Regulations Aware Motion Planning for Autonomous Surface Vessels in Urban Canals

Jitske de Vries

Delft University of Technology

Image: ©MIT/AMS Institute

TU Delft

Regulations Aware Motion Planning for Autonomous Surface Vessels in Urban Canals

by

Jitske de Vries

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 27 September 2021 at 10:00 AM.

Student number:	4477014	
Project duration:	1 March 2021 – 27 September 2021	
Supervisors:	Dr. J. Alonso-Mora,	TU Delft supervisor
	Ir. E. Trevisan,	TU Delft daily supervisor
	R. Doornbusch,	AMS Institute supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

With a growing number of citizens and tourists, the scarce public space, roads, and public transport in Amsterdam are experiencing rising pressure. Instead of utilizing the conventional transportation routes, Autonomous Surface Vessels (ASVs) could transport goods and people via the 165 canals present in Amsterdam. However, urban canals are challenging for motion planning since the space can be narrow and contain other human-operated boats. Unfortunately, there are limited existing works regarding motion planning for ASVs in urban canals with dynamic obstacles. Additionally, motion planners for other applications can not be applied directly because of the lacking traffic structure in the canals, the specific dynamics of ASVs, and the regulations applying to canals. Therefore, the purpose of this thesis is to develop a motion planning framework that can plan dynamically feasible motions for ASVs in urban canals complying with regulations. These rules are not only mandatory but adhering to them makes the ASV's motion socially compliant, and therefore, more predictable by other canal users. We build upon Local Model Predictive Contouring Control (LMPCC) and extend it with regulation compliance. Adherence to rules is achieved by constructing a new cost function for the optimization problem that allocates a higher cost on specific sides of the obstacle vessel. With this new cost function, the Roboat can behave according to the regulations in takeovers, head-on encounters, and crossings with boats from port and starboard. Furthermore, the effect of predicting the obstacle vessel trajectories with a Social Variational Recurrent Neural Network (VRNN) is compared to the constant velocity model. The entire motion planning framework is compared in simulation with LMPCC and the current motion planning and control framework of the Roboat, Breadth First Search (BFS) combined with Nonlinear Model Predictive tracking Control (NMPC). Additionally, the motion planning framework is implemented on a quarter-scale Roboat and is tested in an outdoor environment with disturbances.

Jitske de Vries
Delft, September 2021

Acknowledgments

In my thesis period, I went on a holiday to Austria to do a long-distance hike. During that hike, I realized that there was some resemblance between the hike and my thesis project. Often my thesis project felt like climbing a mountain without a map and a trail. Don't get me wrong, I really do like hiking but it can be hard too. Luckily, I was surrounded by a wonderful team of people who coached me to the top of the mountain.

First of all, my daily supervisor was Elia Trevisan. He was willing to meet with me every week for the past nine months. I think the meetings we missed can be counted on one hand. That is impressive. Thank you for listening to all my updates, answering all my questions, proofreading every single word I wrote, and all the good discussions we had. Next, I would like to thank my supervisor Javier for setting up the TRiLOGy meeting, and giving your honest and valuable feedback every time I asked for it. Part of the TRiLOGy meeting was Bruno Brito, who was like a bonus supervisor for me. He helped me enormously with writing the storyline with the correct arguments and debugging the code. Speaking of the code, I had a pleasant cooperation with Oscar de Groot, regarding the LMPCC GitHub. Also part of the TRiLOGy meeting group were Jules van der Toorn and Tuhin Das. They are the most enthusiastic bachelor students I have ever met, and together with Bruno they made the Social VRNN predictions part of my thesis possible.

My graduate internship was carried out at the Roboat project of AMS Institute, which provided all the facilities I needed for my thesis. Rens Doornbusch convinced me in the first place to take part in this incredible project. Rens, thank you for supervising me but at the same time having faith in my self-reliance. Joshua Jordan and Jonathan Klein Schiphorst embraced me as a fellow software engineer and took their time to explain and debug code if necessary. And on top of all that, Jonathan was dedicated to make the quarter-scale Roboats work. The experimental testing resulted in a lot of frustrations, but also a lot of fun for the both of us. Last, Ynse Deinema, thank you for being endlessly enthusiastic. All in all, it felt like the Roboat team adopted me as a team member.

In addition, my studies and thesis wouldn't have been much fun without my friends and family. I would like to acknowledge my parents Kees and Ella, who supported me financially and mentally throughout my full studies. I could always turn to them for advice or just some small talk. My boyfriend Martijn had the wonderful ability to reduce my stress levels and forget my thesis once in a while. Also, he taught me a very valuable lesson: "Enough is enough.". And if that wasn't enough, he was willing to listen to every single thing I had to say about my thesis. Another significant person during my thesis was my flatmate Suzanne. We started studying six years ago, stuck together, had some amazing adventures, and now we are finishing our thesis together. Suzanne, you have been a great companion, not only during this thesis but throughout our full studies.

Unfortunately, the world experienced a pandemic one-quarter in my masters. I was lucky to have our "Quarantine Club". Thank you, Didi, Roy, and Martijn, to make life bearable during quarantine. Additionally, my rowing team "Strak" has been a major support throughout this pandemic and my full studies. You made two major life achievements possible this year: my graduation and finishing the 100 km rowing tour "de Ringvaart". Thank you for always being interested in how I am feeling, keeping me fit, and inspiring me to be ambitious, just like you all are. But above all, thank you for the lots of fun we had. Lastly, I would like to acknowledge the so-called dreamteam I participated in, Project MARCH. My dreamteam year taught me many skills, without it, my thesis wouldn't be at the same level. In addition to that, I acquired lifelong friendships, which I appreciate very much.

Jitske de Vries
Delft, September 2021

Contents

1 Introduction

1.1. Background

The municipality of Amsterdam presented the Thermometer of Accessibility report in 2021 [1]. This report stated that the number of citizens increased by 14% from the year 2010 to 2020. Moreover, between 2010 and 2019, the number of jobs increased by 21%, and the hotel stays by 89%. This results in more crowdedness on the roads and in public transport. The canals of Amsterdam, on the other hand, are relatively unused. The 165 canals with a total length of 100 km can be used as an alternative to the conventional routes for transporting goods and people. This opens up the opportunity for developing Autonomous Surface Vessels (ASVs), such as Roboat [2], explicitly designed for this urban environment. The different prototype scale Roboats are visualized in Figure 1.1.

(a) Full-scale (b) Quarter-scale

Figure 1.1: Two prototypes of the Autonomous Surface Vessel (ASV) Roboat developed for the Amsterdam canals.

The Roboat project is a collaboration between the Amsterdam Institute for Advanced Metropolitan Solutions and the Massachusetts Institute of Technology. They aim to develop the world's first fleet of ASVs for the city of Amsterdam. Not only could the Roboat serve as a transportation vehicle for people and goods, but it could also function as a waste collection system, a data collection system, or as a floating bridge or stage (Figure 1.2).

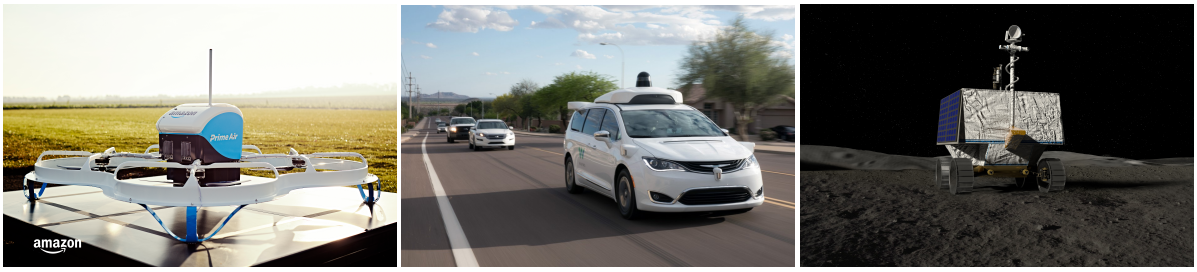
In the last decade, many different types of autonomous vehicles have been developed. Most people are familiar with self-driving cars, drones, or mobile robots (Figure 1.3) as they are being incorporated into our daily lives more and more. A less familiar application is the Autonomous Surface Vessel (ASV), a vessel that can navigate without help of human beings. Currently, most ASVs are being used in a maritime environment; examples are displayed in Fig. 1.4. The primary motivators for automating these vessels are increased efficiency, reduced cost, environmental impact. In addition, improved safety is also a major incentive for automation, since human error causes most maritime accidents [3]. While there are many examples of maritime vessels, there are limited examples of ASVs in cities, which is an entirely different environment.

(a) Trash can be deposited into the floating dumpsters, autonomously transferring the waste.

(b) The Roboat can serve as an on-demand transportation service for people and goods.

(c) The roundAround is a dynamic circular bridge composed of Roboats.

Figure 1.2: There are many possibilities for using ASVs such as the Roboat in a city like Amsterdam. ©MIT/AMS Institute



(a) Prime Air, developed by Amazon, is a future delivery system using autonomous aerial vehicles, also known as drones. [4]

(b) A commercial self-driving taxi service called Waymo. They claim to develop a driver that never gets drunk, tired, or distracted. [5]

(c) Volatiles Investigating Polar Exploration Rover (VIPER), a mobile robot, was developed by NASA to explore the moon's south pole for water ice. [6]

Figure 1.3: In the last decade, many different types of autonomous vehicles have been developed. All of them have a unique way of being incorporated into human society.

1.1.1. Challenges of an Urban Environment for an ASV

The urban environment comes with many challenges. First, the city is a *complex and dynamic environment*. Where in the ocean, obstacles are very sparse; in an urban environment, obstacles are almost always present. These obstacles can be static, for example, quay walls or pillars near bridges. In addition, there are moving obstacles, like the many boats operated by humans that navigate through the canals. In order to ensure safety and avoid any collisions, precise local replanning of the global trajectory is necessary.

Autonomous cars are already able to navigate in dynamic environments. However, autonomous cars can rely on a clear road structure to avoid any collisions with other road users [7–9]. Unfortunately, there are no traffic lanes or traffic lights in the Amsterdam canals. This formulates the second challenge of the urban canals; it is an *unstructured environment*. Where the autonomous car can rely on the traffic lights to avoid collisions, the autonomous surface vessel has to decide on its own when to cross or not to cross.

That is where the *priority regulations* come into play, the third challenge. The Inland Waterways Police Regulations [10] (see Section 1.1.2) describe how to behave in the canals. These laws also specify interaction regulations for encounters with other canal users. Not only is it mandatory to comply with these rules, but it will also enable ASV's motion to be more predictable and socially compliant to the other canal users.

Compared to autonomous cars, mobile robots do navigate in unstructured environments often avoiding pedestrians [11–14]. However, also these methods can not be applied directly to ASVs, because of the fourth challenge. Boats typically have *large inertia and complex dynamics* compared to mobile robots. Therefore, it is not as maneuverable, and slowing down takes more time than for a lightweight robot. The motion planning method should thus anticipate moving obstacles further ahead and cannot use reactive methods that work for mobile robots. On top of that, the planned motions should be dynamically feasible for an ASV.

The last challenge is to navigate in a *crowded environment* with many boats. Crowdedness can result in the freezing robot problem [15] and is often discussed for mobile robots in environments with many moving obstacles [13, 16, 17]. When a motion planning method predicts the future positions of the surrounding vessels without accounting for interactions, a large area could be marked unsafe. As a result, the Roboat would freeze because it is not able to find a secure trajectory. We would still like to find a feasible trajectory, even when the environment is crowded.

(a) The WAM-V 16, an ASV developed by Marine Advanced Robotics. [18]

(b) Surf Bee, a compact low cost marine survey robot. [19]

(c) The Telemetron ASV a high speed vessel can reach velocities up to 18 m/s. [20, 21]

Figure 1.4: A variety of different ASVs, all in a marine environment.

All in all, many challenges are arising from navigating an ASV in urban canals. In addition, current motion planning methods are not sufficient for this specific application. Therefore, the goal for this thesis research is to:

Research, develop and build a regulations aware motion planning framework for ASVs in urban canals.

1.1.2. Inland Waterways Police Regulations

In the city of Amsterdam, the inland waterways police regulations [10] apply. First of all, the boat has to comply with the speed limit in the city, which is 6 km/h. There are some exceptions where it is allowed to navigate with 7.5 km/hour. Additionally, there are priority regulations for avoiding collisions when encountering other boats. These rules apply in a specific order. Note that starboard corresponds to the right side of the boat, while the port side is on the left side.

1. Good seamanship: avoid a collision at all cost, even if you do not need to give way according to the rest of the regulations.
2. Give way to boats longer than 20 m and professional vessels such as canal cruise operators.
3. Navigate on the starboard side of the canal. If the Roboat is not on the starboard side, give way to all boats that do navigate on the starboard side.
4. Give way to sailboats and boats powered by human muscle.
5. When encountering another small motorboat (see Fig. 1.5):
   (a) Overtaking: overtake a boat on the port side.
   (b) Head-on encounter: both boats should move to their starboard side.
   (c) Crossing: give way to a boat from starboard.

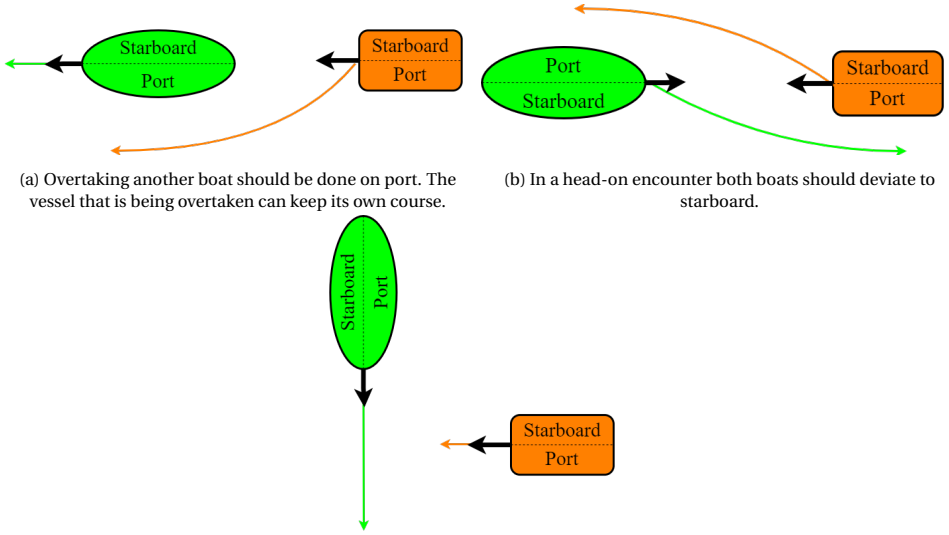
In this thesis, we do not incorporate place-dependent regulations such as Rule 2, navigating on the starboard side of the canal. Instead, we focus on all the other rules, the interaction regulations.

1.2. Related Works

This section discusses a brief overview of the related works regarding four topics: motion planning methods, model predictive control, regulations, and vessel trajectory predictions. A more complete overview can be found in the literature review report written as a foundation for this thesis.

1.2.1. Motion Planning Methods

A motion planning method is responsible for planning a trajectory that satisfies certain demands. Since there are many applications and approaches, it is impossible to comprehensively review all methods. A more extensive overview of motion planning methods for autonomous vehicles can be found in [22–24]. Motion planning methods can be categorized into three different types: hierarchical methods, learning-based methods, and optimization-based methods.



(a) Overtaking another boat should be done on port. The vessel that is being overtaken can keep its own course.

(b) In a head-on encounter both boats should deviate to starboard.

(c) A vessel should give way to boats approaching from starboard.

Figure 1.5: Interaction regulations for equal boat types as described in [10].

