Anette Uttisrud

# Hybrid collision avoidance for autonomous passenger ferries

## Combining the multiple-path velocity planner and the branching-course model predictive control algorithm

Master's thesis

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

Master's thesis in Cybernetics and Robotics

Supervisor: Morten Breivik, Bjørn-Olav Holtung Eriksen, and Emil Hjelseth Thyri

December 2019

NTNU
Norwegian University of Science and Technology

Anette Uttisrud

# Hybrid collision avoidance for autonomous passenger ferries

## Combining the multiple-path velocity planner and the branching-course model predictive control algorithm

Master's thesis in Cybernetics and Robotics
Supervisor: Morten Breivik, Bjørn-Olav Holtung Eriksen, and Emil Hjelseth Thyri
December 2019

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

NTNU
Norwegian University of Science and Technology

# Abstract

The ongoing urbanization raises an increasing need for sustainable passenger transport in the cities. Electrical autonomous passenger ferries can offer a green and adaptable transport solution that can easily be integrated into urban water areas. To achieve the autonomy of a vessel, the detection of the surrounding world, the understanding of it, and the actions made are critical. Besides avoiding collisions, the behavior of an autonomous ferry should comply with passenger comfort and the international regulations for preventing collisions at sea (COLREGs).

This Master's thesis presents a hybrid collision avoidance (COLAV) system, combining the multiple-path velocity planner (MP-VP) and the branching-course model predictive control (BC-MPC) algorithm. The COLAV system is designed to be versatile and should work in both confined environments and at the open sea. Consideration of the COLREGs rules 8 and 13-17, which deals with maneuvering, is also included.

In the hybrid COLAV system, a high-level path planner generates a set of parallel paths that are collision-free of static obstacles. The MP-VP generates a reference trajectory along those paths considering the COLREGs and a predicted future of the moving obstacles. The BC-MPC algorithm serves as a trajectory-tracker and short-term COLAV system. The maneuvers planned by the BC-MPC algorithm complies with the COLREGs and are feasible to the dynamic constraints of the ferry.

To integrate the MP-VP and the BC-MPC algorithm into a common hybrid COLAV system, both algorithms are modified. Consideration of the COLREGs rules 8 and 13-17 is added to the MP-VP. The BC-MPC algorithm is parametrized in 3 degrees of freedom and adapted for a better appliance with the vessel model and velocity ranges. A dynamic representation of moving obstacles are developed for both algorithms, allowing the ferry to pass closer to moving obstacles in confined areas than at the open sea. To complete the hybrid COLAV system, a supervisor that invokes the MP-VP when necessary is implemented.

The hybrid COLAV system is tested through simulations using a vessel model of the milliAmpere ferry test platform. The simulations include multi-obstacle scenarios, also involving speed-maneuvering obstacles. Furthermore, the performance of the hybrid COLAV system is compared with stand-alone COLAV systems of the MP-VP and the BC-MPC algorithm. The results are evaluated using quantitative performance metrics based on an identified set of desirable properties for autonomous passenger ferry COLAV systems. The hybrid COLAV system benefits from both of the algorithms and avoids collisions successfully. When the moving obstacles are maneuvering, the ferry executes necessary maneuvers. Otherwise, the ferry follows the reference trajectory generated by the MP-VP.

# Sammendrag

Den pågående urbaniseringen stiller økende krav til bærekraftig passasjertransport i byene. Elektriske autonome passasjerferger tilbyr en grønn og tilpasningsdyktig transportløsning som enkelt kan integreres i urbane farvann. Deteksjon og forståelse av omgivelsene er kritisk for autonome fartøy, og fartøyet må videre respondere korrekt på omgivelsene og utføre nødvendige manøvrer. Kollisjoner må unngås, og oppførselen bør sikre passasjerkomfort og være i henhold til konvensjonen om internasjonale regler til forebygging av sammenstøt på sjøen (COLREGs).

Denne masteroppgaven presenterer et hybrid kollisjonsunngåelses-system (COLAV-system) som kombinerer hastighetsplanlegging og modellprediktiv kontroll gjennom metodene multiple-path velocity planner (MP-VP) og branching-course model predictive control (BC-MPC). COLAV-systemet er designet for å være fleksibelt, og skal kunne fungere i både begrensede områder og på åpen sjø, samtidig som det tar hensyn til reglene 8 og 13-17 i COLREGs, som omhandler manøvrering.

Det hybride COLAV-systemet består av en høynivå baneplanlegger som genererer et sett parallelle baner som er kollisjonsfrie for statiske hindringer. MP-VP planlegger en referanserute langs disse banene, som er kollisjonsfri for en predikert fremtid for bevegelige hindringer og følger COLREGs. Deretter fungerer BC-MPC algoritmen som en ruteføl ger og et umiddelbart COLAV-system. Den beregner en hastighetsbane til fartøyskontrollerne som følger referanseruten og er kollisjonsfri i den faktiske situasjonen.

For å integrere MP-VP og BC-MPC algoritmene i et felles hybrid COLAV-system, er begge algoritmene modifiserte. MP-VP er endret slik at det tas hensyn til COLREGs Regel 8 og 13-17. BC-MPC algoritmen er parametrisert i tre frihetsgrader, og tilpasset fartøysmodellen og de tilgjengelige hastighetene til fergen. En dynamisk representasjon av bevegelige hindringer er utviklet for begge algoritmene, slik at fergen har mulighet til å passere nærmere hindringene i begrensede områder enn på åpen sjø. Videre er det implementert en modul som sørger for at MP-VP planlegger referanseruten på nytt når nødvendig.

Det hybride COLAV-systemet er testet gjennom simuleringer med en matematisk fartøysmodell av test-plattformen milliAmpere. Simuleringene omfatter ulike scenario med flere bevegelige hindringer, også der noen av hindringene gjør fartsmanøvrer. Videre er prestasjonen til det hybride COLAV-systemet sammenlignet med to COLAV-system som benytter henholdsvis MP-VP og BC-MPC algoritmene. Resultatene er evaluert og diskutert basert på kvantitative metrikker valgt ut i fra et sett med identifiserte ønskelige egenskaper for et COLAV-system på en autonom passasjerferge. Det hybride COLAV-systemet drar nytte av egenskapene til både MP-VP og BC-MPC algoritmene. Fergen unngår kollisjoner, utøver unnamanøvrer når nødvendig og følger ellers referanseruten.

# Preface

This Master's thesis is the culmination of my work conducted during 2019 and is the finalization of my Master's degree in Cybernetics and Robotics at the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU). I have had the pleasure of working with collision avoidance (COLAV) systems for autonomous vessels, and it has been both challenging and inspiring. The Master's thesis has been supervised by Morten Breivik, Bjørn-Olav Holtung Eriksen, and Emil Hjelseth Thyri, and builds upon my project thesis written during the spring of 2019, concerning COLAV for autonomous surface vehicles with the branching-course model predictive control (BC-MPC) algorithm [1].

I would like to express my gratitude to the supervisors for valuable guidance, discussions, and feedback. I would also like to thank my friends for the memorable and enjoyable years in Trondheim.

Previous work, provided material, and recieved support for the work of this Master's thesis is summarized as:

- The supervisors have participated in meetings every fortnight, where the progress of the thesis work and relevant topics were discussed.
- Literature concerning the BC-MPC algorithm, the multiple-path velocity planner (MP-VP), and the milliAmpere test platform were provided by the supervisors. The BC-MPC articles [2] [3] were provided during the spring of 2019, for the work of my project thesis. The literature concerning the MP-VP [4] [5] and the milliAmpere test platform [6] [7] were provided during the fall of 2019.
- Parts of the COLAV literature and theoretical background are similar as for the project thesis, and sections 1.2, 2.1-2.4, and 2.5.2 are based on [1].
- The implementation of BC-MPC in this Master's thesis is based on my own code implemented for the project thesis.
- Emil Hjelseth Thyri provided a complete implementation of the MP-VP and reference filters in MATLAB and SIMULINK. Only minor changes were necessary for the code to work with the interface of the simulator.
- The vessel model parameters were determined by Anders A. Pedersen during the academic year of 2018-2019 [6].

Anette Uttisrud
Trondheim, December 2019

# Table of contents

# List of Tables

# List of Figures

# Abbreviations

**ASV** autonomous surface vehicle

**AUV** autonomous underwater vehicle

**BC-MPC** branching-course model predictive control

**COG** course over ground

**COLAV** collision avoidance

**COLREGs** international regulations for preventing collisions at sea

**CPA** closest point of approach

**DCO** distance to the closest obstacle

**DOF** degrees of freedom

**DP** dynamic positioning

**DW** dynamic window

**HPR** high penalty region

**IASR** integral of absolute speed rate

**IAYR** integral of absolute yaw rate

**IMO** International Maritime Organization

**IW** integral of power consumption

**LOS** line of sight

**LPR** low penalty region

**MDO** minimum distance to obstacle

**MP-VP** multiple-path velocity planner

**MPC** model predictive control

**MR** margin region

**NED** north-east-down

**PID** proportional integral derivative

**ROC** region of collision

**ROO** region of observation

**RRT** rapidly exploring random tree

**SNAME** Society of Naval Architects and Marine Engineers

**SOG** speed over ground

**SP-VP** single-path velocity planner

**SR** safety region

**TT** travel time

**USV** unmanned surface vehicle

**VO** velocity obstacle

**VPP** velocity planning problem

# Chapter 1

# Introduction

## 1.1 Motivation

The world is changing rapidly and will face significant challenges in the future. More than half of the population worldwide are already living in urban areas, and the number is increasing [8]. The urbanization raises requirements for infrastructure capacity, including transport. At the same time, it is essential to ensure sustainable development in compliance with the international sustainability goals and the increasing focus on nature preservation.

Many cities are located on the coast, with harbors or rivers as an integrated part of the city center. Passenger ferries take advantage of the water as a traffic area and offer an adaptable transport solution. They can be used for short, straight-line crossings, so-called water elevators, and longer distances serving as water buses. Hence, autonomous passenger ferries can replace bridges, tunnels, and cabled ferries, which requires more permanent infrastructure and higher installation costs. Furthermore, autonomous, electrical ferries have lower operational costs than manned ferries and produce less emission to the local environment compared with diesel or gas-driven vehicles.

The milliAmpere test platform is developed and used by the Norwegian University of Science and Technology (NTNU) for testing of technology towards a fully autonomous passenger ferry for the transport of people in urban water channels. Figure 1.1 shows an illustration of the intended full-scale ferry and the milliAmpere test platform, which is about half the size. The research includes topics as electrical power and propulsion, sensor systems, collision avoidance, and cyber security.

The success of an autonomous vehicle depends heavily on the detection of the surrounding world, the understanding of it, and the actions made. The collision avoidance (COLAV) system addresses the latter, making maneuvers to avoid collisions with obstacles. Urban ferries operate in high-traffic environments, and a

**(a)** Illustration of the full-scale autonomous passenger ferry.
Courtesy of Petter Mustvedt.

**(b)** The milliAmpere test platform.
Photo: Kai T.Dragland.

**Figure 1.1:** An illustration of the full-scale ferry and the miliAmpere test platform.

COLAV system must handle complex and dynamic situations. Additionally, the maneuvers should be taken considering the "rules of the road" at sea, the international regulations for preventing collisions at sea (COLREGs).

## 1.2 Previous work

A range of COLAV methods is developed over the last decades, many of which are applied for marine vessels. COLAV methods can briefly be divided into reactive and deliberate methods. Reactive methods have limited information and act based on the current state and environment, which gives fast algorithms with low computationally cost. Deliberate methods consider more information, but at the cost of a more computationally heavy algorithm.

Examples of reactive methods are dynamic window (DW) and velocity obstacles (VOs) searching in the velocity space, and the directional potenial field methods. DW was introduced in 1997 as a collision avoidance method for ground vehicles [9], and is modified and applied for autonomous underwater vehicles (AUVs) [10] and autonomous surface vehicles (ASVs) [11]. VOs was introduced in 1998 [12], and an example of application is to unmanned surface vehicles (USVs) considering COLREGs, given in [13], [14]. Potential field methods treats the robot as a paritcle moving in an artificial force field [15]. The method is developed and improved [16] [17], also for marine vessels considering COLREGs[18].

Rapidly exploring random trees (RRTs) and $A*$ are examples of deliberate methods. RRTs are suited for various planning problems and are designed to handle nonholonomic and dynamic constraints, and high degrees of freedom [19]. $A*$ is an informed heuristic algorithm, typically used for pathfinding [20].

Due to the low computationally cost, reactive algorithms are suited for short-

term local planning, typically avoiding suddenly detected or maneuvering obstacles. There is, however, a risk that the locally optimal solution is a dead-end in the global environment, and reactive algorithms can not guarantee convergence towards the goal. Deliberate algorithms are, on the other hand, suitable for long term global planning. A hybrid architecture combining different algorithms can be used to achieve a satisfying COLAV system. An example of application is a combination of RRTs, $A^*$, and modified DW for USVs [21].

Model predictive control (MPC) permits the use of dynamic models of the system to optimize a finite time horizon while satisfying a set of constraints. Applications are within a wide range of motion control, also for marine vessels [22] [23] [2]. The branching-course model predictive control (BC-MPC) algorithm is designed as a short-term COLAV algorithm for marine vessels, to be robust towards noisy estimates of obstacles and to keep the typical marine maneuvers and the COLREGs in mind [2]. Successful simulations and full-scale tests in the Trondheimsfjord are demonstrated [2]. The algorithm is extended to include the consideration of static obstacles [3] and modified to achieve fewer oscillations in turns [24].

The BC-MPC algorithm is proposed as a part of a three-level hybrid architecture, with high-level, mid-level, and short-term COLAV [2]. A three-layered hybrid COLAV system with a high-level trajectory planner generating an energy-optimized trajectory, a mid-level MPC-based COLAV layer considering the COLREGs and moving obstacles, and the BC-MPC algorithm for short-term COLAV is proposed and evaluated through simulations in [24].

Considering passenger ferries, Wärtsilä and Rolls-Royce have both demonstrated successful automated dock-to-dock transits in 2018, which of Rolls-Royce with a COLAV system [25]. The research and development of the field are moving forward, and fully autonomous systems are expected to be in operation within a few years.

A COLAV system based on path velocity decomposition [26] considering a velocity planning problem (VPP) on predefined paths were designed, implemented, and evaluated through simulations and full-scale tests using the milliAmpere test platform during the spring of 2019 [4]. The velocity planner were applied both on a single path (the single-path velocity planner (SP-VP)) and on multiple paths (the multiple-path velocity planner (MP-VP)). The full-scale tests were performed with simulated maneuvering obstacles, and the methods show satisfying results in high-traffic, confined environments [5].

## 1.3 Problem description

The scope of this thesis is COLAV for autonomous passenger ferries, in particular:

- Get familiar with existing COLAV systems for autonomous passenger ferries.
- Investigate which properties are desirable for autonomous passenger ferry COLAV systems.

- Parametrize and adapt the BC-MPC algorithm to work with 3 degrees of freedom (DOF) autonomous passenger ferries.
- Create a hybrid COLAV system combining the multiple-path velocity planner (MP-VP) and the adapted BC-MPC algorithm. This includes necessary modifications to the algorithms and an interface between the algorithms.
- Perform simulations with moving obstacles and evaluate the hybrid COLAV system. Investigate the suitability of the hybrid COLAV system for water elevator and water bus applications.
- Compare the hybrid COLAV system with stand-alone COLAV systems using only the MP-VP or the BC-MPC algorithm through simulations. The COLAV systems should be evaluated with a set of quantitative performance metrics considering the identified desirable properties of autonomous passenger ferry COLAV systems.

## 1.4 Contributions

The contributions of this Master's thesis are considered to be

- A set of desirable properties for autonomous passenger ferry COLAV systems are identified.
- A hybrid COLAV system combining the MP-VP and the BC-MPC algorithm is developed. This includes an interface between the algorithms and a supervisor that invokes the MP-VP when necessary.
- For the MP-VP, a dynamic obstacle representation considering the size of the operating area is developed, keeping the maneuvering COLREGs rules 8 and 13-17 in mind.
- The BC-MPC algorithm is parametrized in 3 DOF and adapted to the vessel model of the milliAmpere test platform. Sway motions are included as emergency maneuvers in a separate emergency search space. Similarly as for the MP-VP, a dynamic representation of moving obstacles is developed for the BC-MPC algorithm.
- A complete simulation environment is implemented and integrated in MATLAB, reusing some implementations of the MP-VP and the BC-MPC algorithm.
- The hybrid COLAV system is evaluated through simulations of both water elevator and water bus scenarios. The simulated scenarios include multi-obstacle traffic pictures with both speed-maneuvering and non-maneuvering obstacles.
- The performance of the hybrid architecture is compared with stand-alone COLAV systems of both the MP-VP and the BC-MPC algorithm through simulations. The algorithms are evaluated using quantitative performance

metrics defined based on the identified desirable properties of autonomous passenger ferry COLAV systems.

## 1.5 Outline

Chapter 2 gives the necessary theoretical background for this Master's thesis, including a summary of the relevant COLREGs rules, and detailed descriptions of the original MP-VP and BC-MPC algorithms. Chapter 3 describes the hybrid COLAV system, including the identified desirable properties, the architecture and the modifications made to MP-VP and BC-MPC. Chapter 4 presents the implemented simulation environment and the simulation results. Chapter 5 gives conluding remarks and suggestions for further work.

# Chapter 2

# Theoretical background

This chapter gives relevant background theory. Section 2.1 describes vessel modeling, and Section 2.2 gives guidance and motion control theory. Section 2.3 describes the closest point of approach (CPA), and the relevant rules of the COLREGs is summarized in Section 2.4. Finally, Section 2.5 explaines the COLAV methods used in this Master's thesis.

## 2.1 Vessel modeling

The dynamics of a vessel are described by the motions caused by geometry, kinematics, and the motions caused by forces, kinetics [27]. A vessel can translate and rotate independently in three dimensions and has up to six DOF. This section describes the kinematics and kinetics of a vessel, relevant reference frames, and notation.

### 2.1.1 Kinematics

The position and orientation, or pose, motions and forces of a vessel are described relative to reference frames. In this thesis, the earth-fixed north-east-down (NED) and the vessel-fixed body frame is used.

The NED frame $\{n\} = (x_n, y_n, z_n)$ is rotated such that $x_n$ points towards true north, $y_n$ points towards true east and $z_n$ points downwards normal to the Earth's surface. A NED-frame fixed to a point on the Earth's surface is often used as the reference frame for local navigation, with $\{n\}$ assumed to be inertial such that Newton's laws apply.

The body frame $\{b\} = (x_b, y_b, z_b)$ is fixed to the craft and is useful to describe velocities and forces acting on the ship. The origin $o_b$ is usually located midships

in the waterline. The $x_b$ axis points forward in the longitudinal direction, $y_b$ is the transversal axis pointing towards starboard, and $z_b$ is directed downwards and completes the right-handed system.

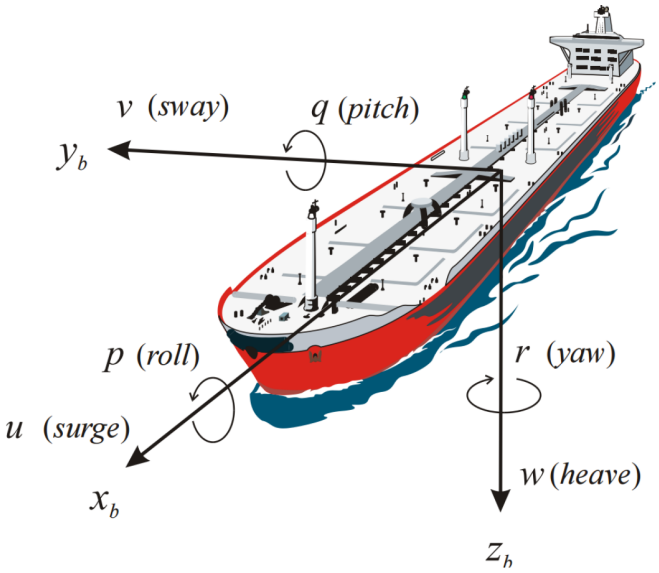



**Figure 2.1:** Motion in 6 DOF [27].

A transformation between $\{b\}$ and $\{n\}$ can be carried out by a rotation

$$\boldsymbol{v^n} = \boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}})\boldsymbol{v^b} \tag{2.1}$$

where $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$ are the Euler angles, and $\boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}})$ is a combination of three principal rotations, one about each axis,

$$\boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi}. \tag{2.2}$$

The reverse transformation is the inverse or the transposed rotation matrix

$$\boldsymbol{R_b^n}(\boldsymbol{\Theta_{nb}})^{-1} = \boldsymbol{R_b^n}^\top(\boldsymbol{\Theta_{nb}}) = \boldsymbol{R_n^b}(\boldsymbol{\Theta_{nb}}) = \boldsymbol{R}_{z,\psi}^\top\boldsymbol{R}_{y,\theta}^\top\boldsymbol{R}_{x,\phi}^\top. \tag{2.3}$$

Figure 2.1 illustrates the 6 DOF of a marine vessel, and Table 2.1 lists the commonly used notation of the Society of Naval Architects and Marine Engineers (SNAME) [28].

**Table 2.1:** SNAME notation for marine vessels

| Motion | Positions and Euler angles | Linear and angular velocities | Forces and moments |
|---|---|---|---|
| Motion in the $x$-direction (surge) | $x$ | $u$ | $X$ |
| Motion in the $y$-direction (sway) | $y$ | $v$ | $Y$ |
| Motion in the $z$-direction (heave) | $z$ | $w$ | $Z$ |
| Rotation about the $x$-axis (roll) | $\phi$ | $p$ | $K$ |
| Rotation about the $y$-axis (pitch) | $\theta$ | $q$ | $M$ |
| Rotation about the $z$-axis (yaw) | $\psi$ | $r$ | $N$ |

The pose, $\boldsymbol{\eta}$, the velocity vector $\boldsymbol{\nu}$, and the force vector $\boldsymbol{\tau}$ becomes

$$\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^\top, \tag{2.4}$$

$$\boldsymbol{\nu} = [u, v, w, p, q, r]^\top, \tag{2.5}$$

$$\boldsymbol{\tau} = [X, Y, Z, K, M, N]^\top, \tag{2.6}$$

and are usually given in $\{n\}$, $\{b\}$ and $\{b\}$, respectively.

The horizontal speed, speed over ground (SOG), is defined by the surge speed $u$ and the sway speed $v$ as

$$U = \left\| \begin{bmatrix} u \\ v \end{bmatrix} \right\|_2, \tag{2.7}$$

and the course over ground (COG) as

$$\chi = \psi + \beta, \tag{2.8}$$

where $\beta$ the sideslip, and $\|\boldsymbol{a}\|_2$ is the Euclidean norm of $\boldsymbol{a} = [a_1, \ldots, a_n]^\top$ given by

$$\|\boldsymbol{a}\|_2 = \sqrt{a_1^2, \ldots, a_n^2}. \tag{2.9}$$

Figure 2.2 illustrates SOG and COG. In this thesis, speed refers to SOG and course to COG.

### 2.1.2 Kinetics

The equations of motion in 6 DOF can be written on a vectorial form

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{2.10}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave}, \tag{2.11}$$

where $\boldsymbol{\eta}, \boldsymbol{\nu}$ and $\boldsymbol{\tau}$ is equal to (2.4)-(2.5). The external forces from wind and waves are denoted as $\boldsymbol{\tau}_{wind}$ and $\boldsymbol{\tau}_{wave}$, and the inertia, coriolis and damping matrices as

$\boldsymbol{M}, \boldsymbol{C}(\boldsymbol{\nu})$ and $\boldsymbol{D}(\boldsymbol{\nu})$, respectively. The generalized gravitation and boyancy forces are denoted as $\boldsymbol{g}(\boldsymbol{\eta})$, and the static restoring forces and moments as $\boldsymbol{g}_0$.

For most vessels, $\phi$ and $\theta$ are small, and $\boldsymbol{R}_b^n\left(\boldsymbol{\Theta}_{nb}\right) \approx \boldsymbol{R}_{z,\psi}$. The model is commonly reduced to a 3 DOF model in two dimensions neglecting heave, roll, and pitch:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.12}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{2.13}$$

where $\boldsymbol{\eta} = [N, E, \psi]^\top$, $\boldsymbol{\nu} = [u, v, r]^\top$, $\boldsymbol{\tau} = [X, Y, N]^\top$ [27].

### 2.1.3 milliAmpere vessel model

The milliAmpere experimental test platform is a prototype of a small autonomous passenger ferry designed and developed by the Norwegian University of Science and Technology (NTNU). Several vessel models of the milliAmpere experimental test platform were developed as Master's thesis during the academic year of 2018-2019 by Anders A. Pedersen [6]. In this thesis, a 3 DOF coupled model identified with optimal control theory is used. The model is given by

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}\left(\boldsymbol{\nu}\right)\boldsymbol{\nu} + \boldsymbol{D}\left(\boldsymbol{\nu}\right)\boldsymbol{\nu} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave}, \tag{2.14}$$

where $\boldsymbol{\nu}, \dot{\boldsymbol{\nu}}$ are the rigid body velocity and acceleration in $\{b\}$, and $\boldsymbol{M}, \boldsymbol{C}(\boldsymbol{\nu}), \boldsymbol{D}(\boldsymbol{\nu})$ are the inertia, centripetal and damping matrices. The control input is given by $\boldsymbol{\tau}$ and the external forces are given as $\boldsymbol{\tau}_{wind}, \boldsymbol{\tau}_{wave}$.

The inertia matrix

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A \tag{2.15}$$

consists of the rigid body mass $\boldsymbol{M}_{RB}$ and the added body mass $\boldsymbol{M}_A$:

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \; \boldsymbol{M}_A = \begin{bmatrix} -X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{r}} \\ -X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{r}} \\ -X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{r}} \end{bmatrix}. \tag{2.16}$$

The parameter identification is however done without separating the inertia matrix, giving

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}. \tag{2.17}$$

The centripetal matrix is skew-symmetric and given by $\boldsymbol{M}$ and $\boldsymbol{\nu}$:

$$\boldsymbol{C}(\boldsymbol{\nu}) = -\boldsymbol{C}(\boldsymbol{\nu})^\top, \tag{2.18}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & c_{13}(\boldsymbol{\nu}) \\ 0 & 0 & c_{23}(\boldsymbol{\nu}) \\ c_{31}(\boldsymbol{\nu}) & c_{32}(\boldsymbol{\nu}) & 0 \end{bmatrix}, \tag{2.19}$$

$$c_{13}(\boldsymbol{\nu}) = -m_{12}u - m_{22}v - m_{23}r, \tag{2.20a}$$
$$c_{23}(\boldsymbol{\nu}) = m_{11}u + m_{12}v + m_{13}r, \tag{2.20b}$$
$$c_{31}(\boldsymbol{\nu}) = -c_{13}(\boldsymbol{\nu}), \tag{2.20c}$$
$$c_{32}(\boldsymbol{\nu}) = -c_{23}(\boldsymbol{\nu}). \tag{2.20d}$$

The hydrodynamic damping matrix $\boldsymbol{D}(\boldsymbol{\nu}) > 0$ is defined as

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} d_{11}(\boldsymbol{\nu}) & d_{12} & d_{13} \\ d_{21} & d_{22}(\boldsymbol{\nu}) & d_{23}(\boldsymbol{\nu}) \\ d_{31} & d_{32}(\boldsymbol{\nu}) & d_{33}(\boldsymbol{\nu}) \end{bmatrix}, \tag{2.21}$$

where

$$d_{11}(\boldsymbol{\nu}) = -X_u - X_{|u|u}|u| - X_{uuu}u^2 \tag{2.22a}$$
$$d_{12} = -X_v \tag{2.22b}$$
$$d_{13} = -X_r \tag{2.22c}$$
$$d_{21} = -Y_u \tag{2.22d}$$
$$d_{22}(\boldsymbol{\nu}) = -Y_v - Y_{|v|v}|v| - Y_{|r|v}|r| - Y_{vvv}v^2 \tag{2.22e}$$
$$d_{23}(\boldsymbol{\nu}) = -Y_r - Y_{|v|r}|v| - Y_{|r|r}|r| \tag{2.22f}$$
$$d_{31} = -N_u \tag{2.22g}$$
$$d_{32}(\boldsymbol{\nu}) = -N_v - N_{|v|v}|v| - N_{|r|v}|r| \tag{2.22h}$$
$$d_{33}(\boldsymbol{\nu}) = -N_r - N_{|v|r}|v| - N_{|r|r}|r| - N_{rrr}r^2. \tag{2.22i}$$

The identified parameters for the fully-coupled 3 DOF model is listed in Table 2.2.

**Table 2.2:** Parameters for the fully-coupled 3 DOF model [6].

| Parameter | Value | Unit |
|---|---|---|
| $m_{11}$ | 2389.173 | $kg$ |
| $m_{12}$ | -12.536 | $kg$ |
| $m_{13}$ | 39.776 | $kgm$ |
| $m_{21}$ | 27.147 | $kg$ |
| $m_{22}$ | 2530.602 | kg |
| $m_{23}$ | -20.612 | $kgm$ |
| $m_{31}$ | 112.965 | $kgm$ |
| $m_{32}$ | -0.606 | $kgm$ |
| $m_{33}$ | 5068.800 | $kgm^2$ |
| $X_u$ | -27.408 | $\frac{kg}{s}$ |
| $X_{\|u\|u}$ | -107.555 | $\frac{kg}{m}$ |
| $X_{uuu}$ | -14.874 | $\frac{kgs}{m^2}$ |
| $X_v$ | 39.398 | $\frac{kg}{s}$ |
| $X_r$ | 104.568 | $\frac{kgm}{s}$ |
| $Y_u$ | -45.036 | $\frac{kg}{s}$ |

| Parameter | Value | Unit |
|---|---|---|
| $Y_v$ | -61.927 | $\frac{kg}{s}$ |
| $Y_{\|v\|v}$ | -84.895 | $\frac{kg}{m}$ |
| $Y_{vvv}$ | -45.394 | $\frac{kg}{s^2}$ |
| $Y_{\|r\|v}$ | -1475.115 | $kg$ |
| $Y_r$ | 35.525 | $\frac{kg}{s}$ |
| $Y_{\|v\|r}$ | 546.700 | $\frac{kg}{s}$ |
| $Y_{\|r\|r}$ | -60.848 | $\frac{kg}{s}$ |
| $N_u$ | 41.789 | $\frac{kg}{s}$ |
| $N_v$ | 16.464 | $\frac{kg}{s}$ |
| $N_{\|v\|v}$ | -18.002 | $\frac{kg}{s}$ |
| $N_{\|r\|v}$ | 320.144 | $\frac{kg}{s}$ |
| $N_r$ | -120.483 | $\frac{kg}{s}$ |
| $N_{\|r\|r}$ | -870.050 | $\frac{kg}{s}$ |
| $N_{rrr}$ | 0.000 | $\frac{kg}{s}$ |
| $N_{\|v\|r}$ | -271.946 | $\frac{kg}{s}$ |

## 2.2 Guidance and motion control

A guidance system is used to calculate the desired states, usually speed and course, for a moving vessel following a target, path or trajectory. The motion controllers calculates a control input for the vessel thrusters, with the aim of driving the vessel state to the desired state.

