Modular Trajectory Planning for Autonomous Surface Vessels in Inland Waters Using Electronic Nautical Charts

THESIS

submitted in partial fulfilment of the
requirements for the degree of

MASTER OF SCIENCE

in

ROBOTICS

by

Jelle Vogel
born in Haarlem, the Netherlands
Student Number: 4459911

Autonomous Multi-Robots Lab
Department of Cognitive Robotics
Faculty ME, Delft University of Technology
Delft, the Netherlands
www.tudelft.nl/me

Damen Shipyards
Automation Team
Avelingen-West 20
Gorinchem, the Netherlands
www.damen.com

Abstract

Autonomous navigation on inland waterways is challenging: channel geometry can vary widely, traffic rules are qualitative, and vessels must interact safely with other traffic while operating near infrastructure. This thesis presents a modular two-stage planning framework for a high-speed passenger ferry (Damen Waterbus) operating between Rotterdam and Dordrecht. A global planner extracts routes from Electronic Navigational Charts (ENCs) and refines them to produce a starboard-biased global path. A local planner uses Biased Model Predictive Path Integral (BIASED-MPPI) with ancillary controllers for path-following, goal reaching, and cruise speed regulation. Its cost function adapts based on the traffic scenario. Together, the planners generate starboard-biased trajectories that respect vessel limits, traffic and waterway constraints.

Global paths are built using the waterway-axis object extracted from the ENCs, and are made starboard-biased by projecting to the Fairway. This path is then refined using simple windowed smoothing schemes (minimum, average, and a hybrid), applied to the projection distances. The best smoothing configurations are explored via a grid search over various parameters. The generated paths are evaluated by metrics that assess feasibility. Across diverse intersections, smoothing noticeably reduces irregularity and sharp turns. The hybrid variant offers the most consistent gains, though feasibility in difficult geometries remains the dominant failure mode.

The local planner runs in real time and handles three encounter types (head-on, crossing, overtaking) using a standardised COLREG state definition to trigger rule-consistent behaviour. Variants without learned priors or with reduced replanning frequency are compared to highlight the effect of priors and update rate on success, stability, and computation time. Human-likeness is assessed against the Triton dataset, consisting of Waterbus AIS data via dock-to-dock distance and travel-time benchmarks.

Overall, the study demonstrates that combining ENC-informed global planning with a COLREG-aware MPPI controller yields safe, efficient, and human-like dock-to-dock trajectories in structured inland waterways, while identifying where the proposed method still limits performance.

Thesis Committee:

Chair:	Dr. J. Alonso-Mora , Faculty ME, TU Delft
University supervisor:	Dr. J. Alonso-Mora , Faculty ME, TU Delft
Company supervisor:	Dr. K. van der El, Damen Shipyards
Committee Member:	Dr. L. Ferranti, Faculty ME, TU Delft

Contents

Contents 3

List of Figures 5

1 Introduction 1
- 1.1 Related Works . . . 2
- 1.2 Research Questions . . . 4

2 Background 7
- 2.1 Mission: Safe Dock-to-Dock Sailing . . . 8
- 2.2 High-level Dynamic Constraints . . . 9
- 2.3 Data . . . 10
- 2.4 System Overview and Constraints . . . 11

3 Global Path Planning 13
- 3.1 Route Planning . . . 13
- 3.2 Refining the Global Path . . . 15
- 3.3 Global Planner evaluation . . . 19
- 3.4 conclusion . . . 24

4 Local Path Planning 25
- 4.1 Local Planner Requirements . . . 26
- 4.2 Model Predictive Path Integral (MPPI) Control . . . 27
- 4.3 Vessel Dynamic Model . . . 28
- 4.4 Sailing Phase Switcher . . . 30
- 4.5 COLREG state-switcher . . . 31
- 4.6 Cost Function Formulation . . . 34
- 4.7 Biased-MPPI . . . 39
- 4.8 Leveraging Biased-MPPI for inland waterway local planning . . . 40
- 4.9 Summary and Method Overview . . . 44

5 Results 45
- 5.1 Dock-to-Dock scenario analysis without traffic encounters . . . 46
- 5.2 Traffic Encounters . . . 50
- 5.3 Dock-to-Dock with Traffic Handling . . . 53
- 5.4 Details of near-dock manoeuvres . . . 58
- 5.5 Discussion . . . 60

6 Conclusions and Future Work 61

6.1 Conclusions 61
6.2 Recommendations 63

Bibliography 65

A Glossary 71

B Preconditions 73

C Parameters of Ancillary controllers 75

D MPPI Cost Function Configuration Parameters 77

E Detailed Test results 79

F Full grid Search Results 83

G Scenarios 85
G.1 Examples of different planning settings 87

List of Figures

2.1 Overview of all docks used when evaluating autonomous dock-to-dock voyages. . . . 7
2.2 Schematic overview of Sailing phases. Blue = Undocking, White = Sailing, Yellow = Approach, Orange = Docking, Red = Berthing 8
2.3 Three traffic scenarios considered in this thesis—overtaking, head-on, and crossing. Own Ship (green), Target Ship (red). 9
2.4 Side view of the Damen Waterbus 9
2.5 IENC of the inland waters at Gorinchem. 10
2.6 All extractable AIS routes from the Triton dataset between stations Merwedekade and Westeind. 11
2.7 High-level system overview and scope of the work (green) and the corresponding inputs and outputs. 12

3.1 Two combined IENC of the inland waters of Rotterdam. 14
3.2 Graph extracted from the waterway axis structure. 14
3.3 Global path found by mission planner using the waterway axis object from the merged ENC data. 14
3.4 Overview of the global path planning system architecture. 15
3.5 Interpolated path projected onto the Fairway. 16
3.6 projected path in Dordrecht 16
3.7 Apply the minimum windowsize method. 17
3.8 Apply the average windowsize smoothing, clipping points outside the Fairway back to their original position. 17
3.9 Apply the average windowsize method for multiple iterations. 18
3.10 Hybrid method which first applies the minimum windowed smoothing and then applies multiple iterations of the average window smoothing method. 18
3.11 Illustration of various performance metrics used to evaluate the global planner. . . . 19
3.12 Projected path with an infeasible line segment, intersecting the Fairway. 20
3.13 maximum Step Size Projection Ratio with varying step sizes for all scenarios. 21
3.14 Commonly caused infeasible paths due to a width change in the lane transition. . . . 23
3.15 Example of a sharp left in the waterway axis leading to crossing projections. 23
3.16 Issues with disjointed Fairway objects with ENCs produced by the Port of Rotterdam. 23

4.1 Overview of the local path planning system architecture 25
4.2 Overview of the various sailing phase switches and COLREG state switches. 31
4.3 The COLREG classification model mainly utilises angles α and β, defined respectively as the angle between the Target Ship's longitudinal axis and the LOS and the Own Ship's longitudinal axis and the LOS. 32

4.4 Hagen's COLREG classification model from the Own Ship's viewpoint. The mappings $\alpha_{360} : [-180°, 180°) \to [0°, 360°)$ and $\beta_{180} : [0°, 360°) \to [-180°, 180°)$ are used to display the symmetry in the situation classification model. [11] 32

4.5 High-level State Machine Diagram for COLREG state switcher, showcasing how the COLREG memory is applied. 33

4.6 Illustration of the Velocity Obstacle VO_B for vessel A avoiding vessel B by Fiorini et al. [9]. The cone spanned by δ_r and δ_f shows all relative velocity directions that would intersect with obstacle B. 37

4.7 CR-GW: Way Heading setpoint set to align with the point two shiplengths behind the TS' stern. 38

4.8 CR-GW: Once positioned behind the TS, change OS heading setpoint perpendicular to current TS heading. 38

4.9 CR-GW: Postcondition with a little more text to yap. 38

4.10 OT-GW: Heading setpoint to align with the point one shiplength to the portside of TS. 39

4.11 OT-GW: Setpoint when more than one shiplength away from TS portside. 39

4.12 OT-GW: Postcondition. 39

4.13 HO: Heading setpoint. 39

4.14 HO: Postcondition. 39

4.15 Overview of local path planning architecture distinguishing three cycle depths: The system cycle, MPPI cycle and rollout cycle. 44

5.1 Trajectory of autonomous vessel compared to AIS logged voyages from Merwedekade to Westeind. 47

5.2 Trajectory of autonomous vessel and the corresponding global path from De Schans to Rijsdijk. 47

5.3 MPPI-Inland-10 overshoots the dock multiple times before berthing. 49

5.4 Head-on Scenario overview. 50

5.5 Crossing Give-way Scenario overview. 50

5.6 Head-on Scenario 50

5.7 Distance to Closest Point of Approach (CPA) over time for the head-on, crossing give-way, and overtaking scenarios. 51

5.8 Current velocities and heading over time for the head-on, crossing give-way, and overtaking scenarios. 52

5.9 Overview of Merwedekade-Westeind with overtaking scenario. 53

5.10 Velocity profiles and Heading of Merwedekade-Westeind with overtaking scenario. Shaded regions correspond to different sailing phases. The Hatched area corresponds to a traffic encounter. 53

5.11 Distance between Own Ship and Target Ship the Closest Point of Approach, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario. . 54

5.12 Cross Track Error, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario. Shaded regious correspond to different sailing phases. The Hatched area corresponds to a traffic encounter. 55

5.13 Overview of Merwedekade-Westeind with overtaking scenario (Priors disabled). . . . 55

5.14 Velocity profiles and Heading of Merwedekade-Westeind with overtaking scenario (Priors disabled). Shaded regions correspond to different sailing phases. The Hatched area corresponds to a traffic encounter. 56

5.15 Distance between Own Ship and Target Ship the Closest Point of Approach, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario (Priors disabled). 56

5.16 Overview of Merwedekade-Westeind with overtaking scenario of MPPI-Inland-10. . . 57

5.17 Velocity profiles and Heading of Merwedekade-Westeind with overtaking scenario of MPPI-Inland-10. Shaded regions correspond to different sailing phases. The Hatched area corresponds to a traffic encounter. 57
5.18 Distance between Own Ship and Target Ship the Closest Point of Approach, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario of MPPI-Inland-10. 58
5.19 Undocking overview. 58
5.20 Docking overview. 58
5.21 Undocking velocity profiles and heading. 59
5.22 Undocking velocity profiles and heading in terms of distance to dock. 59
5.23 Docking velocity profiles and heading. 59
5.24 Docking velocity profiles and heading in terms of distance to dock. 59

G.1 Dordrecht and Dordrecht Flipped projected paths 85
G.2 Hardinxveld Projected Paths 85
G.3 Nieuwe Maas Projected Paths 86
G.4 Rotterdam Projected Paths 86
G.5 Sliedrecht Intersection Projected Paths 86
G.6 Sophiapolder scenarios 86
G.7 Sliedrecht Left, minimum window approach. 87
G.8 Sliedrecht Left, minimum window approach with an offset from the starboard shore line of 10 meters. 88
G.9 Sliedrecht Left, average window approach. 88
G.10 Sliedrecht Left, average window approach using five iterations. 89
G.11 Sliedrecht Left, Combined approach: First minimum window, then average window. 89
G.12 Sliedrecht Left, Combined approach: First minimum window, then average window with five iterations. 89

Chapter 1

Introduction

In recent years, autonomous transportation technology has shifted from research labs to commercial development, fundamentally reshaping the transportation landscape. Major automotive manufacturers and tech companies have invested heavily in self-driving systems, resulting in sophisticated prototypes such as Waymo's autonomous taxis, which are currently operational in Los Angeles and San Francisco [1]. According to a 2023 report by McKinsey, global investment in autonomous vehicle (AV) technology is expected to surpass \$300–\$400 billion in revenue by 2035 [4], with the market reaching over \$200 billion in 2023 alone [30]. This surge has been fueled by the relatively structured nature of road environments—defined lanes, standardised signage, and universally enforced traffic laws—making them well-suited for algorithmic interpretation and control.

In contrast, the maritime industry has only recently begun to adopt automation at a comparable scale. While some progress has been made for coastal and open water applications, autonomous surface vessel (ASV) research for inland waterways remains an emerging field. The global market for autonomous ships was valued at \$5.61 billion in 2023 and is projected to grow to over \$12 billion by 2032 [19]. ASVs now represent a significant portion of this trend, accounting for approximately 64.5% of the autonomous vessel market in 2025 [18]. However, the transition is not straightforward. Maritime cases differ significantly from road networks: they are continuous, unbounded spaces where movement is governed by flexible conventions such as "good seamanship" or "ample time", rather than strict rules. These qualitative guidelines are challenging to translate to decision-making frameworks for autonomous navigation systems.

Furthermore, ASVs face challenges not typically encountered in automotive autonomy. Waterway topology is dynamic and often lacks consistent infrastructure like dedicated lanes, demanding fundamentally different approaches to path planning and control. Unlike road networks, navigable waterways are continuous, irregular spaces with no standardised layout, requiring flexible trajectory generation [3]. Additionally, machine perception systems must operate over significantly longer detection ranges and with fewer fixed landmarks [14].

While road and maritime domains differ, core autonomy principles such as machine perception, planning, and obstacle avoidance still apply. However, adapting for inland navigation requires accounting for shoreline boundaries, maritime navigation rules, and ambiguous infrastructure.

This thesis proposes a modular two-stage planning framework tailored to inland navigation, combining global ENC-informed routing with a COLREG-aware MPPI-based local planner. It performs a case study on the Damen Waterbus in structured, traffic-dense waterways, namely the area between Rotterdam and Dordrecht.

1.1 Related Works

The increasing interest in autonomous surface vessel (ASV) navigation has led to extensive research in various path planning methodologies to test their performance in maritime scenarios. Inland waterways pose unique challenges due to their constrained geometry, traffic interactions according to COLREGs [21, 22], and the necessity for real-time adaptive capabilities. Recent literature addressing these aspects can be broadly categorised into global path planning, local path planning and collision avoidance, regulatory compliance, utilisation of Electronic Navigational Charts (ENCs), specific inland or urban implementations, and benchmarking approaches.

1.1.1 Global Path Planning

Traditional grid-based planning approaches first discretise the environment into a grid, then apply graph search algorithms such as Dijkstra [5], A* [12], and D* [40]. With appropriate heuristics, they can be configured to adhere to environmental and regulatory constraints. To ensure dynamic feasibility and path smoothness, the generated path is often post-processed with a smoothing algorithm as presented in [57, 46, 17, 55, 50]. While these methods usually lead to sufficiently smooth trajectories, there is no guarantee of dynamic feasibility. In some cases [13], steering methods such as Dubins paths [7] are employed to ensure dynamic feasibility after smoothing. Notable inland-focused adaptations include advanced A* algorithms that incorporate depth constraints and vessel traffic separation schemes [60, 55].

Sampling-based methods aim to address dynamic feasibility and environmental complexity more effectively. Rapidly-exploring Random Trees (RRT) [27] and its optimised variants such as RRT* [24], informed RRT* (iRRT*) [2], and RRTX [34], have demonstrated considerable flexibility. However, similarly to grid-based methods, they often require extensive refinement to guarantee dynamic feasibility and regulatory compliance, particularly in constrained waterways [54, 8, 3]. Inland-focused extensions include improved RRT variants that consider river currents and grounding risks [13, 8].

He [13] proposes a Fast Marching (FM) method that solves the Eikonal equation given velocity maps with various constraints, generating cost-optimal paths via gradient descent. While efficient and adaptable to starboard positioning, its constant velocity assumption limits use in dynamic settings.

1.1.2 Local Path Planning and Collision Avoidance

Real-time collision avoidance methods often rely on techniques such as Velocity Obstacles [9], Artificial Potential Fields (APF)[25], or apply local replanning strategies based on RRT or grid search methods to adapt trajectories in response to environmental changes[8, 58, 59, 17]. These methods offer computational efficiency but frequently suffer from issues like local minima and oscillations when handling dense traffic scenarios or narrow passages [31, 29].

Predictive control frameworks such as Model Predictive Control (MPC) [53, 42, 15] and Model Predictive Path Integral (MPPI) control [51, 52, 10] have shown promising results. Recent developments in MPPI demonstrate significant advantages in maritime contexts, including real-time robustness [56], interaction-awareness in multi-vessel scenarios [23, 41], and improved efficiency by leveraging ancillary controllers to escape local minima [44, 43]. For instance, MPPI-based control has shown notable success in urban canal navigation, effectively handling dense multi-agent interactions and dynamic constraints [41, 33].

1.1.3 Regulatory Compliance (COLREGs)

Compliance with COLREGs remains a critical and challenging element of autonomous maritime navigation [21, 22]. Quantifying qualitative regulations, such as ”ample time” or ”good seamanship," for algorithmic implementation remains challenging without international agreements [11]. Numerous strategies have been proposed to bridge this gap. For example, Artificial Potential Field modifications incorporating COLREG constraints [29] have been proposed, as well as the hybrid VORRT-COLREG approach [6], which merges Velocity Obstacles with RRT to balance reactive avoidance and rule-aware planning. More recently, predictive control methods such as Model Predictive Control (MPC) [42] and Model Predictive Path Integral (MPPI) [41, 23] have shown promise in handling dynamic multi-agent interactions by embedding COLREG logic into the cost function.

Learning-based methods, including deep reinforcement learning, have been explored as a means to handle complex multi-agent navigation and ambiguous COLREG scenarios with greater flexibility and efficiency. Meyer et al. [31] and Zaccone [58] have demonstrated substantial progress in utilising learning-based frameworks for COLREG compliance, thereby improving real-time responsiveness and regulatory adherence. Comprehensive reviews by Hu et al. [16] further illustrate how learning-based approaches can systematically incorporate human-like judgment and adaptability in maritime collision avoidance.

1.1.4 Utilisation of Electronic Navigational Charts (ENC)

Electronic Navigational Charts (ENCs) standardised under the IHO's S-57 specification [20, 32, 38] provide detailed environmental and infrastructural information for all commercial shipping. While many path planning approaches now incorporate ENC data, their utilisation often remains superficial or limited in scope, particularly in constrained inland environments.

Grid-based approaches such as [17, 57] commonly use ENC-derived cost layers for A*-style search, but often suffer from coarse resolutions, inadequate handling of turning radii, and limited compliance with COLREGs.

Sampling-based planners [3, 8] incorporate ENC constraints such as depth and obstacle buffers, lack a steering function, assume static environments, and require case-specific tuning. A hybrid method combining A* with potential fields and a velocity adaptive cost function [57] as a local planner offers improved smoothness and efficiency, yet exhibits oscillations.

Despite being essential for representing the physical and regulatory structure of inland waterways, ENCs are often underutilised in motion planning. When they are used, it is usually only for basic environmental constraints, without integrating COLREGs. In particular, the use of inland ENCs (IENCs) for structured, rule-compliant planning in dense or constrained traffic scenarios remains a significant gap.

1.1.5 Urban and Inland ASV Implementations

Several practical implementations have demonstrated the potential and challenges inherent in inland and urban ASV operations. Notable projects, such as Roboat [47, 48, 49, 43], which address Amsterdam's dense and constrained canal systems, highlight the importance of precise manoeuvring and reliable interaction strategies. In addition, social trajectory planning methods developed by Park et al. [35] demonstrate the complexity of human-like interactions required in inland waterway operations. Additionally, case studies from ports such as Ulsan [57] illustrate the necessity of tailoring algorithms to local environmental and operational conditions, as highlighted by the assignment of different berthing locations to various vessel types near the port.

While several global path planning methods have been developed for inland navigation scenarios, local path planning remains largely underexplored. To date, the only identified local planning framework for inland waters that adheres to COLREGs is Trevisan's AI-MPPI and Biased-MPPI approach [43], designed for small autonomous vessels operating in urban canals. However, this work does not address the operational constraints or manoeuvring characteristics of larger commercial vessels in larger-scale inland shipping routes, which require fundamentally different planning strategies due to their reduced manoeuvrability and wider safety margins.

1.1.6 Benchmarking and Validation

Despite substantial methodological progress, standardised benchmarking remains relatively undeveloped, limiting comparability across studies. Few works include reproducible experimental setups, and direct comparisons between competing methods are rare. Current literature highlights several key gaps: the absence of structured testing scenarios [36], limited use of systematic evaluation frameworks for collision avoidance [28, 50], and a lack of consistent metrics to quantify aspects such as collision risk, COLREG compliance, and path smoothness [26, 11].

A particularly persistent issue is the inconsistent definition of COLREG states and encounter types across studies. While some papers rely on hardcoded distance thresholds or angle-based heuristics, these definitions vary widely [11, 16, 29, 58, 31], undermining cross-method evaluation. DNV [36] and Hagen [11] have proposed a step towards a standardised system of COLREG state definitions, which this thesis adopts to ensure consistency in regulatory interpretation.

In summary, this thesis addresses these benchmarking limitations by introducing structured test scenarios, standardised COLREG state classification, and quantitative evaluation metrics. Together with its contributions to global and local path planning, ENC integration, and regulatory compliance, this work presents a reproducible and modular framework designed for practical ASV deployment in inland waterways.

1.1.7 Research Gaps

The literature review [45] on path planning literature, focused on inland scenarios, reveals three persistent gaps:

1. Global planners are typically designed for open-ocean or coastal operations, often employing coarse grids that fail to create paths that comply to the vessel dynamics.
2. Most studies emphasise head-on and overtaking and crossing scenarios in open water or urban inland scenarios, leaving the coordination of larger vessels operating in inland shipping routes unaddressed.
3. Most studies do not incorporate quantitative benchmarks for dynamic feasibility, trajectory quality, and regulatory compliance. Standardised testing environments for constrained inland waterways remain undeveloped.

1.2 Research Questions

The primary objective of this thesis is to design and demonstrate a path planning architecture for autonomous vessels, with a focus on global and local path planning strategies. The development of a low-level controller falls outside the scope of this work.

The proposed architecture is designed to be applicable across a wide range of vessel types and operational environments. This is achieved through a modular design, which enables planners to be configured using only a few high-level dynamic parameters, such as maximum velocity and acceleration. As a result, the controller is the only component within the Guidance, Navigation, and Control (GNC) framework that requires extensive vessel-specific customisation.

Research Question

How can a modular path planning architecture generate desirable trajectories for inland autonomous vessel navigation between docks while safely adapting to marine traffic using ENCs?

Subquestions

1. Which quantitative metrics (e.g., path length, curvature variance, computation latency, COLREG compliance rate) best evaluate both global and local planner performance in inland waterways?

2. How can S-57 ENC layers (depth contours, fairway boundaries, infrastructure) be leveraged to generate safe, starboard-biased global routes that respect vessel dynamic constraints?

3. What effects do different post-processing techniques have on global paths?

4. How can an MPPI-based local controller be designed to balance collision avoidance, COLREG adherence, and real-time execution under multi-vessel encounters?

5. How do the global and local planners interact across sailing phases (undocking, cruising, docking), and what strategies ensure smooth phase transitions and overall mission success?

To address the research questions, a modular, two-stage path planning architecture is proposed for dock-to-dock inland navigation of the Damen Waterbus, operating in the inland Area between Rotterdam and Dordrecht, using S-57 Electronic Navigational Charts (ENCs). A global planner Constructs routes using the Waterway-Axis, extracted from the ENCs and then projects them toward the starboard shore. This process is followed by several refinement steps to increase path quality, which is assessed using various metrics addressing feasibility. Next, a local planner implements a real-time, traffic-aware Biased-MPPI [44] controller, using ancillary controllers which focus on global path following, end goal reaching and cruise control. The Biased-MPPI controller incorporates a cost function to promote following the global path, varying behaviour based on different sailing phases and handling diverse traffic scenarios by changing between policies.

To summarise, this thesis contributes a modular two-stage dock-to-dock navigation pipeline combining ENC-based global path planning with starboard projection and smoothing with a real-time Biased-MPPI local controller, which switches policies based on the scenario.

The remainder of the thesis is organised as follows. Chapter 2 provides background, data sources, vessel and environment models, and performance metrics. Chapter 3 details the strategy for extracting and refining a global path from ENC data. Chapter 4 formulates the design of the cost function and ancillary controllers for the Biased-MPPI system. Chapter 5 presents experimental results across diverse inland Dock-to-Dock and traffic scenarios. Chapter 6 concludes with key insights, limitations, and directions for future inland ASV deployments.

Chapter 2

Background

This chapter defines the context, data, and assumptions used throughout the thesis. It first introduces the dock-to-dock mission, sailing phases and the basic traffic scenarios considered. It then describes the two data sources used (S-57 ENCs and Waterbus AIS data) and the high-level system architecture. Next, it lists the thesis-level metrics used later for evaluation (success rate, safety, travel distance/time, computation time). It states the key dynamic constraints of the vessel (sway, surge, yaw rate). Together, these elements frame how the global and local planners are assessed under realistic inland navigation conditions.

The chapter begins by outlining the scope of the system architecture, followed by a description of the relevant data sources, including Electronic Navigational Charts (ENCs) and the AIS-based Triton dataset. hereafter, the performance metrics used to assess trajectory quality are introduced. Finally, the dynamic constraints and planner-specific requirements are described in detail, covering both global feasibility and local traffic-aware behaviour.

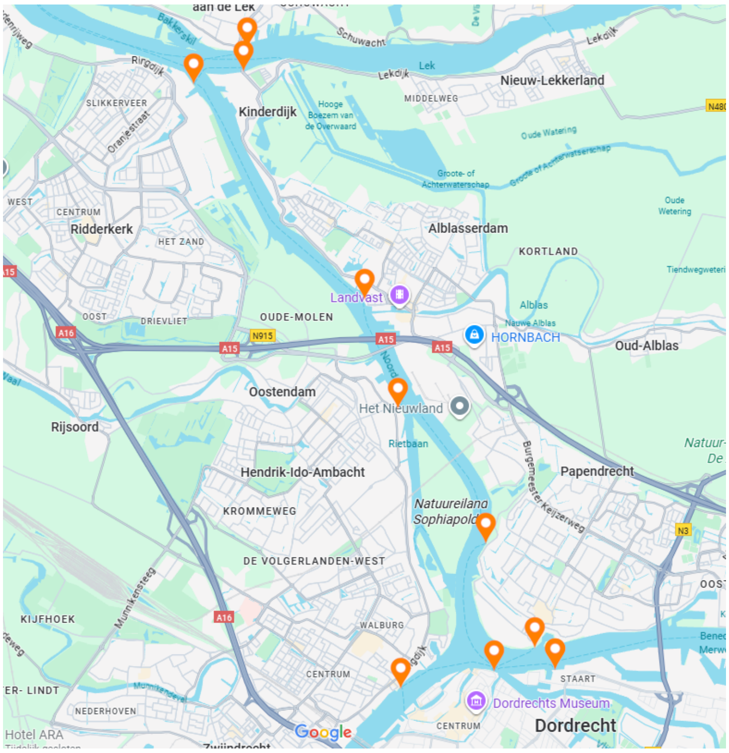


Figure 2.1: Overview of all docks used when evaluating autonomous dock-to-dock voyages.

2.1 Mission: Safe Dock-to-Dock Sailing

The overall dock-to-dock voyage is executed as a sequence of sailing phases with different objectives, speeds and priorities, while interacting with other traffic and adhering to local waterway rules. Figure 2.1 shows an overview of all docks used for dock-to-dock evaluation in this thesis.

This section introduces those phases and the traffic scenarios considered in this thesis. Technical details follow in later chapters.

2.1.1 Sailing phases

We partition a dock-to-dock mission into five phases with simple, observable transitions (radius or geometry-based). Each phase has a clear objective and a small set of safety constraints that remain active throughout.

- Undocking: Leave the berth safely by using low-speed lateral movement until clear of the dock. End phase when clear of obstacles and sufficiently far away from the dock.
- Sailing: Accelerate to and maintain cruising velocity while following the planned route and keeping to starboard. Avoid any obstacles according to the traffic scenario handling policy. Exit this phase when entering the "approach" radius of the destination dock.
- Approach: Reduce speed and head straight to the dock. Exit phase when entering the docking radius.
- Docking: Precise manoeuvring towards the berth at very low speed while aiming to set the heading to align with the dock. When at the berth, switch to the next sailing phase.
- Berthing: Hold position to load and unload passengers.

Figure 2.2 illustrates the sailing phase logic. The voyage begins in the undocking zone (blue), where low-speed lateral motion is used to clear the berth. In open water (white), the vessel cruises along the planned route. Upon entering the approach radius (yellow), speed is reduced and the heading moves directly toward the berth. Inside the docking radius (orange), precise low-speed manoeuvres align the vessel with the dock. The voyage ends at the berth (red), where the vessel holds position for berthing. The two distance-to-dock thresholds trigger phase transitions.

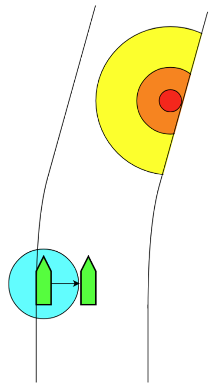


Figure 2.2: Schematic overview of Sailing phases. Blue = Undocking, White = Sailing, Yellow = Approach, Orange = Docking, Red = Berthing

2.1.2 Traffic scenarios

Figure 2.3 illustrates the scenarios examined during this thesis and the corresponding required behaviour. Three basic traffic encounters are evaluated:

- Head-On: The Own Ship gives way by moving towards starboard.
- Crossing: In the crossing scenario evaluated in this thesis, the Own Ship has to give way as it encounters the Target Ship at starboard.
- Overtaking: With a slower vessel ahead on a similar heading, Own Ship bears slightly to port and then overtakes along the Target Ship's port side.

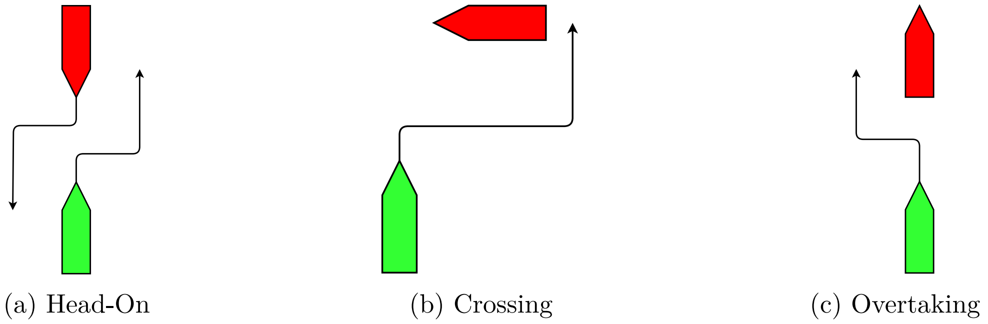

(a) Head-On (b) Crossing (c) Overtaking

Figure 2.3: Three traffic scenarios considered in this thesis—overtaking, head-on, and crossing. Own Ship (green), Target Ship (red).

In the scope of this Thesis, ”Own Ship” refers to the vessel being controlled. ”Target Ship” denotes any encountered vessel. The policies the local planner uses for these scenarios are presented in Chapter 4.

2.2 High-level Dynamic Constraints

The Damen Waterbus, shown in Figure 2.4 is equipped with two rear rudders. The placement of the rear rudders allows the vessel to move laterally when set to the correct angles. This method is used by captains when performing docking and undocking manoeuvres. As low-level controllers are out of the scope of this thesis the lateral movement (sway) is simplified. Lateral movement is only modelled for cases where the surge velocity is sufficiently small.
The following dynamic and physical constraints are assumed for the vessel:

Figure 2.4: Side view of the Damen Waterbus

- Maximum sure velocity: $v_{\text{surge}} = 20$ knots $(10.29\,\mathrm{m/s})$
- Maximum sway velocity: $v_{\text{sway}} = 1.5\,\mathrm{m/s}$

- Dimensions: Length × Breadth × Draught $= 28.6 \times 7.5 \times 1.5\,\mathrm{m}$
- Maximum acceleration/deceleration: $a = 1.11$ knots/s ($0.571\,\mathrm{m/s^2}$)
- Maximum turning rate: $\omega = 5°/\mathrm{s}$ ($0.087\,\mathrm{rad/s}$)
- Drift: Neglected in this study at velocities greater than 2 m/s for simplification

Given these constraints, the vessel's minimum turning radius can be computed using the relationship between linear and angular velocity:

$$r_{\min} = \frac{v}{\omega} = \frac{10.29}{0.087} \approx 118\,\mathrm{m}$$

This minimum turning radius forms a key constraint for determining the feasibility of any generated path.

2.3 Data

This paragraph describes the two data sources used. S-57 ENCs supply the static map layers for routing and feasibility, and the Triton dataset provides one month of the Waterbus AIS positions.

2.3.1 Electronic Nautical Charts

This thesis examines the applicability for path planning of S-57 Electronic Navigational Charts (ENCs), the current standard for digital hydrographic data exchange. Commercial vessels, such as the Damen Waterbus, are required to use navigation systems which have these charts integrated, while leisure vessels are not subject to this mandate.