Hierarchical Methods

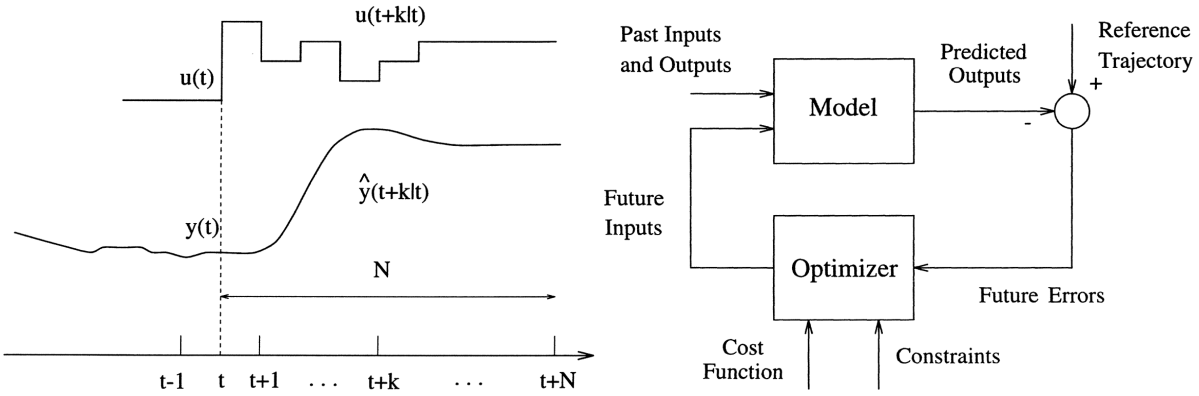
The more classical technique to motion planning is the hierarchical approach. These methods consider planning and path tracking as two separate problems. The path planner will generate a path that will be passed to the tracking controller. The controller makes sure that the path is executed by the vehicle. Planning such path can be done through an A* graph search in combination with motion primitives [25, 26] or an incremental search with Rapdly-exploring Random Tree (RRT) [27]. These hierarchical methods have been used for decades because their approach allows for tractable computation. Because of the Roboat's high inertia, two aspects of the hierarchical methods can result in collisions that could be prevented. First, the dynamics are often not included in the path planning, resulting in dynamically unfeasible paths. Second, the planned motions do not incorporate the future positions of the obstacles, leading to reactive behaviors.

Learning-Based Methods

With the development of increasingly powerful computers, machine learning techniques have gained popularity in all different research fields, also in motion planning. For example, end-to-end planning [28], in which the commands sent towards the actuators are learned from raw sensor data. Another option is Inverse Reinforcement Learning (IRL), in which the weights of a reward function with hand-designed features are learned. IRL is successfully applied to, for example, mobile robots surrounded by pedestrians [13] and autonomous vehicles on public roads [29]. Overall, learning from expert behavior results in natural paths. However, to incorporate the vehicle dynamics into the planning of the path, a lot of Roboat specific data is required. If this specific data is not available, Reinforcement Learning (RL) could be used [30, 31]. RL is capable of learning trajectories that give the most reward evaluated on a hand-constructed reward function. Training of this method can be done in simulation and the real world. Training in the real world can cause risky situations and is very costly. At the same time, the simulation environment might not be representative enough for the real world. Moreover, all learning-based methods might not perform so well on rare edge cases that are not included in the training set.

Optimization-Based Methods

Due to the recent developments in solvers for nonlinear constrained optimization, optimization can be used for real-time path planning. Incorporating both planning and tracking into one problem results in a simple and elegant problem formulation [32]. It is also possible to integrate the dynamic model into the trajectory planning, resulting in more smooth trajectories. However, if the problem is not convex, optimization-based motion planning may result in locally optimal trajectories. Additionally, finding a reasonable solution within real time can also become difficult when the optimization problem becomes too complex. But through simplification of the problem formulation, many methods can find a reasonable solution within a short time [33, 34]. Moreover, an optimization-based method can find trajectories that are dynamically feasible, and the



(a) The receding horizon principle with length N. The control input u and the predicted resulting output y will be computed for the full prediction horizon at each time step. Only the current control input at time t will be sent to the process.

(b) The basic structure of MPC. The future inputs are determined by solving an optimization problem that minimizes the cost. The cost function penalizes errors but also other elements can be minimized. Additionally, the optimization problem is subject to constraints.

Figure 1.6: The receding horizon principle and the basic structure of the Model Predictive Control technique [36].

method will be able to anticipate moving obstacles [11, 35]. These locally optimal trajectories are obtained using a hand-constructed cost function and do not require any data set. Therefore, it is able to work with new scenarios as well.

1.2.2. Model Predictive Control

An example of an optimization-based method that is able to run in real-time, depending on the problem formulation, is Model Predictive Control (MPC) [36]. The method is visualized in Figure 1.6b. The technique of MPC consists of four core elements: the cost function, the constraints, the dynamic model, and the receding horizon.

The cost function or the objective function J consists of every aspect of the problem that should be minimized. Common aspects are the reference error, control input, and collisions. Nonetheless, it also allows for including other important aspects for the designer, in this case, compliance with regulations. The second core element is the constraints c, which help limit the system's inputs u and outputs y. Mainly this element will ensure safety by constraining the position to safe areas and allowing only velocities lower than the maximum velocity. Additionally, a dynamical model can predict the output of the system to a specific control input. This way, a control signal can be determined specific to the dynamics using feed-forward control. The last element is the receding horizon, as illustrated in Figure 1.6a [36]. At each time step t the optimal control inputs and resulting outputs will be calculated for the full length N of the prediction horizon. However, the control signals calculated from t onward will not be used, considering that new outputs are available for the next time instant. Because of the recalculation, MPC can react to disturbances and changes in the environment.

MPC for Autonomous Vehicles

The downside of MPC is that optimization of a complex problem requires time. Therefore, many methods have altered the problem in a specific way to ensure real-time calculation. In order to find the best technique for our application, different types of Autonomous Vehicles (AVs) have been researched.

First, *vessels* are mainly situated in oceanic environments in which less precise navigation is allowed. Therefore, it is possible to use predefined maneuvers and evaluate each possible maneuver's cost [20, 37] instead of solving an optimal control problem. While for our application, more than a small subset of all feasible movements should be considered. Another specific AV is the *racing car*, which operates near their limit of friction. This requires dynamically feasible paths, and thus they often combine planning and path tracking in one MPC problem. The sampling rate is often higher (50 Hz) than with our application (5-10 Hz). To enable fast solving, (a combination of) linear models are utilized to simplify the dynamics [33, 34]. Another approach is a sampling-based technique [32], which will not perform well in a dynamic environment. Where racing cars show us how the dynamics can be simplified, the *self-driving cars* give us examples for simplified environments. These variants of MPC make use of the traffic lanes they have to drive in. Instead of considering the full environment, they consider so-called tubes [7, 8], a convex region without any obstacles.

Another approach is a mixed-integer problem to select the correct lane [9]. Such clear structures, however, are not available in urban waterways. Moreover, for *Micro Air Vehicles* (MAVs), the method Chance Constrained Nonlinear Model Predictive Control (CCNMPC) is often chosen [35, 38]. Instead of using hard constraints, the constraint is defined such that the probability of a collision is below a certain threshold. This approach results in less conservative paths but it also makes the problem more complex. Both properties are not desired for an ASV. *Mobile robots* are the last vehicles that we consider. These vehicles often have to deal with similar unstructured dynamic environments [11, 12]. However, mobile robots usually have much simpler dynamics. Still, Local Model Predictive Contouring Control (LMPCC) [11] is especially interesting for our application since it predicts the motions of the moving obstacles and is, therefore, able to anticipate them.

In conclusion, LMPCC can tackle many of the challenges discussed in Section 1.1.1 and is therefore selected as a foundation for our motion planning method. For instance, the contouring control method is able to anticipate and avoid dynamic obstacles in an unstructured environment. Additionally, the ASV's dynamics can be incorporated into the motion planning, resulting in dynamically feasible paths. Lastly, the priority regulations can be incorporated into the cost function of the MPC optimization problem.

1.2.3. Regulations

The rules that apply to the Amsterdam canals are described in the inland waterways police regulations [10]. The desired behavior for interactions with other vessels is also described in COLlision avoidance REGulations on Sea (COLREGS) [39]. Several methods implement COLREGS compliance for ASVs on the sea.

Starting with a hierarchical approach, COLREGS-RRT [40]. Utilizing virtual obstacles on specific sides of the vessels, the planned path is able to comply with the regulations. However, this is a binary approach that will either result in full compliance with the rules or no compliance at all. This approach might not work in a more crowded environment like the Amsterdam canals because full compliance is not always possible. Two other methods incorporate COLREGS with MPC. Both techniques generate a finite set of trajectories for which the costs will be evaluated separately. In [21, 37], a hazard function is constructed, containing a binary indicator that expresses whether the maneuver handles encounters with other boats in compliance with the regulations. The other method, Branching Course Model Predictive Control [20], defines three discrete penalty regions around the obstacle vessels to allocate a higher cost in front and on the starboard side of the obstacle boat. Because both methods are used on the sea, the optimization can be executed by brute force evaluation of the limited number of trajectories. However, this is not possible for a canal environment, where the navigation should be very precise, because this will result in too many computations to evaluate all the precise movements. To conclude, no COLREGS methods were found that ensure precise navigation, which is necessary on the Amsterdam canals.

A completely different approach to generating interactions according to the rules is through reinforcement learning. For Socially Aware Motion Planning with Deep Reinforcement Learning [31], a norm-inducing reward function is defined. The learning agent is penalized when another agent is within specific regions. This way, left-handed and right-handed interaction regulations are learned for a mobile robot in simulation and the real world. For the Roboat, reinforcement learning in Amsterdam's canals is too dangerous. In addition, learning in simulation requires accurate modeling of the dynamics and the interactions of the vessels.

1.2.4. Vessel Trajectory Predictions

For trajectory planning with receding horizon control, it is necessary to know the other vessel's motion plans in advance. Traditional approaches [11, 35] make use of simple prediction models like the constant velocity model. However, these predicted vessel motions might be oversimplified. Moreover, they do not account for interactions with other agents. As a result, too much of the space can be marked unsafe in a crowded environment. Then the freezing robot problem can occur [15]. This problem can be avoided by accounting for interactions.

For example, there are cooperative methods, utilizing, for instance, multiple goal interacting Gaussian processes [16] or Inverse Reinforcement Learning [13]. These methods plan paths for all nearby agents. As a result, the trajectories reach their goal faster than when using non-cooperative methods. However, unavailable information about hidden states such as the intended goal is needed. Another approach is taken with Collision Avoidance with Deep Reinforcement Learning (CADRL) [31], which learns movements while accounting for interactions. However, RL demands accurate simulation of the environment.

A different option is to evaluate the motion planning problem and the vessel trajectory predictions separately. In [41], it is shown that with only knowledge of the current state and its environment, trajectories of pedestrians can be accurately predicted in real-time by a Social Variational Recurrent Neural Network (VRNN)

learned on real-world data. In theory, accurate vessel motions and interactions can be learned because the static environment and the other agents are also considered for the predicted motions. For the entire motion planning problem, learning was not feasible because of a lack of data. On the other hand, for vessel trajectory predictions an Automatic Identification System (AIS) data set is available.

1.3. Objectives and Contributions

The objective of this thesis is to develop a regulations aware motion planning framework for an ASV in urban canals. The method should generate dynamically feasible motion plans in real-time taking into account the complex dynamics of an ASV. Additionally, static and moving obstacles should be anticipated and therefore avoided by the planner. The main contribution of this thesis is a cost function that encourages adherence to interaction regulations in four different scenarios: take-over, head-on, and crossing with a vessel from starboard or port.

The developed framework will be compared with LMPCC [11] and the current motion planning and control method for the Roboat, Breadth First Search (BFS) in combination with Nonlinear Model Predictive Control (NMPC) [42] to show its effectiveness. Also, the influence of predicting the future positions by Social Variational Recurrent Neural Network (VRNN) compared to the constant velocity model is researched. Moreover, the framework is demonstrated in an outdoor environment with disturbances.

1.4. Thesis Outline

The main body of this thesis is the scientific paper written for the IEEE International Conference on Robotics and Automation 2022 (Chapter 2). The rest of the chapters contain additional information to fully understand the method, the experiments, and the limitations. First, Chapter 3 contains more information about the methods of the developed motion planning method, Regulations Aware Model Predictive Contouring Control (RA-MPCC) and the vessel trajectory predictions. Then, in Chapter 4, the experiments and their results will be discussed in more detail. Potential limitations are discussed in Chapter 5. Eventually, the thesis will be concluded in Chapter 6, which also discusses the future work.

2

Scientific Paper

This chapter contains the scientific paper that is submitted to IEEE International Conference on Robotics and Automation 2022 on 14 September 2021. Additionally, a video showing the experimental results will be submitted on 21 September 2021.

Regulations Aware Motion Planning for Autonomous Surface Vessels in Urban Canals

Jitske de Vries, Elia Trevisan, Jules van der Toorn, Tuhin Das, Bruno Brito, and Javier Alonso-Mora

***Abstract*— In unstructured urban canals, regulation-aware interactions with other vessels are essential for collision avoidance and social compliance. In this paper, we propose a regulations aware motion planning framework for Autonomous Surface Vessels (ASVs) that accounts for dynamic and static obstacles. Our method builds upon local model predictive contouring control (LMPCC) to generate motion plans satisfying kino-dynamic and collision constraints in real-time when a feasible solution is found, while including regulation awareness. To incorporate regulations in the planning stage, we propose a cost function encouraging compliance with rules describing interactions with other vessels similar to COLlision avoidance REGulations at Sea (COLREGS). These regulations are essential to make an ASV behave in a predictable and socially compliant manner with regard to other vessels. We compare the framework against baseline methods and show more effective regulation-compliance avoidance of moving obstacles with our motion planner. Additionally, we present experimental results in an outdoor environment.**

I. INTRODUCTION

With a growing number of citizens and tourists, the scarce public space, roads, and public transport in Amsterdam is experiencing rising pressure [1]. A possible solution to this problem is to use the 165 canals with a total length of 100 km as an alternative to the conventional routes to transport goods and people. This opens up the opportunity for developing Autonomous Surface Vessels (ASVs) explicitly designed for urban environments, such as Roboat [2].

However, urban canals are challenging for motion planning since the space can be narrow and contain other human-operated boats. Also, ASVs have slow dynamics and cannot brake quickly. Therefore, navigation requires precision and planning ahead to avoid any collision with both static and dynamic obstacles. Moreover, interaction regulations [3] apply to Amsterdam's canals (Fig. 2), similar to those described in the COLlision avoidance REGulations at Sea (COLREGS). These regulations are not only mandatory, but adhering to them makes the ASV's motion socially compliant, and therefore, more predictable by other canal users.

While there are many examples of autonomous cars [4] and mobile robots [5] navigating in dynamic urban environments containing human agents, the examples for ASVs are limited as they are primarily designed for marine or coastal areas [6]–[9]. Furthermore, motion planning algorithms for autonomous vehicles mostly rely on the road structure [4], [10], [11]. On urban waterways there are no traffic lanes or traffic lights, leading to more interactions with other vessels. In order to interact appropriately with obstacle vessels, the motion planning method should be aware of the interaction regulations. Mobile robots often have to deal with similar unstructured dynamic environments [5], [12]–[14]. However, compared to a mobile robot, a vessel has large inertia and complex dynamics.

Fig. 1: A visualization of two Roboats navigating Amsterdam's canals. The blue arrows represent their planned motion. ©MIT/AMS Institute.

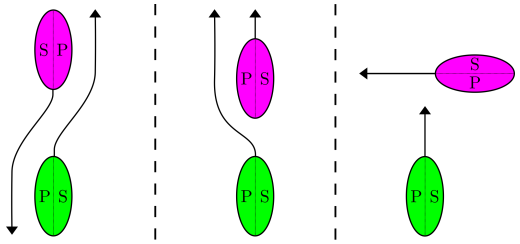


Fig. 2: Two boats interacting in a (from left to right) head-on, overtaking, and crossing scenario according to the regulations. The starboard and port side of the boats are denoted as respectively S and P.

In this paper, we propose a motion planning framework for ASVs in urban canals. Our method employs model predictive contouring control (MPCC) to generate regulation-aware dynamically feasible motions in real-time. The complete system overview is displayed in Fig. 3. Our main contribution is a method for collision-free and regulations-aware motion planning.

This work was supported by the TRiLOGy project and the Amsterdam Institute for Advanced Metropolitan Solutions (AMS) in the Netherlands.

The authors are with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD, Delft, The Netherlands. {e.trevisan; bruno.debrito; j.alonsomora}@tudelft.nl

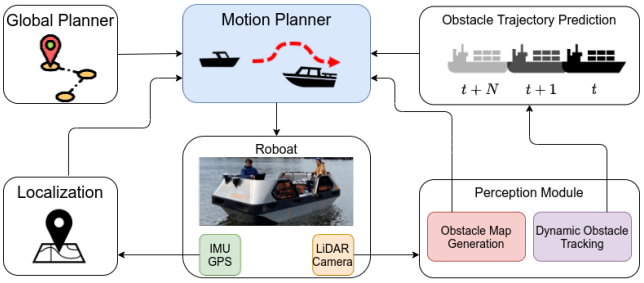


Fig. 3: The proposed framework for priority regulations aware motion planning for ASVs in urban canals. The motion planner receives a global path, the current Roboat position, a static obstacle map, and the position and predicted trajectories of the dynamic obstacles. With this information, it generates inputs for Roboat's thrusters.

