.18).

4.6 JIPDAF

The Joint Integrated Probabilistic Data Association Filter (JIPDAF) [46] is a multi-target tracking filter which includes a track quality measure - a direct generalization of the IPDAF and the JPDAF. Being based on mostly the same concepts as in the PDAF, a high number of similarities is seen. The kinematic prediction and update steps (section 4.4.1) are, for a given target, the same as in the PDAF. For each target, all prior measurement information are approximated by a single Gaussian which defines a validation area around the predicted measurement. This is then used to determine if a measurement is to be identified with a target or not. The main difference is the calculation of the association probabilities (as in the JPDAF), and the addition of track existence probabilities (from the IPDAF).

The following derivation is based on the work from [10, 46, 6].

4.6.1 Assumptions

The number of existing targets is assumed to be n_{k-1} at time step $k-1$, and we assume that no target births occur. At each time step, the set of new measurements is denoted Z_k, while the cumulative set of measurements up to and including time step k is $Z_{1:k}$. In addition, each track is described by a state estimate $\hat{\mathbf{x}}_k^t$, an error covariance matrix $\mathbf{P}_k^t$ and a probability of track existence. The number of false measurements are assumed to be Poisson distributed with intensity λ_P, and each existing target survive from time step $k-1$ to k with probability P_S.

4.6.2 Target existence and visibility modeling

Target existence propagation is modeled as a Markov chain. Musicki and Evans [46] uses a Markov model which distinguishes three possibilities: the target exists and is detectable (with a known probability of detection, P_D), the target exists but is not detectable and the target does not exist.

In this thesis, a similar approach for target existence is used, but track existence and track visibility (detectability) are separated and assumed decoupled. This means that death probability is assumed to not depend on the prior visibility state. If we denote the predicted existence probability by $\varepsilon^t_{k|k-1}$ and the predicted visibility probability by $\eta^t_{k|k-1}$, the prediction equations are given by

$$\begin{bmatrix} \varepsilon^t_{k|k-1} \\ 1-\varepsilon^t_{k|k-1} \end{bmatrix} = \begin{bmatrix} p^\varepsilon_{11} & p^\varepsilon_{12} \\ p^\varepsilon_{21} & p^\varepsilon_{22} \end{bmatrix} \begin{bmatrix} \varepsilon^t_{k-1} \\ 1-\varepsilon^t_{k-1} \end{bmatrix} \tag{4.20}$$

$$\begin{bmatrix} \eta^t_{k|k-1} \\ 1-\eta^t_{k|k-1} \end{bmatrix} = \begin{bmatrix} p^\eta_{11} & p^\eta_{12} \\ p^\eta_{21} & p^\eta_{22} \end{bmatrix} \begin{bmatrix} \eta^t_{k-1} \\ 1-\eta^t_{k-1} \end{bmatrix}. \tag{4.21}$$

For both models, the Markov chain coefficients must satisfy

$$\begin{aligned} p^\varepsilon_{11} + p^\varepsilon_{12} &= p^\varepsilon_{21} + p^\varepsilon_{22} \\ &= p^\eta_{11} + p^\eta_{12} \\ &= p^\eta_{21} + p^\eta_{22} = 1 \end{aligned} \tag{4.22}$$

By assuming no-target births, the multi-target Markov model is given entirely given by the survival of existing targets. The death probability for a given target is assumed to be $P_S = 0.25$.

4.6.3 JIPDAF data association

As for the JPDAF, the JIPDAF relies on the calculation of the joint association probabilities, $P\{a_k|Z_{1:k}\}$. Using Bayes' formula, the joint association probabilities can be written

$$\begin{aligned} P\{\boldsymbol{a}_k|Z_{1:k}\} =& P\{\boldsymbol{a}_k|Z_k, m_k, Z_{1:k-1}\} \propto p[Z_k|\boldsymbol{a}_k, m_k, Z_{1:k-1}]P\{\boldsymbol{a}_k|Z_{1:k-1}, m_k\} \\ =& p[Z_k|\boldsymbol{a}_k, m_k, Z_{1:k-1}]P\{\boldsymbol{a}_k|m_k\}. \end{aligned} \tag{4.23}$$

The next step is to find concrete expressions for each term in (4.23).

The likelihood of measurement function

Let φ denote the number of false measurements in event $\boldsymbol{a}_k$. Following along the lines of [10], the first term of (4.23) can be written as

$$p[Z_k|\boldsymbol{a}_k, m_k, Z_{1:k-1}] = \frac{1}{V^\varphi} \prod_{t:\boldsymbol{a}_k(t)>0} \left[\lambda l^{t,\boldsymbol{a}_k(t)}\right] \tag{4.24}$$

where V is the volume of the surveillance region and $l^{t,\boldsymbol{a}_k(t)}$ is the likelihood ratio defined by

$$l^{t,\boldsymbol{a}_k(t)} = \frac{1}{\lambda}\mathcal{N}[\mathbf{z}_k^{\boldsymbol{a}_k(t)}; \mathbf{H}\hat{\mathbf{x}}_k^{t,\boldsymbol{a}_k(t)}, \mathbf{H}\mathbf{P}_k^{t,\boldsymbol{a}_k(t)}\mathbf{H}^T + \mathbf{R}]. \tag{4.25}$$

The posterior event-conditional states and covariances used to define the likelihood ratio, are given by

$$\hat{\mathbf{x}}_k^{t,\boldsymbol{a}_k(t)} = \begin{cases} \hat{\mathbf{x}}_{k|k-1}^t & \text{if } \boldsymbol{a}_k(t) = 0 \\ \hat{\mathbf{x}}_{k|k-1}^t + \mathbf{K}_k^t(\mathbf{z}_k^{\boldsymbol{a}_k(t)} - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}^t) & \text{if } \boldsymbol{a}_k(t) > 0 \end{cases} \tag{4.26}$$

$$\mathbf{P}_k^{t,\boldsymbol{a}_k(t)} = \begin{cases} \mathbf{P}_{k|k-1}^t & \text{if } \boldsymbol{a}_k(t) = 0 \\ (\mathbf{I} - \mathbf{K}_k^t\mathbf{H})\mathbf{P}_{k|k-1}^t + & \text{if } \boldsymbol{a}_k(t) > 0 \end{cases} \tag{4.27}$$

where $\mathbf{K}_k^t = \mathbf{P}_{k|k-1}^t\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}^t\mathbf{H}^T + \mathbf{R})^{-1}$ is the Kalman gain for track number t.

The prior joint association probabilities

Given the hypothesis $\boldsymbol{a}_k$, the number of false measurements φ and the number of assumed detected targets $m_k - \varphi$ are given. Defining a new vector $\delta(\boldsymbol{a}_k)$, used to indicate which targets have an associated measurement corresponding to event $\boldsymbol{a}_k$, the second term of equation (4.23) can be rewritten as

$$P\{\boldsymbol{a}_k|m_k\} = P\{\boldsymbol{a}_k|\delta(\boldsymbol{a}_k), m_k\}P\{m_k|\delta(\boldsymbol{a}_k)\}P\{\delta(\boldsymbol{a}_k)\}. \tag{4.28}$$

Continuing on, the unconditional probability of the combined events in $\delta(\boldsymbol{a}_k)$, is based entirely on the individual detection events. This yields

$$P\{\delta(\boldsymbol{a}_k)\} = \prod_{t:\boldsymbol{a}_k(t)=0}(1 - \varepsilon_{k|k-1}^t P_D^t \eta_{k|k-1}^t) \prod_{t:\boldsymbol{a}_k(t)>0} \varepsilon_{k|k-1}^t P_D^t \eta_{k|k-1}^t \tag{4.29}$$

where $\varepsilon_{k|k-1}^t$ and $\eta_{k|k-1}^t$ are the prior existence and visibility probability of target t, respectively. Assuming that the number of false measurements are Poission distributed with the rate $\lambda_P V$, we have

$$P\{m_k|\delta(\boldsymbol{a}_k)\} = e^{V\lambda_P}\frac{(V\lambda_P)^\varphi}{\varphi!}. \tag{4.30}$$

The second term in (4.28) is obtained from the following reasoning: Given the targets-with-measurement vector $\delta(\boldsymbol{a}_k)$, we have $\frac{m_k!}{\varphi!}$ equally likely permutations of the measurements assumed to be target originated, where each permutation will lead to a different association hypothesis ($\boldsymbol{a}_k$). Therefore,

$$P\{\boldsymbol{a}_k|\delta(\boldsymbol{a}_k), m_k\} = \frac{\varphi!}{m_k!} \tag{4.31}$$

The posterior joint association probabilities

Combining the equations (4.29)-(4.31) to one equation and multiplying it by the likelihood (4.24), the posterior probabilities of the association hypothesis are given by

$$P\{\boldsymbol{a}_k|Z_{1:k}\} \propto \prod_{t:\boldsymbol{a}_k(t)=0} (1-\varepsilon^t_{k|k-1}P_D\eta^t_{k|k-1}) \prod_{t:\boldsymbol{a}_k(t)>0} \varepsilon^t_{k|k-1}P_D\eta^t_{k|k-1}l^{t,\boldsymbol{a}_k(t)} \quad (4.32)$$

where it is seen that the V terms have been canceled and the λ_P terms have been moved into the proportionality constant. This can be done as the power of λ_P is always m_k [6].

4.6.4 Marginal association probabilities

Based on the same principle as in the PDAF, we want to merge different hypothesis such that one is left with one Gaussian for each track. Moreover, we want to find the marginal probabilities, i. e. the probabilities for measurement j originating from target t, given by

$$\beta^{t,j}_k \propto \sum_{\boldsymbol{a}_k \text{ s.t. } \boldsymbol{a}_k(t)=j} P\{\boldsymbol{a}_k|Z_{1:k}\}. \quad (4.33)$$

Once these have been calculated, the posterior Gaussian approximation for each track can be found by

$$p^t_k(\mathbf{x}^t_k) \propto \sum_{j=0}^{m_k} \beta^{t,j}_k \mathcal{N}(\mathbf{x}^t_k; \hat{\mathbf{x}}^{t,j}_k, \mathbf{P}^{t,j}_k) \approx \mathcal{N}(\mathbf{x}^t_k; \hat{\mathbf{x}}^t_k, \mathbf{P}^t_k). \quad (4.34)$$

The merged state $\hat{\mathbf{x}}^t_k$ and covariance $\mathbf{P}^t_k$ are found by using the state update equations from section 4.4.1.

4.6.5 Posterior existence and visibility probabilities

The last step of the tracking scheme is to update each tracks existence and visibility probabilities. First, we define the quantities

$$\varepsilon^{t,0}_k = \frac{(1-P_D\eta^t_{k|k-1})}{1-\varepsilon^t_{k|k-1}P_D\eta^t_{k|k-1}}\varepsilon^t_{k|k-1} \quad (4.35)$$

$$\eta^{t,0}_k = \frac{(1-P_D)}{1-P_D\eta^t_{k|k-1}}\eta^t_{k|k-1} \quad (4.36)$$

representing the posterior existence and visibility probabilities given that no measurement originates from target t. The marginal existence and visibility probabilities can then be

calculated by

$$\varepsilon_k^t = \beta_k^{t,0}\varepsilon_k^{t,0} + \sum_{j=1}^{m_k}\beta_k^{t,j} \tag{4.37}$$

$$\eta_k^t = \frac{1}{\varepsilon_k^t}\left(\beta_k^{t,0}\varepsilon_k^{t,0}\eta_k^{t,0} + \sum_{j=1}^{m_k}\beta_k^{t,j}\right). \tag{4.38}$$

The derivation of (4.37) and (4.38) can be found in [10].

4.6.6 Efficiency improvements

As the number of association hypothesis grows exponentially with respect to the number of measurements and tracks, the calculation of the marginal association probabilities (4.33) may be too expensive to calculate completely. If the calculation is based on, for example, 2 tracks and 2 measurements, the calculation is perfectly feasible. However, if the number of tracks and/or measurements are increased by only a few numbers, approximations is needed. To mitigate this problem, several techniques can be used.

The first efficiency improvement is to use validation gating, the same concept as in the PDAF, section 4.4.1. This ensures that, for each track, only the most likely measurements are included in the association calculation. Another similar approach for multi-target methods, is to cluster all tracks which shares associated measurements, together (Figure 4.5). As the number of tracks and measurements are reduced, the resulting number of association hypothesis is significantly reduced.

Several other techniques for complexity mitigation exist [13, 32, 45], where we will take a deeper look at Murty's M-best method [13] as this is used in the implemented tracking system.

Murty's M-best method

Murty's M-best method tries to generate the M-most probable association hypothesis. This is done by utilizing another method, called Auction, which is used for solving the more general optimal assignment problem. The main concept in the action method is that we have a set of customers bidding for a larger set of items. The algorithm terminates when all customers are satisfied with their items.

Continuing with Murty's method, the method maintains a list $\mathcal{L}$ of problem-solution pairs, $\mathcal{A}_M = (\mathcal{A}_p, \mathcal{A}_s)$, where the problem $\mathcal{A}_p$ is a score matrix and the solution $\mathcal{A}_s$ represents an assignment of measurements to tracks. The values of the score matrix $\mathcal{A}_p$ is based on the association probabilities from the hypothesis represented by $\mathcal{A}_s$. The idea is to

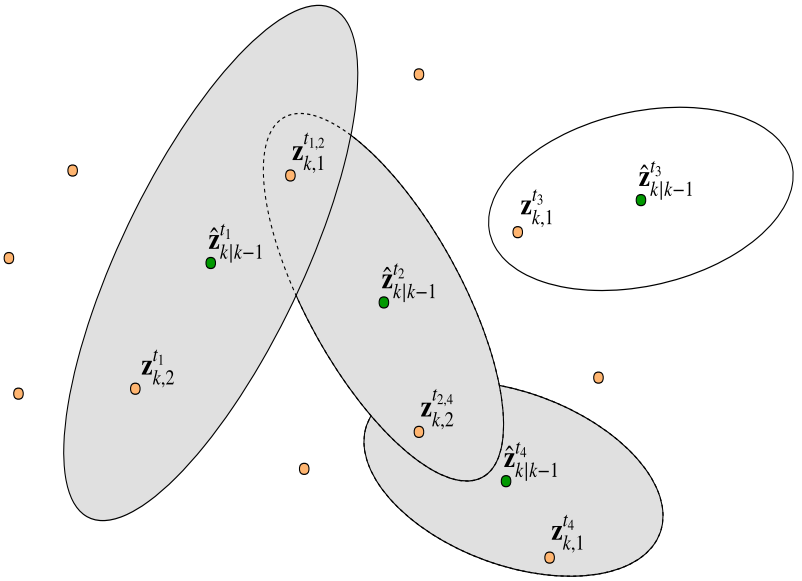


Figure 4.5: All tracks which have associated measurements in common, are connected to one cluster. This is seen as the grey colored union of the given validation regions.

first solve the original assignment problem by using the Auction method. The method then takes the current best problem-solution pair and removes the current assignments before recalculating this new modified assignment problem. This is used to gradually move towards a better assignment solution. The method terminates when a given number of solutions have been found or there are no problems left in $\mathcal{L}$.

4.7 The JIPDAF implementation

In addition to the JIPDAF theory given in section 4.6, the implemented JIPDAF tracking system needs some additional functionality in order to work. The next section describes the method used to initiate tracks, followed by a section describing how radar measurements are emulated. In the last section, all tracking parameters are given.

4.7.1 Track initiation

The track initiation procedure used in this thesis is a simple, yet effective approach. At each time steps, all measurements not associated with any target are stored to the next time step. At the next time step, for each measurement from the new set of non-associated measurements, the distance to all the previous non-associated measurements are calculated. If the distance for any measurement are closer than some specified max range, a new track is formed with a set of predefined initiation parameters.

Denoting the set of non-associated measurements, ϱ_{k-1}, we have

$$|\varrho_k^d - \varrho_{k-1}^b| < Tv_{max} \qquad d = 1, ..., \varphi_k, b = 1, ..., \varphi_{k-1} \tag{4.39}$$

where v_{max} is chosen to be larger than the maximum velocity of the vessel and $T = t_k - t_{k-1}$ is the tracker sample time. If equation (4.39) is satisfied, the associated measurements are used to generate the initial values (covariances etc.) for a new track. For each new track, the existence and observability probabilities are set to $\varepsilon_{init}^t = 0.5$ and $\eta_{init}^t = 0.8$.