### 2.2.1 Path following and trajectory tracking

A piecewise linear path is typically defined by $n$ waypoints $\boldsymbol{p} = \left[\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\right]$ where $\boldsymbol{p}_i = \left[N_i, E_i\right]$ for all $i = \{1, \ldots, n\}$. A path is time-invariant, and the object of path following is to track the path with no temporal constraints. A trajectory is a time parametrized path, and forces the system output $\boldsymbol{y}(t)$ to track the trajectory $\boldsymbol{y}_d(t)$ [27].

#### Absolute and relative tracking

Relative tracking adjusts the desired trajectory by adding a time delay $t_b$ if the vessel lags behind the reference, avoiding the desired trajectory to "run away." The relative desired trajectory is

$$\bar{\boldsymbol{p}}(t) = \boldsymbol{p}_d\left(t + t_b\right). \tag{2.23}$$

The time delay is calculated by minimizing the eucleadian distance between $\boldsymbol{p}(t_0)$ and $\bar{\boldsymbol{p}}(t_0)$ by solving the optimization problem

$$t_b(t_0) = \underset{t_b}{\operatorname{argmin}} \left\|\boldsymbol{p}_d(t_0 + t_b) - \boldsymbol{p}(t_0)\right\|_2, \tag{2.24}$$

by a simple line search algorithm [29]. Absolute tracking is simply to track the trajectory without a time delay.

### 2.2.2 Line of sight (LOS) guidance

Line of sight (LOS) guidance is a three-point scheme, illustrated in Figure 2.2. It typically takes a stationary reference point $\boldsymbol{p}_t = \left[N_t, E_t\right]^\top$, a target $\boldsymbol{p}_{t+1} = \left[N_{t+1}, E_{t+1}\right]^\top$ and the interceptor with the position $\boldsymbol{p}(t)$, all in $\{n\}$. The control objective is formulated as [30]

$$\lim_{t\to\infty} \left[\boldsymbol{p}(t) - \boldsymbol{p}_{t+1}(t)\right] = \boldsymbol{0}. \tag{2.25}$$

The path is intercepted from $\boldsymbol{p}_0$ to $\boldsymbol{p}_t$ along the LOS vector from the interceptor to a point $(x_{LOS}, y_{LOS})$ on the path. For lookahead-based steering, the desired course is calculated based on the cross-track error $e$ and the lookahead-distance

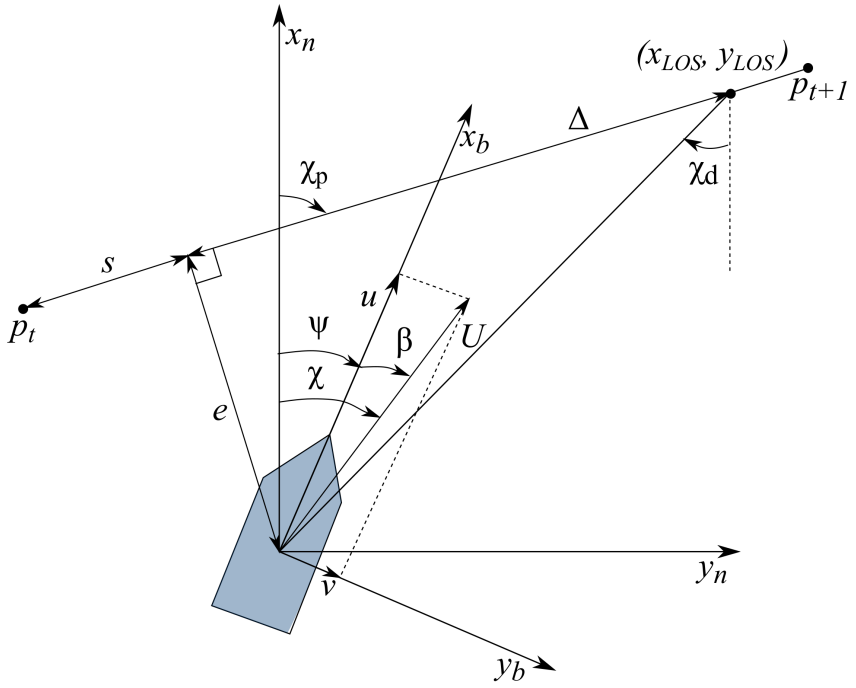



**Figure 2.2:** Illustration of LOS guidance.

$\Delta(t) > 0$, which decides how fast the convergence towards the LOS vector should be. The desired yaw angle $\chi_d$ is

$$\chi_d(e) = \chi_p + \chi_r(e), \tag{2.26}$$

where $\chi_p = \alpha_k$ is the path-tangential angle and $\chi_r(e)$ the velocity-path relative angle given by

$$\chi_r(e) := \arctan\Big(-\frac{e}{\Delta}\Big). \tag{2.27}$$

### 2.2.3 Motion control

A controller calculates an input to the actuators with the goal of driving the vessel state to a desired state. The proportional integral derivative (PID) controller uses feedback of the current state, and the input is given by

$$\boldsymbol{\tau}_{PID} = -\boldsymbol{K}_p\boldsymbol{e} - \boldsymbol{K}_i\int_0^t \boldsymbol{e}d\tau - \boldsymbol{K}_d\dot{\boldsymbol{e}} \tag{2.28}$$

$$\boldsymbol{e} = \boldsymbol{R}^\top(\psi)(\boldsymbol{\eta} - \boldsymbol{\eta}_d) \tag{2.29}$$

$$\dot{\boldsymbol{e}} = \boldsymbol{\nu} - \boldsymbol{\nu}_d. \tag{2.30}$$

where the gain matrices $\boldsymbol{K}_p, \boldsymbol{K}_i, \boldsymbol{K}_d > 0 \in \mathbb{R}^{3x3}$ are diagonal

$$\boldsymbol{K}_p = \begin{bmatrix} k_{p_{11}} & 0 & 0 \\ 0 & k_{p_{22}} & 0 \\ 0 & 0 & k_{p_{33}} \end{bmatrix}, \boldsymbol{K}_i = \begin{bmatrix} k_{i_{11}} & 0 & 0 \\ 0 & k_{i_{22}} & 0 \\ 0 & 0 & k_{i_{33}} \end{bmatrix}, \boldsymbol{K}_d = \begin{bmatrix} k_{d_{11}} & 0 & 0 \\ 0 & k_{d_{22}} & 0 \\ 0 & 0 & k_{d_{33}} \end{bmatrix}. \tag{2.31}$$

A feedforward control law uses the vessel model to calculate the control input based ithe desired state, and is given by

$$\boldsymbol{\tau}_{FF} = \boldsymbol{M}\boldsymbol{\nu}_d + \boldsymbol{C}\left(\boldsymbol{\nu}_d\right)\boldsymbol{\nu}_d + \boldsymbol{D}\left(\boldsymbol{\nu}_d\right)\boldsymbol{\nu}_d. \tag{2.32}$$

## 2.3 Closest point of approach (CPA)

The time and distance to the CPA of a moving obstacle with position $\boldsymbol{p}_o$ and velocity $\boldsymbol{v}_o$ in $\{n\}$ can be calculated and used to decide whether a collision situation is likely to happen in near future. The time to CPA, $t_{CPA}$, is given by

$$t_{\text{CPA}} = \begin{cases} 0 & \text{if } \|\boldsymbol{v} - \boldsymbol{v}_o\| \leq \epsilon \\ \frac{(\boldsymbol{p}_o - \boldsymbol{p}) \cdot (\boldsymbol{v} - \boldsymbol{v}_o)}{\|\boldsymbol{v} - \boldsymbol{v}_o\|^2} & \text{otherwise} \end{cases}, \tag{2.33}$$

where $\boldsymbol{p}$ and $\boldsymbol{v}$ are the position and velocity of the ownship in $\{n\}$. The distance to CPA, $d_{CPA}$, by

$$d_{CPA} = \|(\boldsymbol{p} + \boldsymbol{v}t_{CPA}) - (\boldsymbol{p}_o + \boldsymbol{v}_o t_{CPA})\|. \tag{2.34}$$

A collision situation is then likely to happen within $t_{CPA_{max}}$ seconds if

$$0 \leq t_{CPA} \leq t_{CPA_{max}} \ \wedge \ d_{CPA} \leq d_{CPA_{min}}. \tag{2.35}$$

## 2.4 International regulations for preventing collisions at sea (COLREGs)

The international regulations for preventing collisions at sea (COLREGs) were published by International Maritime Organization (IMO) in 1972 and acts as the rules of the road at sea [31]. They establish navigation rules for vessels at sea to prevent collisions and consider both general behavior and specific obligations on how to act in different situations. The rules that are of the most relevance for this thesis are the ones that consider maneuvering, and is summarized from [32] as follows:

**Rule 8: Action to Avoid Collision** Any action to avoid collision shall be taken following the maneuvering rules, and if the circumstances of the case admit, be positive, made in ample time and with due regard to the observance of good seamanship. Any alteration of course and speed shall be large enough to be readily apparent to another vessel, and if there is sufficient sea-room, alteration of course alone may be the most effective action.

**Rule 13: Overtaking** A vessel is overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam. The overtaking vessel should keep clear of the overtaken vessel until she is finally past and clear.

**Rule 14: Head-On Situations** When two power-driven vessels are meeting on (nearly) reciprocal courses, each shall alter her course to starboard so that each shall pass on the port side of each other.

**Rule 15: Crossing Situations** When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

**Rule 16: Action by Keep-Way 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: Actions by Stand-On Vessel** The stand-on vessel shall keep her course and speed. As soon as it becomes apparent that the vessel required to keep out of the way is not taking appropriate action in compliance with the rules, the stand-on vessel shall take action to avoid a collision. If possible, the stand on the vessel should not alter the course to port for a vessel on her port side.

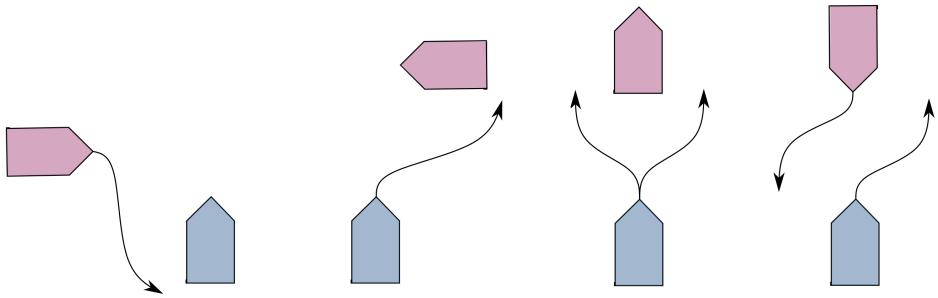


**Figure 2.3:** COLREGs situations seen from ownship, with the keep-way actions of Rules 13-15 indicated. From left to right: crossing from port, crossing from starboard, overtaking, and head-on.

Figure 2.3 illustrates the keep-way obligations of Rule 13-15, seen from the ownship. The complete set of rules includes rules for situations or actions that are not easily

measurable or not relevant to the scope of this thesis. Therefore, for this thesis, COLREGs compliance refers to the rules listed in this section unless otherwise is specified.

## 2.5 Collision avoidance methods

As mentioned in Section 1.2, a COLAV system can be organized in a hybrid architecture utlizing multiple COLAV methods. The hybrid architecture can exists of a high-level, a mid-level and a low-level COLAV system, as proposed in [2], and illustrated in Figure 2.4.

The high-level COLAV system is typically a path or trajectory planner, considering the destination request and static environmental data. The mid-level layer modifies this trajectory, planning collision avoidance maneuvers in ample time before the collision is due to happen, and should, therefore, consider COLREGs. The low-level COLAV considers the immediate situation and moving obstacles, and deliver a feasible velocity trajectory to the vessel controllers.

The levels run at different frequencies, with $F_{hl} < F_{ml} < F_{ll}$. The high-level path or trajectory planner commonly replans only when a new destination is requested. For the mid-layer, a supervisor with COLAV supporting functions can be used to request a new trajectory when the situations change, i.e., when new moving obstacles are detected. The short-time COLAV system should run frequently, to ensure that collisions with suddenly detected or maneuvering obstacles are avoided.

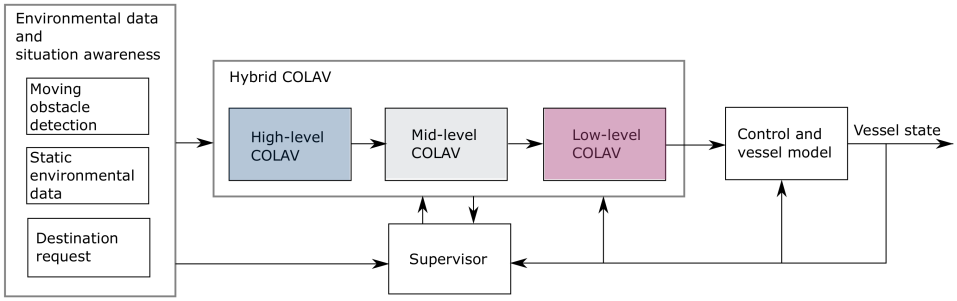



**Figure 2.4:** Collision avoidance architecture overview.

This section describes the two collision avoidance methods considered in this thesis, the velocity planner and the branching-course model predictive control (BC-MPC) algorithm. The MP-VP runs the single-path velocity planner (SP-VP) algorithm along several paths, and the details of SP-VP is therefore not explained explicit in this thesis, as MP-VP is the utilized planner. The rest of this section gives an overview of the MP-VP and the BC-MPC algorithm, before the steps are further explained.

### 2.5.1 Multiple-path velocity planner (MP-VP)

The MP-VP algorithm solves a VPP along a predifined set of paths, assumed to be collision-free of static objects, and returns a trajectory that is collision-free with respect to the predicted future of a set of moving obstacles $\mathcal{O}$ [4]. The VPP is formulated as a node-search problem in the $path \times time$ space, finding the optimal path from a start node $n_s \in \mathcal{N}_\mathcal{S} \subset \mathcal{N}$ to an end node $n_e \in \mathcal{N}_\mathcal{E} \subset \mathcal{N}$ among all nodes $n = [P, t] \in \mathcal{N}$, where $P$ is the distance travelled along the path at the time $t$. The set of nodes is

$$\mathcal{N} = \mathcal{N}_\mathcal{S} \cup \mathcal{N}_\mathcal{E} \cup \mathcal{N}_\mathcal{O} \tag{2.36}$$

where $\mathcal{N}_\mathcal{S}$ is the set of start nodes at the current position, $\mathcal{N}_\mathcal{E}$ is the set of end nodes at the destination, and $\mathcal{N}_\mathcal{O}$ is a set of obstacle nodes in the proximity of the moving obstacles. A set of collision-free edges $\mathcal{E}$, that are feasible with respect to velocity and time constraints, is generated between the nodes. Each edge is assigned a cost according to a cost function, before the node-search problem is solved. Last, a reference trajectory $\boldsymbol{\eta}_d$ in $\{n\}$ is generated from the optimal node-path, with the desired heading aligned with the trajectory. Figure 2.5 shows an overview of the algorithm, and the steps are further explained in the following of this section.

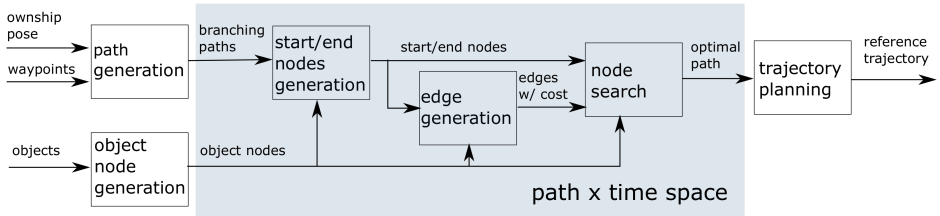



**Figure 2.5:** Overview of the multiple-path velocity planner (MP-VP)

#### Moving obstacle representation

The moving obstacle representation assumes that the size, pose, and velocity of the moving obstacles are known. The obstacle is encapsulated by a region of collision (ROC), a high penalty region (HPR) and a low penalty region (LPR) to improve the robustness towards disturbances and non-constant moving obstacle behavior [4]. Four corners define the regions:

$$co_{fore} = \left[N_o + k_f l_f \cos\left(\psi_o\right), E_o + k_f l_f \sin\left(\psi_o\right)\right], \tag{2.37}$$

$$co_{starboard} = \left[N_o + k_s l_s \cos\left(\psi_o + \frac{\pi}{2}\right), E_o + k_s l_s \sin\left(\psi_o + \frac{\pi}{2}\right)\right], \tag{2.38}$$

$$co_{aft} = \left[N_o + k_a l_a \cos\left(\psi_o + \pi\right), E_o + k_a l_a \sin\left(\psi_o + \pi\right)\right], \tag{2.39}$$

$$co_{port} = \left[N_o + k_p l_p \cos\left(\psi_o + 3\frac{\pi}{2}\right), E_o + k_p l_p \sin\left(\psi_o + 3\frac{\pi}{2}\right)\right], \tag{2.40}$$

where $[N_o, E_o, \psi_o]^\top$ is pose of the moving obstacle, and $k_d l_d$ the axis length for each direction $d \in \{f, s, a, p\}$:

$$k_f l_f = k_f(l_{os} + l_o + l_{R,f}), \tag{2.41}$$
$$k_s l_s = k_s(l_{os} + w_o + l_{R,s}), \tag{2.42}$$
$$k_a l_a = k_a(l_{os} + l_o + l_{R,a}), \tag{2.43}$$
$$k_p l_p = k_p(l_{os} + w_o + l_{R,p}), \tag{2.44}$$

where $(l_o, w_o)$ is the length and width of the moving obstacle, and $l_{os}$ the length of the ownship. The gains for each direction are given by $k_f, k_s, k_a, k_p \geq 0$, respectively, and $l_{R,d}$ is the added length for each region $R = \{ROC, HPR, LPR\}$ in the direction $d$, with $0 \leq l_{ROC,d} \leq l_{HPR,d} \leq l_{LPR,d}$, for all directions $d \in \{f, s, a, p\}$. The moving obstacle representation is illustrated in Figure 2.6.

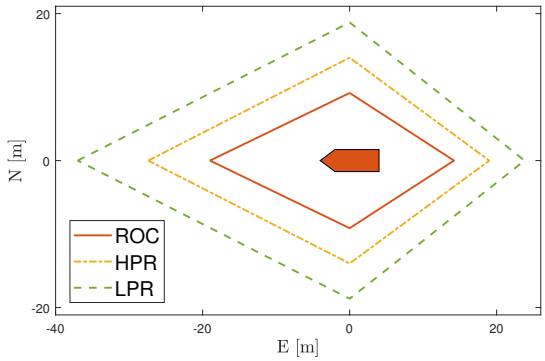



**Figure 2.6:** Moving obstacle representation with the region of collision (ROC), the high penalty region (HPR), and the low penalty region (LPR).

### Transformation to path-time space

A linear path $P$ between the points $P_s = [N_s, E_s]^\top$ and $P_e = [N_e, E_e]^\top$ in $\{n\}$ can be parametrized by

$$P := \frac{N - N_{start}}{a} = \frac{E - E_{start}}{b}, \tag{2.45}$$

for $N \in [N_{start}, N_{end}]^\top$, $E \in [E_{start}, E_{end}]^\top$ by choosing $a$ and $b$ as

$$a = \frac{(N_e - N_s)}{l}, \tag{2.46}$$

$$b = \frac{(E_e - E_s)}{l}, \tag{2.47}$$

where

$$l = \sqrt{(N_{end} - N_{start})^2 + (E_{end} - E_{start})^2}, \tag{2.48}$$

is the length of $P$.

The position $[N(t), E(t)]^\top$ of a point following a path with constant speed $U_o$, course $\chi_o$ and initial position $[N_o, E_o]^\top$ at $t_o = 0$ can be described as a function of time by

$$N(t) = N_o + U_o \cos(\chi_o)(t - t_o), \tag{2.49}$$
$$E(t) = E_o + U_o \sin(\chi_o)(t - t_o). \tag{2.50}$$

Substituting $N(t), E(t)$ from (2.45) gives

$$Pa - N_s = N_o + U_o \cos(\chi_o)(t - t_o), \tag{2.51}$$
$$Pb - E_s = E_o + U_o \sin(\chi_o)(t - t_o), \tag{2.52}$$

which can be solved for every point $[N_o, E_o]^\top$ in $\{n\}$ to get a point $[P, t]$ in the *path* $\times$ *time* space.

The moving obstacles can hence be transformed onto the *path* $\times$ *time* space with the assumption of constant known course $\chi_o$ and speed $U_o$.

### Adding nodes

A set of nodes $\mathcal{N}$ (2.36) is added to the *path* $\times$ *time*. Figure 2.7 illustrates an example of nodes, obstacle regions and edges for a *path* $\times$ *time* space.

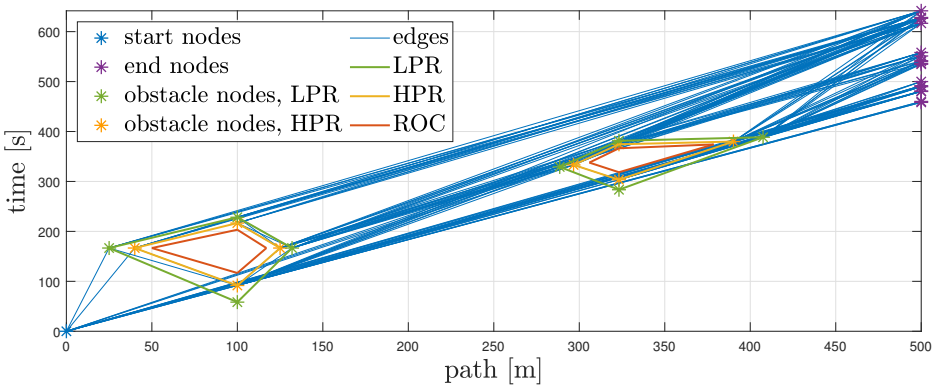



**Figure 2.7:** Example of nodes, moving obstacle representation and edges in the *path* $\times$ *time* space.

Minimum one start node is added at the start position $P = 0$:

$$n_s = [0, t_s], \tag{2.53}$$

where $t_0$ is the current time and $t_0 \leq t_s$. The current time is denoted by $t_0$, and the earliest possible start time for layer $i$, $t_{0_i}$, is equal to the time required to travel

from the current position to the current path, $t_c$, and from the current path to the layer $i$, with a branching angle $\alpha_b$ and speed $U_d$:

$$t_c = \frac{|\epsilon_{n_{cp}}|}{\sin(\alpha_b)U_d} \tag{2.54}$$

$$t_s = t_c + \frac{l_b|c-i|}{\sin(\alpha_b)U_d}, \tag{2.55}$$

where $\epsilon_{n_{cp}}$ is the cross track error of the current path and $c$ is the index of the current layer. Nodes with $t_0 < t_s$ can be added to allow the option of waiting at the start postion before starting the transit at time $t_s$.

A set of end nodes $\mathcal{N}_\mathcal{E}$ at the destination of the transit, with $P$ equal to the path length $l$ is added:

$$n_e = [l, t_e]. \tag{2.56}$$

To ensure that there exists an end node in transit velocity from each node in $\mathcal{N}_\mathcal{S}$ and $\mathcal{N}_\mathcal{O}$, the time $t_e$ can be calculated for all $n \in \mathcal{N}_\mathcal{S} \cup \mathcal{N}_\mathcal{O}$ by

$$t_e = t_n + \frac{l - P_n}{U_{d,T}} \tag{2.57}$$

where $[P_n, t_n]$ is the node $n$ and $U_{d,T}$ is the desired transit velocity.

A set of moving obstacle nodes, $\mathcal{N}_\mathcal{O}$, is added around the moving obstacles to ensure that there exists a collision-free solution to the node search problem. One approach is to transform the corners of HPR and LPR (2.40) onto the $path \times time$ space and add them to $\mathcal{N}_\mathcal{O}$, as illustrated in Figure 2.7.

### Edge generation

A set of feasible directed edges between the nodes in $\mathcal{N}$ is added to connect the start nodes $\mathcal{N}_\mathcal{S}$ with the end nodes $\mathcal{N}_\mathcal{E}$.For an edge from $n_a = [P_a, t_a]$ to $n_b = [P_b, t_b]$ to be feasible, the following criterias are required

1. $n_b$ must be later in time than $n_a$.
2. $n_a$ must be later in time than the current node.
3. The velocity required to travel along the vertex is not greater than the maximum velocity of the ferry.
4. The vertex does not pass through any Regions of Collision

Criterias $1-3$ are trivial to check. For details on how to calculate criteria 4, please refer to [4].

### Edge cost function

The edge cost function for en edge $e$ from $n_a$ to $n_b$ is given by

$$J = cost_{vel} + cost_{time} + cost_{length} + cost_{penalty} + cost_{node}, \tag{2.58}$$

where

$$cost_{vel} = w_{vel}|U_{edge} - U_{d,T}|, \tag{2.59}$$

$$cost_{time} = w_{time} \min\left(\frac{l_p}{|U_{edge}|}, time\_cost_{max}\right) \tag{2.60}$$

$$cost_{length} = w_{length} l_p, \tag{2.61}$$

$$cost_{penalty} = w_{penalty} penalty_{bool}, \tag{2.62}$$

$$cost_{node} = cost_{n_b}, \tag{2.63}$$

The objective terms are weighted by $w_{vel}, w_{time}, w_{length}, w_{penalty} \geq 0$. The desired transit velocity is $U_{d,T}$, and $cost_{n_b}$ is an added node-cost if $n_b \in \mathcal{N}_O$. The boolean value $penalty_{bool}$ is true if the edge passes through a penalty region, which is a line between two HPR nodes at opposite corners. The node search algorithm can be solved by a line-search algorithm, as Dijkstra.

### Multiple path generation

The multiple predefined paths are defined as parallell paths, separated by a separation length $l_s$. Each path is defined by a start point $P_{n,s} = [N_{n,s}, E_{n,s}]^\top$ and an end point $P_{n,e} = [N_{n,e}, E_{n,e}]^\top$ in $\{n\}$, where $n$ is the path number, numbered from left to right seen in the travel direction. The center path is the path from the initial position to the destination, and the current path is the one of which the vessel has the smallest cross-track error:

$$e_{n,cp} = l_\eta \sin(\theta_{path} - \theta_{l_\eta}), \tag{2.64}$$

where $\theta_{path}$ is the course of the path, $l_\eta$ the euclidean error from the current vessel position $[N, E]^\top$ to $P_{n,s}$ and $\theta_{l_\eta}$ the realtive bearing:

$$\theta_{l_\eta} = 2((E - E_{n,s}), (N - N_{n,s})), \tag{2.65}$$

$$l_\eta = \sqrt{(N - N_{n,s})^2 + (E - E_{n,s})^2}. \tag{2.66}$$

Figure 2.8 shows an example of multiple paths. From the predefined paths, a set of possible branching paths is generated. The branching paths allows the ownship to branch out from the current path with an angle $\alpha_b$, and merge back to the destination at the center-path at the same angle. Thus, the branching paths exist of parallel subpaths along the predefined paths, branching subpaths branching out from the current path and merging subpaths merging back to the center path.

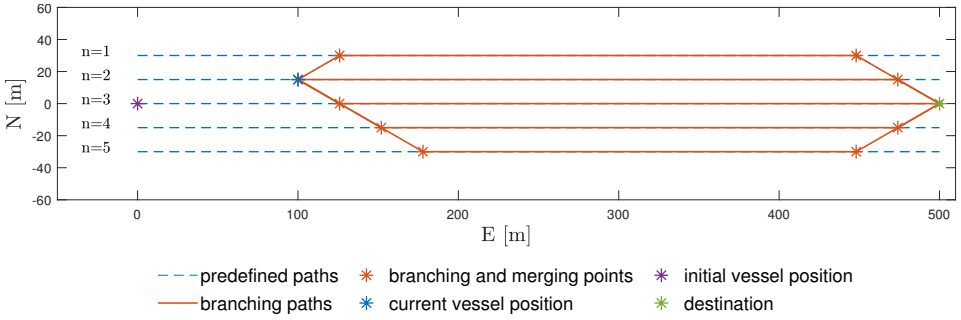



**Figure 2.8:** Example of a set of predefined paths and branching paths with $l_s = 15m$ and $\alpha_b = 30°$ in $\{n\}$. In this example, the current path number is $n_{current} = 2$ and the center path number $n_{center} = 3$.

Each branching subpath is defined by a start point $P_{bn,s} = [N_{bn,s}, E_{bn,s}]^\top$ at path number $n_{current}$ and an end point $P_{bn,e} = [N_{bn,e}, E_{bn,e}]^\top$ at path number $n$ in $\{n\}$:

$$N_{bn,s} = N + \frac{\epsilon_{n_{cp}}}{\sin(\alpha_b)} \cos\left(\theta_{path} - sgn\left(\epsilon_{n_{cp}}\right)\alpha_b\right), \tag{2.67}$$

$$E_{bn,s} = E + \frac{\epsilon_{n_{cp}}}{\sin(\alpha_b)} \sin\left(\theta_{path} - sgn\left(\epsilon_{n_{cp}}\right)\alpha_b\right), \tag{2.68}$$

$$N_{bn,e} = N_{bn,s} + \frac{l_s|k|}{\sin(\alpha_b)} \cos\left(\theta_{path} - (k)\,\alpha_b\right), \tag{2.69}$$

$$E_{bn,e} = E_{bn,s} + \frac{l_s|k|}{\sin(\alpha_b)} \sin\left(\theta_{path} - (k)\,\alpha_b\right), \tag{2.70}$$

where $k = n - n_{current}$, and $[N, E]^\top$ is the current vessel position.