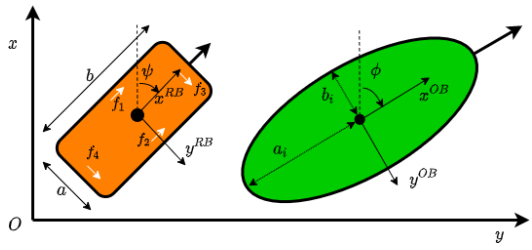


Fig. 4: The Roboat and an obstacle vessel. Angles ψ and ϕ denote the orientation of the boats. a and b are the width and the length of the Roboat. a_i and b_i are the semi-major and minor axis respectively of obstacle boat i. The Roboat's thrusters can exert force f in two directions, the white arrows denote the positive direction.

A. Related Work

Traditional motion planning methods employ a hierarchic planning architecture decomposing the navigation pipeline into a sequence of blocks performing different sub-tasks such as motion planning and control [15]. For instance, [16] employed A* to search a state lattice and motion primitives for control and [17] employed A* for path planning and Nonlinear Model Predictive Control (NMPC) as a tracking controller. However, the first is a reactive method which can be troublesome for collision avoidance in high inertia systems. The second plans along a prediction horizon, but it does not account for static or dynamic obstacles.

Receding-horizon approaches such as Model Predictive Control (MPC) [4], [5], [18], [19] can be directly deployed in real environments by dealing with the model inaccuracies and environments changes by continuous re-planning online. However, when navigating in urban canals, ASVs must comply with the inland waterways police regulations [3] which these methods neglect.

COLlision avoidance REGulations on Sea (COLREGS) describe the same rules for interactions with other vessels. Several methods implement COLREGS compliance for ASVs in oceanic and coastal environments [20]–[24]. Nevertheless, these approaches are not feasible for a crowded and complex environment like Amsterdam's urban waterways. [20] relies on virtual obstacles for COLGRES compliance which makes the problem unfeasible in crowded canals. [21] and [22] use a small set of motions primitives that would not be rich enough to navigate dense environments. [23] and [24] employ geometrical rules resulting in highly reactive motions. Hence, in this paper, we propose a regulations-aware motion planning framework employing receding-horizon trajectory optimization for static and dynamic collision avoidance.

B. Contribution

The main contribution is Regulations Aware Model Predictive Contouring Control (RA-MPCC), a motion planning framework for an ASV in urban canals, which includes a cost function that encourages adherence to the interaction regulations in four different scenarios: take-over, head-on encounter, and crossing with a vessel from starboard or port.

The system is compared in simulation with LMPCC [5] and the current motion planning and control method for the Roboat, Breadth First Search (BFS) in combination with NMPC [17] (Section IV-B). Moreover, the framework is demonstrated in an outdoor environment with disturbances (Section IV-C).

II. PRELIMINARIES

Vectors are denoted with bold lowercase symbols, matrices with bold uppercase symbols, and sets with scripted symbols. Superscript W denotes coordinates in World frame, while B indicates the body-fixed frame.

A. Robot Description and Dynamics

Let B represent an ASV on the plane $\mathscr{W} = \mathbb{R}^2$. The vessel is visualized in Fig. 4. $\mathbf{p}^W$ denotes the position of the Roboat, and $\mathbf{R}_B^W$ is the rotation matrix corresponding to its orientation. The area occupied by the Roboat is represented with a union of $n_d = 3$ discs. Each disc j is centered at a position $\mathbf{p}_j^W = \mathbf{p}^W + \mathbf{R}_B^W(\mathbf{z})\mathbf{p}_j^B$, in inertial frame. Moreover, *port* will describe the left side of a boat looking forward, and *starboard* will be used for the right. Additionally, the ASV's dynamics are defined by a discrete-time nonlinear differential equation, described in detail in [17].

$$\begin{aligned} \eta(t+1) &= \mathbf{R}(\psi)\mathbf{v}(t) \\ \mathbf{v}(t+1) &= \mathbf{M}^{-1}\mathbf{B}\mathbf{u}(t) - \mathbf{M}^{-1}(\mathbf{C}(\mathbf{v}(t)) + \mathbf{D}(\mathbf{v}(t)))\mathbf{v}(t) \end{aligned} \tag{1}$$

Where the state vector of the vessel is $\mathbf{z}(t) = [x\ y\ \psi\ u\ v\ r]^T$. $\eta(t) = [x\ y\ \psi]^T$ represents the configuration given by the position and the orientation of the robot in the inertial frame, and $\mathbf{v}(t) = [u\ v\ r]^T$ is respectively the surge velocity, sway velocity, and yaw rate of the vehicle in the body-fixed frame. Converting a state from body frame to inertial frame can be done by the transformation matrix $\mathbf{R}(\psi)$. The inputs are given by the four thrusters, the applied forces are described in the control vector $\mathbf{u} = [f_1\ f_2\ f_3\ f_4]^T$. The control matrix $\mathbf{B}$ describes the thruster configuration. $\mathbf{M} = \text{diag}\{m_{11}, m_{22}, m_{33}\}$ is the positive-definitive symmetric added mass and inertia

matrix. $\mathbf{C}(\mathbf{v}) \in \mathbb{R}^{3\times3}$ is the skew-symmetric vehicle matrix of Coriolis and centripetal terms. It is assumed that the origin O_b corresponds to the center of mass of the Roboat. $\mathbf{D}(\mathbf{v}) \in \mathbb{R}^{3\times3}$ is the positive-semi-definite drag matrix-valued function with linear damping terms on its diagonal. In short, the dynamics can be summarized as:

$$\mathbf{z}(t+1) = f(\mathbf{z}(t), \mathbf{u}(t)) \tag{2}$$

B. Static Obstacles

$\mathscr{O}^{\text{static}} \subset \mathscr{W}$ is the area that is occupied by static obstacles. This area is represented in an occupancy grid map. This map can either be created beforehand based on map segments of Amsterdam or can be generated in real time from sensor readings.

C. Dynamic Obstacles

The area occupied by dynamic obstacles, such as boats, is described by $\mathscr{O}^{\text{dyn}} \subset \mathscr{W}$. These dynamic obstacles are represented by ellipsoids with a semi-major axis a and semi-minor axis b. For each dynamic obstacle i the current position, rotation matrix $R_i(\phi)$ and velocity v_i are assumed to be known. Future positions of the dynamic obstacles are obtained using a constant velocity model.

D. Global Reference Path

A global reference path $\mathscr{P}$ is given to our local planner. A global planner could generate this path. The reference is built up from M way-points $p^r_m = [x^p_m, y^p_m] \in \mathscr{W}$ with $m \in \mathscr{M} := \{1, ..., M\}$. As described in [5], cubic polynomials describe the path segments for smoothness. A variable θ represents the traveled distance along the reference path.

III. MOTION PLANNING

This section presents the Regulation Aware Model Predictive Contouring Control (RA-MPCC) method, based on [5]. This method is used for planning collision-free, dynamically feasible, and regulation-aware motion plans.

At every time step t, a receding horizon constrained optimization problem with an N length prediction horizon T_{horizon} is solved.

$$\begin{aligned} J^* = \min_{\mathbf{z}_{0:N}, \mathbf{u}_{0:N-1}, \theta_{0:N}} & \sum_{k=0}^{N-1} J(\mathbf{z}_k, \mathbf{u}_k, \theta_k) + J_N(\mathbf{z}_N, \theta_N) \\ \text{s.t.} \quad & \mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{u}_k),\ \theta_{k+1} = \theta_k + v_k\tau, \\ & \mathscr{B}(\mathbf{z}_k) \cap \left(\mathscr{O}^{\text{static}} \cup \mathscr{O}^{\text{dyn}}_k\right) = \emptyset, \\ & \mathbf{u}_k \in \mathscr{U},\ \mathbf{z}_k \in \mathscr{Z},\ \mathbf{z}_0, \theta_0 \text{ given}. \end{aligned} \tag{3}$$

J is the cost function with $\mathscr{U}$ and $\mathscr{Z}$ the sets of admissible inputs and states and $\mathbf{z}_{0:N}$, $\mathbf{u}_{0:N-1}$ the sequence of state and control inputs, respectively, for the prediction horizon. The predicted progress along the reference path is θ_k with k being the time-step. v_k denotes the forward velocity of the Roboat. The output of the optimization is an optimal control input sequence $[\mathbf{u}^*_t]^{t=N-1}_{t=0}$.

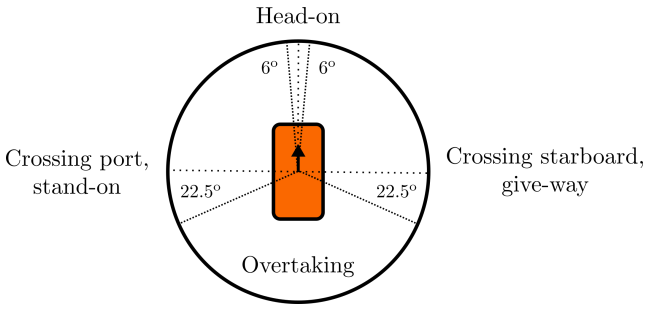


Fig. 5: The position of an obstacle boat with respect to the Roboat can be categorized in one of the four regions: head-on, crossing port, crossing starboard or overtaking.

A. Cost Function

The cost function consists of four elements: the tracking, the speed, the input, and the regulations costs (Section III-B).

$$J_{\text{tracking}}(\mathbf{z}_k, \theta_k) = \mathbf{e}_k^T \mathbf{Q}_\varepsilon \mathbf{e}_k \tag{4a}$$
$$J_{\text{speed}}(\mathbf{z}_k, \mathbf{u}_k) = Q_v(u_{\text{ref}} - u_k)^2 \tag{4b}$$
$$J_{\text{input}}(\mathbf{z}_k, \theta_k) = \mathbf{u}_k^T \mathbf{Q}_u \mathbf{u}_k \tag{4c}$$

The tracking cost penalizes error vector e_k containing the estimated contour $\tilde{\varepsilon}^c$ and lag $\tilde{\varepsilon}^l$ error. Second, J_{speed} contains the deviation of the surge velocity u_k from the reference velocity u_{ref}. Furthermore, the inputs are penalized with J_{input}. $\mathbf{Q}_\varepsilon$, Q_v, and $\mathbf{Q}_u$ denote design weights. The tracking, velocity, and input costs are further described in [5]. The stage cost and the terminal cost are, respectively, $J(\mathbf{z}_k, \mathbf{u}_k, \theta_k) := J_{\text{tracking}} + J_{\text{speed}} + J_{\text{input}} + J_{\text{reg}}$ and $J_N(\mathbf{z}_N, \theta_N) := J_{\text{tracking}} + J_{\text{reg}}$, where J_{reg} is the regulation cost described in the next section.

B. Priority Regulation Compliance

In the city of Amsterdam, the inland waterways police regulations [3] apply. Humans operating the other vessels will assume that the Roboat will behave according to these regulations. Thus, complying with these rules will result in social compliance, and therefore, in fewer collisions. In this work, we consider the specific regulations that describe interactions with other vessels. The first interaction regulation describes specific boats to which a small motorboat like the Roboat should always give way. So-called priority boats are sailboats, boats powered by muscle, professional vessels such as canal cruises, and vessels longer than 20 meters. Still, most of the vessels on the canals are of the same type as the Roboat. When encountering another small motorboat, there are three different situations (Fig. 2). First, when having a head-on encounter with another boat, both boats should move to their starboard side. Second, overtaking another boat has to be done on the port side. Furthermore, in a crossing, one should give way to a boat from starboard.

While in [25] and [22], discrete parameterized cost functions of different shapes were defined with respect to obstacles, a continuous cost function with a simple shape is constructed for our method to facilitate the optimization. We

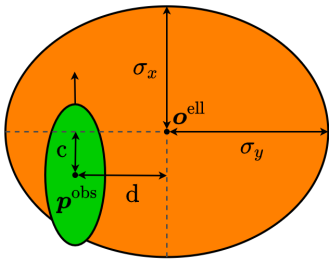


Fig. 6: Geometry of the regulations cost function (7). A higher cost is allocated to the obstacle's starboard and front by shifting the ellipse's center from the center of the obstacle p^{obs} by some parameters c and d in the x- and y-direction of the obstacle, respectively. The standard deviation of the 2D Gaussian for x and y axes are respectively σ_x and σ_y.

have selected an off-center ellipsoidal 2D Gaussian function (7) as displayed in Fig. 6. The idea is to use this smooth function to penalize specific positions with respect to the obstacle boats. We define two types of costs: J_{HT} for Head-on and Take-over encounters and J_{RoW} for Right of Way (RoW). Both use the same cost function but with different parameters. These two costs together make the regulation costs $J_{reg} = J_{HT} + J_{RoW}$.

1) Head-On and Take-Over: First, the cost function J_{HT} is used to allocate a higher cost to both the starboard side and the front of the obstacle boat. This asymmetric cost will help to achieve the desired trajectories for head-on encounters and takeovers, displayed in Fig. 2. The ellipsoid is shifted from the $\mathbf{p}^{\text{obs}}$ with c in the x-direction of the vessel and with d in the negative y-direction. This results in the origin of the ellipse $\mathbf{o}^{\text{ell},\,W}$ in world frame W.

$$\mathbf{o}_{k,i}^{\text{ell},\,W} = R(\phi_{k,i}) \begin{bmatrix} c \\ d \end{bmatrix} + \mathbf{p}_{k,i}^{\text{obs}} = \begin{bmatrix} x_{k,i}^{\text{ell}} \\ y_{k,i}^{\text{ell}} \end{bmatrix} \tag{5}$$

The standard deviation of the 2D Gaussian ellipsoid in the x- and y-direction are σ_x and σ_y, respectively. These values are dependent on the size of the obstacle (major-axis a_i and minor-axis b_i) and the disc representing the Roboat, and are scaled with parameters g and h.

$$\begin{aligned} \sigma_{x,i} &= g(a_i + r_{disc}) \\ \sigma_{y,i} &= h(b_i + r_{disc}) \end{aligned} \tag{6}$$

The cost function J_{HT} can be constructed using scalars λ, μ and ν to shape and rotate the ellipsoid with the obstacle's size and orientation ϕ_i.

$$\begin{aligned} J_{HT}(\mathbf{z}_k) = Q_{HT} \sum_{i=1}^{n} \exp(-(&\lambda (x_k^{\text{Roboat}} - x_{k,i}^{\text{ell}})^2 \\ &+ 2\mu (x_k^{\text{Roboat}} - x_{k,i}^{\text{ell}})(y_k^{\text{Roboat}} - y_{k,i}^{\text{ell}}) \\ &+ \nu (y_k^{\text{Roboat}} - y_{k,i}^{\text{ell}})^2)) \end{aligned} \tag{7}$$

$$\begin{aligned} \lambda &= \frac{cos(\phi_i)^2}{2\sigma_x^2} + \frac{sin(\phi_i)^2}{2\sigma_y^2} \\ \mu &= \frac{sin(2\phi_i)}{4\sigma_x^2} - \frac{sin(2\phi_i)}{4\sigma_y^2}; \\ \nu &= \frac{sin(\phi_i)^2}{2\sigma_x^2} + \frac{cos(\phi_i)^2}{2\sigma_y^2}; \end{aligned} \tag{8}$$

2) Right of Way: The RoW costs will only be allocated to priority vessels depending on their type and length. Additionally, a vessel will also be marked as a priority boat if the Roboat sees the obstacle vessel in the crossing starboard giving way area and the Roboat is seen by the obstacle vessel in the crossing port stand-on area (Fig. 5). The cost function will be allocated to the boat for the full prediction horizon. The cost function is similar to the function for J_{HT}, but with different parameters. In this case, a long ellipsoidal cost J_{RoW} weighted with Q_{RoW} will be placed in front of the priority boat to discourage crossing paths. For this scenario, the ellipsoid's center will only be shifted in the x-direction of the vessel by parameter f. Moreover, σ_x is equal to parameter e, and σ_y is $b_i + r_{disc}$.