The implemented tracking system does not directly distinguish between a tentative track and a confirmed track. All existing tracks are treated the same inside the tracking system, but for a track to be delivered to the COLAV algorithm, the existence probability needs to exceed 0.5. This is to ensure that only high probability tracks are treated as true obstacles. Setting this threshold to be higher than 0.5 is also to make sure newly initiated tracks are not included.

4.7.2 Radar measurement emulation

The radar measurement emulation is done by adding random measurements to the measurement vector from a simulated radar grid, shown in Figure 4.6. The grid resolution is set to 1000×512, where 1000 is the number of sections in the radial direction and 512 is the number of sections in azimuth.

As the simulator environment doesn't include support for a Poisson distribution, the number of clutter measurement is drawn from a uniform distribution with a given average. The average number of clutter measurements is calculated based on the probability that a false measurement appears in a given cell, denoted P_{FA}. By varying P_{FA} and the detection probability P_D, different tracking conditions can be emulated.

4.7.3 Tracking parameters

As the implemented tracking system utilizes the motion and measurement model in section 4.3, methods from the PDAF in subsection 4.4.1 in addition to JIPDAF theory from

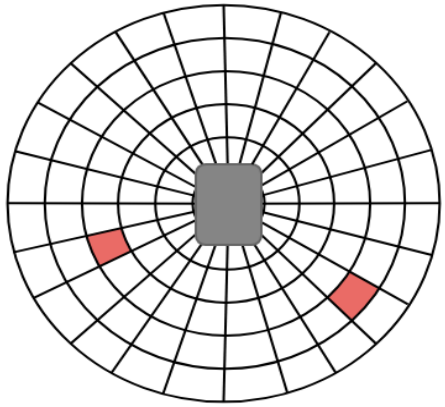


Figure 4.6: A radar grid around the vessel is created, where a random number of measurement are detected.

section 4.6, a number of parameters has to be defined. These are listed in Table 4.1.

Parameter	Value
σ_a	1.5
σ_r	1
P_G	0.99
p_{11}^{ε}	0.99
p_{12}^{ε}	0.01
p_{21}^{ε}	0
p_{22}^{ε}	1
p_{11}^{η}	0.9
p_{12}^{η}	0.1
p_{21}^{η}	0.48
p_{22}^{η}	0.52

Table 4.1: Parameters used in the implemented tracking system.

4.8 Other tracking methods

4.8.1 Multi Hypothesis Tracker

The Multi Hypothesis Tracker (MHT) method uses a completely different procedure for state-update and measurement handling; alternative hypothesis are maintained and propagated further in time. This makes it possible to postpone which measurement to choose if

the information available at that time is uncertain or missing. So, whenever observation-to-track conflict situations occur, a new data association hypothesis is formed. Instead of merging the hypothesis as in PDAF/JPDAF, the hypothesis is propagated into the future in anticipation that upcoming data will resolve the uncertainty [8]. For each hypothesis, a probability is given based on how well the measurements fits the tracks which are contained in the hypothesis.

The logic which the MHT relies on, namely having several hypothesis at the same time, is both its weakness and strength. The number of hypothesis (and tracks within this hypothesis) may see an exponential growth because of the addition and propagation of hypothesis. Techniques like clustering, track pruning and merging have been developed to mitigate this problem.

Chapter 5

Simulation setup

This chapter presents the modules and methods used to simulate and evaluate the COLAV system. The simulator environment is first presented, followed by a description of how the scenarios are generated. The final section introduces the evaluation metrics used to quantify COLREGS compliance. Figure 5.1 shows the overall modules and the chronological order of the testing process.

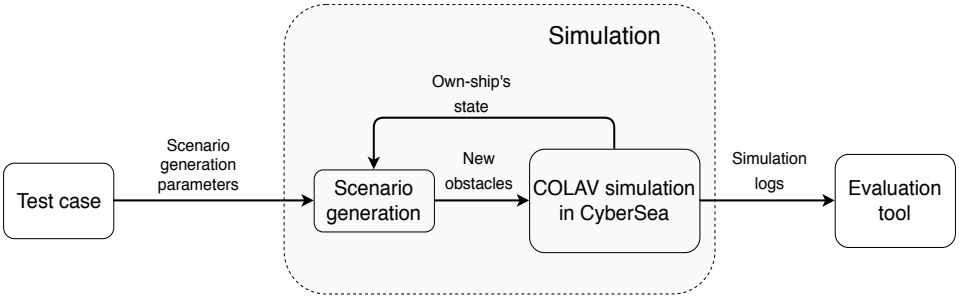


Figure 5.1: Overview of the testing procedure. The test case decides what type of scenarios are to be simulated, which is defined by the scenario generation parameters. During the simulation process, the scenario generation module generates scenarios based on the initial scenario generation parameters given by the test case and the own-ship's state. After finishing the simulation process, the simulation logs are delivered to the evaluation tool for further analysis. The analysis is based on a number of evaluation metrics used to determine COLREGS compliance.

5.1 Simulation environment

5.1.1 HIL and SIL testing

The testing of the COLAV system is done in the framework of a simulator provided by DNV GL, called CyberSea. CyberSea enables what is called Hardware-In-the-Loop (HIL) testing. HIL testing means that there is something physical connected to the real-time simulation [12]. When performing HIL-testing in the maritime sector, the physical vessel is often replaced by a HIL-simulator. The HIL-simulator emulates the vessels motion in response to simulated environmental conditions, failure modes injected by the user etc., while the control software typically runs on a seperate hardware module [36]. HIL testing is not directly used in this thesis, instead the testing technique called Software-In-the-Loop is used.

In general, SIL testing is used to test the software of a system. All required hardware modules are emulated - making it easier to control the process execution [12]. In addition, not depending on physical equipment, sometimes heavy or impossible to ship to the same location, makes the testing process much more flexible; all testing may be done at one location, maybe fully virtual in a cloud-based solution. One of the main challenges when it comes to SIL-testing is the accuracy of the emulated hardware-modules. To be a reliable testing method, the model accuracy has to be as close to the real-life modules as possible.

As seen by the gray colored box in Figure 5.2, the distinction between the emulated hardware and the rest of the system is evident. The main point is that the control system should not be able to tell the difference between the emulated hardware and a real marine vessel. In the simulator we find accurate ship dynamic modelling together with actuators and sensors.

In general, simulation compared to real-life testing is much more effective. The simulation can be much faster than real-time, quality control is easier to introduce, specific tests cases are easier to provoke, time and money savings etc. Also, more dangerous situations can be tested without damaging equipment or people.

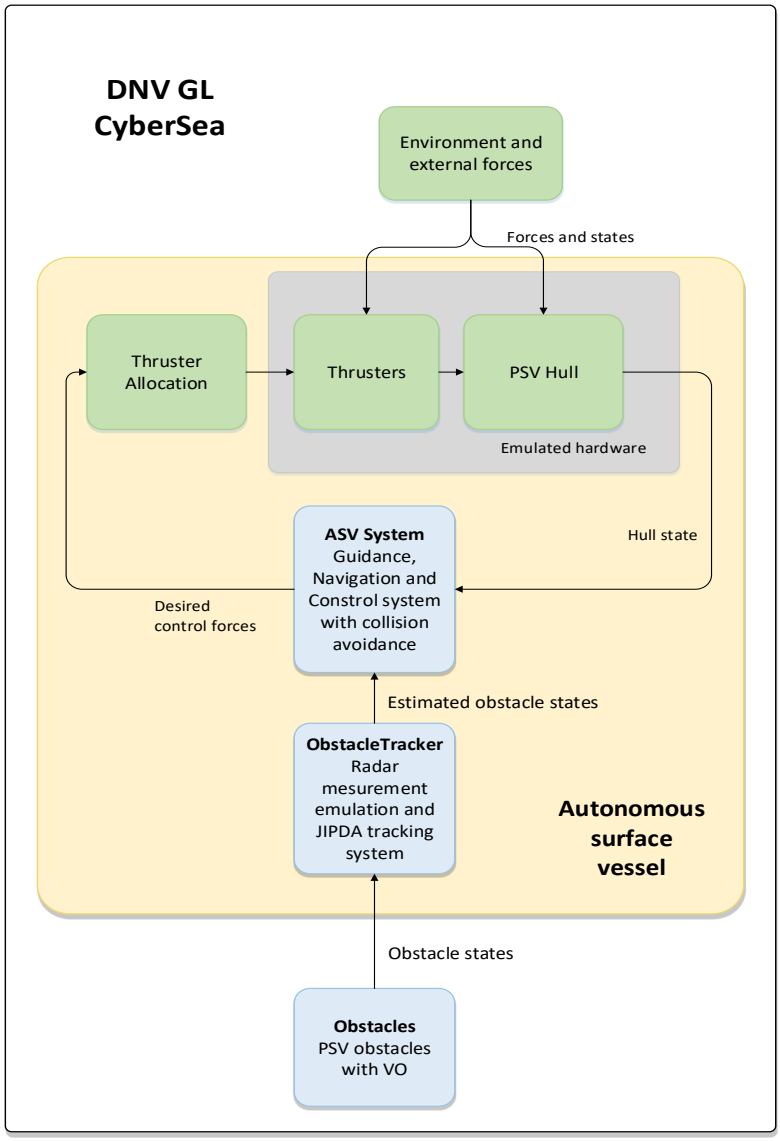


Figure 5.2: Overview of the modules in the CyberSea simulator. The green boxes represents modules provided by DNV GL, while the ”blue” modules have been implemented by Minne and in this thesis. The figure is based on a figure from [44].

5.1.2 CyberSea

An overview of the modules used in the CyberSea simulator are presented in Figure 5.2. The ”PSV Hull” module models the dynamics of the hull of a PSV, the ”Thrusters” module models the thruster dynamics, the ”Thruster Allocation” module implements the distribution of generalized forces to the thrusters, while the environmental forces such as waves, current and wind are modeled by the last green colored module.

The blue colored boxes are additional modules implemented to enable testing of COLAV systems. The yellow colored box defines the entire emulated ASV (the own-ship), while the boxes ”outside” of this are assumed separated from the ASV system. It is important to notice that the module and signal information shown in Figure 5.2 is not complete, i.e. only the most important information are presented.

5.1.3 Vessel models

In order to make the simulation process more realistic, two vessels with significantly different maneuverability are used as obstacle vessels. A PSV will serve as the large slowly state changing vessel, while the NTNU-boat Gunnerus will provide the simulation with a smaller faster moving vessel with higher maneuverability.

The PSV modeled in CyberSea is used as the System Under Test (SUT) in this thesis. This model is an accurate 6 DOF model with thruster allocation. The obstacles are modeled by the simpler 3 DOF maneuvering model described in section 2.2.1. Thus, the obstacle PSVs are not modeled with the same accuracy as the own-ship PSV. This is because the 3 DOF model will give sufficient behavior accuracy for this application.

The 3 DOF model relies on maneuvering information such as hydrodynamic coefficients etc. This information is not listed explicitly as the values of the different parameters may be hard to interpret. If the reader find it crucial to know, all PSV information can be found in section 4.2 in [44]. The data for Gunnerus are based on a number of Planar Motion Mechanism (PMM) tests performed prior to Gunnerus' refit [50, 30].

Both vessels use the same low level controllers and ILOS guidance method, but the controller gains are adjusted for the specific vessel. This is due to the fact that Gunnerus doesn't have the same power system as the PSV. The controller gains are listed in subsection 2.2.2.

Figure 5.3: A PSV similar to the one used in this thesis. Image source: `https://www.flickr.com/photos/zeesenboot/10317626005/`.

Platform supply vessel

The first vessel is a Platform supply vessel (PSV), normally used to supply offshore oil and gas platforms with equipment and personnel. The PSV weights about 15500 tons, has a length of 116 meters and is 25 meters wide. A similar vessel as the PSV is shown in Figure 5.3.

Gunnerus

The second vessel is NTNU's research vessel R/V Gunnerus (Figure 5.4). Gunnerus is equipped with a dynamic positioning system and the latest technology for a variety of research activities. It is also used for educational purposes for the marine courses at NTNU, as it contains both a lecture room and a set of useful laboratories. Gunnerus has a length of 31 meters, a width of about 10 meters and weights 75 tons.

5.2 Testing methodology

5.2.1 Iterative geometric testing

Woerner [63] presented a testing scheme called Iterative Geometric Testing (IGT), where the purpose was to make a testing scheme that challenges a COLAV system to a wide range of collision scenarios. The COLAV system was to be exposed to multi-vehicle, multi-

Figure 5.4: R/V Gunnerus starboard side view. Image source: `https://www.ntnu.edu/oceans/gunnerus`.

rule and conflicting scenarios. The main idea is to set an initial configuration (speed and heading) of the obstacles while varying the configuration of the own-ship. After defining the time to collision, a common collision point is calculated - a point where all ships will collide if no action to avoid collision are taken. To add some sort of randomness to the scenarios, uniformly or normally distributed noise is added to the initial obstacle configuration. A large number of collision scenarios are generated by keeping constant obstacle configuration while alter the initial configuration of the own-ship.

Minne [44] proposed a testing scheme based almost entirely on the IGT presented by Woerner, but with the difference that instead of altering the own-ships initial configuration, this was kept constant while varying the obstacles configuration. Except for this distinction, the methods basically works the same way.

5.2.2 New testing scheme

The testing schemes presented in the previous section have shown to give satisfactory results, but there seems to be some potential regarding the connection to real-life scenarios. The previous work by [44] only considered situations with up to 3 vessels approaching the same point. While additional complexity due to more vessels approaching each other simultaneously is unlikely, situations where a larger number of vessels interact in rapid succession are more common in the real world. Based on this argument and by using the concepts from the IGT scheme, a new testing scheme is presented.

The main new feature is to not initialize the own-ship to the same state in every scenario. Having a situation where the own-ship needs to alter the course to avoid collision (or to

comply with the COLREGS), the resulting own-ship state after completing this scenario may be in great distance from the desired state. If then there's a new obstacle vessel nearby, we have a new collision situation where the own-ship is not in a "steady state". By implementing this concept to the testing scheme, a more realistic course of events will happen. Figure 5.5 illustrates the main concepts of the proposed testing scheme.

Scenario generation algorithm

At first, we need to define the number of of succeeding scenarios, denoted n_S, and the duration time for each scenario, denoted t_S. The own-ship is assumed to be in transit, heading directly north (this assumption could be relaxed to use the desired heading). For each scenario, the following steps are performed:

1. Calculate the future point where the own-ship is to reach the path given by the way-points, denoted $\boldsymbol{P}_{est} = [x_{est}, y_{est}]^T$. This is done by assuming straight line motion from the position of the own-ship at the time it crosses from the previous scenario to the next ($\boldsymbol{P}_0 = [x_0, y_0]^T$). This is seen as the crossing of the dotted line in Figure 5.5. Since the own-ship uses a ILOS controller to follow the path, the estimated point $\boldsymbol{P}_{est}$ will not lie at the exact position the own-ship would've reached without any vessel interaction, but it will be so close that avoiding maneuvers has to be taken in order to avoid collision. $\boldsymbol{P}_{est}$ is assumed to lie on the north axis ($y_{est} = 0$), and the calculation of x_{est} is given by

$$\boldsymbol{x}_{est} = \begin{cases} x_0 + t_{ttp} u_0, & \text{if own-ship is close to path or heading away from path} \\ x_0 + \frac{|y_0|}{v_0} u_0, & \text{otherwise} \end{cases} \tag{5.1}$$

   where u_0 and v_0 is the speed in the north and east direction at the time the scenario begins, respectively. The parameter deciding the time to path (t_{ttp}) is set to a value based on the time of the scenario (t_S).

2. Pick the number of obstacle vessels to be included in the scenario, denoted n_O. This could be set to a specific value or chosen randomly.