The merging subpaths all ends at the endpoint of the center path:

$$P_{mn,e} = [N_{n_{center},e}, E_{n_{center},e}]^\top. \tag{2.71}$$

The start point $P_{tn,e}$ of the merging path from path number $n$ becomes

$$N_{mn,s} = N_{t,e} + \frac{l_s|j|}{\sin(\alpha_b)} \cos\left(\theta_{path} - sgn\left(j\right)\alpha_b + \pi\right), \tag{2.72}$$

$$E_{mn,s} = E_{t,e} + \frac{l_s|j|}{\sin(\alpha_b)} \sin\left(\theta_{path} - sgn\left(j\right)\alpha_b + \pi\right), \tag{2.73}$$

where $j = n - n_{center}$.

The parallel subpath along path number $n$ starts in the end point of the branching subpath ending at that path, $P_{pn,s} = P_{bn,e}$, and ends at the start point of the merging path from that path, $P_{pn,e} = P_{mn,s}$.

Furthermore, when solving the VPP along multiple paths, a third dimension is added to the $path \times time$ space, namely the layer dimension. Each path is assigned a $path \times time$ space and represents a layer.

### 2.5.2 Branching-course model predictive control (BC-MPC)

The BC-MPC algorithm is based on model predictive control (MPC). It plans multiple maneuvers respecting dynamic constriants over a finite prediction horizon, forming a tree of branching trajectories with a depth equal to the number of planned maneuvers. The trajectory which minimizes the objective function is selected, and the first maneuver is executed before the algorithm replans. The object function is designed considering COLREGs and robustness towards measurement noise. Hence, the two main tasks of the algorithm are 1) generating a set of trajectories and 2) choose the optimal trajectory. Figure 2.9 shows an overview of the algorithm and the two tasks. BC-MPC is originally developed and evaluated using an underactuated, high speed ASV [2]. The following of this section is based on the original article and explains the main steps of the trajectory generation and the optimization in terms of the original parametrization, $(U, \chi)$. Note that a constant sideslip is assumed, $\dot{\chi} = r + \dot{\beta} = r$.

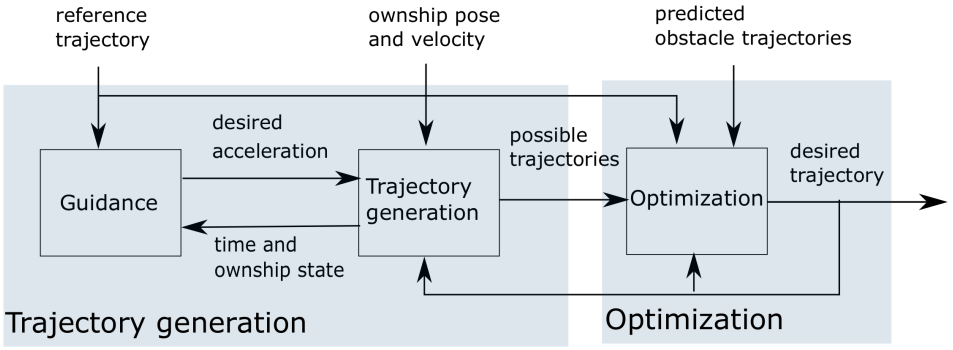



**Figure 2.9:** Overview of BC-MPC

#### Trajectory generation

An example of a tree of branching paths width a number of maneuvers, or levels, equal to three is illustrated in Figure 2.10. For each level, a number of $N_U$ speed accelerations and $N_\chi$ course accelerations is considered, and the example is generated with $\boldsymbol{N}_U = [1, 1, 1]$ and $\boldsymbol{N}_\chi = [3, 3, 3]$.

The trajectories are generated iteratively, and for each node in the tree, the subtrajectories is generated by the following steps:

- A number of $N_U$ speed acceleration and $N_\chi$ course accelerations is sampled from a set of feasible accelerations, $\mathcal{A}$, considering dynamic constraints.
- The desired accelerations $(\dot{U}'_d, \dot{r}'_d)$ are calculated by a guidance scheme so that an acceleration that converges to the desired trajectory can be included in the search space, if applicable.

- A set of continuous acceleration trajectories is generated from each sample, assuming an actuator change of $T_{ramp}$ seconds, and a total maneuvering time of $T_U$ and $T_\chi$, respectively. The acceleration trajectories are integrated into desired velocity trajectories $\mathcal{U}_d$, using the previously desired velocity as the initial condition to ensure continuity.
- The desired velocity trajectories are feedback-corrected considering the current state of the vessel, and a set of predicted velocity trajectories $\bar{\mathcal{U}}$ and a set of pose trajectories $\bar{\mathcal{H}}$ are generated.

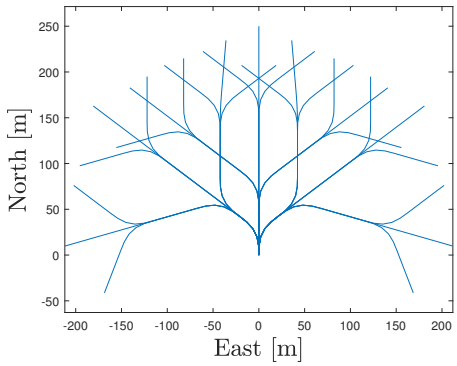



**Figure 2.10:** Branching trajectories generated with $\boldsymbol{N}_U = [1, 1, 1]$ and $\boldsymbol{N}_\chi = [3, 3, 3]$.

### Acceleration samples

The minimum and maximum possible accelerations $(\dot{\boldsymbol{U}}_{min}, \dot{\boldsymbol{U}}_{max}), (\dot{\boldsymbol{r}}_{min}, \dot{\boldsymbol{r}}_{max})$ are determined from the vessel model and the minimum and maximum available thrust $(\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max})$, where the thruster limits also should consider thrust rate limitations $(\dot{\boldsymbol{\tau}}_{min}, \dot{\boldsymbol{\tau}}_{max})$:

$$\boldsymbol{\tau}_{min} = \text{sat}(\boldsymbol{\tau}_0 + T_{ramp}\dot{\boldsymbol{\tau}}_{min}, \boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}) \tag{2.74}$$

$$\boldsymbol{\tau}_{max} = \text{sat}(\boldsymbol{\tau}_0 + T_{ramp}\dot{\boldsymbol{\tau}}_{max}, \boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}). \tag{2.75}$$

The saturation function $\text{sat}(\boldsymbol{a})$ is defined as

$$a_i = \begin{cases} a_{min,i} & , a_i < a_{min,i} \\ a_{max,i} & , a_i > a_{max,i} \\ a_i & , \text{otherwise.} \end{cases} \tag{2.76}$$

where $\boldsymbol{a} = \left[a_1, \dots a_n\right]^\top$, $\boldsymbol{a}_{min} = \left[a_{min,1}, \dots a_{min,n}\right]^\top$, $\boldsymbol{a}_{max} = \left[a_{max,1}, \dots a_{max,n}\right]^\top$ for $i = \{1, \dots, n\}$.

The set of possible accelerations is

$$\mathcal{A} = \left\{ \left(\dot{U}, \dot{r}\right) \in \mathbb{R}^2 \middle| \dot{U} \in \left[\dot{U}_{min}, \dot{U}_{max}\right], \dot{r} \in [\dot{r}_{min}, \dot{r}_{max}] \right\}, \tag{2.77}$$

and a set of uniform samples $\dot{\boldsymbol{U}}_{samples}, \dot{\boldsymbol{r}}_{samples}$ is sampled from $\mathcal{A}$

$$\dot{\boldsymbol{U}}_{samples} = \left\{\dot{U}_1, \ldots, \dot{U}_{N_U}\right\}, \tag{2.78}$$

$$\dot{\boldsymbol{r}}_{samples} = \left\{\dot{r}_1, \ldots, \dot{r}_{N_\chi}\right\}. \tag{2.79}$$

To ensure that at least one trajectory in the search space converge towards the goal, if applicable, the desired speed and/or course acceleration $\dot{U}'_d, \dot{r}'_d$ is included in $\dot{\boldsymbol{U}}_{samples}$ if $\dot{U}_d \in \mathcal{A}$ and/or $\dot{\boldsymbol{r}}_{samples}$ if $\dot{r}_d \in \mathcal{A}$, respectively. In that case, the sample nearest $(\dot{U}'_d, \dot{r}'_d)$ is removed to keep the same number of samples.

### Desired acceleration

The desired acceleration $(\dot{U}'_d, \dot{r}'_d)$ is calculated based on LOS guidance, and a path particle fixed to the path $P$ of which the velocity is controlled. Equations (2.26)-(2.27) gives the desired heading $\chi_{d,LOS}$, and the path particle velocity by the vessel speed $U$, vessel orientation $\chi$, the alongtrack-error $s$ and a tuning parameter $\gamma_s$:

$$U_{pp} = U\cos\left(\chi - \chi_{path}\right) + \gamma_s s, \tag{2.80}$$

letting the vessel converge towards $P$ with constant speed $U$. However, we would preferably control the vessel speed to converge to the desired trajectory. The path particle position and velocity is therefore fixed at the desired position on the trajectory, giving the desired vessel velocity

$$U_{d,LOS} = \begin{cases} \text{sat}\left(\frac{U_t - \gamma_s s}{\cos(\chi - \chi_p)}, 0, U_{max,LOS}\right) & \text{if } |\cos(\chi - \chi_p)| > \epsilon \\ \text{sat}\left(\frac{U_{pp} - \gamma_s s}{\epsilon}, 0, U_{max,LOS}\right) & \text{else,} \end{cases} \tag{2.81}$$

where $U_t$ is the trajectory velocity, and $U_{max,LOS}$ the maximum vessel speed. Division by zero is avoided using the small constant $\epsilon$. The desired acceleration becomes

$$\dot{U}'_d = \frac{U_{d_{LOS}} - U_{d0}}{T_U - T_{ramp}}, \tag{2.82}$$

$$\dot{r}'_d = \frac{\chi_{d_{LOS}} - \chi_{d0}}{T_{ramp}(T_\chi - 2T_{ramp})}. \tag{2.83}$$

### Desired trajectories

A set of piecewise-linear speed and course acceleration trajectories, representing the maneuvers, are calculated from the sampled accelerations $\dot{U}_i, i = \{1, \cdots, N_U\}$ and $\dot{\chi}_i, i = \{1, \cdots, N_\chi\}$ , ramp time $T_{ramp}$ and maneuver times $T_U$ and $T_\chi$:

$$\dot{U}_{d,i}(t) = \begin{cases} k_{U,i}t & , 0 \leq t < T_{ramp} \\ \dot{U}_i & , T_{ramp} \leq t < T_U - T_{ramp} \\ \dot{U}_i - k_{U,i}(t - (T_U - T_{ramp})) & , T_U - T_{ramp} \leq t < T_U \\ 0 & , T_U \leq t \leq T, \end{cases} \tag{2.84}$$

$$\dot{r}_{d,i}(t) = \begin{cases} k_{r,i}t & ,0 \leq t < T_{ramp} \\ 2\dot{r}_i - k_{r,i}t & ,T_{ramp} \leq t < 2T_{ramp} \\ 0 & ,2T_{ramp} \leq t < T_\chi - 2T_{ramp} \\ -k_{r,i}(t - (T_\chi - 2T_{ramp})) & ,T_\chi - 2T_{ramp} \leq t < T_\chi - T_{ramp} \\ -2\dot{r}_i + k_{r,i}(t - (T_\chi - 2T_{ramp})) & ,T_\chi - T_{ramp} \leq t < T_\chi \\ 0 & ,2T_\chi \leq t \leq T, \end{cases} \tag{2.85}$$

where $k_{U,i} = \frac{\dot{U}_{d,i}}{T_{ramp}}$ and $k_{\chi,i} = \frac{\dot{\chi}_{d,i}}{T_{ramp}}$ are slopes of the acceleration trajectories.

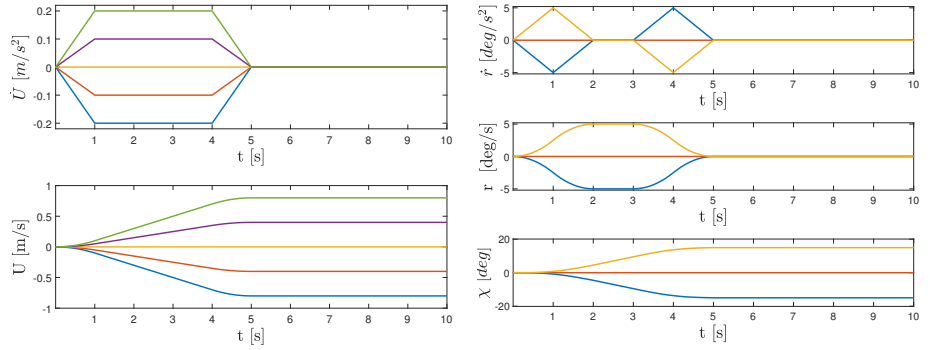
The trajectories of the desired speed, course rate and course are found by integrating the acceleration trajectories (2.84)- (2.85):

$$U_{d,i}(t) = U_{d,0} + \int_{t_0}^{t} \dot{U}_{d,i}(\gamma)d\gamma, i \in \left[1, N_U\right], \tag{2.86}$$

$$r_{d,i}(t) = r_{d,0} + \int_{t_0}^{t} \dot{r}_{d,i}(\gamma)d\gamma, i \in \left[1, N_\chi\right], \tag{2.87}$$

$$\chi_{d,i}(t) = \chi_{d,0} + \int_{t_0}^{t} \dot{\chi}_{d,i}(\gamma)d\gamma, i \in \left[1, N_\chi\right]. \tag{2.88}$$

The initial values $U_{d,0}, r_{d,0}, \chi_{d,0}$ of the first level correspond to the desired values of the previous BC-MPC iteration. For the other levels, the initial values correspond to the previous sub trajectory. This ensures continuity of the reference passed to the controller. Figure 2.11 shows an example of desired acceleration, velocity and course trajectories. Looking at the graphs for $\dot{\boldsymbol{r}}$, it can be seen that the integral of $\dot{r}_{d,i}(t)$ (2.85) is zero. Thus, a maneuver will start and end with the same course rate, and setting $r_{d,0} = 0$ ensures that each maneuver starts and ends with a constant course motion.




**(a)** Speed trajectories. **(b)** Course trajectories.

**Figure 2.11:** Examples of acceleration, speed and course trajectories for $\dot{\boldsymbol{u}}_{samples} = [-0.2, -0.1, 0, 0.1, 0.2]$, $\dot{\boldsymbol{r}}_{samples} = [-5, 0, 5]$, $T_{ramp} = 1s$, $T_U = 5s$, $T_\chi = 5s$, $T = 10s$.

Infeasible velocities are removed before the desired speed and course trajectories form a set of desired velocity trajectories:

$$\mathcal{U}_d = \{U_{d,1}(t), U_{d,2}(t), ..., U_{d,N_U}(t)\} \times \left\{\chi_{d,1}(t), \chi_{d,2}(t), ..., \chi_{d,N_\chi}(t)\right\}. \tag{2.89}$$

**Trajectory prediction**

Next, the velocity and course trajectories $\bar{U}(t)$ and $\bar{\chi}(t)$ are predicted, using the first order closed-loop error model and feedback of the current vessel speed $U_0$ and course $\chi_0$. The error model is

$$\dot{\tilde{U}} = \frac{1}{T_u}\tilde{u}, \tag{2.90}$$

$$\dot{\tilde{r}} = \frac{1}{T_r}\tilde{r}, \tag{2.91}$$

where $\tilde{U} = \bar{U} - U_d$, $\tilde{\chi} = \bar{\chi} - \chi_d$ and $T_{\tilde{U}}, T_{\tilde{\chi}}$ are time constants. This is a simple error mode, but considered sufficient assuming small tracking errors [2]. The predicted trajectories becomes

$$\bar{U}_i = \tilde{U}_0 e^{-\frac{1}{T_{\tilde{U}}}(t-t_0)} + U_{d,i}(t), i \in [1, N_U], \tag{2.92}$$

$$\bar{\chi}_i = \tilde{\chi}_0 e^{-\frac{1}{T_{\tilde{\chi}}}(t-t_0)} + \chi_{d,i}(t), i \in [1, N_\chi], \tag{2.93}$$

where $\tilde{U}_0 = U_0 - U_{d,0}$, $\tilde{\chi}_0 = \chi_0 - \chi_{d,0}$. The set of predicted celocity trajectories is made:

$$\bar{\mathcal{U}} = \{\bar{U}_1(t), \bar{U}_2(t), \ldots, \bar{U}_{N_U}(t)\} \times \{\bar{\chi}_1(t), \bar{\chi}_2(t), \ldots, \bar{\chi}_{N_\chi}(t)\}. \tag{2.94}$$

Last, the position trajectories are cacluated from the predicted velocity trajectories:

$$\dot{\bar{\boldsymbol{p}}} = \begin{bmatrix}\cos(\bar{\chi})\\ \sin(\bar{\chi})\end{bmatrix}\bar{U}, \tag{2.95}$$

where $\bar{\boldsymbol{p}} = \left[\bar{N}(t), \bar{E}(t)\right]^\top$ is obtained by integrating (2.95) with the current vessel position as the initial condition. The predicted vessel pose trajectories becomes:

$$\bar{\boldsymbol{\eta}} = \left[\bar{N}(t), \bar{E}(t), \bar{\chi}(t)\right]^\top, \tag{2.96}$$

and the set of predicted pose trajectories:

$$\bar{\mathcal{H}} = \left\{\bar{\boldsymbol{\eta}}\left(t; \bar{U}(t), \bar{\chi}(t)\right) \mid \left(\bar{U}(t), \bar{\chi}(t)\right) \in \bar{\mathcal{U}}\right\}. \tag{2.97}$$

**Optimization**

A cost for each trajectory $\bar{\boldsymbol{\eta}}(t) \in \bar{\mathcal{H}}$, $\boldsymbol{u}_d(t) \in \mathcal{U}_d$ is calculated according to the objective function

$$\begin{aligned} J(\bar{\boldsymbol{\eta}}(t); \boldsymbol{u}_d(t); \boldsymbol{p}_d(t)) = {} & w_{al}\text{align}(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_d(t)) \\ & + w_{av}\text{avoid}(\bar{\boldsymbol{\eta}}(t)) \\ & + w_{tc}\text{trans}(\boldsymbol{u}_d(t)), \end{aligned} \tag{2.98}$$

where $\boldsymbol{p}_d(t)$ is the reference trajectory. The different objective terms is calculated by the functions align$(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_d(t))$, avoid$(\bar{\boldsymbol{\eta}}(t))$, and trans$(\boldsymbol{u}_d(t))$, weighted by the respective parameters $w_{al}, w_{av}, w_{tc} \geq 0$. The trajecotry minimizing (2.98) is sent to the controller as the reference:

$$\boldsymbol{u}_d^*(t) = \underset{(\bar{\boldsymbol{\eta}}_k(t), \boldsymbol{u}_{d,k}(t)) \in (\bar{\mathcal{H}}, \mathcal{U}_d)}{\operatorname{argmin}} J\left(\bar{\boldsymbol{\eta}}(t); \boldsymbol{u}_d(t); \boldsymbol{p}_d(t)\right). \tag{2.99}$$

The algorithm is extended with static obstacle consideration in [3].

### Trajectory alignment

Trajectory alignment considers Euclidian and angular error between the predicted pose trajectory and the desired path. The desired course is

$$\chi_d(t) = \arctan(\dot{E}_d(t), \dot{N}_d(t)), \tag{2.100}$$

and the alignment cost is

$$\text{align}(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_d(t)) = \int_{t_0}^{t_0+T_{full}} \left( w_p \left|\left| \begin{bmatrix} \bar{N}(\gamma) \\ \bar{E}(\gamma) \end{bmatrix} - \boldsymbol{p}_d(\gamma) \right|\right|_2 + w_\chi |\Upsilon(\bar{\chi}(\gamma) - \chi_d(\gamma))| \right) \tag{2.101}$$

where $w_p, w_\chi > 0$ controls the relative weighting of the Euclidean and course error, $T_{full}$f is the total prediction horizon for the predicted trajectories, and $\Upsilon(\cdot)$ wraps to $[-\pi, \pi)$.

### Moving obstacle avoidance

The obstacle avoidance term is given by

$$\text{avoid}(\bar{\boldsymbol{\eta}}(t)) = \sum_{i=1}^{N_o} \int_{t=t_0}^{t_0+T_{full}} w_{mo,i}(\gamma) \text{penalty}_i(\bar{\boldsymbol{\eta}}(\gamma)) d\gamma, \tag{2.102}$$

where penalty$_i(\bar{\boldsymbol{\eta}}(\gamma))$ and $w_{mo,i}(\gamma)$ is the cost and weight for obstacle $o_i$, and $N_o$ the number of obstacles.

The penalty for each obstacle is calculated based on the assumption of constant obstacle speed and course. The distance and realtive bearing to obstacle $i$ at time $t$ becomes

$$d_i(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_{o,i}(t)) = \left|\left| \boldsymbol{r}_i(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_{o,i}(t)) \right|\right|_2 \tag{2.103}$$

$$\beta_i(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_{o,i}(t)) = \Upsilon(\arctan\left(r_{E,i}\left(\bar{\boldsymbol{\eta}}\left(t\right)\right), r_{N,i}\left(\bar{\boldsymbol{\eta}}\left(t\right)\right) - \chi_{o,i}\left(t\right)\right) \tag{2.104}$$

where $\chi_{o,i} = \arctan\left(\dot{E}_{o,i}\left(t\right), \dot{N}_{o,i}\left(t\right)\right)$ and $\boldsymbol{p}_{o,i}(t) = [N_{o,i}(t), E_{o,i}(t)]^\top$ are the course and position of obstacle $i$, and $\boldsymbol{r}_i(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_{o,i}(t))$ the vector between the obstacle $i$ and the predicted pose trajectory $\bar{\boldsymbol{\eta}}(t)$:

$$\boldsymbol{r}_i(\bar{\boldsymbol{\eta}}(t); \boldsymbol{p}_{o,i}(t)) = \boldsymbol{p}_{o,i}(t) - \begin{bmatrix} \bar{N}(t) \\ \bar{E}(t) \end{bmatrix}. \tag{2.105}$$

The penalty function calculates a region of collision (ROC), a safety region (SR), and margin region (MR) for each obstacle. The regions consist of three elliptical and one circular region and have a greater radius on the starboard and bow side of the obstacle, as illustrated in Figure 2.12a. The radius $D_k$ of the region $k$ varies with the relative bearing, and is given by

$$D_k(\beta_i) = \begin{cases} b_k & \text{if} \quad \beta_i < -\frac{\pi}{2} \\ \frac{a_k b_k}{\sqrt{(b_k \cos\beta_i)^2 + (a_k \sin\beta_i)^2}} & \text{if} \quad -\frac{\pi}{2} \leq \beta_i < 0 \\ \frac{a_k c_k}{\sqrt{(c_k \cos\beta_i)^2 + (a_k \sin\beta_i)^2}} & \text{if} \quad 0 \leq \beta_i < \frac{\pi}{2} \\ \frac{b_k c_k}{\sqrt{(c_k \cos\beta_i)^2 + (b_k \sin\beta_i)^2}} & \text{if} \quad \frac{\pi}{2} \leq \beta_i \end{cases} \tag{2.106}$$

where $a_k, b_k$ and $c_k = b_k + d_{\text{COLREGs}}$ are the minor, major and COLREGs axes of the collision, margin and safety region. The elliptical regions motivate for passing the obstacle on her port or aft side, or keeping a greater distance if passing abaft or on her starboard side, in compliance with COLREGs. An option to the elliptical regions is circular regions that does not consider COLREGs [2].

For the elliptical cost, an additional inner penalty is added to avoid constant cost inside the collision region, given by

$$\text{inner\_penalty}_i(\bar{\boldsymbol{\eta}}(t)) = \begin{cases} 1 & \text{if} \quad d_i < D_0^* \\ 1 - \frac{y_b(d_i, \beta_i)}{d_{\text{COLREGs}}} & \text{if} \quad D_0^* \leq d_i < D_0 \\ 0 & \text{else.} \end{cases} \tag{2.107}$$

The distance in the $y$-direction of the obstacle's body frame from the $D_0^*$ region to the point $(d_i, \beta_i)$ is givven by $y_b(d_i, \beta_i)$. The distance $D_0^*$ is calculated as

$$D_0^*(\beta_1) = \begin{cases} \frac{a_0 b_0}{\sqrt{(b_0 \cos\beta_i)^2 + (a_0 \sin\beta_i)^2}} & \text{if} \, |\beta_i| < \frac{\pi}{2} \\ b_0 & \text{else.} \end{cases} \tag{2.108}$$

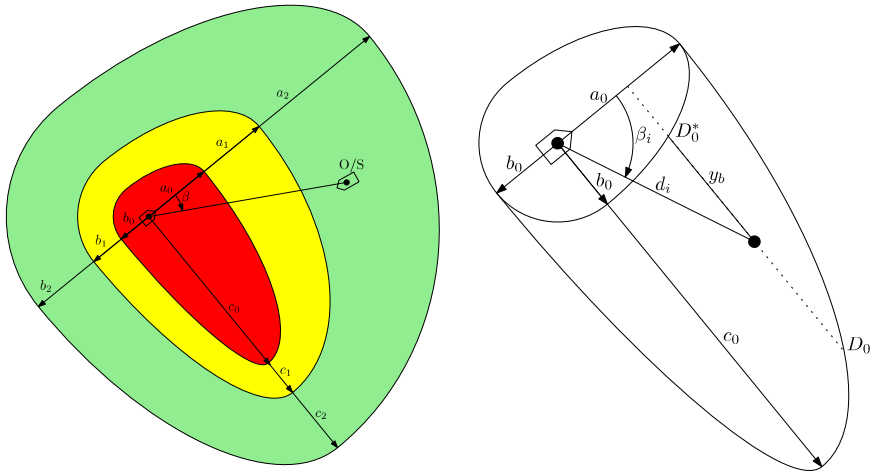
Figure 2.12b illustrates the variables of the inner elliptical penalty function.

Finally, the total penalty is

$$\begin{aligned} \text{penalty}_{i,\text{COLREGs}}(\bar{\boldsymbol{\eta}}(t)) &= \text{inner\_penalty}_i(\bar{\boldsymbol{\eta}}(t)) \\ &+ \begin{cases} 1 & \text{if} \;\; d_i < D_0 \\ 1 + \frac{\gamma_1 - 1}{D_1 - D_0} & \text{if} \;\; D_0 \leq d_i < D_1 \\ \gamma_1 - \frac{\gamma_1 (d_i - D_1)}{D_2 - D_1} & \text{if} \;\; D_1 \leq d_i < D_2 \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{2.109}$$

**Transitional cost**

Transitional cost is included to increase robustness, and considers changes in the desired velocity trajectory $[U_d, \chi_d]^\top$ compared to the previous desired velocity tra-




**(a)** Elliptical COLREGs function with minor and major axis. The region of collision (ROC) is shown in red, the safety region (SR) in yellow and the margin region (MR) in green.

**(b)** Variables for the inner elliptical col-penalty function.

**Figure 2.12:** Elliptical COLREGs regions and variables [2].

jectory $\left[U_d^-, \chi_d^-\right]^\top$ for the first maneuver. The minimum speed and heading difference for all candidates $\boldsymbol{u}_d(t) \in \mathcal{U}_d$ are

$$e_{U,\min} = \min_{\boldsymbol{u}_d(t)\in\mathcal{U}_d} \int_{t_0}^{t_0+T_1} \left|U_d(\gamma) - U_d^-(\gamma)\right| \mathrm{d}\gamma \tag{2.110}$$

$$e_{\chi,\min} = \min_{\boldsymbol{u}_d(t)\in\mathcal{U}_d} \int_{t_0}^{t_0+T_1} \left|\chi_d(\gamma) - \chi_d^-(\gamma)\right| \mathrm{d}\gamma. \tag{2.111}$$

where $T_1$ is the step time of the first maneuver. Considering the translational cost for speed and course individually [3], a transitional error is added if the error is greater than the minimum value:

$$\begin{aligned} \mathrm{trans}\left(\boldsymbol{u}_d(t)\right) = &\begin{cases} w_{tc,U} & \text{if} \quad \int_{t_0}^{t_0+T_1} \left|U_d(\gamma) - U_d^-(\gamma)\right| \mathrm{d}\gamma > e_{U,\min} \\ 0 & \text{else} \end{cases} \\ + &\begin{cases} w_{tc,\chi} & \text{if} \quad \int_{t_0}^{t_0+T_1} \left|\chi_d(\gamma) - \chi_d^-(\gamma)\right| \mathrm{d}\gamma > e_{\chi,\min} \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{2.112}$$

where $w_{tc,U}$ and $w_{tc,\chi}$ weights the translational error for speed and course, respectively.

# Chapter 3

# Hybrid collision avoidance system

This chapter explains the desirable properties, architectures and algorithms considered in the proposed hybrid COLAV system. The identified desirable properties of a COLAV system for passenger ferries are presented in Section 3.1. Section 3.2 presents a hybrid architecture combining the MP-VP and the BC-MPC algorithm, and two stand-alone systems using each of the algorithms. Section 3.3 describes the modifications made on the MP-VP to fit with the hybrid architecture, and Section 3.4 outlines the modifications made on the BC-MPC algorithm to work with 3 DOF, slow speed vessels and the hybrid architecture.