C. Static Collision Avoidance

The static obstacles are represented with an occupancy grid map, which is then divided into convex shapes. After that, the points of these convex shapes that are the closest to the Roboat are selected. At last, the linear constraints are determined such that they are normal to the vector pointing from the Roboat to the closest points. These constraints are defined by a vector $\mathbf{A}$ and a scalar b. Resulting in the following equation, in which $\mathbf{p}_j^W$ denotes the position of disc j representing Roboat and δ is a safety margin.

$$c^{\text{stat},j}(\mathbf{z}_0) = A * \mathbf{p}_j^W - b + r_{\text{disc}} + \delta \leq 0 \tag{9}$$

D. Dynamic Collision Avoidance

For dynamic collision avoidance, it is assumed that the moving obstacles can be represented with an ellipse, having semi-axes a_i and b_i. The position and rotation of each obstacle $i \in \{1,...,n\}$ are $\mathbf{p}_i(t)$ and $\mathbf{R}_i(\phi)$. $\Delta x_{k,i}^j$ and $\Delta y_{k,i}^j$ denote the distance in x and y-direction between the center of the disc j and the obstacle i. The semi-major axis $\alpha_i = a_i + \delta$ and semi-minor axis $\beta_i = b_i + \delta$ are defined such that any collision will be avoided. The inequality constraint for each disc of the robot with respect to the obstacle is

$$c_{k,i}^{\text{dyn},j}(\mathbf{z}_{k,i}) = \begin{bmatrix} \Delta x_{k,i}^j \\ \Delta y_{k,i}^j \end{bmatrix}^{\text{T}} R(-\phi_i)^{\text{T}} \begin{bmatrix} \frac{1}{\alpha_i^2} & 0 \\ 0 & \frac{1}{\beta_i^2} \end{bmatrix} R(-\phi_i) \begin{bmatrix} \Delta x_{k,i}^j \\ \Delta y_{k,i}^j \end{bmatrix} > 1 \tag{10}$$

IV. RESULTS

This section presents simulation results for different navigation scenarios. First, we introduce the experimental setup used. Then, in Section IV-B, we compare our framework with two baseline approaches. We present results demonstrating

our method's ability to perform collision avoidance and regulation compliance. Moreover, in Section IV-C, we present experimental results on the MIT's Roboat platform [17].

A. Experimental Setup

1) Hardware Setup: We use the quarter-scale Roboat described in [17] for real-world experiments. The entire framework can run on one onboard computer equipped with an Intel i7 CPU. A Velodyne LiDAR is used for perception, and a Swift Nav GPS in combination with a LORD Microstain IMU is used for localization.

2) Software Setup: The motion planner is implemented as a ROS node in C++ and Python. The planner runs onboard at 5 Hz. We rely on FORCES PRO [26] for solving the non-convex model predictive contouring control equation (3). If the solver does not find a feasible solution within the maximum number of iterations, a zero command will be sent towards the thrusters, resulting in a deceleration of the Roboat. We use a planning horizon of 10 s and 20 steps. Further, we employ OpenCV [27] to divide the occupancy grid into convex shapes for static collision avoidance.

B. Simulation

We compare our approach (RA-MPCC) with the following baseline methods:

- *LMPCC* [5] without regulation awareness. A repulsive cost similar to J_{HT} is still employed, but centered on the obstacle.
- *Breadth First Search (BFS) local planner and an NMPC tracking controller* [17]. Since BFS does not allow for dynamic obstacles in its planning, the space occupied by obstacle boats is added to the static occupancy grid.

To simulate the other boats, we replay real vessel trajectories navigating the in the Amsterdam canals, as presented in Fig. 7, collected using the Automatic Identification System (AIS) [28]. The global path is designed such that the Roboat has to interact with the obstacle boat. The eight situations include head-on encounters, taking over other vessels, crossing from starboard, and crossing from port, evenly represented. Furthermore, we show multi-robot coordination with RA-MPCC in Fig. 10 where two vessels run our method and perform a head-on (left figure) and a crossing scenario (right figure) while respecting the regulations.

1) Dynamic Collision Avoidance: The results presented in Table I demonstrate that our approach outperforms the baseline methods in terms of percentage of collisions. Qualitative results presented in Fig. 8, shows that RA-MPCC and the original LMPCC method can avoid dynamic obstacles in different situations. However, the head-on and take over cost function J_{HT} can help the solver choose between avoidance on the starboard or the port side, therefore starting the manoeuvre earlier on and resulting in a safer motion. In contrast, the BFS combined with the NMPC closely follows the global path without considering the future obstacle positions, resulting in a high number of collisions.

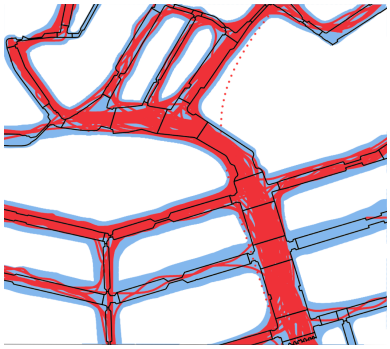


Fig. 7: Map of the Amsterdam canals and trajectories collected using the Automatic Identification System (AIS). The canals are depicted in black and in red the vessel's trajectories. Segments of these canals were used as simulation environments.

Method	% Right-Handed Violations	% Collisions
LMPCC	64.20	1.23
BFS + NMPC	68.32	36.36
RA-MPCC	**19.38**	**0.00**

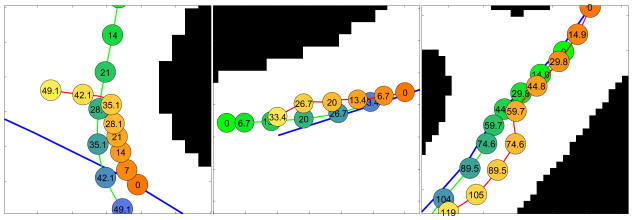
TABLE I: Results for eight different scenarios, each ran ten times. These test cases include head-on, take-over, crossing from port and starboard. Out of all violations (left and right-handed), the percentage of right-handed ones is calculated for each run. The percentage shows the mean over all runs.

2) Compliance to Regulations: Similar to [25], we define situations where the regulations are breached for both the right-handed and the left-handed rules. Right-handed regulations are the norm for vessels, for example, giving way to someone from starboard. In contrast, left-handed regulations require giving way to a vessel nearing from port. Violations to these regulations are registered when an obstacle vessel is in one of the rectangular regions with a specific orientation with respect to the Roboat (see Fig. 9) for more than dt seconds. For this experiment we have used $dt = 0.17$.

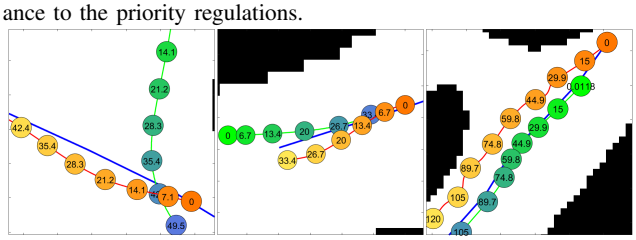
The quantitative results presented in Table I and qualitative results displayed in Fig. 8, show that RA-MPCC incurs in the lower number of right-handed violations relative to the left-handed ones. RA-MPCC mainly produces paths that pass the other boat on their port side for head-on encounters and take-overs. LMPCC and BFS, instead, are more likely to cross with the obstacle vessel's starboard side, not complying with the inland waterway regulations. Moreover, our method will plan a trajectory that does not cross the trajectory of an obstacle approaching from starboard. Furthermore, Fig. 10 shows two decentralized multi-robot coordination scenarios where all agents ran our RA-MPCC without communicating with each other. In both the head-on and crossing situations, the vessels avoid each other while complying with the regulations.

C. Real-World Experiments

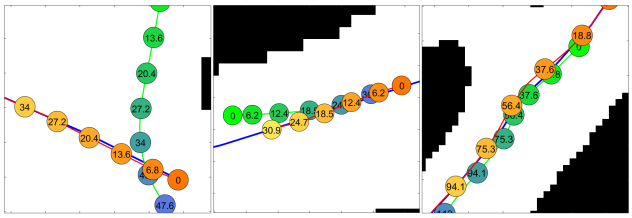
RA-MPCC was implemented on a quarter-scale Roboat [17] for testing at the Marineterrein in Amsterdam (Fig. 11).



(a) *RA-MPCC* achieves successful obstacle avoidance and compliance to the priority regulations.



(b) *LMPCC* also results in successful obstacle avoidance but does not generate regulation-aware trajectories.



(c) *BFS combined with NMPC* follows the global path closely and is not able to anticipate the moving obstacles. We see more dangerous situations and no regulation compliance.

Fig. 8: The Roboat (orange-yellow) avoiding an obstacle boat (green-blue) while following a global reference path. The obstacle boat is executing a trajectory taken from a real-world data set. Timestamps are displayed in seconds, and the static obstacles are represented in black.

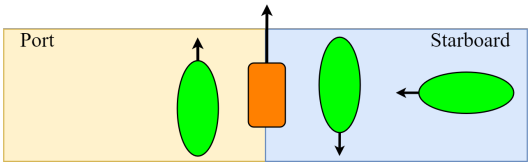


Fig. 9: Violations of the right-handed priority regulations. The Roboat should never be in a situation where an obstacle boat is in one of the configurations shown in the figure.

For more details on the real-world experiments, we refer the reader to the accompanying video.

V. CONCLUSIONS AND FUTURE WORK

This paper proposed a motion planning framework called RA-MPCC for ASVs in urban canals based on LMPCC. This framework is able to plan local trajectories that avoid dynamic obstacles according to the regulations. We compare our method to the original LMPCC method and a BFS local planner combined with an NMPC tracking controller. Simulated experiments, executed on Amsterdam's canal segments with real vessel trajectory data, have shown that RA-MPCC outperforms both methods in urban canals. Moreover, we have shown that RA-MPCC also performs well in a multi-agent coordination scenario where all vessels run the proposed method. As future work, better predictions of the obstacle vessels' motion can be incorporated into the framework to plan more efficient trajectories in crowded environments.

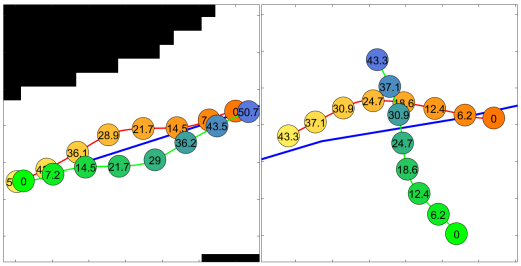


Fig. 10: Two vessels both running the proposed RA-MPCC method. On the left, the two vessels encounter each other head-on. On the right, a crossing is performed. In both scenarios the regulations are satisfied. Timestamps are displayed in seconds.

Fig. 11: The quarter-scale Roboat [17] using RA-MPCC at the Marineterrein in Amsterdam.

ACKNOWLEDGMENT

We would like to acknowledge the staff from the Roboat project at AMS Institute for assisting with the experimental tests. In particular, we would like to thank Jonathan Klein Schiphorst, Joshua Jordan, and Rens Doornbusch. In addition, we would like to acknowledge the MIT team for providing the quarter-scale Roboat and the NMPC code.

REFERENCES

[1] Gemeente Amsterdam, "Amsterdamse Thermometer van de Bereikbaarheid," Gemeente Amsterdam, Tech. Rep., 2021.

[2] W. Wang, T. Shan, P. Leoni, D. Fernandez-Gutierrez, D. Meyers, C. Ratti, and D. Rus, "Roboat II: A Novel Autonomous Surface Vessel for Urban Environments," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[3] Rijksoverheid, "wetten.nl - Regeling - Binnenvaartpolitiereglement - BWBR0003628," 2017. [Online]. Available: https://wetten.overheid.nl/BWBR0003628/2017-01-01

[4] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah, "Trajectory Planning for Autonomous High-Speed Overtaking in Structured Environments Using Robust MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2310–2323, 2020.

[5] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 4459–4466, 2019.

[6] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016.

[7] Y. Gu, J. C. Goez, M. Guajardo, and S. W. Wallace, "Autonomous vessels: state of the art and potential opportunities in logistics," *International Transactions in Operational Research*, p. itor.12785, 2020.

[8] C. Zhou, S. Gu, Y. Wen, Z. Du, C. Xiao, L. Huang, and M. Zhu, "The review unmanned surface vehicle path planning: Based on multi-modality constraint," *Ocean Engineering*, vol. 200, p. 107043, 2020.

[9] W. Jing, C. Liu, T. Li, A. B. Rahman, L. Xian, X. Wang, Y. Wang, Z. Guo, G. Brenda, and K. W. Tendai, "Path Planning and Navigation of Oceanic Autonomous Sailboats and Vessels: A Survey," *Journal of Ocean University of China*, vol. 19, no. 3, pp. 609–621, 6 2020.

[10] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Control Engineering Practice*, vol. 61, pp. 307–316, 4 2017.

[11] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *IEEE Intelligent Vehicles Symposium*. IEEE, 6 2017, pp. 174–179.

[12] N. E. Du Toit and J. W. Burdick, "Robot Motion Planning in Dynamic, Uncertain Environments," *IEEE Transactions on Robotics (T-RO)*, vol. 28, no. 1, pp. 101–115, 2 2012.

[13] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 9 2016.

[14] A. Turnwald and D. Wollherr, "Human-Like Motion Planning Based on Game Theoretic Decision Making," *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, 1 2019.

[15] B. Paden, M. Cap, S. Z. Yong, D. Yershov, E. Frazzoli, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles (T-IV)*, vol. 1, no. 1, pp. 33–55, 4 2016.

[16] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern, and M. Webster, "Odin: Team VictorTango's entry in the DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 467–492, 8 2008.

[17] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle," in *IEEE International Conference on Robotics and Automation (ICRA)*, 5 2018, pp. 6189–6196.

[18] H. Zhu and J. Alonso-Mora, "Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 2, pp. 776–783, 2019.

[19] E. Alcalá, V. Puig, and J. Quevedo, "LPV-MP planning for autonomous racing vehicles considering obstacles," *Robotics and Autonomous Systems*, vol. 124, p. 103392, 2 2020.

[20] H.-T. L. Chiang and L. Tapia, "COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 2024–2031, 7 2018.

[21] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, 12 2016.

[22] B. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "The branching-course model predictive control algorithm for maritime collision avoidance," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 10 2019.

[23] Y. Zhao, W. Li, and P. Shi, "A real-time collision avoidance learning system for Unmanned Surface Vessels," *Neurocomputing*, vol. 182, pp. 255–266, 3 2016.

[24] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 1 2014.

[25] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially Aware Motion Planning with Deep Reinforcement Learning," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 2017-Septe, pp. 1343–1350, 3 2017.

[26] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: an efficient implementation of interior-point... methods for multistage nonlinear nonconvex programs," *International Journal of Control*, pp. 1–17, 2017.

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] "Automatic Identification System (AIS) Overview." [Online]. Available: https://www.navcen.uscg.gov/?pageName=AISmain

3

Additional Information on Methods

This chapter provides additional information to elucidate some of the methods presented in the scientific paper (Chapter 2). Section 3.1 explains how the static obstacles are represented. Additionally, in Section 3.2, the manner in which the priority boats are determined is demonstrated. Moreover, as an additional method, a Social Variational Recurrent Neural Network (VRNN) network is used to predict the vessel trajectories (Section 3.3).

3.1. Static Obstacle Representation

To simplify the optimization problem, the occupancy grid map containing the space occupied by static obstacles $\mathcal{O}^{\text{static}}$ is translated into linear constraints as in [43]. The process is visualized in Fig. 3.1. Starting with converting the static obstacle into a convex hull using OpenCV [44]. Then, a line is drawn from the closest point of the convex hull to the center of circle j representing the Roboat. The resulting constraint is normal to this drawn line and at a $r_{\text{disc}} + \delta$ distance from the convex hull with δ being a safety margin. This linear constraint can be defined by a vector $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$ and scalar b. Then the produced constraint is the following, in which $\mathbf{p}_j$ denotes the position of the disc j.

$$c^{\text{stat},j}(\boldsymbol{z}_0) = \boldsymbol{a} * \mathbf{p}_j - b + (r_{\text{disc}} + \delta) \leq 0 \tag{3.1}$$

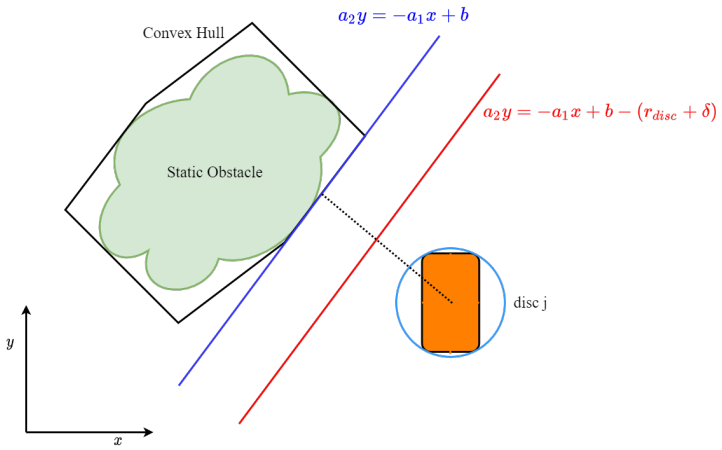


Figure 3.1: The static obstacles are represented by convex hulls, from which linear constraints are determined. The Roboat is represented with one disc for simplification.