3. Given the estimated future path position $\boldsymbol{P}_{est}$, a collision point for each obstacle $\boldsymbol{P}_{col} = [x_{col}, y_{col}]^T$ is found by adding Gaussian noise to $\boldsymbol{P}_{est}$. This may be done in both spatial directions, but in this thesis we would like a collision point at the north axis ($y_{col} = 0$), which result in

$$x_{col} = x_{est} + w_x \tag{5.2}$$

   where w_x is the Gaussian noise described by the pdf $p(w_x) = \mathcal{N}(w_x; 0, \sigma^2_{w_x})$.

4. For each obstacle, a random time to collision is picked from a desired set $t_{ttc} \in [t^{min}_{ttc}, t^{max}_{ttc}]$. The set of desired time to collision values is chosen based on the time of the scenario t_S.

5. For each obstacle, pick a random initial course $\chi_{init} \in [0°, 360°)$ and an initial speed $v_{init} \in [v_{init}^{min}, v_{init}^{max}]$.

6. Given the initial course and heading of the obstacles, an initial starting point $\boldsymbol{P}_{init} = [x_{init}, y_{init}]^T$ of all obstacles are calculated. For each obstacle, the starting point is calculated by

$$x_{init} = x_{col} - t_{ttc} v_{init} \cos(\chi_{init}) \tag{5.3}$$
$$y_{init} = y_{col} - t_{ttc} v_{init} \sin(\chi_{init}) \tag{5.4}$$

7. Determine obstacle types. Two different types of vessels are used in this thesis. All obstacle can be the same vessel type or the obstacle type can be chosen randomly.

8. When having multiple obstacle vessels in a scenario, the likelihood of them being in close vicinity is present. Thus, if an obstacle vessel is defined to be too close to another vessel, the obstacle vessel is removed from the scenario.

By making the the number of scenarios a large value, this testing scheme will provide a wide range of collision situations with different active COLREGS rules. One of the drawbacks is that because of the extensive use of noise (varying parameters) there's no guarantee that all desired collision situation types are inspected thoroughly, but if the total number of obstacles in all scenarios is large, the likelihood of this happening is small.

Minne [44] states that his testing scheme easily can be altered to generate this kind of successive interaction. The problem with his proposition is that the collision point is calculated beforehand, making it not reliable due to the non-predictable behavior of the collision avoidance methods. When predicting the collision position based on real-time state information (as the testing scheme presented here), the likelihood of the estimated collision point being a real collision point increases compared to a testing scheme where the collision point is calculated in advance.

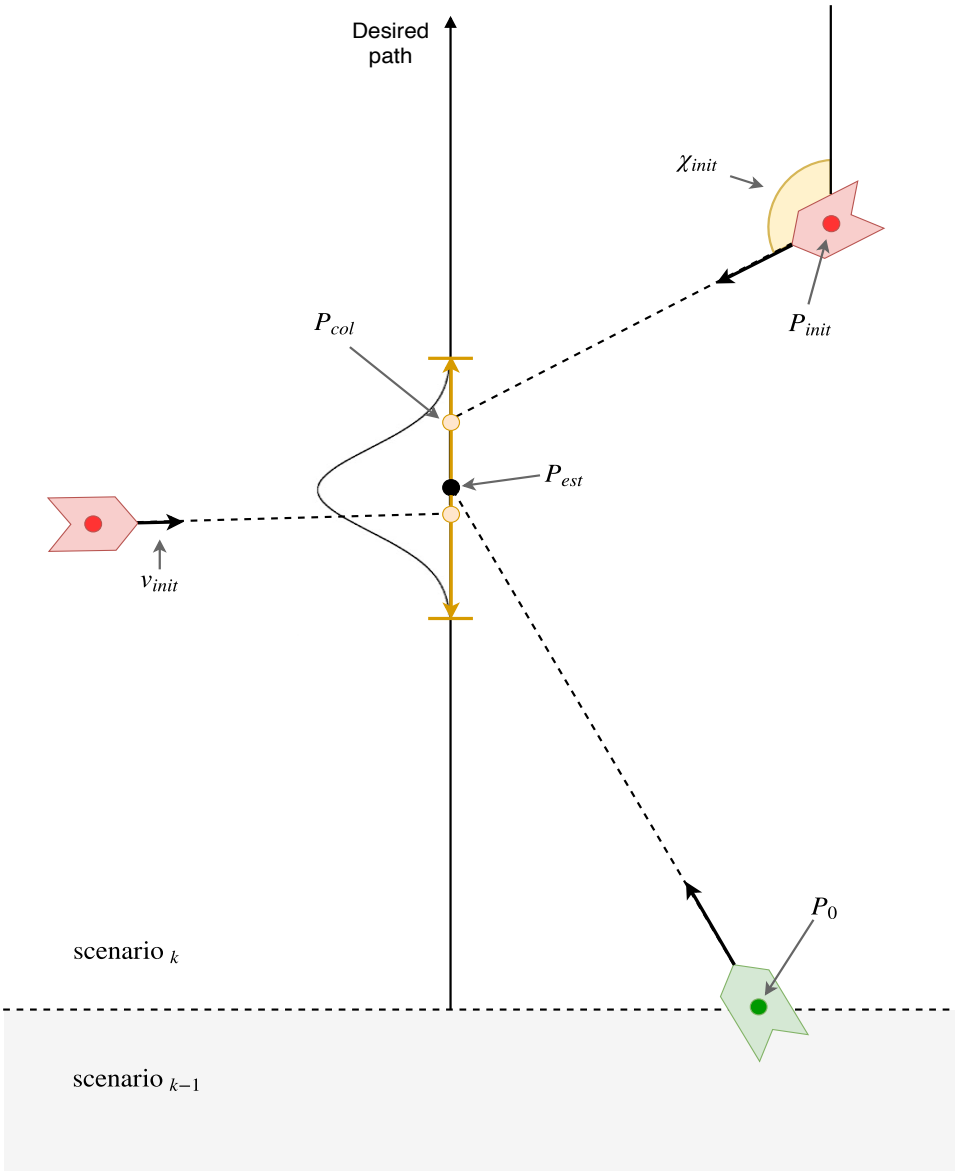


Figure 5.5: Illustration showing the most important concepts of the testing scheme. The estimated future own-ship position P_{est} is calculated based on the (green colored) own-ships state at the time the scenario starts. Based on P_{est}, 2 collision positions are found for the two (red colored) obstacle vessels. The obstacles are given a random initial speed and heading, and their initial position are calculated based on this and a random time to collision. The rotated bell-shaped form on the left side of the collision points is used to illustrate the variance in distance between the estimated P_{est} and the possible collision points.

5.3 Maritime COLAV evaluation

The evaluation of a COLAV system may in general include the aspects of safety, efficiency and protocol compliance. Protocol compliance often includes the safety aspect, but additional evaluation may be required. The efficiency aspect refers to track deviation distance, transit time, fuel consumption etc. This however, is not a part of this thesis. The focus will lie on protocol compliance, more specifically, COLREGS compliance.

This section gives an overview of the metrics used to determine COLREGS compliance and is heavily based on [44, 31].

5.3.1 Evaluation overview

The evaluation process of a scenario is divided into three steps. First, determine which rules apply for each encounter in the scenario. The next step is to evaluate the encounters by the use of the appropriate metrics for the rules found in step one. At last, the total score of the scenario is given as the lowest encounter score for each scenario.

Some additional evaluation information is listed here:

- Only the COLREGS rules specified in section 2.3 are covered by the metrics.
- All metrics give a score between 0 and 1. $\mathcal{S}$ denotes a score, where 0 is the lowest score and 1 the highest score. $\mathcal{P}$ denotes a penalty, where 0 is no penalty and 1 is the highest penalty.
- A scenario or a situation both describe the same. In this report that is when the own-ship approaches one or more obstacles and has to take action. An encounter is defined as the interaction between the own-ship and one obstacle.
- In an encounter, the stand-on vessel should keep her course and speed, while the give-way vessel should initiate a maneuver to avoid collision.
- In general, the evaluation of an encounter is done based on the own-ships maneuvering from the time the obstacle vessel is detectable (observable) for the own-ship (t_{detect}) to the time of the closest point of approach (t_{CPA}). The exceptions are clearly mentioned.

In the first step of the evaluation process, the encounters are categorized according to Table 5.1.

Encounter type	Description
Rule 13 + 16	Overtaking, own-ship as give-way
Rule 13 + 17	Overtaking, own-ship as stand-on
Rule 14	Head-on
Rule 15 + 16	Crossing, own-ship as give-way
Rule 15 + 17	Crossing, own-ship as stand-on

Table 5.1: Encounter categorization.

5.3.2 Evaluation metrics

This subsection presents the evaluation metrics used to evaluate the encounters from the simulations. Most of the metrics are the exact same as in chapter 3 in [44], but some have seen a number of modifications. The most important metrics and the modified ones have been given a more thorough explanation.

Safety metric

The safety metric is used in all encounters, and tries to quantify the safety of an encounter. The metric is a score function given by

$$S_{safety}(r_{CPA}) = \begin{cases} 1, & \text{if } R_{min} \leq r_{CPA} \\ 1 - \gamma_{nm} \frac{R_{min} - r_{CPA}}{R_{min} - R_{nm}} & \text{if } R_{nm} \leq r_{CPA} \leq R_{min} \\ 1 - \gamma_{nm} - \gamma_{col} \frac{R_{nm} - r_{CPA}}{R_{nm} - R_{col}} & \text{if } R_{col} \leq r_{CPA} \leq R_{nm} \\ 0, & \text{else.} \end{cases} \tag{5.5}$$

r_{CPA} is the range to the obstacle at CPA, R_{min} is the minimum acceptable passing distance and R_{col} is a distance at CPA where a collision most likely would have occurred. The parameter values are shown in Table 5.2. A more detailed description of this metric can be found in section 3.2 in [44].

Parameter	Value	Unit
R_{min}	1000	m
R_{nm}	800	m
R_{col}	200	m
γ_{nm}	0.4	
γ_{col}	0.6	

Table 5.2: Parameters used in the safety metric (5.5).

Non-readily apparent course change penalty

For vessels in a collision situation, the maneuvers taken should be readily apparent. This is especially important for the give-way vessel; it should make a significant alteration of course and speed such the stand-on vessel understands her intention. Three penalty metrics are used to detect non-readily apparent movements: one for course, one for speed and one for the maneuver overall. The metric for detecting non-readily apparent course changes is defined by

$$\mathcal{P}_{\Delta\chi_{app}}(\Delta\chi) \triangleq \min\left(0, 1 - \left(\frac{|\Delta\chi|^2}{\Delta\chi_{app}^2}\right)\right) \tag{5.6}$$

where $|\Delta\chi|$ is the absolute course alteration in a maneuver, and $\Delta\chi_{app}$ is a course alteration threshold considered readily apparent. 30° is a default value used by Minne [44]. $|\Delta\chi|$ is identified by finding the maximum deviation in course between t_{detect} (the time the obstacle was detected) and t_{cpa} (the time of closest point of approach).

Non-readily apparent speed change penalty

The metric used for determining non-readily apparent speed change

$$\mathcal{P}_{\Delta U_{app}} \triangleq \frac{\delta_{app} - \delta_U}{\delta_{app}}, \quad \text{where} \quad \delta_U = \frac{U_0 - \min\left(U_0\delta_{app}, |\Delta U|\right)}{U_0} \tag{5.7}$$

U_0 is the speed at t_{detect}, ΔU is the maximum deviation in speed from U_0 in the interval $[t_{detect}, t_{cpa}]$, and δ_{app} is an apparent speed reduction threshold. The default value is 0.5. This metric is a modified version of the same metric used by Minne. The modifications are done in order to prevent penalty given if the maximum deviation in speed $|\Delta U|$ is larger than $U_0\delta_{app}$. In other words, if the speed reduction or increase is larger in magnitude than what is considered readily apparent, it should not be penalized as it was in Minne's penalty. This metric is used to penalize non-readily apparent speed change, it is not used to penalize nominal speed deviation.

Non-readily apparent maneuver penalty

Both the non-apparent course and speed penalties are combined to a new penalty, which should detect non-readily apparent maneuvers. This is defined as

$$\mathcal{P}_\Delta \triangleq \mathcal{P}_{\Delta U_{app}}\mathcal{P}_{\Delta\chi_{app}} \tag{5.8}$$

Non-readily apparent penalty modifications

In the preliminary project, during heavy oscillating vessel maneuvers caused by high uncertainty in obstacles state estimates, the non-readily apparent penalties didn't work as intended. Due to the large variation in course and speed (induced only by uncertain obstacle estimates), the resulting penalties ended up being close to zero (which was not correct). To prevent this from occurring in this thesis' evaluation, all states of the own-ship used in the calculation of non-readily apparent penalties are filtered by the use of a moving average filter. When applying a moving average filter, it is important to not remove the true underlying maneuvering done by the vessel, and only filter out undesired oscillating behaviour. Based on the time constants of the own-ship, the span of the filter is chosen to be 100 seconds.

Stand-on course change penalty

The stand-on course change penalty is used to penalize deviation in course when the vessel should keep her desired course. The metric is obtained from Woerner [63] and is defined as

$$\mathcal{P}^{17}_{\Delta\chi} \triangleq \frac{|\Delta\chi| - \chi_{md}}{\Delta\chi_{app} - \chi_{md}} \tag{5.9}$$

where $\Delta\chi$ is the maximum course change from t_{steady} to t_{cpa}. $\Delta\chi_{app} = 30^\circ$ is the apparent course change defined earlier and $U_{md} = 4^\circ$ is the minimum detectable course change. The metric is saturated between 0 and 1, and t_{steady} is defined in the modification paragraph below.

Stand-on speed change penalty

The stand-on speed change penalty is used to penalize deviation in speed when the vessel should keep her desired speed. Minne describes a metric given by

$$\mathcal{P}^{17}_{\Delta U} \triangleq \begin{cases} 0, & \text{if } |\Delta U| < \Delta U_{md} \\ 1 - \left(\frac{U_0}{U_{max}}\right)^2, & \text{if } \Delta U > 0 \\ \frac{|\Delta U|}{U_0}, & \text{if } \Delta U < 0 \end{cases} \tag{5.10}$$

where ΔU is the maximum deviation in speed from t_{steady} to t_{cpa}. U_{max} is the maximum speed and U_{md} is the minimum detectable speed change (0.2 m/s is used). The metric is saturated between 0 and 1, and t_{steady} is defined in the modification paragraph below.

Stand-on maneuver penalty

In situations where the give-way vessel does not comply with the COLREGS, the stand-on vessel should not be penalized for taking action to avoid collision. Minne [44] developed a metric which is used to weight the stand-on maneuvering penalties based on the distance to the obstacle at the time the maneuver was initiated. The penalty is defined as

$$\mathcal{P}_{\Delta}^{17} \triangleq 1 - \left(\frac{r_{maneuver} - R_{stand-on,max}}{R_{stand-on,max} - R_{stand-on,min}} \right)^2 \tag{5.11}$$

where $r_{maneuver}$ is the range to the obstacle at the time the maneuver was initiated. A maneuver initiated with a range of more than $R_{stand-on,max} = 5000$m results in no penalty reduction, while a range of less than $R_{stand-on,min} = 3000$m yields no penalty (full reduction). In between these ranges, there is a smooth transition of penalty reduction.

Stand-on maneuver penalty modifications

Based on the new testing scheme, the stand-on maneuver penalties from [44] need some adjustments to work as intended. Since the own-ship now may start in a position not on the nominal path, a maneuver taken in order to get closer to the nominal path should not be penalized. Thus, we have to figure out what maneuvers are made to get closer to the nominal path and what maneuvers are made to avoid the obstacles. Since the evaluation process is done entirely separated from the COLAV simulation (seen in Figure 5.1), it is not possible to know the exact reasoning behind the maneuvers being taken.