## 3.1 Desirable properties of passenger ferries

A definition of a ferry is *"a boat or ship for taking passengers and often vehicles across an area of water, especially as a regular service"* [33]. Considering the function of a ferry as a transport system at regular service, the limited operational space, and safety, the following properties are considered desirable for a COLAV system for autonomous passenger ferries

1. The ferry should reach the destination in ample time.
2. The computational cost must be small enough to run real-time on an on-board computer.
3. The behavior must be safe, and collisions must be avoided.
4. The transit should have comfortable speed, acceleration, and jerk for standing passengers.
5. The COLAV system should be compliant with the COLREGs maneuvering

rules 8 and 13-17.

6. When operating in limited areas, speed changes should be favored over course or path changes. Especially close to the pier, a stop-and-go approach is preferable.
7. The behavior should be energy-efficient.

Properties 1-2 is about achieving the desired function of the ferry. If the destination is not reached, the purpose of passenger transport vessels is not achieved. Reasonable travel time is essential for the autonomous passenger ferry to fulfill its purpose of passenger transport, as passengers most likely will not use a ferry that is not time-reliable. Furthermore, the COLAV system must run in real-time to be used at the physical ferry.

Property 3 considers safety, which is critical for passenger ferries. Property 4 considers passenger comfort, which is essential for getting passengers to travel with the ferry. Furthermore, some aspects of passenger comfort are also vital for safety, as passengers losing their balance due to rapid accelerations can get hurt. Considering 3 DOF, slow speed ferries, operating short transits in urban waterways with calm water, it is relevant to consider passenger comfort related to horizontal speed accelerations. In [4], a survey of studies on comfortable longitudinal accelerations for ground transportation vehicles, suggesting a maximum longitudinal acceleration of about $1.1 - 1.5\ m/s^2$ and a maximum jerk of $3\ m/s^3$ [34], is used as a guideline for autonomous passenger ferries.

Properties 4-6 consider the maneuvers of the ferry, which should comply with the COLREGs rules 8 and 13-17. Compliance with these rules is essential to achieve safe behavior. Furthermore, a stop-and-go approach is desireable in confined areas close to the pier. It is more energy-efficient and is experienced as the most comfortable approach by test passengers through sea trials [4]. Note that this conflicts with the COLREGs Rule 8, which suggests course maneuvers when possible.

Property 7 considers energy-efficiency behavior, which is desirable considering the electrical passenger ferries as a sustainable and green transport solution.

## 3.2 COLAV architectures

The MP-VP sets requirements to the previous COLAV layer, assuming that the paths are collision-free of static objects. Furthermore, the method is not allowed to plan trajectories leaving those paths, and could encounter a situation where the ferry gets trapped. This problem can be solved by a short-term COLAV layer modifying the trajectory generated from the MP-VP. The hybrid, three-layer architecture combining path planning, the MP-VP and the BC-MPC algorithm is therefore proposed, listed as Architecture 1 in Table 3.1. Architectures 2 and 3 are two-layer architectures, where one algorithm handles all COLAV of moving obstacles, and are included such that the three-layer Architecture 1 can be compared

with stand-alone COLAV systems of the respective algorithms. Section 3.2.1-3.2.2 describes the architecture options further.

**Table 3.1:** Collision avoidance architecture options.

| | **High level** | **Mid level** | **Low level** |
|---|---|---|---|
| Architecture 1 | Path planner | MP-VP | BC-MPC |
| Architecture 2 | Path planner | - | MP-VP |
| Architecture 3 | Trajectory planner | - | BC-MPC |

### 3.2.1 Architecture 1: hybrid architecture combining the MP-VP and the BC-MPC algorithm

In this architecture, illustrated in Figure 3.1, three layers are included in the hybrid COLAV system. The high-level layer, or the path planner, generates time-invariant paths that are collision-free concerning the static environmental data. The paths are generated as parallel paths to the center-path, which connects the initial position and the destination, with a separation length of $l_{sep}$, as described in Section 2.5.1. The mid-level layer, or the MP-VP, consider the detected moving obstacles. It plans a trajectory for the next $T_{plan}$ seconds along the paths received from the path planner, that is collision-free with respect to a predicted future of moving obstacles with constant speed and course. Last, the low-level layer, or the BC-MPC algorithm, serves as a short-term COLAV system, returning a feasible desired velocity trajectory to the controller that is collision-free to the immediate situation. The BC-MPC algorithm provides trajectories with continuous acceleration, and no additional reference filter is needed.

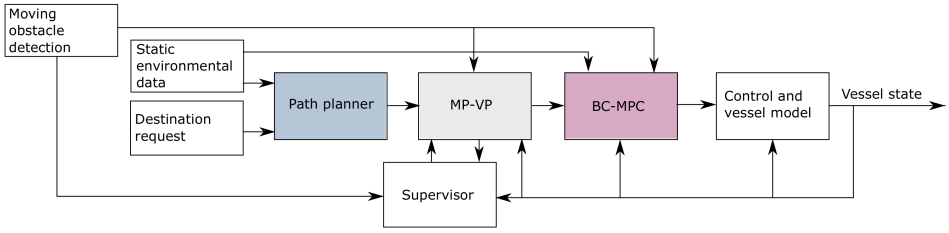



**Figure 3.1:** COLAV Architecture 1: Hybrid architecture with the MP-VP and the BC-MPC algorithm.

The path planner runs initially when a new destination is requested. The MP-VP is invoked by the supervisor, which of the details is explained in the following of this section. Last, the BC-MPC algorithm runs frequently to handle immediate situations.

### Supervisor

The supervisor is based on the application of the MP-VP in [4], and monitors the information received from the moving obstacle tracker, the current reference trajectory calculated by the MP-VP, and the vessel state. It calculates which moving obstacles that should be considered by the COLAV algorithms and if the reference trajectory can or should be replanned. The mid-layer is locked in the hybrid architecture when the ferry is considered to be in a collision situation, to avoid a new reference path that will change the objective function of the BC-MPC algorithm, and possibly lead to a sudden change of behavior. When the lock is open, and a new trajectory is requested, the supervisor invokes the MP-VP, and a new reference trajectory is generated and passed to the low-level COLAV algorithm.

### Moving obstacle monitoring

Moving obstacles are considered detected when they are within a region of observation (ROO). For all detected moving obstacles, it is evaluated if they should be added to the set of obstacles that are considered by the MP-VP, $\mathcal{O}_c$. This evaluation is done based on CPA. The $d_{CPA}$ is calculated by (2.34), and the minimum distance to the CPA within the next $T_{plan}$ seconds becomes:

$$d_{CPA,T_{plan}} = \min_{t \in [t_0, T_{plan}]} (d_{CPA}(t)), \tag{3.1}$$

where $t_0$ is the current time and $T_{plan}$ is the prediction horizon of the MP-VP. A moving obstacle is to be considered at time $t$ if

$$\|\boldsymbol{p}(t) - \boldsymbol{p}_o(t)\| < ROO \wedge d_{CPA,T_{plan}} \leq d_{CPA_{min}}(l_{sep}), \tag{3.2}$$

where $d_{CPA_{min}}(l_{sep})$ is the minimum distance at the CPA a moving obstacle can have without beeing considered. This distance is dependent on the separation length between the paths, and is given as

$$d_{CPA_{min}}(l_{sep}) = C_{CPA} l_{sep}, \tag{3.3}$$

where $C_{CPA}$ is a proportional constant. The approach of using CPA for moving obstacle monitoring is adapted from [4]. However, $d_{CPA_{min}}$ is made dependent on $l_{sep}$ to work with transits encountering paths with variable separation length.

### Opening and locking the mid-layer

The mid-layer is locked if the ownship is in a collision situation, that is, if the distance to the closest obstacle (DCO) is too small, and opened again when the DCO is sufficient, as illustrated in Figure 3.2.

The risk of using this lock is if the MP-VP plans a zero-speed reference to be followed by the BC-MPC algorithm and never gets to replan because the mid-layer

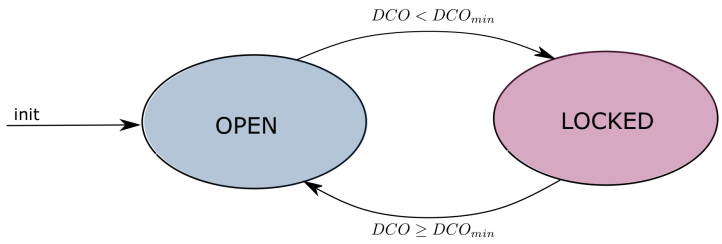



**Figure 3.2:** Overview of when the mid-layer is open or locked.

remains locked. Then, neither the MP-VP or the BC-MPC algorithm will be able to save the ferry from getting stuck. This issue should be considered when setting the value for $DCO_{min}$. A simple approach to solve this problem can be to let the lock remain open if the ferry speed is lower than a lower bound, as the ferry is then most likely not performing an emergency maneuver.

#### Requesting a new trajectory

Figure 3.3 illustrates the logic of when a new trajectory is requested. A new trajectory is requested if any of the following criteria are fulfilled

1. $\mathcal{O}_c^- \neq \mathcal{O}_c$
2. $VALIDATE\_CURRENT\_WAYPOINTS() == false$
3. $EE \geq EE_{max}$
4. $T_{plan} - (t_0 - t_{traj,0}) < T_{full}$.

Where $\mathcal{O}_c^-$ is the previously considered set of obstacles, $EE$ the Euclidean tracking error, $t_{traj,0}$ is the initial time of the current reference trajectory, and $t_0$ the current time. $VALIDATE\_CURRENT\_WAYPOINTS()$ returns $false$ if any $ROC_o,\ o \in \mathcal{O}_c$ intersects with any subpath of the current desired trajectory, and $true$ otherwise [5]. The first and second criteria are similar as for [4], and the third and fourth criteria are added for the hybrid algorithm. The third criteria request a new trajectory when the Euclidean error is greater than a minimum value, and the fourth criterion ensures that the BC-MPC algorithm receives a reference trajectory that has at least the same prediction horizon as the of BC-MPC algorithm.

### 3.2.2 Architecture 2: MP-VP COLAV system

Architecture 2 is similiar to the application in [4], and is illustrated in Figure 3.4. It exists of the same high-level path planner as for Architecture 1 (Section 3.2.1), and the MP-VP as the low-level COLAV layer. The MP-VP is the last instance

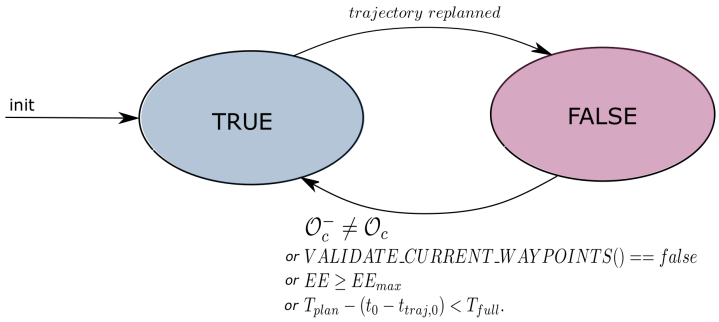



**Figure 3.3:** Overview of the status of requested trajectories

of COLAV, and it is critical that the supervisor invokes the MP-VP when needed, to avoid collisions. Moreover, a reference filter is added to provide continuous acceleration and velocity trajectories to the vessel controllers.

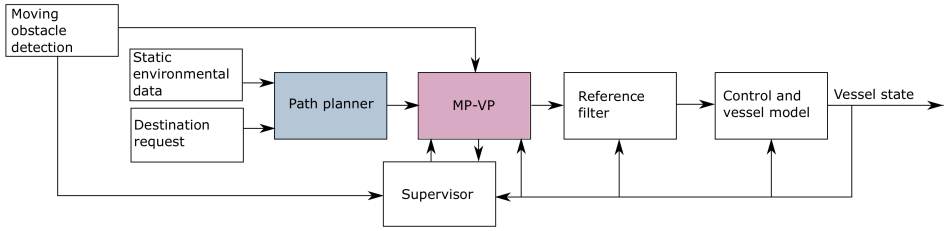



**Figure 3.4:** COLAV Architecture 2: path planner and the MP-VP.

### Supervisor

The supervisor used in Architecture 2 is similar to the supervisor used in Architecture 1. However, the lock is not active and remains open, as an updated trajectory from the MP-VP is critical in collision situations. Furthermore, the third and fourth criteria are not relevant when checking if a new trajectory should be requested.

### Reference filter

The MP-VP provides a reference trajectory with infinite accelerations, and a third-order reference filter is needed to provide a continuous acceleration reference to the controllers. The reference filter is a first-order low-pass filter cascaded with a mass-damper-spring-system

$$\dddot{\boldsymbol{\eta}}_d + (2\Delta + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\eta}}_d + (2\Delta + \boldsymbol{I})\boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d + \boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d = \boldsymbol{\Omega}^3\boldsymbol{r}^n, \tag{3.4}$$

where $\boldsymbol{\eta}_d$, $\dot{\boldsymbol{\eta}}_d$, $\ddot{\boldsymbol{\eta}}$, and $\dddot{\boldsymbol{\eta}}_d$ are the desired pose, velocity, acceleration and jerk, and $\boldsymbol{r}^n$ the pose reference in $\{n\}$ [27]. The diagonal matrices of relative damping ratios and

natural frequencies are denotes $\boldsymbol{\Delta} > 0$ and $\boldsymbol{\Omega} > 0$, respectively. The trajectories generated by the MP-VP have piecewise constant velocities and a dynamic position reference. The reference filter is therefore augmented with a velocity reference

$$\dddot{\boldsymbol{\eta}}_d + (2\Delta + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\eta}}_d + (2\Delta + \boldsymbol{I})\boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d + \boldsymbol{\Omega}^2\dot{\boldsymbol{\eta}}_d = \boldsymbol{\Omega}^3\boldsymbol{r}^n + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^2\dot{\boldsymbol{r}}^n. \quad (3.5)$$

The desired jerk is calculated from (3.5), using the current pose of the ferry, $\boldsymbol{\eta}$:

$$\dddot{\boldsymbol{\eta}}_d = \boldsymbol{\Omega}^3(\boldsymbol{r}^n - \boldsymbol{\eta}) + (2\Delta + \boldsymbol{I})\boldsymbol{\Omega}^2(\dot{\boldsymbol{r}}^n - \dot{\boldsymbol{\eta}}) - (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\eta}}. \quad (3.6)$$

To consider passenger comfort and vessel acceleration limitations, the desired jerk is saturated according to (2.76), and the reference filter is discretized and calculated by

$$\ddot{\boldsymbol{\eta}}_d = \text{sat}(\ddot{\boldsymbol{\eta}} + \dddot{\boldsymbol{\eta}}_d\Delta_t), \quad (3.7)$$
$$\dot{\boldsymbol{\eta}}_d = \text{sat}(\dot{\boldsymbol{\eta}} + \ddot{\boldsymbol{\eta}}_d\Delta_t), \quad (3.8)$$
$$\boldsymbol{\eta}_d = \boldsymbol{\eta} + \dot{\boldsymbol{\eta}}_d\Delta_t. \quad (3.9)$$
$$(3.10)$$

For further remarks on using the reference filter on the MP-VP generated trajectory and tuning, please refer to [4].

### 3.2.3 Architecture 3: BC-MPC COLAV system

Figure 3.5 illustrates Architecture 3, which has two levels, a high-level trajectory planner and the BC-MPC algorithm as a low-level COLAV system. The trajectory planner gives a piecewise linear, constant-speed trajectory connecting the destination waypoints when a new destination is requested, and only then. All tracking and collision avoidance of moving obstacles are handled by the BC-MPC algorithm. The reference trajectory generated by the trajectory planner is not replanned, but relative tracking, as described in Section 2.2.1, can be activated if the tracking error is significant.

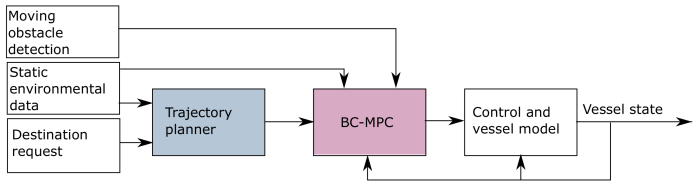



**Figure 3.5:** COLAV Architecture 3: trajectory planner and the BC-MPC algorithm.

## 3.3 Adapting the MP-VP

The trajectory generated by the MP-VP is originally fed to a reference filter and used as a reference for the controller, as described in Architecture 2 (Section 3.2.2). For Architecture 1 (Section 3.2.1), the trajectory is used as a reference for the short-term BC-MPC algorithm. For the latter case, the trajectory is not required to be frequently updated. Furthermore, as the operation space and transit length is extended to include open-sea, and not only confined environments, the consideration of COLREGs is added by increasing the fore and starboard region axis of the moving obstacle representations, similarly to the moving obstacle regions of the BC-MPC algorithm. Lastly, the desired heading is aligned with the path. Other than that, only minor adjustments are applied to the velocity-planning algorithm, as it is previously designed and evaluated using the milliAmpere test platform [4] [5].

### 3.3.1 Moving obstacle representation

The moving obstacles are represented as described in Section 2.5.1, but with modified axis lengths. It is assumed that obstacles keep a lower speed in confined areas and that the paths generated by the path planner correspond with the distance considered necessary to obstacles. The obstacle region sizes should correspond to the environment and separation length between the predefined paths. They need to be large enough to cover the moving obstacles and small enough not to occupy all of the predefined paths. The added lengths for every region $R = \{ROC, HPR, LPR\}$, and every direction $d = \{f, s, a, p\}$ is therefore made proportional to the separation length, $l_{sep}$, of the predefined paths in the area where the moving obstacle is located. The added lengths become:

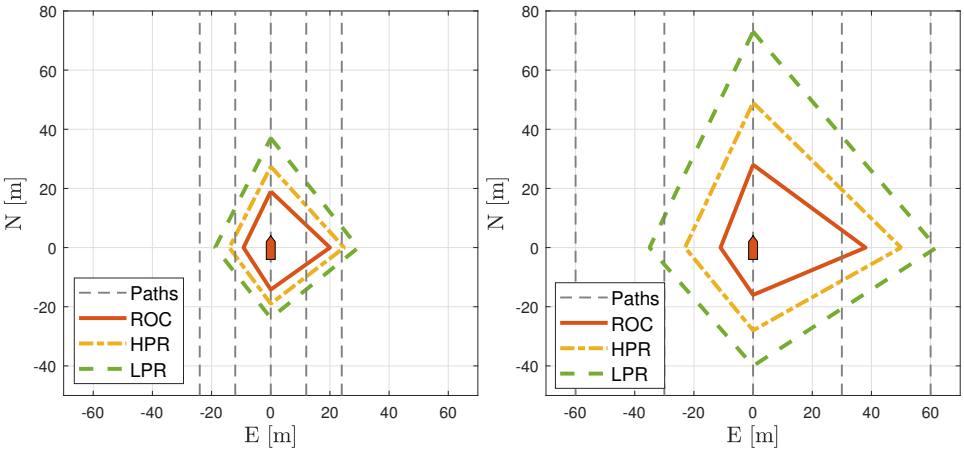
$$l_{R,f} = C_{R,f} l_{sep} \tag{3.11}$$
$$l_{R,s} = C_{R,s} l_{sep} \tag{3.12}$$
$$l_{R,a} = C_{R,a} l_{sep} \tag{3.13}$$
$$l_{R,p} = C_{R,p} l_{sep}, \tag{3.14}$$

where $C_{R,d} \geq 0$ is the proportional constant. The added length $l_{R,s} \geq 0$ is defined similarly as for the BC-MPC algorithm, with an extended length for starboard, motivating the ownship to pass on the port side as suggested in the COLREGs rules 14-15. The proportional constant $C_{R,s}$ is given as:

$$C_{R,s} = \begin{cases} C_{R,p} & \text{if } l_{sep} < l_{sep,COLREGs} \\ C_{R,p} + C_{COLREGs} & \text{if } l_{sep} \geq l_{sep,COLREGs} \end{cases} \tag{3.15}$$

where $C_{COLREGs} \geq 0$ is the proportional COLREGs constant. By choosing $l_{sep,COLREGs} > 0$, the COLREGs added length is only added for moving obstacles operating in spaces where paths have a separation length greater than

$l_{sep,COLREGs}$. Hence, the moving obstacle regions are not made too wide in narrow passages. Figure 3.6 shows an example of moving obstacles regions with separation length of $l_{sep} = 12m$ and $l_{sep} = 30m$, respectively.




**(a)** $l_{sep} = 12m$ **(b)** $l_{sep} = 30m$

**Figure 3.6:** Dynamic representaion of moving obstacles considering COLREGs for the MP-VP.

To simplify the transformation onto the $path \times time$ space, the obstacle representation is given by its current position, and is constant for the predicted future, even if the separation lengths between the paths and the nature of the environment changes. This is a simple representation of moving obstacles, and other factors as velocity could be included. Additionally, more advanced geographical factors based on expected behavior in certain areas can be used instead of the separation length. Further development of the representation of moving obstacles is, however, left to future work.

### 3.3.2 Trajectory generation

In the original application [4], the desired heading is aligned with the desired course:

$$\psi_d = \arctan\left(\dot{E}, \dot{N}\right). \tag{3.16}$$

For zero-speed references, (3.16) gives a zero heading reference. Aligning the heading to the course of the trajectory approaching the point of zero speed reference results in better convergence to the point. Hence, the desired heading is calculated as

$$\psi_d = \arctan\left(\left(E_{k+1} - E_k\right), \left(N_{k+1} - N_k\right)\right), \tag{3.17}$$

where $[N_k, E_k]^\top, [N_{k+1}, E_{k+1}]^\top$ are the waypoints making the current path of the piecewise linear reference path.

## 3.4 Adapting the BC-MPC algorithm

The milliAmpere ferry is fully actuated with dynamic positioning (DP) controllers, and controls $(u, v, \psi)$ independently. A parametrization in 3 DOF is necessary to benefit from all degrees of freedom, but increase the search space and the computational cost. The main topics addressed when adapting the BC-MPC algorithm to the ferry are parametrization, sampling and modeling, and the dilemma of simplifications versus complexity. Furthermore, the objective function must handle the reference trajectory of non-constant speed and course, and the possibilities of negative surge speed and sway speeds. Different solutions to these topics were explored before the following modifications were made:

- Parametrize in 3 DOF
- Calculate the set of feasible accelerations $\mathcal{A}$ using the milliAmpere model
- Generate an emergency search space that includes emergency sway maneuvers
- Modify the trajectory alignment term to track a pose trajectory $\boldsymbol{\eta}_d$
- Adjust the prediction horizon to better track trajectories with a non-constant speed and/or course reference
- Add a velocity term to the objective function.

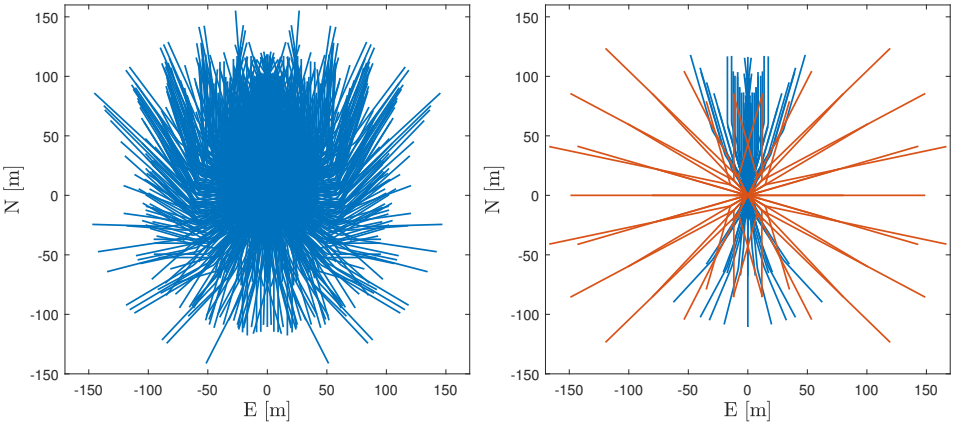
The parametrization in 3 DOF use $(u, v, \psi)$. The main search space is, however, sampled with $v = 0$, utilizing only $u$ and $\psi$ motions. Mostly, the equations for $U$ in Section 3.4 is used for $(u, v)$ and the equations considering $\chi$ for $\psi$ for most of the equations. The remaining of this section describes the further applied modifications.

### 3.4.1 Emergency search space

An emergency search space is sampled in addition to the main search space, with sway maneuvers included in all trajectories. Furthermore, only minimum, maximum, and zero acceleration of $(u, \psi)$ are considered, and the samples of the emergency search space are not aligned with the desired acceleration. Hence, the emergency space introduces sway maneuvers, but only for a few emergency maneuvers. The main search space without sway accelerations and motions is then considered sufficient, and the total search-space is significantly reduced. The numbers of samples in the emergency space of $(u, v, \psi)$ are denoted by $\boldsymbol{N}_{u,EM}, \boldsymbol{N}_{v,EM}, \boldsymbol{N}_{\psi,EM}$.

Figure 3.7a illustrates an example where sway samples are included in the main search space, resulting in a total of 1215 trajectories, and Figure 3.7b an example where sway maneuvers are included only in an emergency space, giving a total of 189 trajectories.

In this thesis, the emergency space is used only for sway emergency maneuvers. If the vessel allows accelerations above the range considered comfortable for pas-




**(a)** Trajectory search space sampled in 3DOF.

**(b)** Trajectory search space with an emergency search space.

**Figure 3.7:** Figure (a) shows a trajectory search space where sway maneuvers are included in the main search space, giving 1215 trajectories. The search space is sampled with $\boldsymbol{N}_u = [5, 3, 1]\,, \boldsymbol{N}_v = [3, 3, 1]\,, \boldsymbol{N}_\psi = [3, 3, 1]$. Figure (b) shows a search space with a 2DOF search space and an emergency space, sampled by $\boldsymbol{N}_u = [5, 3, 1]\,, \boldsymbol{N}_v = [1, 1, 1]\,, \boldsymbol{N}_\psi = [3, 3, 1]$, $\boldsymbol{N}_{u,EM} = [3, 1, 1]\,, \boldsymbol{N}_{v,EM} = [2, 3, 1]\,, \boldsymbol{N}_{\psi,EM} = [3, 1, 1]$, giving a total of 189 trajectories.

sengers, the emergency space can be used to sample emergency maneuvers with maximum accelerations, such that the main search space can be limited to comfortable accelerations.

### 3.4.2 Modeling and acceleration sampling

Several vessel models of the milliAmpere test platform exist [6], also including thruster modeling and control allocation [7]. Considering this will introduce the azimuth angle $\alpha$ as a new state. The system is considered sufficient for the scope of this thesis, neglecting control allocation. The vessel model is identified using the current milliAmpere test platform, which has an under dimensioned thruster system. Thus, the thruster limits $(\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max})$ are assigned some values considered realisitc for simulations. Neglecting external forces, the vessel model (2.14) gives

$$\dot{\boldsymbol{\nu}}_{min} = \boldsymbol{M}^{-1} \left( \boldsymbol{\tau}_{min} - \boldsymbol{C}\left(\boldsymbol{\nu}\right) \boldsymbol{\nu} - \boldsymbol{D}\left(\boldsymbol{\nu}\right) \boldsymbol{\nu} \right) \tag{3.18}$$

$$\dot{\boldsymbol{\nu}}_{max} = \boldsymbol{M}^{-1} \left( \boldsymbol{\tau}_{max} - \boldsymbol{C}\left(\boldsymbol{\nu}\right) \boldsymbol{\nu} - \boldsymbol{D}\left(\boldsymbol{\nu}\right) \boldsymbol{\nu} \right). \tag{3.19}$$

where $\dot{\boldsymbol{\nu}}_{min} = [\dot{u}_{min}, \dot{v}_{min}, \dot{r}_{min}]^\top$ and $\dot{\boldsymbol{\nu}}_{max} = [\dot{u}_{max}, \dot{v}_{max}, \dot{r}_{max}]^\top$ are the minimum and maximum possible accelerations.

The set of accelerations is then sampled as

$$\dot{\boldsymbol{u}}_{samples} = \{\dot{u}_1, \ldots, \dot{u}_{N_u}\} \tag{3.20}$$

$$\dot{\boldsymbol{v}}_{samples} = \{\dot{v}_1, \ldots, \dot{v}_{N_v}\} \tag{3.21}$$

$$\dot{\boldsymbol{r}}_{samples} = \left\{\dot{r}_1, \ldots, \dot{r}_{N_\psi}\right\} \tag{3.22}$$

from the set of possible accelerations

$$A = \left\{(\dot{u}, \dot{v}, \dot{r}) \in \mathbb{R}^3 | \dot{u} \in [\dot{u}_{min}, \dot{u}_{max}], \dot{v} \in [\dot{v}_{min}, \dot{v}_{max}], \dot{r} \in [\dot{r}_{min}, \dot{r}_{max}]\right\}. \tag{3.23}$$

The accelerations are aligned with the desired accelerations described in Section 3.4.3.

### 3.4.3 Desired acceleration

The LOS approach described in Section 2.5.2 is used to find the desired accelaration $\dot{\boldsymbol{\nu}}'_d = [\dot{u}'_d, \dot{v}'_d, \dot{r}'_d]$. Zero sway speed and sideslip is desired, giving

$$v_{d,LOS} = 0. \tag{3.24}$$

Equation (2.26) and (2.81) calculates $\psi_{LOS} = \chi_{LOS}$ and $u_{LOS} = U_{d,LOS}$, respectively. The desired accelerations $\dot{v}_{d,LOS}, \dot{u}_{d,LOS}$ is calculated by (2.82) and $\dot{\psi}_{LOS}$ by (2.83).