3.2. Right of Way

There are two types of priority vessels. The first one always has priority status, because it is a large boat (longer than 20 meters), professional vessel, sailing boat, or a boat powered by muscle. The second type is a

small motor boat that is approaching from starboard side. For this last option, we define two angles α and β to describe the relative position and heading of the obstacle boat in Section 3.2.1. Then, in Section 3.2.2, the full algorithm allocating priority statuses to vessels is shown.

3.2.1. Priority Through Position

Small motor boats can deserve right of way depending on their position and orientation with respect to the Roboat. To measure this relative position and orientation, α and β are defined. α is the angle between the x'-axis of the local Roboat coordinate system and the position of the obstacle boat defined in the Roboat body frame $\boldsymbol{p}^{\text{obs, } RB}$. Similarly, β is the angle between the x'-axis and the position of the Roboat both in the obstacle coordinate system $\boldsymbol{p}^{\text{Roboat, } OB}$. These two angles are visualized in Figure 3.2. The angles should be computed for each obstacle vessel i. However, i is omitted in the equations for readability. To calculate these angles, first the relative positions in both coordinate systems are derived derived from the global positions and the rotation matrix R.

$$\begin{aligned} \boldsymbol{p}^{\text{obs, } RB} &= \boldsymbol{R}(-\psi)(\boldsymbol{p}^{\text{obs, } W} - \boldsymbol{p}^{\text{Roboat, } W}) \\ \boldsymbol{p}^{\text{Roboat, } OB} &= \boldsymbol{R}(-\phi)(\boldsymbol{p}^{\text{Roboat, } W} - \boldsymbol{p}^{\text{obs, } W}) \end{aligned} \tag{3.2}$$

Then, the angles can be obtained the following equation in which $\boldsymbol{e}'_x$ denotes the unit vector in the x'-direction of the body-fixed frame. The sign of the y-coordinate of the other boat in body frame is added to cover the full 2π rad range around the boats, except for only the 0 to π rad range the arccosine is able to output.

$$\begin{aligned} \alpha &= \text{sgn } y^{\text{obs, RB}} * \cos^{-1}\left(\frac{\boldsymbol{e}^{RB}_{x'} \cdot \boldsymbol{p}^{\text{obs, } RB}}{|\boldsymbol{p}^{\text{obs, } RB}|}\right) \\ \beta &= \text{sgn } y^{\text{Roboat, OB}} * \cos^{-1}\left(\frac{\boldsymbol{e}^{OB}_{x'} \cdot \boldsymbol{p}^{\text{Roboat, } OB}}{|\boldsymbol{p}^{\text{Roboat, } OB}|}\right) \end{aligned} \tag{3.3}$$

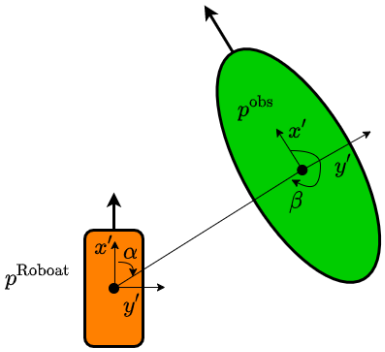


Figure 3.2: The variables α and the β describe the relative orientation of the Roboat and the obstacle boat. These angles are used to determine whether it is necessary to give way to an obstacle boat.

3.2.2. Algorithm

To determine which boats should be allocated a priority status, Algorithm 1 is constructed. First, the algorithm checks whether the boat is of a specific priority type or length. If not, it is analyzed in what area the Roboat and the obstacle boat see each other. The angles defining these areas are taken from COLREGS [39]. If α is larger than $\frac{1}{30}\pi$ rad, but smaller than $\frac{5}{8}\pi$ rad, the obstacle boat is in the crossing giving-way area from the perspective of the Roboat. However, the Roboat only has to give way if β is larger than $-\frac{5}{8}\pi$ rad and smaller than $-\frac{1}{30}\pi$ rad, which means that the Roboat is seen in the crossing stand-on area by the obstacle boat. Thus, if both α and β are within the right range, the boat is assigned the priority status.

Note that the priority boats are determined for the current time step and do not change for the entire prediction horizon. This way, the cost function does not change drastically during the optimization.

Algorithm 1 Determine priority boats

Sense positions and orientation of the Roboat and obstacles
Determine length and type of the obstacle boats
for Each obstacle boat i **do**
  Calculate relative positions $\boldsymbol{p}_i^{\text{obs, } RB}$ and $\boldsymbol{p}_i^{\text{Roboat, } OB}$ (3.2)
  Calculate α_i and β_i (3.3)
  if The obstacle boat is a professional vessel **or** boat length > 20 **then**
    Allocate boat i as a priority boat
  else if $\frac{1}{30}\pi \leq \alpha \leq \frac{5}{8}\pi$ **and** $-\frac{5}{8}\pi \leq \beta \leq -\frac{1}{30}\pi$ **then**
    Allocate boat i as a priority boat
  end if
end for

3.3. Vessel Trajectory Predictions

The Regulations Aware Model Predictive Contouring Control (RA-MPCC) method, like other trajectory receding horizon planners, needs to know the future positions of the obstacle vessels in advance. Traditional approaches, like the original LMPCC [11], estimate the vessel's motion plans with simple prediction models such as the constant velocity model. There are two problems with these simple prediction models. First, they can oversimplify the vessel's trajectories. Second, they do not account for interactions. As discussed in Section 1.2.4, this can lead to the freezing robot problem in crowded environments. As a solution to this, we explored using a Social Variational Recurrent Neural Network (VRNN) [41] to generate open-loop and environment-aware predictions. The neural network is employed to learn vessel's trajectories from data (see Section 4.1.1). Since the relative positions and velocities of the other agents are considered for the trajectory predictions, interactions can be learned from this data set.

4

Additional Information on Results

In order to show the performance of RA-MPCC, three different types of experiments are performed. Two experiments are performed in simulation, and one experiment is done in an outdoor environment with disturbances. In this chapter, the implementation of the methods, design of the experiments, and the results will be further elaborated.

In Section 4.1, the details regarding the implementation of the methods are discussed. This section includes the training and the dataset of the Social VRNN, specific parameter values for the MPCC methods, and the original Roboat local planning and control method. After that, the details on the experiment design and its results are discussed in Sections 4.2 and 4.3 for the simulation experiments and Section 4.4 for the real world experiments.

4.1. Experimental Setup

4.1.1. Social VRNN

The Social VRNN was trained on a real-world data set of vessel trajectories gathered from the Automatic Identification System (AIS) [45] (Fig. 4.1b). For training, a subsection containing various canal segments with different widths and orientations was selected (see Fig. 4.1a). The length of the dataset for training was 1 hour. Unfortunately, the AIS trajectories available to us contained sparse data points. Because of interpolation between those data points, the paths sometimes seem to cross the boundaries of the canals. Therefore, the accompanying map was adjusted, so no paths overlapped with the occupancy grid (Figure 4.1b).

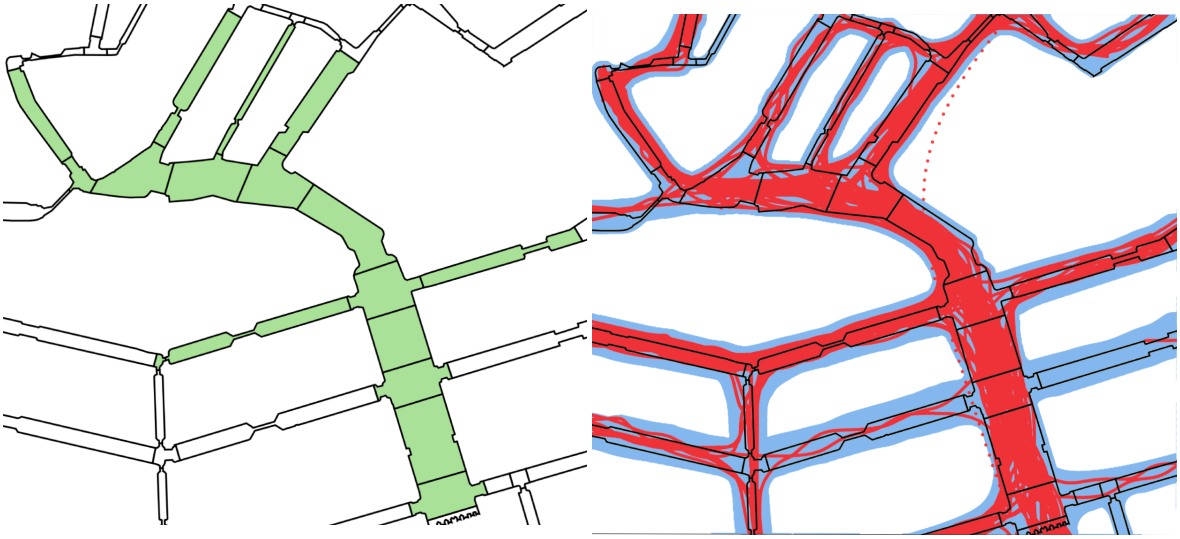
The original Social VRNN method was trained on pedestrian trajectories. However, vessels behave differently than walking humans. For example, vessels are less maneuverable. While on the other hand, humans operating a vessel tend to consider a larger space around them. Therefore, some hyper parameters were adjusted. For instance, the time step is increased from 0.4 s to 1.0 s, enlarging the prediction horizon is from 4.8 s to 12.0 s. Additionally, the part of the occupancy grid surrounding the agent that is considered is enlarged from 6x6 to 60x60 meters. The predictions node is running at 1 Hz.

4.1.2. RA-MPCC and LMPCC

This section discusses the implementation of both the Model Predictive Contouring Control (MPCC) methods. Since RA-MPCC is built upon LMPCC, the implementation is similar; only the cost function is different. It must be noted that we do not compare the proposed RA-MPCC with the original LMPCC method. The original repulsive cost function was a round exponential function. Instead, the repulsive regulations function J_{HT} was brought back to the center by setting g and h to zero because the new repulsive cost function is a better fit for the vessels' long shapes. Also, σ_x and σ_y are decreased for a smaller cost function. In addition to that, the right-of-way costs J_{RoW} were removed for the LMPCC version.

Parameter Values

As presented in the preliminaries of the scientific paper, a discrete-time nonlinear differential equation is used to describe the dynamics of the boat. The corresponding parameter values are obtained through grey-box system identification in [42] and displayed in Table 4.1a. Additionally, the dimensions of both the quarter-



(a) Visualized is a part of Amsterdam's canals. Various canal segments (green) with different shapes, sizes and orientations were selected for training.

(b) An example of the vessel trajectories in red. The black lines display the canal segments, and in blue, the enlarged canals.

Figure 4.1: The Automatic Identification System (AIS) data set.

scale and full-scale boat are displayed in Table 4.1b. The radius of the discs representing the Roboat is chosen so that the discs cover the full boat.

The optimization problem described in Section 2 contains many design weights and variables to define the shape of the regulations cost function. The design weights and reference velocities used for the input, tracking, and speed costs can be found in Table 4.1c. These values are the same for both the LMPCC and the RA-MPCC variation. However, only RA-MPCC contains the head-on and takeover costs J_{HT}, and Right of Way (RoW) costs J_{RoW}. For the LMPCC version, the RoW costs are omitted. Additionally, the values of the head-on and takeover costs are adjusted such that it results in a central repulsive cost function with an ellipsoidal shape. The values for the corresponding parameters can be found in Table 4.1d.

The design weights are tuned such that obstacle vessels are avoided according to the regulations. The motivation behind the choice of parameters is as follows. First of all, the thrusters on the front and the rear of the boat (thrusters 3 and 4) have a much higher penalty than those on the side (thrusters 1 and 2). This difference is because the idea behind the design of the full-scale Roboat is that thrusters 3 and 4 will only be used for small sideways motions, like the mooring on the dock. Second, the contour error weight Q_{contour} is much smaller than the lag error weight Q_{lag}. In order to avoid obstacles, we allow deviation from the reference path. Moreover, the velocity reference u_{ref} is higher than the maximum allowed velocity on the canals (1.67 m/s). Otherwise, the vessel moves rather slow. Lastly, the parameters' values, a through f, defining the shape of the repulsive cost function, are determined such that there is a higher cost on the front and the starboard side of the obstacle vessel. The costs for crossing the path of a priority vessel Q_{RoW} is higher than the regulation costs Q_{reg} encouraging to choose the right side for head-on and takeover encounters.

4.1.3. Breadth First Search combined with Nonlinear Model Predictive Control

For the motion planning and control of the Roboat, a hierarchical approach is taken, displayed in Fig. 4.2. In this section, this approach is briefly discussed together with the relevant settings we choose for our experiment.

The current local planning method utilized for the Roboat is a Breadth First Search (BFS). First, the occupancy grid is transformed into a graph. After that, a path will be searched from the current position of the Roboat to a point on the global path that is further away without touching any occupied areas. The found local path will be sent towards the Nonlinear Model Predictive Control (NMPC) tracking controller [42], with a reference velocity of 1.39 m/s. This is smaller than the u_{ref} (1.95 m/s) for the MPCC methods because the NMPC is more likely to follow its path at the reference speed (see Section 4.3.4).

In order to model exactly the same scenarios for the BFS combined with NMPC, some efforts were made. The inputs that Roboat's local planner required were slightly different. The most important adjustment worth mentioning is the change in the occupancy grid. Where the MPCC local planners received the dynamic ob-

m_{11}	m_{22}	m_{33}	d_{11}	d_{22}	d_{33}
12.982	23.318	1.273	6.012	7.112	0.771

(a) The diagonal mass and drag matrix values for the dynamical model of the quarter-scale Roboat [42].

	a	b	r_{disc}	# discs
Quarter-scale	0.45	0.90	0.28	3
Full-scale	2.00	4.00	1.38	3

(b) The dimensions of the quarter-scale and full-scale boat, and r_{disc} the radius of the discs representing the Roboat.

$Q_{\text{u1, 2}}$	$Q_{\text{u3, 4}}$	Q_{contour}	Q_{lag}	Q_{v}	u_{ref}
0.50	10.00	0.03	2.00	22.00	1.95 m/s

(c) Design weights for input, tracking, and speed costs of the MPCC method. $\boldsymbol{Q}_c$ consists of a weight for the contour Q_{contour} and lag error Q_{lag}. Also, the $\boldsymbol{Q}_{\text{u}}$ is divided into $Q_{\text{u1,2}}$ and $Q_{\text{u3,4}}$, the numbers denoting the thruster number. u_{ref} is the surge velocity reference.

Method	Q_{HT}	Q_{RoW}	g	h	c	d	e	f
RA-MPCC	23.00	41.00	2.00	4.80	2.00	6.40	44.40	14.40
LMPCC	23.00	0.00	1.60	3.00	0.00	0.00	-	-

(d) Value for the regulation costs $J_{\text{reg}} = J_{\text{HT}} + J_{\text{RoW}}$. The motion planning method is evaluated in the LMPCC version with a central repulsive cost function and the proposed RA-MPCC with a decentralized cost function. Since Q_{RoW} is zero for LMPCC, the values for e and f are not used for that variation.

Table 4.1: The parameter values used for the RA-MPCC and LMPCC method.

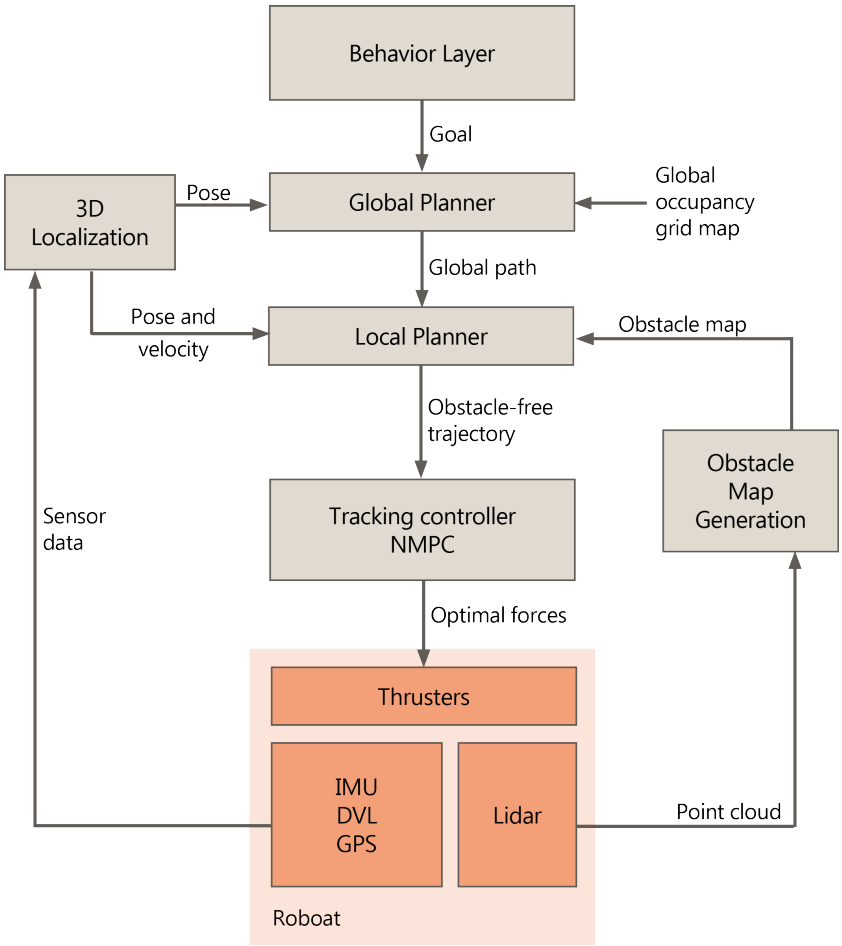


Figure 4.2: Software structure of the Roboat methods. The Breadth First Search (BFS) local planner receives the obstacle map and the current state of the Roboat. This BFS method plans an obstacle-free trajectory, which will be sent towards the Nonlinear Model Predictive Control (NMPC) tracking controller. An optimization problem will be solved to calculate the optimal forces. These are then sent towards the thrusters of the Roboat.

stacles separately, for the BFS method, the dynamic obstacles had to be added to the occupancy grid. To determine which cells were occupied with the dynamic obstacles, the dynamic constraint c^{obst} was used. 2.0 m padding is added around the obstacles, which is r_{disc} = 1.38 m with some extra margin.