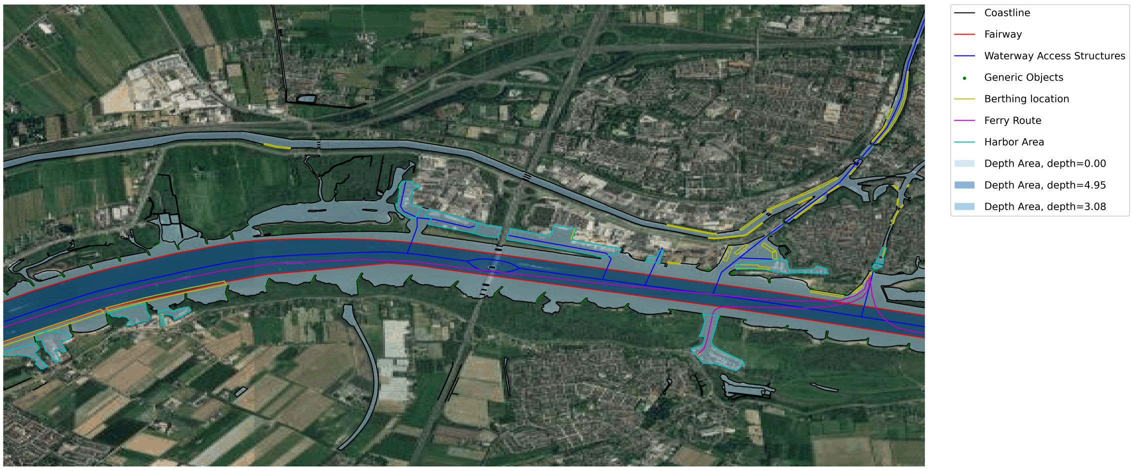


Figure 2.5: IENC of the inland waters at Gorinchem.

S-57 ENCs include a variety of over 180 object classes, such as depth areas, land areas, coastlines, and static waterborne objects like buoys, pylons, and bridges [38]. Figure 2.5 showcases an example of some layers found in Rijkswaterstaat's IENCs. This thesis uses two of these layers as a foundation of the path planning architecture: The Fairway and the Waterway Axis.

In this research, the Fairway object is designated as the navigable area of the canal. According to Rijkswaterstaat's guidelines, the Fairway object is defined as:

”Part of the navigable waterway area where a certain depth within a certain width is available for continuous navigation That part of a river where the main navigable channel for vessels of larger size lies” [38].

Rijkswaterstaat’s ENC encoding practices ensure that the Fairway object shares its geometry with the Depth Area object. Areas lacking assigned depth information are considered unsafe for navigation within the scope of this research.

The Waterway Axis is an S-57 ENC object, which is defined as the central line of a Fairway. In the absence of a designated Fairway, it represents the central line of the waterway. This axis forms a network of interconnected lines, which is used by the mission planner to construct a graph of the waterway system

2.3.2 Triton Dataset

Another data source used in this thesis is the Triton dataset. This dataset consists of historical AIS recordings of the Damen Waterbus over one month, sampled every 10 s. Data recorded includes GPS position, speed-over-ground, heading. Individual dock-to-dock voyages are extracted by detecting stationary periods at docks as shown in Figure 2.6.
The AIS trajectories provide a reference for human navigation: they show consistent manoeuvres between docks, show a clear starboard-side bias, and exhibit greater spread near bends and intersections. This dataset serves as a real-world benchmark for trajectory comparison and motivates design choices such as a starboard-shore bias in the global planner.

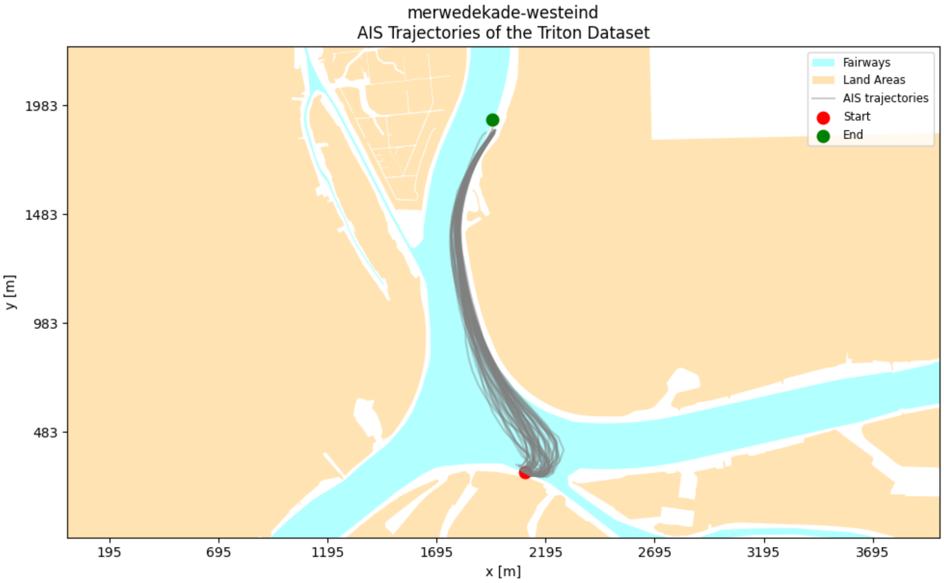


Figure 2.6: All extractable AIS routes from the Triton dataset between stations Merwedekade and Westeind.

2.4 System Overview and Constraints

Figure 2.7 provides a concise overview of the scope of the work presented in this thesis. This thesis focuses on a path planning system combining local and global path planning. The global planner uses infrastructural constraints extracted from ENCs and high-level dynamic constraints corresponding to the vessel configuration and extracts a list of waypoints near the starboard shore.

The local planner uses the list of waypoints generated by the global planner as a heuristic, taking the high-level vessel dynamics into account. The Local planner replans the path in real-time based on current traffic conditions, adhering to the traffic laws. As a result, the local planner delivers a list of waypoints including heading and corresponding velocities which is fed to the motion controller.

In-depth mission management and lower-level motion controllers are excluded from this work.

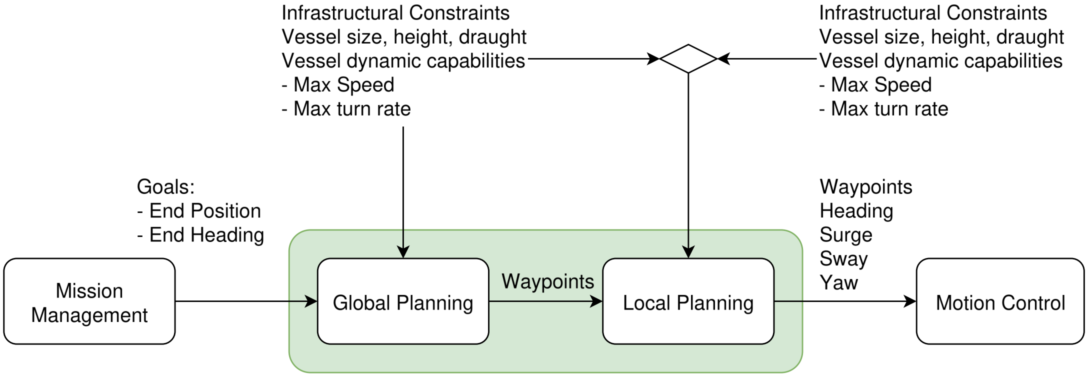


Figure 2.7: High-level system overview and scope of the work (green) and the corresponding inputs and outputs.

2.4.1 Performance Metrics

To evaluate path planning performance, high-level indicators are presented in Table 2.1.

Table 2.1: High-level performance metrics.

Metric	Stage	How it is computed (summary)
Computation Time	Local	Mean/maximum computation time per local replanning cycle.
Success Rate	Local	Fraction of runs that reach the dock.
Safety	Global & Local	Share of runs with no safety violations (no Fairway crossings, no collision, keep a safe distance from other vessels.
Travel Distance	Local	Executed trajectory length (dock-to-dock).
Travel Time	Local	Elapsed time from departure to docking.
Human-likeness	Local	Deviation of executed distance and travel time from AIS benchmarks for the same leg (e.g., difference from AIS mean/median across historical runs).

Computation time is included to verify that the local planner can suffice the real-time constraints for inland marine navigation. Success rate directly measures the planner's ability to complete the intended voyage, while safety ensures compliance with navigational rules, keeping safe distances from other vessels. Travel distance and travel time quantify the efficiency of the executed path in terms of route efficiency and speed efficiency. Finally, human-likeness compares autonomous trajectories against historical AIS data, to check how well the vessel's behaviour aligns with established navigation practices and its interpretability to other waterway users. Together, these metrics provide a balanced evaluation of efficiency, safety, realism, and computational feasibility.

Chapter 3

Global Path Planning

To develop a global path for inland maritime navigation, the unique challenges of waterway environments require addressing. Unlike automotive navigation, where vehicles adhere to well-defined lanes and traffic laws, maritime navigation involves open waterways without fixed lanes and more flexible traffic regulations. For instance, while vessels typically keep to the starboard side of a channel, the exact positioning can vary based on factors like vessel shape and size, waterway width and traffic conditions. Additionally, captains may adjust their paths to optimise travel time and fuel efficiency when permitted by the conditions. Therefore, global paths for inland maritime scenarios must adhere to general navigation practices with adaptability to dynamic and less structured environments.

3.1 Route Planning

One ENC displays information within the bounds of a particular area. The area where the Damen Waterbus operates requires the use of multiple ENCs created by Rijkswaterstaat and the Port of Rotterdam [39]. To merge the data, layers of interest are extracted from each ENC and stored in a CSV. The polygon-based Fairway objects are merged into one large polygon to prevent any projection issues at the ENC boundaries. Figure 3.1 shows an example of two merged ENCs.

To plan the route between docks, the waterway axis object found in the S-57 Inland ENCs is used as a foundation. This Linestring object represents the centreline of a Fairway and creates a network of interconnected waterways. This object is extracted from each ENC and added to a list. Then, a connected graph, shown in Figure 3.2, is created by using the Networkx Python library, where each point along the linestring functions as a node and the edges are weighted by the distances between them.

To plan a route between docks, the graph is expanded with locations of the start and end dock by adding an edge between the dock node and its nearest node on the graph. The result of the mission planner can be seen in Figure 3.3 This allows the mission planner to find a high-level route from one dock to another.

effectively

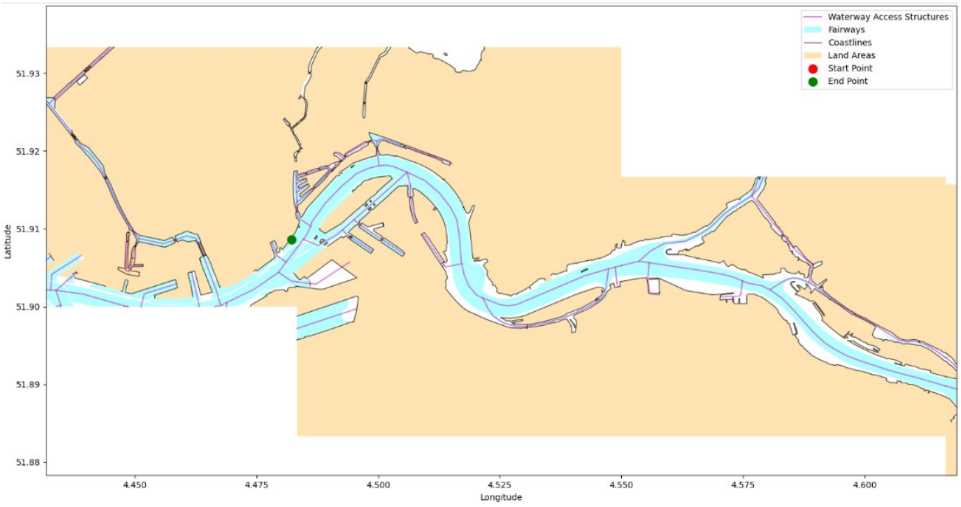


Figure 3.1: Two combined IENC of the inland waters of Rotterdam.

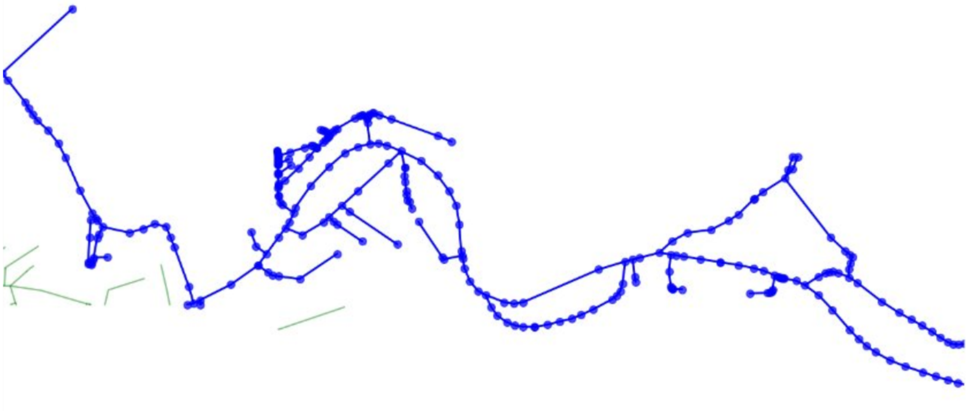


Figure 3.2: Graph extracted from the waterway axis structure.

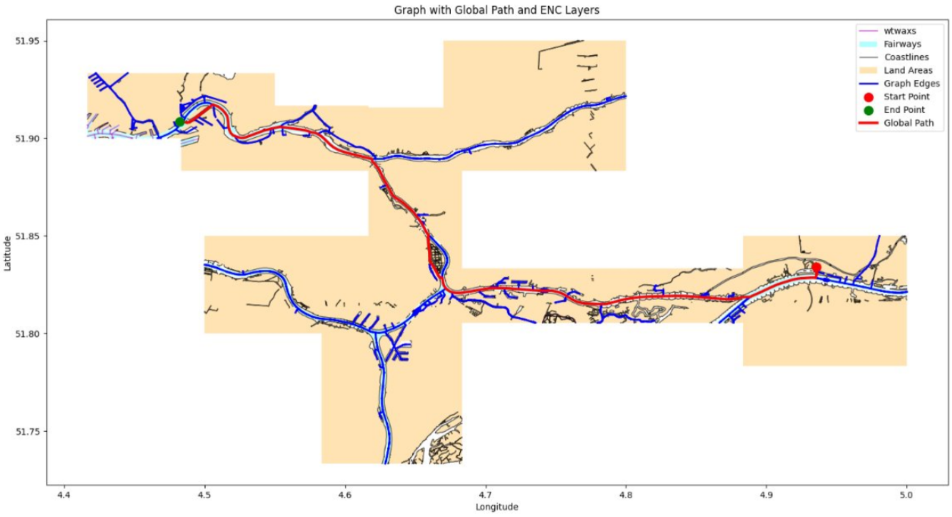


Figure 3.3: Global path found by mission planner using the waterway axis object from the merged ENC data.

3.2 Refining the Global Path

The high-level global path generated by the mission planner may not fully meet the requirements outlined in Section 3.4, particularly starboard shore positioning and the presence of sharp turns. To address these issues, two approaches can be considered:

1. This approach involves refining the global path before it is processed by the local planner to ensure feasibility and meeting navigational constraints. The goal is to smooth out sharp turns and adjust the path to maintain the preferred starboard shore position. This allows the global path to become more aligned with the vessel's dynamic capabilities and standard sailing practices.

2. Alternatively, the global path generated by the mission planner can serve as a baseline, with the local planner addressing feasibility issues and navigational constraints in real-time.

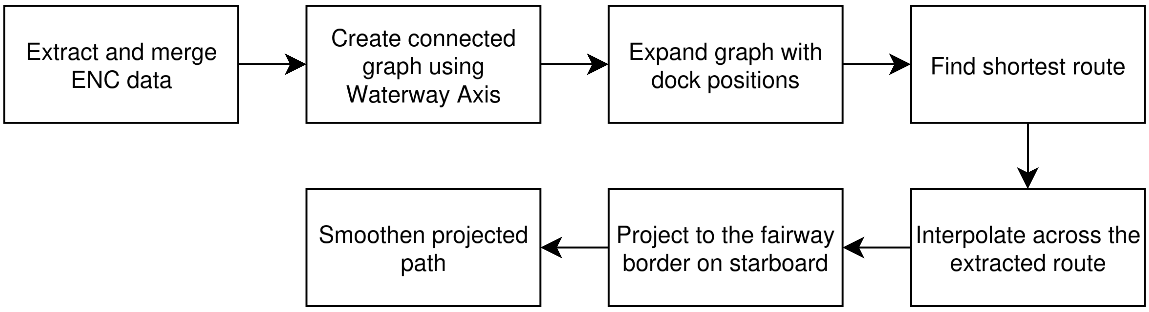

Extract and merge ENC data
Create connected graph using Waterway Axis
Expand graph with dock positions
Find shortest route
Interpolate across the extracted route
Project to the fairway border on starboard
Smoothen projected path

Figure 3.4: Overview of the global path planning system architecture.

This section explores the first approach, focusing on the process and benefits of refining the global path.

3.2.1 Projection

As shown in 3.4, the found path is interpolated after the shortest route is extracted to ensure equidistant points. The distance between the interpolation points is a tunable parameter. These points are then projected onto the edge of the Fairway orthogonally towards starboard, aligning the path with the preferred starboard shore position. A slight lateral offset is applied, keeping the path close to the Fairway edge (but not coincident) to discourage crossings. An overview of the result after interpolation and projection is shown in Figure 3.5

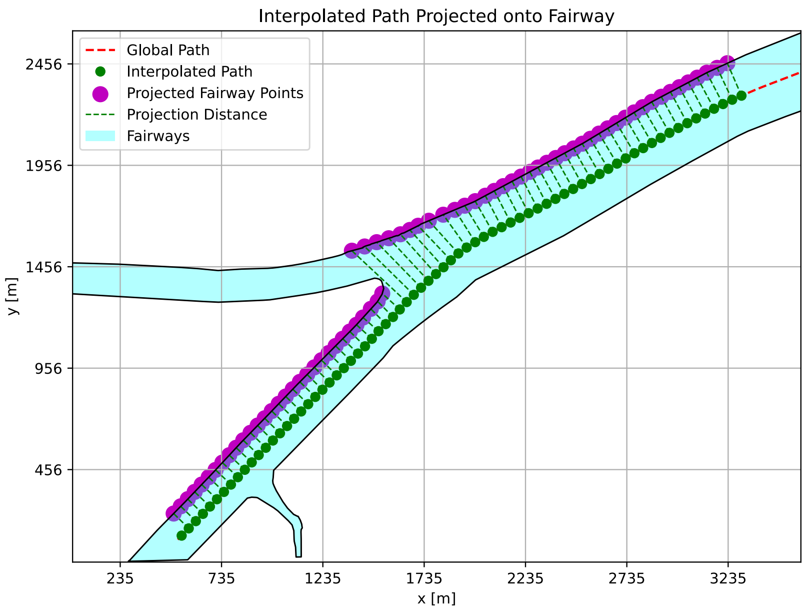


Figure 3.5: Interpolated path projected onto the Fairway.

Figure 3.5 shows an example of the result of projecting the interpolated path onto the Fairway in one of the testing scenarios. Upon evaluation, a significant discontinuity is found at the left-turning intersection. This large gap is caused by the abrupt change in projection caused by the vessel's left turn, leading to a sudden shift in the Fairway segment onto which the path points are projected.

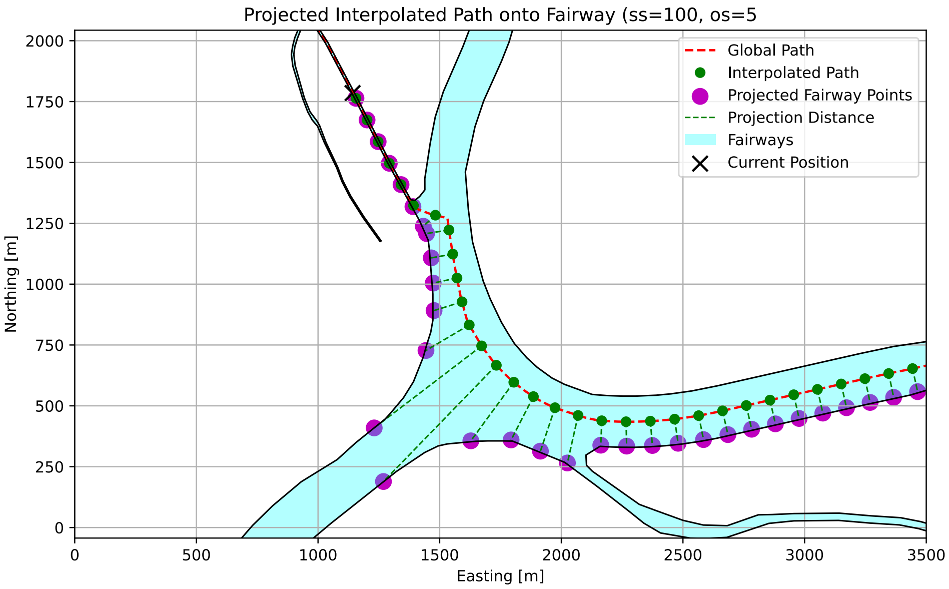


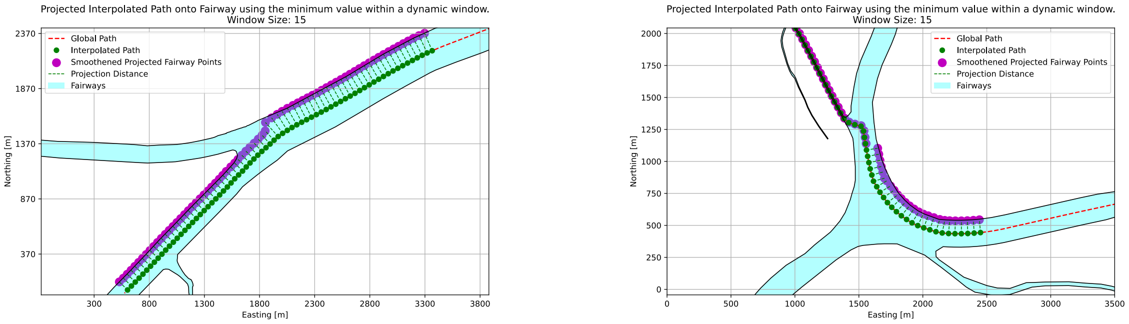
Figure 3.6: projected path in Dordrecht

Another example of issues due to projection is shown in Figure 3.6. The triangular shape of this intersection causes the points to be projected into the wrong canal, creating a very irregular path.

3.2.2 smoothing methods

In this section, two straightforward smoothing techniques are introduced. They are designed to enhance the consistency of the distance between projected points along a Fairway object. These methods adjust the projection distance by analysing a dynamic window of neighbouring points to accommodate variations in canal width and intersections.

The first method is the minimum distance approach. Its effect is shown in Figure 3.7. This technique recalculates the projection distance for each point by setting it to the minimum distance among all points within a specified window. This method aims to prevent abrupt deviations.

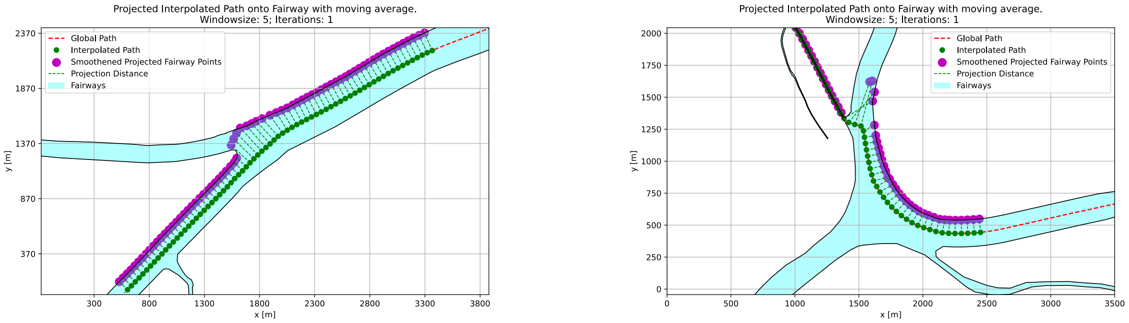


(a) Sliedrecht

(b) Dordrecht

Figure 3.7: Apply the minimum windowsize method.

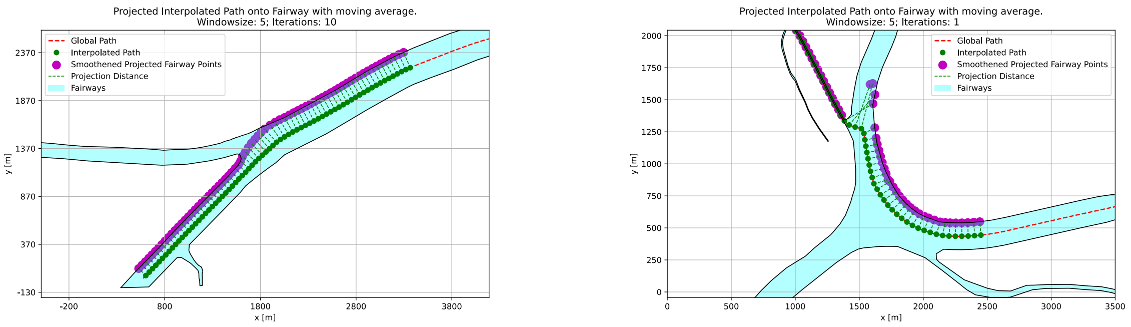
The second method is the average distance approach, shown in Figure 3.8. This method computes the average distance of all points within the dynamic window and uses this mean value for projection. Whenever points are moved outside of the Fairway object using this method, the point is clipped back to the intersection of the orthogonal line and the Fairway object. This approach promotes a smoother transition through varying canal widths and intersections. Additionally, applying this method over multiple repetitions (Figure 3.9) is examined to assess its impact on path refinement.



(a) Sliedrecht

(b) Dordrecht

Figure 3.8: Apply the average windowsize smoothing, clipping points outside the Fairway back to their original position.

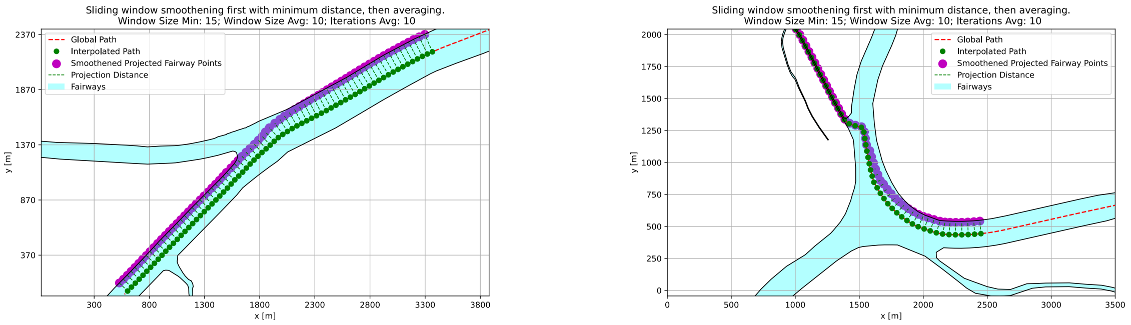


(a) Sliedrecht

(b) Dordrecht Flipped

Figure 3.9: Apply the average windowsize method for multiple iterations.

A hybrid method, shown in Figure 3.10, integrates both minimum and average distance strategies to leverage their respective advantages.



(a) Sliedrecht

(b) Dordrecht Flipped

Figure 3.10: Hybrid method which first applies the minimum windowed smoothing and then applies multiple iterations of the average window smoothing method.

The effectiveness of these smoothing methods is influenced by parameters such as the step size of the interpolated path, the dynamic window size, offset from shoreline and the amount of repetitions of the average distance approach. As illustrated in Figure 3.10, the hybrid method appears visually smoother than the other approaches. The subsequent section presents quantitative metrics to evaluate this observation objectively.

3.3 Global Planner evaluation

To evaluate the proposed methods, paths are generated using a grid search approach where combinations of parameter settings with defined ranges, found in Table 3.1, were systematically explored. Each unique combination of settings is applied across ten testing scenarios along the route from Damen Shipyards in Gorinchem to the Erasmusbrug in Rotterdam. These scenarios encompass intersections with various characteristics, including steep and sharp angles, as well as transitions from narrow waterways to wider channels. A path is considered sufficient if it meets all the requirements (Paragraph 3.3.2) across all evaluated scenarios. Visual representations of each scenario are found in Appendix G.

Table 3.1: Tunable parameters used in the grid search for global-path refinement.

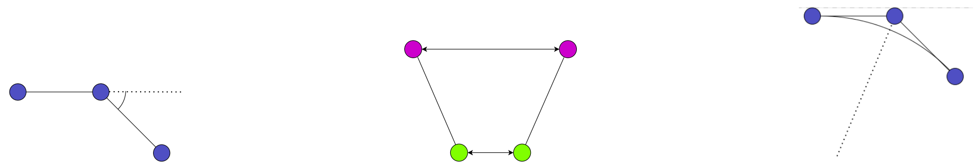
Parameter	Unit	Description	Grid
Interpolation step size	m	Point spacing along the extracted shortest path prior to projection.	$\{20, 40, \ldots, 300\}$
Dynamic window size	points	Neighbour count used to compute local statistics (min/mean).	$\{5, 10, \ldots, 30\}$
Shoreline offset	m	Lateral offset from the Fairway boundary after orthogonal projection.	$\{5, 10, \ldots, 30\}$
Avg-method iterations	—	Number of repetitions of the average-distance smoothing pass.	$\{1, 2, \ldots, 10,$ $15, 20, \ldots, 50\}$

3.3.1 Global Planner Metrics

The path planning system must adhere to the following criteria to ensure feasible and efficient navigation. An overview of all metrics is found in Table 3.2. Figure 3.11 presents a supporting illustration to help visualise the respective metrics.

Table 3.2: Performance metrics, applicability, and descriptions

Metric	Unit	Description
Angle Between Consecutive Segments	°	Maximum interior angle formed by any two consecutive path segments (sharp angles indicate low smoothness).
Step-Size Projection Ratio	–	Max ratio of distance between raw waypoints to distance between their projections onto the Fairway centreline (irregularity indicator).
Turning Radius	m	Average across the trajectory indicates smoothness, and Minimum provides feasibility checks.



(a) Angle between segments (b) Stepsize Projection Ratio (c) Turning Radius

Figure 3.11: Illustration of various performance metrics used to evaluate the global planner.

3.3.2 Global Planner Requirements

The metrics presented in Section 3.3.1 are used to set up detailed requirements. A path is found infeasible when it fails to meet these requirements:

1. Feasibility: When a line segment between two consecutive waypoints intersects the Fairway, the path is found infeasible.

2. Minimum Turning Radius: Any path with a turning radius less than 118 m is deemed unfeasible, as this is the vessel's minimum turning capability.

3. Segment Angle Limit: The angle between two consecutive line segments must not exceed 90° to maintain navigational stability.

4. Maximum Step Size Projection Ratio: The maximum ratio across all projections must not exceed 2, ensuring that the refined path closely follows the intended trajectory without significant deviations.

Requirement 1 ensures that every consecutive line segment lies within the Fairway, the area of the canal which guarantees sufficient under-keel clearance. An example of an infeasible segment is shown in Figure 3.12.

Requirement 2 supports sufficiently smooth trajectories when sailing at maximum velocity. Requirement 3 supports Requirement 2 because there are cases where a turning radius exceeds 108 meters and has a large angle between consecutive segments. This can cause a reversing manoeuvre, which is undesirable for safe and efficient operation. The Step size Projection Ratio (Figure 3.11c) is used to measure irregularities between the original and projected line segments.

A large Step size projection ratio indicates irregular behaviour. For example, waypoints are projected across an intersection into an adjacent channel, as in the Dordrecht scenario of Figure 3.6. Requirement 4 is created to ensure sufficiently spaced projections. Requirement 4 constrains this ratio to maintain regular waypoint spacing and prevent such cross-channel projections.

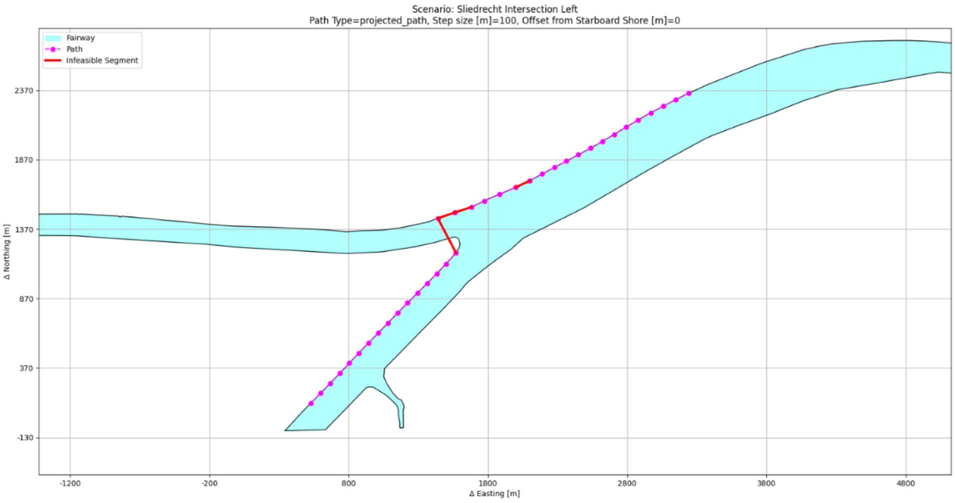


Figure 3.12: Projected path with an infeasible line segment, intersecting the Fairway.

3.3.3 Result projected paths without smoothing

This method has one tunable parameter, namely the interpolation step size. The step size projection ratio is a good indicator to test sufficient path quality across different step sizes.