### 3.4.4 Desired and predicted trajectories

Desired acceleration, velocity and heading trajectories for $(u, v, r)$ are calculated according to (2.84)-(2.88), and the set of desired velocities becomes:

$$\mathcal{U}_d = \{u_{d,1}(t), \ldots, u_{d,N_u}(t)\} \times \{v_{d,1}(t), \ldots, v_{d,N_v}(t)\} \times \left\{\psi_{d,1}(t), \ldots, \psi_{d,N_\psi}(t)\right\}, \tag{3.25}$$

with a desired velocity trajectory denoted as $\boldsymbol{u}_d(t) = [u_d, v_d, \psi_{d,i}]^\top$. The milliAmpere vessel model is unstable in heading when operating at maximum surge or sway velocities. To avoid such unstable behavior, the set of feasible trajectories is given as

$$\mathcal{U}_{d,f} = \{\mathcal{U}_d | u_{d,i}(t) \in [u_{min}, u_{max}], v_{d,i}(t) \in [v_{min}, v_{max}], r_{d,i}(t) \in [r_{min}, r_{max}]\} \tag{3.26}$$

The limits $\boldsymbol{\nu}_{min} = [u_{min}, v_{min}, r_{min}]^\top$, $\boldsymbol{\nu}_{max} = [u_{max}, v_{max}, r_{max}]^\top$ are determined through simulations of step inputs. This is a simple approach to removing infeasible velocities, but proved suffient for the scope of this thesis through simulations. Another approach is using the vessel model and actuator saturation constraints [2].

The predicted velocity and heading trajectories are calculated from $\mathcal{U}_{d,f}$ as described in Section 2.5.2, and the set of predicted velocity and heading trajectories becomes

$$\bar{\mathcal{U}} = \{\bar{u}_1(t), \ldots, \bar{u}_{N_u}(t)\} \times \{\bar{v}_1(t), \ldots, \bar{v}_{N_v}(t)\} \times \left\{\bar{\psi}_1(t), \ldots, \bar{\psi}_{N_\psi}(t)\right\}. \tag{3.27}$$

The predicted pose trajectories are calculated according to:

$$\dot{\bar{\boldsymbol{p}}} = \begin{bmatrix} \cos(\bar{\psi}) & -\sin(\bar{\psi}) \\ \sin(\bar{\psi}) & \cos(\bar{\psi}) \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix}. \tag{3.28}$$

Integrating (3.28), the set of predicted pose trajectories becomes

$$\bar{\mathcal{H}} = \left\{ \bar{\boldsymbol{\eta}} \left( t; \bar{u}\left(t\right), \bar{v}\left(t\right), \bar{\psi}\left(t\right) \right) \mid \left( \bar{u}\left(t\right), \bar{v}\left(t\right), \bar{\psi}\left(t\right) \right) \in \bar{\mathcal{U}} \right\}. \tag{3.29}$$

### 3.4.5 Trajectory alignment

The trajectory alignment is calculated similarly to (2.101), considering the Euclidean and heading error:

$$\begin{aligned} \text{align}(\bar{\boldsymbol{\eta}}(t); \boldsymbol{\eta}_d(t)) = w_p \int_{t_0}^{t_0+T_{al}} \left\| \begin{bmatrix} \bar{N}(\gamma) \\ \bar{E}(\gamma) \end{bmatrix} - \begin{bmatrix} N_d(\gamma) \\ E_d(\gamma) \end{bmatrix} \right\|_2 d\gamma \\ + w_\psi \int_{t_0}^{t_0+T_{al}} |\Upsilon(\bar{\psi}(\gamma) - \psi_d(\gamma))| \, d\gamma, \end{aligned} \tag{3.30}$$

where $w_p, w_\psi \geq 0$ controls the relative weighting of the Euclidean and heading error, $\boldsymbol{\eta}_d(t) = [N_d, E_d, \psi_d]^\top$ is the reference trajectory received from the trajectory planner, and $T_{al}$ is the considered prediction horizon.

Tracking a pose reference trajectory containing speed and/or course maneuvers, it might be impossible for the BC-MPC algorithm to generate a trajectory that aligns with those maneuvers if they appear at different time intervals than for the prediction horizon, as illustrated in Figure (3.8). The first maneuver of the prediction horizon is considered more critical to align with the trajectory than the last one, as only the first maneuver executes before the BC-MPC algorithm replans. As the reference trajectory is replanned in the mid-layer when the tracking error grows too big, as described in Section 3.2.1, it is considered sufficient with a limited prediction horizon in the calculation of trajectory alignment for the hybrid architecture. The prediction horizon considered for the trajectory alignment is therefore

$$T_{al} = \begin{cases} T_1 + T_2 & \text{for Hybrid} \\ T_{full} & \text{for BC-MPC.} \end{cases} \tag{3.31}$$

An approach considering along-track and cross-track error was implemented and tested. However, it is not included in the algorithm as limiting the prediction horizon gives satisfying tracking results, and further consideration of along-track and cross-track errors are left to future work.

### 3.4.6 Moving obstacle representation

Similarly to the moving obstacle representation of the MP-VP, the axis of the moving obstacle regions for the BC-MPC algorithm is made proportional to the

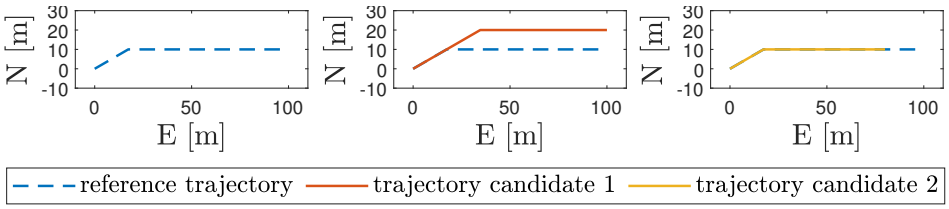



**Figure 3.8:** Example of a reference trajectory with an initial position $[E_0, N_0]^\top = [0, 0]^\top$, and a desired speed of $2m/s$, that makes a course maneuver of $30°$ after $10s$, or $20m$ from the initial position. The two trajectory canditates are maneuvering after $20s$. Candidate 1 keeps the desired speed of $2m/s$, and hence makes the course maneuver after $40m$ of travel. Candidate 2 keeps a speed of $1m/s$ until the maneuver, such that the course maneuver happens at the desired position, at a cost of $10s$ time lag.

separation length of the predefined paths for all regions:

$$a_k = l_{a,i} + C_{a,k} l_{sep} \tag{3.32}$$
$$b_k = l_{b,i} + C_{b,k} l_{sep} \tag{3.33}$$
$$c_k = l_{b,i} + C_{c,k} l_{sep} \tag{3.34}$$

where $C_{a,k}, C_{b,k}, C_{c,k} \geq 0$ are proportional constants, with

$$C_{c,k} = \begin{cases} C_{b,k} & \text{if } l_{sep} < l_{sep,COLREGs} \\ C_{b,k} + C_{COLREGs} & \text{if } l_{sep} \geq l_{sep,COLREGs}. \end{cases} \tag{3.35}$$

The lengths $l_{a,i}, l_{b,i}$ are given as

$$l_{a,i} = l_{o,i} + l_{os} \tag{3.36}$$
$$l_{b,i} = w_{o,i} + l_{os}, \tag{3.37}$$

where $(l_{o,i}, w_{o,i})$ are the length and width of the moving obstacle $o_i \in \mathcal{O}$, and $l_{os}$ the length of the ownship, similarly to the definitions of the MP-VP moving obstacle regions in (2.44). The weight is given as

$$w_{mo,i}(t) = \begin{cases} w_{mo,T_1}\ w_{mo} & \text{if } t_0 \leq t \leq T_1 \\ w_{mo} & \text{if } T_1 < t \leq T_{full}, \end{cases} \tag{3.38}$$

where $w_{mo}$ is a constant weight for all moving obstacles $o_i \in \mathcal{O}$, and $w_{mo,T_1}$ is an additional weight giving a higher cost for collisions due to happen during the first maneuver.

### 3.4.7 Velocity cost

The objective function adds a velocity term $\text{vel}(\boldsymbol{u}_d(t)))$ weighted by $w_{vel} \geq 0$

$$\begin{aligned} J\left(\bar{\boldsymbol{\eta}}(t); \boldsymbol{u}_d(t); \boldsymbol{\eta}_d\left(t\right)\right) = & \, w_{al}\text{align}\left(\bar{\boldsymbol{\eta}}(t); \boldsymbol{\eta}_d(t)\right) \\ & + w_{av}\text{avoid}(\bar{\boldsymbol{\eta}}(t)) \\ & + w_{tc}\text{trans}(\boldsymbol{u}_d(t)) \\ & + w_{vel}\text{vel}(\boldsymbol{u}_d(t)). \end{aligned} \tag{3.39}$$

The desired motion for the ferry is a positive surge speed and a zero sway speed. The velocity term penalizes the negative surge speeds and nonzero sway speeds:

$$\text{vel}(\boldsymbol{u}_d(t)) = \int_{t_0}^{t_0+T_{full}} \text{speed\_penalty}(\boldsymbol{u}_d(\gamma))d\gamma \tag{3.40}$$

$$\text{speed\_penalty}(\boldsymbol{u}_d(t)) = w_{sway}|v_d| + \begin{cases} 0 & \text{if} \quad u_d \geq 0 \\ |u_d| & \text{else.} \end{cases} \tag{3.41}$$

# Chapter 4

# Simulation results

This chapter describes the implemented simulation environment and the simulation results. Section 4.1 gives details on the simulator implementation. Section 4.2 lists the simulation parameters and explains the considerations made when tuning the parameters. The simulation scenarios are introduced in Section 4.3, and a set of performance metrics is defined in 4.4. The simulation results of the hybrid architecture are presented in Section 4.5, and the performance of the COLAV architectures are compared for some of the scenarios in Section 4.6. Finally, the results are summarized in Section 4.7 and discussed in Section 4.8.

## 4.1 Simulator

The simulator is implemented in MATLAB and SIMULINK 2019b, and a discrete-time solver is used to perform the simulations.

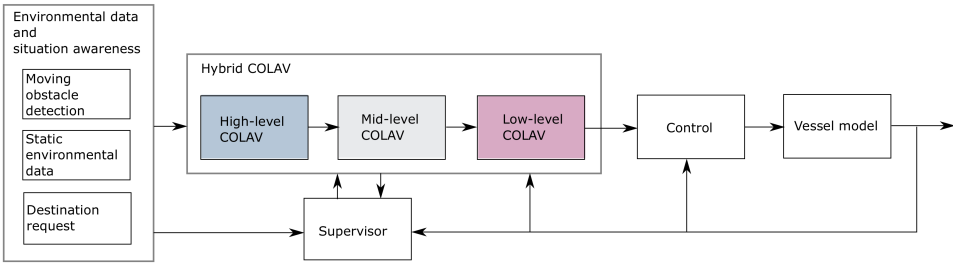



**Figure 4.1:** Simulator overview.

Figure 4.1 shows an overview of the simulator, and the elements of the simulator is summarized as follows:

- The obstacle detection module is significantly simplified. Information about obstacle size, position, course, and speed are assumed known, and measurements assumed perfect. The details of obstacle implementation is given in Section 4.1.1.

- The static environmental data is neglected in this thesis, and testing the system with static obstacles is left for future work. However, BC-MPC is previously demonstrated with successful avoidance of static obstacles through simulations [3], also by the author [1].

- The destination request is given as waypoints.

- The COLAV system is described in Chapter 3. The implementation of MP-VP and the path planner is provided by Emil Hjelseth Thyri, with only minor changes applied.

- The implemented controller is a PID and model feedforward control law, given in Section 4.1.2.

- The implemented vessel model is a fully coupled 3DOF model [6], described in Section 2.1.3.

- The implemented supervisor is described in Section 3.2.1.

For all simulations, external forces are neglected, and all measurements are assumed perfect.

### 4.1.1 Obstacle modeling and maneuvering

The targets in the simulator are defined by an initial pose $\boldsymbol{\eta}_{o,0}$ and velocity $U_o$, and modeled by the 3 DOF model in (2.13) The obstacle maneuvering implementation is significantly simplified: the speed and/or course is changed at a time $t_{man,U}$ and $t_{man,\chi}$, respectively. This gives an instant speed and/or course change with infite acceleration, which is not realistic. However, the approach is considered sufficient for the scope of this Masters's thesis, and is chosen to keep things simple.

### 4.1.2 Motion control

The controller implemented for the simulator combines the PID controller (2.28) and the model-feedfoward controller (2.32), giving the control law

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{PID} + \boldsymbol{\tau}_{FF}. \tag{4.1}$$

## 4.2 Simulation setup

### 4.2.1 Vessel parameters

The vessel parameters are given in Table 4.1. The existing milliAmpere test platform has an under dimensioned thruster system, and $\boldsymbol{\tau}_{min}$, $\boldsymbol{\tau}_{max}$ is scaled up to avoid the limitation of the physical ferry. When operating at maximum surge or sway speed, drifting in the heading is experienced. Therefore, the velocity is limited by $\boldsymbol{\nu}_{min}, \boldsymbol{\nu}_{max}$ to avoid considerable drifting in heading. The minimum and maximum values for thrust are chosen, as stated in Table 4.1, and the minimum and maximum operation velocities are determined from simulations.

**Table 4.1:** Vessel parameters.

| Parameter | Value | Unit |
|---|---|---|
| $\boldsymbol{\tau}_{min}$ | $[-1800, -1800, -1800]^\top$ | Nm |
| $\boldsymbol{\tau}_{max}$ | $[1800, 1800, 1800]^\top$ | |
| $u_{min}$ | -2 | m/s |
| $u_{max}$ | 2 | m/s |
| $v_{min}$ | -2 | m/s |
| $v_{max}$ | 2 | m/s |
| $r_{min}$ | -70 | deg/s |
| $r_{max}$ | 70 | deg/s |

### 4.2.2 COLAV parameters

The COLAV parameters are tuned to fit with the area of operation, the velocity range of the ferry, and the desirable properties from Section 3.1. For both algorithms, the behavior is dependent on the parameters. Several simulations are performed to determine the parameters used in this thesis, but extensive analysis of parameter sensitivity is left for future work. In general, the parameters are chosen such that the BC-MPC algorithm does not penalize behavior that the MP-VP does not. This includes setting the axis of the obstacle regions slightly larger for the MP-VP than for the BC-MPC algorithm. Furthermore, the MP-VP has a longer prediction horizon than the BC-MPC algorithm. The cost function parameters are chosen such that speed maneuvers should be favorized over course maneuvers if the time cost is not too high. Tables 4.2-4.3 list the parameters.

**Table 4.2:** MP-VP parameters.

| Parameter | Value | Unit | Description |
|---|---|---|---|
| $\boldsymbol{T}_{plan}$ | 400 | s | prediction horizon |
| $k_f$ | 1 | | fore gain |
| $k_a$ | 1 | | aft gain |
| $k_s$ | 1 | | starboard gain |
| $k_p$ | 1 | | port gain |
| $C_{ROC,f}$ | 0.5 | | ROC fore length gain |
| $C_{ROC,a}$ | 0.1 | | ROC aft length addition gain |
| $C_{ROC,p}$ | 0.1 | | ROC port length addition gain |
| $C_{HPR,f}$ | 1.2 | | HPR fore length addition gain |
| $C_{HPR,a}$ | 0.5 | | HPR aft length addition gain |
| $C_{HPR,p}$ | 0.5 | | HPR port length addition gain |
| $C_{LPR,f}$ | 2.0 | | LPR fore length addition gain |
| $C_{LPR,a}$ | 0.9 | | LPR aft length addition gain |
| $C_{LPR,p}$ | 0.9 | | LPR port length addition gain |
| $C_{COLREGs}$ | 0.9 | | COLRREGs length addition gain |
| $\alpha_b$ | 30 | deg | branching angle |
| $w_{vel}$ | 100 | | velocity cost weight |
| $w_{time}$ | 10 | | travel time weight |
| $w_{length}$ | 100 | | travel length weight |
| $w_{penalty}$ | 500 | | penalty region weight |

**Table 4.3:** BC-MPC parameters.

| Parameter | Value | Unit | Description |
|---|---|---|---|
| $\boldsymbol{N}_u$ | $[5, 3, 1]$ | | number of surge speed samples |
| $\boldsymbol{N}_v$ | $[1, 1, 1]$ | | number of sway speed samples |
| $\boldsymbol{N}_\psi$ | $[3, 3, 1]$ | | number of heading samples |
| $\boldsymbol{N}_{u_{em}}$ | $[3, 1, 1]$ | | number of surge speed emergency samples |
| $\boldsymbol{N}_{v_{em}}$ | $[2, 3, 3]$ | | number of sway speed emergency samples |
| $\boldsymbol{N}_{\psi_{em}}$ | $[3, 1, 1]$ | | number of heading emergency samples |
| $\boldsymbol{T}$ | $[20, 40, 60]$ | s | prediction horizon |
| $C_{a,0}$ | 0.07 | | collision region major axis gain |
| $C_{a,1}$ | 0.44 | | safety region major axis gain |
| $C_{a,2}$ | 0.90 | | margin region major axis gain |
| $C_{b,0}$ | 0.02 | | collision region minor axis gain |
| $C_{b,1}$ | 0.27 | | safety region minor axis gain |
| $C_{b,2}$ | 0.57 | | margin region minor axis gain |
| $C_{COLREGs}$ | 0.7 | | COLREGs gain |
| $\gamma$ | 0.08 | | obstacle cost parameter |
| $w_\psi$ | 20 | | angular error weight |
| $w_{al}$ | 1000 | | trajectory alignment weight |
| $w_{mo}$ | 6000 | | moving obstacle avoidance weight |
| $w_{mo,T_1}$ | 10 | | moving obstacle avoidance weight, maneuver one |
| $w_{tc,u}$ | 50 | | surge translational cost weight |
| $w_{tc,v}$ | 500 | | sway translational cost weight |
| $w_{tc,\psi}$ | 500 | | heading translational cost weight |
| $w_{vel}$ | 5000 | | velocity cost weight |
| $w_{sway}$ | 100 | | sway speed weight |
| $\Delta$ | 50 | m | lookahead distance |
| $\gamma_s$ | 0.005 | | LOS along-track distance gain |

## 4.3 Scenarios

A set of scenarios is simulated to evaluate the performance of the hybrid COLAV system. The scenarios include water elevator scenarios with shorter, straight-line crossings, and a water bus scenario over a longer distance with varying separation lengths between the paths. All scenarios include multiple obstacles.

For all transits, the desired transit velocity is set to $1.5m/s$. The starboard axis of the dynamic representation of moving obstacles are extended with a gain $C_{COLREGs}$ for all separation lengths, and $l_{sep,COLREGs} = 0$ for all scenarios. Table 4.4 gives an overview of the simulated scenarios, which are further explained in the following of this section.

**Table 4.4:** Overview of the simulated scenarios.

| Scenario | Transit | Traffic picture | Obstacle behavior |
|---|---|---|---|
| Scenario 1 | Water elevator | CS-O-CP | Constant |
| Scenario 2 | Water elevator | CS-HO-CP | Constant |
| Scenario 3 | Water elevator | High-traffic | Speed-maneuvering |
| Scenario 4 | Water bus | Varying | Constant |

### Scenario 1-2: water elevator with constant obstacle behavior

Scenario 1-2 covers multi-obstacle traffic pictures, challenging different maneuvering rules of the COLREGs. The obstacles keep constant speed and course, and it is expected that the trajectory provided by the MP-VP is collision-free and sufficient to handle the situations. In Scenario 1, the ferry faces an obstacle crossing from port (CP), an obstacle being overtaken (O), and an obstacle crossing from starboard (CS), giving the traffic picture CP-O-CS. Scenario 2 is similar, but the second obstacle is in a head-on (HO) situation instead of being overtaken, giving the traffic picture CP-HO-CS.

### Scenario 3: water elevator with speed-maneuvering obstacles

In Scenario 3, the ferry is in a high-traffic situation, with an obstacle being overtaken, and several obstacles crossing from port and starboard. Most of the obstacles keep constant speed and course, except for two of the crossing obstacles, which makes speed maneuvers. Firstly, one of the obstacles crossing from port speeds up, initiating a collision situation. Secondly, an obstacle crossing from starboard reduces its speed significantly, and the ferry is in a risk of getting stuck. The scenario challenges the hybrid architecture's capabilities of avoiding collision situations with suddenly maneuvering obstacles, and saving the ferry from getting stuck when facing slowly moving obstacles.

### Scenario 4: water bus with constant obstacle behavior

In this scenario, the ferry travels a longer transit with varying separation lengths between the paths, and several obstacles entering and leaving the set of considered obstacles. The obstacles have constant behavior, and this scenario challenges the capability of the MP-VP to maneuver in accordance with the COLREGs rules and the dynamic representation of moving obstacles.

## 4.4 Performance metrics

A set of quantitative performance metrics are used to evaluate the COLAV systems. They are chosen based on the identified desirable properties of passenger ferries in Section 3.1, and consider aspects as safety, passenger comfort, and efficiency.

The total travel time (TT) denotes the total travel time is defined as

$$TT(t) = t - t_0. \tag{4.2}$$

The DCO between the ownship and the obstacles at time $t$ is calculated as

$$DCO(t) = \min_{o,i \in \mathcal{O}} \left\| \boldsymbol{p}(t) - \boldsymbol{p}_{o,i}(t) \right\|_2, \tag{4.3}$$

where $\mathcal{O}$ is the set of all moving obstacles. The minimum distance to obstacle (MDO) is the minimum distance to any obstacle during the transit, and is calculted as

$$MDO(t) = \min_{\gamma \in [t_0, t]} DCO(\gamma). \tag{4.4}$$

$DCO(t)$ reflects the risk taken at the time instance $t$, and plotting $DCO(t)$ for the entire transit gives a picture on the risk taken throughout the transit. $MDO(t)$ gives the minimum value, and reflects tha maximum risk taken during the transit.

Similarly to [4] and [1], The integral of absolute speed rate (IASR) and integral of absolute yaw rate (IAYR) are used to measure the accumulated changes in yaw and speed, respectively, and are given as

$$IASR(t) = \int_{t_0}^{t} |\dot{U}(\gamma)| d\gamma, \tag{4.5}$$

$$IAYR(t) = \int_{t_0}^{t} |r(\gamma)| d\gamma. \tag{4.6}$$

It is desirable to keep both integral of absolute yaw rate (IAYR) and integral of absolute speed rate (IASR) low, as a lot of speed- and course- maneuvering is neither good for the readability of the ferry behavior, the passenger comfort or the power consumption.

Furthermore, the absolute value of the linear acceleration, $|\dot{U}(t)|$, and the absolute value of the angular acceleration, $|\dot{r}(t)|$, are considered, as they are important for passenger comfort, as described in Section 3.1.

The accumulated power consumption is directly related to the power efficency of the passenger ferry, and is calculated as the integral of power consumption (IW):

$$IW(t) = \int_{t_0}^{t} \boldsymbol{\tau}(\gamma)^\top \boldsymbol{\nu}(\gamma) d\gamma. \tag{4.7}$$

Finally, the computational time of the algorithms should be considered as they must be able to run real-time. A quantitative measurement is not included in this Master's thesis, but consideration of computational time is included in the discussion of the results.

## 4.5 Hybrid COLAV system simulations

### 4.5.1 Scenario 1-2: water elevator with constant obstacle behavior

In these scenarios, all obstacles keep constant speed and course and the ferry follows the reference trajectory generated by the MP-VP algorithm.

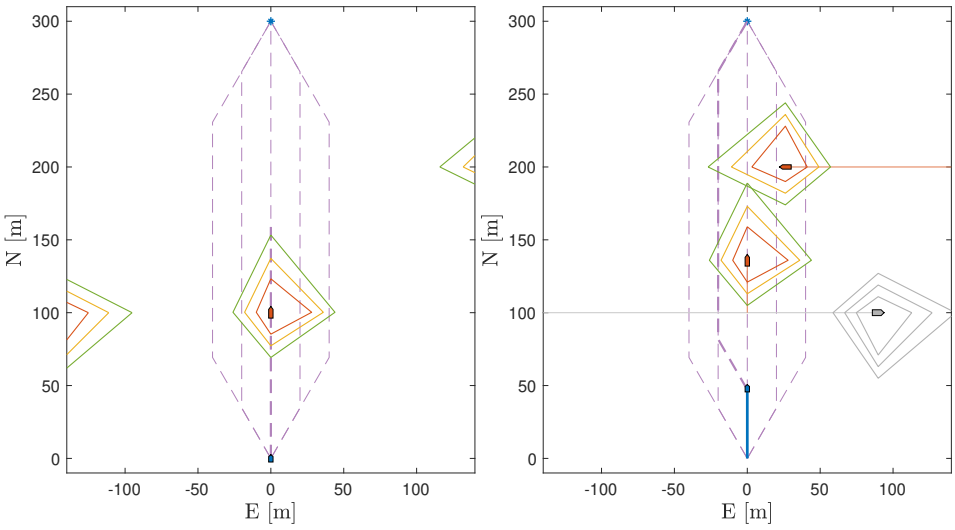
#### Scenario 1: CS-O-CP

In this scenario, the ferry faces the traffic picture CS-O-CP, with one obstacle crossing from the port, one obstacle being overtaken, and one obstacle crossing from starboard, all with constant speed and course. Figure 4.2 shows the situation at time $t = 1s$, $t = 120s$, $t = 200s$, and $t = 296s$, respectively, and Figure 4.3 shows the velocity profiles and the time of replanning by the MP-VP.

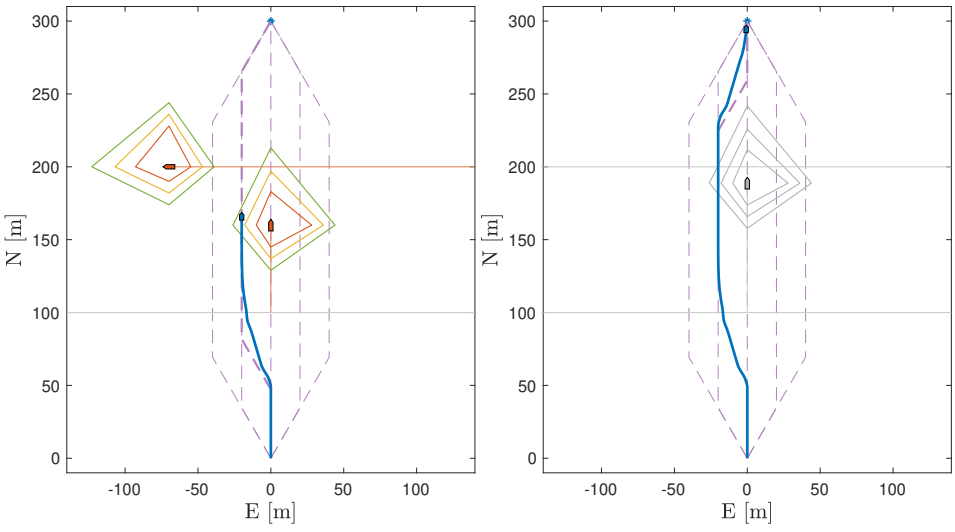
As shown in Figure 4.2a, all moving obstacles are considered by the MP-VP at time $t = 1s$, and the initial planned trajectory is along the center-path, following behind the moving obstacle being overtaken at a slow speed. At $t = 120s$, the moving obstacle crossing from port has passed and is no longer considered. Numbering the paths 1 to 5 from port to starboard seen in the traveling direction, the MP-VP replans the trajectory along $path2$, as shown in Figure 4.2b. The ferry follows this trajectory at desired transit speed, and takes a port maneuver and passes the overtaken obstacle on its port side, as seen in Figure 4.2c. It passes behind the obstacle crossing from starboard before it replans the trajectory at $t = 239s$ when the obstacle crossing from starboard is no longer to be considered. Figure 4.2d shows that the ferry maneuvers back to the center path, slows down, and reaches the destination. The tracking of the reference path is not perfect, as the ferry cuts corners in the turns. This is because the maneuvers of the BC-MPC algorithm does not match the reference-path exactly.

As expected, the trajectories planned by the MP-VP are sufficient to avoid collisions in this scenario, as the obstacles keep constant speed and course. Furthermore, the maneuvers comply with the COLREGs rules. For the initial velocity planning at $t = 1s$, the ferry has a stand-on obligation towards the moving obstacle crossing from left. The ferry does not keep its initial zero-speed until the obstacle has passed. However, with such small distances and velocities, it can not be expected to wait for a vessel crossing $100m$ away. The ferry keeps its course and slow constant speed, and avoids making a port maneuver until the obstacle has passed. Hence,

the stand-on obligation of the COLREGs Rule 17 is fulfilled. Next, the ferry passes the obstacle being overtaken on its port side, and behind the obstacle crossing from starboard. The maneuvers are apparent and made in ample time, and the ferry acts under the COLREGs rules 8, 13, and 15-17.




**(a)** $t = 1s$ **(b)** $t = 120s$




**(c)** $t = 200s$ **(d)** $t = 296s$

**Figure 4.2:** Scenario 1: North-East plots of the transit with the traffic picture CP-O-CS and constant behavior of the moving obstacles. The plots show the results using the hybrid COLAV system. The objects considered by the MP-VP at the current time are shown in colors, and the obstacle that are not considered in grey.