4.2. Simulation Experiment Design

To measure the performance of our proposed RA-MPCC method, experiments in simulation were executed. In this section, the design of these experiments and the motivation behind the metrics chosen to evaluate the performance.

4.2.1. Experiment Design

Eight different scenarios displayed in Fig. 4.3 are designed to evaluate the proposed method. Scenarios 1 and 2 are crossings with a vessel from port, while those with a vessel nearing from starboard are displayed in scenarios 3 and 4. Moreover, in scenarios 5 and 6, the Roboat will take over another boat. Lastly, a head-on encounter is shown in scenarios 7 and 8. The obstacle vessel trajectory and the static obstacle map are taken from the real-world AIS data set. The reference trajectory is designed such that an encounter with the moving obstacle will happen.

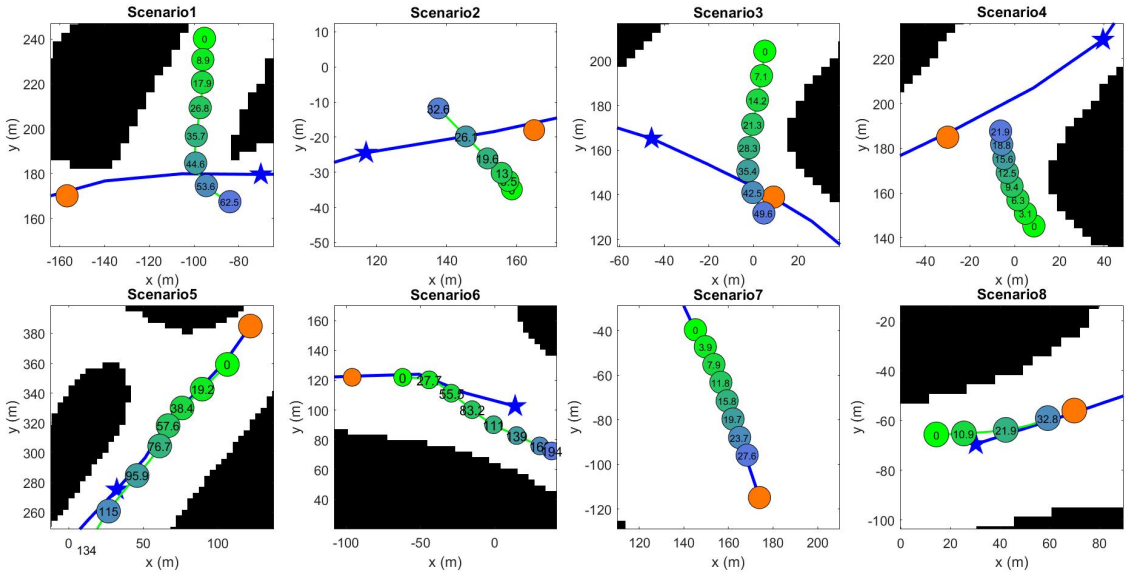


Figure 4.3: The different scenarios used for the simulation experiments. The obstacle follows a trajectory taken from the AIS data set. This trajectory is displayed in green to blue. The starting point of the Roboat is an orange circle. The reference path given to the Roboat is dark blue with a blue star for the goal.

4.2.2. Metrics

Four different metrics were selected to evaluate the performance concerning the obstacle avoidance, regulations compliance, and interaction-awareness. In this section, the motivation behind this selection and the calculation of these metrics are further elaborated. First, the percentage of collisions and the minimum separation distance are often used to represent the performance regarding collision avoidance. Second, as explained in Section 2, the percentage of right-handed rule violations inspired from [31] is chosen to evaluate the regulation compliance. Eventually, it was possible to evaluate the interaction awareness with the traveled distance and the minimum separation distance. When accounting for interactions, it can be assumed that the obstacle vessel also is planning on avoiding the Roboat in real-life situations. Therefore it would be less necessary for the Roboat to avoid the obstacle vessel, resulting in a smaller traveled distance [13]. Because in our simulation, the obstacle boats only follow the trajectory from the data set without avoiding any other obstacles, the minimum separation distance would therefore be smaller.

Method	% Right-Handed Violations	% Collisions	Traveled Distance in m (std)	Mean Min Separation Distance in m (std)
LMPCC	64.20	1.23	71.10 (50.52)	13.42 (6.45)
BFS + NMPC	68.32	36.36	57.14 (63.42)	12.98 (14.32)
RA-MPCC				
const. vel.	**19.38**	**0.00**	67.95 (53.30)	10.56 (3.59)
S. VRNN	34.77	**0.00**	69.84 (47.18)	11.11 (4.84)

Table 4.2: Statistical results for 80 test cases, including head-on, take-over, and crossing from port and starboard. Eight different scenarios are run 10 times. The percentage of right-handed violations shows how many times the right-handed norm violation is breached with respect to both left-handed and right-handed violations. What number of runs resulted in collisions is shown by the percentage of collisions. The traveled distance denotes the distance traveled by the Roboat. The mean minimum separation distance is the distance between the center of the Roboat and the center of the obstacle boat.

4.3. Additional Simulation Results

In the results section of the scientific paper, a brief overview was given of the results. In this section, more detailed results of the performance for each separate scenario are given. In addition to that, the results on randomly varying the initial conditions and the global path are discussed. Furthermore, the velocity of the vessel and the computation time necessary are analyzed. Lastly, the predictions generated by the Social VRNN are compared with constant velocity predictions.

4.3.1. Expanded Statistical Results

The scientific paper (see Chapter 2) gave a brief overview of the most important statistical results. In Table 4.2 also the traveled distance and the mean minimum separation distance are shown. Additionally, the results for the two types of vessel trajectory predictions for RA-MPCC are compared, the constant velocity model and Social VRNN. The minimum separation distance is the smallest measured distance between the center of the Roboat and the center of the obstacle vessel for one test run. The traveled distance denotes the length of the executed path by the Roboat.

The percentages of right-handed violations and collisions show that RA-MPCC outperforms the other methods in regulations compliant collision avoidance of vessels. More specifically, the constant velocity prediction model results in more adherence to the rules than the Social VRNN. On the other hand, BFS combined with NMPC produces the most efficient paths because the traveled distance is the smallest. However, this also results in the most collisions; see Section 4.3.3 for more details. The mean minimum separation distance shows that the MPCC methods keep a decent distance to the obstacle boat. Still, it is hard to draw conclusions from the mean minimum separation distance because this depends on how the cost function is tuned.

4.3.2. Regulation Compliance

There were eight scenarios in total: two crossing from starboard, two crossing from port, two head-on encounters, and two takeovers. To better understand the percentage of right-handed rule violations per method, we had a closer look at each scenario in Table 4.3. For the crossing scenarios, it was counted in how many runs a crossing occurred. It is desired not to cross in a situation with a boat nearing from starboard, while the Roboat is allowed to cross when a boat comes from port. When looking at the crossing from port section, we see that RA-MPCC has approximately the same amount as crossing or even more than the other two methods. However, for a situation where the obstacle vessel nears from starboard, the RA-MPCC method clearly has fewer crossings. According to the regulations, this is the desired outcome since the Roboat should give way an obstacle vessel from starboard.

For the takeovers and head-on encounters, is it desired to deflect towards the port side of the other vessel. Therefore the amount of takeovers and head-on encounters where the Roboat deflected towards the vessel's port side is compared to the total amount of encounters. It can be concluded that RA-MPCC outperforms the other methods because there are more deviations towards the correct side for this method. It is interesting to see that the social VRNN predictions give a better result in head-on encounters. At the same time, the constant velocity prediction model performs better in giving way.

4.3.3. Collisions

There were no registered collisions for the RA-MPCC method. However, there was one with the LMPCC method in a head-on encounter (scen. 7). Still, the BFS combined with NMPC resulted in the most colli-

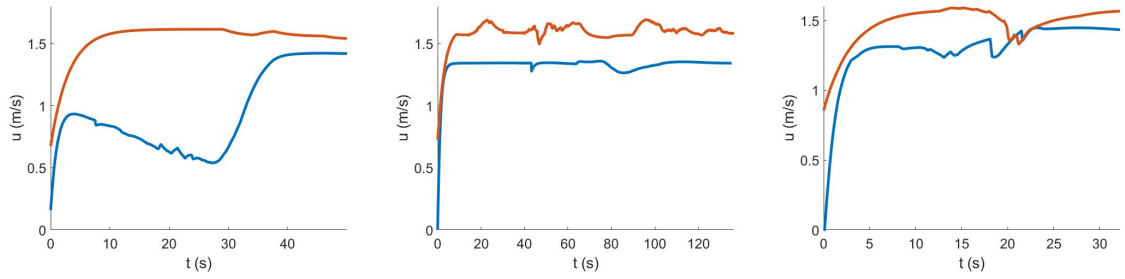
Method	**% Cross. Port**	scen. 1	scen. 2	**% Cross. Star.**	scen. 3	scen. 4
LMPCC	76.19	100.00	50.00	100.00	100.00	100.00
BFS + NMPC	50.00	100.00	0.00	100.00	100.00	100.00
RA-MPCC						
const. vel.	85.00	100.00	70.00	**0.00**	0.00	0.00
S. VRNN	55.00	100.00	10.00	65.00	50.00	20.00
	% Takeover Norm Pref.	scen. 5	scen. 6	**% Head-On Norm Pref.**	scen. 7	scen. 8
LMPCC	50.00	0.00	100.00	0.00	0.00	0.00
BFS + NMPC	14.29	0.00	100.00	9.09	18.18	0.00
RA-MPCC						
const. vel.	**60.00**	100.00	20.00	95.00	100.00	90.00
S. VRNN	55.00	100.00	10.00	**100.00**	100.00	100.00

Table 4.3: Behaviour for all four scenarios: crossing an obstacle from port and starboard, taking over another vessel, and a head-on encounter. Each scenario is run ten times with the same initial conditions. For the crossing scenarios, it is determined for each run whether the Roboat did or did not cross. Then the percentage of crossings is calculated. Crossing should not happen in the starboard situation, while they are allowed in the port situation. For the takeover situation, the percentage of right-handed norm takeovers compared to all takeovers are determined. The same goes for the head-on encounters. Note that the percentages are calculated for the two scenarios together and separately. Sometimes there are more encounters in one scenario, which results in the overall percentage not being the average of the two separate percentages.

sions: 36,36 % in total. Every run with scenario 8 (head-on) and scenario 1 (cross. port) resulted in a collision. Additionally, all runs with scenario 7 (head-on) except for one ended in an accident with the obstacle vessel.

4.3.4. Velocity

During the simulations, it was noticed that the BFS combined with NMPC reached higher velocities than the MPCC methods. In order to further analyze this behavior, the velocities are plotted in Fig. 4.4. The reference velocity for the BFS method was set to 1.39 m/s, while for the MPCC method, this was set to 1.95 m/s. Both methods do not have a stable speed at their reference velocity but also do not exceed the maximum velocity (Section 1.1.2). The original Roboat planning method, BFS combined with NMPC, often reached a higher speed. On the other hand, the RA-MPCC method results in a lower velocity. On the other hand, the RA-MPCC method results in a lower velocity because other aspects of the cost function are also being optimized.



(a) Scenario 3, Crossing Starboard (b) Scenario 5, takeover (c) Scenario 8, Head-On

Figure 4.4: The surge velocity u plotted against time in seconds for RA-MPCC with constant velocity predictions (blue) and the BFS combined with NMPC (red). The BFS combined with NMPC reaches a higher velocity for all situations. In scenario 3, it is visible that the Roboat slows down to give way to the obstacle vessel.

4.3.5. Random Initial Conditions

The planned trajectory depends on the boat's initial conditions and the hand-crafted global reference path. To counterbalance this dependency, the initial conditions and global reference path were adjusted by adding a random value from a uniform distribution range to all parameters. The value ranges are displayed in Table 4.4. In total all eight scenarios were started 24 times, with new initial conditions every four runs. So six different random settings per scenario. Experiment runs, which did not have an entire run, for example because the scenario was accidentally cut off early, were removed from the data. In total, we had between 144

	x_0 (m)	y_0 (m)	ψ_0 (rad)	u_0 (m/s)	x_{ref} (m)	y_{ref} (m)	path angle (rad)
Min	−5.0	−5.0	$-\pi/8$	0.0	−3.0	−3.0	$-\pi/10$
Max	5.0	5.0	$\pi/8$	0.5	3.0	3.0	$\pi/10$

Table 4.4: The experiment scenarios are randomized by adding a value to the initial conditions and the global reference path coordinates. In this table the minimum and maximum boundaries of these additional values are displayed per parameter.

Method	**% Right-Handed Violations**	**% Collisions**	**Traveled Distance in m (std)**	**Mean Min Separation Distance in m (std)**
LMPCC	60.25	1.23	67.83 (53.58)	13.83 (6.06)
BFS + NMPC	65.36	32.64	78.50 (58.59)	9.07 (7.84)
RA-MPCC				
const. vel.	**32.25**	0.70	67.64 (47.53)	10.92 (6.44)
S. VRNN	39.29	**0.00**	63.86 (42.54)	12.09 (7.77)

Table 4.5: Statistical results for approximately 150 test cases, including head-on, take-over, and crossing from port and starboard per method. All eight scenarios were started 24 times, with new initial conditions every four runs. The percentage of right-handed violations shows how many times the right-handed norm violation is breached with respect to both left-handed and right-handed violations. What number of runs resulted in collisions is shown by the percentage of collisions. The traveled distance denotes the distance traveled by the Roboat. The mean minimum separation distance is the distance between the center of the Roboat and the center of the obstacle boat.

and 162 runs per method.

The results regarding the mean minimum separation distance, percentage of right-handed rule violations, percentage of collisions, and the traveled distance can be found in Table 4.5. Moreover, the detailed results for all scenarios separately are displayed in Table 4.6. Compared to fixed scenarios in Tables 4.3 and 4.2, the numbers are quite different. The most considerable difference is for the head-on scenario 7 performed with LMPCC, which had a percentage difference of 45. Overall, the numbers show a lot more nuance instead of completely following or not following the regulations. Nevertheless, we see that the RA-MPCC is still outperforming the other methods with a smaller percentage of right-handed violations. Again, we see the Social VRNN version performing better than the constant velocity prediction alternative for the head-on encounters and as well for the takeovers. Still, it is difficult to draw hard conclusions from these randomized results, considering that the random conditions were not the same for every scenario. The performances can easily be biased because there were only six different random settings per scenario and the range for the random initialization was quite large (see Table 4.4).

4.3.6. Computation Time

A common concern with optimization-based methods is the ability to solve the problem within the realtime constraints. To take away this concern, the solver time was researched for all four methods in one of the most difficult scenarios to solve, a direct head-on encounter. The results are visualized in Fig. 4.5. First of all, it is visible that the optimization problem can be solved within the loop rate of our controller (200 ms) on an Intel Core i7 CPU. The NMPC method performs rather consistently regarding the computation time. For the MPCC methods, on the other hand, it depends on the situation. Because this was a direct head-on encounter, the LMPCC had difficulty choosing between the two different sides. As a result, we see higher computation times around 70 on the x-axis, even crossing the real-time constraint. In this situation, this did not happen for the RA-MPCC, still the same could happen in another situation. Moreover, it is interesting to see that the calculation period rises every time a new Social VRNN prediction is received.