To determine the stand-on metrics, we define a new parameter, called t_{steady}. As the name implies, this parameter is used to define when the own-ship is in a steady state. "Steady state" is a relative concept, and may be defined in a various of ways. In this thesis, we define steady state as when the vessel's heading and speed are close (based on value) to the nominal heading and speed over a given number of time steps. If a "steady state" is not found, another test is performed. This is to check if the vessel's state (position and velocity) at t_{CPA} are closer to the desired state compared to the vessel's state at the time of detection, t_{detect}. If this is found to be true, the encounter is given no penalty for stand-on maneuvers. This is not a perfect approach, as the vessel may perform obstacle-evasive maneuvers in between the encounter start and end, resulting in no penalty for these maneuvers.

All stand-on encounters have been adjusted based on the method described above.

Other metrics

The most important metrics used in this thesis is presented above, but a number of other metrics are also investigated in Chapter 6. All metrics can be found in [44], but a short

description is given here:

- $\mathcal{P}_{DetectPortTurn}$, binary penalty given if the own-ship makes a port turn when being stand-on vessel in a crossing encounter.
- $\mathcal{P}_{LateManeuver}$, penalty to determine if actions are made in ample time.
- $\mathcal{P}_{PassAhead}$, binary penalty given in crossing situations where the starboard vessel is passed ahead by the other vessel.
- $\mathcal{S}_{PortToPortPassing}$, a score to determine the quality of the port-to-port passing.

Chapter 6

Results and discussion

This chapter presents the results from the testing of the COLAV system. The first section describes the testing procedure and the overall reasoning behind the testing method. The next sections present and discuss the results from a set of test cases used to test the COLAV system, ending with an overall discussion which tries to sum up the most important results.

6.1 Testing overview

The testing has been done by using a set of test cases. A test case is defined as a set of scenarios where the testing parameters are the same. The testing parameters include the scenario generation parameters and possibly tracking parameters depending on tracking being enabled or not. An example of such a set of parameters is shown in table Table 6.1. The scenarios are generated with the testing scheme presented in subsection 5.2.2.

Three different test cases are used in the testing:

1. **Test case - ground truth**. This test case is used to see how the COLAV algorithms perform under ideal conditions, i.e. where the exact state of the surrounding obstacles are known (tracking is disabled). It is then used as a reference base for comparison for the next test cases.
2. **Test case - normal conditions**. This test case tries to emulate the conditions that is usually taking place when a vessel of the size of the COLAV is in transit. The exact state of the surrounding obstacles are not known, hence, tracking is enabled for this test case.

3. **Test case - difficult conditions**. This test case is made such that the COLAV system is challenged to difficult tracking and COLAV situations. The main point is to test the limits of the system and find it's weaknesses.

For each test case, both the VO and the SBMPC COLAV algorithm have been tested. For every scenario in each test case, the evaluation metrics described in subsection 5.3.2 are used to quantify COLREGS compliance for the own-ship (and NOT the obstacles). The scenario scores are then collected such that it is possible to compare the results between the different COLAV algorithms.

Both the COLAV and the Gunnerus vessel are used as obstacles in the scenarios, and they are chosen randomly. Including both vessels in the same test case challenges the own-ship to react to different types of maneuvering. All obstacles use the VO algorithm for collision avoidance and no tracking is performed, i.e. the obstacles are always aware of the own-ship's exact state and the other obstacles exact state.

To make the understanding of the conditions for each test case easier, a figure showing a set of the scenarios are shown. The path of the own-ship together with true obstacle paths and tracks are included. Each of these figures is to show what's taken place in a fixed interval of time. It is then possible to compare these plots for the different test cases to see the differences in the number of obstacles, tracking uncertainty etc. An example of such a figure is Figure 6.1.

6.1.1 Scenario simulation

In all the upcoming scenarios, the own-ship's nominal (desired) path lies on the north axis, and the desired speed is set to 5 m/s. Hence, in the scenario figures, the own-ship will always try to steer upwards while also moving as close to the north axis as possible. All obstacles are also set to follow a straight path from their initial position.

For all test cases, the number of obstacles in a scenario is chosen to be 1. During the simulation, this will result in only 1 obstacle being present. In the testing phase, it was seen that the removal of the obstacle when the duration time of the scenario was reached, resulted in an unnatural behaviour as the obstacle suddenly disappearing short time after being in a collision situation. This could be avoided by choosing a larger scenario duration time (t_S) while keeping the same range for time to collision (t_{ttc}), but as this also will result in the own-ship being able to reach the desired path before the next scenario begins, this is not done. One of the main points of the testing scheme was to make the own-ship initial state not being on the desired path. Therefore, this issue was solved by letting each obstacle be present for two successive scenarios, making the obstacles gradually move away and out of range of the own-ship. As a result, all scenario figures includes two obstacles - the new obstacle for this specific scenario (numbered 1) and the obstacle from the previous scenario (numbered 2). Only the new obstacle is evaluated for the scenario

under consideration.

6.1.2 Results presentation structure

For each test case, the presentation of the results follows the same structure. At first, the overall scores are presented and discussed. Then, each COLAV algorithm has it's own subsection used to investigate and discuss the reasoning behind the scores given to this COLAV algorithm. If there are any particularly bad results, this is investigated further by including a specific scenario simulation.

6.2 Test case - ground truth

This test case is used to evaluate the COLAV system when there is no uncertainty regarding the obstacles positions and velocities. The own-ship and the obstacles always know the exact positions and velocities of the other vessels.

Parameter	Value	Unit	Description
n_S	200		Number of scenarios.
t_S	1000	s	Duration time of a scenario.
σ_{w_x}	500		Collision point noise strength.
n_O	1		Number of obstacles in a scenario.
t_{ttc}^{min}	700	s	Minimum initial time to collision for obstacles.
t_{ttc}^{max}	800	s	Maximum initial time to collision for obstacles.
v_{init}^{min}	2	m/s	Minimum initial obstacle speed.
v_{init}^{max}	12	m/s	Maximum initial obstacle speed.

Table 6.1: Test parameters for the ground truth test case.

The test parameters used in this test case is given in Table 6.1. The number of scenarios is chosen such that the likelihood of all the different COLREGS rules being applied is large. Also, the obstacles initial speed range is made relatively wide such that both types of overtaking situations may occur. This will also introduce quite different crossing situations; if an obstacle with high speed crosses from the right, the own-ship may not need to slow down and/or turn that much to starboard compared to a situation where the speed of the obstacle is slower. The time to collision parameters are chosen such that the own-ship should have a reasonable amount of time to adjust course and take action if needed. σ_{w_x} is given a value that will result in a great variation in collision points. This will produce, in addition to direct collision course situations, situations where obstacles will be such close

that actions have to be made in order to comply with the COLREGS. The time of each scenario is chosen such that the collision situation is likely to be finished ($t_S > t_{ttc}^{max}$). An overview of the degeneration of the test case is shown in Figure 6.1.

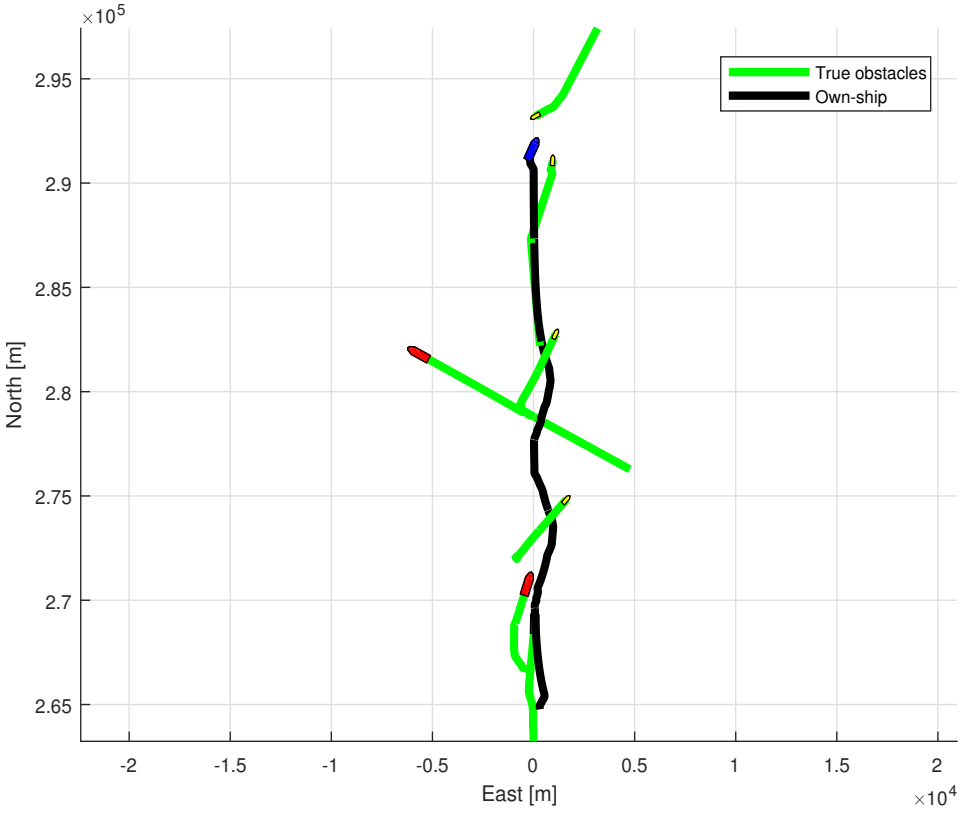


Figure 6.1: 5000 seconds from the simulation of the ground truth test case. The own-ship is colored in blue, the obstacle PSVs are red and the Gunnerus obstacles have been given the color yellow.

By looking at the overall mean scores from Figure 6.2 it is clear that the VO algorithm gets a significantly better score than the SBMPC algorithm. By using the mean rule scores (also in Figure 6.2), we get more insight in the overall calculation. VO performs better for all rules, except for Rule 14 which covers head-on situations. As described in subsection 3.2.3, the tuning of the SBMPC parameters is an extensive task because of the number of parameters and the nature of the cost function. By focusing more on parameter tuning, the SBMPC algorithm is expected to get higher evaluation scores. Nevertheless, we will look into the details and try to find the reasoning behind the scores, for both COLAV algorithms.

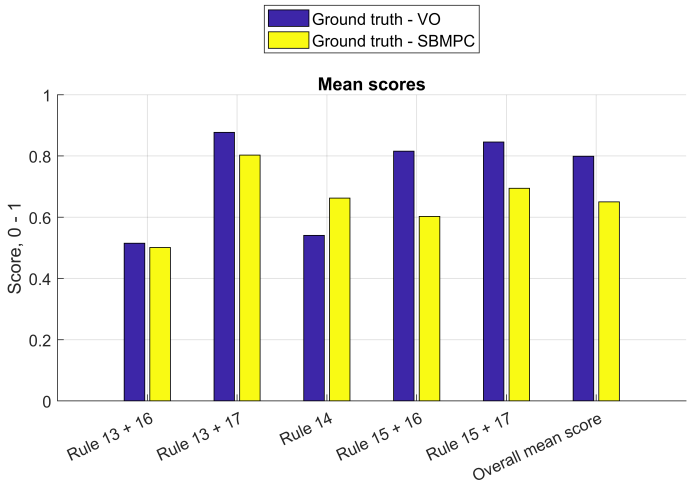


Figure 6.2: Mean encounter score for each COLREGS rule and the overall mean score, for the ground truth test case.

6.2.1 Ground truth - VO

A distribution of the encounter types for the scenarios in this simulation is given in Table 6.2. We see that most of the encounters are determined to be crossing, while overtaking encounter when own-ship is give-way vessel is only occurring in 4 scenarios. This distribution is almost directly given by looking at the testing parameters in combination with the testing scheme. Since the initial course of the obstacles are chosen uniformly from all possible courses, and because the combined crossing sections during encounter determination are larger than the other sections, this is the most likely distribution in the case of which category got the most number of encounters. As a result, the number of Rule 13+16 encounters are so low that the combined scores governing this encounter type should not be considered to be exact.

Encounter type	Number	Description
Rule 13 + 16	4	Overtaking, own-ship as give-way
Rule 13 + 17	20	Overtaking, own-ship as stand-on
Rule 14	15	Head-on
Rule 15 + 16	76	Crossing, own-ship as give-way
Rule 15 + 17	75	Crossing, own-ship as stand-on

Table 6.2: Ground truth - VO, encounter categorization.

As seen from Figure 6.2, the VO algorithm is given high scores on all rules except for Rule

14. To find out what lies behind this rule score calculation, we look at Figure 6.3. The safety and port-to-port passing scores are satisfying, but we see that the penalties covering non apparent course change and non starboard maneuver are large. To find out the reason for these high penalty values, we need to look at a specific head on scenario.

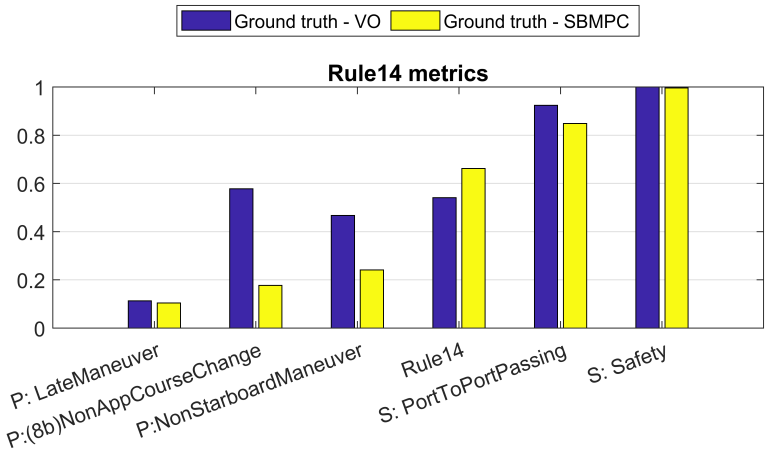
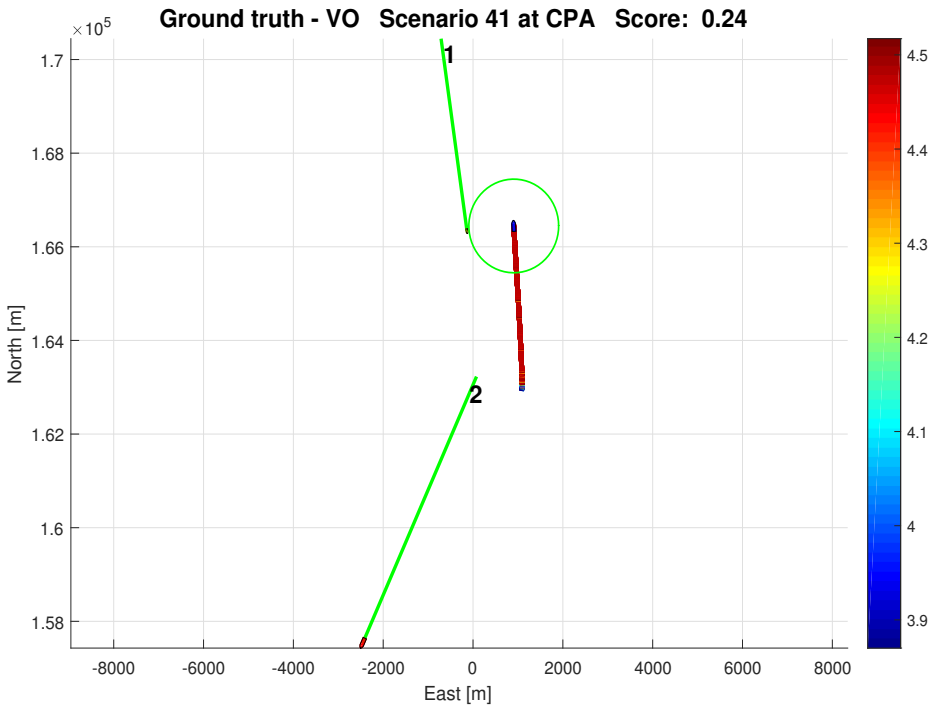


Figure 6.3: Score and penalties for rule 14, in the ground truth test case.