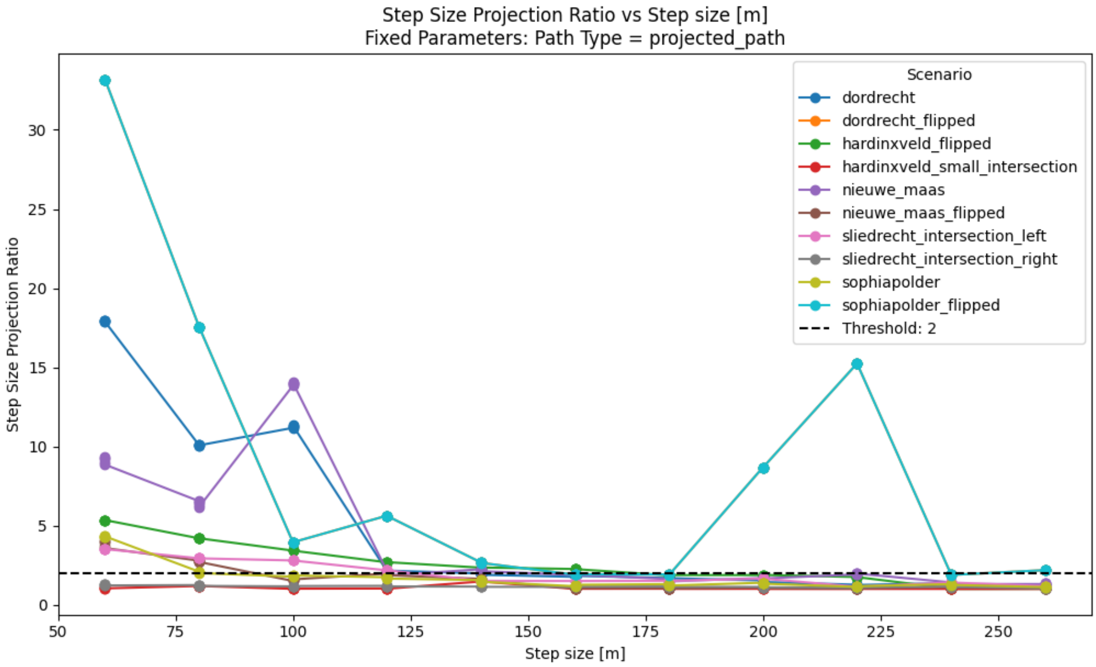


Figure 3.13: maximum Step Size Projection Ratio with varying step sizes for all scenarios.

Modifying the interpolation step size improved compliance with performance metrics. However, Figure 3.13 reveals that in some instances, line segments between consecutive points intersected the Fairway boundaries, implying infeasibility. Increasing the distance between waypoints generally enhances path smoothness. Yet, in scenarios involving left turns where the vessel crosses the waterway, such as the scenario shown in Figure 3.6, high step size projection ratios, large angles between line segments and large discrepancies between refined and global path lengths were detected.

To conclude, projection alone isn't enough to guarantee feasibility and smoothness; additional refinement steps are required.

3.3.4 Results of smoothing methods

Table 3.3 provides an overview of scenarios with the corresponding smoothing method. A checkmark indicates that at least one parameter configuration satisfied all requirements. A complete, per-scenario listing of feasible parameter ranges is provided in Appendix F (Tables F.1–F.10).

Across the ten scenarios, only five yielded at least one feasible configuration: Dordrecht (flipped), Hardinxveld, Hardinxveld (flipped), Sliedrecht Intersection Left, and Sliedrecht Intersection Right. No feasible settings were found for Dordrecht, Nieuwe Maas, Nieuwe Maas (flipped), Sophiapolder, or Sophiapolder (flipped) within the tested ranges. For scenarios with feasible outcomes, the appendix tables report the corresponding parameter ranges per method.

Table 3.3: Method feasibility across scenarios (✓= Contains configuration which satisfies all requirements).

Scenario	Projected	Min	Avg	Hybrid
Dordrecht	✗	✗	✗	✗
Dordrecht (flipped)	✗	✓	✓	✓
Hardinxveld	✓	✓	✓	✓
Hardinxveld (flipped)	✗	✓	✓	✓
Nieuwe Maas	✗	✗	✗	✗
Nieuwe Maas (flipped)	✗	✗	✗	✗
Sliedrecht Intersection Left	✓	✓	✓	✓
Sliedrecht Intersection Right	✓	✓	✓	✓
Sophiapolder	✗	✗	✗	✗
Sophiapolder (flipped)	✗	✗	✗	✗
Feasible scenarios (of 10)	3	5	5	5

Table 3.3 shows that smoothing methods improve the number of feasible scenarios compared to projection without smoothing. Considering feasibility, there is no difference between all smoothing methods. Table 3.4 summarises the grid-search results into parameter bounds that yielded feasible paths across scenarios.

Table 3.4: Feasible parameter ranges (robust across most scenarios).

Parameter	Recommended range
Step size	100–160 m
Offset	5–20 m
Minimum Window Size	5–15 points
Average Window Size	5–20 points
Number of Iterations	5–10 Iterations

Table 3.5 summarises failures by method across all scenarios and parameter settings. The "Fails" represent the number (and percentage of trials) that violated at least one requirement. Columns R1–R4 report, for each requirement, the number of trials that violated that specific requirement and include the corresponding percentage out of all trials for that method.

Table 3.5: Requirement 3.3.2 violations and rates per method

Method	Trials	Fails	R1	R2	R3	R4
projected_path	550	491 (89.3%)	236 (42.9%)	58 (10.5%)	241 (43.8%)	194 (35.3%)
windowed_min_path	1760	1306 (74.2%)	994 (56.5%)	303 (17.2%)	76 (4.3%)	47 (2.7%)
windowed_avg_path	12900	9822 (76.1%)	7581 (58.8%)	2091 (16.2%)	632 (4.9%)	468 (3.6%)
windowed_min_avg_path	33899	24159 (71.3%)	18709 (55.2%)	5296 (15.6%)	1180 (3.5%)	457 (1.3%)

Overall, the projected path without smoothing performs worst (89.3% failed). Any smoothing sharply reduces Requirement 3 (Angle Segments) and Requirement 4 (Step Size Projection Ratio) (to 3–5%), indicating clear regularisation benefits. Among the smoothing strategies, the hybrid variant yields the lowest overall failure rate (71.3%) and the fewest Requirement 4 violations (1.3%), with Requirement 2 comparable to the other smoothers. The most common reason for failure across all smoothers is feasibility (Requirement 1 of 55–59%). Hence, if a single method must be chosen, the hybrid variant is preferable; its advantage over the other smoothing strategies is consistent but modest. Residual failures are largely feasibility-related (Requirement 1), which is not eliminated by smoothing alone.

When evaluating the five scenarios that yield zero feasible results visually, it was found that these scenarios. Examples shown in Figures 3.14-3.16 share one of three common issues:

1. The scenario presented in Figure 3.14 contains a switch between moving through a very narrow canal of 10 meters wide and a very wide waterway of 300 to 500 meters wide. This causes the path to intersect with the Fairway object more often.

2. The waterway axis object contains a sharp corner, therefore, the projected points intersect and cause a very spurious path. An example of this occurrence is shown in Figure 3.15.

3. Data generated by the port of Rotterdam contains Fairway objects which do not share any line segments (shown in Figure 3.16. Therefore, the unionising step of the Fairway objects does not work. This step is crucial for the functionality of the global planner.

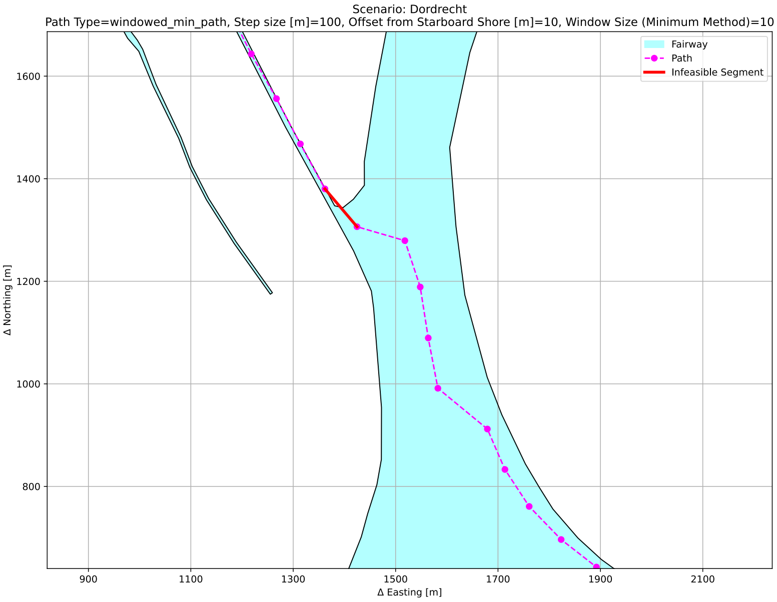


Figure 3.14: Commonly caused infeasible paths due to a width change in the lane transition.

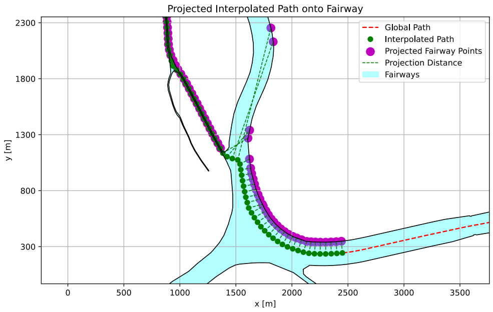


Figure 3.15: Example of a sharp left in the waterway axis leading to crossing projections.

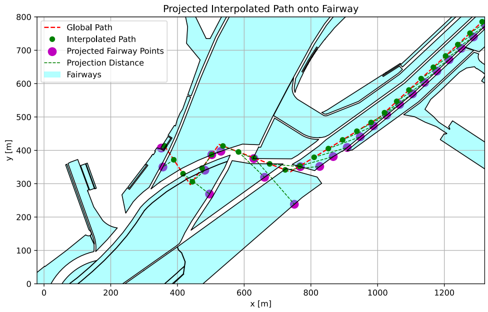


Figure 3.16: Issues with disjointed Fairway objects with ENCs produced by the Port of Rotterdam.

3.4 conclusion

This chapter presents a method for global path planning by using the waterway axis object found in Inland ENCs produced by Rijkswaterstaat [39]. Then various methods for global path refinement were evaluated using projection and three smoothing variants across ten scenarios. Feasible configurations were found in five out of ten scenarios within the tested parameter ranges.

Smoothing substantially reduced angle (Requirement 3) and step-size-projection (Requirement 4) violations, while feasibility (Requirement 1) remained the dominant failure across methods. Among the smoothers, the hybrid variant achieved the lowest overall failure rate (71.3%), but the margin over the other smoothing strategies was modest (Table 3.5). Recommended parameter ranges that generalise best are summarised in Table 3.4.
In the scenarios without any feasible results, three recurring reasons of failure were observed:

1. Width transitions: The canal abruptly changes in width.

2. Sharp kinks in the waterway axis yielding spurious projections and segment intersections.

3. Disjoint Fairway polygons in ENC data.

smoothing improves the overall path performance, but it cannot guarantee feasibility everywhere; even the best method (hybrid) yields many failures due to feasibility under challenging geometries. If a single method must be chosen, the hybrid method with a stepsize between 100 and 160, an offset between 5 and 20, smoothing window sizes (maximum and average) between 5 and 20, and performing the average smoothing method between 5 and 10 repetitions.

Chapter 4

Local Path Planning

Building on the global path planning approach presented in the previous chapter, this chapter presents the design and formulation of a local path planning approach for autonomous surface vessels (ASVs) operating in inland waterways. It introduces a model-based control framework that computes local trajectories by minimising a cost function. The cost function is constructed from multiple heuristics, including proximity to the global path, static and dynamic obstacle avoidance, and rule-based penalties for COLREG compliance violations.

The local planner used in this thesis builds upon the Biased-MPPI algorithm by Trevisan [44]. In his PhD thesis, Trevisan tested this method in a simulated environment of Amsterdam's urban canals, showing promising results. One shortcoming was mentioned, where it was stated that avoiding collision altogether was as good as impossible. This was mainly due to the chaotic nature of the environment, where operators sometimes intentionally pushed each other for navigation purposes. [43]

The primary difference between the environment used in Trevisan's work and that of the Damen Waterbus lies in both scale and vessel type: the inland waterway width can span up to 500 m and is primarily used by commercial cargo ships. This traffic tends to be less chaotic and more strictly rule-abiding, unlike the small, tightly-packed pleasure craft of Amsterdam's canals. The traffic might be less dense than in the urban canals of Amsterdam, but the vessels are larger and move at greater velocity, making them less versatile in manoeuvrability. Given these environmental differences, a hypothesis arises regarding the applicability of the Biased-MPPI method and the necessary adaptations that need to be made.

An overview of the local path planning system architecture introduced in this chapter is provided in Figure 4.1. The following sections describe the structure and function of each system component, the mathematical formulation of the cost function and its components, the heuristics employed, and the implementation and evaluation of the overall local planner design.

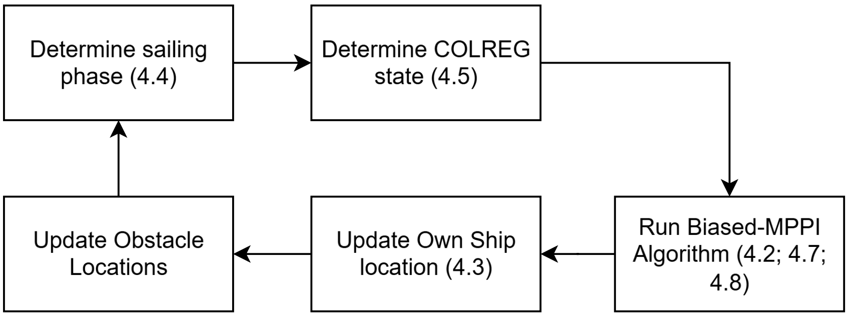

Determine sailing phase (4.4)
Determine COLREG state (4.5)
Run Biased-MPPI Algorithm (4.2; 4.7; 4.8)
Update Own Ship location (4.3)
Update Obstacle Locations

Figure 4.1: Overview of the local path planning system architecture

4.1 Local Planner Requirements

The local planner must satisfy the following functional and performance requirements to ensure safe, rule-compliant dock-to-dock navigation.

Table 4.1: Local Planner Behavioural Phases

Phase	Vel Setpoints	Expected Behaviour	Postcondition
UNDOCKING	Surge $v = 0$, Sway $v_{\text{sway}} = 1$	Hold heading and move laterally away from dock laterally	$d_{\text{start}} \geq R_{\text{undock}}$
SAILING	Surge $v = v_{\text{cruising}}$	Track global path at cruising speed	$d_{\text{rem}} \leq R_{\text{approach}}$
APPROACH	Surge $v = v_{\text{approach}}$	Slow down and prioritise direct line to dock	$d_{\text{rem}} \leq R_{\text{dock}}$
DOCKING	Surge $v = v_{\text{dock}} \ll v_{\text{approach}}$	Align heading; precise slow movement into berth	$d_{\text{rem}} \leq R_{\text{berth}}$
BERTHING	Surge $v = 0$, small sway v_{sway}	Hold position at berth with fine lateral adjustments	Terminal hold (no exit)

Table 4.2: COLREGS Encounter Handling

Encounter	Required Response
Head-On (HO)	Turn to starboard to give way
Crossing Give-Way (CR)	Allow down and Alter heading astern of target.
Overtaking Give-Way (OT)	Maintain or increase velocity. Overtake on the port side of the Target Ship.

COLREG detection criteria are Further discussed in Sections 4.5 and 4.6.8.

Table 4.3: Hard Safety & Real-Time Requirements

Requirement	Bound / Value	Purpose
Setpoint velocity	$v \leq v_{\text{set}}$	Respect vessel's setpoint speeds in each Sailing Phase.
Maximum acceleration	$\lvert a \rvert \leq a_{\max}$	Avoid dynamic infeasibility
Workspace containment	Fairway Boundaries	Never exit navigable channel
Minimum distance to traffic	$d(OS, TS) \geq 2 \cdot \text{Length}$	Maintain safe distance from other vessels: Two Shiplengths
MPPI replanning frequency	$\geq 0.1\,\text{Hz}$ (cycle $\leq 10\,\text{s}$)	Provide timely trajectory updates

4.2 Model Predictive Path Integral (MPPI) Control

The Model Predictive Path Integral (MPPI) method, first derived by Williams et al. [51], reformulates MPC as a stochastic optimal control problem by sampling a set of noisy control sequences around a nominal plan and re-optimising via cost-weighted averaging. At time t, given state $q_t \in \mathbb{R}^n$ and horizon length T, let the nominal control sequence be

$$U = [u_0, u_1, \dots, u_{T-1}] .$$

A collection of K rollouts is generated by applying Gaussian noise to the nominal control sequence

$$\tilde{u}_t^k \sim \mathbb{N}\big(u_t, \Sigma\big), \quad t = 0, \dots, T-1, \;\; k = 1, \dots, K, \tag{4.1}$$

and then applying the system dynamics to each rollout, generating trajectories

$$q_{t+1}^k = f\big(q_t^k, \tilde{u}_t^k\big), \quad q_0^k = q_t. \tag{4.2}$$

Each trajectory $Q^k = [q_0^k, \dots, q_T^k]$ gets assigned a total cost

$$S^k = \phi\big(q_T^k\big) + \sum_{t=0}^{T-1} L\big(q_t^k, \tilde{u}_t^k\big), \tag{4.3}$$

where L denotes the running cost and ϕ the terminal cost. Let w^k be importance weight for rollout k:

$$w^k = \frac{\exp\big(-\frac{1}{\lambda}(S^k - S_{\min})\big)}{\sum_{j=1}^{K} \exp\big(-\frac{1}{\lambda}(S^j - S_{\min})\big)}, \quad S_{\min} = \min_j S^j, \tag{4.4}$$

with tunable parameter $\lambda > 0$. The weighted average determines the update for the nominal sequence:

$$u_t = \sum_{k=1}^{K} w^k \tilde{u}_t^k, \quad t = 0, \dots, T-1. \tag{4.5}$$

The first control input u_0 is applied to advance the system, after which the horizon shifts forward and the procedure repeats.

This formulation requires no gradient information, accommodates non-convex and non-differentiable costs, and permits parallel rollout evaluation, enabling real-time performance on modern hardware.

Algorithm 1 illustrates the inner workings of the MPPI algorithm as it is applied in this thesis. Note that the algorithm predicts future positions of the obstacles, assuming that the obstacle remains a constant heading and velocity, and calculates the cost accordingly.

Algorithm 1 Model Predictive Path Integral (MPPI) Control with obstacle prediction.

```
Input: Current state $q_0$, nominal control sequence $U = [u_0, \dots, u_{T-1}]$
Parameters: Horizon length $T$, number of rollouts $K$, noise covariance $\Sigma$, temperature $\lambda$
Output: Control input $u_0$
for $k = 1$ to $K$ do
    $q_0^k \leftarrow q_0$
    $S^k \leftarrow 0$
    for $t = 0$ to $T-1$ do
        Sample noisy control: $\tilde{u}_t^k \sim \mathbb{N}(u_t, \Sigma)$
        Simulate ownship: $q_{t+1}^k \leftarrow f(q_t^k, \tilde{u}_t^k)$
        Update obstacle states at time $t$
        Calculate cost: $S_t^k \leftarrow L(q_t)$
    end for
    Reset obstacle states
end for
$S_{\text{min}} \leftarrow \text{min}_k S^k$
for $k = 1$ to $K$ do
    Compute weight $w_k$.
end for
for $t = 0$ to $T-1$ do
    Update control: $u_t \leftarrow \sum_{k=1}^{K} w^k \tilde{u}_t^k$
end for
return $u_0$
```

4.3 Vessel Dynamic Model

This vessel dynamic model is formulated purely in kinematic form: it updates position, heading and velocity commands directly into next-step states under the given acceleration and rate limits, without any explicit hydrodynamic-based equations. By focusing on high-level dynamics, the same local planner can be paired with different low-level models. This decoupling emphasises the flexibility and reusability of the planner across vessels, controllers, and operating conditions.

4.3.1 State, Inputs and Parameters

At time step k with step-size Δt, the vessel's state $\mathbf{s}_k$ and control inputs $\mathbf{a}_k$ are

$$\mathbf{s}_k = \left[x_k, y_k, \theta_k, v_{x,k}, v_{y,k}, \omega_k\right]^\top, \quad \mathbf{a}_k = \left[u_{f,k}, u_{\ell,k}, u_{\omega,k}\right]^\top \tag{4.6}$$

where

- (x, y) is the world-frame position,
- θ is the heading,
- (v_x, v_y) are world-frame velocity components,
- ω is the yaw-rate,
- $u_f \in [-v_{\text{surge\_max}}, v_{\text{surge\_max}}]$ is the body-frame surge velocity control inputs.
- $u_\ell \in [-v_{\text{sway\_max}}, v_{\text{sway\_max}}]$ is the body-frame sway velocity control input.
- $u_\omega \in \left[-\omega_{\text{max}}, \omega_{\text{max}}\right]$ is the control input for the yaw-rate.

The key parameters are shown in Table 4.4.

Symbol	Meaning and units
Δt	time step [s]
L	vessel length [m]
B	vessel breadth [m]
T	vessel draught [m]
$v_{\max}$	maximum speed [m/s]
$\omega_{\max}$	maximum turning rate [rad/s]
$a_{\max}$	maximum linear acceleration [m/s^2]
$\alpha_{\max}$	maximum rotational acceleration [rad/s^2]
$r_{\min}$	minimum turning radius [m]

Table 4.4: High-level vessel parameters.

4.3.2 Kinematic Equations

The vessel motion is modelled as a modified kinematic bicycle model [37] augmented with an additional degree of freedom to represent drift. This allows the model to explicitly represent sideways drift at low speeds, while simplifying to standard bicycle kinematics when surge velocity exceeds a threshold. The discrete-time update is expressed in state–space form as

$$\mathbf{s}_{k+1} = \mathbf{A}(\mathbf{s}_k)\,\mathbf{s}_k + \mathbf{B}(\mathbf{s}_k)\,\mathbf{u}_k \tag{4.7}$$

where $\mathbf{u}_k = [u_{f,k},\, u_{\ell,k},\, u_{\omega,k}]^\top$ is the velocity command computed by the MPPI algorithm. These commands are drawn from a Gaussian distribution centred at the previous input, each with its standard deviation. Since this sampling does not ensure dynamic feasibility, the surge and sway velocity commands are subsequently clamped to the vessel's maximum acceleration limits. To reach this goal, first, the current body-frame velocities are computed since these are not given directly by the state vector $\mathbf{s}_k$.

$$v_{b,x} = v_{x,k}\cos\theta_k + v_{y,k}\sin\theta_k \tag{4.8}$$

$$v_{b,y} = -v_{x,k}\sin\theta_k + v_{y,k}\cos\theta_k \tag{4.9}$$

Then the accelerations are computed, given the current velocities and command velocities. These are then clamped to accelerations, as shown in equations, and are purposefully clamped to the maximum acceleration of the vessel to ensure dynamic feasibility.

$$a_f = \operatorname{clamp}\left(\frac{u_f - v_{b,x}}{\Delta t},\, -a_{\max},\, a_{\max}\right) \tag{4.10}$$

$$a_\ell = \operatorname{clamp}\left(\frac{u_\ell - v_{b,y}}{\Delta t},\, -a_{\max},\, a_{\max}\right) \tag{4.11}$$

$$\alpha = \operatorname{clamp}\left(\frac{u_\omega - \omega_k}{\Delta t},\, -\alpha_{\max},\, \alpha_{\max}\right) \tag{4.12}$$

After establishing the dynamically feasible accelerations corresponding to the MPPI-generated control inputs, the commands are updated.

$$u'_f = v_{b,x} + a_f\,\Delta t \tag{4.13}$$

$$u'_\ell = \mathbf{1}_{\{u'_f \leq 2\}}\,(v_{b,y} + a_\ell\,\Delta t) \tag{4.14}$$

$$u'_\omega = \omega_k + \alpha\,\Delta t \tag{4.15}$$

At surge velocities of above 2 m/s, the vessel's hydrodynamic forces greatly outweigh any lateral thrust. As a result, sideways steering is not very effective. Therefore, the kinematic model is simplified by setting $u'_l = 0$ (as per Equation 4.14) once this threshold is exceeded, essentially simplifying the kinematics to a bicycle model. To determine the next state, first, the velocities are determined as such:

$$\omega_{k+1} = u'_\omega \tag{4.16}$$

$$v_{x,k+1} = u'_f \cos\theta_{k+1} - u'_\ell \sin\theta_{k+1} \tag{4.17}$$

$$v_{y,k+1} = u'_f \sin\theta_{k+1} + u'_\ell \cos\theta_{k+1} \tag{4.18}$$

Lastly, the position and heading are computed by:

$$\theta_{k+1} = \theta_k + u'_\omega \Delta t \tag{4.19}$$

$$x_{k+1} = x_k + v_{x,k+1}\,\Delta t \tag{4.20}$$

$$y_{k+1} = y_k + v_{y,k+1}\,\Delta t \tag{4.21}$$

4.4 Sailing Phase Switcher

To perform sufficient Dock-to-Dock behaviour, we employ a sailing phase switcher that adapts the cost function throughout the mission. Between every MPPI iteration, it computes the distance between the vessel's current position and the start dock and the remaining distance from the vessel's current position to the end dock:

$$d_{\text{start}} = \left\| (x,y) - (x_0,y_0) \right\| \tag{4.22}$$

$$d_{\text{rem}} = L_{\text{path}} - (x_g, y_g), \tag{4.23}$$

where (x_0, y_0) is the path start and (x_g, y_g) the path goal, L_{path} is the total global path length and $s(x,y)$ is the along-track distance from the start. Given the distance to the start d_{start} and the remaining distance d_{rem}, one of five sailing phases is selected using Algorithm 2.

Algorithm 2 Sailing Phase Switcher

```
Compute $d_{\text{start}}$, $d_{\text{rem}}$
if $d_{\text{start}} < R_{\text{undock}}$ then
    phase $\leftarrow$ UNDOCKING
else if $d_{\text{rem}} > R_{\text{approach}}$ then
    phase $\leftarrow$ SAILING
else if $d_{\text{rem}} > R_{\text{dock}}$ then
    phase $\leftarrow$ APPROACH
else if $d_{\text{rem}} > R_{\text{berth}}$ then
    phase $\leftarrow$ DOCKING
else
    phase $\leftarrow$ BERTHING
end if
return phase
```

The following behaviour is expected for each sailing phase

- UNDOCKING: In this initial phase, the vessel keeps its heading while slowly moving away from the dock laterally. $d_{\text{start}} \geq R_{\text{undock}}$, we switch to the undocking phase is exited.

- SAILING: Here, the vessel tracks the global path at cruising speed. Once within R_{approach} of the goal, the planner enters the Approach phase.
- APPROACH: In the Approach, the velocity setpoint is set significantly lower than the cruising speed to approach the dock safely. Euclidean distance is prioritised over the along-track distance of the global path. Upon crossing into R_{dock}, the DOCKING phase begins.
- DOCKING: During Docking, the surge speed is limited to a low setpoint, and the vessel aims to align its heading with the dock. When $d_{\text{rem}} \leq R_{\text{berth}}$, we transition to Berthing.
- BERTHING: During the final Berthing phase, the vessel's goal is to hold its position at the berth with minimal movement.

Figure 4.2 shows the state-machine between the sailing phase switches, including switches between COLREG states (which will be explained in more depth in paragraph 4.5. Within the scope of this thesis, the sailing phase is the only phase where COLREG-specific behaviour is assumed.

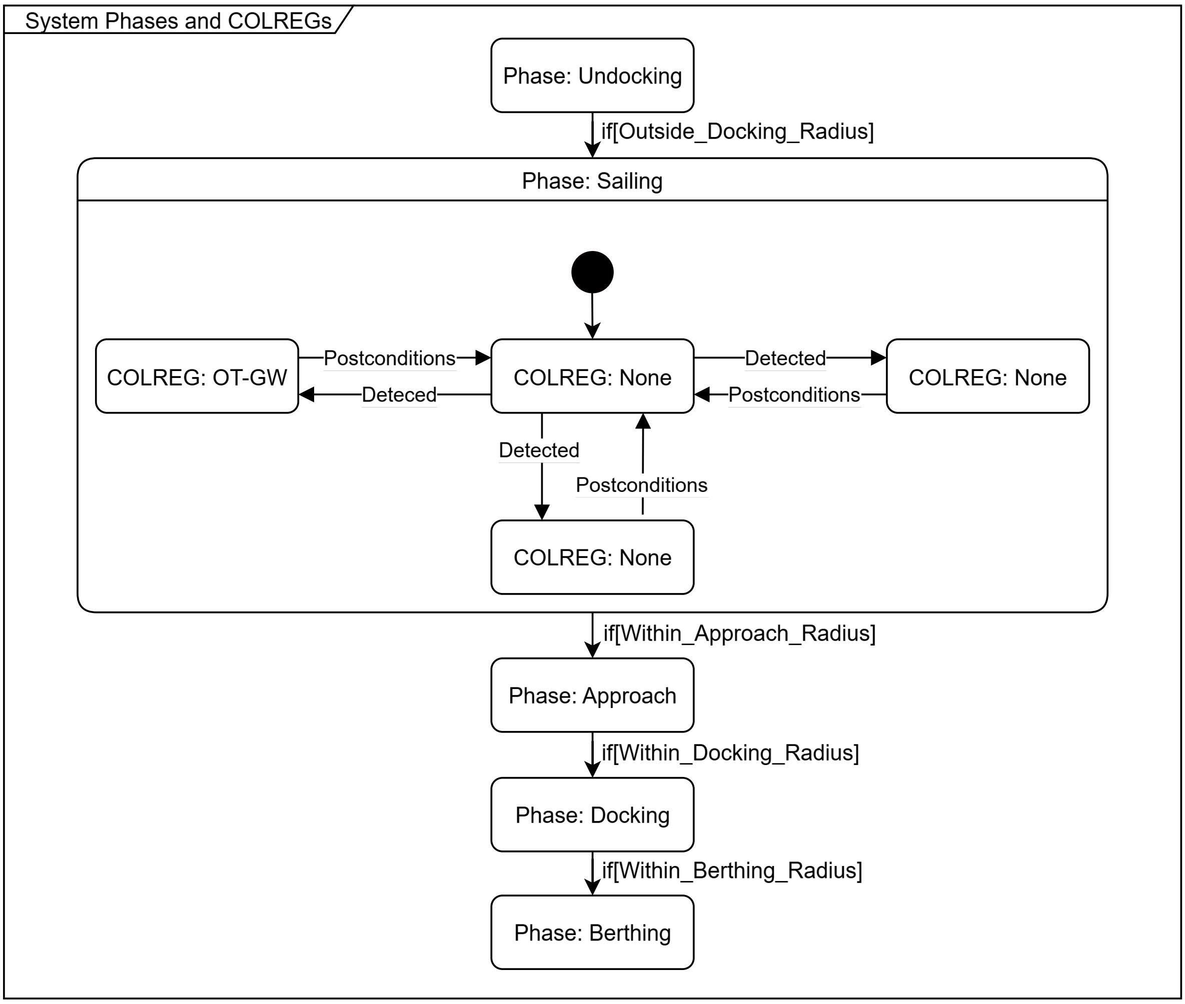


Figure 4.2: Overview of the various sailing phase switches and COLREG state switches.

4.5 COLREG state-switcher

Some of the issues identified in the literature studies included the limited research on local path planning for COLREG-compliance in inland situations and the lack of standardisation regarding the definitions of the COLREG states. DNV [36], a leading classification society, is actively working towards a standard to test path planning methods for COLREG-compliance. They also propose a system for COLREG state definitions, as proposed by Hagen [11], which is used in this work to define COLREG states.

To define the COLREG states, two angles are determined by using the Line of Sight (LOS) as shown in Figure 4.3. The LOS is defined as the line from the centre point of the Own Ship to the centre point of the Target Ship. β is defined as the angle between the Own Ship and the LOS, and α is defined as the angle between the Target Ship and the LOS.

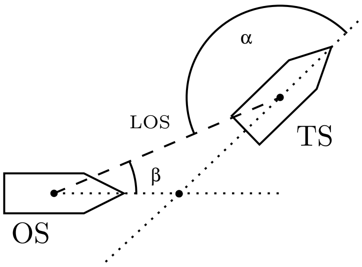


Figure 4.3: The COLREG classification model mainly utilises angles α and β, defined respectively as the angle between the Target Ship's longitudinal axis and the LOS and the Own Ship's longitudinal axis and the LOS.

The angles α and β are used in combination to classify the COLREG state when in range of a set of precondition thresholds. The definitions of these preconditions are illustrated in Figure 4.4. Appendix B shows a complete list of pre- and postconditions and the corresponding numeric values of these preconditions.