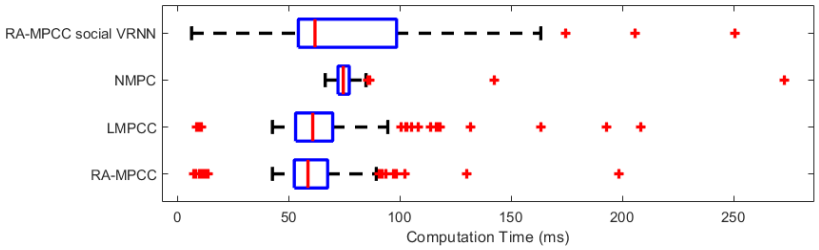
4.3.7. Performance of the Social VRNN Predictions

Social VRNN is employed as a prediction model to generate more accurate predictions than simple models like the constant velocity model. Moreover, because Social VRNN takes other agent's the relative positions and velocities as an input, the network might account for interactions.

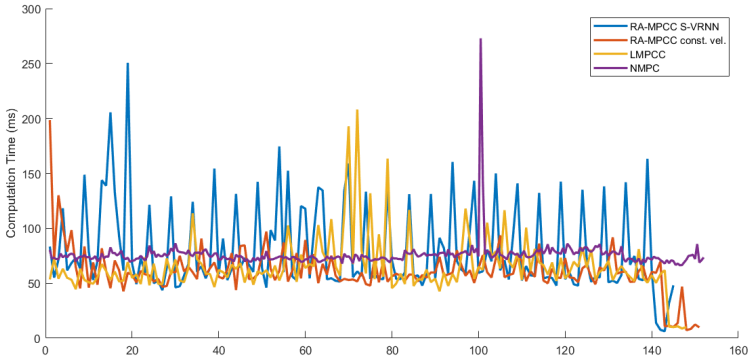
In order to evaluate the performance of the Social VRNN predictions, it will be compared to the traditional constant velocity model. The resulting trajectories of the two prediction models with RA-MPCC often seem similar, as visualized in Fig. 4.6. However, Tables 4.2 and 4.6 show a worse performance regarding the right-handed violations. More specifically, the Social VRNN version does not perform well in right of way situations (see Tables 4.3 and 4.5), but slightly outperforms the constant velocity model in head-on situations.

Method	**% Cross. Port**	scen. 1	scen. 2	**% Cross. Star.**	scen. 3	scen. 4
LMPCC	68.57	68.75	68.42	89.19	100.00	78.95
BFS + NMPC	58.33	100.00	21.05	100.00	100.00	100.00
RA-MPCC						
const. vel.	59.38	92.31	70.59	**39.47**	15.79	70.59
S. VRNN	58.97	85.00	33.33	62.16	47.37	73.68
	% Takeover Norm Pref.	scen. 5	scen. 6	**% Head-On Norm Pref.**	scen. 7	scen. 8
LMPCC	35.90	0.00	100.00	44.00	45.16	42.11
BFS + NMPC	21.05	5.88	78.57	21.05	42.11	0.00
RA-MPCC						
const. vel.	80.00	100.0	57.14	71.05	100.00	42.11
S. VRNN	**88.89**	100.00	78.57	**76.32**	84.21	68.42

Table 4.6: Behaviour for all four scenarios with random initial conditions: crossing an obstacle from port and starboard, taking over another vessel and a head-on encounter. Each scenario is run 24 times with the different initial conditions every four runs. For the crossing scenarios, it is determined for each run whether the Roboat did or did not cross. Then the percentage of crossings is calculated. Crossing should not happen in the starboard situation, while they are allowed in the port situation. For the takeover situation, the percentage of right-handed norm takeovers compared to all takeovers are determined. The same goes for the head-on encounters. Note that the percentages are calculated for the two scenarios together and separately. Sometimes there are more encounters in one scenario, which results in the overall percentage not being the average of the two separate percentages.

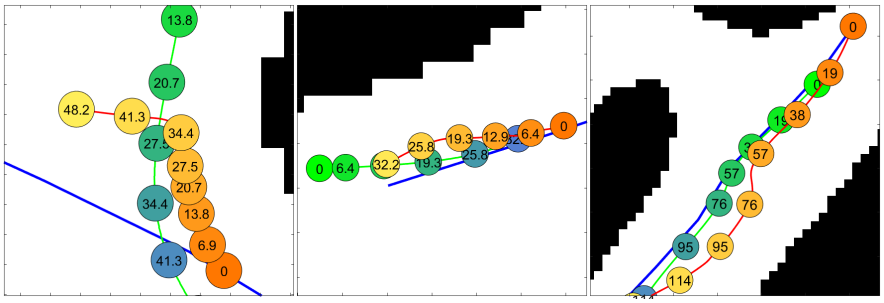


(a) Boxplot comparing the computation times for all four methods.

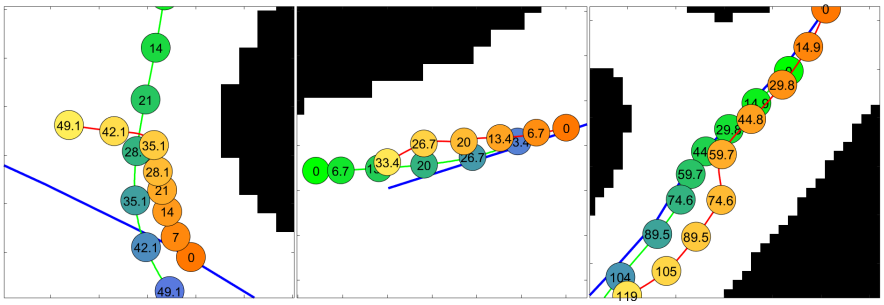


(b) The computation times plotted for the full duration of the scenario.

Figure 4.5: The time it took to execute the computation is analyzed for all four methods for one head-on scenario (scen. 7). The NMPC method had a relatively constant computation duration, while the calculation periods for the MPCC methods depended a lot on the situation.



(a) Social VRNN



(b) Constant Velocity Model

Figure 4.6: The Roboat (orange-yellow) running RA-MPCC with two different prediction models avoiding an obstacle boat (green-blue) while following a global reference path. The obstacle boat is executing a trajectory taken from a real world data set. Timestamps are displayed in seconds and the static obstacles are represented in black.

	Cross. Port (scen. 1)	**Cross. Star. (scen. 3)**	**Takeover (scen. 5)**	**Head-On (scen. 8)**
Const. Vel.	8.11 (12.17)	4.42 (5.81)	2.36 (2.8)	9.79 (5.55)
Social VRNN	15.38 (10.36)	7.07 (6.27)	19.35 (10.44)	4.81 (4.51)

Table 4.7: The Mean Squared Error calculated between the actual vessel trajectory and the predictions as an outcome of the social VRNN or the constant velocity assumption. The corresponding standard deviation is displayed between brackets.

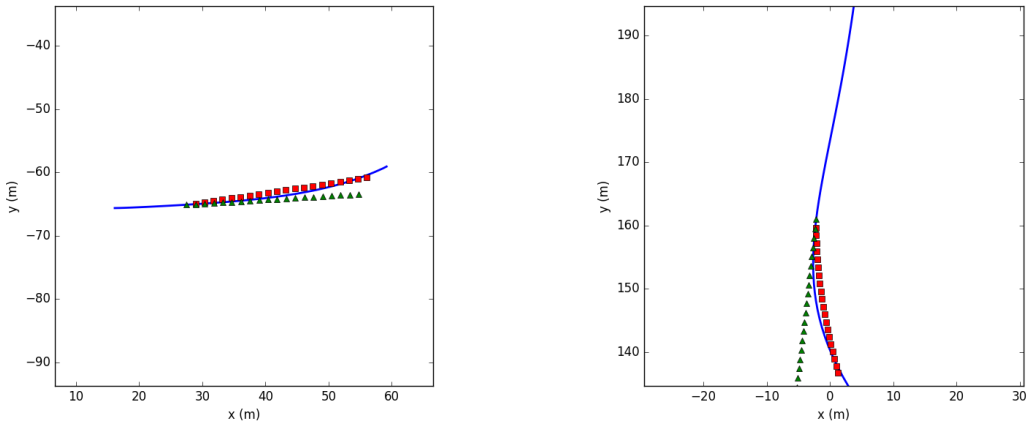
If a motion planning method accounts for interactions, the traveled paths would be more effective, and thus the traveled distance would be smaller [13]. Also, because the predefined obstacle trajectory does not react to the Roboat, the minimum separation distance would be smaller. It is hard to tell based on the metrics whether the predictions account for interactions. Which prediction model results in the smallest traveled distance and mean minimum separation distance depends on the scenario.

In order to analyze where this difference in performance comes from, the Mean Squared Error (MSE) between the predicted trajectory and the executed trajectory was computed for four different runs with each a different scenario (Table 4.7). We do expect that the MSE will be worse for the neural network. This network should account for the vessel's environment, reacting to the Roboat. On the contrary, the obstacle vessel does not react and follows a predetermined trajectory. However, we can conclude from this table that overall the constant velocity predictions are overall better at predicting the future vessel positions than the Social VRNN predictions. From Fig. 4.7 and Fig. 4.8, it is visible that the Social VRNN often predicts a curve, while the vessel's motions in these scenarios are straight. These predicted curves are often not a logical reaction to the static environment or the Roboat. Therefore, we can conclude that the predictions generated by the Social VRNN are inaccurate.

4.4. Additional Information on Real-World Experiments

To test our motion planning framework in the real world we selected the quarter-scale Roboat [42] as the experimental platform displayed in Figure 4.9. Since the software is the same for both the quarter-scale and the full-scale Roboat, our method could also be used for the full-scale vessel. The only adjustments that would have to be made are the dynamics, the dimensions, and the weights of the optimization problem.

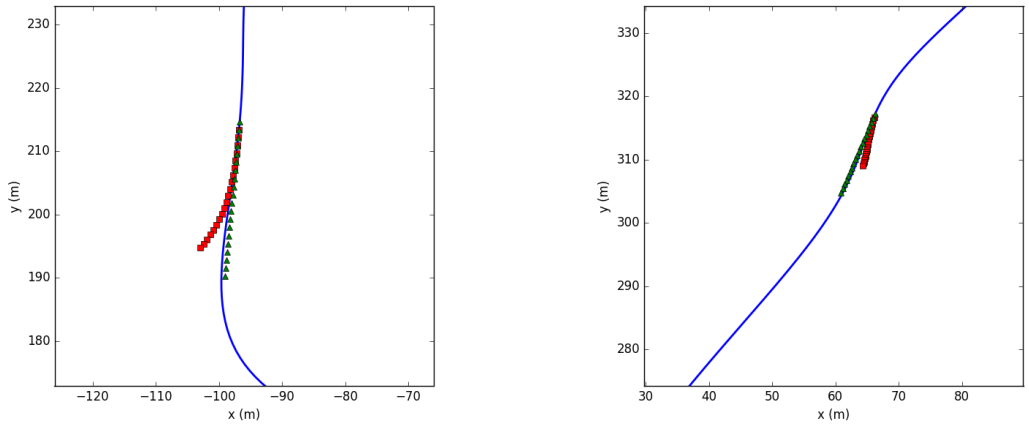
The aim was to show static and regulations compliant dynamic collision avoidance. Because the perception software module was not complete, the Roboat was not able to identify dynamic obstacles. As a solution



(a) Head-on scenario, in which the Social VRNN outperforms the constant velocity predictions.

(b) Crossing with a boat from starboard, a scenario in which the MSEs are close to each other.

Figure 4.7: The examples of successful Social VRNN predictions (red squares) compared to the constant velocity predictions (green triangles), displayed for a specific time step. The actual trajectory of the obstacle vessel is drawn in blue.



(a) Crossing with a boat from port, a scenario in which the MSE is almost double for the Social VRNN compared to the constant velocity model.

(b) Takeover scenario, in which the Social VRNN MSE is eight times bigger with respect to the constant velocity model.

Figure 4.8: Inaccurate Social VRNN predictions (red squares) compared to the constant velocity model (green rectangle). In the figure, a specific time step is displayed. The actual trajectory of the obstacle vessel is drawn in blue.

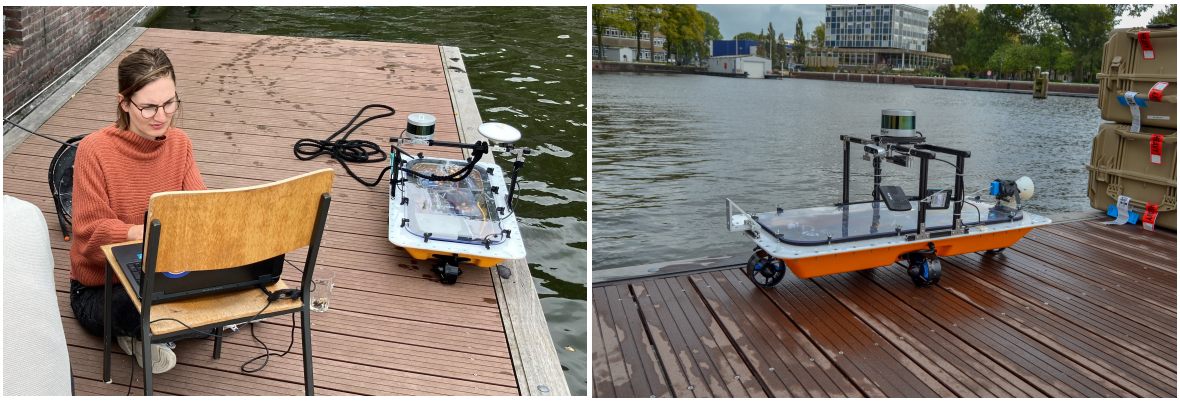


Figure 4.9: The quarter-scale Roboat is used for the outdoor experiments.

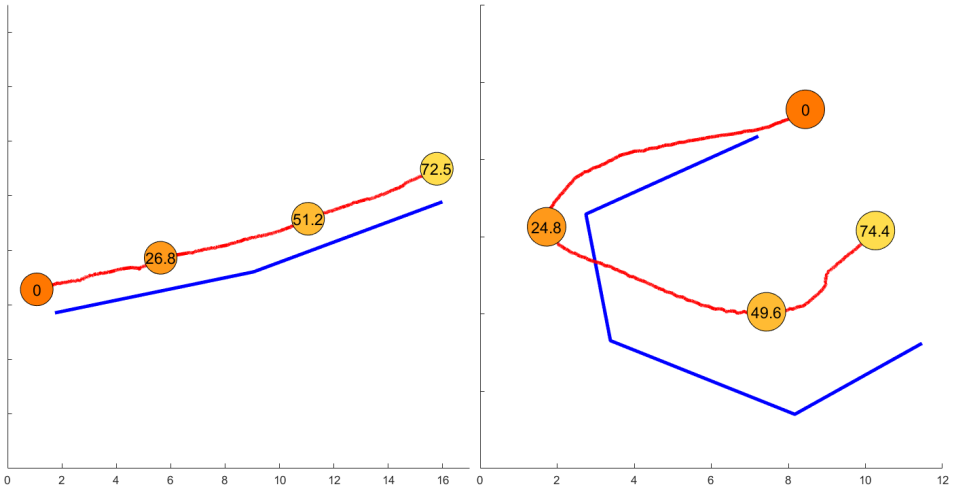


Figure 4.10: The quarter-scale Roboat following two paths in a real-world environment, including wind (originates in the left) with strengths of 8.0-11.0 m/s. The scale is in meters.

to that, we planned on using another quarter-scale Roboat as a moving vessel. This vessel would then communicate its pose and velocity to the other Roboat, running the RA-MPCC method. Because of hardware and software development and limited testing time, few experiments with the real quarter-scale Roboat have been performed as of yet.

However, we were able to run the RA-MPCC method in the real world in an outdoor environment following a reference path. On the testing day, the wind was quite strong. In the environment of Amsterdam, wind velocities of 8.0-11.0 m/s were measured. Despite this, we achieved global path following with minimal tuning of the weights and the dynamics. For the dynamic parameters, the values found in [42] were used. The results of the Roboat following a global path defined by waypoints are displayed in Fig. 4.10. As can be concluded from the data, even with the strong disturbances and the minimal tuning, the boat followed the global path in general. Because there is no substantial penalty on the contour error in our method to achieve obstacle avoidance, we allow for some deviation from the global path. The effect of the small contour design weight is visible in the resulting paths.

In simulation, we already proved that we are able to find a locally optimal path. Together with the real-world reference path following, the proposed method could work in theory if the perception module is finished.

5
Discussion

During the evaluation of RA-MPCC, several limitations of the current method were noticed. In this chapter, these limitations and corresponding solutions will be addressed. This chapter is divided in three different sections: Section 5.1 Obstacle Avoidance, Section 5.2 Regulation-Compliance, and Section 5.3 Vessel Trajectory Predictions.