Figure 6.4 shows a head on situation from the VO simulation. It is clear that the distance between the obstacle (1) and own-ship at t_{CPA} is at a desired range, apparent alteration of course is seen. What is causing the own-ship to not alter the course apparently (to starboard) seems to be based on its initial state; the own-ship starts in a position located about 1200 meters east, heading straight north. At this time, it seems like obstacle 1 and the own-ship find themselves just inside the head-on section. The evaluation system determines which rules apply for a scenario at the start of the scenario and it is not changed during the scenario. The VO algorithm however, updates the situation type at regular intervals. In this specific case, this will result in the VO first taking head-on considerations, but as the situation continues, the situation type inside the VO will be updated to a crossing from left situation, resulting in the own-ship not taking any action to comply with Rule 14.

By looking at similar situations, this problem seems to be present, but the number of such situations are limited. This is because this type of situation only occurs if all the conditions described in the previous paragraph is present. In addition, mitigating this problem may be difficult as the COLAV algorithm has to update which COLREGS rules apply to be able to take action if other obstacles appears. However, a solution can be to store which COLREGS rule apply to a specific obstacle at the time the obstacle was detected, and not change this during the situation.



Encounter: 1

Active COLREGS: 14

Metric scores
Rule14: 0.24

SubMetric scores
P: LateManeuver: 0.02
P:(8b)NonAppCourseChange: 0.80
P:NonStarboardManeuver: 1.00
S: PortToPortPassing: 1.00
S: Safety: 1.00

Figure 6.4: Head-on situation where the own-ship does not make a readily apparent course change. The green circle is the desired passing distance (1000 m). The colour of the own-ship path is given by the barplot on the right which visualizes the speed (in m/s) of the own-ship.

6.2.2 Ground truth - SBMPC

The encounter distribution is shown in Table 6.3. Again it is clear that the testing scheme generates more crossing encounters than overtaking and head-on encounters. As a result, the Rule13+16 combined scores is not discussed here.

Encounter type	Number	Description
Rule 13 + 16	7	Overtaking, own-ship as give-way
Rule 13 + 17	16	Overtaking, own-ship as stand-on
Rule 14	13	Head-on
Rule 15 + 16	85	Crossing, own-ship as give-way
Rule 15 + 17	54	Crossing, own-ship as stand-on

Table 6.3: Ground truth - SBMPC, encounter categorization.

The SBMPC algorithm gets on average lower scores than the VO algorithm. It seems like the algorithm suffers from making course changes when being stand on vessel in crossing situations (Figure 6.5) and that the maneuvers when being give-way vessel is not as apparent as they should be (Figure 6.6) in addition of passing ahead encounters. This not so consistent behaviour may be caused by the poor tuning of the COLAV parameters. Based on the available parameters in the SBMPC algorithm, it should be possible to make the vessels maneuvering more apparent; if being stand-on vessel, no course changes nor speed changes should be initiated, and if having give-way responsibility, the course and speed changes should be large in magnitude. In total, this should make the vessel's intention clear. The suggestions for accomplishing this are:

- Increasing the path-following parameters k_P, k_χ such that path deviation only occurs when the vessel is near collision situations or if stand-on responsibility from COLREGS is active.
- Raise the value of p in order to weight risks that are far away in time less. This should make the handling of stand-on situations better.
- The penalty functions Δ_P, Δ_χ should be further increased such that a maneuver is made only if it deviates by a large value compared to the previous propulsion and course commands.

As described in subsection 3.3.4, the VO algorithm is not taking enough alteration of course when being give-way vessel. The SBMPC algorithm seems to be more sensitive in such cases, resulting in the SBMPC calculating a higher risk of collision compared to the VO algorithm and therefore adjusting its speed and course more. This may be an additional factor affecting the penalty scores.

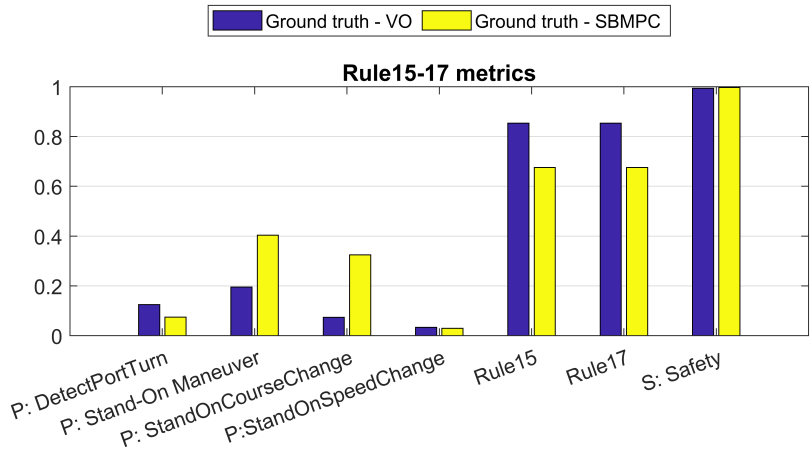


Figure 6.5: Score and penalties for rule 15-17, in the ground truth test case. NB, the ”Rule 15” and ”Rule 17” bars denote scores.

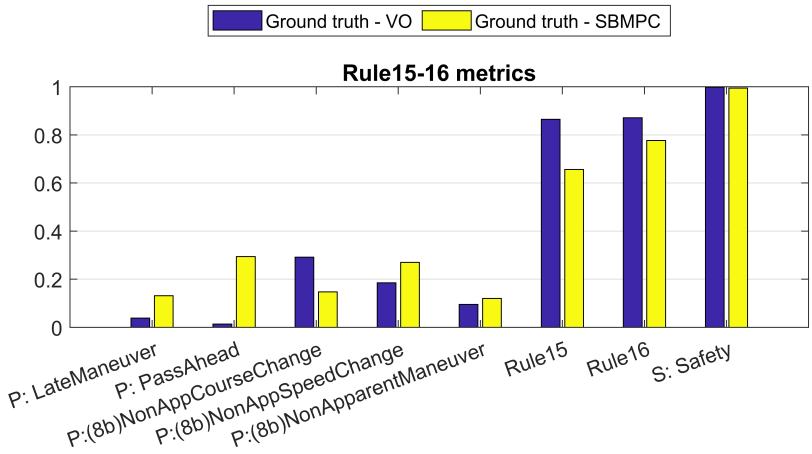


Figure 6.6: Score and penalties for rule 15-16, in the ground truth test case.

6.3 Test case - normal conditions

This test case is used to evaluate the performance of the COLAV system when ”normal conditions” are applied. To make comparison between this test case and the ground truth test case easier, the same scenario generation parameters as given in Table 6.1 is used, and

in addition, the two tracking parameters P_D and P_{FA} (Table 6.4).

Parameter	Value	Unit	Description
P_D	0.9		Obstacle detection probability.
P_{FA}	$3 \cdot 10^{-4}$		False measurement probability.

Table 6.4: Tracking parameters for the normal conditions test case.

Figure 6.7 shows a set of the scenarios from the simulation. The own-ship's COLAV algorithm is now using the track information as a decision base to find the new desired velocity. Because of the somewhat easy parameter settings, the tracks lies almost perfectly on top of the true obstacle paths. Consequently, the results for this test case compared to the ground truth test case is expected to be quite similar.

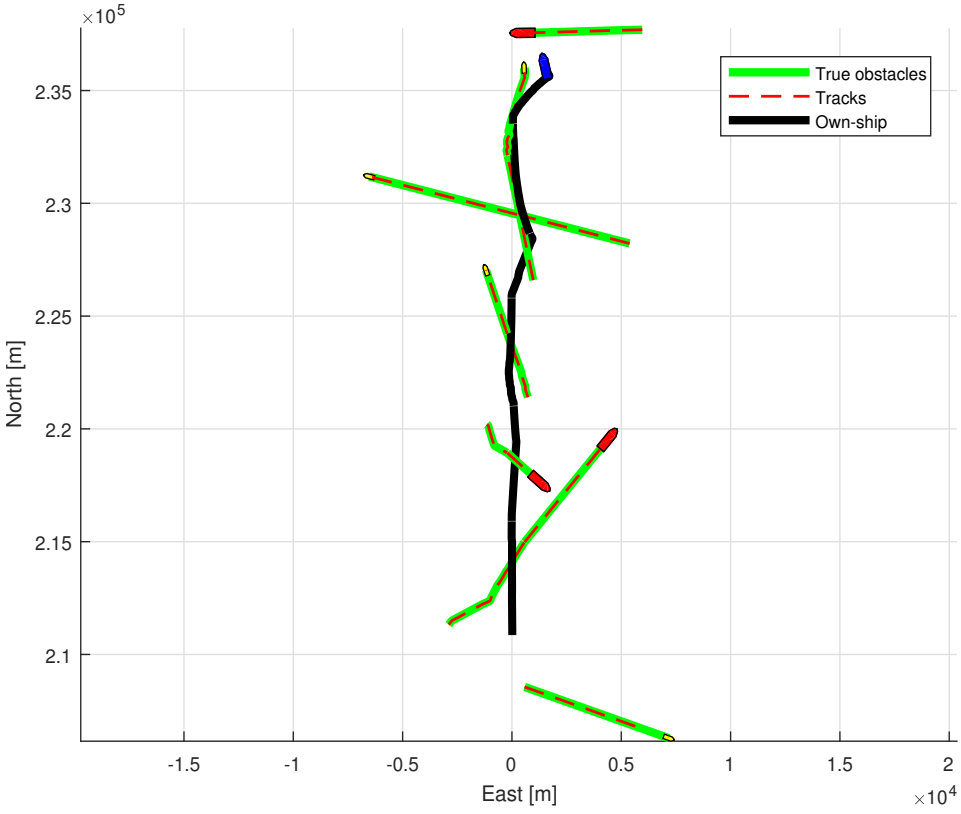


Figure 6.7: 5000 seconds from the simulation of the normal conditions test case.

The mean scenario scores (Figure 6.8) are, for both COLAV algorithms, higher compared

to the previous test case. The SBMPC algorithm gets a significantly higher mean score, where it seems like the Rule 15+16 score is the largest contributor to this positive difference. For the VO algorithm, we have a massive increase in the score for Rule 14.

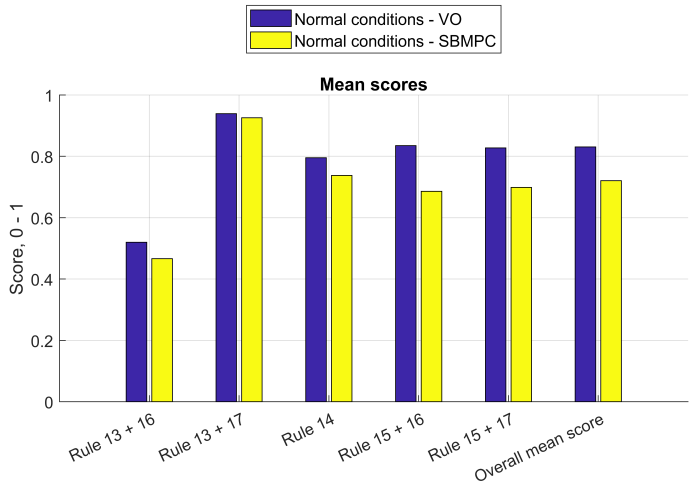


Figure 6.8: Mean encounter score for each COLREGS rule and the overall mean score, for the normal conditions test case.

6.3.1 Normal conditions - VO

Since the scenario generation parameters are chosen the same as in the ground truth test case, the encounter type distribution given in Table 6.5 follows the same pattern as before. As for the mean scores in Figure 6.8, the algorithm still performs pretty well in most cases, and we see a marked increase for the Rule14 score.

By iterating through all scenarios which is classified as head-on, some insight on the improved score for Rule 14 can be deducted. It seems like the specific situation described under section 6.2.1 where the VO changed COLREGS situation type during the scenario is not present in any of the scenarios in this test case. This is certainly affecting the score positively, and is likely the main contributor to the improved score.

6.3.2 Normal conditions - SBMPC

For the SBMPC, the encounter distribution (Table 6.6) is seen to be almost the same as in the previous test case. On the other hand, the rule scores have generally increased, especially the score for Rule 15+16. By comparing Figure 6.9 with Figure 6.6, it is clear

Encounter type	Number	Description
Rule 13 + 16	6	Overtaking, own-ship as give-way
Rule 13 + 17	27	Overtaking, own-ship as stand-on
Rule 14	25	Head-on
Rule 15 + 16	65	Crossing, own-ship as give-way
Rule 15 + 17	75	Crossing, own-ship as stand-on

Table 6.5: Normal conditions - VO, encounter categorization.

that all the penalty values have decreased by a considerable amount. As the number of Rule15+16 encounters is large for both test cases, this result cannot only be based on random coincidences.

Encounter type	Number	Description
Rule 13 + 16	10	Overtaking, own-ship as give-way
Rule 13 + 17	30	Overtaking, own-ship as stand-on
Rule 14	20	Head-on
Rule 15 + 16	70	Crossing, own-ship as give-way
Rule 15 + 17	70	Crossing, own-ship as stand-on

Table 6.6: Normal conditions - SBMPC, encounter categorization.

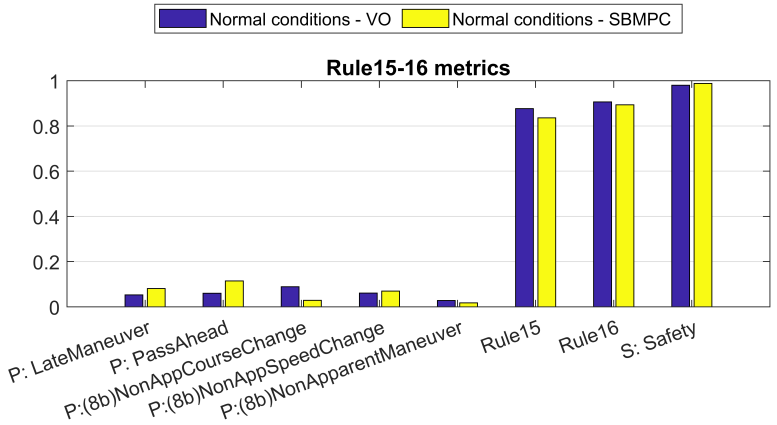
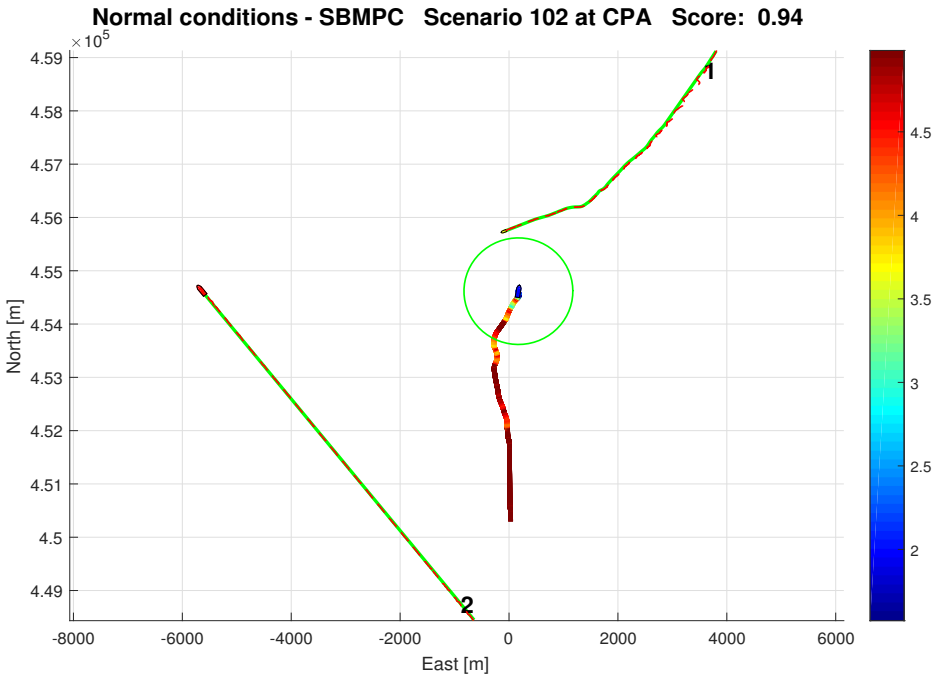


Figure 6.9: Score and penalties for rule 15-16, in the normal conditions test case.