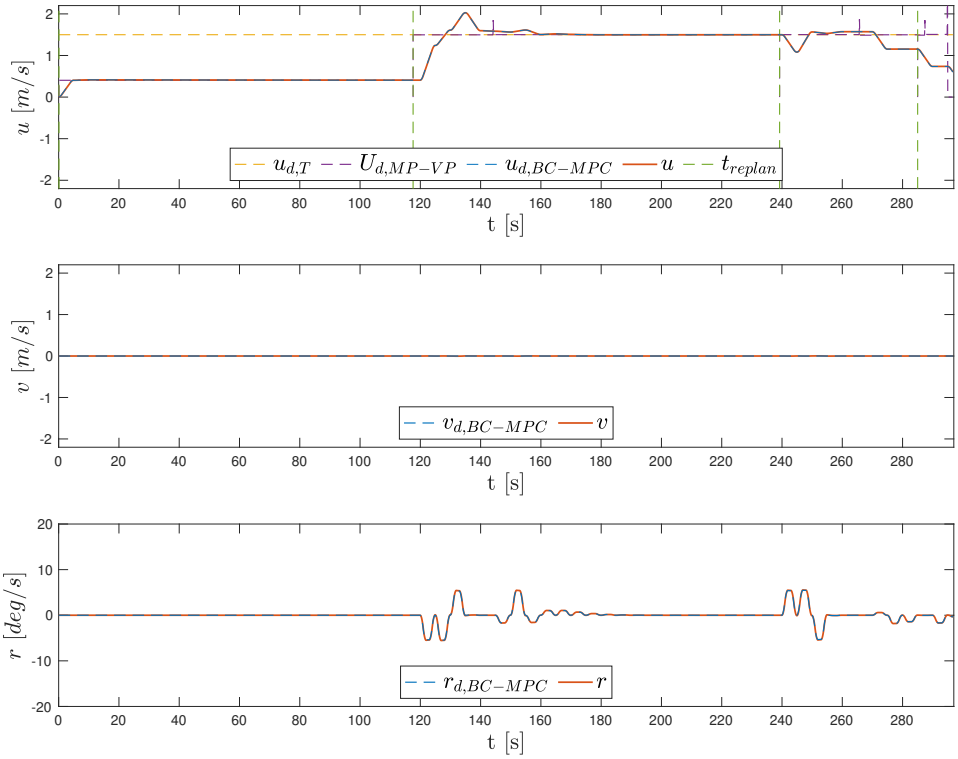
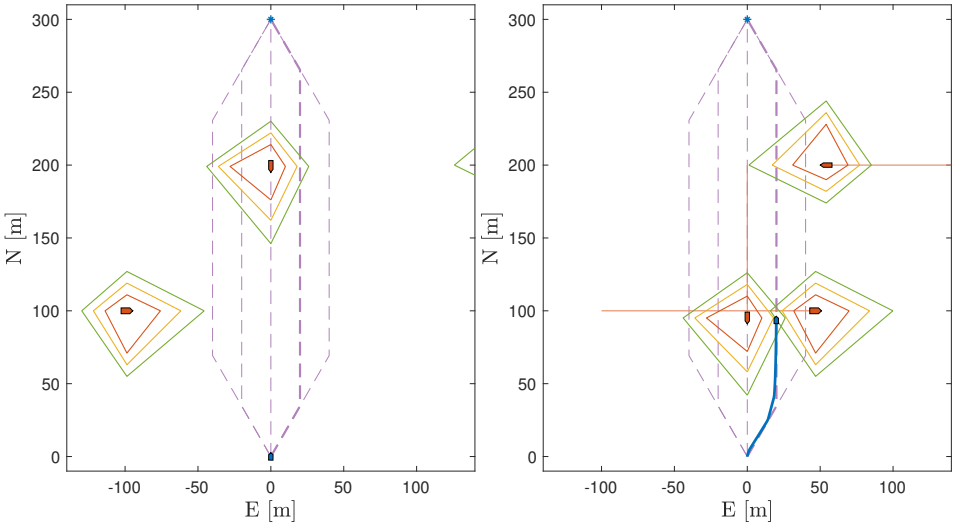



**Figure 4.3:** Scenario 1: velocity plots of the transit with the traffic picture CP-O-CS and constant behavior of moving obstacles. The plots show the results using the hybrid COLAV system. The desired transit speed is denoted $u_{d,T}$, the desired speed calculated by the MP-VP by $U_{d,MP-VP}$, and the time of replanning by the MP-VP by $t_{replan}$. The reference velocities calculated by the BC-MPC algorithm are denoted by $[u_{d,BC-MPC}, v_{d,BC-MPC}, r_{d,BC-MPC}]^\top$, and the velocities of the ferry by $[u, v, r]^\top$.

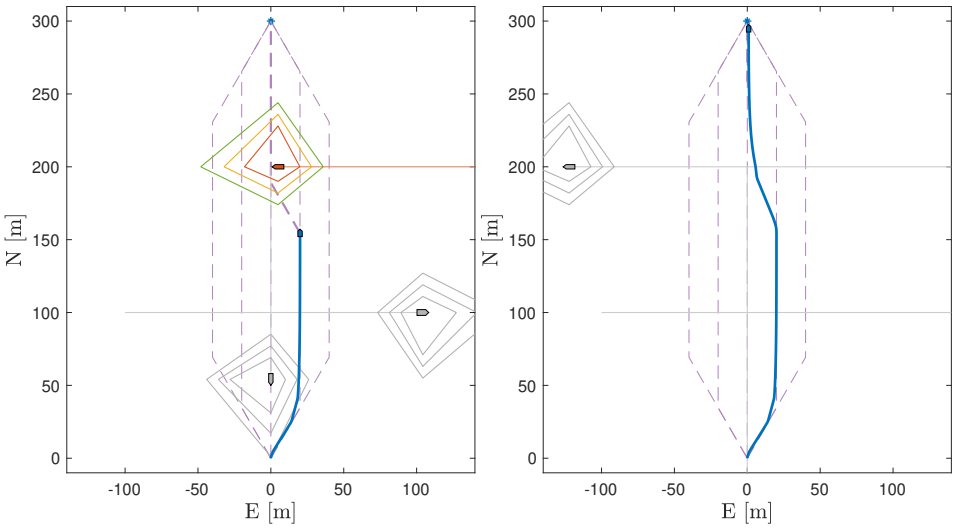
**Scenario 2: CP-HO-CS**

Scenario 2 is similar to Scenario 1, with moving obstacles crossing from port and starboard. The obstacle traveling along the center path is, however, in a head-on situation with the ferry. Figures 4.4 - 4.5 shows North-East plots and the velocity profiles. At $t = 1s$, all three obstacles are considered, and the MP-VP plans a trajectory along $path4$, with a starboard maneuver, as shown in Figure 4.4a. The ferry travel with a reduced speed until the obstacle crossing from port has passed the ferry at about $t = 90s$. The ferry passes behind the obstacle crossing from port, and on the correct side of the obstacle in the head-on situation, as shown in Figure 4.4b. When the passed obstacles are no longer considered, the trajectory is replanned to the center path at the desired transit speed. The ferry follows this trajectory to the destination, passing behind the last obstacle, as seen in Figures 4.4c-4.4d.

Similarly to Scenario 1, the ferry travels at a slow speed until the obstacle crossing from port has passed, once again avoiding port maneuvers. The ferry is, however, in a head-on situation and makes a starboard maneuver, before returning to the center pathw passing behind the obstacle crossing from starboard. The maneuvers are apparent and made in ample time. Hence, the behavior in Scenario 2 is in accordance with the COLREGs rules 8, 14-17.




**(a)** $t = 1s$ **(b)** $t = 105s$




**(c)** $t = 146s$ **(d)** $t = 252s$

**Figure 4.4:** Scenario 2: North-East plots of the transit with the traffic picture CP-HO-CS and constant behavior of the moving obstacles. The plots show the results using the hybrid COLAV system. The obstacles considered by the MP-VP at the current time are shown in colors, and the obstacles that are not considered in grey.

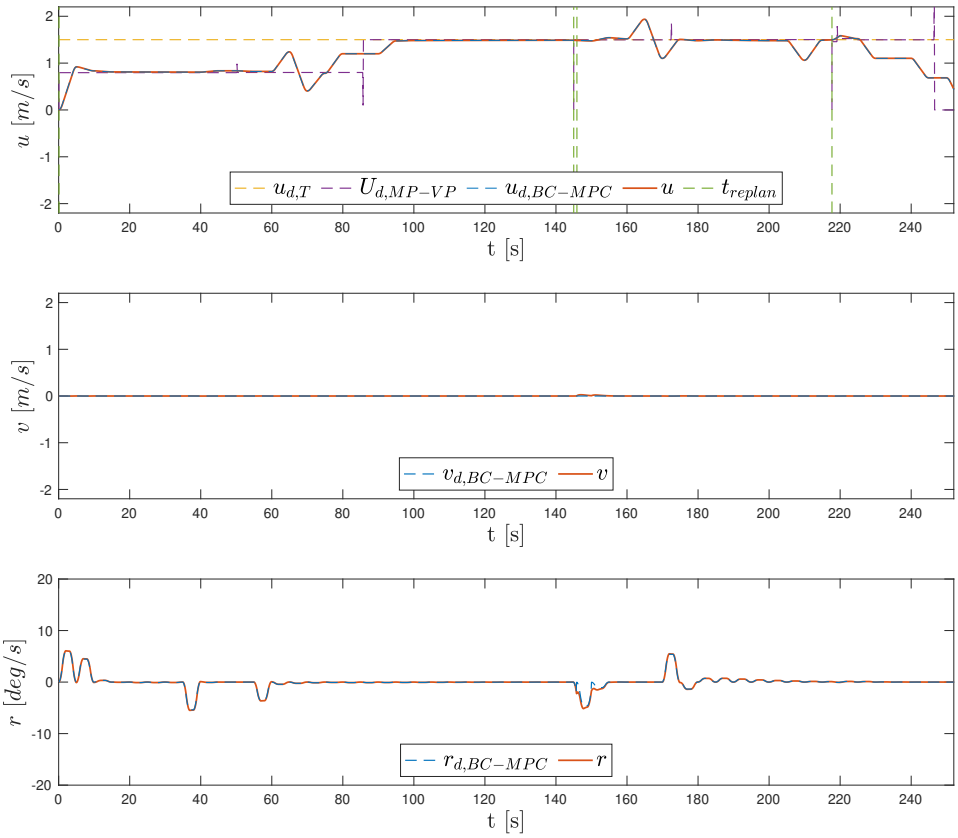


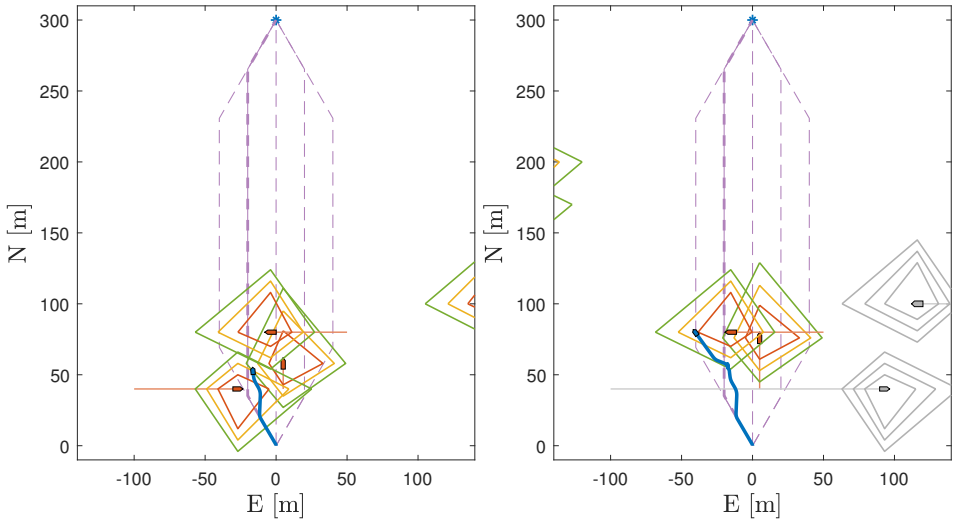

**Figure 4.5:** Scenario 2: velocity plots of the transit with the traffic picture CP-HO-CS and constant behavior of the moving obstacles. The plots show the results using the hybrid COLAV system. The desired transit speed is denoted $u_{d,T}$, the desired speed calculated by the MP-VP by $U_{d,MP-VP}$, and the time of replanning by the MP-VP by $t_{replan}$. The reference velocities calculated by the BC-MPC algorithm are denoted by $[u_{d,BC-MPC}, v_{d,BC-MPC}, r_{d,BC-MPC}]^\top$, and the velocities of the ferry by $[u, v, r]^\top$.

### 4.5.2 Scenario 3: water elevator with speed-maneuvering obstacles

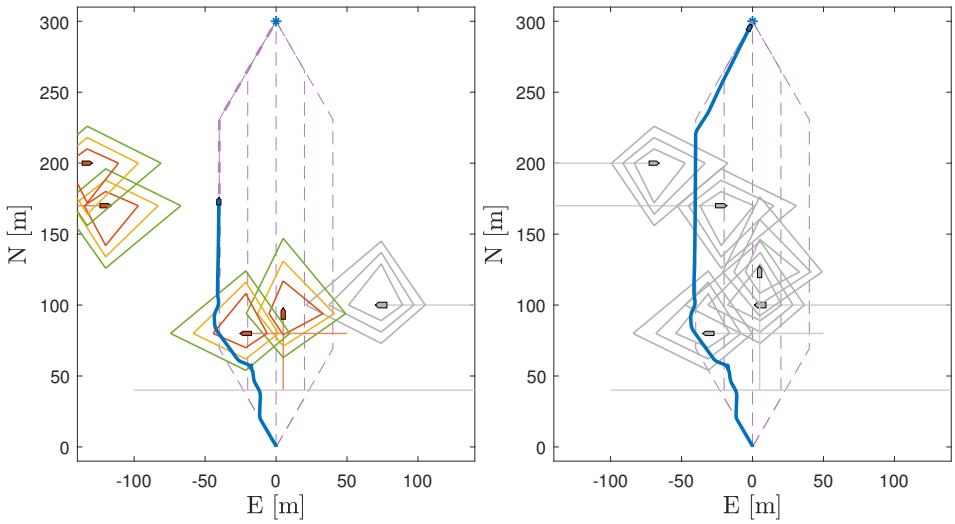
In this scenario, the ferry faces a high-traffic picture with speed maneuvering obstacles, as seen from the North-East plots in Figure 4.6. At $t = 1s$, the MP-VP plans a trajectory along $path2$. However, after about $30s$, the obstacle crossing from port at $N = 40m$ speeds up, and the ferry modifies the turn towards $path2$ to avoid passing to close to the obstacle, as seen in Figure 4.6a. At about $t = 50s$, the ferry stops and waits for the obstacle crossing from starboard at $E = 80m$. Between $t = 75s$ and $t = 90s$, the ferry has a very slow negative surge speed to get outside the obstacle region. Note that the MP-VP does not replan during this period, as the distance to the obstacle is too short. At about $t = 100s$, the ferry had planned to speed up and pass behind the obstacle crossing from starboard. However, the obstacle has slowed significantly down, and the BC-MPC algorithm modifies the maneuvers and makes a port maneuver to pass in front of the obstacle, as seen in Figure 4.6b. At about $t = 130s$, the MP-VP replans the trajectory along $path1$, due to the Euclidean error. The ferry follows this trajectory until the destination is reached, as seen in figures 4.6c-4.6d.

The collision situation is first solved with speed-maneuvers by the MP-VP, but when the obstacles change their speed, this plan is no longer valid. The BC-MPC algorithm solves the situations with course maneuvers and succeeds in maneuvering away from the moving obstacle which increases its speed and saving the ferry from getting stuck. In this situation, it would be more effective to make the port maneuver and pass in front of the obstacle passing from starboard as soon as it slows down, instead of waiting for a while before doing the port maneuver. As seen from the velocity profiles in Figure 4.7, this results in many maneuvers of both speed and course.

In this scenario, it is hard for the MP-VP to make maneuvers in compliance with the COLREGs, as most of the paths are blocked. An example is the initial port maneuver, which obeys COLREGs Rule 17. The best options would probably be a starboard maneuver, but the branching paths with the fixed branching angle at $\alpha_b = 30^\circ$ are blocked by the moving obstacle being overtaken. Furthermore, the added COLREGs-length of the moving obstacle representation participates to block $path5$.




**(a)** $t = 60s$ **(b)** $t = 120s$




**(c)** $t = 180s$ **(d)** $t = 278s$

**Figure 4.6:** Scenario 3: North-East plots of the transit with speed-manuvering moving obstacles and the hybrid COLAV system. The obstacles considered by the MP-VP at the current time are shown in colors, and the obstacles that are not considered in grey.

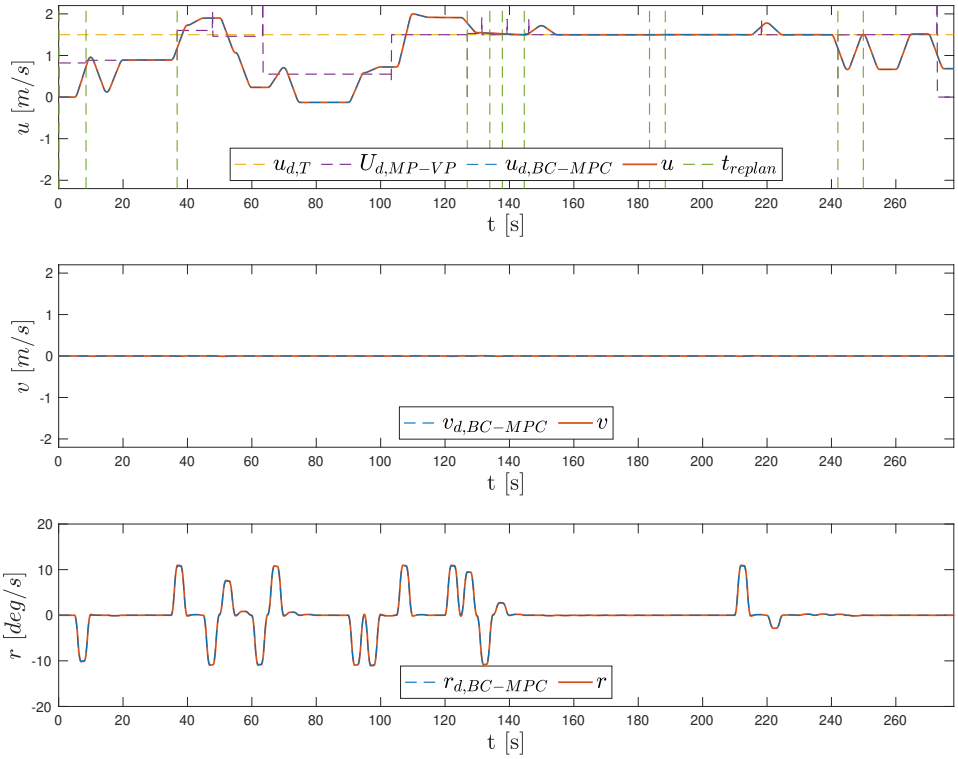



**Figure 4.7:** Scenario 3: velocity plots from the transit with speed-manuvering moving obstacles and the hybrid COLAV system. The desired transit speed is denoted $u_{d,T}$, the desired speed calculated by the MP-VP by $U_{d,MP-VP}$, and the time of replanning by the MP-VP by $t_{replan}$. The reference velocities calculated by the BC-MPC algorithm are denoted by $[u_{d,BC-MPC}, v_{d,BC-MPC}, r_{d,BC-MPC}]^\top$, and the velocities of the ferry by $[u, v, r]^\top$.

### 4.5.3 Scenario 4: water bus with constant obstacle behavior

In this scenario, the ferry encounters a longer transit, starting out in an area with paths separated by $12m$, before making a turn and entering the open sea with a separation length of $30m$ between the paths. The path planner has modified the paths, such that all five paths are continuous. Several obstacles are crossing paths with the ferry along the way, all keeping constant speed and course. The North-East plots and the velocities from the simulation are shown in figures 4.8-4.9.

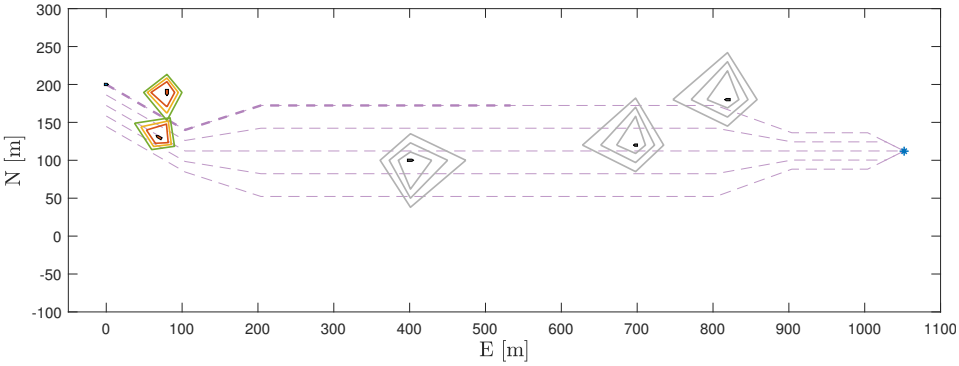
At $t = 1s$, only two obstacles are considered, and the MP-VP plans a trajectory along $path1$, with a slower speed than the desired transit speed. Hence, the ferry avoids a collision situation with the obstacle passing from port, as illustrated in Figure 4.8a. The ferry follows this trajectory and speeds up to the desired transit speed when the obstacle crossing from port has passed. The trajectory is replanned several times as the set of considered obstacles changes. However, the reference trajectory is not significantly changed before $t = 156s$, when the obstacle approaching along $path1$ is considered. As seen from the situation overview in Figure 4.8b, and the velocities in Figure 4.9, the ferry now plans to stop and wait. When the obstacle crossing from starboard has passed, the trajectory is replanned at $t = 234s$ (Figure 4.8c), following $path2$ with desired transit speed. Figure 4.8d illustrates the next replanning worth commenting. When the obstacle crossing from starboard at $E = 960m$ is considered, the trajectory is replanned along $path2$ at a reduced speed. At $t = 736s$, the trajectory is replanned as the prediction horizon of the BC-MPC algorithm exceeds the reference trajectory. The ferry follows the center path at desired transit speed until the destination is reached, as seen in figures 4.8e-4.8f.

The initial maneuver of driving at a slow speed until the two first obstacles have passed complies with the COLREGs, as following any path at transit speed or faster will result in getting inside the ROC or passing in front of the obstacle crossing from port. Considering the replanning at $156s$ in Figure 4.8b, when new obstacles are detected, $path4$ and $path5$ are the only paths that are not blocked by a ROC. However, to reach those paths, the ferry has to cross ahead of both the obstacle traveling in the north direction at $E = 300m$ and the obstacle traveling in the west direction at $N = 120m$. This should be avoided according to the COLREGs Rule 15. When the two obstacles have passed, the ferry makes a starboard maneuver to pass the obstacle approaching at $N = 180$ on its port side, which is the correct action for head-on situations, according to the COLREGs Rule 14. Lastly, the ferry passes abaft the obstacle driving in the north direction at $E = 960m$.

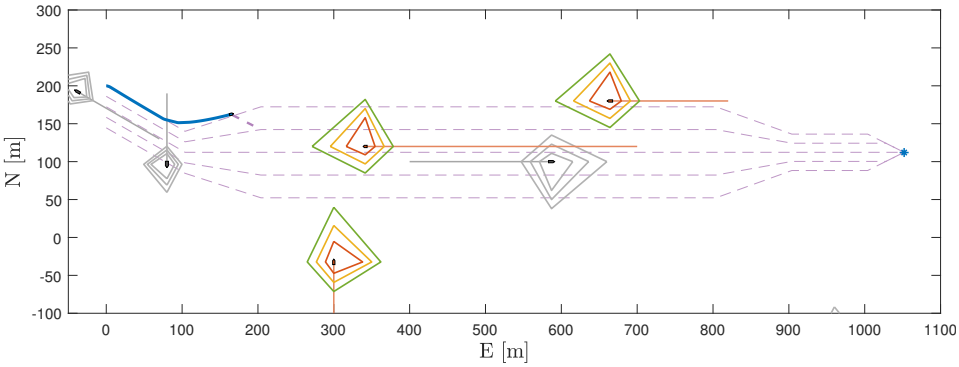
The maneuvers are made in ample time, and as seen from the velocity profiles in Figure 4.9 keeps a constant velocity and course during most of the transit. Hence, the behavior can be considered to be in compliance with the COLREGs rules 8 and 13-17. A better solution to this scenario might be to wait at the initial position until the two first obstacles have passed and then encounter the full transit along $path5$. However, waiting at the transit considering obstacles far away is probably not very effective, as there is a chance the detected obstacles will maneuver, or

that new obstacles appear. Furthermore, in this scenario, only the two obstacles operating in the first region of $l_{sep} = 12$ are considered until the ferry has left this region, which is reasonable.

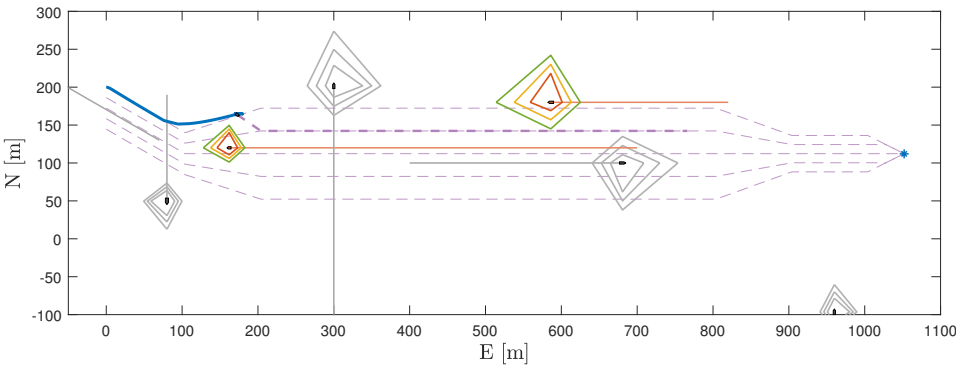
Note that the region sizes change according to the position of the moving obstacles. The ferry passes closer to the moving obstacles in the first and last part of the transit, where the separation length between the paths is the smallest.



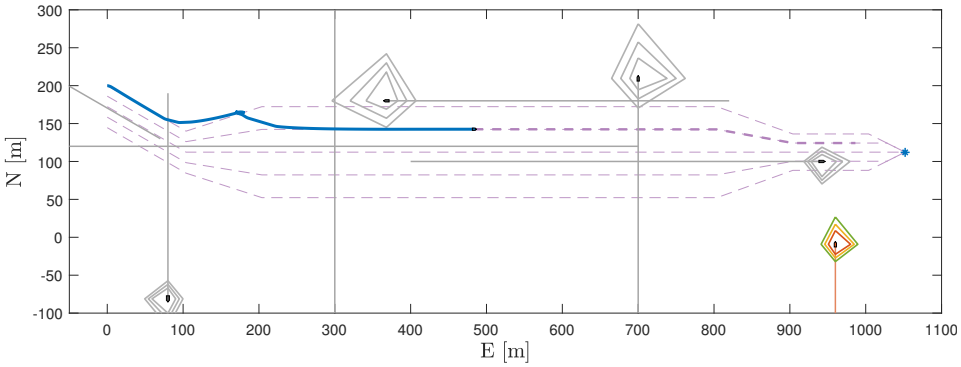

**(a)** $t = 1s$



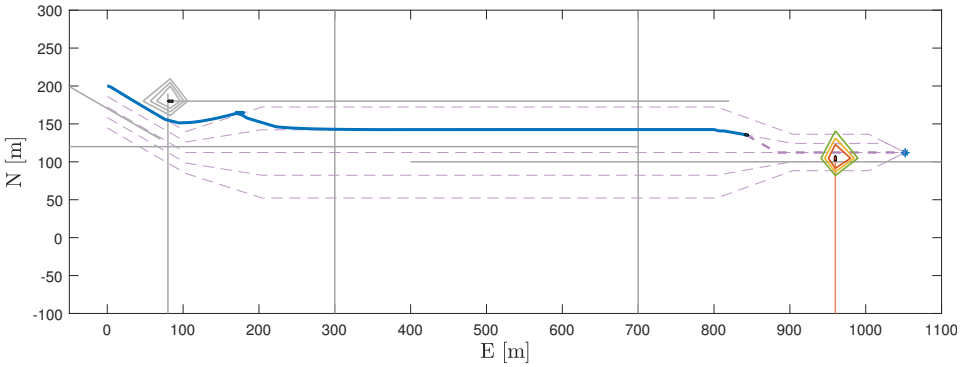

**(b)** $t = 156s$



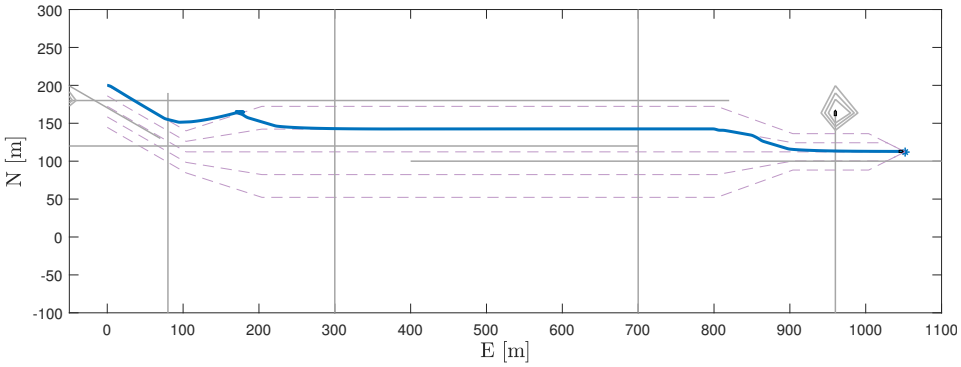

**(c)** $t = 234s$




**(d)** $t = 452s$




**(e)** $t = 737s$




**(f)** $t = 884s$

**Figure 4.8:** Scenario 4: North-East plots of the water bus transit with constant obstacle behavior and the hybrid COLAV system. The obstacles considered by the MP-VP at the current time are shown in colors, and the obstacles that are not considered in grey.