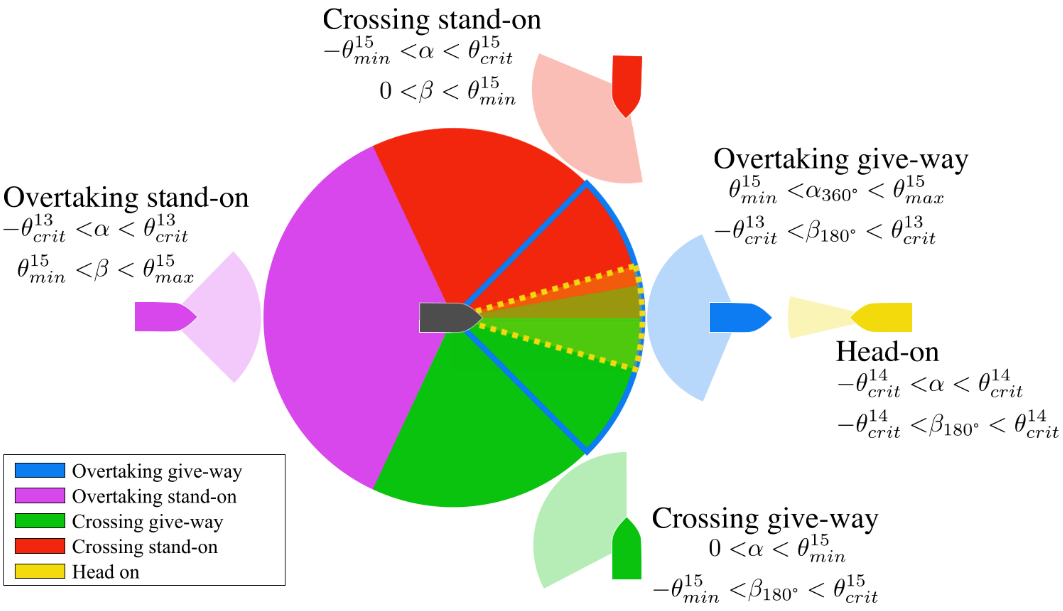

Crossing stand-on
$-\theta^{15}_{min} < \alpha < \theta^{15}_{crit}$
$0 < \beta < \theta^{15}_{min}$
Overtaking give-way
$\theta^{15}_{min} < \alpha_{360^\circ} < \theta^{15}_{max}$
$-\theta^{13}_{crit} < \beta_{180^\circ} < \theta^{13}_{crit}$
Overtaking stand-on
$-\theta^{13}_{crit} < \alpha < \theta^{13}_{crit}$
$\theta^{15}_{min} < \beta < \theta^{15}_{max}$
Head-on
$-\theta^{14}_{crit} < \alpha < \theta^{14}_{crit}$
$-\theta^{14}_{crit} < \beta_{180^\circ} < \theta^{14}_{crit}$
Crossing give-way
$0 < \alpha < \theta^{15}_{min}$
$-\theta^{15}_{min} < \beta_{180^\circ} < \theta^{15}_{crit}$
🟦 Overtaking give-way
🟪 Overtaking stand-on
🟩 Crossing give-way
🟥 Crossing stand-on
🟨 Head on

Figure 4.4: Hagen's COLREG classification model from the Own Ship's viewpoint. The mappings $\alpha_{360} : [-180^\circ, 180^\circ) \rightarrow [0^\circ, 360^\circ)$ and $\beta_{180} : [0^\circ, 360^\circ) \rightarrow [-180^\circ, 180^\circ)$ are used to display the symmetry in the situation classification model. [11]

Hagen [11] distinguishes five various scenarios:

- Crossing Give-Way (CR-GW)
- Crossing Stay-On (CR-SO)

- Overtaking Give-Way (OT-GW)
- Overtaking Stay-On (OT-SO)
- Head-On (HO)

In this thesis, the Stay-On scenarios for Crossing and Overtaking are classified, but no adaptive behaviour is programmed; these scenarios require the Own Ship to stay on its current trajectory. A policy for these scenarios is recommended for future work. However, they are neglected within this work.

Lastly, the distance between the Own Ship and the Target Ship must be addressed, as otherwise specific heading configurations could cause the system to think it is in a traffic scenario while the Target Ship is very far away from the Own Ship. Therefore each COLREG is assigned a radius (R_{CR_GW}, R_{OT_GW}, R_{HO}), when the distance between the Own Ship and the Target Ship is smaller than this radius and the other preconditions are met, the COLREG sate classifier returns a positive classification for the respective COLREG state.

Figure 4.5 illustrates the high-level system architecture of the COLREG state switcher. The system begins in a default Sailing state, where no specific traffic interaction is assumed. During this state, a COLREG state check is performed at every iteration. If the preconditions of a particular COLREG scenario, as defined by Hagen's model [11], are satisfied, the system transitions to the corresponding COLREG state.

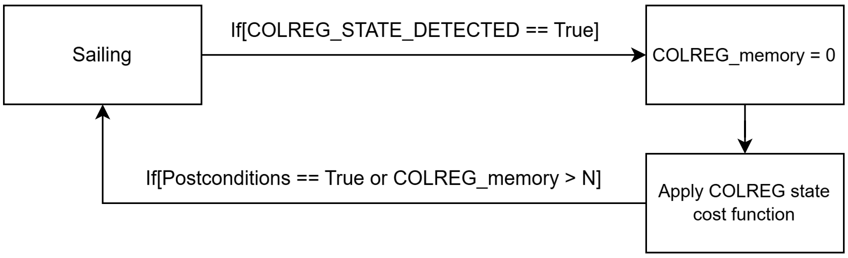

Sailing
If[COLREG_STATE_DETECTED == True]
COLREG_memory = 0
If[Postconditions == True or COLREG_memory > N]
Apply COLREG state cost function

Figure 4.5: High-level State Machine Diagram for COLREG state switcher, showcasing how the COLREG memory is applied.

Once a COLREG state is active, the cost function is modified to encourage behaviours that align with that specific rule and remains in effect until the postconditions are met. To ensure stability and prevent rapid or unintended transitions, the COLREG state detector is temporarily disabled for a predefined number of iterations (N in Figure 4.5) after a transition. At the end of every iteration, the COLREG_memory counter is incremented by one. Once N iterations have passed, the COLREG state detector is enabled.

This mechanism prevents premature exits from a COLREG state due to momentary fluctuations in perception or vessel positioning. For instance, during an overtaking manoeuvre, the Target Ship will move out of the defined overtaking zone while manoeuvring. However, it is still necessary to safely complete the manoeuvre before reverting to the Sailing state until the set postconditions are met. This built-in memory ensures such transitions are handled safely. In-depth definitions on the postconditions are found in Paragraph 4.6.8.

The pseudo-algorithm 3 summarises the COLREG state switcher logic on a slightly deeper level compared to Figure 4.5. The COLREG state-switcher operates outside of the MPPI passes and influences the MPPI cost functions directly. Therefore, future states inside of the MPPI planner still assume the current COLREG-encounter even if that future state does not comply with Hagen's system [11]. This design choice was made as otherwise the MPPI planner was able to quickly determine which state would result in a lower cost and therefore avoid entering the COLREG state altogether, which causes unwanted behaviour such as turning back. Forcing the entire MPPI loop to assume the same COLREG was found to lead to more wanted behaviour.

Algorithm 3 COLREG State Switcher

```
Input: Current vessel state $\mathbf{s}_k$, Target Ship states
Output: Updated COLREG state current_state
current_state ← 'NONE'
COLREG_memory ← 0
Define detection radii $r_{\text{OT-GW}}, r_{\text{CR-GW}}, r_{\text{HO}}$
for $k = 1$ to $N$ do
    Compute CPA (DCPA, TCPA) with all targets
    if current_state = 'CR-GW' and Postcondition for CR-GW holds then
        current_state ← 'NONE'
        COLREG_memory ← 0
    else if current_state = 'OT-GW' and Postcondition for OT-GW holds then
        current_state ← 'NONE'
        COLREG_memory ← 0
    else if current_state = 'HO' and Postcondition for HO holds then
        current_state ← 'NONE'
        COLREG_memory ← 0
    end if
    if COLREG_memory > 0 then
        COLREG_memory ← COLREG_memory − 1
        return current_state
    end if
    new_state ← classify_current_encounter($\mathbf{s}_k$, targets)
    if new_state ≠ current_state and new_state ≠ 'NONE' then
        current_state ← new_state
        COLREG_memory ← 0
    end if
    return current_state
end for
```

4.6 Cost Function Formulation

The cost function J used in the MPPI planner guides the vessel by penalising undesirable behaviour and promoting adherence to navigation rules and safe operational practices. It comprises multiple components, each weighted according to the sailing phase (e.g., UNDOCKING, SAILING, APPROACH, DOCKING, BERTHING):

$$\begin{aligned} J(\mathbf{s}) =& w_{\text{valid}} J_{\text{valid}}(\mathbf{s}) + w_{\text{dist}} J_{\text{dist\_to\_goal}}(\mathbf{s}) + w_{\text{vel}} J_{\text{velocity}}(\mathbf{s}) + w_{\text{sway\_vel}} J_{\text{sway\_velocity}}(\mathbf{s}) \\ &+ w_{\text{heading}} J_{\text{heading}}(\mathbf{s}) + w_{\text{yaw}} J_{\text{yaw\_rate}}(\mathbf{s}) + J_{\text{global\_path}}(\mathbf{s}) + J_{\text{colreg}}(\mathbf{s}) \end{aligned} \tag{4.24}$$

Note that the costs $J_{\text{global\_path}}$ and J_{colreg} do not have a weight assigned, as these functions are dependent on multiple factors, which are weighed internally. A comprehensive list of all weights applied in each sailing phase is found in Appendix D.

The remainder of this section explains each of the cost functions and their phase-dependent tuning.

4.6.1 Workspace Validity

Penalises states outside of the valid workspace:

$$J_{\text{valid}}(\mathbf{s}) = \begin{cases} 1, & \text{if outside workspace} \\ 0, & \text{otherwise} \end{cases} \tag{4.25}$$

The weight associated with this cost function is set relatively high to ensure the workspace boundary is not exceeded. In the context of this work, the workspace is set to be the Fairway object. All berthing locations are programmed with a set position and heading.

4.6.2 Euclidean Distance to Goal

Penalises the normalised distance from the vessel to the navigation goal:

$$J_{\text{dist\_to\_goal}}(\mathbf{s}) = \frac{\|\mathbf{s}_{xy} - \mathbf{g}\|}{d_{\text{init}}} \tag{4.26}$$

Where $\mathbf{s}_{xy}$ is the current position, $\mathbf{g}$ is the goal position, and d_{init} is the initial distance to the goal from the undocking position.

This cost function is only applied once the vessel is near the dock, such as in the APPROACH, DOCKING and BERTHING phases. In earlier phases, such as SAILING, the Along Track Distance (ATD) is used instead of Euclidean distance to encourage forward progression along the planned path.

4.6.3 Set Velocity Deviation

Penalises deviation from a surge velocity setpoint:

$$J_{\text{velocity}}(\mathbf{s}) = \frac{|v_{\text{fwd}} - v_{\text{fwd\_set}}|}{\max(v_{\text{fwd\_set}}, 1)} \tag{4.27}$$

where v_{fwd} and v_{set} are current and setpoint surge velocities. The $\max(v_{\text{fwd\_set}}, 1)$ term applies normalisation. The surge velocity setpoint varies based on the sailing phase to promote different speeds based on dock proximity.

4.6.4 Set sway velocity Deviation

Penalizes deviation from a sway velocity setpoint:

$$J_{\text{sway\_velocity}}(\mathbf{s}) = \frac{|v_{\text{lat}} - v_{\text{sway\_set}}|}{\max(v_{\text{sway\_set}}, 1)} \tag{4.28}$$

where v_{lat} and $v_{\text{sway\_set}}$ are current and setpoint sway velocities. The $\max(v_{\text{sway\_set}}, 1)$ term applies normalisation. The sway setpoint is set to zero in most cases except for the undocking and docking state. In these cases the sway velocity setpoint is set to promote moving away or towards the dock in a safe, precise manner.

4.6.5 Heading Adjustment

Penalizes heading deviation, fading based on the distance to goal:

$$J_{\text{heading}}(\mathbf{s}) = \exp\left(-\frac{1}{2}\left(\frac{d}{\sigma}\right)^2\right) \cdot \frac{|\theta - \theta_{\text{desired}}|}{\pi} \tag{4.29}$$

where d is distance to goal, σ controls fading, θ is current heading, and θ_{desired} is the desired heading. This heading cost is designed to encourage the vessel to gradually align its heading with the dock's orientation as it nears the goal. The fading component causes the cost to increase gradually within a set radius, promoting the vessel into making slight heading corrections when approaching the dock. When closer to the dock the full heading penalty is applied to get into the proper position.

4.6.6 Yaw Rate Cost

Penalizes excessive yaw rate:

$$J_{\text{yaw\_rate}}(\mathbf{s}) = \frac{|\omega|}{\omega_{\max}} \tag{4.30}$$

where ω is yaw rate, and $\omega_{\max}$ is the maximum allowed yaw rate possible according to the vessel dynamics. This cost promotes smoother trajectories by discouraging sharp or frequent changes in heading.

4.6.7 Global Path Tracking

Encourages adherence to the global path, considering lateral and along-track errors:

$$J_{\text{global\_path}}(\mathbf{s}) = w_{\text{xtr}} \frac{\text{XTR}}{R_{\text{xtr}}} + w_{\text{atd}} \left(1 - \frac{\text{ATD}}{L_{\text{total}}}\right) \tag{4.31}$$

where XTR is the cross-track error at the current time step, ATD is the along-track distance traveled, and L_{total} is the total global path length, R_{xtr} is a normalisation radius that defines how far off the path is considered fully penalised. The value of R_{xtr} is set manually dependant on the average widths of the Fairway object.

The XTR is determined by finding the intersection between the line perpendicular to the vessel's heading and the global path and determining the corresponding distance between the intersection and the current position of the vessel.

Increasing w_{xtr} stimulates stricter global path following and therefore causes the vessel to move more towards the starboard shore. Increasing w_{adt} promotes forward progression along the planned global path.

Each sailing phase balances along-track distance (ATD, Eq. 4.31) against Euclidean distance to goal (Eq. 4.26) according to its navigational objective. For example, in the SAILING phase only the ATD cost is active since a Euclidean distance penalty alone could lead to unwanted behaviour when the straight-line distance intersects Fairway boundaries. In contrast, during DOCKING only the Euclidean distance cost is applied, driving the vessel straight toward the goal.

Noteworthy is that by using the total path length as a normalisation factor, the weight of Along track distance when approaching the goal. This has been accounted for by the weights of the APPROACH phase. In the DOCKING and BERTHING phase, the global path cost is fully neglected to promote proper near-dock manoeuvring. Global Path cost is not applied for the UNDOCKING phase to promote lateral movement while undocking. The full list of weights for each sailing phase is shown in D

4.6.8 COLREGS-Based Cost Components

During encounters governed by COLREGS rules, additional cost terms are introduced to enforce compliant behaviour. These costs apply selectively depending on the detected encounter type (e.g., crossing, overtaking, head-on). The total COLREG-related cost is:

$$J_{\text{colreg}}(\mathbf{s}) = J_{\text{colreg\_vo}}(\mathbf{s}) + J_{\text{colreg\_heading}}(\mathbf{s}) + J_{\text{colreg\_brake}}(\mathbf{s}) \tag{4.32}$$

where each term activates conditionally depending on the current encounter state.

Velocity Obstacle (VO) Cost

The velocity obstacle [9] (VO) of an obstacle B for our vessel A is the set of all velocities of A that would lead to a collision with B at some future time (assuming A and B continue at their current velocities). To create the velocity obstacle, one first constructs the collision cone $\mathrm{CC}_{A,B}$ by expanding B by the sum of the radii of A and B (plus a safety margin), then forming the planar sector with apex at A whose sides are the two lines tangent to this enlarged obstacle. The VO is simply that cone with its origin in the centre of A shifted by B's velocity:

$$\mathrm{VO}_B \;=\; \{\mathbf{v_A} \mid \mathbf{v_A} - \mathbf{v}_B \in \mathrm{CC}_{A,B}\}.$$

Any absolute velocity $\mathbf{v}_A$ lying inside this cone will intersect B if both continue straight.

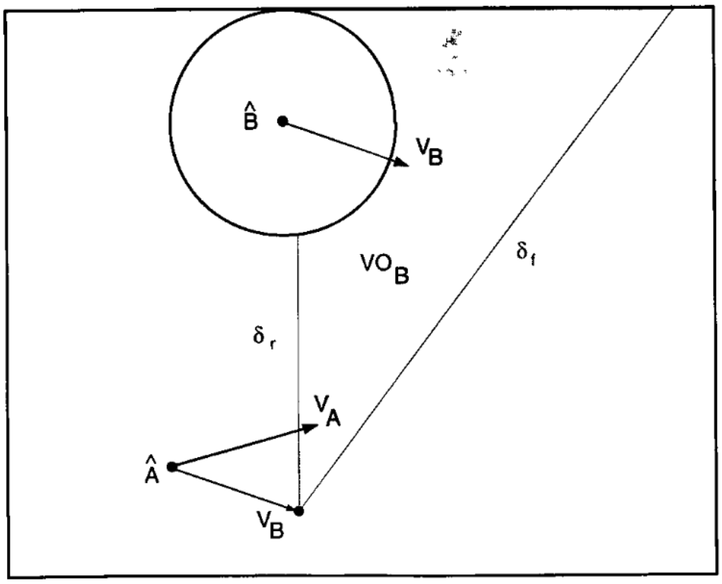


Figure 4.6: Illustration of the Velocity Obstacle VO_B for vessel A avoiding vessel B by Fiorini et al. [9]. The cone spanned by δ_r and δ_f shows all relative velocity directions that would intersect with obstacle B.

In the cost function the VO-penalty weight w_{vo} is applied to any sampled velocity that lies in VO_B:

$$J_{\mathrm{colreg\_vo}}(\mathbf{s}) = w_{\mathrm{vo}} \cdot \mathbb{I}\big[\mathbf{v}_A \in \mathrm{VO}_B\big] \tag{4.33}$$

Here, $\mathbb{I}_{\mathrm{VO}} = 1$ if the trajectory lies within a VO cone (defined by relative position and velocity with respect to the obstacle), and 0 otherwise. The weight w_{vo} is encounter-type dependent.

Heading Adjustment Cost (COLREG)

Encourages the vessel to steer in a direction compliant with the COLREGS scenario:

$$J_{\mathrm{colreg\_heading}}(\mathbf{s}) = w_{\mathrm{colreg\_hdg}} \cdot \frac{|\theta - \theta_{\mathrm{t}}|}{\pi} \tag{4.34}$$

where θ is the vessel's current heading and θ_{t} is a dynamically computed target heading based on the COLREG encounter.

Brake Cost (CR-GW only)

Encourages slowing down during crossing give-way encounters:

$$J_{\text{colreg\_brake}}(\mathbf{s}) = w_{\text{brake}} \cdot \frac{v_{\text{own}}}{v_{\text{set}}} \tag{4.35}$$

where v_{own} is the vessel's speed and v_{set} is the colreg-dependant velocity setpoint. Note that in this case w_{vel} is set to zero.

COLREG State-Specific Behaviour

The parameters in the equations 4.33–4.35 depend on the type of encounter detected by the COLREG state-switcher, particularly the setpoint of the heading. Therefore, the weights and target heading are set based on the COLREG state, shown by Table 4.5. The heading setpoint is tailored to the specific scenario and will be described in detail for each case below.

Component	Symbol	CR-GW	OT-GW	HO	Effect
VO-cone penalty	w_{vo}	1	0.5	0.25	Penalises trajectories inside the VO cone (Eq. 4.33).
Heading setpoint	$w_{\text{colreg\_hdg}}$	5	10	1	Aims to align heading to an adaptive setpoint (Eq. 4.34).
Brake cost	w_{brake}	0.01	—	—	Encourages deceleration in the crossing give-way scenario (Eq. 4.35).

Table 4.5: Global-path & COLREGS cost components.

When a Crossing Give-Way (CR-GW) scenario is reached, the Own Ship is required to give way as per the COLREGs. This behaviour is supported by braking as per equation 4.35 and aiming to align the Own Ship's heading with a point two ship lengths behind the Target Ship as illustrated in Figure 4.7. Once the vessel moves into an area beyond this point, such as in Figure 4.9, defined by line k, the give way manoeuvre is completed and the COLREG state-switcher is set back to None.

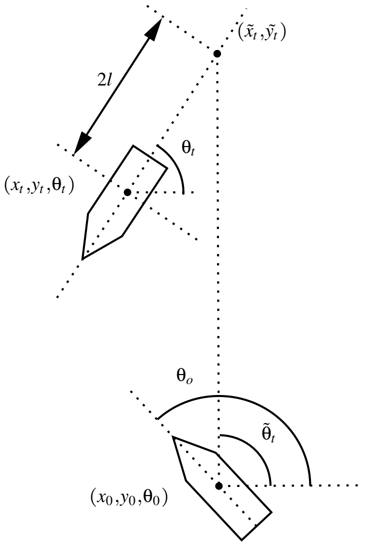


Figure 4.7: CR-GW: Way Heading setpoint set to align with the point two shiplengths behind the TS' stern.

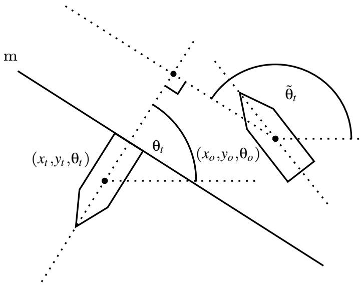


Figure 4.8: CR-GW: Once positioned behind the TS, change OS heading setpoint perpendicular to current TS heading.

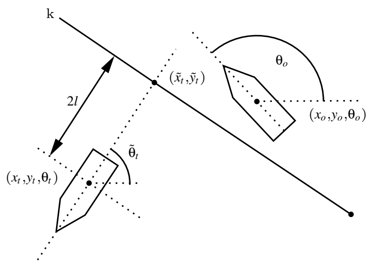


Figure 4.9: CR-GW: Post-condition with a little more text to yap.

When an Overtaking Give-Way (OT-GW) scenario is reached, the Own Ship is allowed to overtake the Target Ship by altering its course. The Target Ship is to be overtaken on its port side as per the COLREGs. This behaviour is supported by aiming to align the Own Ship's heading to a point one ship length to the port side of the Target Ship, shown in 4.10. Once the Own Ship's position is more than one shiplength to the port side of the Target Ship, such as in Figure 4.11, a different policy is adopted where the Own Ship's heading is set to match the heading of the Target Ship. When the Own Ship is two shiplengths or more ahead of the Target Ship, such as in Figure 4.12, the postconditions of this scenario are met.

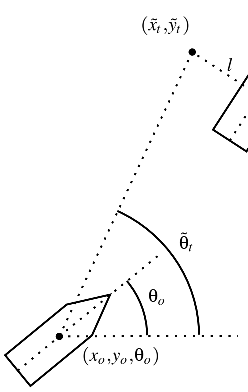


Figure 4.10: OT-GW: Heading setpoint to align with the point one shiplength to the portside of TS.

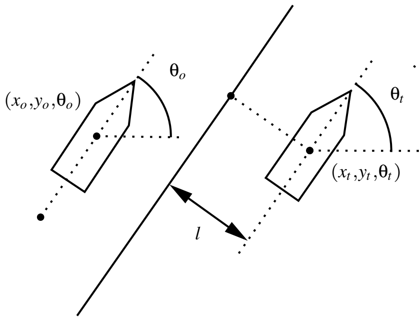


Figure 4.11: OT-GW: Setpoint when more than one shiplength away from TS portside.

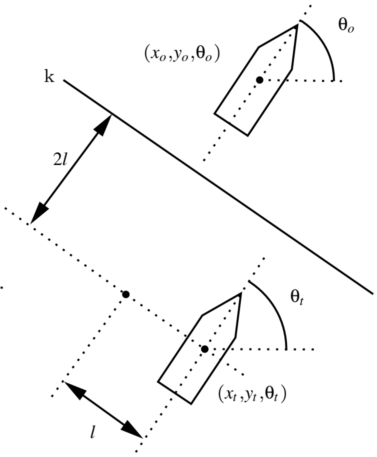


Figure 4.12: OT-GW: Postcondition.

When a Head-On (HO) scenario is reached, the Own Ship is required to give way to the Target Ship by moving its heading towards starboard. This behaviour is supported by aiming to align the Own Ship's heading to a point one ship length to the port side of the Target Ship as shown in Figure 4.13. Once the vessel is beyond the heading setpoint as per Figure 4.14, the postconditions are met.

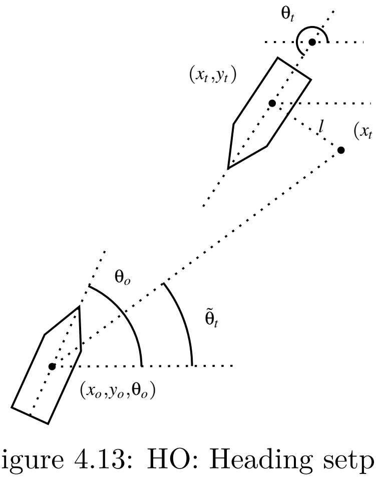


Figure 4.13: HO: Heading setpoint.

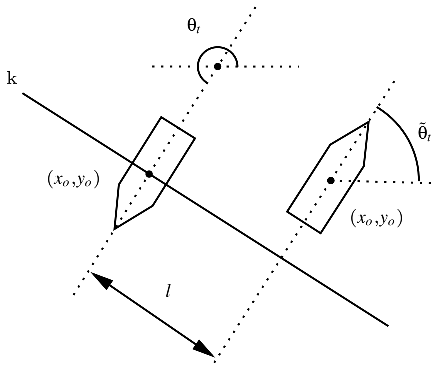


Figure 4.14: HO: Postcondition.

4.7 Biased-MPPI

Biased-MPPI [44] was developed to address one key shortcoming of classical MPPI: its lack of adaptability to sudden changes to the environment. This is due to the nature of the MPPI algorithm 4.2, where noise with a set standard deviation is sampled around each control input. Therefore, each rollout will have control input values near the corresponding nominal trajectory. If the system has to make an emergency stop of an object that suddenly appears in front of it, the trajectory with the lowest velocity set points will win each iteration. However, that trajectory will most probably not brake as fast as the system dynamics allow due to the Gaussian sampling around higher velocities. This lack of adaptability increases the risk of sudden environmental changes.

Biased-MPPI tackles this problem by adding ancillary controllers to the sampling distribution. In the example of the sudden appearance of an object, sampling the null input as an ancillary controller (in addition to the Gaussian sampled rollouts) enables the system to perform an emergency stop at the limit of its dynamic capabilities.

In the case of a small vessel navigating a highly dynamic environment, such as the urban canals of Amsterdam, Trevisan designs four Ancillary controllers to help escape local minima and improve sampling efficiency:

- Go-slow: A sequence of input commanding a small thrust.
- Go-Fast: A sequence of input commanding a large thrust.
- Braking: Give a zero velocity reference.
- Go-to-Goal: Compute a velocity reference that moves the vessel towards its corresponding goal at each time step of the planning horizon.

Adding these Ancillary controllers significantly decreased the travelling distance and the number of collisions in multi-vessel interactions in a simulation of urban canals in Amsterdam.

4.8 Leveraging Biased-MPPI for inland waterway local planning

To address the differences in the Biased-MPPI method proposed by Trevisan in urban canals and the case of the Damen Waterbus, this thesis proposes an additional set of Ancillary controllers:

- Move Forward: Set the surge command velocity to the surge velocity setpoint of the current sailing phase.
- Move Laterally: Includes two ancillary controllers which assign a constant sway velocity towards port and starboard.
- Rotate Set the rotational command velocity towards the port or starboard side based on the error between the current heading and the heading setpoint.
- Go-to-Goal-and-brake: Compute a velocity reference using a PID controller that adapts the vessel's heading towards the set point. When the set heading is reached, move towards the goal at the set velocity. When nearing the goal, brake fully to come to a standstill.
- Follow-Global-Path: Compute a velocity reference using a PID controller that aims to minimise the cross-track error of the global path, with progress measured along-track rather than by Euclidean distance.

The Full description, including values for all control gains, is found in Appendix C. The remainder of this section provides an in-depth mathematical formulation of the ”Go-to-Goal-and-brake” and ”Follow-Global-Path” ancillary controllers.

4.8.1 Mathematical Formulation of the New Ancillary Controllers

Let the vessel state at time step t be given by:

$$\mathbf{s}(t) = \left[x(t), y(t), \theta(t), v_x(t), v_y(t), \omega(t)\right]^\top, \tag{4.36}$$

where $\theta(t)$ is the heading, v_x, v_y are body-frame velocities, and ω is angular velocity. The surge velocity in the body frame is computed as:

$$v_f(t) = v_x(t)\cos\theta(t) + v_y(t)\sin\theta(t). \tag{4.37}$$

All controllers run at a discrete time step Δt and use the following PD gains and thresholds: K_p^l, K_d^l for sway control; K_p^ω, K_d^ω for yaw control; and δ_{lat}, θ_{th} for deadbanding. A maximum deceleration $a_{\max}$ is used for braking behaviour.

4.8.2 Go-to-Goal-and-Brake Prior

Given a static goal position $\mathbf{p}_g = [x_g, y_g]^\top$ and current vessel state $\mathbf{s}(t)$, define the goal vector $\mathbf{e}$ and Euclidean distance d to goal as:

$$\mathbf{e}(t) = \mathbf{p}_g - \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \quad d(t) = \|\mathbf{e}(t)\|_2 + \varepsilon \tag{4.38}$$

The desired heading θ_d and current heading error e_θ are:

$$\theta_d(t) = \text{atan2}(e_y(t), e_x(t)) \tag{4.39}$$

$$e_\theta(t) = \big(\theta_d(t) - \theta(t) + \pi\big) \bmod 2\pi - \pi \tag{4.40}$$

Where ε is a small positive constant to avoid numerical division by zero. To prevent abrupt motion before alignment, a heading deadband δ_θ is applied. If $|e_\theta(t)| > \delta_\theta$, no surge or sway is commanded:

$$u_f(t) = 0, \quad u_\ell(t) = 0 \tag{4.41}$$

$$u_\omega(t) = K_p^\omega e_\theta(t) + K_d^\omega \dot{e}_\theta(t) \tag{4.42}$$

This promotes behaviour where the vessel's heading is aligned with the goal before moving forward. Once aligned, the surge velocity is applied based on the distance to goal. Then surge velocity command $v_{\text{cmd}}(t)$ is determined as:

$$v_{\text{cmd}}(t) = \begin{cases} u_{f,\max}, & \frac{v_f^2}{2a_{\max}} < d(t) \cdot M_b \\ \text{clamp}\left(\sqrt{2a_{\max} d(t)}, 0, u_{f,\max}\right), & \text{otherwise} \end{cases} \tag{4.43}$$

Equation 4.43 is designed to set the surge velocity at its maximum allowed value (given the sailing state) as long as the distance (times a margin factor M_b) is greater than the minimum braking distance. This is designed such that the vessel can come to a full stop at its final position as fast as theoretically possible. The velocity is clamped due to the added margin in the velocity command conditions. Next, the command is further clamped to prevent single-step overshoot:

$$u_f(t) = \text{clamp}\left(v_{\text{cmd}}(t), 0, \frac{d(t)}{\Delta t}\right) \tag{4.44}$$

For sway control, define the cross-track error c, the and its derivative:

$$c(t) = \sin(e_\theta(t)) \cdot d(t) \tag{4.45}$$

$$\dot{c}(t) = \frac{c(t) - c(t-1)}{\Delta t} \tag{4.46}$$

The lateral offset c is minimised to promote the vessel to approach the dock in a straight line. A deadband δ_{lat} is applied to stop any lateral movement when sufficiently on course, otherwise apply a PD controller clamped to comply with the vessel dynamics:

$$u_\ell(t) = \begin{cases} 0, & \text{if } |c(t)| < \delta_{\text{lat}} \\ \text{clamp}\left(K_p^\ell c(t) + K_d^\ell \dot{c}(t), -u_{\ell,\max}, u_{\ell,\max}\right), & \text{otherwise} \end{cases} \tag{4.47}$$

Finally, during tracking, the angular velocity command u_ω is:

$$u_\omega(t) = \text{clamp}\left(K_p^\omega e_\theta(t) + K_d^\omega \dot{e}_\theta(t), -\omega_{\max}, \omega_{\max}\right) \tag{4.48}$$

This controller combines braking-in-time, lateral correction, and fine heading adjustments in a unified structure that adapts based on current alignment and proximity to the goal.

4.8.3 Follow-Global-Path Prior

This controller follows a predefined global path represented as a sequence of waypoints $\{\mathbf{p}_i\}_{i=0}^N$ in UTM coordinates. The control inputs are determined based on the nearest path point and a lookahead strategy.

At time t, let the vessel position be $\mathbf{p}(t) = [x(t), y(t)]^\top$. First, the index i^* of the closest waypoint on the global path is found by:

$$i^* = \arg\min_i \|\mathbf{p}_i - \mathbf{p}(t)\|_2 \tag{4.49}$$

The the arc-length from i^* until a set lookahead distance L_{LA} is exceeded to find index j:

$$\sum_{i=i^*}^{j-1} \|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2 \geq L_{\text{LA}} \tag{4.50}$$

Let $\mathbf{p}_{\text{LA}} = \mathbf{p}_j$ be defined as the lookahead waypoint. The Look-ahead error is found by:

$$\mathbf{e}_{\text{LA}}(t) = \mathbf{p}_{\text{LA}} - \mathbf{p}(t) \tag{4.51}$$

The Euclidean distance to the lookahead point is:

$$d_{\text{LA}}(t) = \|\mathbf{e}_{\text{LA}}(t)\|_2 + \varepsilon \tag{4.52}$$

where ε is a small positive constant added for numerical stability to avoid division by zero.