The increased Rule 15+16 score may be a result of poor velocity estimates. Consider Figure 6.10, the tracks seems to follow more or less on top of the true obstacle paths, but some deviation is seen at the start of the track for obstacle 1. This may contribute to poor velocity estimates and therefore large variation in the future path estimation, resulting in

the own-ship making larger alteration of course and speed. Additionally, the adjustment of d_i^{safe} and d_i^{cl} based on the existence probability (described in subsection 3.2.3) will also affect this behaviour positively. The combination of these aspects will make the own-ship take avoiding maneuvers earlier and more drastically while also keeping a larger distance to the obstacles. Most of the penalty reduction seen for Rule 15+16 is likely to be a result of this.



Encounter: 1

Active COLREGS: 15, 16

Metric scores

Rule15: 0.94
Rule16: 0.94

SubMetric scores

P: LateManeuver: 0.06
P: PassAhead: 0.00
P:(8b)NonAppCourseChange: 0.00
P:(8b)NonAppSpeedChange: 0.00
P:(8b)NonApparentManeuver: 0.00
S: Safety: 1.00

Figure 6.10: A scenario where obstacle 1's estimate shows some fluctuation when being far away in distance, resulting in the own-ship making larger alteration of course and speed.

6.4 Test case - difficult conditions

This last test case is made to challenge the COLAV system, both the tracking system and the COLAV algorithms. The idea is to make the simulation conditions so hard that unusual situations will occur, and maybe force undiscovered weaknesses to appear.

Parameter	Value	Unit	Description
n_S	200		Number of scenarios.
t_S	800	s	Duration time of a scenario.
σ_{w_x}	500		Collision point noise strength.
n_O	1		Number of obstacles in a scenario.
t_{ttc}^{min}	500	s	Minimum initial time to collision for obstacles.
t_{ttc}^{max}	600	s	Maximum initial time to collision for obstacles.
v_{init}^{min}	2	m/s	Minimum initial obstacle speed.
v_{init}^{max}	12	m/s	Maximum initial obstacle speed.
P_D	0.4		Obstacle detection probability.
P_{FA}	$3 \cdot 10^{-3}$		False measurement probability.

Table 6.7: Test parameters for the difficult conditions test case.

All testing parameters are listed in Table 6.7. The number of scenarios and obstacles are the same as in the previous test cases, including the collision point variance and the initial obstacle speed range. The time to collison range is lowered in order to challenge the ownship (and obstacle vessels) to less maneuvering time. To make the tracking conditions more difficult, the detection probability (P_D) is reduced to a significantly lower value, and the clutter density (P_{FA}) is increased by a factor of 10. As seen in Figure 6.11, the number of false tracks has certainly increased. The plot only shows the longest false tracks, meaning that an additional large number of shorter tracks are present. All these false tracks will affect the COLAV algorithm as they are considered to be true obstacles. Also, observe that all the true obstacles have a corresponding track, even though there are some deviation and a smaller number of missing parts.

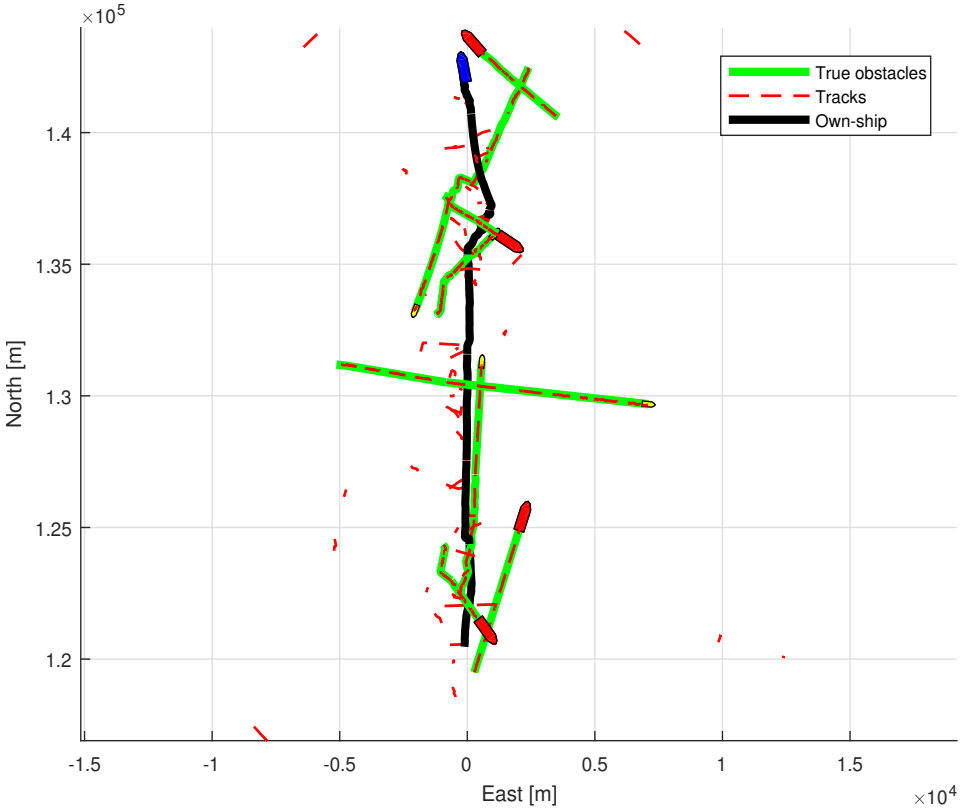


Figure 6.11: 5000 seconds from the simulation of the difficult conditions test case.

The overall mean scenario scores in Figure 6.12 show a decrease compared to the other test cases. As the simulation parameters for this test case are more challenging, this result is to be expected. On the other hand, while the decrease in score for the SBMPC algorithm seems to be in accordance with the adjustment of the testing parameters, the VO algorithm performs remarkably well. Given the number of false tracks and their influence on the decision making, the evaluation scores should be expected to decrease by a larger amount.

By directing the focus to the rule specific mean scores in Figure 6.12, more insight on the overall behaviour can be deducted. Whereas the VO algorithm seems to perform almost as good as earlier, the SBMPC algorithm still handles the stand-on responsibility in a good way, but the give-way encounters seems to suffer from the more challenging conditions. Notice that the score for Rule 13+16 is again the lowest for both algorithms. This is discussed later on.

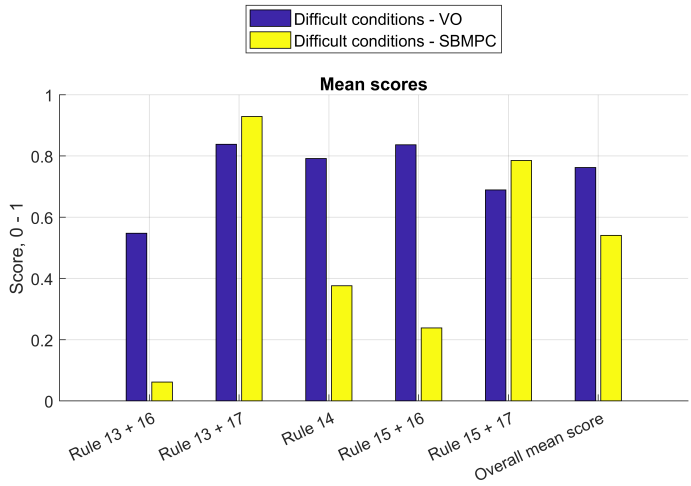


Figure 6.12: Mean encounter score for each COLREGS rule and the overall mean score, for the difficult conditions test case.

6.4.1 Difficult conditions - VO

Table 6.8 shows the encounter distribution for the VO simulation, which is seen to be more or less the same as earlier.

Encounter type	Number	Description
Rule 13 + 16	5	Overtaking, own-ship as give-way
Rule 13 + 17	36	Overtaking, own-ship as stand-on
Rule 14	16	Head-on
Rule 15 + 16	56	Crossing, own-ship as give-way
Rule 15 + 17	76	Crossing, own-ship as stand-on

Table 6.8: Difficult conditions - VO, encounter categorization.

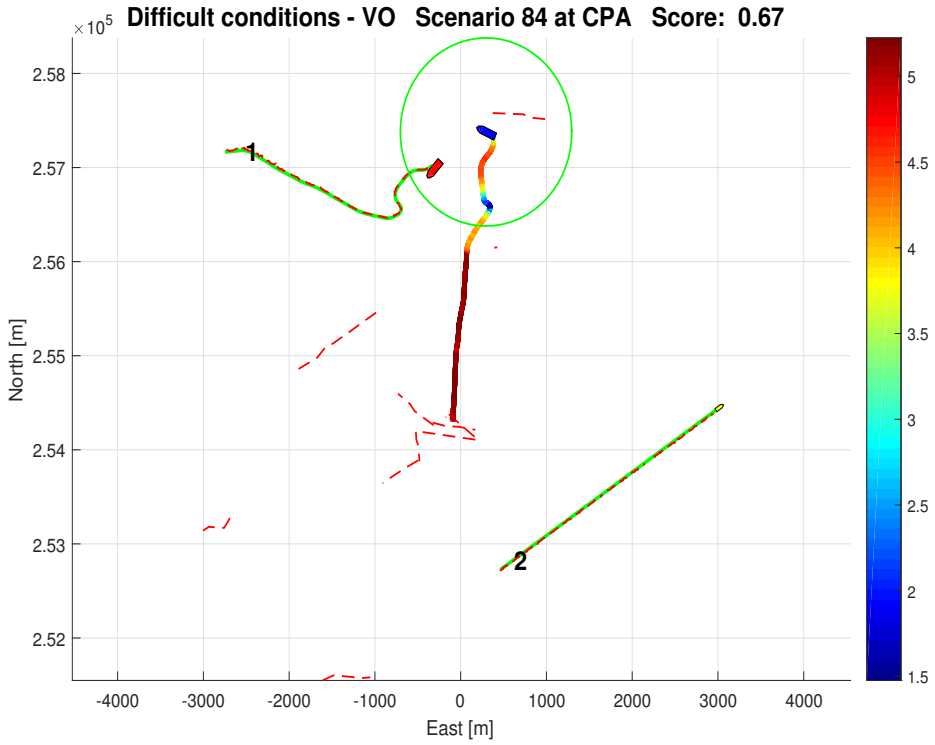
As mentioned earlier, the performance of the VO algorithm for this test case was very good; all encounter types are handled well and there is only a slight reduction in the evaluation scores. Comparing the scores to the SBMPC algorithm, in addition of having good mean scores, the small variation in the scores of the VO algorithm is to prefer, especially from a safety point of view.

Even though the overall behaviour was satisfying, there were some situations that were not handled well. This was mainly due to the challenging conditions, but it shows that there exist weaknesses. Figure 6.13 shows a scenario where the own-ship makes a sudden

course change towards an obstacle vessel, mainly because of a neighbouring false track suddenly appearing. As seen from the figure, the obstacle vessel almost immediately turns the other way around in order to not collide with the own-ship. Since this was the closest point between these vessels, and given how the false track appeared, one can argue that the situation was solved in a satisfying manner.

A small number of other similar situations did occur, and even though the COLREGS evaluation wasn't always good, the safety distances were maintained at a high level. As the method weights collision avoidance, and therefore safety, the most, this behaviour is as expected, but the VO algorithm showed a surprisingly high degree of robustness.

Another aspect which is important to mention is that the scenario (Figure 6.13) may had seen a totally different outcome if the obstacle vessels also needed to track the other vessels with the same amount of noise and uncertainty as the own-ship is experiencing. In general, for all the more challenging scenarios, the obstacle vessels may end up "saving" the scenarios (from a safety perspective) as they are always aware of the exact state of the other vessels. To give an example: if the own-ship suddenly alter the course towards an obstacle (as seen in Figure 6.13), the obstacle vessel will immediately initiate collision avoiding maneuvers that will (most likely) result in no collision occurring. The point is, situations where all vessels have uncertain surrounding state information should be investigated. This is proposed as future work.



Encounter: 1

Active COLREGS: 15, 17

Metric scores

Rule15: 0.67

Rule17: 0.67

SubMetric scores

P: DetectPortTurn: 0.00

S: Safety: 0.52

Figure 6.13: A scenario where the own-ship makes a sudden course change towards the obstacle vessel because of a nearby false track appearing.

6.4.2 Difficult conditions - SBMPC

As for all previous test cases, the encounter distribution is more or less equal (Table 6.9). Furthermore, Figure 6.12 shows that the SBMPC algorithm still handles both stand-on encounter types well, but struggles when it comes to the give-way responsibility. More specifically, Rule 15+16 and Rule 13+16 get so low evaluation score that further investigation is needed. But, as for the previous test cases, the number of Rule 13+16 encounters is so low that a discussion for this specific simulation will not be reliable.

Encounter type	Number	Description
Rule 13 + 16	8	Overtaking, own-ship as give-way
Rule 13 + 17	15	Overtaking, own-ship as stand-on
Rule 14	19	Head-on
Rule 15 + 16	66	Crossing, own-ship as give-way
Rule 15 + 17	86	Crossing, own-ship as stand-on

Table 6.9: Difficult conditions - SBMPC, encounter categorization.

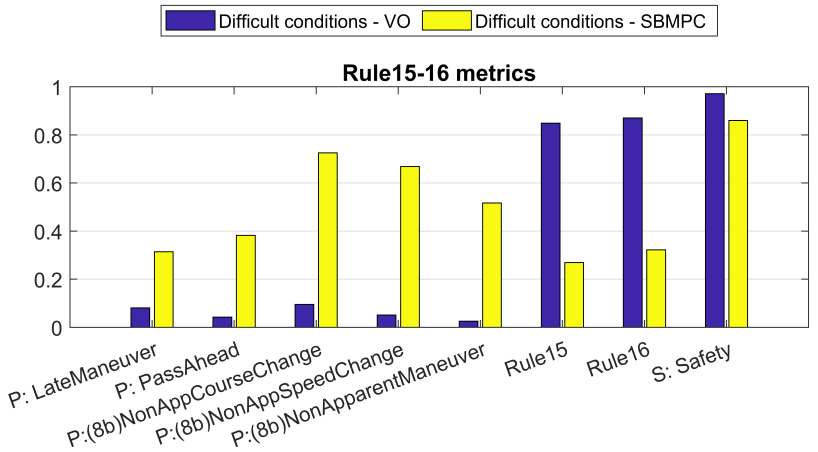


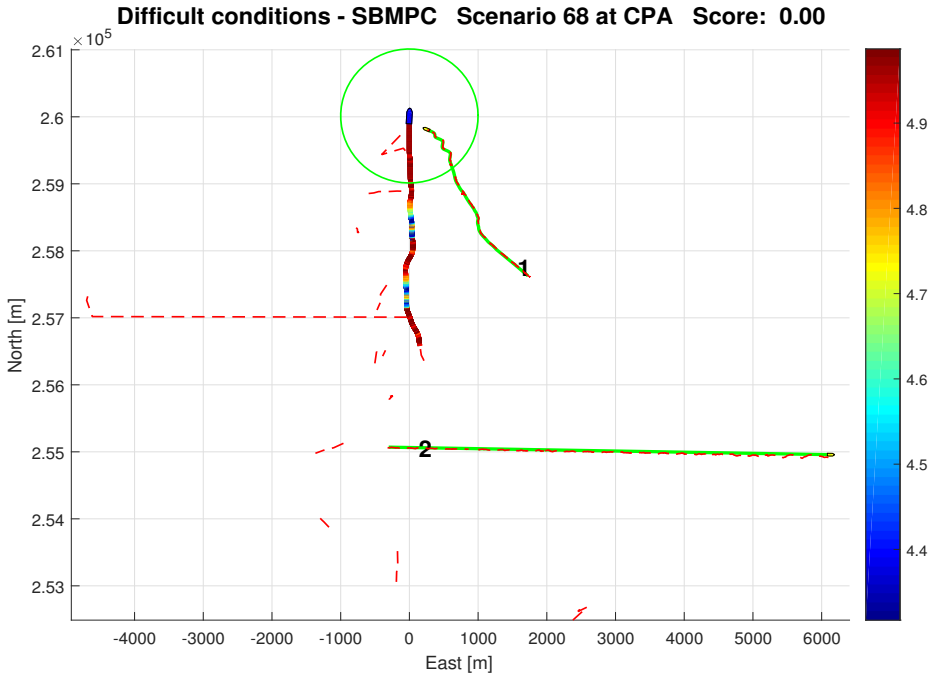
Figure 6.14: Score and penalties for rule 15-16, in the difficult conditions test case.