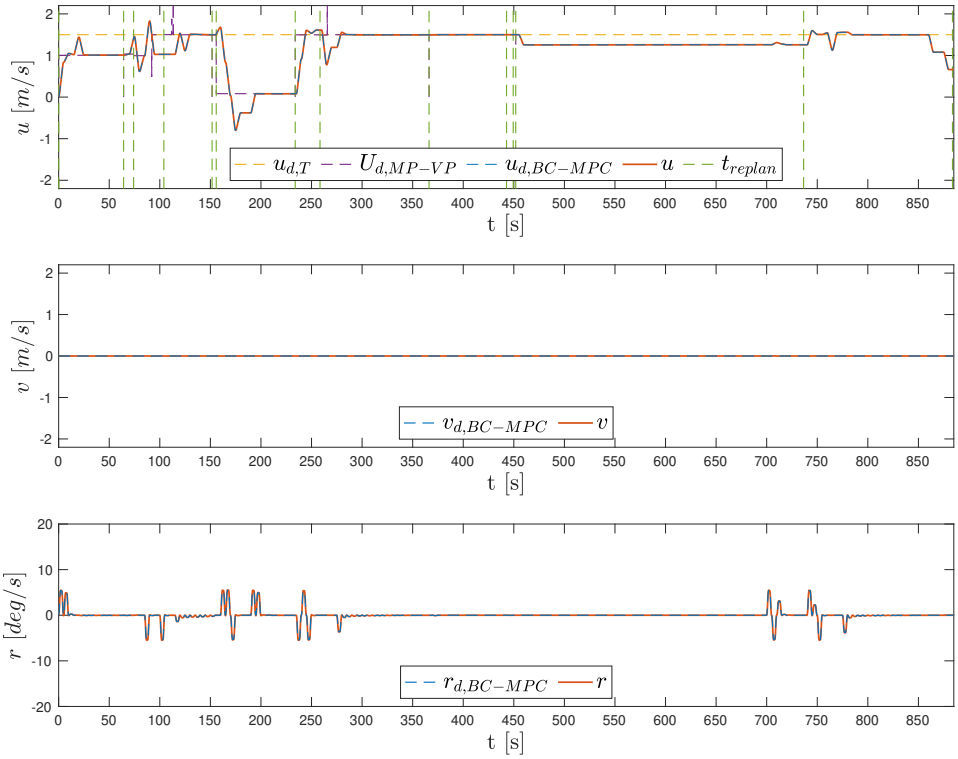



**Figure 4.9:** Scenario 4: velocity plots from the water bus transit with constant obstacle behavior and the hybrid COLAV system. The desired transit speed is denoted $u_{d,T}$, the desired speed calculated by the MP-VP by $U_{d,MP-VP}$, and the time of replanning by the MP-VP by $t_{replan}$. The reference velocities calculated by the BC-MPC algorithm are denoted by $[u_{d,BC-MPC}, v_{d,BC-MPC}, r_{d,BC-MPC}]^\top$, and the velocities of the ferry by $[u, v, r]^\top$.

## 4.6 Comparing COLAV architectures

The water elevator scenarios 1-3 are now simulated with all three architectures, such that the hybrid COLAV system can be compared with the stand-alone systems of the MP-VP and the BC-MPC algorithm.

### 4.6.1 Scenario 1-2: water elevator with constant obstacle behavior

**Scenario 1: CP-O-CS**

This section presents the result of Scenario 1 with the three different COLAV architectures, and figures 4.10-4.11 shows North-East plots and velocities from the simulations. As expected, the hybrid COLAV system and the MP-VP COLAV system have similar trajectories, due to the constant speed and course of the moving obstacles.

The BC-MPC COLAV system is not restricted to the predefined paths and chooses a different trajectory. The ferry makes a starboard maneuver instead of slowing down and maneuvering to port after the obstacle crossing from port has passed. With this starboard maneuver, the ferry crosses ahead of the obstacle, which should be avoided according to the COLREGs Rule 8. A port maneuver before the obstacle has passed must, however, be avoided according to the COLREGs Rule 17, and a starboard maneuver is hence the best course maneuver in the situation.

The BC-MPC COLAV system reaches the goal about $40s$ before the hybrid COLAV system and the MP-VP COLAV system. The MP-VP aim to travel at desired transit-speed, whereas the BC-MPC COLAV system speeds up if it is lagging behind the reference.

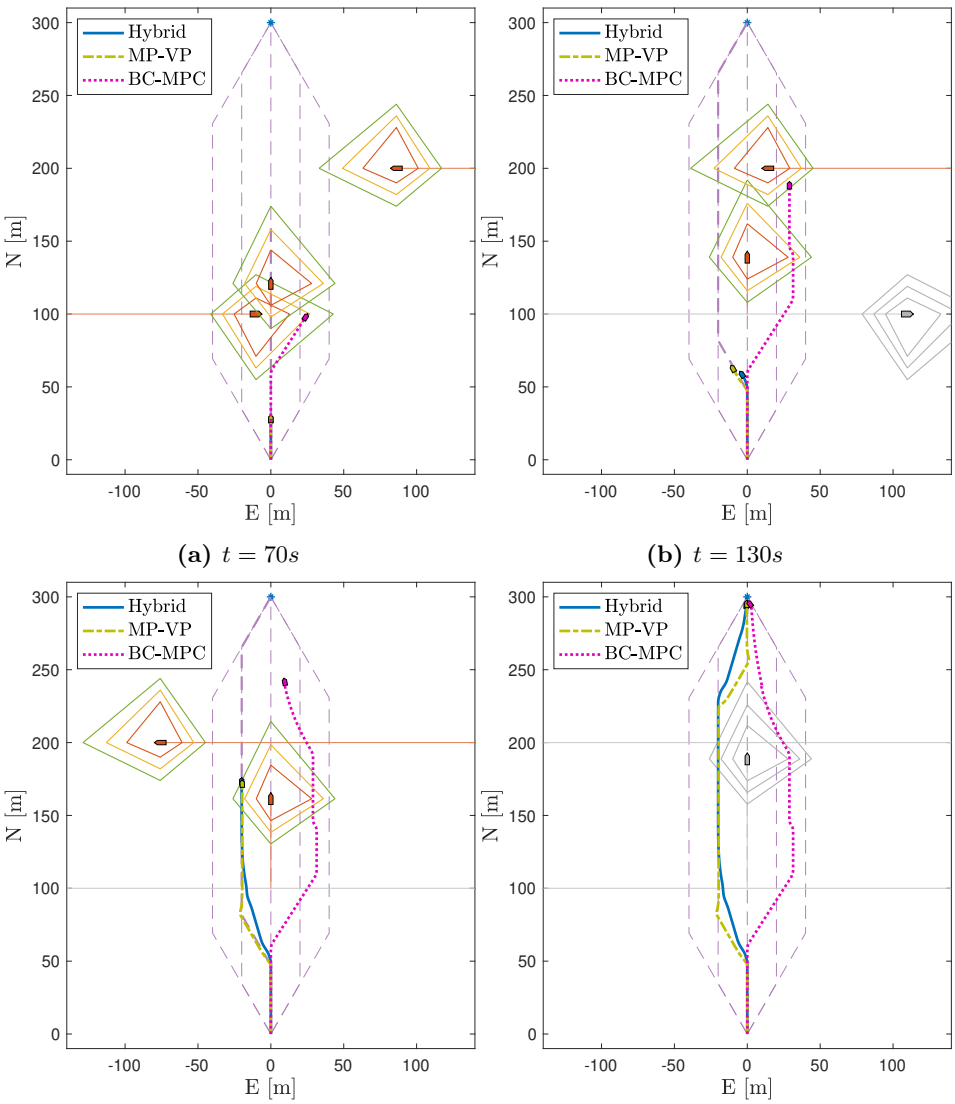
Looking at the accelerations, Figure 4.12 shows that the MP-VP algorithm has the highest peaks of linear acceleration. This is, however, dependent on the reference filter, which is tuned to have a fast response, as a tight tracking of the trajectory is essential. Some sway motion is experienced by the MP-VP COLAV system when course-maneuvers are executed, much due to the same reason.

The obstacle regions of the BC-MPC algorithm are tuned to fit inside the obstacle regions of MP-VP. As they have different shapes, where the regions of the BC-MPC algorithm cuts the corners, the BC-MPC algorithm can get closer to the obstacle than MP-VP without beeing penalized. As seen in Figure 4.13a, the BC-MPC COLAV system passes closer to the obstacles, and has the shortest MDO.

The BC-MPC COLAV system achieves the significantly highest values for IASR, IAYR, and IW as seen in Figure 4.13b-4.13c.

Table 4.5 summarizes the results, and the BC-MPC COLAV system finish the transit significantly faster than the MP-VP COLAV system and the hybrid COLAV

system. However, the BC-MPC COLAV system performs worst for all performance metrics except for the TT. The hybrid COLAV system performs best for most of the metrics, but it is a close race with the MP-VP for TT, IAYR, IW, and MDO.




**(a)** $t = 70s$ **(b)** $t = 130s$

**(c)** $t = 205s$ **(d)** $t = 296s$

**Figure 4.10:** Scenario 1: North-East plots of the transit with the traffic picture CP-O-CS and constant obstacle behavior. The plots shows the results using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm. The obstacles considered by the hybrid COLAV system at the current time are shown in colors, and the obstacles that are not considered in grey.

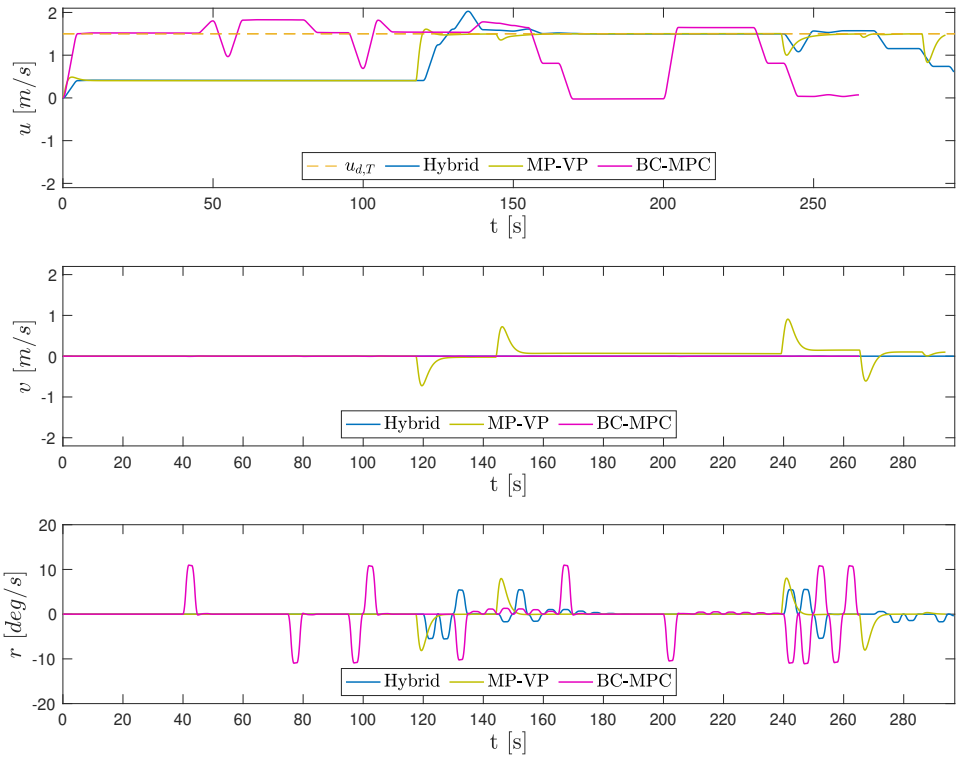



**Figure 4.11:** Scenario 1: velocity plots of the transit with the traffic picture CP-O-CS and constant behavior of the moving obstacles. The plot shows the velocities using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm.

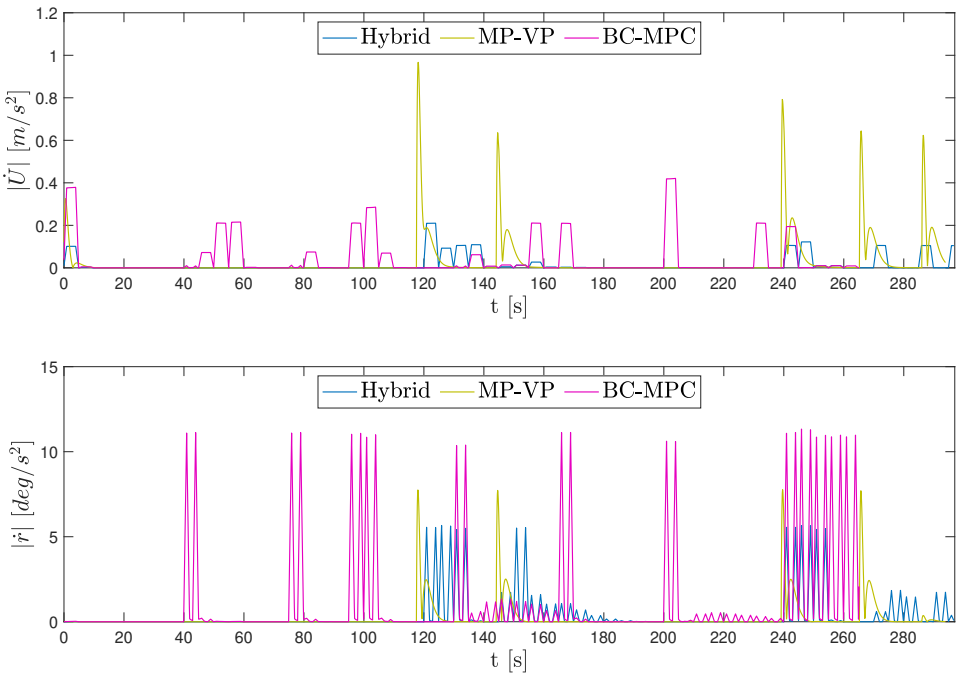



**Figure 4.12:** Scenario 1: absolute linear and angular accelerations of the transit with the traffic picture CP-O-CS and constant obstacle behavior. The plot shows the accelerations using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm.

**Table 4.5:** Performance metrics for Scenario 1. The table shows the travel time (TT), the minimum distance to obstacle (MDO), the integral of absolute speed rate (IASR), the integral of absolute yaw rate (IAYR), and the integral of power consumption (IW). The best values is highlighted in green, and the worst in red.

| COLAV system | TT [s] | MDO [m] | IASR | IAYR | IW [kJ/s] |
|---|---|---|---|---|---|
| Hybrid | 296 | 20.0 | 4.70 | 2.64 | 86.7 |
| MP-VP | 294 | 19.6 | 9.04 | 2.00 | 92.1 |
| BC-MPC | 265 | 18.8 | 11.66 | 7.21 | 125.2 |

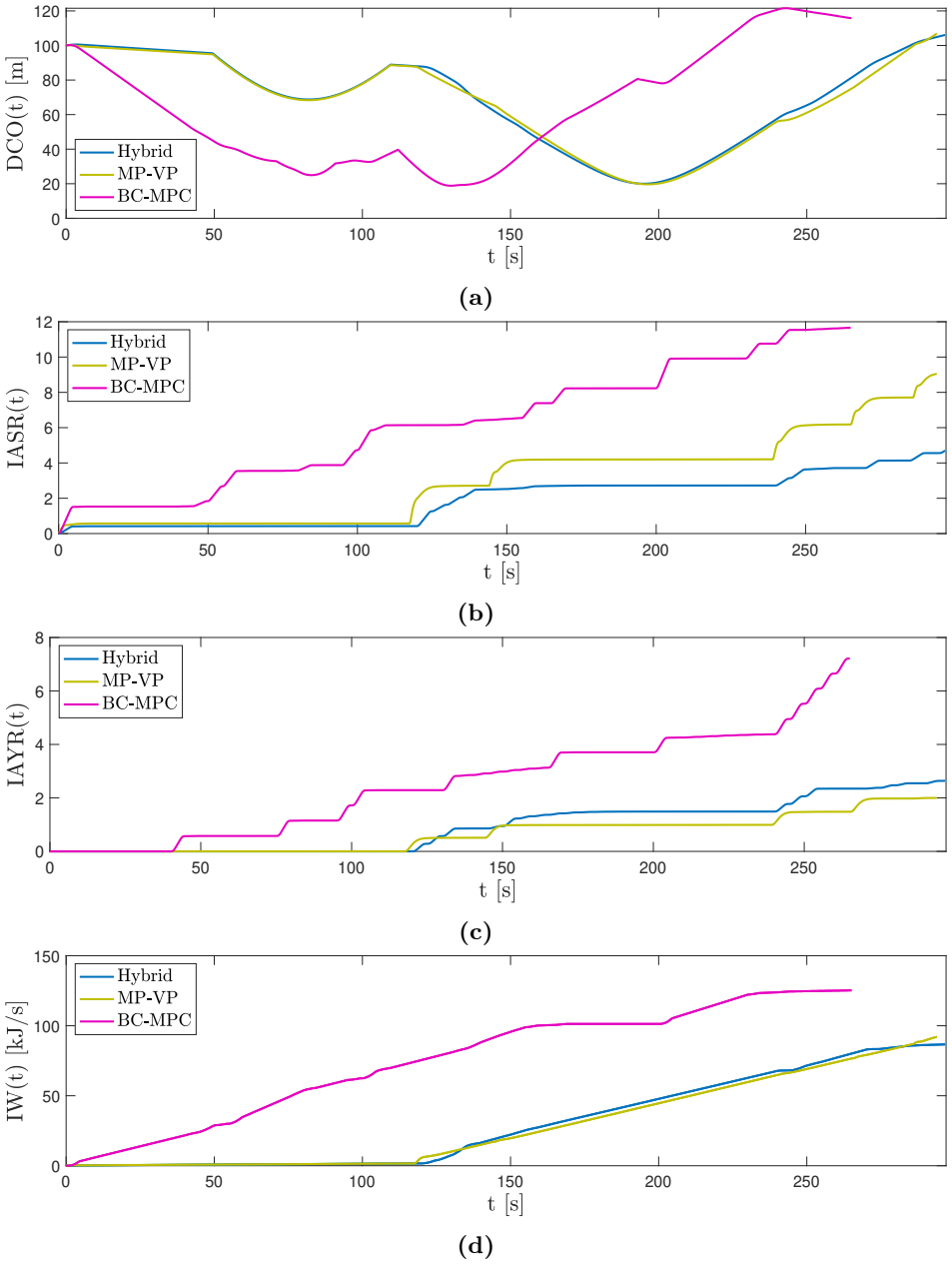


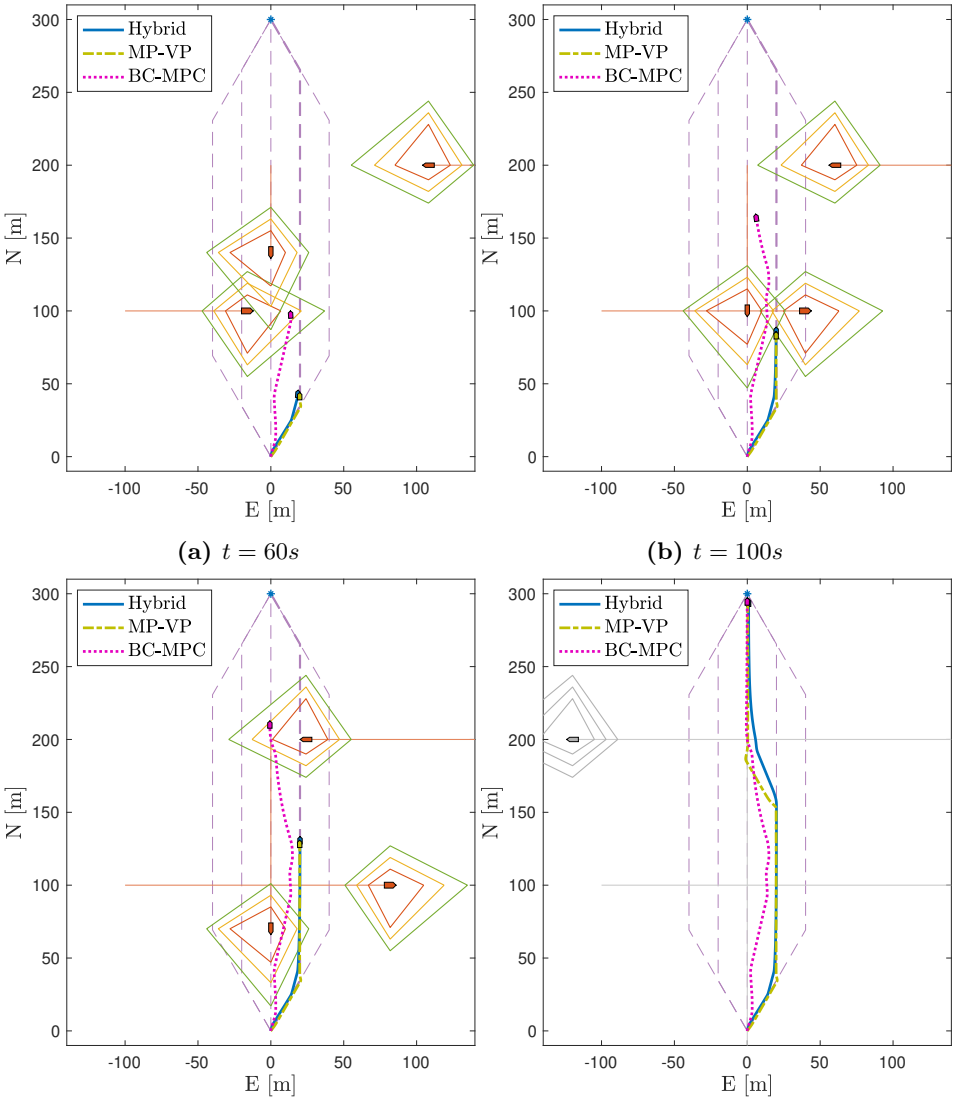

**Figure 4.13:** Scenario 1: the performance metrics for the three COLAV systems. The figure shows (a) the distance to the closest obstacle (DCO), (b) the integral of absolute speed rate (IASR), (c) the integral of absolute yaw rate (IAYR), and (d) the integral of power consumption (IW)).

**Scenario 2: CP-HO-CS**

In this scenario, the three architectures are compared for Scenario 2. North-East plots of the transit and the velocities are shown in Figures 4.14-4.15.

Similarly to the results of Scenario 1, the BC-MPC algorithm cuts corners, and reaches the destination first, at the cost of passing closer to the obstacles, executing more maneuvers, and higher power consumption, as seen in figures 4.17a-4.17d. Furthermore, the MP-VP has the highest absolute accelerations, as seen in Figure 4.16.

Looking at the performance metrics in Table 4.6, the hybrid COLAV system performs best for all performance metrics except for the TT, with the MP-VP following close behind.




**(a)** $t = 60s$ **(b)** $t = 100s$

**(c)** $t = 130s$ **(d)** $t = 250s$

**Figure 4.14:** Scenario 2: North-East plots of the transit with the traffic picture CP-HO-CS and constant behavior of the moving obstacles. The plots show the results using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm. The obstacles considered by the hybrid COLAV system at the current time are shown in colors, and the obstacles that are not considered in grey.

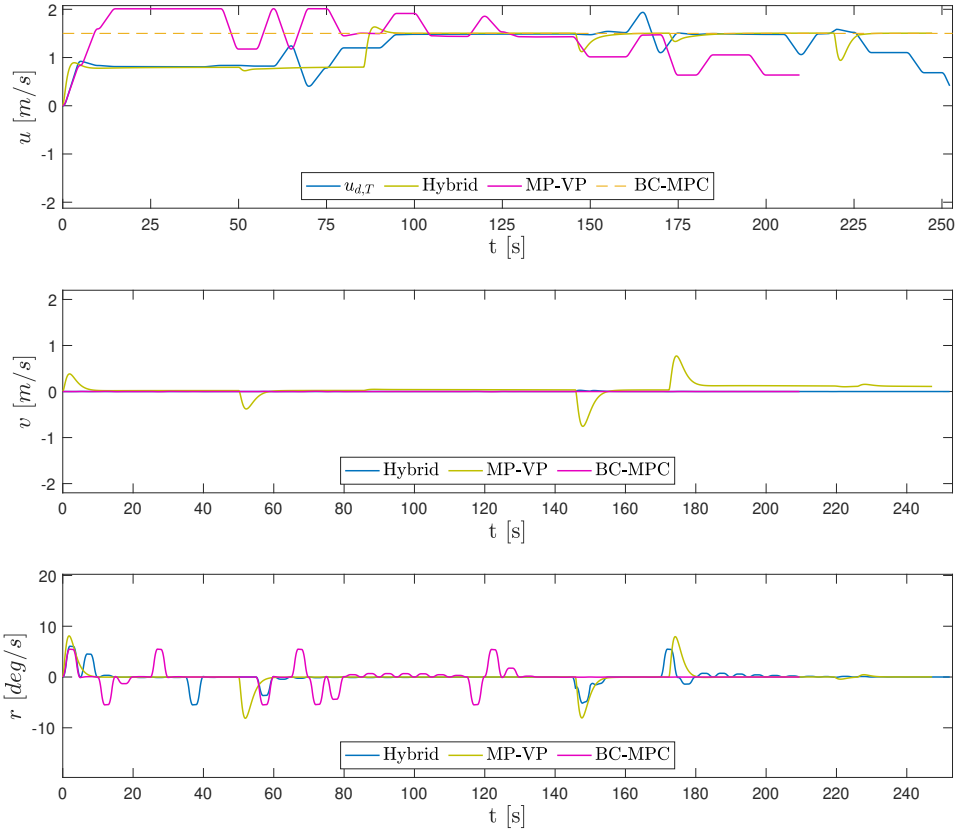



**Figure 4.15:** Scenario 2: velocity plots of the transit with the traffic picture CP-HO-CS and constant behavior of the moving obstacles. The plots show the velocities using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm.

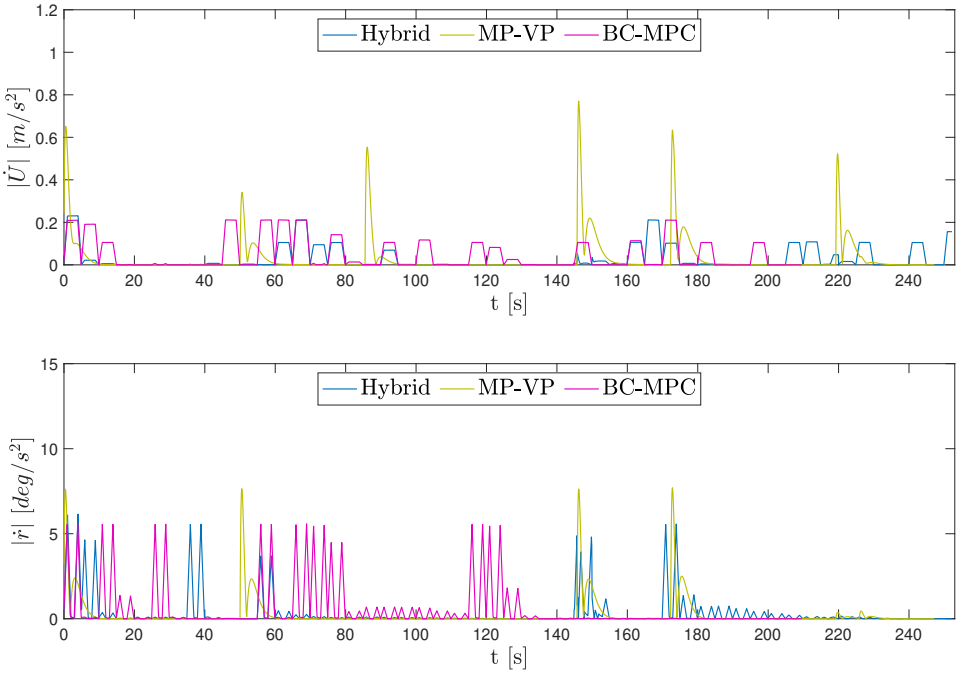



**Figure 4.16:** Scenario 2: absolute linear and angular accelerations of the transit with the traffic picture CP-HO-CS and constant behavior of the moving obstacles. The plots show the accelerations using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm.