The desired heading toward the lookahead point is computed via the atan2 function:

$$\theta_{\text{LA}}(t) = \text{atan2}(e_{\text{LA},y}(t), e_{\text{LA},x}(t)) \tag{4.53}$$

From this, the heading error e_θ^{LA} with respect to the vessel's current heading θ is constrained into the interval $(-\pi, \pi]$ by:

$$e_\theta^{\text{LA}}(t) = (\theta_{\text{LA}}(t) - \theta(t) + \pi) \bmod 2\pi - \pi \tag{4.54}$$

The cross-track error c_{LA} is given by:

$$c_{\text{LA}}(t) = \sin(e_\theta^{\text{LA}}(t)) \cdot d_{\text{LA}}(t) \tag{4.55}$$

Then, the derivatives corresponding to the cross-track error c_{LA} and the heading error e_θ^{LA} are computed:

$$\dot{c}_{\text{LA}}(t) = \frac{c_{\text{LA}}(t) - c_{\text{LA}}(t-1)}{\Delta t}, \tag{4.56}$$

$$\dot{e}_\theta^{\text{LA}}(t) = \frac{e_\theta^{\text{LA}}(t) - e_\theta^{\text{LA}}(t-1)}{\Delta t} \tag{4.57}$$

The sway and yaw control inputs for the path following prior u_ℓ^{PF} u_ω^{PF} and are determined via the following PD controllers:

$$u_\ell^{\text{PF}}(t) = \text{clamp}\left(K_p^\ell c_{\text{LA}}(t) + K_d^\ell \dot{c}_{\text{LA}}(t),\, -u_{\ell,\max},\, u_{\ell,\max}\right) \tag{4.58}$$

$$u_\omega^{\text{PF}}(t) = \text{clamp}\left(K_p^\omega e_\theta^{\text{LA}}(t) + K_d^\omega \dot{e}_\theta^{\text{LA}}(t),\, -\omega_{\max},\, \omega_{\max}\right) \tag{4.59}$$

The surge command is designed to gradually accelerate to cruising speed:

$$u_f^{\text{PF}}(t) = \text{clamp}\left(v_f(t) + a_{\max}\Delta t,\, 0,\, u_{f,\max}\right) \tag{4.60}$$

The choice of the Lookahead Point involves a trade-off: A larger lookahead distance promotes smoother trajectories but can result in persistent lateral offsets from the global path. When applying a smaller lookahead distance, the path is followed more strictly but may cause oscillatory behaviour around the path, which in its turn introduces a cumulative offset as well. To balance smoothness and accuracy, the lookahead distance was tuned to an intermediate value.

This prior aims to promote the vessel to find its route across the global path more quickly. This is particularly prominent when switching out of a COLREG state or entering the sailing state after undocking. By rapidly identifying a low-cost trajectory along the path, the prior facilitates faster convergence, which can then be refined by using MPPI.

4.9 Summary and Method Overview

This chapter has presented the complete design of the local path planning module using high-level dynamics, allowing adaptation to various vessel geometries.

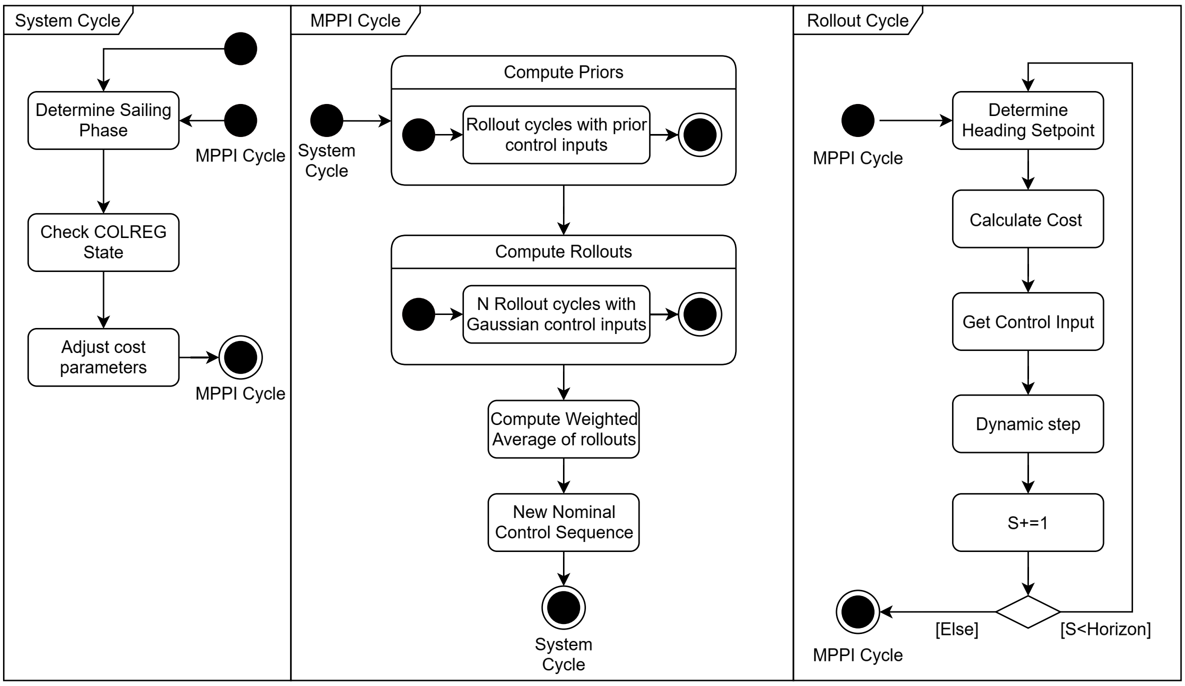


Figure 4.15: Overview of local path planning architecture distinguishing three cycle depths: The system cycle, MPPI cycle and rollout cycle.

Figure 4.15 shows the overview of the method. First Sailing phase and COLREG state checks are performed, and cost weights are adapted based on the current combination of scenarios. Then the MPPI cycle is started, which first computes rollouts using control inputs generated by the ancillary controllers. Next, the rollouts using Gaussian noise around the previous nominal control input are computed. Using the weighted average of all rollouts, the new nominal control sequence is determined.

Inside one rollout cycle, the only cost parameter that could change is the heading setpoint. The only cases for which this applies are the Overtaking Give-Way scenario and the Crossing Give-Way scenario (Paragraph 4.6.8), which have a two-phase policy based on geometry relative to the Target Ship.

Within the MPPI rollout cycle, it is assumed that a single combination between sailing phase and COLREG state remains fixed for the entire rollout, even if the simulated vessel pose during that rollout would normally trigger a sailing phase or COLEG state change. Allowing such switches within the rollout tends to drive the Own Ship into local minima, as it will exploit whichever state yields the lowest cost within the planning horizon.

The next chapter evaluates the performance of this planner in a simulation.

Chapter 5

Results

To test the proposed method, multiple simulations were run with different scenarios, including 14 dock-to-dock scenarios without basic encounters, three basic traffic encounters with no global path, and one dock-to-dock scenario, which includes an overtaking scenario, to evaluate the method's overall performance. The MPPI configuration, which is equal across all these scenarios, is found in Appendix D.

The experiments consist of three parameter configurations:

1. MPPI-Inland: All scenarios are executed with one fixed parameter configuration, including learned priors and the default MPPI period of running every iteration.

2. MPPI-Inland-NP: Identical to MPPI-Inland, except that the learned priors are turned off, isolating their impact on overall performance.

3. MPPI-Inland-10: Identical to MPPI-Inland, except that the MPPI replanning period is set to 10 (i.e. the controller replans once every 10 time-steps), thereby reducing the frequency of MPPI updates.

4. MPPI-Inland-wtwaxs: Identical to MPPI-Inland, except that the global path used is the interpolated waterway axis without using any projection or smoothing.

MPPI-Inland-NP was added to the list of testing configurations to test the effect of the designed ancillary controllers. The MPPI-Inland-10 configuration decreases the replanning frequency to produce a steadier, more predictable track. This allows clearer communication with other vessels about the course of the vessel. Another reason for less frequent updates is to reduce the controller's computational load and latency. Lastly, MPPI-Inland-wtwaxs was added to test the effects of projection and smoothing on the output of the local path planner.

By running MPPI less frequently, the controller's computational load and latency decrease, freeing up processing capacity for other critical subsystems such as machine perception.

When running one scenario, metrics such as the Own Ship's trajectory, the global path used, velocity profile, obstacle positions and sailing states are logged. On these logs, a data analysis is performed to find a set of metrics, of which the following metrics are presented:

- Terminal Error [m].

- Total travelled distance [m].

- Total Travelling Time [s].

- Minimum distance at Closest Point of Approach [m].

The data generated for each simulated scenario is then compared to the Triton dataset, which contains GPS coordinates, speed-over-ground and heading of the Damen Waterbus recorded at 10-second intervals over the course of one month. Individual docking events were classified by detecting prolonged stationary periods. Using these slicing conditions, individual voyages are identified. Given one scenario, all corresponding voyages are extracted. Metrics are produced for each voyage to compare real-life voyages to the simulated dock-to-dock scenarios:

- Total travelled distance [m].
- Total Travelling Time [s].

These performance metrics can be used to assess how well the proposed Path planning architecture performs against humanly operated vessels.

Every global path used in this test is generated with a step size of 100 meters, an offset from the Fairway of 30 meters, a minimum window size of 15, an average window size of 10, and with the average dynamic window applied for five iterations.

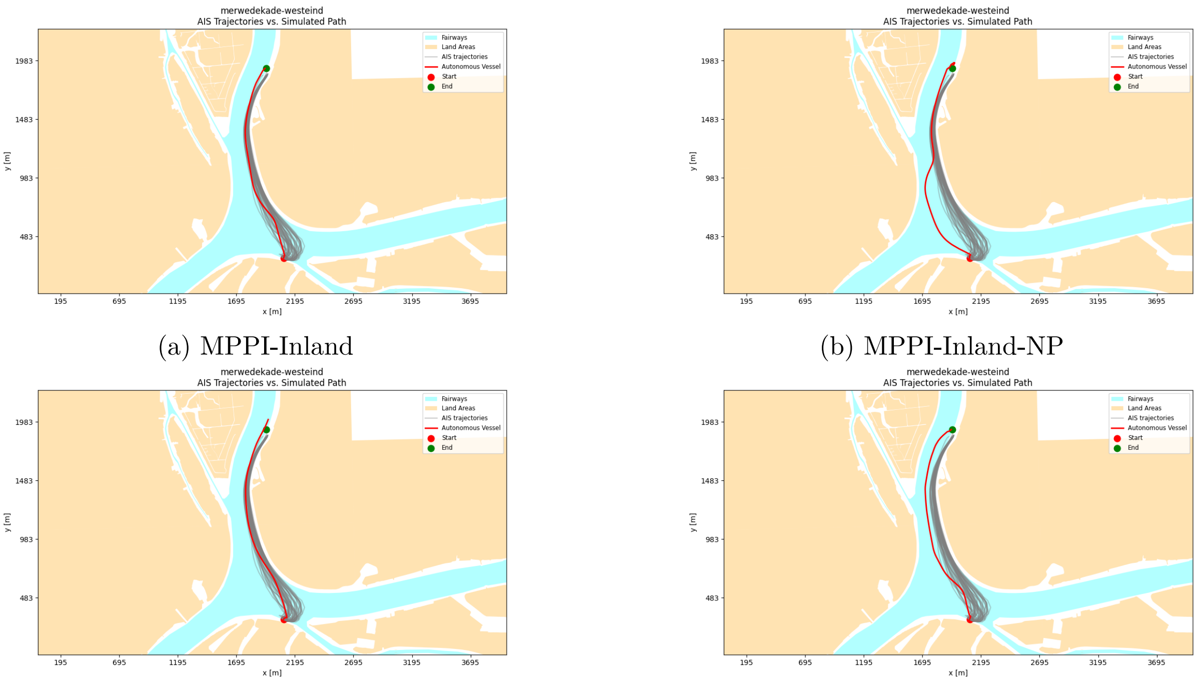
The remainder of this chapter discusses details of the tests performed for each type of scenario and the corresponding results.

5.1 Dock-to-Dock scenario analysis without traffic encounters

This section presents the tests performed and the corresponding results for Dock-to-Dock scenarios. The docks analysed are the stations used by the Damen Waterbus, except for the stations in areas where the corresponding IENC author is Port of Rotterdam. Note that, for simplification, the start and goal waypoints set in this scenario do not fully correspond to the dock locations but approach the dock locations because many docks are outside the Fairway object.

Figure 5.1 displays a relatively longer voyage between two docks. By visual inspection, the trajectories generated by MPPI-Inland and MPPI-Inland-10 seem similar to the routes preferred by captains. The MPPI-Inland-NP method shows a larger deviation from the voyages, especially after undocking. When the global path is set to the waterway axis (MPPI-Inland-wtwaxs), the own ship deviates more from the commonly sailed routes as trajectories across the starboard shoreline are no longer the preferred course.

Figure 5.2 shows an example of a scenario where the distance between docks is relatively low, which yields various outcomes across the methods applied. The waterway axis contains a sharp turn, which the projection and smoothing method were unable to resolve. Visually, the trajectories generated by MPPI-Inland and MPPI-Inland-wtwaxs seem to get stuck in a local minimum, with their heading straight to the border of the Fairway. The trajectory corresponding to MPPI-Inland-10 appears to follow the global path as well, with some slight deviation. However, the version generated by MPPI-Inland-NP without priors fails to take the sharp turn and continues moving in a straight line.



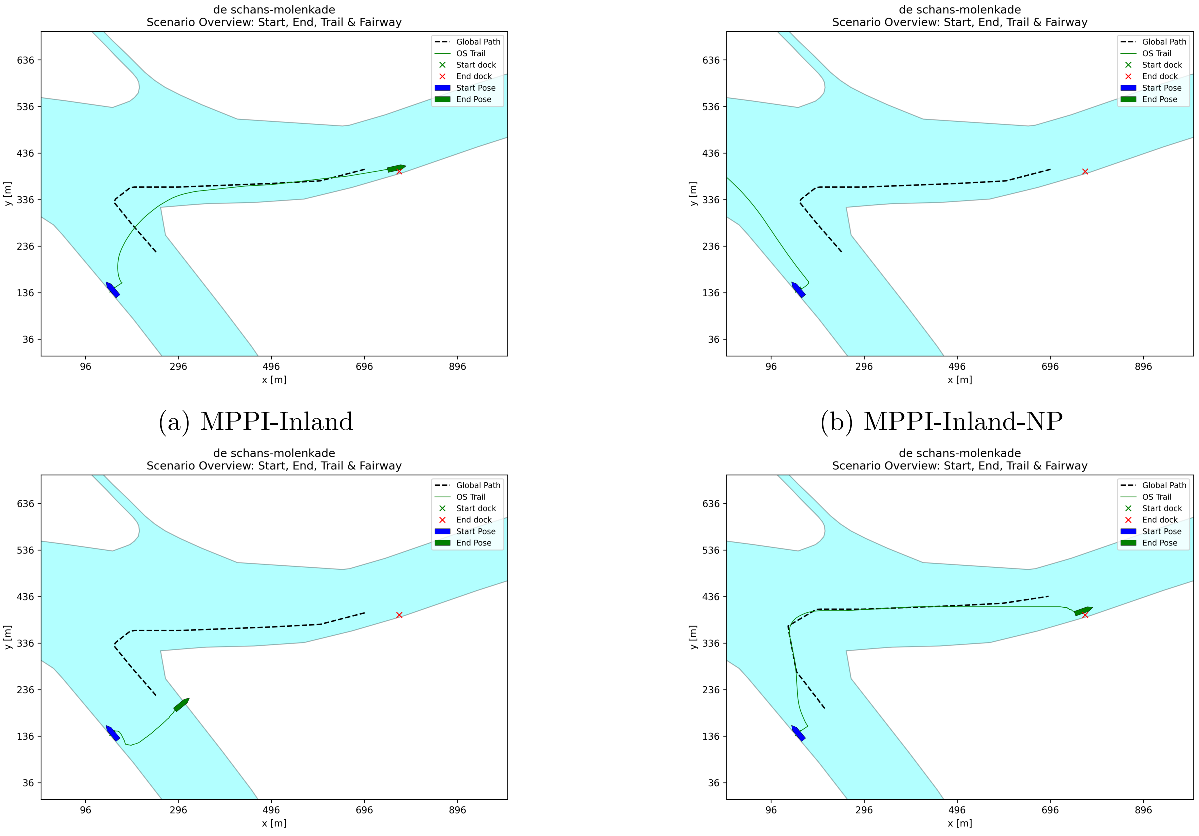
(a) MPPI-Inland

(b) MPPI-Inland-NP

(c) MPPI-Inland-10

(d) MPPI-Inland-wtwaxs

Figure 5.1: Trajectory of autonomous vessel compared to AIS logged voyages from Merwedekade to Westeind.



(a) MPPI-Inland

(b) MPPI-Inland-NP

(c) MPPI-Inland-10

(d) MPPI-Inland-wtwaxs

Figure 5.2: Trajectory of autonomous vessel and the corresponding global path from De Schans to Rijsdijk.

In Appendix E, detailed test results for each scenario are shared in Tables E.1–E.4. They list the performance metrics for every individual scenario under each parameter configuration (MPPI-Inland, MPPI-Inland-NP, MPPI-Inland-10 and MPPI-wtwaxs). The mean and standard deviation of each metric across all scenarios are used to indicate overall performance. Note that for many cases, the standard deviation is relatively high due to large outliers (e.g. cases where the Own Ship moves past the dock, causing a relatively large terminal error). Still these performance metrics can be used to give an overall indication of performance between methods.

Table 5.1 shows that each parameter configuration reaches the goal within 10 meters in 11, 3, 3 and 8 out of the 14 scenarios, respectively, which implies that all methods are not perfect. By visual inspection, it was found that in some cases where the end goal was not reached, there was a sharp turn somewhere in the global path which the MPPI controller, given its dynamics and Gaussian constraints, does not recognise, therefore getting stuck in a local minimum such as prioritising keeping to the set velocity over moving towards the goal. In some cases, this could lead to the ship taking a wrong turn.

In many cases, the system reaches a local minimum when near the dock, causing it to remain in the area, but never get the dock. Adding the Ancillary controllers (MPPI-Inland) appears to help the system escape these local minima (when comparing to MPPI-Inland-NP). Therefore, the goal is more frequently found with the MPPI-Inland and the MPPI-Inland-wtwaxs configurations. When running MPPI only once per ten iterations (MPPI-Inland-10), the system can become trapped in local minima more frequently. In many of these cases, this aligns with a switch between sailing phases. For example, in cases where the sailing phase switches from approach to dock, the controller policy is not updated in sync with the sailing phase switcher, assuming the previous policy for a few more iterations. This causes the target ship to overshoot the dock, finding itself out of docking range once the next MPPI iteration begins.

This phenomenon does not only apply to MPPI-Inland-10, the two cases where the dock is not reached due to getting stuck in a local minimum with the MPPI-Inland configuration both have the sailing state switch from ”Sailing” to ”Approach” right before a sharp turn.

Table 5.1: High-level outcomes across all scenarios for each MPPI configuration. A case is classified as a ”Success” if the Terminal Error (TE) is smaller than 10 meters, where it is classified as ”Near Dock” when the Terminal Error is in between 10 and 200 meters.

scenario	MPPI-Inland	MPPI-Inland-NP	MPPI-Inland-10	MPPI-Inland-wtwaxs
Success (TE $<$ 10 m)	11	3	3	8
Cases Near Dock (10 m $<$ TE $<$ 300 m)	1	6	7	5
Cases taking wrong turn	0	3	0	0
Cases Stuck in local minimum	2	2	4	1

Quantitative comparisons are only meaningful in scenarios where a method succeeds, since metrics such as docking time are undefined if the vessel does not reach the dock. Because the success sets of MPPI-Inland-NP and MPPI-Inland-10 do not overlap, results are reported case-by-case: for each scenario completed by a variant, its outcome is compared against MPPI-Inland on the same scenario; no cross-scenario averages are reported. The results of this pairwise comparison are presented in Table 5.2

Table 5.2: Pairwise comparison of path length and travel time, averaged over the common-success subset of scenarios for each pair (i.e., only scenarios where both methods reached the goal with terminal error $< 10\,\text{m}$ are included).

Method	Path length [m]	AIS mean traj length [m]	Travel time [s]	AIS mean travel time [s]
MPPI-Inland	1311	1296	228	224
MPPI-Inland-NP	1404	1296	208	224
MPPI-Inland	1097	1100	239	196
MPPI-Inland-10	1220	1100	425	196
MPPI-Inland	1931	1834	311	263
MPPI-Inland-wtwaxs	1937	1834	295	263

Table 5.3 presents the computation times for each of the methods. Most methods seem to receive similar results, except for MPPI-Inland-10, which shows computation times at least twice as fast.

Table 5.3: Computation times across all methods.

Method	Minimum Computation Time [s]	Average Computation Time [m]	Maximum Computation Time [s]
MPPI-Inland	0.91	0.97	1.18
MPPI-Inland-NP	0.95	0.95	1.42
MPPI-Inland-10	0.39	0.45	0.58
MPPI-Inland-wtwaxs	0.91	0.95	1.23

The average path length of MPPI-Inland is lower than the average path length of MPPI-Inland-NP and MPPI-Inland-10. The path lengths of MPPI-Inland and MPPI-Inland-wtwaxs are slightly lower than the average trajectory length of a human-operated vessel. Across the methods, travel time shows similar results, except for MPPI-Inland-10, which has a significantly larger travel time than MPPI-Inland. The probable cause of this discrepancy is presented in Figure 5.3, which shows an overview of one of the scenarios where the dock was reached using MPPI-Inland-10. Near the dock, the vessel's trajectory seems to overshoot the dock various times until settling in its berthing zone.

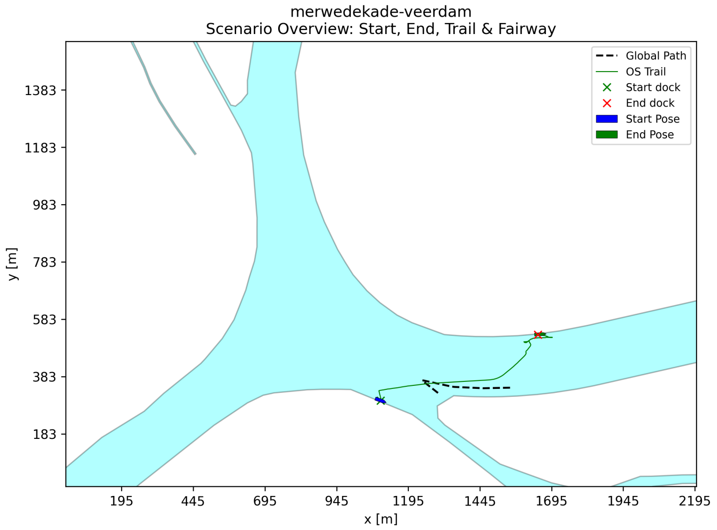


Figure 5.3: MPPI-Inland-10 overshoots the dock multiple times before berthing.

5.2 Traffic Encounters

Figures 5.4-5.6 illustrate an overview of the behaviour of Own-Ship and Target-Ship positions at key COLREG milestones in each traffic scenario. Plotted vessel markers in pink and yellow correspond to (1) the instant the COLREG state detector recognises the encounter, (2) the point at which the post-condition for that COLREG rule is satisfied, and (3) the midpoint between those two instants, highlighting the vessel's trajectory during the manoeuvre.

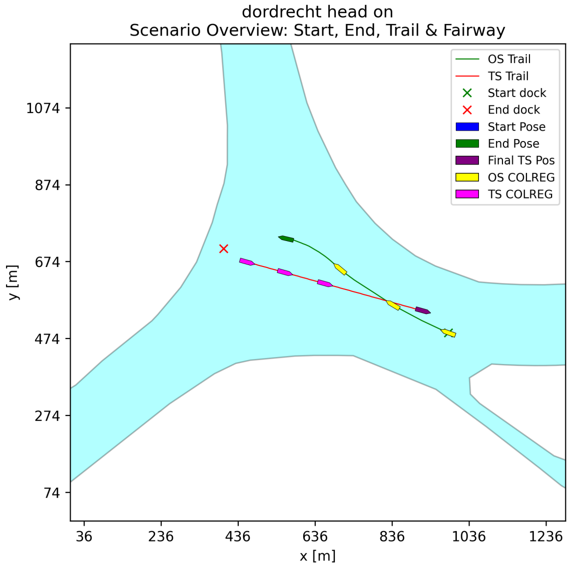


Figure 5.4: Head-on Scenario overview.

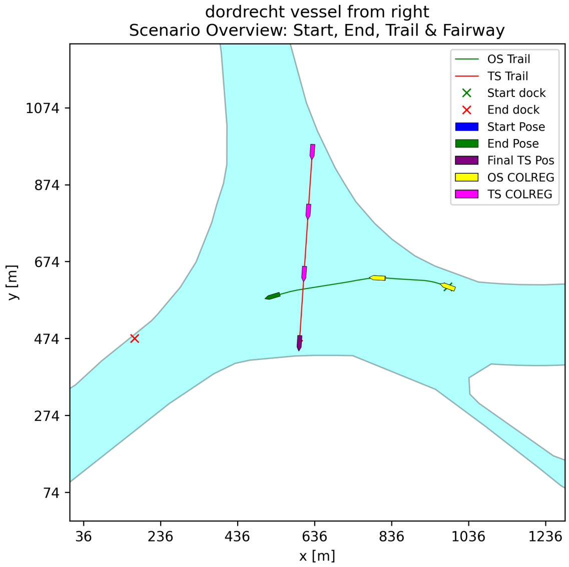


Figure 5.5: Crossing Give-way Scenario overview.

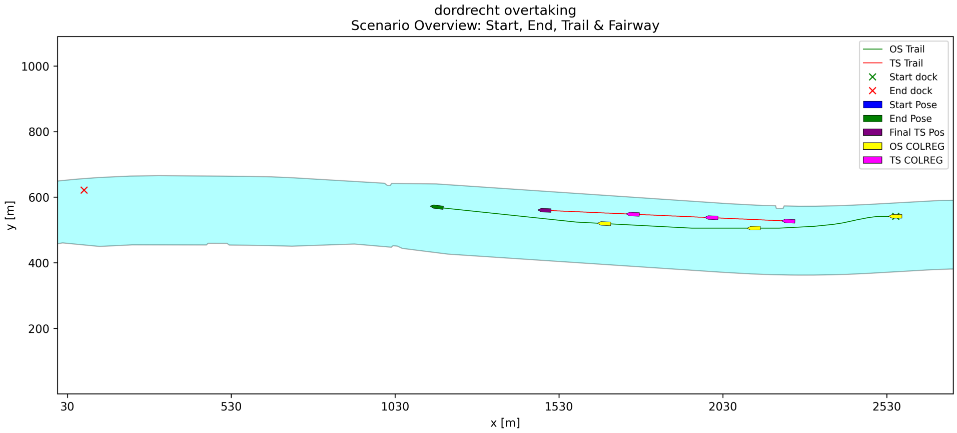
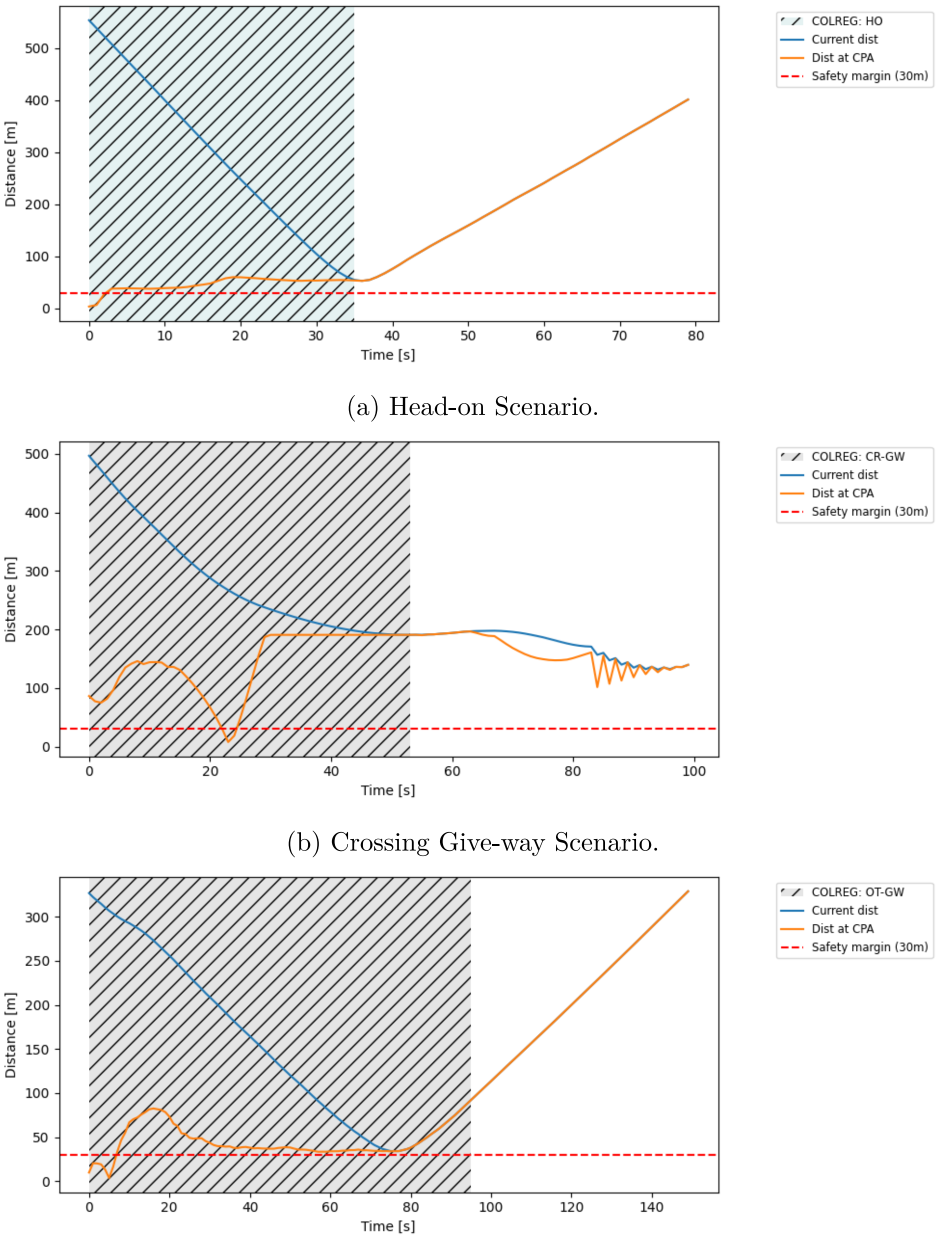


Figure 5.6: Head-on Scenario

Figures 5.4-5.6 show successful obstacle avoidance according to the COLREGs for each scenario by using our adaptation on Biased-MPPI. Figure 5.7 shows the distance between the Own Ship and the Target ship at the Closest Point of Approach over time. For all scenarios, the manoeuvre was performed keeping a safety distance from the Target Ship.

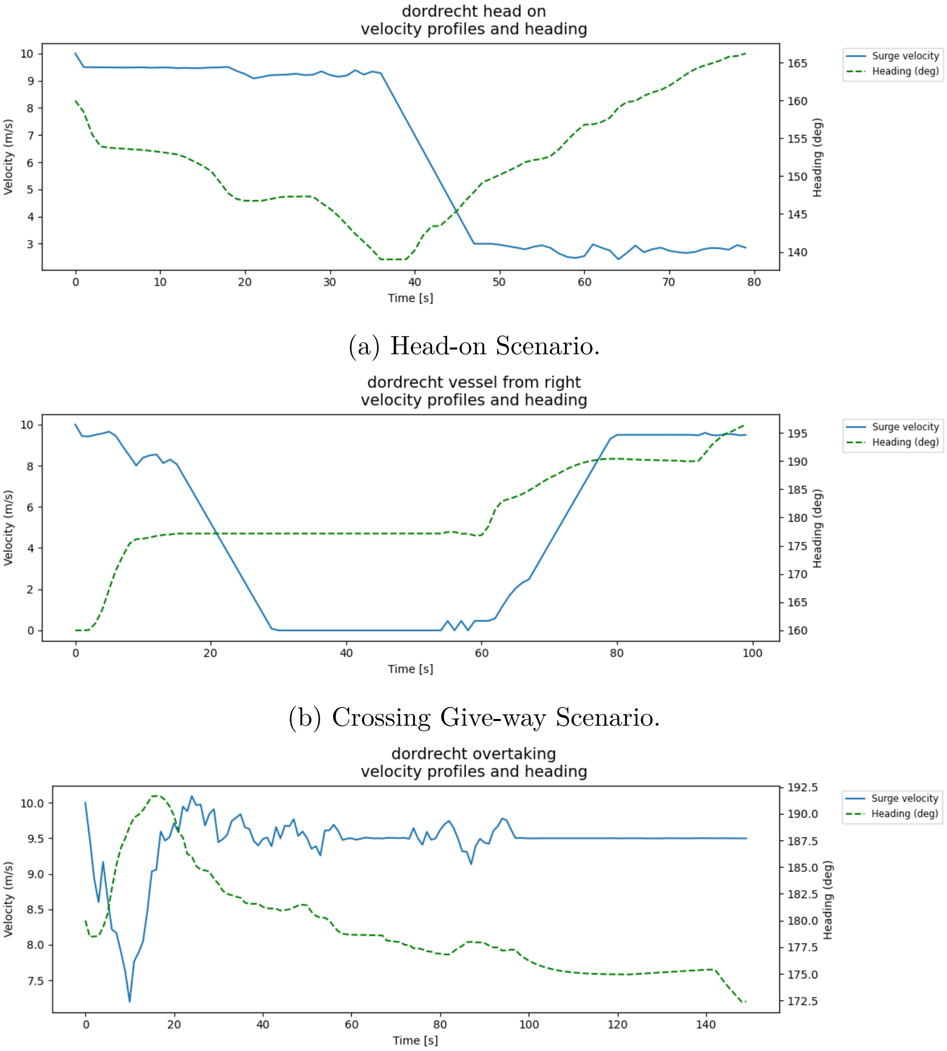


(a) Head-on Scenario.