The overall behaviour of the SBMPC algorithm seems to coincide with the discussion done under the ground truth test case. The lack of consistent behaviour seems to be amplified for this test case, but what may surprise is that only the give-way handling is poorer. This may be a result of the update frequency of the algorithm. Only updating the desired velocity every 10 seconds (compared to every 0.05 seconds in the VO algorithm) may make the vessel not respond to the more noisy and faster tracks in this test case compared to the normal test case. So, instead of making sudden course and speed alterations, as

the VO algorithm is able to do, the SBMPC algorithm will seem to not react to the other obstacles when such maneuvering is taking place. Testing the algorithm with a higher update frequency in such scenarios may be investigated in future work.

A specific dangerous scenario was found in the simulation, shown in Figure 6.15. The own-ship has give-way responsibility for the obstacle on the right, and at the same time a number of false tracks appear on the port side of the own-ship. As the own-ship makes collision avoidance maneuvers for both the false tracks and the true obstacle, the result is the own-ship being ”squeezed” between the tracks. Ideally, the own-ship should have adjusted speed and course such that the true obstacle was passed behind with a large margin. Based on this encounter handling, it seems like the cost of violating stand-on responsibility is higher than violating give-way responsibility. This is not a desired behaviour and should be taken care of. A proposed solution is to decrease the penalty for starboard turning and maybe decrease the nominal course change penalty.

Continuing with the scenario in Figure 6.15, it is not only the own-ship that makes the situation more risky that it should have been. Based on the small distance between the own-ship and the obstacle vessel, the obstacle vessel should have initiated collision avoidance maneuvers, even though the own-ship does not comply with the COLREGS. This may be a result of the modification of the $f_{collision}$ function described in subsection 3.3.4, where the velocity obstacle is not considered during stand-on when the distance is more than 2000 meters. As the distance here is clearly less than 2000 meters, this should not have been a problem, but as a situation like this is able to occur, adjustments to the collision handling should be done.



Encounter: 1

Active COLREGS: 15, 16

Metric scores
Rule15: 0.00
Rule16: 0.06

SubMetric scores
P: LateManeuver: 0.03
P: PassAhead: 1.00
P:(8b)NonAppCourseChange: 0.56
P:(8b)NonAppSpeedChange: 0.85
P:(8b)NonApparentManeuver: 0.48
S: Safety: 0.11

Figure 6.15: A dangerous scenario where the own-ship is near a collision with obstacle 1. The combination of having a high number of false tracks on the left side and at the same time having a true obstacle on crossing course to the right, result in the own-ship keeping it's desired velocity straight ahead.

6.5 Overall discussion

6.5.1 VO vs SBMPC

The overall performance of the COLAV algorithms seems to be in favor of the VO algorithm. It handles almost all encounter types in all test cases in a very good way, a consistency that is highly appreciated. This is also supported by Figure 6.16, where the spread of the scenario scores for all test cases are shown. All the VO simulations are seen to get about the same distribution of scores, while the SBMPC simulations show a higher score variation. To sum up, the overall trends in Figure 6.16 are in accordance with the mean scores presented in the previous test cases.

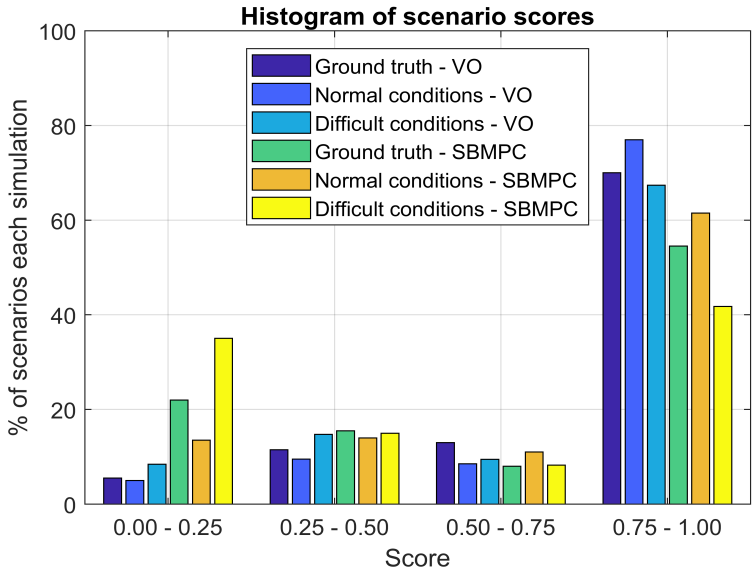


Figure 6.16: Histogram showing the spread of scores for all test cases.

A repeating aspect mentioned multiple times earlier in this thesis, is the tuning of the SBMPC algorithm. Having the combination of little time and a high number of tuning parameters, the task of finding the best parameters was difficult. The resulting parameters was seen to give desired behaviour in most cases, but as the tuning process was done when the obstacles state were perfectly known (ground truth), the algorithm has not been adopted to the more challenging situations.

As described in subsection 3.3.4, the VO algorithm was modified such the behaviour was more in compliance with the evaluation metrics. Given that the evaluation metrics are a perfect representation of the COLREGS, this approach is logical and should improve

the algorithms COLREGS compliance. But, as the evaluation metrics may not always comply perfectly with the COLREGS, this approach of algorithm tuning may not be ideal. Different tuning approaches may give better behaviour for realistic scenarios, but as these evaluation metrics have been adopted by the COLAV community as the evaluation metrics to be used, this approach is reasonable.

What may be an additional ”advantage” for the VO algorithm compared to the SBMPC algorithm is that the obstacles use the VO algorithm for COLAV. Being able to adapt the VO behaviour through testing to the obstacles which also were using VO, was easier than adapting the SBMPC to the VO. This is mostly due to the simpler COLAV approach used in the VO algorithm. Nevertheless, a COLAV method should be able to adapt and function well in a setting where other COLAV algorithms are operating.

6.5.2 Rule 13+16

During all the test cases, the number of encounters where the own-ship was overtaken by an obstacle vessel (Rule 13+16) was too small to give a reliable discussion based on the result from a specific test case. However, by combining the results for this encounter type for all the test cases, a discussion can be done.

With one exception, the rule score for encounter Rule 13+16 was the lowest for both COLAV algorithms in all test cases. The VO algorithm gets almost the same score in all test cases, while the SBMPC algorithm finds itself close to zero for the last test case. This is a critically low encounter score and should be investigated.

For a vessel of the size of the PSV, the number of real life encounters of this type may be relatively more than represented in these test cases. As a large vessel often needs to follow the shipping lanes, overtaking other large vessels presented in the shipping lanes may occur relatively often. With this is mind, the number of Rule 13+16 encounters should be increased. Modifying the existing testing scheme to choose more of the initial obstacle courses which will yield overtaking scenarios is easy to do. With the addition of decreasing the initial speed of the obstacles, this will give more Rule 13+16 encounters and should be considered as a part of the future work.

6.5.3 Metric modifications

The metrics used for evaluation of the COLAV system are heavily based on the metrics from [63]. As described in section subsection 5.3.2, being able to reuse the same metrics for the new testing scheme used in this thesis, some modifications to the stand-on vessel evaluation had to be done. The modifications used may result in a higher score being given than what should have been given. The point is, comparing the results from the test cases done in this thesis directly to other work on the same topic may not be applicable. How-

ever, the comparison given internally in this thesis between the two COLAV algorithms and the different test cases certainly is applicable.

6.5.4 Domain discussion

Having a large vessel as the PSV used in this thesis as the SUT, a discussion of what domains are useful to investigate is reasonable. When a PSV is in transit out on the open sea where there only exist other large vessels equipped with AIS, the collision avoidance problem is not that hard (relatively speaking) and may be assumed solved by [22] or [37]. On the contrary, vessels like the PSV may find itself near shore or in other situations where smaller faster vessels are present. A fully autonomous vessel should also be able to handle such environments, with the addition of situations with poor visibility properties. The test cases presented in this section is used to imitate such situations.

The ground truth test case can be seen as a situation where the PSV moves in an area with a high number of large vessels. This can be near oil rigs where other PSVs and vessels are present. As for the normal conditions test case, a situation with clear sky and good visible conditions, but no good AIS information available, could be the realistic counterpart. The last test case may be a real life situation where the PSV is approaching a large town in heavy foggy weather and a high number of nearby vessels is present.

Chapter 7

Conclusion and future work

7.1 Conclusion

A COLAV system including a tracking system based on the JIPDAF and the COLAV algorithms VO and SBMPC have been tested in a wide variety of relevant ASV scenarios. The tracking is done by emulating noisy radar measurements from the true obstacle measurements. Based on the track information from the tracking module, a COLAV algorithm enables automatic collision avoidance. The testing has been executed in a simulated environment where a high number of succeeding collision situations have been simulated. To evaluate the COLAV system's scenario execution, a number of evaluation metrics used to determine COLREGS compliance have been applied.

When exposed to no or little amount of noise, the tracking system provides accurate estimates of the obstacles positions and velocities. The resulting performance of the COLAV system depends on which COLAV algorithm is being used, where the VO algorithm is seemed to handle all encounter types in a more satisfying way than the SBMPC algorithm. However, the SBMPC method may suffer from poor parameter tuning, thus having a potential of significant performance improvement.

In more challenging scenarios where the noise is raised to considerably higher values, the tracking system's obstacle estimates are highly fluctuating and some times completely missing, in addition of having a high number of false tracks. These challenging conditions result in lower evaluation scores for both COLAV algorithms, especially the SBMPC algorithm. However, given the high number of false tracks present in the simulations and despite getting lower evaluation scores, the VO algorithm performed remarkably well.

7.2 Future work

As this thesis covers a wide range of research areas, there is a great potential for improvements and future work. In the following, a number of suggestions for future work are given.

Dealing with land

Constraints posed by the sea shore may limit the possible actions the COLAV methods can choose between. While the SBMPC method easily can include handling of land [37], the VO method may need additional tweaks to work at all in the presence of land.

SBMPC parameter tuning

The SBMPC algorithm may have a considerable potential regarding the tuning parameters. This should be investigated, or other COLAV methods based on MPC [28] may be looked into for future implementation.

Include tracking for obstacles

To further enhance the realistic aspects of the noisy scenario simulations, target tracking should be enabled on all vessels in a scenario. It is then possible to identify and investigate the more dangerous situations which may appear in similar real life situations.

COLAV while path following

In this thesis, all obstacles and the own-ship follow straight-line paths in the scenario simulations. Investigation on how the COLAV system handles non-straight obstacle and own-ship paths should be done as this is how most of real-life situation is carried out. The evaluation metrics may need additional modifications to work in such situations.

Additional sensors

The implemented system only contains radar measurement emulation. A number of other sensors (cameras, LIDAR etc.) may be implemented, as this is common on real life vessel. A sensor fusion module combining all the sensor information may then also be needed.

Overtaking when own-ship has give-way responsibility (Rule 13+16)

During the simulations, it was seen that the handling of Rule 13+16 encounters was quite poor for both COLAV algorithms. As this may be one of the more common encounter types for vessels with the size of the PSV, further investigation is needed. This can be accomplished by using the testing scheme with a restricted range of initial obstacle courses and a low maximum initial obstacle speed.

Nomenclature

Collision avoidance

α_c	Parameter used to determine the cost related to a collision.
β	Bearing angle.
β^{180}	Normalized bearing angle.
β_c	Parameter used to determine the cost related to COLREGS violation.
$\boldsymbol{p}_A$	Position vector for own-ship, in VO algorithm.
$\boldsymbol{p}_B$	Position vector for obstacle B, in VO algorithm.
$\boldsymbol{v}_A$	Velocity vector for own-ship, in VO algorithm.
$\boldsymbol{v}_B$	Velocity vector for obstacle B, in VO algorithm.
$\chi_{ca,last}$	Previous course offset.
χ_{ca}	Course offset.
$\Delta_{\chi,port}$	Parameter used to prioritize readily apparent port maneuvers.
$\Delta_{\chi,starboard}$	Parameter used to prioritize readily apparent starboard maneuvers.
Δ_P	Parameter used to prioritize readily apparent speed maneuvers.
ϵ_{VO}	Distance used to determine if two vessels are close.
ι_c	Parameter used to weight cost of speed penalty.
κ_i	Cost function parameter used to weight COLREGS violation.

λ	Parameter used to determine the length of the velocity vector in the VO algorithm.
$\mathcal{A}$	Set representation of own-ship.
$\mathcal{B}$	Set representation of obstacle B.
$\mathcal{C}_{ij}$	New VO cost function.
$\mathcal{C}_{ij}^{'}$	Original VO cost function.
$\mathcal{H}^a$	Total cost associated with scenario a.
$\mathcal{R}_i^a$	Collision risk factor.
$\mathcal{V}_1$	Region on the left side of the obstacle seen from the own-ship.
$\mathcal{V}_2$	Region on the right side of the obstacle seen from the own-ship.
$\mathcal{V}_3$	Region in a direction away from the obstacle.
μ_i^a	Binary term used to determine COLREGS violation.
$\psi_{d,colreg}$	Desired heading the last time the COLREG situation was updated.
$\tilde{\boldsymbol{u}}$	Deviation from the desired speed and heading.
$\varkappa_c$	Parameter used to weight cost of port penalty.
$\vec{L}_i^a$	Unit vector pointing in the LOS-direction of own-ship to obstacle i in scenario a.
$\vec{v}_0^a$	Own-ships velocity in scenario a, in SBMPC algorithm.
$\vec{v}_i^a$	Velocity of obstacle i in scenario a, in SBMPC algorithm.
D	The set of discrete sample times in the SBMPC algorithm.
$d_{0,i}^a$	Distance from own-ship to obstacle i in scenario a.
d_{CPA}	Distance between the vessels at CPA.
d_i^{cl}	Distance parameter used to determine the maximum range of where COLREGS apply.
d_{min}	Maximum distance for a collision situation to be examined.
f	Cost related to path following.
$f_{collision}$	Modified version of $f_{collision}^{'}$.
$f_{collision}^{'}$	Boolean function used to determine if a collision will occur.
$f_{COLREGS}$	Boolean function used to determine if the COLREGS is violated.

f_{port}	Boolean function used to determine if port penalty is to be added.
f_{speed}	Boolean function used to determine if speed penalty is to be added.
k_P	Parameter used to weight nominal speed.
$k_{\chi,port}$	Parameter used to penalize port maneuvers.
$k_{\chi,starboard}$	Parameter used to penalize starboard maneuvers.
k_χ	Parameter used to prioritize nominal course.
P_C	Propulsion command.
p_r	Collision risk factor parameter used to weight time.
P_{last}	Previous propulsion command.
q_c	Parameter used to weight cost of velocity deviation.
q_r	Collision risk factor parameter used to weight distance.
r_A	Radius of the circular approximation of own-ship.
r_B	Radius of the circular approximation of obstacle B.
t	Time index.
T_s	Sample interval in the SBMPC algorithm.
t_{CPA}	Time to CPA.
t_{max}	Maximum time for a collision situation to be examined.
$u_{d,last}$	Desired velocity from the previous time step.
VO_B^A	The velocity obstacle induced by obstacle B for obstacle A.
CLOSE	Boolean variable used to determine if an obstacle is close.
CROSSED	Boolean variable used to determine if the own-ship is in a crossing situation.
HEAD-ON	Boolean variable used to determine if the own-ship is in a head-on situation.
OVERTAKEN	Boolean variable used to determine if the own-ship is being overtaken.
RULE14	Boolean variable used to determine if COLREGS rule 14 is violated.
RULE15	Boolean variable used to determine if COLREGS rule 15 is violated.