**Table 4.6:** Performance metrics for Scenario 2. The table shows the travel time (TT), the minimum distance to obstacle (MDO), the integral of absolute speed rate (IASR), the integral of absolute yaw rate (IAYR), and the integral of power consumption (IW). The best values is highlighted in green, and the worst in red.

| COLAV system | TT [s] | MDO [m] | IASR | IAYR | IW [kJ/s] |
|---|---|---|---|---|---|
| Hybrid | 252 | 19.9 | 7.48 | 2.06 | 85.3 |
| MP-VP | 247 | 19.8 | 7.69 | 2.10 | 90.4 |
| BC-MPC | 209 | 14.6 | 10.39 | 2.89 | 123.0 |

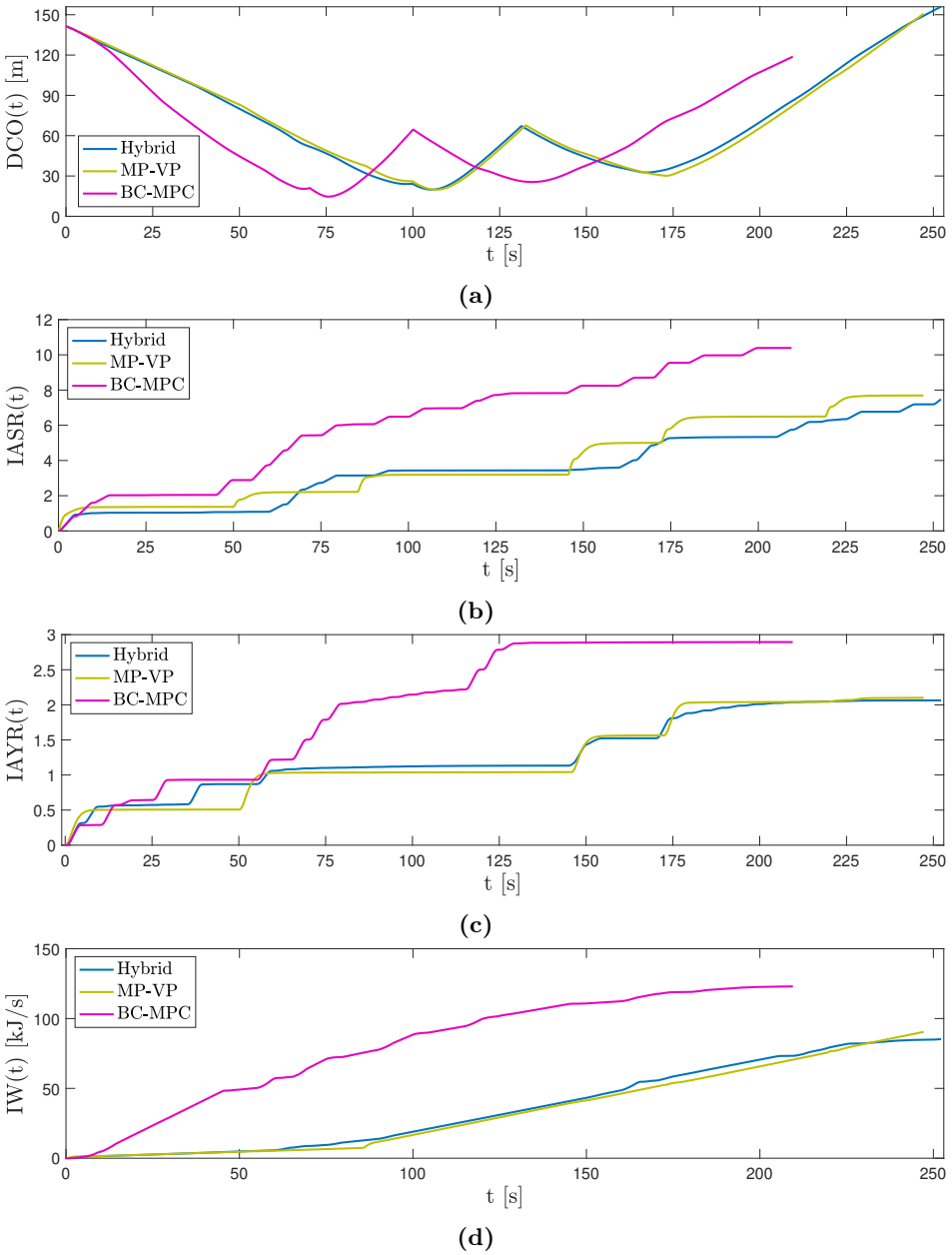


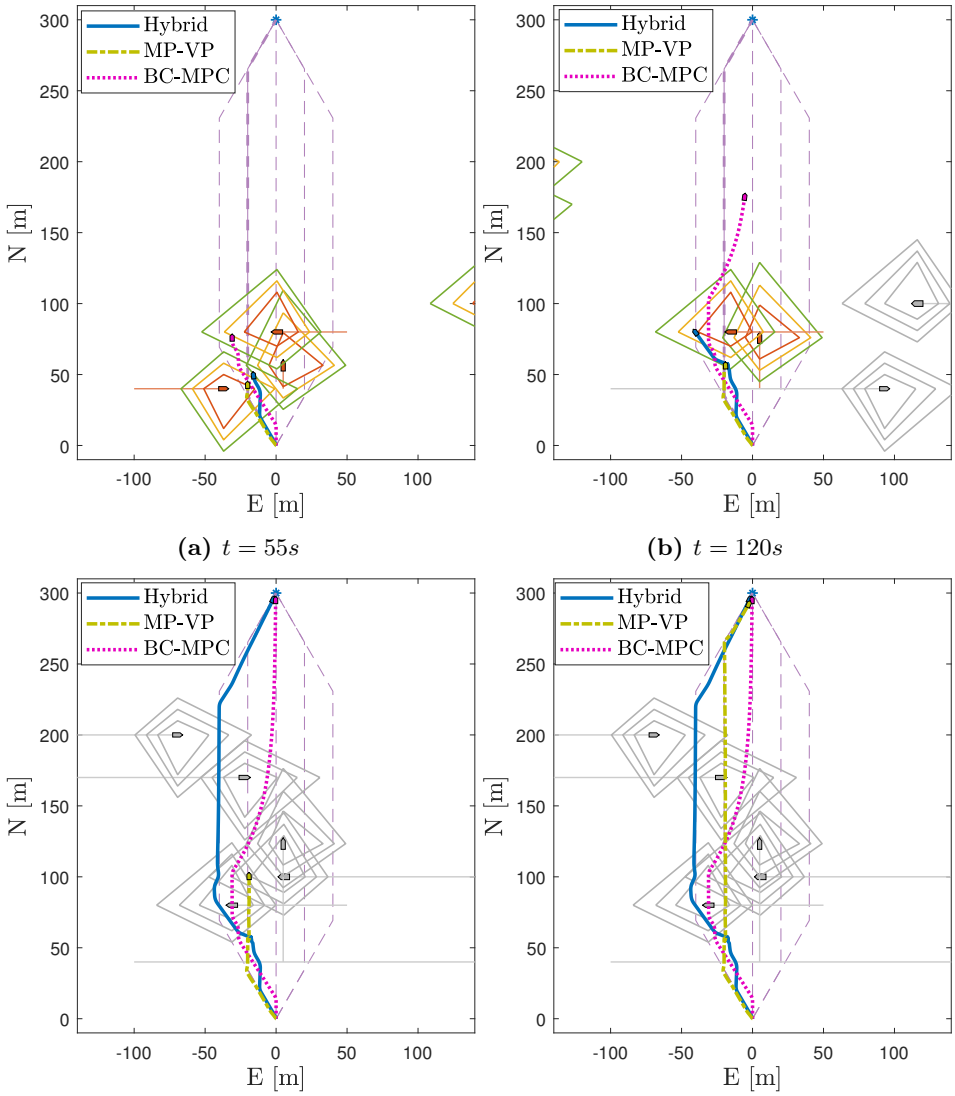

**Figure 4.17:** Scenario 2: the performance metrics for the three COLAV systems. The figure shows (a) the distance to the closest obstacle (DCO), (b) the integral of absolute speed rate (IASR), (c) the integral of absolute yaw rate (IAYR), and (d) the integral of power consumption (IW)).

### 4.6.2 Scenario 3: water elevator with speed-maneuvering obstacles

In this scenario, two of the crossing obstacles make speed-maneuvers, and the three COLAV systems choose different trajectories, as seen in the North-East and velocity plots in figures 4.18-4.19. The hybrid COLAV system and the MP-VP COLAV system follows the same trajectory initially. However, when the obstacle crossing from port at $N = 40m$, increases its speed, the MP-VP COLAV system continues along the predefined path, and pass the crossing obstacle at a closer distance than the hybrid COLAV system, as seen in Figure 4.18a. The MP-VP COLAV system continues along $path2$, and waits for a long time at the slowly moving obstacle, as all paths are blocked. The BC-MPC COLAV system achieve COLAV through course maneuvers, and crosses in front of all the crossing obstacles without slowing down.

Figure 4.20 shows the absolute linear and angular accelerations. Ones again, the peaks are within what is considered acceptable. The absolute linear acceleration are highest for the MP-VP COLAV system, and the absolute angular acceleration are highest for the hybrid COLAV system and the BC-MPC COLAV system.

Looking at the performance metrics in figures 4.21a-4.21d, the hybrid COLAV system saves the ferry from getting stuck, and achieves a shorter TT than the MP-VP COLAV system. Combining the speed-maneuvers of the MP-VP and the course-maneuvers of the BC-MPC algorithm, the hybrid COLAV system maneuvers significantly more than the stand-alone systems, achieving the highest values for IASR and IAYR. However, the IASR increases significantly at the end of the transit for both the hybrid and the BC-MPC COLAV systems, which is not directly related to COLAV maneuvers, but caused by the sometimes oscillating behavior by the BC-MPC algorithm when approaching a speed maneuver of the reference trajectory. The power consumption is also highest for the hybrid COLAV system, with the BC-MPC COLAV system following close behind. The MP-VP COLAV system has the lowest power consumption, at the cost of the longest TT. Finally, the MP-VP COLAV system has the shortest DCO, as all paths are blocked and there is nowhere to escape when the crossing obstacle increases its speed.




**(a)** $t = 55s$ **(b)** $t = 120s$

**(c)** $t = 410s$ **(d)** $t = 1543s$

**Figure 4.18:** Scenario 3: North-East plots of the transit with the high-traffic picture and speed-maneuvering obstacles. The plots show the results using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm. The obstacles considered by the hybrid COLAV system at the current time are shown in colors, and the obstacles that are not considered in grey.

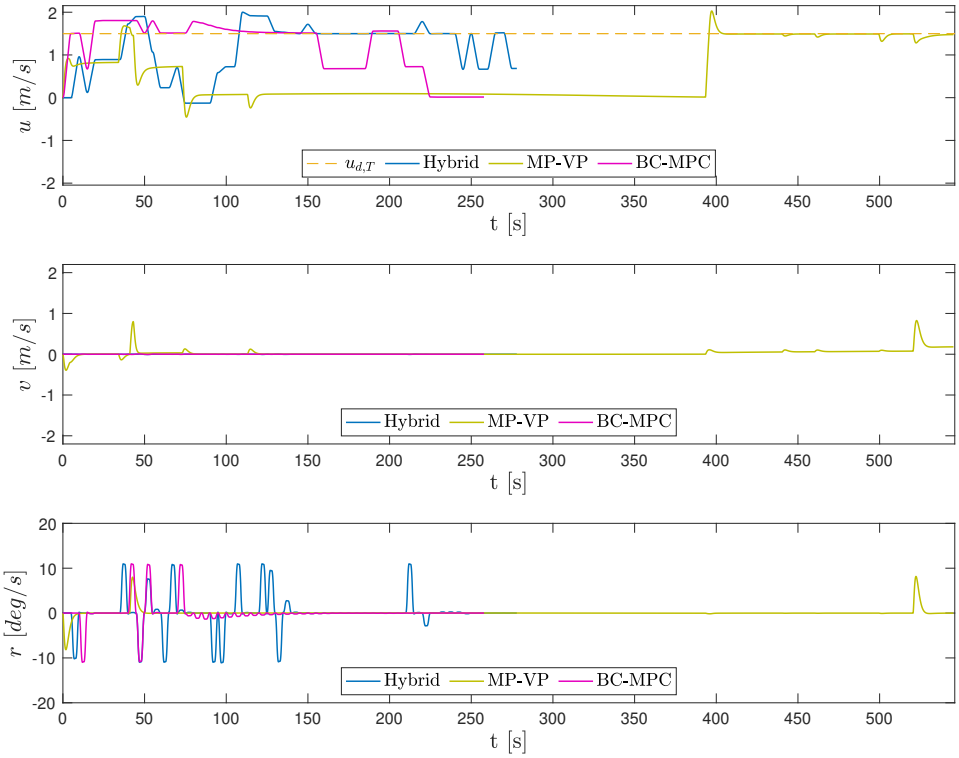



**Figure 4.19:** Scenario 3: velocity plots of the transit with the high-traffic picture and speed-maneuvering obstacles. The plots show the velocities using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm.

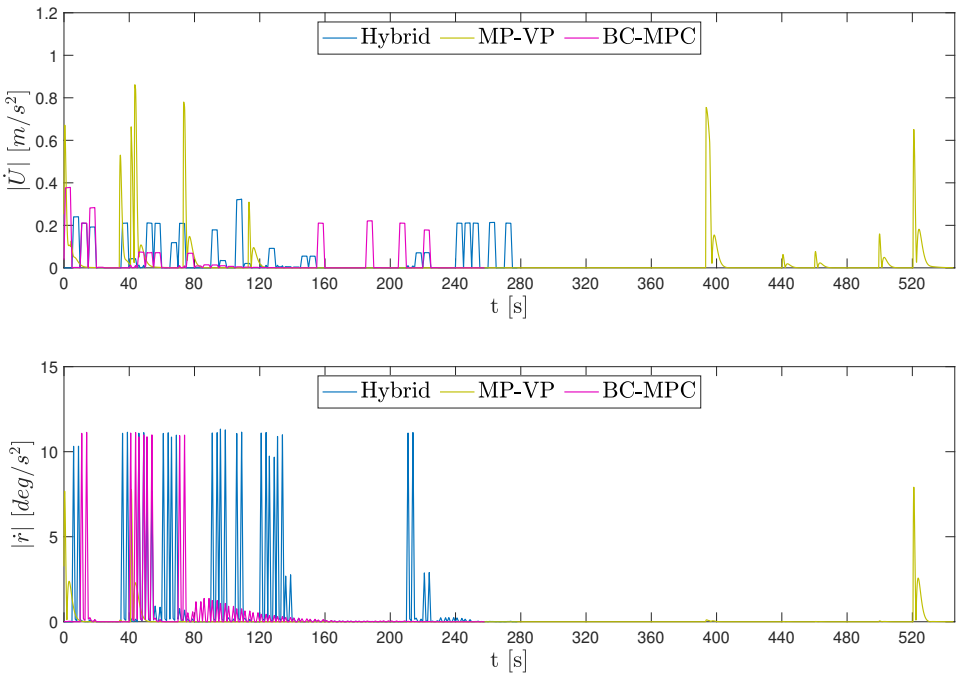



**Figure 4.20:** Scenario 3: absolute linear and angular accelerations of the transit with the high-traffic picture and speed-maneuvering obstacles. The plots show the accelerations using the hybrid COLAV system and the stand-alone systems with the MP-VP and the BC-MPC algorithm.

**Table 4.7:** Performance metrics for Scenario 3. The table shows the travel time (TT), the minimum distance to obstacle (MDO), the integral of absolute speed rate (IASR), the integral of absolute yaw rate (IAYR), and the integral of power consumption (IW). The best values is highlighted in green, and the worst in red.

| **COLAV system** | **TT [s]** | **MDO [m]** | **IASR** | **IAYR** | **IW [kJ/s]** |
|---|---|---|---|---|---|
| Hybrid | 278 | 13.7 | 14.56 | 7.60 | 121.2 |
| MP-VP | 543 | 8.2 | 12.50 | 1.65 | 94.9 |
| BC-MPC | 255 | 22.66 | 8.25 | 3.40 | 115.4 |

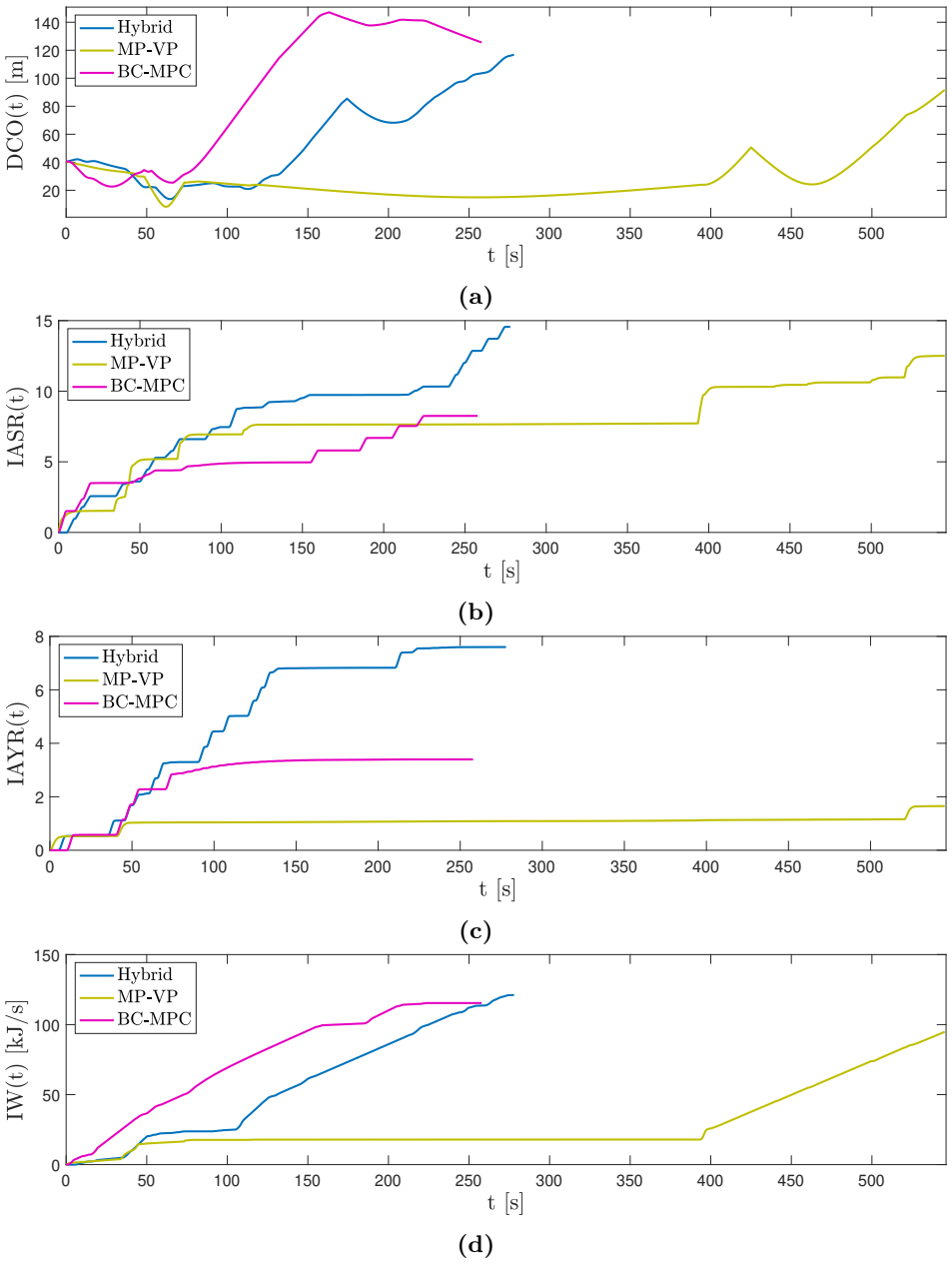



**Figure 4.21:** Scenario 3: the performance metrics for the three COLAV systems. The figure shows (a) the distance to the closest obstacle (DCO), (b) the integral of absolute speed rate (IASR), (c) the integral of absolute yaw rate (IAYR), and (d) the integral of power consumption (IW)).

## 4.7 Summary of results

The hybrid COLAV system avoids collisions with moving obstacles successfully. For scenarios where the obstacles keep constant speed and course, the ferry follows the trajectory generated by the MP-VP. For the scenarios with speed-maneuvering obstacles, the trajectory is modified by the BC-MPC algorithm to avoid collisions or to get stuck waiting for slowly moving obstacles. When possible, the ferry is guided back on a trajectory generated by the MP-VP.

The piecewise linear paths provide trajectories with a few, readably apparent course maneuvers, and the hybrid COLAV system maneuvers in compliance with the COLREGs rules 8 and 13-17.

The results of the three COLAV systems from scenarios 1-3 are summarized in Table 4.8, with the best and worst performance for each scenario marked with green and red, respectively. The results are sorted by the COLAV system, to show the strengths and weaknesses of each COLAV system better.

For constant behavior of the moving obstacles, the hybrid COLAV system has the best performance, with slightly better values than the MP-VP COLAV system for most of the metrics. Comparing the hybrid COLAV system to the BC-MPC COLAV system, the total manuvering and power consumption are significantly reduced with lower values for IASR, IAYR, and IW.

For Scenario 3 with the speed-maneuvering obstacles, the hybrid COLAV system has a significantly reduced travel time compared to the MP-VP COLAV system, at the cost of more maneuvers and higher power consumption. The BC-MPC COLAV system acheives COLAV mainly by course-maneuvering, resulting in less maneuvers in total.

**Table 4.8:** Summary of results, sorted by the COLAV system. The table shows the travel time (TT), the minimum distance to obstacle (MDO), the integral of absolute speed rate (IASR), the integral of absolute yaw rate (IAYR), and the integral of power consumption (IW). The best values for each scenario is highlighted in green, and the worst in red.

| Scenario | COLAV system | TT [s] | MDO [m] | IASR | IAYR | IW [kJ/s] |
|---|---|---|---|---|---|---|
| Sc. 1 | Hybrid | 296 | 20.0 | 4.70 | 2.64 | 86.7 |
| Sc. 2 | Hybrid | 252 | 19.9 | 7.48 | 2.06 | 85.3 |
| Sc. 3 | Hybrid | 278 | 13.7 | 14.56 | 7.60 | 121.2 |
| Sc. 1 | MP-VP | 294 | 19.6 | 9.04 | 2.00 | 92.1 |
| Sc. 2 | MP-VP | 247 | 19.8 | 7.69 | 2.10 | 90.4 |
| Sc. 3 | MP-VP | 543 | 8.2 | 12.50 | 1.65 | 94.9 |
| Sc. 1 | BC-MPC | 265 | 18.8 | 11.66 | 7.21 | 125.2 |
| Sc. 2 | BC-MPC | 209 | 14.6 | 10.39 | 2.89 | 123.0 |
| Sc. 3 | BC-MPC | 255 | 22.66 | 8.25 | 3.40 | 115.4 |

## 4.8 Discussion

The hybrid COLAV system avoids collisions successfully in both shorter water elevator transits, and at water bus scenarios of longer transits. However, combining different algorithms and desirable behaviors into one system raises the challenge of achieving optimal behavior in all situations. The hybrid COLAV system would most likely benefit from prioritizing either speed or course maneuvers depending on the situation. A more consistent and predictable behavior will improve the passenger comfort and the compliance with COLREGs, and lower the power consumption. This should be considered in the development of the algorithms, the supervisor, and the representation of the surroundings.

When using the trajectories generated by the MP-VP as a reference for the BC-MPC algorithm, there is a risk that the reference trajectory gets outdated when the moving obstacles maneuvers. If the mid-layer is locked, and the reference trajectory can not be replanned, the outdated reference trajectory can create more damage than help. A solution can be to reparametrize the current reference trajectory with constant transit speed if a new trajectory is requested while the mid-layer is locked. This allows the BC-MPC algorithm to operate without considering the velocity plan from the MP-VP, and can reduce conflicts between the algorithms and contribute to more predictable behavior.

The hybrid COLAV system does not track the reference trajectory perfectly but cuts corners of the turns. Furthermore, oscillations are experienced when approaching a change in the speed reference. This is not critical as the hybrid COLAV system is not dependent on tracking the reference trajectory perfectly to ensure COLAV. However, clear maneuvers and fewer oscillations are desirable considering the COLREGs Rule 8, passenger comfort, and power consumption. The performance of the BC-MPC algorithm tracking trajectories with changes in speed and/or course can be improved by making sure that the best possible trajectory is available, and that the best available trajectory is selected. To ensure the availability of the best possible trajectory, the calculation of the desired acceleration can be improved. Furthermore, for generating trajectories that match perfectly with the reference, a dynamic prediction horizon can be introduced. This will, however, increase the complexity of the BC-MPC algorithm. Considering the objective function, cross-track and along-track errors can be included in the trajectory alignment term.

As for the BC-MPC algorithm, the output of the MP-VP algorithm is dependent on which trajectories that are considered, and which trajectory that is chosen by the objective function. Hence, the behavior of both algorithms is strongly dependent on the terms and weights of the objective functions, and the choice of parameters should be considered carefully.

The BC-MPC algorithm allow for planning trajectories that branch at another angle than $\alpha_b$ or goes outside the predefined outer paths, which can be a necessity in critical situations. Another approach to branch at another branching angle is to implement a dynamic branching angle in the mid-layer, as suggested in [4]. When

the ferry goes outside the predefined outer paths, it leaves the area that is assumed free of static obstacles. Therefore, the operation area of the ferry should be limited to areas where there are no static obstacles, or the term considering static obstacle avoidance introduced in [3] should be added to the objective function of the BC-MPC algorithm.

Considering the COLREGs, the modified representation of the moving obstacles for the MP-VP results in a behavior of the hybrid COLAV system that mostly complies with the COLREGs rules 8 and 13-17. An improved obstacle monitoring and representation of moving obstacles could probably improve the COLREGs compliance further.

The dynamic obstacle representation allows the ferry to pass closer to the obstacles in confined areas, which is reasonable as the obstacles are expected to keep a slow speed. This lowers the number of predefined paths that can be blocked by one obstacle. However, to better reflect the characteristics of each obstacle, considerations as the speed and course of the moving obstacle could be included in the representation of moving obstacles. One option is not to add the COLREGs gain to the starboard axis of the moving obstacle representation for moving obstacles with a course aligned with the ferry. Then, moving obstacles being overtaken will block less of the predefined paths, and avoid motivating the ferry to take a port maneuver when overtaking the obstacle.

Considering passenger comfort, the accelerations of the hybrid COLAV system are within limits. Besides, the accelerations can be saturated in the main search space if necessary, leaving the maneuvers of maximum accelerations to the emergency search space.

Quantitative measurement of the computationally cost of the algorithms are not included, but it is evident through the simulations that the BC-MPC algorithm is more complex and requires a longer computationally time than the MP-VP. The supervisor is simple, and the hybrid COLAV system's run time depends mainly on the BC-MPC algorithm. With the emergency search space, a third DOF is included while still keeping the size of the trajectory search space reasonable. Furthermore, the object monitoring keeps the moving obstacles that are not relevant out of consideration by the COLAV algorithms. Hence, the hybrid COLAV system is still within reasonable complexity and computationally time.

# Chapter 5

# Conclusion and future work

A hybrid collision avoidance (COLAV) system for autonomous passenger ferries combining the multiple-path velocity planner (MP-VP) and the branching-course model predictive control (BC-MPC) algorithm is developed and evaluated through simulations of both elevator and water bus shuttle scenarios with multiple obstacles. The COLAV system avoids collisions successfully, and is compliant with the international regulations for preventing collisions at sea (COLREGs) rules 8 and 13-17.

The hybrid COLAV system is compared with stand-alone COLAV systems of the MP-VP and the BC-MPC algorithm, and evaluated using quantitative performance metrics based on a set of identified desirable properties. In situations where the moving obstacles keep constant speed and course, the hybrid COLAV system follows the trajectories generated by the MP-VP, and achieves a more energy-efficient and consistent behavior compared to the BC-MPC COLAV system. For situations where the moving obstacles make maneuvers such that the reference trajectory is blocked, the BC-MPC algorithm ensures that the hybrid COLAV system avoids collisions and that the ferry gets stuck at the cost of more maneuvering and higher power consumption.

For further development of the hybrid COLAV system combining the MP-VP and the BC-MPC algortihm, the following is proposed:

- Further development of the hybrid COLAV system to enhance either stop-and-go approach or course maneuvers.
- Perform simulations of various scenarios with maneuvering obstacles and investigate the parameter sensitivity.
- Improve the BC-MPC algorithms tracking capabilities of reference trajectories with maneuvers that do not align with the maneuver intervals of the prediction horizon.
- If relevant, implement the consideration of static obstacles for the BC-MPC

algorithm.

- Further development of the moving obstacle representation to better reflect the properties of the moving obstacle and the surrounding environment.
- Integrate the hybrid COLAV system with the existing ROS-system on the milliAmpere test platform and perform full-scale tests to validate the simulation results.

# Bibliography


[1] A. Uttisrud, "Comparing collision avoidance methods for ASVs: BC-MPC versus velocity obstacles," project report, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2019.

[2] B.-O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "The branching-course model predictive control algorithm for maritime collision avoidance," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 2019.

[3] B.-O. H. Eriksen and M. Breivik, "Short-term ASV collision avoidance with static and moving obstacles," *Modeling, Identification and Control*, vol. 40, no. 3, pp. 177–187, 2019.

[4] E. H. Thyri, "A path-velocity decomposition approach to collision avoidance for autonomous passenger ferries," Master's thesis, Norwegian Univeristy of Science and Technology (NTNU), Trondheim, Norway, 2019.

[5] E. H. Thyri, M. Breivik, and A. Lekkas, "A path-velocity decomposition approach to collision avoidance for autonomous passenger ferries: Concepts and full-scale experiments," *Submitted to IFAC World Congress 2020*, 2020.

[6] A. A. Pedersen, "Optimization based system identification for the milliAmpere ferry," Master's thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2019.

[7] T. R. Torben, "Control allocation and observer design for autonomous ferries," Master's thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2019.

[8] United Nations, Department of Economic and Social Affairs, Population Division, "World urbanization prospects: The 2018 revision," 2019.

[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[10] 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)*, Buenos Aires, Argentina, Sep. 2016.

[11] B.-O. H. Eriksen, E. F. Wilthil, A. L. Flaten, E. F. Brekke, and M. Breivik, "Radar-based maritime collision avoidance using dynamic window," in *2018 IEEE Aerospace Conference*, Big Sky, Montanta, USA, Mar. 2018.

[12] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998.

[13] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime navigation with COLREGS using velocity obstacles," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, Sep. 2011.

[14] ——, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, Jan. 2014.

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*, Springer New York, 1986, pp. 396–404.

[16] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Sacramento, USA.

[17] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989. [Online]. Available: `http://dx.doi.org/10.1109/21.44033`.

[18] S.-M. Lee, K.-Y. Kwon, and J. Joh, "A fuzzy logic for autonomous navigation of marine vehicle satisfying COLREG guidelines," *International Journal of Control, Automation, and Systems*, vol. 2, Jun. 2004.

[19] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[20] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[21] Ø. A. G. Loe, "Collision avoidance for unmanned surface vehicles," Master's thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2008.

[22] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[23] B.-O. H. Eriksen and M. Breivik, "MPC-based mid-level collision avoidance for asvs using nonlinear programming," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Hawaii, USA: IEEE, Aug. 2017.

[24] B. O. H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, "Hybrid collision avoidance for ASVs compliant with COLREGs rules 8 and 13–17," *submitted to Frontiers in Robotics and AI*, 2019.

[25] The Maritime Executive. (Dec. 2018). Rolls-Royce and Wartsila in close race with autonomous ferries, [Online]. Available: `https://www.maritime-executive.com/article/rolls-royce-and-wartsila-in-close-race-with-autonomous-ferries` (visited on 12/14/2019).

[26] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, Sep. 1986.

[27] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Trondheim, Norway: John Wiley & Sons, Ltd, Apr. 2011.

[28] SNAME, "Nomenclature for treating the motion of a submerged body through a fluid," 1950.

[29] G. Bitar, B.-O. H. Eriksen, A. M. Lekkas, and M. Breivik, "Energy-optimized hybrid collision avoidance for ASVs," in *2019 18th European Control Conference (ECC)*, Jun. 2019.

[30] M. Breivik and T. I. Fossen, "Guidance laws for autonomous underwater vehicles," in *Underwater Vehicles*. InTech, Jan. 2009.

[31] International Maritime Organization, Ed. (). Convention on the international regulations for preventing collisions at sea, 1972 (COLREGs), [Online]. Available: `http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx` (visited on 06/18/2019).

[32] "COLREGS - International Regulations for Preventing Collisions at Sea," *Lloyd's Register Rulefinder, 2005. Version 9.4.*,

[33] Cambridge University Press, Ed. (). Cambridge online english dictionary, ferry, [Online]. Available: `https://dictionary.cambridge.org/dictionary/english/ferry` (visited on 11/20/2019).

[34] L. L. Hoberock, "A survey of longitudinal acceleration comfort studies in ground transportation vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 76–84, Jun. 1977.

Anette Uttisrud

Hybrid collision avoidance for autonomous passenger ferries

NTNU
Norwegian University of
Science and Technology