(b) Crossing Give-way Scenario.

(c) Overtaking Scenario.

Figure 5.7: Distance to Closest Point of Approach (CPA) over time for the head-on, crossing give-way, and overtaking scenarios.



(a) Head-on Scenario.

(b) Crossing Give-way Scenario.

(c) Overtaking Scenario.

Figure 5.8: Current velocities and heading over time for the head-on, crossing give-way, and overtaking scenarios.

Figure 5.8 provides more insight into the behaviour of the Own Ship for each scenario. In the Head-on scenario, the velocity remains stable around the setpoint while the heading is adjusted. When approaching the target vessel, a slight heading correction is made, then the COLREG state exits.

In the Overtaking Give-way scenario, the Own ship's surge velocity remains decently stable, and the heading is adjusted to start performing the overtaking manoeuvre. Once the Own Ship is sufficiently positioned to the port side of the Target Ship, it adjusts its heading to match the heading of the Target Ship. When the Own Ship is two shiplengths in front of the Target Ship, the COLREG state is exited, after which the Own Ship readjusts its heading back to course.

In the Crossing Give-way scenario, the Own Ship slightly brakes and adjusts its heading at first. Next, the Own Ship comes to a complete stop. Once the Target Ship moves past, the Own Ship continues its voyage.

Figure 5.7 emphasises that every manoeuvre is performed safely since all of these cases remain outside the safety margin. These methods show effective for obstacle avoidance, adhering to the COLREGs.

5.3 Dock-to-Dock with Traffic Handling

This paragraph examines a full dock-to-dock scenario from Merwedekade to Westeind with a vessel moving across the global path at 1.5 m/s. The Own Ship will have to overtake this vessel somewhere along its voyage.

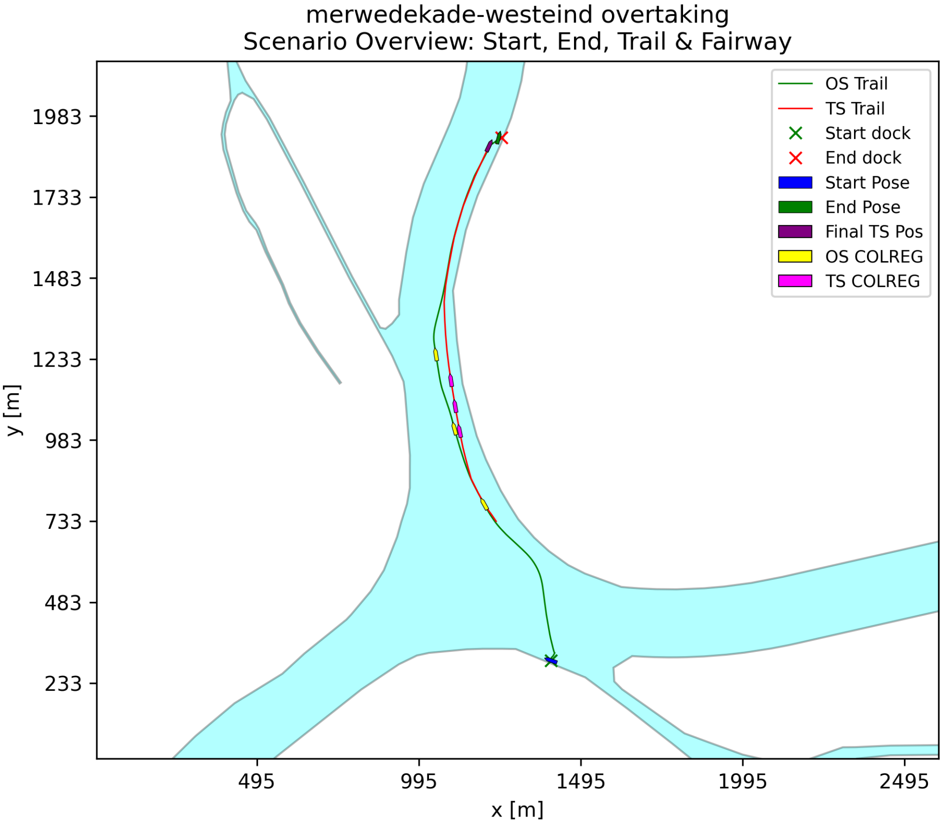


Figure 5.9: Overview of Merwedekade-Westeind with overtaking scenario.

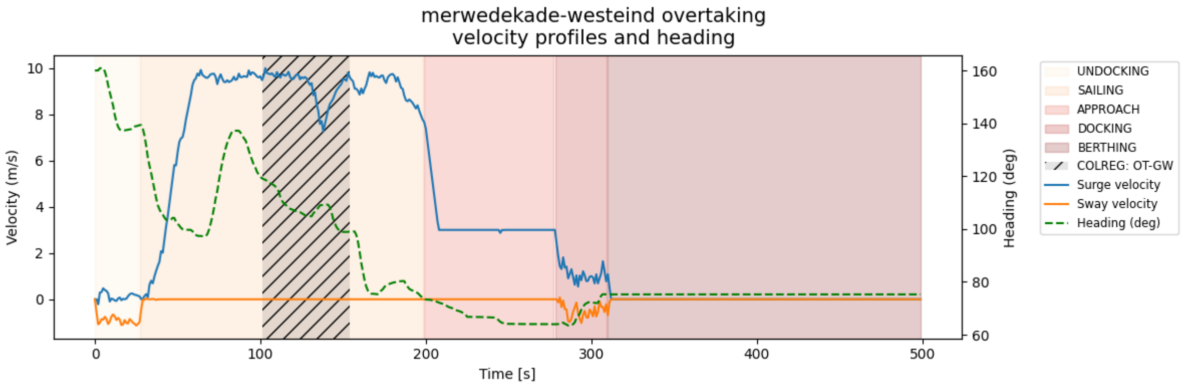


Figure 5.10: Velocity profiles and Heading of Merwedekade-Westeind with overtaking scenario. Shaded regions correspond to different sailing phases. The Hatched area corresponds to a traffic encounter.

Figures 5.9-5.10 collectively show the overall dynamic behaviour of the Own Ship during all sailing phases. During undocking, the Own Ship moves laterally in the starboard direction, away from the dock. Once sufficiently clear from the dock, the sailing phase is reached, and the

Own Ship moves its heading to align with the global path. A new velocity setpoint is applied, causing the own ship to apply maximum acceleration.

When aligned with the global path, the COLREG state switcher classifies an Overtaking Give-way scenario. While performing the overtaking manoeuvre, the global path weight is disregarded, allowing larger deviations from the global path, causing the Cross Track Error to increase as shown in Figure 5.12. Once the Own Ship is sufficiently in front of the Target Ship, the postconditions of the COLREG state are reached and the regular cost conditions are applied. This causes the Own Ship to change its course to realign with the global path. The overtaking manoeuvre was performed in a sufficiently safe manner, as the minimum distance between the Own Ship and the Target Ship is greater than the set safety margin of 30 m.

Once the approach stage is reached, the Own Ship adjusts its surge velocity to approach velocity. The Own Ship allows for more deviation from the global path to set up for docking.

When in proximity of the dock, the docking stage is started, further slowing down the Own Ship. Starting this stage, a slight cost is applied to the heading error, promoting the vessel to move slightly into place. Figure 5.12 shows an increase in cross-track error here because the final point of the global path has been passed by the Own Ship, and it is moving toward its final waypoint.

When the berthing scenario is reached, the heading cost is amplified. A very small setpoint sway velocity is applied to promote moving laterally to align with the goal. This setpoint and the associated weight are set such that the positional cost significantly outweighs the sway velocity, such that an equilibrium close to the final waypoint is reached.

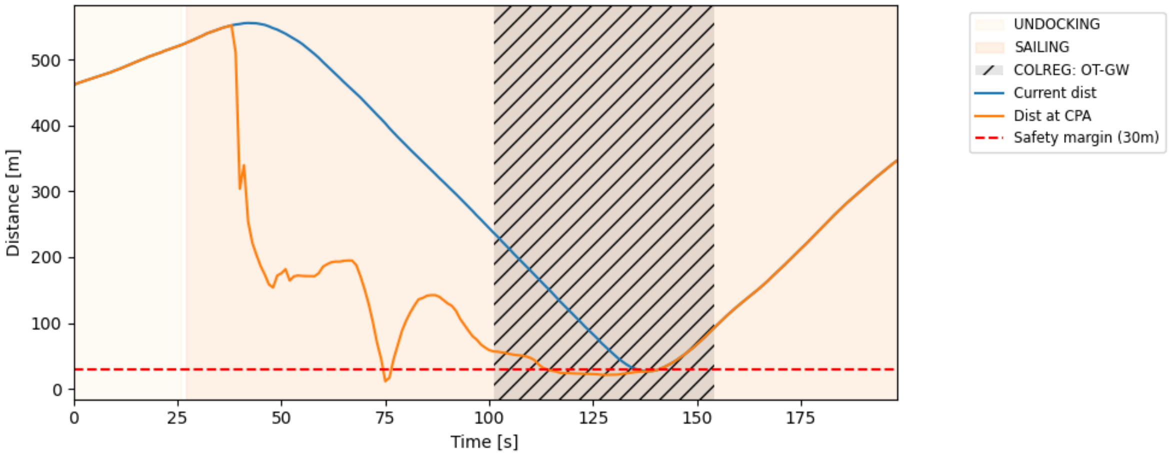


Figure 5.11: Distance between Own Ship and Target Ship the Closest Point of Approach, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario.

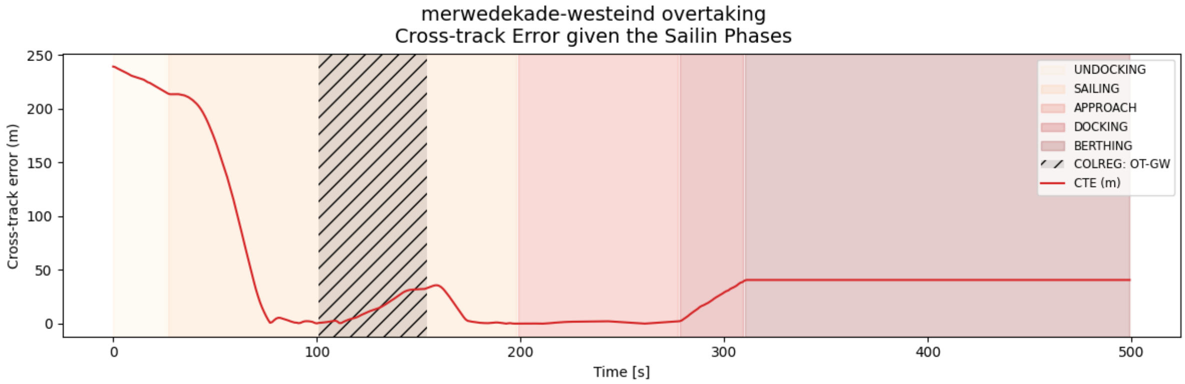


Figure 5.12: Cross Track Error, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario. Shaded regious correspond to different sailing phases. The Hatched area corresponds to a traffic encounter.

5.3.1 Disabling Priors

Figures 5.13-5.14 provide an overview of the overall behaviour of the Own Ship in the Dock-to-Dock with traffic handling scenario without active priors. Similar to the scenario presented in Figure 5.2b, the Own Ship starts the voyage remaining on roughly the same course as the starting heading until the maximum velocity is reached. The vessel begins adjusting its course towards the global path. This change of movement causes the Own Ship to approach the Target Vessel from a different angle as opposed to the scenario with the MPPI-Inland configuration.

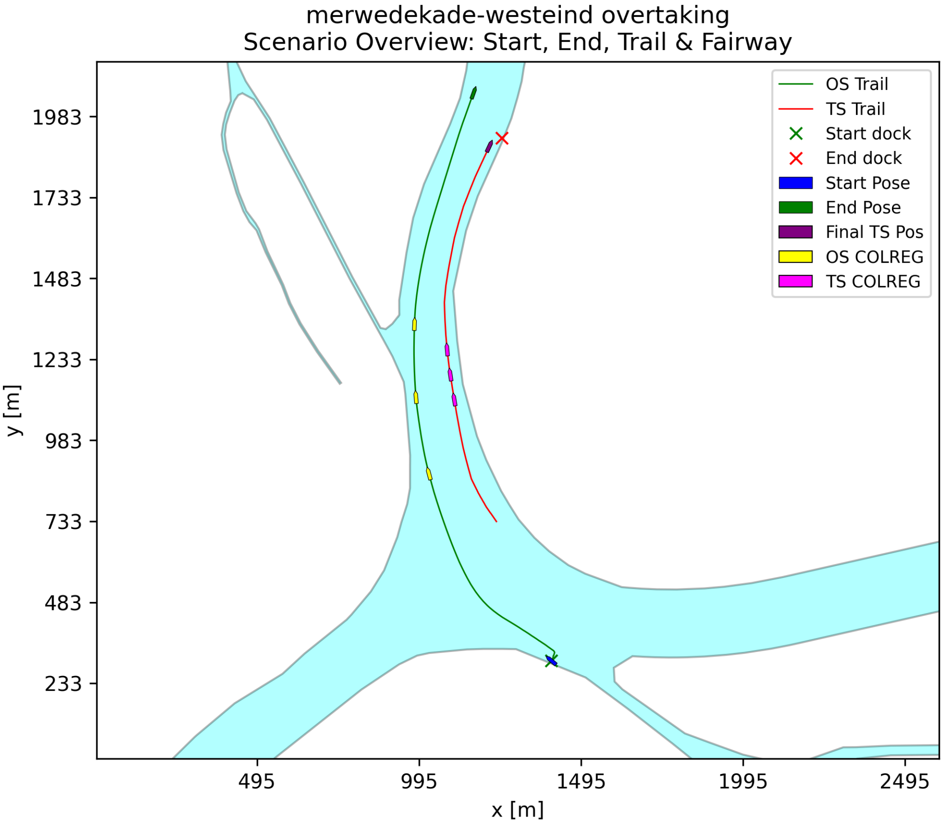


Figure 5.13: Overview of Merwedekade-Westeind with overtaking scenario (Priors disabled).

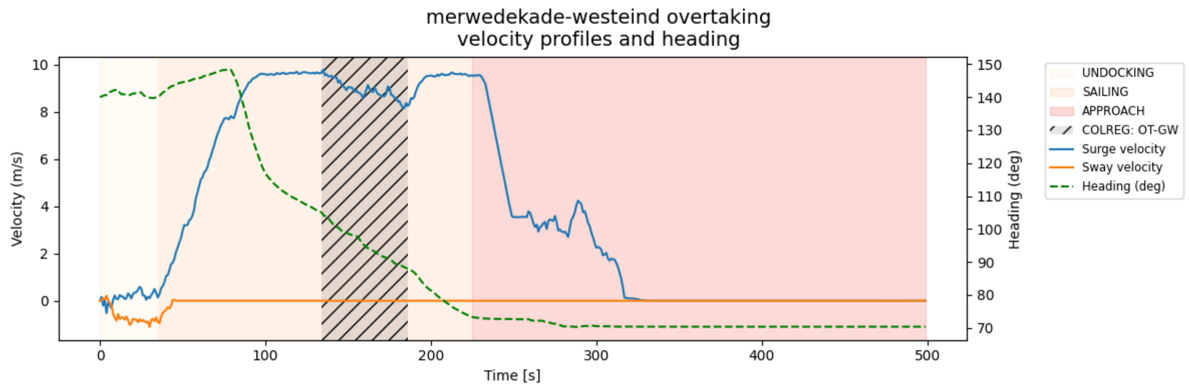


Figure 5.14: Velocity profiles and Heading of Merwedekade-Westeind with overtaking scenario (Priors disabled). Shaded regions correspond to different sailing phases. The Hatched area corresponds to a traffic encounter.

The overtaking manoeuvre is performed safely as the distance between the Own Ship and the Target Ship does not fall within the safety margin, as shown by Figure 5.15. When evaluating the methods' trajectory, shown in Figure 5.13, the vessel seems to have trouble moving towards the global path after the overtaking manoeuvre is reached. It does enter the Approach state, but the Docking state is never reached as it moves past the dock. This is likely due to the controller being stuck in a local minimum.

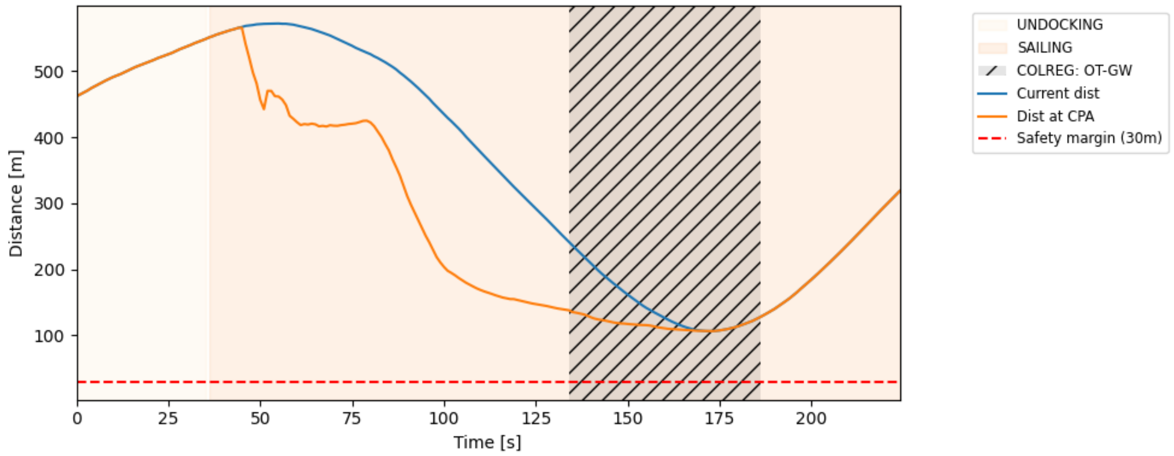


Figure 5.15: Distance between Own Ship and Target Ship the Closest Point of Approach, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario (Priors disabled).

5.3.2 Running Biased-MPPI every 10 iterations

Figures 5.16-5.17 provide an overview of the overall behaviour of the Own Ship in the Dock-to-Dock with traffic handling scenario. Similarly to the basic scenario of Figures 5.9-5.12, the Own Ship aligns with the global path and classifies an Overtaking Give-way scenario, adjusting its heading setpoint. Due to the Own Ship's trajectory being updated less frequently, the Own Ship runs the Biased-MPPI algorithm a few iterations later than the exit of the Overtaking Give-way scenario. Due to the lower responsiveness of MPPI-Inland-10, the Own Ship cannot detect the pre- and postconditions as quickly as the other methods. This causes the Own Ship to have a more unstable velocity profile (Figure 5.17) slightly cross the safety boundary (Figure 5.18) with the Target Ship and take more time to move back to the course given by the global path.

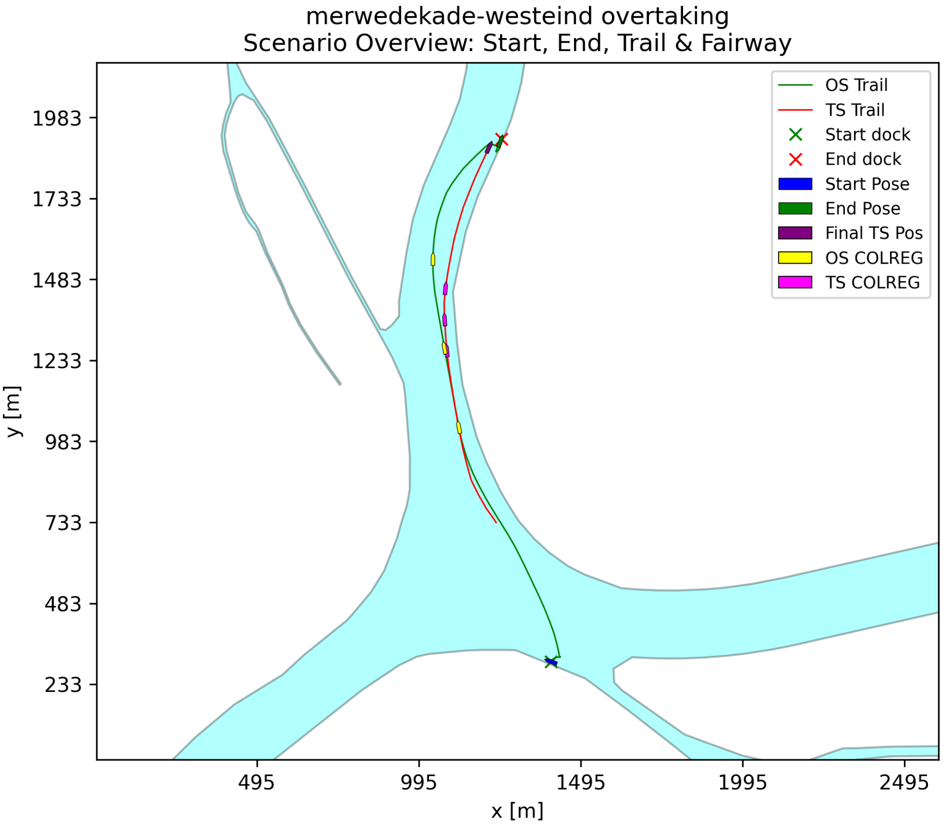


Figure 5.16: Overview of Merwedekade-Westeind with overtaking scenario of MPPI-Inland-10.

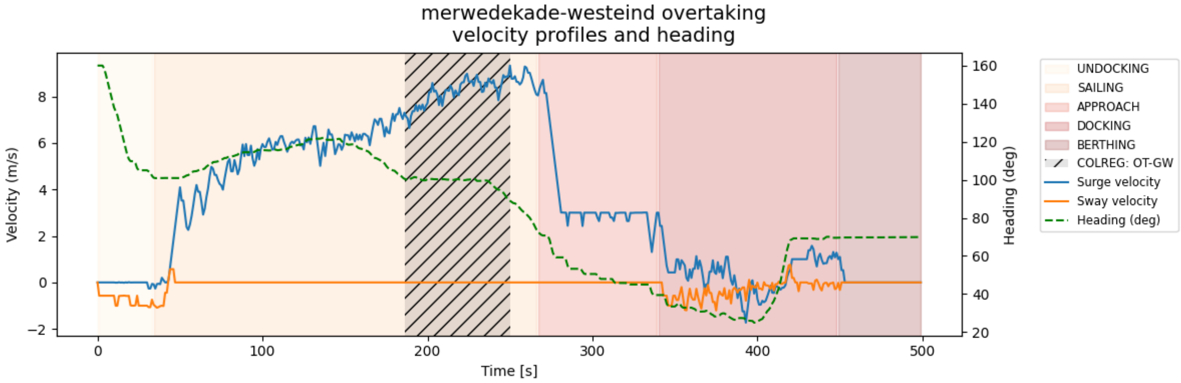


Figure 5.17: Velocity profiles and Heading of Merwedekade-Westeind with overtaking scenario of MPPI-Inland-10. Shaded regions correspond to different sailing phases. The Hatched area corresponds to a traffic encounter.

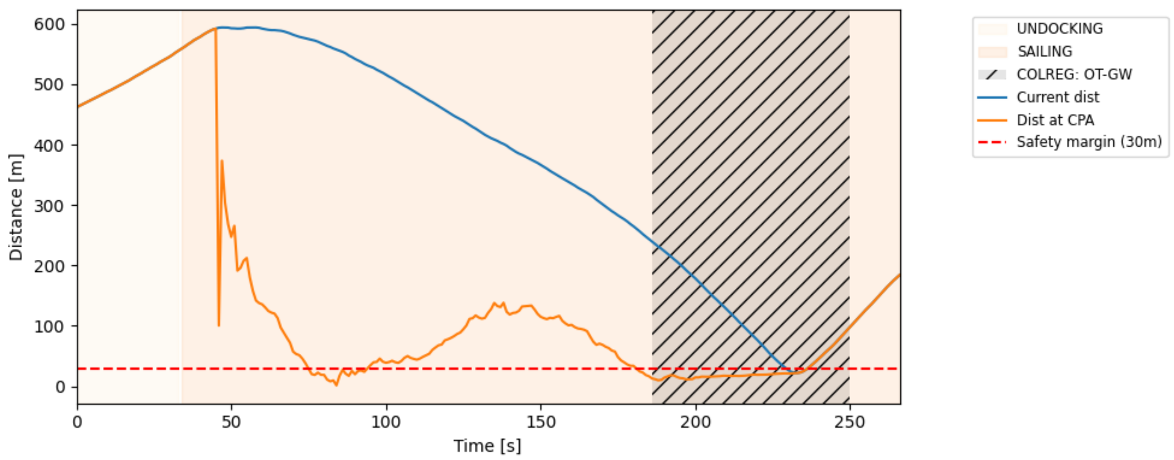


Figure 5.18: Distance between Own Ship and Target Ship the Closest Point of Approach, Sailing and COLREG states over time of Merwedekade-Westeind with overtaking scenario of MPPI-Inland-10.

5.4 Details of near-dock manoeuvres

This section presents details on the Docking and Undocking manoeuvres using the basic configuration. Figures 5.19 and 5.21 show the overview and velocity profiles for the undocking scenario. While undocking, the heading remains stable until the sailing phase is reached. Then it starts moving its heading to align the vessel with the global path when the sailing phase is reached.

Figures 5.20 and 5.21 show the overview and velocity profiles for the docking scenario, and Figures 5.22-5.24. First, the dock is approached at a straight line during the approach phase, combining global path following and Euclidean distance tracking. Once the docking phase is entered, the vessel slows down and starts incorporating a sway velocity to align with the final dock. Once the berthing phase is reached, the vessel is assigned to remain in the current position. Within a system architecture beyond the model presented in this thesis, this could also be interpreted as ”goal is reached”.

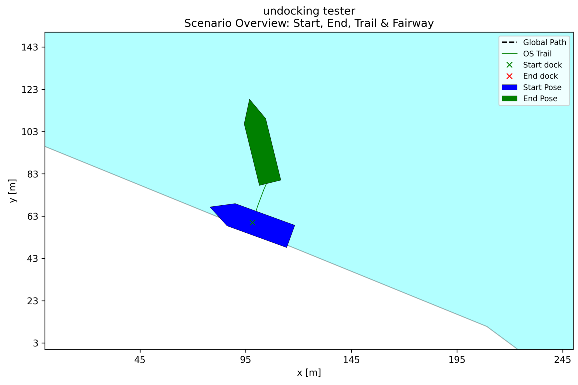


Figure 5.19: Undocking overview.

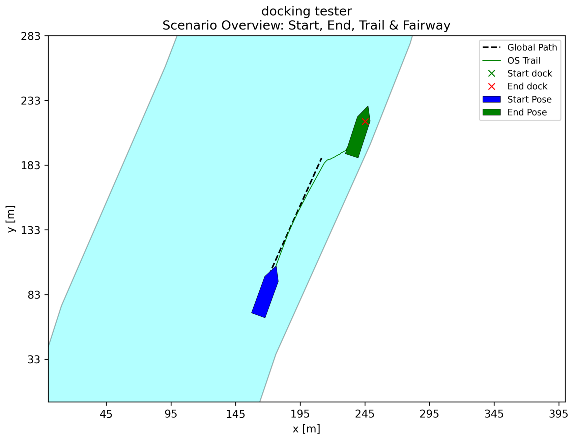


Figure 5.20: Docking overview.

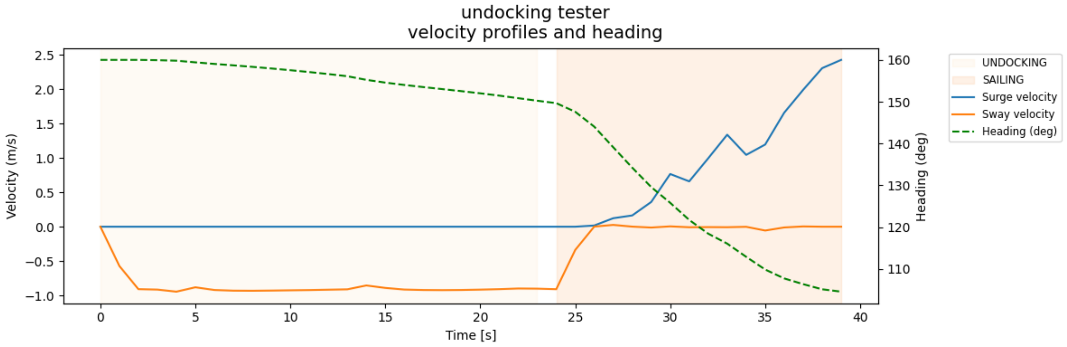


Figure 5.21: Undocking velocity profiles and heading.

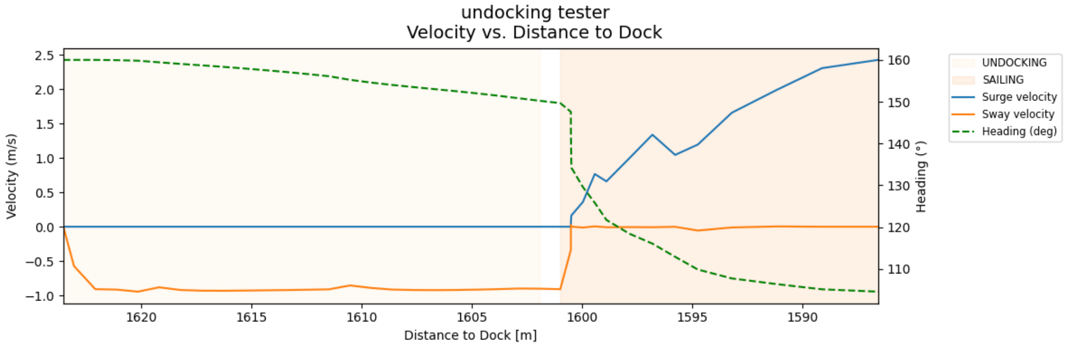


Figure 5.22: Undocking velocity profiles and heading in terms of distance to dock.

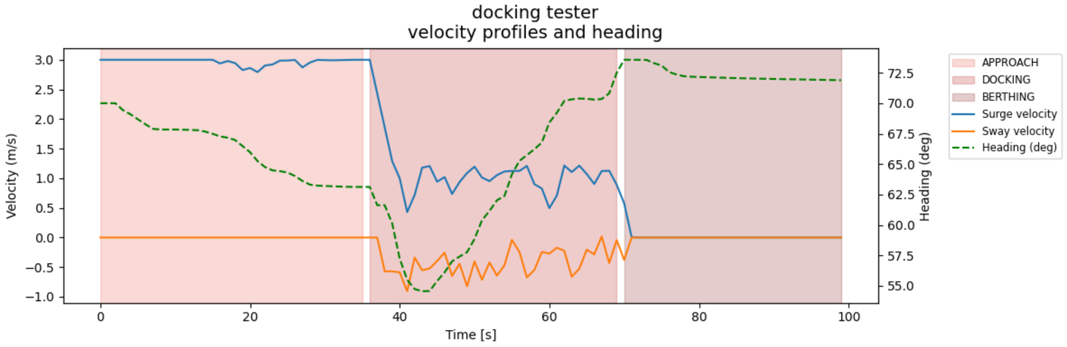


Figure 5.23: Docking velocity profiles and heading.

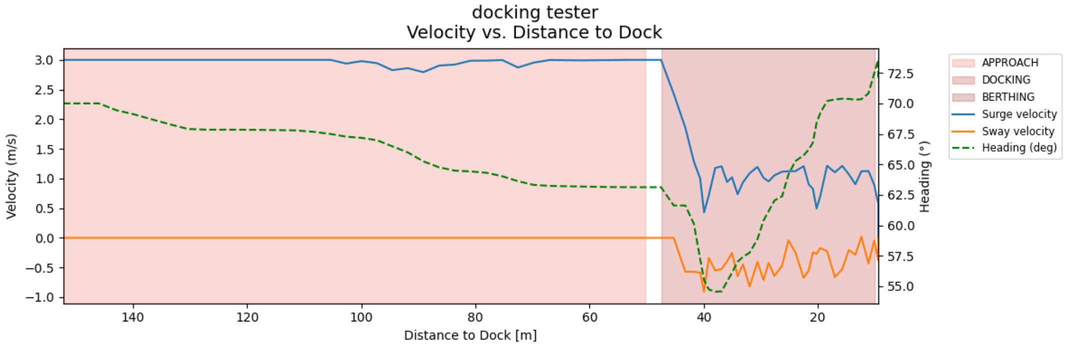


Figure 5.24: Docking velocity profiles and heading in terms of distance to dock.

5.5 Discussion

The results presented in this chapter demonstrate the performance and limitations of the proposed path-planning architecture under a variety of conditions. Across the different parameter configurations, MPPI-Inland achieved the highest number of successful dock-to-dock completions, suggesting that the combination of ancillary controllers and full-rate replanning provides a tangible benefit in avoiding local minima and ensuring that the vessel reaches its goal. This effect was particularly prominent in scenarios near the dock, where ancillary controllers appeared to provide additional guidance to escape local minima.