Evaluation metrics and scenario generation parameters

$\mathcal{P}_{\Delta\chi_{app}}$	Non-readily apparent course change penalty.
$\mathcal{P}_{\Delta U_{app}}$	Non-readily apparent speed change penalty.
$\mathcal{P}^{17}_{\Delta U}$	Stand-on speed change penalty.
$\mathcal{P}^{17}_{\Delta\chi}$	Stand-on course change penalty.
$\mathcal{P}_{\Delta}$	Non-readily apparent maneuver penalty.
$\mathcal{P}^{17}_{\Delta}$	Stand-on maneuver penalty.
$\mathcal{P}_{DetectPortTurn}$	Port turn penalty.
$\mathcal{P}_{LateManeuver}$	Late maneuver penalty.
$\mathcal{P}_{PassAhead}$	Pass ahead penalty.
$\mathcal{S}_{PortToPortPassing}$	Port to port passing score.
$\boldsymbol{P}_0$	Position of own-ship at the start of a scenario.
$\boldsymbol{P}_{col}$	Collision point between own-ship and obstacle.
$\boldsymbol{P}_{est}$	Estimated path position of own-ship.
$\boldsymbol{P}_{init}$	Initial obstacle position.
χ_{init}	Initial obstacle course.
$\sigma^2_{w_x}$	Collision point noise strength.
n_O	Number of obstacle in a scenario.
n_S	Number of succeeding scenarios in a testcase.
S_{safety}	Safety metric.
t_S	Duration time of scenario.
t_{ttc}	Time to collision.
t^{max}_{ttc}	Maximum time to collision.
t^{min}_{ttc}	Minimum time to collision.
t_{ttp}	Time to path.
v_{init}	Initial obstacle speed.
v^{max}_{init}	Maximum initial obstacle speed.
v^{min}_{init}	Minimum initial obstacle speed.

w_x	Collision point noise.

Vessel modeling and control

α_k	Path-tangential angle.
$\boldsymbol{\epsilon}$	Along-track distance and cross-track error vector.
$\boldsymbol{\eta}$	Position and orientation vector.
$\boldsymbol{\tau}$	Generalized forces and moments from the actuators.
$\boldsymbol{\tau}_{wind}, \boldsymbol{\tau}_{wave}$	Environmental forces from wind and waves.
$\boldsymbol{\varsigma}$	Linear and angular velocity vector.
$\boldsymbol{\varsigma}_d$	Reference values to the feedback linearizing MIMO controller.
$\boldsymbol{C}_A$	Added mass coriolis-centripetal matrix.
$\boldsymbol{C}_{RB}$	Rigid body coriolis-centripetal matrix.
$\boldsymbol{D}$	Damping matrix.
$\boldsymbol{K}_p$	Proportional gain in the feedback linearizing MIMO controller.
$\boldsymbol{K}_p^G$	Proportional gain in the feedback linearizing MIMO controller for the Gunnerus vessel.
$\boldsymbol{K}_p^{PSV}$	Proportional gain in the feedback linearizing MIMO controller for the PSV vessel.
$\boldsymbol{M}$	Inertia matrix.
$\boldsymbol{M}_A$	Added mass matrix.
$\boldsymbol{M}_{RB}$	Rigid body inertia matrix.
$\boldsymbol{n}$	Non-linear component of the feedback linearizing MIMO controller.
$\boldsymbol{p}^n$	Vessel position vector in ILOS guidance.
$\boldsymbol{p}_{b/n}^n$	Position of origo in body-fixed frame expressed in the inertial frame.
$\boldsymbol{p}_k^n$	Waypoint position vector.
$\boldsymbol{R}$	Rotation matrix.
χ_d	Desired course.
χ_r	Velocity-path relative angle.
Δ	Lookahead distance in ILOS guidance.
ψ	Euler angle about the z axis.

ψ_d	Desired heading, in the heading controller.
σ	Tuning parameter in ILOS guidance.
$\tilde{\boldsymbol{\varsigma}}$	Error between reference values $\boldsymbol{\varsigma}_d$ and the actual values $\boldsymbol{\varsigma}$.
$\tilde{\psi}$	Error between desired heading and actual heading.
$\vec{r}_g$	The vector from center of gravity of the vessel to origo in the $\{b\}$ frame.
$\{b\}$	Body-fixed reference frame.
$\{n\}$	Inertial reference frame.
e	Cross-track error.
e_{int}	Integral term in ILOS guidance.
F_x, F_y, F_z	Force and moment components of the $\boldsymbol{\tau}$ vector.
I_z	Moment of inertia about the z-axis.
$K_{p,1}, K_{p,2}, K_{p,3}$	Components of the proportional gain $\boldsymbol{K}_p$.
$K_{p,\psi}$	Proportional gain in the heading controller.
$K_{p,\psi}^{G}$	Proportional gain in the heading controller for the Gunnerus vessel.
$K_{p,\psi}^{PSV}$	Proportional gain in the heading controller for the PSV vessel.
m	Mass of vessel.
N	Body-fixed moment about the z axis.
r	Body-fixed angular velocity about the z axis.
s	Along-track distance.
u, v	Body-fixed linear velocities in the x and y direction.
u_d, v_d, r_d	Components of the reference values vector $\boldsymbol{\varsigma}_d$.
X, Y	Body-fixed forces in the x and y direction.
x, y	Body-fixed positions in the x and y direction.
x_g, y_g, z_g	x, y and z components of the $\vec{r}_g$ vector.
x_k, y_k	Inertial-fixed waypoint position in the x and y direction.

Target tracking

$(\hat{\cdot})_{k\|k-1}$	Predicted (a priori) estimate.

$(\hat{\cdot})_k$	Updated (a posteriori) estimate.
β_k^i	Probability of measurement i being target originated.
$\beta_k^{t,j}$	Probability of measurement j originating from target t.
$\boldsymbol{\nu}_k$	Combined innovation.
$\boldsymbol{\nu}_k^i$	Measurement innovation for measurement i.
$\boldsymbol{a}_k$	An association hypothesis.
δ	Targets-with-measurement vector.
η_{init}^t	Initial visibility probability.
η_k^t	Visibility probability for target t.
γ_G	Gate threshold.
$\hat{\mathbf{x}}_k$	Estimated state vector.
$\hat{\mathbf{x}}_k^{t,\boldsymbol{a}_k}$	Posterior event-conditional state.
λ_P	Poisson distribution intensity of false measurements.
$\mathcal{A}_M$	A problem-solution pair.
$\mathcal{A}_p$	A score matrix.
$\mathcal{A}_s$	A solution matrix.
$\mathcal{L}$	List of problem-solution pairs, $\mathcal{A}_M$.
$\mathcal{N}$	Normal distribution.
σ_a	Process noise strength.
σ_r	Measurement noise strength.
$\mathbf{F}_t$	State transition matrix.
$\mathbf{H}$	State observation matrix.
$\mathbf{K}_k$	Kalman gain.
$\mathbf{P}_k$	State covariance matrix.
$\mathbf{P}_k^c$	Covariance of state updated with the true measurement.
$\mathbf{Q}_T$	Process noise covariance matrix.
$\mathbf{R}$	Measurement noise covariance matrix.
$\mathbf{S}_k$	Predicted measurement covariance matrix.

$\mathbf{v}_k$	Process noise vector.
$\mathbf{w}_k$	Measurement noise vector.
$\mathbf{x}_k$	State vector at time step k.
$\mathbf{x}_k^{t,\boldsymbol{a}_k}$	Posterior event-conditional covariance.
$\mathbf{z}_k$	Measurement vector.
$\mathbf{z}_k^i$	I-th measurement at time step k.
$\tilde{\mathbf{P}}_k^c$	Spread of innovations.
ε_{init}^t	Initial existence probability.
ε_k^t	Existence probability for target t.
φ	Number of false measurements.
ϱ_k	The set of non-associated measurements.
i	Measurement index.
j	Target index.
k	Discrete time index.
$l^{t,\boldsymbol{a}_k}$	Likelihood ratio.
m_k	Number of validated measurements.
n	Number of targets.
p	Probability density function.
P_D	Detection probability.
P_G	Gate probability.
P_S	Death probability.
p_{FA}	False measurement rate.
T	Time step in motion model.
V	Volume of the surveillance region.
v_{max}	A parameter chosen to be larger than maximum velocity.
Z_k	The current set measurements.
$Z_{1:k}$	The cumulative set of measurements up until time k.
NIS	Defines the elliptic region called the validation region.

Bibliography

[1] C. Allen. *Farwell's Rules of the Nautical Road*. Naval Institute Press, 2005.

[2] R. Aoki. Report on autonomous mode disengagements for waymo self-driving vehicles in california. Technical report, WAYMO, 2017.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb 2002.

[4] Y. Bar-Shalom, F. Daum, and J. Huang. The probabilistic data association filter, estimation in the presence of measurement origin uncertainty. *2009 IEEE Control Systems Magazine (CSM)*, pages 82–100, Dec 2009.

[5] Y. Bar-Shalom, T. Kirubarajan, and X. Lin. Probabilistic data association techniques for target tracking with applications to sonar, radar and eo sensors. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):37–56, Aug 2005.

[6] Y. Bar-Shalom and X. Li. *Multitarget-Multisensor-Tracking: Principles And Techniques*. YBS, 1995.

[7] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, 09 1975.

[8] S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, Jan 2004.

[9] A. Boniske. Annual disengagement report for autonomous vehicles. Technical report, GM Cruise LLC, 2017.

[10] E. Brekke. Compendium in sensor fusion. teaching material for a future course in sensor fusion, 20..

[11] R. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wile & Sons, Inc, 2012.

[12] H. Chang. Power electronics control design and testing in the 21st century. *Typhoon HIL Blog*, 05 2016.

[13] I. J. Cox and M. L. Miller. On finding ranked assignments with application to multitarget tracking and motion correspondence. *IEEE Transactions on Aerospace and Electronic Systems*, 31(1):486–489, Jan 1995.

[14] C. Denker, M. Baldauf, S. Fischer, A. Hahn, R. Ziebold, E. Gehrmann, and M. Semann. e-navigation based cooperative collision avoidance at sea: The mtcas approach. In *2016 European Navigation Conference (ENC)*, pages 1–8, May 2016.

[15] DNV. The revolt - a new inspirational ship concept, 2017. Accessed: 05-04-2018.

[16] W. Elmenreich. An introduction to sensor fusion. *ResearchGate*, 04 2018.

[17] B.-O. Eriksen and M. Breivik. Mpc-based mid-level collision avoidance for asvs using nonlinear programming, 08 2017.

[18] B.-O. Eriksen, E. F. Wilthil, A. Lindahl Flåten, E. Brekke, and M. Breivik. Radar-based maritime collision avoidance using dynamic window, 03 2018.

[19] B.-O. H. Eriksen. Horizontal collision avoidance for autonomous underwater vehicles. Master's thesis, Norwegian University of Science and Technology, 2015.

[20] B. O. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig. A modified dynamic window algorithm for horizontal collision avoidance for auvs. In *2016 IEEE Conference on Control Applications (CCA)*, pages 499–506, Sept 2016.

[21] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using the relative velocity paradigm. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 560–565 vol.1, May 1993.

[22] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[23] Food and Agriculture Organization of the United Nations. Vessel model, 2013. [Online; accessed April 27, 2018].

[24] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, Jul 1983.

[25] T. Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wile & Sons, Inc, 2011.

[26] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33, Mar 1997.

[27] S. Ge and Y. Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3):207–222, Nov 2002.

[28] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen. Mpc-based collision avoidance strategy for existing marine vessel guidance systems, 2017.

[29] J. Han, J. Kim, and N. s. Son. Persistent automatic tracking of multiple surface vessels by fusing radar and lidar. In *OCEANS 2017 - Aberdeen*, pages 1–5, June 2017.

[30] V. Hassani, A. Ross, O. Selvik, D. Fathi, F. Sprenger, and T. Berg. Time domain simulation model for research vessel Gunnerus. In *34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015), St. John's, Newfoundland, Canada, 2015.*, 2015.

[31] E. S. Henriksen. Automatic testing of sensor fusion-based maritime collision avoidance systems. unpublished report, 2017.

[32] P. Horridge and S. Maskell. Real-time tracking of hundreds of targets with efficient exact jpdaf implementation. In *2006 9th International Conference on Information Fusion*, pages 1–8, July 2006.

[33] C. Ilas. Electronic sensing technologies for autonomous ground vehicles: A review. In *2013 8th International Symphosium on Advances Topics in Electrical Engineering (ATEE)*, pages 1–6, May 2013.

[34] IMO. Convention on the international regulations for preventing collisions at sea, 1972 (colregs). *Lloyd's Register*, 1972. Accessed: 04-03-2018.

[35] IMO. Ais transponders. *International Maritime Organization*, 2018.

[36] T. Johansen, T. Fossen, and B. Vik. Hardware-in-the-loop testing of dp systems. 01 2005.

[37] T. A. Johansen, T. Perez, and A. Cristofaro. Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3407–3422, Dec 2016.

[38] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. Safe maritime autonomous navigation with colregs, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, 39(1):110–119, Jan 2014.

[39] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct 2003.

[40] V. J. Lumelsky and T. Skewis. Incorporating range sensing in the robot navigation function. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5):1058–1069, Sep 1990.

[41] J. Luo, K. R. Pattipati, P. K. Willett, and F. Hasegawa. Near-optimal multiuser detection in synchronous cdma using probabilistic data association. *IEEE Communications Letters*, 5(9):361–363, Sept 2001.

[42] J. E. Manley. Unmanned surface vehicles, 15 years of development, 2008. Accessed: 15-02-2018.

[43] K. Maritime. Autonomous ship project, key facts about yara birkeland, 2018. Accessed: 05-04-2018.

[44] P. K. E. Minne. Automatic testing of maritime collision avoidance algorithms. Master's thesis, Norwegian University of Science and Technology, 2017.

[45] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[46] D. Musicki and R. Evans. Joint integrated probabilistic data association: Jipda. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, July 2004.

[47] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, Jun 1994.

[48] C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):560–576, Jun 2001.

[49] Rolls-Royce. Remote and autonomous ships - the next steps, 2017.

[50] A. Ross, V. Hassani, O. Selvik, E. Ringen, and D. Fathi. Identification of non-linear manoeuvring models for marine vessels using planar motion mechanism tests. In *34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015), St. John's, Newfoundland, Canada, 2015.*, 2015.

[51] M. Schuster, M. Blaich, and J. Reuter. Collision avoidance for vessels using a low-cost radar sensor. In *19th IFAC World Congress - Cape Town, South Africa*, Aug 2014.

[52] E. Serigstad. Hybrid collision avoidance for autonomous surface vessels. Master's thesis, Norwegian University of Science and Technology, 2017.

[53] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 508–513 vol.1, 2002.

[54] T. Statheros, G. Howells, and K. M. Maier. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of Navigation*, 61(1):129–142, 2008.

[55] T. Stenersen. Guidance system for autonomous surface vehicles. Master's thesis, Norwegian University of Science and Technology, 2015.

[56] D. Stormont. Analyzing human trust of autonomous systems in hazardous environments. *Association for the Advancement of Artificial Intelligence*, 01 2008.

[57] C. Tam, R. Bucknall, and A. Greig. Review of collision avoidance and path planning methods for ships in close range encounters. In *The Journal of Navigation*, volume 62, pages 455–476, 2009.

[58] C. S. Tan, R. Sutton, and J. Chudley. Collision avoidance systems for autonomous underwater vehicles part b: A review of obstacle avoidance. *Journal of Marine Science and Environment*, 2004b.

[59] J. van den Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, May 2008.

[60] J. van den Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin. Interactive navigation of multiple agents in crowded environments. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, I3D '08, pages 139–147, New York, NY, USA, 2008. ACM.

[61] T. J. J. van den Boom and T. C. P. M. Backx. Model predictive control, 2005.

[62] E. Wilthil, A. Flåten, and B. E.F. A target tracking system for asv collision avoidance based on the pdaf. *Sensing and Control for Autonomous Vehicles*, pages 269–288, 2017.

[63] K. Woerner. Multi-contact protocol-constrained collision avoidance for autonomous marine vehicles. ph.d. thesis. *Massachusetts Institute of Technology*, 2016.