Disabling the priors (MPPI-Inland-NP) led to a notable decrease in success rate, with more cases of either missing the dock entirely or becoming trapped in local minima. This confirms that the priors contribute meaningfully to overall performance, particularly during docking and in scenarios with complex geometries. However, MPPI-Inland-NP did occasionally produce shorter travel times in the subset of cases where it did succeed, suggesting that the absence of priors can sometimes allow for more direct, but riskier, manoeuvres.

Reducing the replanning frequency (MPPI-Inland-10) achieved the expected improvement in computational efficiency, roughly halving computation time compared to other methods. However, this came at the cost of reduced responsiveness, docking overshoot events, and a temporary safety margin violation during traffic encounters, particularly the overtaking scenarios. Here, trajectory updates prevented timely transitions between COLREG states and cruising. This suggests that while reduced replanning can be viable for computational load management, it requires an additional method to ensure responsiveness to dynamic events.

The configuration using the waterway axis instead of the post-processed starboard-biased global path (MPPI-Inland-wtwaxs) resulted in more deviations from human-preferred routes. The absence of projection and smoothing causes a slightly lowered success rate compared to MPPI-Inland, but not as much as the other tested configurations. Nevertheless, travel times and path lengths remained comparable to MPPI-Inland in successful runs; the controller is still able to produce feasible solutions in a considerable number of cases.

Traffic encounter scenarios confirmed that MPPI-Inland, with the proposed COLREG policies, can successfully perform manoeuvres in compliance with basic COLREG rules, maintaining safe separation in all testing scenarios. The differences between methods became more pronounced in combined dock-to-dock with traffic handling scenarios. Here, MPPI-Inland displayed smoother integration between COLREG-driven manoeuvres and global path following, whereas MPPI-Inland-10 demonstrated being the only method to cross the safety boundary around the Target Ship.

Manoeuvres near the docks showed predictable vessel behaviour: Lateral movement to undock until the sailing phase is reached, and a structured slow-down during the approach phase with a combination of slow forward and lateral movement to bring the vessel in position to reach the berthing location in the docking phase. Failures to dock were frequently linked to either suboptimal global path geometry near the target or control policies that were not updated synchronously with sailing phase changes. These findings suggest that improved coordination between sailing phases could further improve docking reliability.

Overall, the results indicate that MPPI-Inland, with full-rate replanning and learned priors, offers the most robust performance across diverse scenarios compared to the other configurations.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis set out to answer the overarching research question:

”How can a modular path planning architecture generate desirable trajectories for inland autonomous vessel navigation between docks while safely adapting to marine traffic using electronic navigational charts (ENCs)?”

Below, main findings and reflections are presented by addressing each research sub-question.

6.1.1 Which quantitative metrics best capture global and local planner performance?

A comprehensive set of performance indicators was defined and implemented for both the global mission planner (computation time, path length, turning radius, projection regularity) and the local trajectory generator (terminal error, minimum distance at CPA, COLREG compliance rate). Experimental results across fourteen diverse inland scenarios demonstrated that these metrics provide insight on the path quality and local vessel behaviour regarding traffic.

Time measurements were observed to be misrepresented due to overhead caused by data logging in each iteration. To accurately quantify latency due to the MPPI algorithm, future evaluations should include dedicated runs with minimum logging.

6.1.2 How can S-57 ENC layers be leveraged to generate safe, starboard-biased global routes that respect vessel dynamic constraints?

A graph representing the centre of all canals in the area between Rotterdam and Dordrecht was constructed from the ENC WaterwayAxis feature, and waypoints were projected onto the boundaries of the Fairway object to move the global path to the starboard side. Hereafter, various smoothing techniques were applied, mainly a hybrid method combining minimum and average window smoothing with multiple iterations. By definition of the Fairway geometry, which is specifically designed for larger marine vessels, under-keel clearance is guaranteed. Two challenges were identified:

1. Waterway axis structure discontinuities at intersections and narrow channel transitions leading to infeasible segment jumps.

2. Some ENC producers, like Port of Rotterdam, tend to draw the Fairway object disjointed, causing issues with projection and feasibility checks.

6.1.3 What effects do post-processing techniques have on global paths?

A grid search over three smoothing strategies (minimum window, average window, hybrid) showed that smoothing reduces abrupt changes and significantly reduces projection issues in intersections. Across the various smoothing methods, failure rates deviate slightly. The method with the lowest failure rate was the hybrid method, applying multiple iterations of the average window method after performing one iteration of the minimum window method. Limitations remain in cases with sharp turns or data discontinuities inherent to the ENC provider.

6.1.4 How can an MPPI-based local controller be designed to balance collision avoidance, COLREG adherence, and real-time execution under multi-vessel encounters?

The MPPI-Inland local controller presented in this thesis, with ancillary controllers including the go-to-goal-and-brake and follow-global-path controllers, demonstrated the best performance across all tested methods, reaching the end dock in 11 out of 14 scenarios. The MPPI-Inland method was compared to three different methods, which adapted one aspect from MPPI-Inland, including turning off the ancillary controllers, running MPPI once per ten iterations, and using the waterway axis as a global path instead of the projected and post-processed global path. Key observations include:

- COLREG rules (head-on, crossing, overtaking) were enforced via a state-switcher architecture and a cost function which penalises based on Velocity obstacles and a geometry-based heading setpoint. The results show a safe minimum distance to CPA margins in every encounter.
- The architecture reliably classifies traffic encounters and executes effective manoeuvres under most policies except for MPPI-Inland-10. The Own Ship safely performs overtaking and head-on avoidance, and the Crossing Give-Way behaviour generally conforms to COLREGs. However, when this COLREG is detected, the Own Ship comes to a full stop, continuing after the post conditions are met. This is safe but not preferred behaviour, as reducing velocity is energy inefficient.
- Sharp global-path turns near docks occasionally trapped the controller in suboptimal local minima. Changing the nature of the reference global path may mitigate the issue.
- A common reason for failure was the controller reaching a local minimum when near the dock, coming to a standstill between 300 and 50 meters away from the dock.
- Dock-to-dock performance declined significantly when not implementing ancillary controllers. This indicates that the ancillary controllers help the MPPI controller move out of local minima by promoting alternative trajectories. Thereafter, MPPI further refines the chosen path.
- When reducing the MPPI update rate by a factor of ten, sailing behaviour remained acceptable when cruising. However, in critical situations such as traffic encounters and near docks, this decrease in update rate can lead to unwanted behaviour. Measured computation times appear roughly twice as high when MPPI is invoked only once every ten iterations. However, this increase is misrepresented by logging overhead rather than reflecting actual computational cost.
- When switching the projected and post-processed global path used by MPPI-Inland with the simpler interpolated waterway axis, the planner's performance slightly drops to 8 success cases out of 14, indicating that the post-processing steps, as explained in Chapter 3, lead to an improvement in robustness.

6.1.5 How do the global and local planners interact across sailing phases, and what strategies ensure smooth phase transitions and overall mission success?

A hierarchical phase switcher changes the vessel state between the Undocking, Sailing, Approach, Docking and Berthing phases based on dock proximity thresholds. Behaviour in every sailing phase was shown to be sufficient. Most of the runs had terminal errors below 2 m. Remaining issues include:

- During sailing state transitions, a new velocity setpoint is set, causing the vessel to accelerate or decelerate at full capacity. In transitions like from sailing to approach, this behaviour could be excessive and uncomfortable for passengers. Adjusting the transition policies to adopt smoother phase changes to reduce peak G-forces would greatly enhance comfort.

6.1.6 Overall Conclusions

This thesis has presented a modular framework that performs graph-based route planning using Inland ENCs produced by Rijkswaterstaat [39], finds a static global path with starboard preference by projection onto the Fairway object and post-processing refinements. A Biased-MPPI controller uses this global path as a cost heuristic, incorporating various ancillary controllers to encourage moving away from local minima. The method has been shown to produce smooth, starboard-biased trajectories, good dock-to-dock performance and achieve real-time, COLREG-compliant local control within a set of sailing phases.

6.2 Recommendations

The following directions are recommended to extend the contributions presented in this thesis:

Currently, the Own Ship comes to a complete stop when the Crossing Give Way scenario is detected. It then waits until the Target Ship has passed before recommencing course. This is safe, but undesirable behaviour as braking and accelerating leads to more fuel consumption than remaining on course. Therefore, a different policy or a policy that builds on the current policy is recommended for further development to improve upon the proposed method.

The Current solution has only been tested in a simulation. The method has been designed to allow easy redeployment across different vessel types and sizes. To test this robustness, it is recommended to adapt the proposed planner for the Damen Autonomous VEssel (DAVE), a small model vessel with different dynamics than the Damen Waterbus. This helps to assess real-world performance and the robustness of the high-level dynamics.

One of the most common reasons global paths fail feasibility checks is an abrupt change in waterway width. Especially when waypoints are far apart, the chance of line segments intersecting increases. Therefore, it is recommended to develop a method with adaptive adjustment of the global path interpolation step size based on local canal geometry.

Within the scope of this thesis, the sailing phase is the only phase where COLREG-specific behaviour is assumed. To expand on this work, COLREG conditions in other sailing phases need to be defined. For example, waiting at the dock until the area is sufficiently clear when undocking.

The MPPI-Inland-10 method has shown good trajectories while cruising in the sailing phase. However, it does not reach the end-dock in many scenarios. Due to the lowered frequency, there are many cases where the state switcher is misaligned with the MPPI loop, causing many overshoots and oscillations around the dock. A scenario-adaptive MPPI schedule, which dynamically adjusts the planning frequency and increases during docking and COLREG encounters, while lowering during cruising, can be an interesting step forward. This allows the Own Ship to send clear signals to other traffic about its course when cruising while having the same performance as MPPI-Inland when docking or dealing with closer-proximity traffic.

The planning architecture shows good performance in simple continuous trajectories. Issues arise at intersections or areas with complex geometries. One approach to tackling this issue is to identify these situations and change policy accordingly. For example, when near an intersection, removing the global reference and instead setting an Euclidean setpoint after the intersection.

This thesis presents both MPPI-Inland and MPPI-Inland-wtwaxs, which respectively use the projected path and the waterway axis as a global reference trajectory. One interesting adaptation could be to combine these methods with a slight adaptation to the cost function, using the waterway axis as a global reference in terms of along-track distance, while minimising the cross-track error for the projected path.

Given the current state of autonomous shipping research, it may be better to define dedicated routes instead of relying only on algorithmic extraction. Autonomous shipping routes or shipping lanes with standardised rules for geometry, width, offsets, and continuity would avoid issues found in this thesis, such as projection errors and breaks in the waterway axis. In that case, the global planning problem is addressed, allowing autonomous shipping research to focus on safely following a predefined path while adhering to various complex traffic scenarios.

Currently, evaluation is limited to the three most common COLREG situations, while many other traffic scenarios remain to be tested. Pedersen [36] has listed 55 different multi-vessel scenarios for maritime traffic interaction. To ensure robust COLREG-compliant traffic avoidance in all cases, additional policies should be designed to provide all possible scenarios are met.

Bibliography

[1] Chris Bonelli and Caroline Petrow-Cohen. Waymo expands service area in Los Angeles and San Francisco. Los Angeles Times. URL https://www.latimes.com/business/story/2025-06-17/waymo-expands-service-area-in-los-angeles-and-san-francisco.

[2] Jürgen Brandstetter, Péter Rácz, Clay Beckner, Eduardo B. Sandoval, Jennifer Hay, and Christoph Bartneck. A peer pressure experiment: Recreation of the asch conformity experiment with robots. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1335–1340, 2014. doi: 10.1109/IROS.2014.6942730.

[3] Shengshi Cao, Pingyi Fan, Tao Yan, Cheng Xie, Jian Deng, Feng Xu, and Yaqing Shu. Inland waterway ship path planning based on improved rrt algorithm. Journal of Marine Science and Engineering, 10(10):1460, 2022.

[4] McKinsey Company. Autonomous driving's future: Convenient and connected, 2023. URL https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected.

[5] E. W. Dijkstra. A note of two problems in connection with graphs. Numerische Matematik, 1:269–271, 1959.

[6] Rahul Dubey and Sushil J Louis. Vorrt-colregs: A hybrid velocity obstacles and rrt based colregs-compliant path planner for autonomous surface vessels. In OCEANS 2021: San Diego–Porto, pages 1–8. IEEE, 2021.

[7] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of mathematics, 79(3):497–516, 1957.

[8] Thomas T Enevoldsen and Roberto Galeazzi. Grounding-aware rrt for path planning and safe navigation of marine crafts in confined waters. IFAC-PapersOnLine, 54(16):195–201, 2021.

[9] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. The international journal of robotics research, 17(7):760–772, 1998.

[10] Manan S Gandhi, Bogdan Vlahov, Jason Gibson, Grady Williams, and Evangelos A Theodorou. Robust model predictive path integral control: Analysis and performance guarantees. IEEE Robotics and Automation Letters, 6(2):1423–1430, 2021.

[11] Inger Berge Hagen. Topics on marine collision avoidance. 2022.

[12] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

[13] Hongwei He, Marc Mansuy, Jeroen Verwilligen, Guillaume Delefortrie, and Evert Lataire. Global path planning for inland vessels based on fast marching algorithm. Ocean Engineering, 312:119172, 2024.

[14] R. Herrmann, S. Bose, I. Filip, D. Medina, R. Ziebold, S. Gehrig, T. Lenhard, and M. Gardill. Near-range environmental perception for inland waterway vessels: A comparative study of lidar and automotive fmcw radar sensors. In Proceedings of the Positioning and Navigation for Intelligent Transport Systems (POSNAV 2024). DGON, November 2024. Conference paper included in POSNAV 2024.

[15] Hannes Homburger, Stefan Wirtensohn, Moritz Diehl, and Johannes Reuter. Energy-optimal planning and shrinking horizon mpc for vessel docking in river current fields. In 2024 European Control Conference (ECC), pages 1125–1130. IEEE, 2024.

[16] Liang Hu, Huosheng Hu, Wasif Naeem, and Zidong Wang. A review on colregs-compliant navigation of autonomous surface vehicles: From traditional to learning-based approaches. Journal of Automation and Intelligence, 1(1):100003, 2022.

[17] Xiao Hu, Kai Hu, Datian Tao, Yi Zhong, and Yi Han. Gis-data-driven efficient and safe path planning for autonomous ships in maritime transportation. Electronics, 12(10):2206, 2023.

[18] Coherent Market Insights. Autonomous vessels market, by type (fully autonomous, semi-autonomous), 2025. URL https://www.coherentmarketinsights.com/industry-reports/autonomous-vessels-market.

[19] Fortune Business Insights. Autonomous ship market size, share trends [2023-2032], 2023. URL https://www.fortunebusinessinsights.com/industry-reports/autonomous-ship-market-101797.

[20] International Hydrographic Organization. S-57 Appendix B.1 – ENC Product Specification. URL https://iho.int/en/standards-and-specifications.

[21] International Maritime Organization. Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs), October 20 1972. URL https://www.imo.org/en/About/Conventions/Pages/International-Regulations-for-Preventing-Collisions-at-Sea-(COLREG).aspx. Adopted 20 October 1972; entered into force 15 July 1977.

[22] International Maritime Organization. Convention on the International Regulations for Preventing Collisions at Sea, 1972: Consolidated Edition 2018. International Maritime Organization, London, consolidated edition edition, 2018. ISBN 9789280116309. IMO publication number: 200E.

[23] Walter Jansma, Elia Trevisan, Álvaro Serra-Gómez, and Javier Alonso-Mora. Interaction-aware sampling-based mpc with learned local goal predictions. In 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), pages 15–21. IEEE, 2023.

[24] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. The international journal of robotics research, 30(7):846–894, 2011.

[25] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, 5(1):90–98, 1986. doi: 10.1177/027836498600500106. URL https://doi.org/10.1177/027836498600500106.

[26] Hak-Chan Kim, Woo-Ju Son, Jeong-Seok Lee, and Ik-Soon Cho. Identification of maritime areas with high vessel traffic based on polygon shape similarity. IEEE Access, 2024.

[27] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. Research Report 9811, 1998.

[28] Huanhuan Li and Zaili Yang. Incorporation of ais data-based machine learning into unsupervised route planning for maritime autonomous surface ships. Transportation Research Part E: Logistics and Transportation Review, 176:103171–103171, 2023.

[29] Hongguang Lyu and Yong Yin. Colregs-constrained real-time path planning for autonomous ships using modified artificial potential fields. The Journal of navigation, 72(3):588–608, 2019.

[30] Market.us. Autonomous vehicles statistics - market size growth 2023, 2023. URL https://www.news.market.us/autonomous-vehicles-statistics/.

[31] Eivind Meyer, Amalie Heiberg, Adil Rasheed, and Omer San. Colreg-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning. Ieee Access, 8: 165344–165364, 2020.

[32] National Oceanic and Atmospheric Administration. NOAA ENC® – Electronic Navigational Charts. URL https://nauticalcharts.noaa.gov/charts/noaa-enc.html.

[33] Godert Notten. Risk-aware motion planning: towards safe autonomous surface vessels. Msc thesis, Delft University of Technology, 2023.

[34] Michael Otte and Emilio Frazzoli. Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning. The International Journal of Robotics Research, 35(7):797–822, 2016.

[35] Shinkyu Park, Michal Cap, Javier Alonso-Mora, Carlo Ratti, and Daniela Rus. Social trajectory planning for urban autonomous surface vessels. IEEE Transactions on Robotics, 37(2):452–465, 2020.

[36] Tom Arne Pedersen, Chanjei Vasanthan, Kristian Karolius, Øystein Engelhardtsen, Koen Pieter Houweling, and Are Jørgensen. Generating structured set of encounters for verifying automated collision and grounding avoidance systems. In Journal of Physics: Conference Series, volume 2618, page 012013. IOP Publishing, 2023.

[37] Rajesh Rajamani. Vehicle Dynamics and Control. Springer Science & Business Media, 2012. doi: 10.1007/978-1-4614-1433-9.

[38] Rijkswaterstaat. Specificaties IENC-NL: Specificaties van en codeerinstructies voor IENC's in Nederland op basis van IES 2.5.1, 3.0 edition, December 2021. Final edition based on IES 2.5.1 with feedback and harmonization with RIS Index NL 3.0.

[39] Rijkswaterstaat. Elektronische vaarwegkaarten. https://www.rijkswaterstaat.nl/zakelijk/zakendoen-met-rijkswaterstaat/werkwijzen/werkwijze-in-gww/data-eisen-rijkswaterstaat contracten/elektronische-vaarwegkaarten, 2025.

[40] Anthony Stentz. Optimal and efficient path planning for unknown and dynamic environments. In Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pages 3310–3317, 1994.

[41] Lucas Streichenberg, Elia Trevisan, Jen Jen Chung, Roland Siegwart, and Javier Alonso-Mora. Multi-agent path integral control for interaction-aware motion planning in urban canals. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 1379–1385. IEEE, 2023.

[42] Trym Tengesdal, Sébastien Gros, and Tor A Johansen. Real-time feasible usage of radial basis functions for representing unstructured environments in optimal ship control this work was supported by the research council of norway through the autoship centre for research-based innovation, project number 309230, and by kongsberg maritime through the university technology center on ship performance and cyber physical systems. In 2024 American Control Conference (ACC), pages 4050–4057. IEEE, 2024.

[43] Elia Trevisan. Model Predictive Path Integral Control for Interaction-Rich Local Motion Planning in Dynamic Environments. Phd thesis, Delft University of Technology, 2025.

[44] Elia Trevisan and Javier Alonso-Mora. Biased-mppi: Informing sampling-based model predictive control by fusing ancillary controllers. IEEE Robotics and Automation Letters, 2024.

[45] Jelle Vogel. Modular trajectory planning for autonomous surface vessels in inland waters using electronic nautical charts - literature review. Unpublished literature review, Delft University of Technology, 2025.

[46] Gang Wang, Jingheng Wang, Xiaoyuan Wang, Quanzheng Wang, Junyan Han, Longfei Chen, and Kai Feng. A method for coastal global route planning of unmanned ships based on human-like thinking. Journal of Marine Science and Engineering, 12(3):476, 2024.

[47] Wei Wang, Banti Gheneti, Luis A Mateos, Fabio Duarte, Carlo Ratti, and Daniela Rus. Roboat: An autonomous surface vehicle for urban waterways. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6340–6347. IEEE, 2019.

[48] Wei Wang, Tixiao Shan, Pietro Leoni, David Fernández-Gutiérrez, Drew Meyers, Carlo Ratti, and Daniela Rus. Roboat ii: A novel autonomous surface vessel for urban environments. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1740–1747. IEEE, 2020.

[49] Wei Wang, David Fernández-Gutiérrez, Rens Doornbusch, Joshua Jordan, Tixiao Shan, Pietro Leoni, Niklas Hagemann, Jonathan Klein Schiphorst, Fabio Duarte, Carlo Ratti, et al. Roboat iii: An autonomous surface vessel for urban transportation. Journal of Field Robotics, 40(8):1996–2009, 2023.

[50] Zhenyang Wang, Ping Yang, Diju Gao, and Chunteng Bao. Path-planning algorithm based on elastic force contractions for autonomous navigation of unmanned container ships in waterborne transportation. Ocean Engineering, 310:118646, 2024.

[51] Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. arXiv preprint arXiv:1509.01149, 2015.

[52] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. Journal of Guidance, Control, and Dynamics, 40(2):344–357, 2017.

[53] Liu Xiaodong and Zhang Qingling. H/sub /spl infin// control for t-s fuzzy systems: Lmi approach. In Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), volume 2, pages 987–988 vol.2, 2002. doi: 10.1109/ACC.2002.1023146.

[54] Tie Xu, Jun Ma, Peiqiang Qin, and Qinyou Hu. An improved rrt∗ algorithm based on adaptive informed sample strategy for coastal ship path planning. Ocean Engineering, 333: 121511, 2025.

[55] Lichao Yang, Jingxian Liu, Zhao Liu, Yukuan Wang, Yang Liu, and Qin Zhou. Ship global path planning using jump point search and maritime traffic route extraction. Expert Systems with Applications, 284:127885, 2025.

[56] Ji Yin, Charles Dawson, Chuchu Fan, and Panagiotis Tsiotras. Shield model predictive path integral: A computationally efficient robust mpc method using control barrier functions. IEEE Robotics and Automation Letters, 8(11):7106–7113, 2023.

[57] Sang-Woong Yun, Dong-Ham Kim, Se-Won Kim, Dong-Jin Kim, and Hye-Jin Kim. Global path planning for autonomous ship navigation considering the practical characteristics of the port of ulsan. Journal of Marine Science and Engineering, 12(1):160, 2024.

[58] Raphael Zaccone. Colreg-compliant optimal path planning for real-time guidance and control of autonomous ships. Journal of Marine Science and Engineering, 9(4):405, 2021.

[59] Jinfen Zhang, Han Zhang, Jiongjiong Liu, Da Wu, and C Guedes Soares. A two-stage path planning algorithm based on rapid-exploring random tree for ships navigating in multi-obstacle water areas considering colregs. Journal of Marine Science and Engineering, 10 (10):1441, 2022.

[60] Rong Zhen, Qiyong Gu, Ziqiang Shi, and Yongfeng Suo. An improved a-star ship path-planning algorithm considering current, water depth, and traffic separation rules. Journal of Marine Science and Engineering, 11(7):1439, 2023.

Appendix A

Glossary

This appendix gives an overview of frequently used terms and abbreviations.

AIS: Automatic Identification System – a maritime communication system used for tracking ship positions and navigation data.

APPROACH: A sailing phase in which the vessel slows down and begins alignment with the docking point, transitioning from cruising to final docking.

ASV: Autonomous Surface Vessel – a watercraft capable of operating and navigating without human intervention.

ATD: Along Track Distance – cumulative line distance of a trajectory.

BERTHING: Final sailing phase where the vessel holds its position at the dock using minimal movement and lateral corrections.

COLREGs: International Regulations for Preventing Collisions at Sea – a set of navigational rules ensuring safe vessel encounters.

CPA: Closest Point of Approach – the minimum distance between two vessels if they continue on their current paths.

CTE: Cross Track Error – lateral offset perpendicular to a trajectory.

DAVE: Damen Autonomous Vessel.

DCPA: Distance at the Closest Point of Approach – between the Own Ship and the Target Ship.

DOCKING: Sailing phase where the vessel moves into the berthing position in a controlled manner.

ENC: Electronic Navigational Chart – digital maps in standardized formats (e.g., S-57) used for marine navigation.

GNC: Guidance, Navigation, and Control – a control framework governing how a vessel determines, follows, and maintains its path.

GPS: Global Positioning System.

HO: Head-On – a COLREG-defined scenario where two vessels approach each other directly and must take action to avoid collision.

IENC: Inland Electronic Navigational Chart – inland digital maps in standardized formats (e.g., S-57) used for marine navigation.

LOS: Line of Sight – line from the centre of the Own Ship to the centre of the Target Ship.

Lookahead Point: Point a set distance ahead across the global path.

MPC: Model Predictive Control – a control algorithm that minimizes a cost function given a dynamic model.

MPPI: Model Predictive Path Integral – a control algorithm that samples trajectories to minimize cost over a planning horizon.

OS: Own Ship – the autonomous vessel controlled by the system and under evaluation in the planning framework.

OT-GW: Overtaking Give-Way – a COLREG-defined scenario where one vessel overtakes another and must ensure safe passing.

SAILING: Sailing phase in which the vessel follows the global path at cruising speed under normal navigation conditions.

TCPA: Time to Closest Point of Approach – the amount of time it takes for the Own Ship to reach the Closest Point of Approach.

TS: Target Ship – the other vessel considered in encounter scenarios such as collision avoidance under COLREGs.

UNDOCKING: Initial phase where the vessel departs from the dock, typically by lateral movement before entering cruising state.

VO: Velocity Obstacle – the set of velocities which results in a collision with an obstacle.

XTR: Cross Track Error – lateral offset perpendicular to a trajectory.

Appendix B

Preconditions

Scenario	α-condition	β-condition	Other Conditions
Crossing Give-Way (CR-GW)	$0 < \alpha_{180} < \theta_{15}^{\min}$	$-\theta_{15}^{\min} < \beta_{180} < \theta_{15}$	$d < R_{\mathrm{CR\_GW}}$
Crossing Stay-On (CR-SO)	$-\theta_{15}^{\min} < \alpha_{180} < \theta_{15}$	$0 < \beta_{360} < \theta_{15}^{\min}$	$d < R_{\mathrm{CR\_GW}}$
Overtaking Give-Way (OT-GW)	$\theta_{15}^{\min} < \alpha_{360} < \theta_{15}^{\max}$	$-\theta_{13} < \beta_{180} < \theta_{13}$	$v_{\mathrm{own}} > v_{\mathrm{target}}$, $d < R_{\mathrm{OT\_GW}}$
Overtaking Stay-On (OT-SO)	$-\theta_{13} < \alpha_{180} < \theta_{13}$	$\theta_{15}^{\min} < \beta_{360} < \theta_{15}^{\max}$	$v_{\mathrm{own}} < v_{\mathrm{target}}$, $d < R_{\mathrm{OT\_GW}}$
Head-On (HO)	$-\theta_{14} < \alpha_{180} < \theta_{14}$	$-\theta_{14} < \beta_{180} < \theta_{14}$	$d < R_{\mathrm{HO}}$

Table B.1: Preconditions for each COLREG encounter, where $\alpha_{180} \in [-180, 180)$, $\alpha_{360} \in [0, 360)$, $\beta_{180} \in [-180, 180)$, $\beta_{360} \in [0, 360)$, $\theta_{13}, \theta_{14}, \theta_{15}$ are critical angle thresholds, d is the Own–Target distance, and $R_{\cdot}$ the encounter radii.

Variable definitions and values:

- $\alpha_{180}, \alpha_{360}$: Angle between the Target Ship's longitudinal axis and the line-of-sight (LOS) to the Own Ship, wrapped into $[-180°, 180°)$ and $[0°, 360°)$, respectively.
- β_{180}, β_{360}: Angle between the Own Ship's heading and the LOS to the Target Ship, wrapped into $[-180°, 180°)$ and $[0°, 360°)$, respectively.
- $\theta_{13} = 45°$: Critical crossing/overtaking threshold. [11]
- $\theta_{14} = 13°$: Critical head-on threshold. [11]
- $\theta_{15} = 10°$: Threshold for Crossing Give-Way. [11]
- $\theta_{15}^{\min} = 120°$, $\theta_{15}^{\max} = 240°$: Minimum and maximum thresholds behind the Own Ship. [11]
- d: Euclidean distance between the centres of the Own Ship and Target Ship, in meters.
- $R_{safety} = 30$ m: Safety margin around the Target Ship.
- $R_{\mathrm{CR\_GW}} = R_{\mathrm{OT\_GW}} = 8 \times R_{safety} = 240$ m, $\quad R_{\mathrm{HO}} = 30 \times R_{safety} = 900$ m: Encounter detection radii.
- $v_{\mathrm{own}}, v_{\mathrm{target}}$: Forward speeds of the Own and Target vessels, in m/s.

Appendix C

Parameters of Ancillary controllers

Table C.1: Summary of PD Gains, Values and Effects of the Ancillary controllers used in Biased-MPPI.

Parameter	Symbol	Value	Effect on System Behaviour
Lateral proportional gain	K_p^l	1.0	Increases responsiveness to cross-track error; larger values result in stronger lateral corrections.
Lateral derivative gain	K_d^l	0.1	Dampens lateral oscillations by reacting to the rate of change in cross-track error.
Angular proportional gain	K_p^ω	2.0	Increases aggressiveness of heading correction.
Angular derivative gain	K_d^ω	0.1	Reduces overshoot and smooths angular responses.
Velocity proportional gain	K_p^v	5.0	Adjusts responsiveness in velocity tracking.
Maximum braking deceleration	a_{max}	$0.572\,\frac{\text{m}}{\text{s}^2}$	Limits deceleration rate during emergency braking (set by vessel dynamics).
Heading deadband	θ_{th}	0.05 rad	Prevents overreaction to small heading deviations.
Lateral deadband	δ_{lat}	1.0 m	Prevents unnecessary corrections when cross-track error is small.

Appendix D

MPPI Cost Function Configuration Parameters

$$\begin{aligned}
J(\mathbf{s}) = {} & w_{\text{valid}} J_{\text{valid}}(\mathbf{s}) + w_{\text{dist}} J_{\text{dist\_to\_goal}}(\mathbf{s}) + w_{\text{vel}} J_{\text{velocity}}(\mathbf{s}) \\
& + w_{\text{lat\_vel}} J_{\text{lat\_velocity}}(\mathbf{s}) + w_{\text{heading}} J_{\text{heading}}(\mathbf{s}) + w_{\text{yaw}} J_{\text{yaw\_rate}}(\mathbf{s}) \\
& + J_{\text{global\_path}}(\mathbf{s}) + J_{\text{colreg}}(\mathbf{s})
\end{aligned} \tag{D.1}$$

Table D.1: Phase-Independent MPPI Weights

Parameter	Symbol	Value	Effect
Validity penalty weight	w_{valid}	10	Ensures states outside the workspace incur high cost (Eq. 4.25).
Yaw-rate penalty weight	w_{yaw}	5	Discourages sudden heading changes (Eq. 4.30).
Deceleration penalty weight	w_a	1	Penalizes strong braking deceleration.
Braking-zone radius	z_b	10 m	Within this distance of the goal, deceleration cost applies.
Lookahead distance	d_{LA}	500 m	Distance ahead along the path for dynamic goal placement.

Table D.2: Phase-Dependent MPPI Weights

Cost	Symbol	Undocking	Sailing	Approach	Docking	Berthing	Effect
Distance-to-goal	w_{dist}	0	0	5	1	2	Scales $J_{\text{dist\_to\_goal}}$ (Eq. 4.26).
Forward-speed deviation	w_{vel}	1	1	0.5	1	1	Scales J_{velocity} (Eq. 4.27).
Lateral-speed deviation	$w_{\text{lat\_vel}}$	1	0	0	1	1	Scales $J_{\text{lat\_velocity}}$ (Eq. 4.28).
Heading alignment	w_{heading}	1	0	0	2.5	4	Scales J_{heading} fade-in (Eq. 4.29).
Along-track error	w_{atd}	0	1.0	0	0	0	Scales ATD in $J_{\text{global\_path}}$ (Eq. 4.31).
Cross-track error	w_{xtr}	0	2.0	0	0	0	Scales XTR in $J_{\text{global\_path}}$ (Eq. 4.31).

Table D.3: Phase-dependent speed setpoints used in MPPI.

Quantity [m/s]	Undocking	Sailing	Approach	Docking	Berthing
Forward speed setpoint, v_{set}	0	9.5	3	1	0
Lateral speed setpoint, $v_{\text{lat,set}}$	1	0	0	1	0

Table D.4: Radii used for phase switching.

Parameter	Symbol	Value [m]
Undocking radius	r_{undock}	25
Approach radius	r_{approach}	300
Docking radius	r_{dock}	50
Berthing radius	r_{berth}	10

Table D.5: MPPI configuration.

Parameter	Value	Notes
Time step, Δt [s]	1	Integration step.
Simulation steps, T	550	Steps per simulation rollout.
Samples, N	300	Number of control sequences per update.
Horizon, H	50	Prediction horizon (controls per sequence).
λ	0.5	Tuning parameter allowing for different weight discrepancies.
Control lower bound, $\mathbf{u}_{\text{min}}$	$[-7.2, -1.5, -0.087266]$	Thrusters & yaw-rate limits (last entry ≈ −5°/s).
Control upper bound, $\mathbf{u}_{\text{max}}$	$[10.2889, 1.5, 0.087266]$	Last entry ≈ 5°/s.
Noise covariance, Σ	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$	Sampling noise on controls.

Appendix E

Detailed Test results

This Appendix provides testresults for all scenarios in greater detail.

Table E.1: Performance metrics corresponding to the MPPI-IL method given each evaluated Dock-to-Dock scenario.

scenario	total path length [m]	AIS mean traj length [m]	global path length [m]	total travel time [s]	AIS mean travel time [s]	Terminal Error [m]	min radius phase [m]	AIS mean min radius [m]	auc ang acc sailing [Rad/s]	AIS mean avg XTE [m]	AIS mean max XTE [m]
de schans-kade	3671	3541	3504	491	386.9	8.92	67.76	172.2	0.5412	36.89	117.9
de schans-molenkade	746.8	815.5	696.5	120	170	9.032	30.64	2.771	0.26	27.22	39.27
de schans-rijsdijk	873.1	-	917.5	187	-	9.006	14.39	-	0.295	-	-
kade-de schans	3636	3470	3500	489	364.9	9.452	21.96	161	0.2696	39.31	109.5
kade-noordeinde	1282	1294	1194	219	210.5	9.569	24.25	202.3	0.2371	20.07	57.16
merwedekade-veerplein	1348	1283	1450	244	264.1	8.869	17.44	7.537	0.307	30.1	80.96
merwedekade-westeind	1727	1767	1522	281	250	9.825	0	23.13	0.4146	35.57	96.07
merwedekade-veerdam	709.3	695.3	384.6	276	180.5	9.263	32.65	75.6	0.1065	56.79	118.6
molenkade-de schans	609.8	-	829.9	512	-	290.8	59.6	-	0.2719	-	-
noordeinde-kade	1301	1310	1204	221	198	9.605	49.48	380.1	0.2115	27.62	64.81
noordeinde-westeind	2476	2398	2350	517	404.3	50.64	27	26.94	0.2827	50.16	143.6
veerdam-merwedekade	704.3	-	376	523	-	255.5	52.91	-	0.08052	-	-
veerplein-merwedekade	1533	1128	1541	394	248.9	9.357	0	100.9	0.3703	64.32	154.2
westeind-noordeinde	2552	-	2448	405	-	9.267	0	-	0.437	-	-
mean	1655	1770	1565	348.5	267.8	49.94	28.44	115.2	0.2918	38.8	98.2
std	1044	1031	1027	142.2	87.09	95.45	22.17	118	0.1225	14.16	37.57

Table E.2: Performance metrics corresponding to MPPI-IL-NP method given each evaluated Dock-to-Dock scenario.

scenario	total path length [m]	AIS mean traj length [m]	global path length [m]	total travel time [s]	AIS mean travel time [s]	Terminal Error [m]	min radius phase [m]	AIS mean min radius [m]	auc ang acc sailing [Rad/s]	AIS mean avg XTE [m]	AIS mean max XTE [m]
de schans-kade	3678	3541	3504	515	386.9	82.68	73.58	172.2	0.5478	39.51	112.3
de schans-molenkade	2947	815.5	696.5	515	170	3143	0	2.771	0.2082	211.8	618.3
de schans-rijsdijk	2969	-	917.5	512	-	3138	0	-	0.2252	-	-
kade-de schans	3672	3470	3500	511	364.9	128.4	114.9	161	0.4804	44.34	123.4
kade-noordeinde	1396	1294	1194	236	210.5	4.269	125.5	202.3	0.2094	27.8	83.9
merwedekade-veerplein	1403	1283	1450	222	264.1	7	107.2	7.537	0.2347	27.52	73.27
merwedekade-westeind	1946	1767	1522	512	250	50.91	78.77	23.13	0.3737	70.91	209.4
merwedekade-veerdam	4501	695.3	384.6	513	180.5	3797	124.1	75.6	0.6118	226.8	477.6
molenkade-de schans	829.9	-	829.9	513	-	271.4	39.84	-	0.2381	-	-
noordeinde-kade	1416	1310	1204	166	198	6.208	138.2	380.1	0.1308	59.84	123.8
noordeinde-westeind	2444	2398	2350	515	404.3	78.78	85.55	26.94	0.3768	55.27	144.2
veerdam-merwedekade	732.9	-	376	523	-	293.9	83.48	-	0.1335	-	-
veerplein-merwedekade	1119	1128	1541	518	248.9	317.2	0	100.9	0.3257	83.92	263.9
westeind-noordeinde	2568	-	2448	514	-	79.21	22.8	-	0.4542	-	-
mean	2259	1770	1565	448.9	267.8	814.2	71	115.2	0.325	84.77	223
std	1180	1031	1027	131.4	87.09	1391	49.92	118	0.1531	73.18	183.5

Table E.3: Performance metrics corresponding to the MPPI-10 method given each evaluated Dock-to-Dock scenario.

scenario	total path length [m]	AIS mean traj length [m]	global path length [m]	total travel time [s]	AIS mean travel time [s]	Terminal Error [m]	min radius phase [m]	AIS mean min radius [m]	auc ang acc sailing [Rad/s]	AIS mean avg XTE [m]	AIS mean max XTE [m]
de schans-kade	3641	3541	3504	442	386.9	38.06	15.2	172.2	0.9286	36.15	109.7
de schans-molenkade	203.7	815.5	696.5	506	170	505.3	38.43	2.771	0.1516	163.4	485.2
de schans-rijsdijk	450.9	-	917.5	508	-	675.6	0	-	0.6975	-	-
kade-de schans	3646	3470	3500	463	364.9	21.48	257.1	161	0.7141	39.32	116.3
kade-noordeinde	1457	1294	1194	403	210.5	4.115	23.56	202.3	0.1791	16.97	47.39
merwedekade-veerplein	1260	1283	1450	453	264.1	23.05	0	7.537	0.7616	47.14	112.9
merwedekade-westeind	1832	1767	1522	515	250	87.28	42.58	23.13	0.5386	30.66	82.54
merwedekade-veerdam	810.3	695.3	384.6	413	180.5	8.394	25.38	75.6	0.1861	54.31	98.81
molenkade-de schans	789.1	-	829.9	477	-	182.2	31.42	-	0.1544	-	-
noordeinde-kade	1394	1310	1204	459	198	2.577	47.58	380.1	0.2219	29.13	64.49
noordeinde-westeind	2517	2398	2350	419	404.3	50.95	319.1	26.94	0.6254	45.95	146.1
veerdam-merwedekade	513.1	-	376	522	-	245.4	25.38	-	0.1889	-	-
veerplein-merwedekade	1490	1128	1541	520	248.9	34.25	0	100.9	0.8757	68.37	154.7
westeind-noordeinde	62.36	-	2448	518	-	2270	0	-	0.3183	-	-
mean	1433	1770	1565	472.7	267.8	296.4	58.98	115.2	0.4673	53.14	141.8
std	1148	1031	1027	42.79	87.09	604.1	99.16	118	0.2948	41.29	125.1

Table E.4: Performance metrics corresponding to the MPPI-Inland-wtwaxs method given each evaluated Dock-to-Dock scenario.

scenario	total path length [m]	AIS mean traj length [m]	global path length [m]	total travel time [s]	AIS mean travel time [s]	Terminal Error [m]	min radius phase [m]	AIS mean min radius [m]	auc ang acc sailing [Rad/s]	AIS mean avg XTE [m]	AIS mean max XTE [m]
de schans-kade	3621	3541	3499	467	386.9	9.191	23.87	172.2	0.6108	41.68	77.75
de schans-molenkade	895.7	815.5	770.2	197	170	9.651	30.36	2.771	0.3456	24.27	66.49
de schans-rijsdijk	988.3	-	970.2	520	-	50.51	17.88	-	0.3252	-	-
kade-de schans	3620	3470	3499	464	364.9	9.857	23.87	161	0.5598	40.26	78.91
kade-noordeinde	1294	1294	1200	218	210.5	7.522	18.46	202.3	0.2965	45.39	70.11
merwedekade-veerplein	1274	1283	1383	203	264.1	8.733	11.95	7.537	0.3554	50.79	94.8
merwedekade-westeind	1938	1767	1522	463	250	174.2	113.3	23.13	0.1692	66.45	115
merwedekade-veerdam	683.8	695.3	398.9	520	180.5	50.62	0	75.6	0.1942	66.53	103.4
molenkade-de schans	860.3	-	779.4	521	-	262.1	14.13	-	0.1632	-	-
noordeinde-kade	1295	1310	1200	215	198	9.119	19.69	380.1	0.2975	38.23	70.95
noordeinde-westeind	2533	2398	2399	427	404.3	9.032	28.75	26.94	0.4575	57.49	93.78
veerdam-merwedekade	694.1	-	397	523	-	50.89	68.98	-	0.1819	-	-
veerplein-merwedekade	1559	1128	1552	304	248.9	9.492	29.86	100.9	0.3949	79.54	210.9
westeind-noordeinde	115.2	-	2399	518	-	2187	0	-	0.2147	-	-
mean	1527	1770	1569	397.1	267.8	203.4	28.65	115.2	0.3262	51.06	98.21
std	1062	1031	1019	136.5	87.09	575.9	29.44	118	0.1421	16.49	42.64

Appendix F

Full grid Search Results

This appendix reports the full grid-search test results referenced in Section 3.3. For each scenario, the tables list the parameter ranges that produced paths satisfying all requirements. Columns: SS (interpolation step size, m), OS (shoreline offset, m), WS_{min} (minimum-distance window, points), WS_{avg} (average-distance window, points), and AvgIt (number of averaging iterations). Ranges are inclusive; hyphens denote continuous intervals over the tested grid and commas denote discrete values (e.g., “100, 140–160, 260”). Entries shown as “nan–nan” indicate that no feasible settings were found for that path type in the given scenario or that the path type does not the option to adjust the range of the corresponding metric.

Table F.1: Successful setting ranges for scenario Dordrecht

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected_path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed_avg_path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed_min_avg_path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed_min_path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan

Table F.2: Successful setting ranges for scenario Dordrecht Flipped

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected_path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed_avg_path	100.0, 140.0 - 160.0, 220.0, 260.0	5.0	nan - nan	5.0 - 20.0	5.0 - 50.0
windowed_min_avg_path	100.0, 140.0 - 160.0, 260.0	0.0 - 10.0	5.0 - 15.0	5.0 - 20.0	1.0 - 50.0
windowed_min_path	100.0 - 140.0, 260.0	10.0	5.0 - 15.0	nan - nan	nan - nan

Table F.3: Successful setting ranges for scenario Hardinxveld

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected_path	60.0, 100.0 - 120.0, 160.0 - 260.0	10.0 - 30.0	nan - nan	nan - nan	nan - nan
windowed_avg_path	60.0 - 260.0	5.0 - 30.0	nan - nan	5.0 - 20.0	1.0 - 50.0
windowed_min_avg_path	60.0 - 260.0	5.0 - 30.0	5.0 - 15.0	5.0 - 20.0	1.0 - 50.0
windowed_min_path	60.0 - 260.0	5.0 - 30.0	5.0 - 15.0	nan - nan	nan - nan

Table F.4: Successful setting ranges for scenario Hardinxveld Flipped

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected_path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed_avg_path	60.0 - 220.0	5.0 - 30.0	nan - nan	5.0 - 20.0	1.0 - 50.0
windowed_min_avg_path	60.0 - 220.0	5.0 - 30.0	5.0 - 15.0	5.0 - 20.0	1.0 - 50.0
windowed_min_path	60.0 - 220.0	5.0 - 30.0	5.0 - 15.0	nan - nan	nan - nan

Table F.5: Successful setting ranges for scenario Nieuwe Maas

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan

Table F.6: Successful setting ranges for scenario Nieuwe Maas Flipped

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan

Table F.7: Successful setting ranges for scenario Sliedrecht Intersection Left

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected__path	140.0 - 160.0, 220.0, 260.0	20.0 - 30.0	nan - nan	nan - nan	nan - nan
windowed__avg__path	60.0 - 260.0	5.0 - 30.0	nan - nan	5.0 - 20.0	1.0 - 50.0
windowed__min__avg__path	60.0 - 260.0	5.0 - 30.0	5.0 - 15.0	5.0 - 20.0	1.0 - 50.0
windowed__min__path	60.0 - 260.0	5.0 - 30.0	5.0 - 15.0	nan - nan	nan - nan

Table F.8: Successful setting ranges for scenario Sliedrecht Intersection Right

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected__path	80.0 - 260.0	5.0 - 30.0	nan - nan	nan - nan	nan - nan
windowed__avg__path	80.0 - 260.0	5.0 - 30.0	nan - nan	5.0 - 20.0	1.0 - 50.0
windowed__min__avg__path	80.0 - 260.0	5.0 - 30.0	5.0 - 15.0	5.0 - 20.0	1.0 - 50.0
windowed__min__path	80.0 - 260.0	5.0 - 30.0	5.0 - 15.0	nan - nan	nan - nan

Table F.9: Successful setting ranges for scenario Sophiapolder

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan

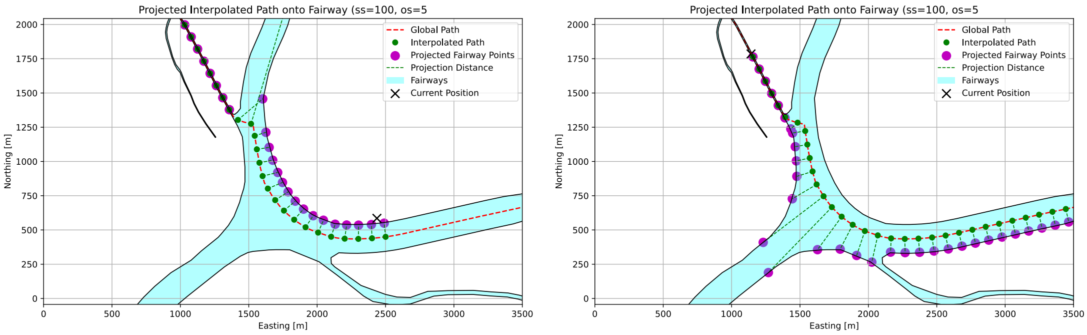
Table F.10: Successful setting ranges for scenario Sophiapolder Flipped

Path Type	SS Range	OS Range	WSmin Range	WSavg Range	Avgit Range
projected__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__avg__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan
windowed__min__path	nan - nan	nan - nan	nan - nan	nan - nan	nan - nan

Appendix G

Scenarios

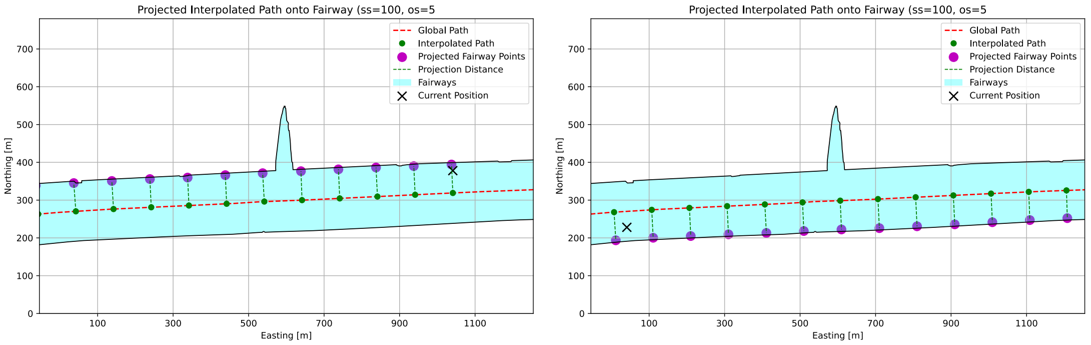
In this appendix an overview of the scenarios used for evaluation of global path planning methods is presented. The scenario's shown also contain an example trajectory using the projection method at a step size of 100 meters.



Projected Interpolated Path onto Fairway (ss=100, os=5

Projected Interpolated Path onto Fairway (ss=100, os=5

Figure G.1: Dordrecht and Dordrecht Flipped projected paths



Projected Interpolated Path onto Fairway (ss=100, os=5

Projected Interpolated Path onto Fairway (ss=100, os=5

Figure G.2: Hardinxveld Projected Paths

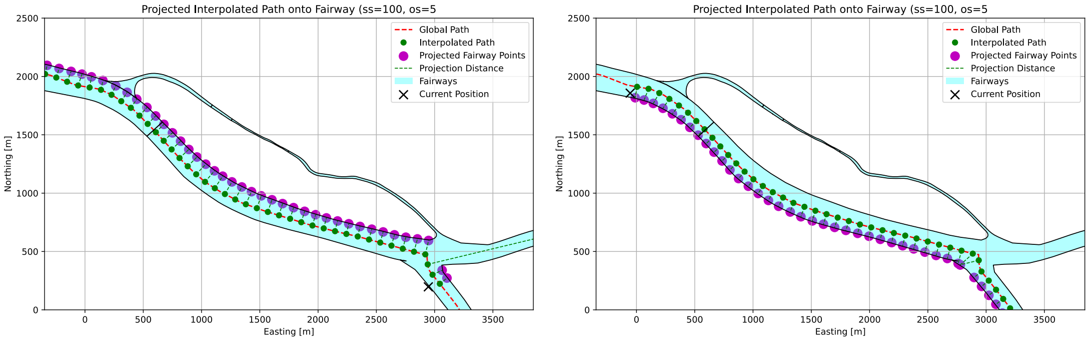


Figure G.3: Nieuwe Maas Projected Paths

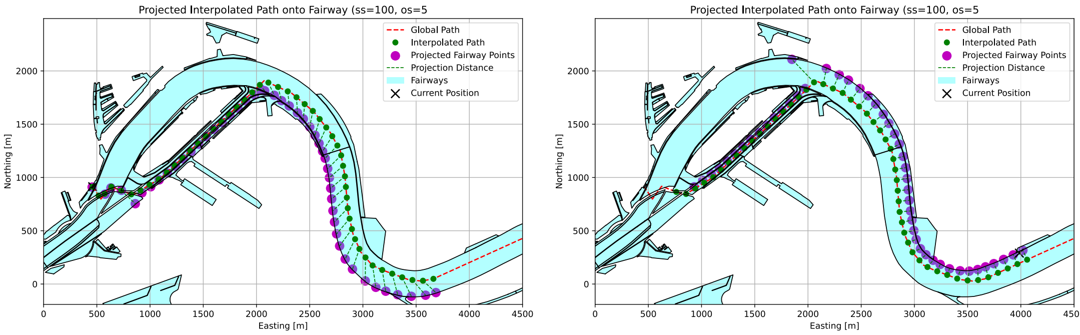


Figure G.4: Rotterdam Projected Paths

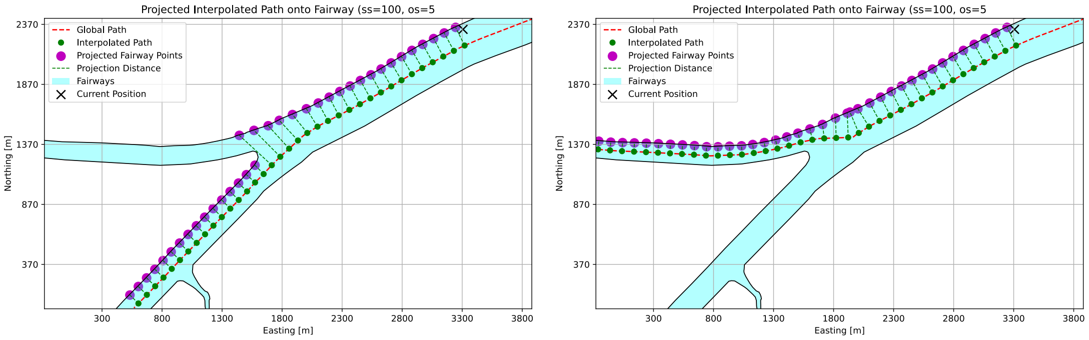


Figure G.5: Sliedrecht Intersection Projected Paths

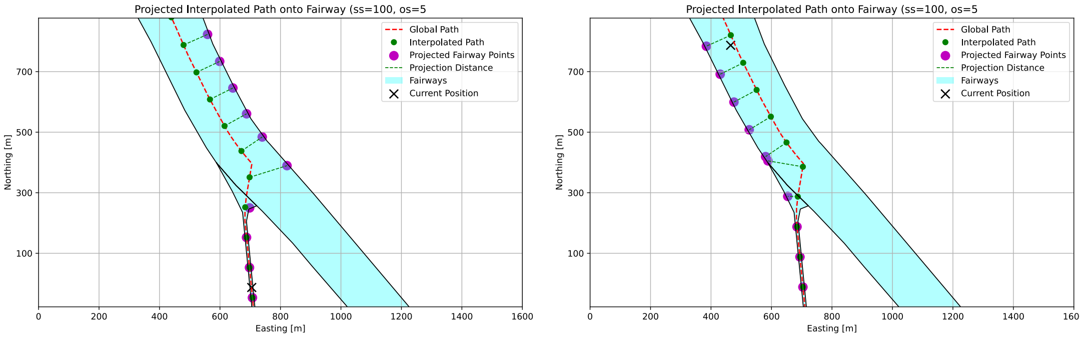


Figure G.6: Sophiapolder scenarios

G.1 Examples of different planning settings

This Section shows the effect of various global path smoothening strategies and its results. First the minimum window approach is shown, which creates smooth segments on both sides of the intersection, however a small discontinuity remains.

When using the average window approach, reprojecting any points that went off the fairway back to the edge, a slightly smoother path is shown but infeasible segments remain. Applying this type of smoothing leads to good results but slight imperfections remain at the intersection.

When first applying the minimum window size smoothening method and then performing an average dynamic window smoothening method, the discontinuity of the minimum window size method becomes less prominent. When applying the average dynamic window smoothening method five times, the seems to disappear and the trajectory leaves the starboard shore more gradually at first to provide a smoother path.

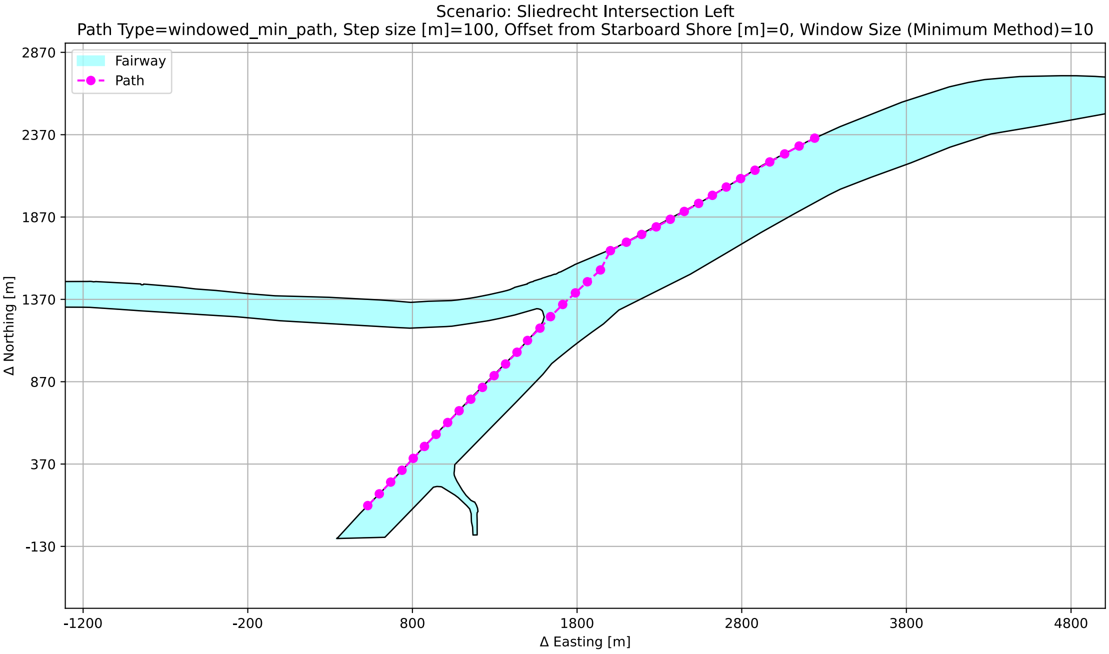


Figure G.7: Sliedrecht Left, minimum window approach.

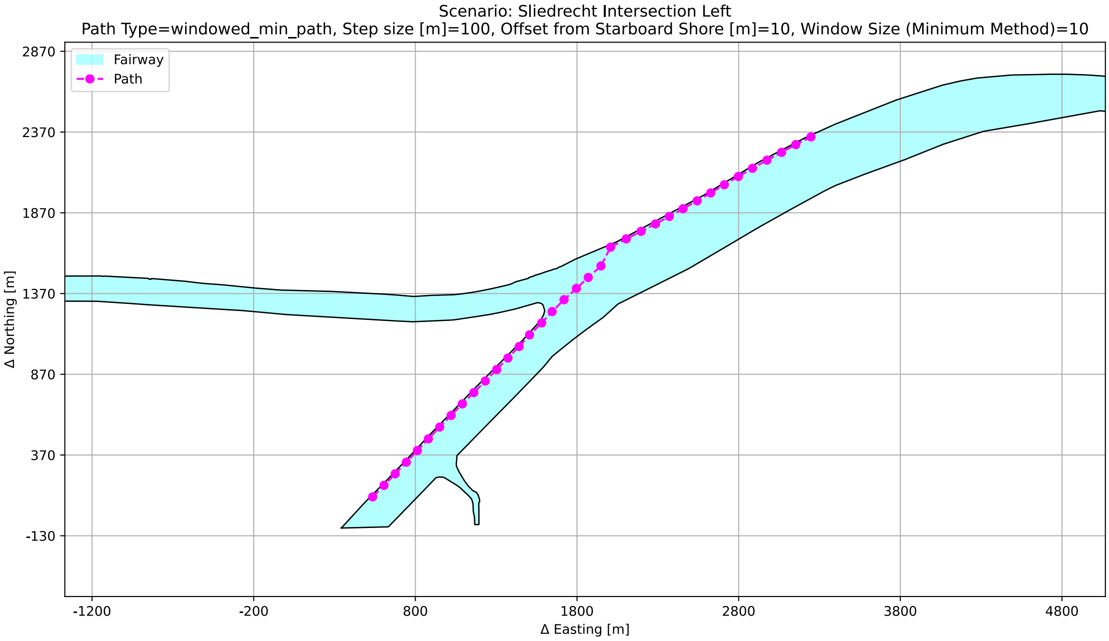


Figure G.8: Sliedrecht Left, minimum window approach with an offset from the starboard shore line of 10 meters.

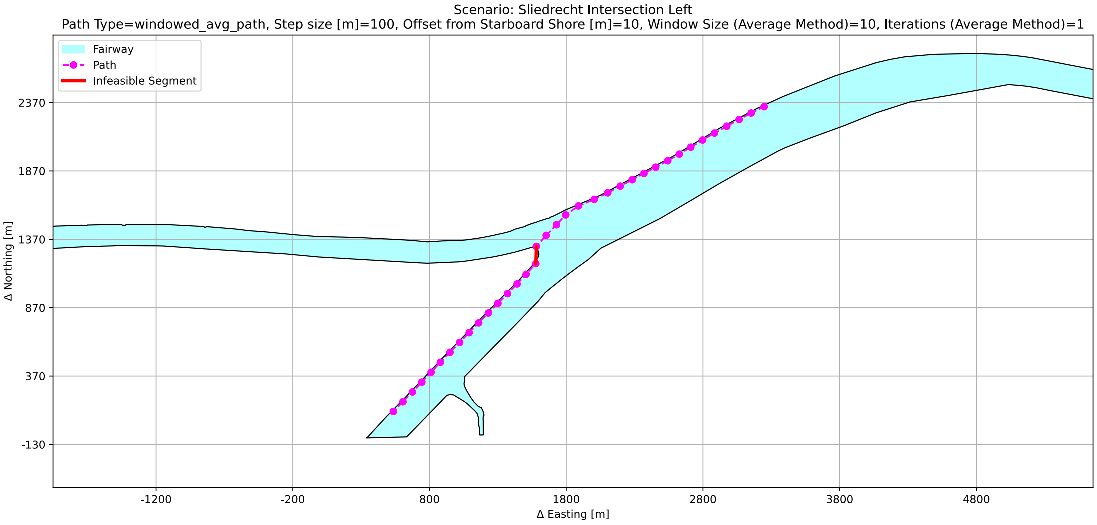


Figure G.9: Sliedrecht Left, average window approach.

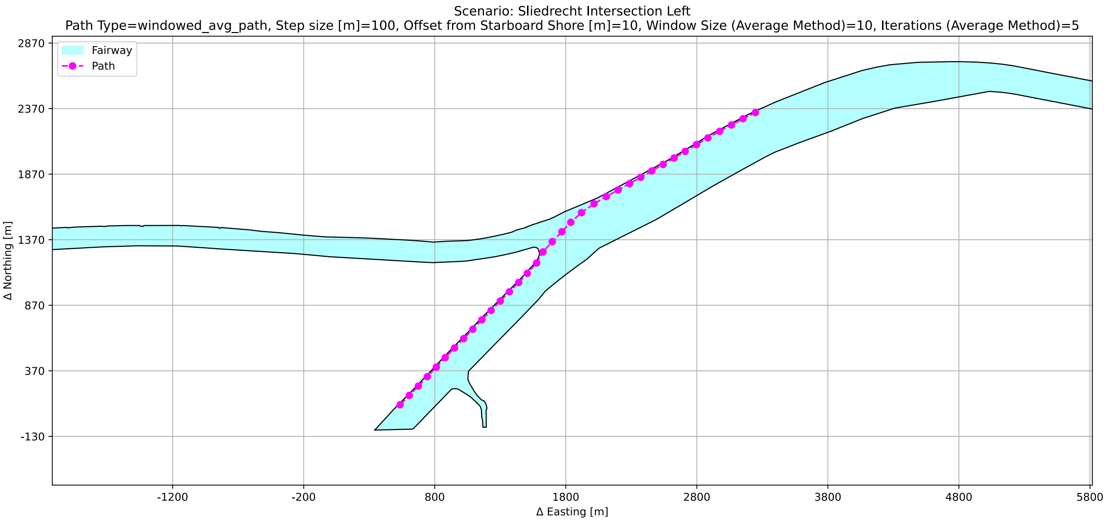


Figure G.10: Sliedrecht Left, average window approach using five iterations.

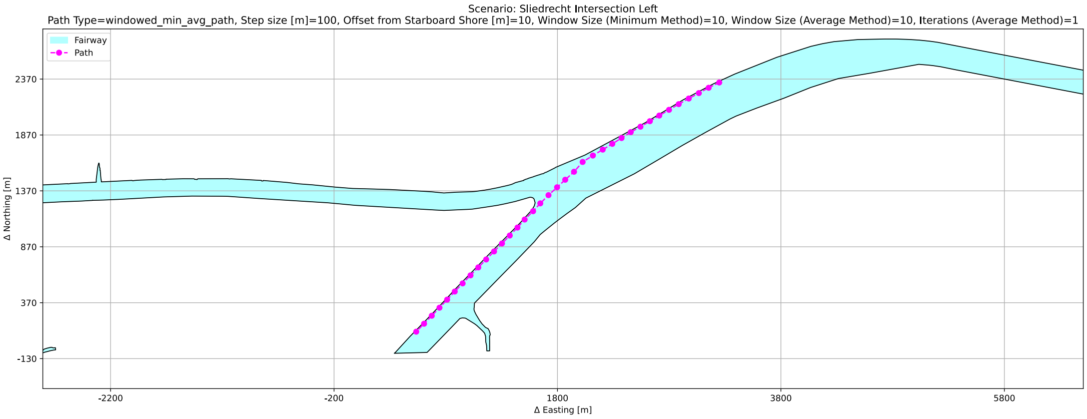


Figure G.11: Sliedrecht Left, Combined approach: First minimum window, then average window.

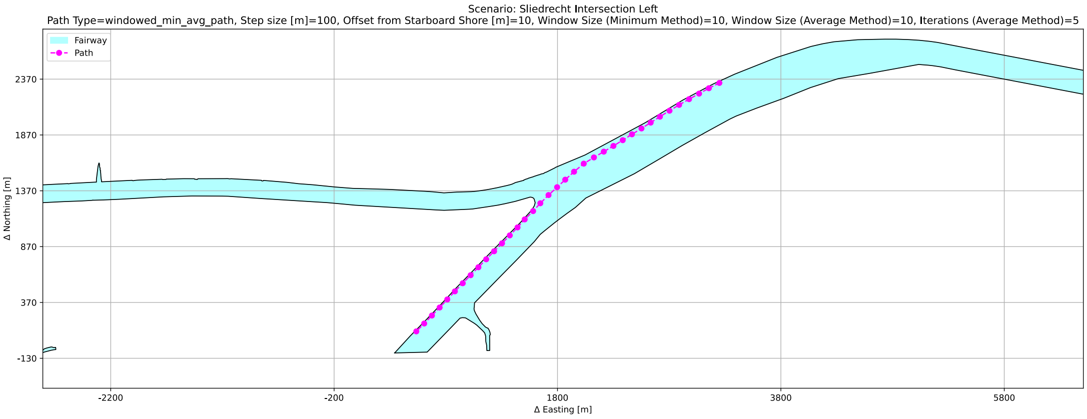


Figure G.12: Sliedrecht Left, Combined approach: First minimum window, then average window with five iterations.