5.1. Obstacle Avoidance

5.1.1. Local Optima

What can be seen from the results in the scientific paper is that RA-MPCC is good at avoiding obstacles compared to the other methods. However, there are some situations in which collisions still occur. A well-known problem for real-time optimization-based methods is that they are able to find a local optimum instead of the globally optimal trajectory. Because there are multiple local optima, it can happen that the solver will switch between two local optimal trajectories, for example between passing the other boat on starboard or port. If this happens for too long and the motion planner does not commit to either of the trajectories, no avoidance will happen. Eventually, this results in a collision. Our asymmetric cost function helps in the case of a direct head-on encounter with another boat. The trajectory will be pushed towards the side with the smaller costs. However, when there is a direct head-on encounter with the center of the cost function, the solver still experiences a difficult time.

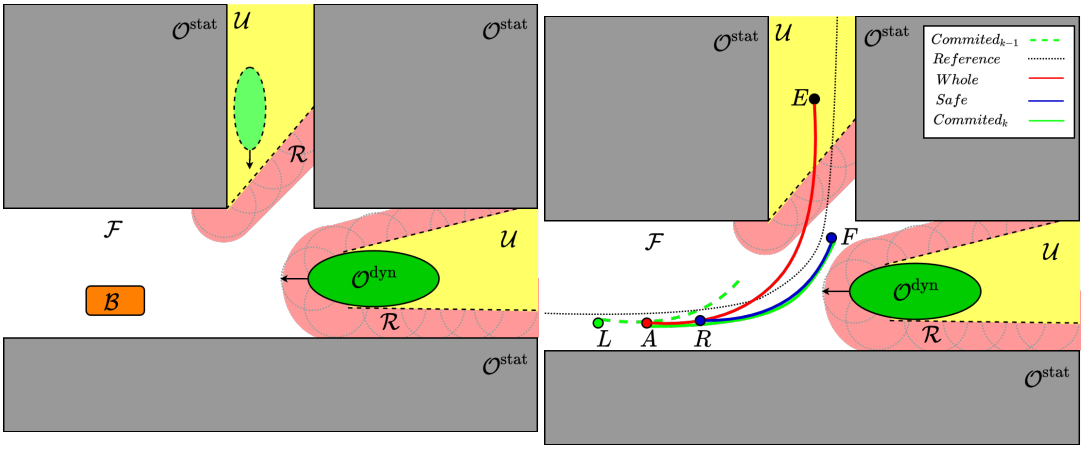
Two solutions might help in such a case. The first solution is a global planner that also plans its way-points approximately around the obstacles. Then, the motion planner will have an extra incentive for choosing one of the sides. The second solution is combined with a place-dependent regulation that was not implemented in our motion planning method. In the urban canals, it is preferred that you navigate on the starboard side of the canal. Adding this regulation to the optimization problem either through a cost function or a cost map also encourages choosing one side over the other.

5.1.2. Perception

In order to avoid obstacles, they first have to be perceived. In this thesis, it was assumed that the obstacles' positions, size, and velocities were known. However, in the current software of the Roboat project, there is no distinction between static and dynamic obstacles. The LiDAR produces a point cloud, and all obstacles are seen as static obstacles. For the future, it is highly recommended that this perception module is further developed so that it is able to detect and track moving obstacles. A good start was made during my internship, during which an object detector was developed.

Limited Field of View

Another assumption made during this thesis is that the dynamic obstacles are always within the field of view. While in practice, there are many bridges, intersections, and narrow canals with obstacles. This might lead to the situation where an obstacle suddenly appears in an area that was previously believed to be safe to plan a trajectory in. While always planning in a safe space leads to unnecessary slow and conservative paths. On the other hand, planning in an unknown space can lead to dangerous situations. In [46], an answer was found to this problem; an additional backup-trajectory was calculated in safe space. However, this specific method



(a) The Roboat situated in the free known space $\mathcal{F}$. The most righter obstacle boat occupies the known dynamically occupied space $\mathcal{O}^{dyn}$, while the other moving obstacle is situated in the unknown space $\mathcal{U}$. Knowing the maximum velocities on the canals, an additional risk region $\mathcal{R}$ is defined. $\mathcal{O}^{stat}$ denotes the space occupied by static obstacles.

(b) Each time step two trajectories will be calculated. Whole from the current position of the Roboat A to goal point E. And a second Safe backup trajectory from the next point in time on the Whole trajectory R to a point in the safe environment F. The safe area is the known free space minus the risk area.

Figure 5.1: When planning in both the free known space as in the unknown space, a safe backup trajectory can help in case the optimization problem is not feasible in the next time step.

did not account for moving obstacles. Inspired by that method, a possible answer to motion planning with limited field of view is first identifying an additional risk region in which a possible collision might happen (see Fig. 5.1a). We can identify this risk region since we know the maximum velocity on the canals. Next, the safe space can be defined as the known free space minus the risk region. Eventually, each time step, not only a trajectory will be planned in the to-be-believed free space, but also a backup trajectory is planned in the safe space as in Fig. 5.1b. Then, in a case where the solver fails to find a solution, the backup trajectory can be used instead.

5.2. Regulation-Compliance

As shown in the scientific paper (Chapter 2) and the results (Chapter 4), the proposed RA-MPCC can comply with the inland waterway regulations that apply in the Amsterdam canals. Still, there are some discussion points regarding the regulation-compliance. Namely, it is hand-crafted and not suitable for all situations. Moreover, the cost function does not incorporate place-dependent rules and is dependent on the global reference path. These issues and possible solutions will be discussed in this section.

5.2.1. Inverse Reinforcement Learning

The presented regulations cost function is a hand-crafted cost function that can achieve regulations compliance measured on the amount of hand-crafted violations. While more methods come up with their own cost function [30, 31] to adhere to regulations, these cost functions are still objective to what the designer sees as good trajectories. The rules seem straightforward. For example, give way to a vessel approaching from starboard. But when exactly does a boat approach from starboard, and when is it far away enough so that it is not necessary to give way? Humans do keep the regulations in mind but often handle them with their intuition. To create more human-like avoidance according to the rules, the cost function presented in this thesis could be used for inverse reinforcement learning. The parameters and weights of the cost function will then be learned from a human vessel trajectory data set.

5.2.2. Generalized Cost Function

Furthermore, it is challenging to design a cost function suitable for all different kinds of situations. The proposed cost function is suitable for many situations, as presented in the results. But there are two situations in which the current cost function is lacking. Starting with being taken over by another boat. Namely, in that situation, it is preferred to continue the route it was following, maybe move a bit towards starboard. But the vehicle taking over should do the maneuver. However, the presented cost function is designed to react to obstacle boats and even push Roboat towards the port side. This behavior is the opposite of what is desired.

The second situation in which the cost function is lacking is crossing with a vessel coming from starboard. In this situation, it is desired to slow down and wait for the other vehicle to pass. But because of the velocity reference costs in our optimization problem, this will not happen. The Roboat will a little bit slow down but not enough to let the boat cross. Therefore, it will move around the rear of the other boat to not cross the other vehicle's path and minimize the velocity reference costs at the same time (see Fig. 4.6b). However, the weight on the velocity reference should be high enough to act adequately in the case of take-overs and head-on encounters.

There might be a different cost function that is more fitting for all situations. However, the chances are that none of the cost functions are perfect for all possible scenarios. There is, however, a solution to make this cost function work for more situations. The optimization problem being solved every time step from scratch allows for completely changing the weights of the cost function depending on the situation. Changing weights is similar to the so-called gain scheduling [47], often used to control a nonlinear plant with one controller while switching gains. This changing of weights is already proven to be effective for the right of way costs that can be turned on or off, depending on the obstacle's current relative position and orientation. This way, for example, the velocity reference weight could be decreased in a right of way encounter. Moreover, for being taken over, the regulation costs weights could be reduced.

5.2.3. Global Reference Trajectory

In this thesis, it was assumed that this global reference path was provided. Still, the resulting local trajectories are highly dependent on the global reference path. Therefore, the global planner should not generate routes that are the fastest from A to B. It should create more natural, safe, and regulation-compliant trajectories. The social trajectory planner [48], for example, could help plan natural global trajectories. As a bonus, these trajectories naturally consider the place-dependent regulation, which dictates that a vessel should navigate on the starboard side of the canal.

5.3. Vessel Trajectory Predictions

We hypothesized that more accurate vessel trajectories, including interactions with other agents, could be learned by the Social VRNN from an AIS data set. However, during the simulation experiments (Section 4.3) it was found that the regulation compliance was worse for the RA-MPCC with Social VRNN predictions than with the baseline constant velocity model. This was a result of inaccurate vessel trajectory predictions. Often the predicted trajectories did not represent the actual trajectories. Additionally, there was no proof for interaction awareness.

The cause of these inaccurate predictions and not accounting for interactions can have three reasons: the network itself, the training, or the data set. The network and the training were not within the scope of this thesis work and will therefore not be discussed. However, we believe that adjusting the network and the training can result in accurate predictions, outperforming the simple constant velocity model. The third reason for inaccurate predictions without interaction awareness lies within the AIS data set. In Section 5.3.1, the results of gathering data via AIS are discussed. Moreover, Section 5.3.2 elaborates on the way the static obstacles are represented. At last, we suggest constructing an improved data set in Section 5.3.3.

5.3.1. Automatic Identification System

There are several problems caused by using AIS [45] to gather information about the vessel trajectories. First of all, the data points generated by the AIS system are very sparse. Therefore, we had to linearly interpolate the different data points to create enough data points for the Social VRNN training. Therefore, the vessel trajectories were not detailed, and occasionally they crossed the boundaries of the static occupancy grid.

Additionally, AIS is only mandatory for large and professional vessels [10], resulting in two problems. First, there are fewer trajectories in the data set than when small ships would also be required to have an AIS. Therefore, few interactions are being captured in the data set. Secondly, these specific ships with an AIS are often identified as priority vessels, which do not have to give way to smaller vessels. Moreover, their trajectories are much different from trajectories of smaller vessels.

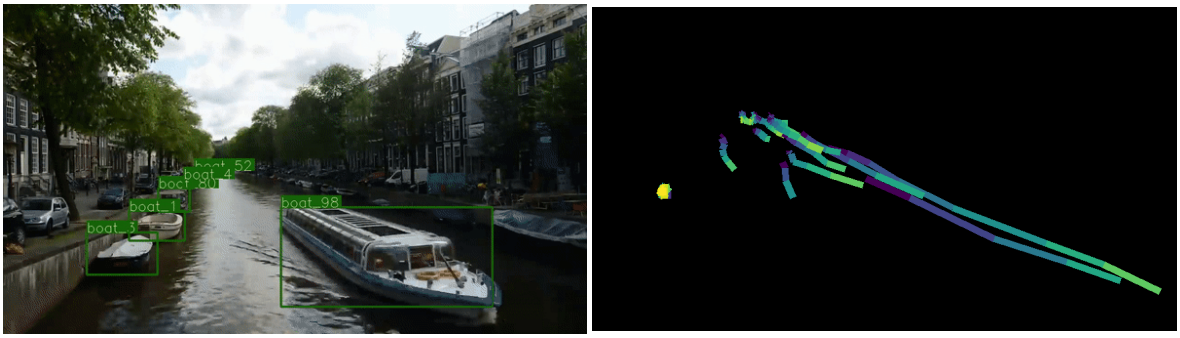
5.3.2. General Map

A general map from Amsterdam was used to represent the static obstacles in the environment of the subject vessel. However, as visible in Figure 5.2, this general map did not include any details. Not only the permanent bridges and piers were not included in the map, but also the semi-permanent houseboats were ignored. It

(a) General occupancy grid used for training of the Social VRNN

(b) The actual situation with bridges, moored houseboats and piers. [49]

Figure 5.2: Part of the Amstel between the Doelensluis and the Blauwbrug. The actual situation shows more semi-permanent static obstacles than the general occupancy grid captures.



(a) Detected boats with YOLOv3 [51]

(b) Tracking trajectories with SORT [52]

Figure 5.3: An improved dataset could be generated by setting up cameras. Vessel trajectories can be determined by detecting and tracking vessels. [50]

doesn't make sense to make the static map more detailed since the data points given by the AIS system are very sparse. However, for a carefully constructed dataset, it would be useful to construct a detailed static occupancy map for every ten minutes or so. This way, also large moored boats are included as static obstacles.

5.3.3. Improved Data Set

To conclude, the AIS is not able to accurately capture the interactions of between boats. Therefore, a different solution must be searched to generate a vessel trajectory data set. One option would be to set up cameras and use object detection and tracking to capture more detailed trajectories (see Figure 5.3 [50]). Moreover, the static environment can then also be determined from the images.

6

Conclusions and Future Work

6.1. Conclusions

In this thesis, a regulations aware motion planning framework for ASVs in urban canals. The method is called Regulations Aware Model Predictive Contouring Control (RA-MPCC) and is based on LMPCC. RA-MPCC plans dynamically feasible trajectories in real-time that avoid static and dynamic obstacles. One of the main contributions of this work is regulation compliance through the addition of a new cost function.

The framework is compared against LMPCC and the method the Roboat currently uses for motion planning and control, BFS combined with NMPC. For this comparison, eight different scenarios were designed with all different types of encounters with obstacle vessels. It became clear that the BFS combined with NMPC had difficulty dealing with moving obstacles, resulting in many collisions. These collisions were because this method followed the reference path closely, only deviating when there was no other option. Moreover, the future positions were not predicted, leading to reactive behavior. To evaluate the regulations compliance, the percentage of right-handed rule violations is defined. From this percentage, we can conclude that the proposed cost function improved adherence to the rules for almost all scenarios. In addition, we showed that two boats, both running RA-MPCC, were successfully able to avoid each other according to the regulations.

For the MPCC method, it is necessary to know in advance the other vessel's motion plans. Often a simple prediction model such as the constant velocity model is used. However, this assumption is not always accurate, for example, when the vessel takes a turn. Moreover, these methods do not account for interactions. Therefore, we researched the difference between the constant velocity predictions and predictions generated by Social VRNN. Eventually, the output of the Social VRNN was less accurate than the constant velocity predictions. Therefore, the RA-MPCC performed worse combined with Social VRNN.

At last, the method is validated in a real-world outdoor environment with disturbances. During these experiments, we were able to show that the RA-MPCC was able to run in real-time and onboard the quarter-scale Roboat platform. Additionally, the Roboat followed a reference path with two different shapes even with severe wind.

To conclude, the method presented in this thesis is a new step in the direction of ASVs navigating autonomously in the urban canals.

6.2. Future Work

Still, the motion planning framework can be improved in the future. This section summarizes the future work discussed in Chapter 5. The improvements can be categorized into three categories: obstacle avoidance, regulation compliance, and vessel trajectory predictions. First, dynamic obstacles have to be detected and tracked by a perception module before they can be avoided. Additionally, the planning algorithm could account for the limited field of view by planning a backup trajectory in a safe space, as discussed in Section 5.1.2. Second, the regulation compliance can be extended with a global planner that plans waypoints around obstacles and on the starboard side of the canal. Moreover, the hand-crafted cost function can result in even more social-compliant behavior by learning the weights and parameters with IRL. In addition to that, different weights could be employed for different scenarios inspired by gain scheduling to make the cost function even more useful for all types of encounters. The last category is the vessel trajectory predictions. These can be improved by further training and tuning of the network. On top of that, a more detailed vessel

trajectory data set can be captured by cameras. With this improved data set, it might be possible to learn interactions with the Social VRNN.

Bibliography

[1] Gemeente Amsterdam, "Amsterdamse Thermometer van de Bereikbaarheid," Gemeente Amsterdam, Tech. Rep., 2021.

[2] W. Wang, T. Shan, P. Leoni, D. Fernandez-Gutierrez, D. Meyers, C. Ratti, and D. Rus, "Roboat II: A Novel Autonomous Surface Vessel for Urban Environments," in *IEEE International Conference on Intelligent Robots and Systems*, 7 2020.

[3] O. Levander, "Autonomous ships on the high seas," *IEEE Spectrum*, vol. 54, no. 2, pp. 26–31, 2 2017.

[4] Amazon, "Amazon Prime Air." [Online]. Available: https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011

[5] Waymo, "Waymo on Public Roads." [Online]. Available: https://waymo.com/press/

[6] National Aeronautics and Space Administration, "VIPER," 2020. [Online]. Available: http://www.nasa.gov/viper/images

[7] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah, "Trajectory Planning for Autonomous High-Speed Overtaking in Structured Environments Using Robust MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2310–2323, 6 2020.

[8] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Control Engineering Practice*, vol. 61, pp. 307–316, 4 2017.

[9] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *IEEE Intelligent Vehicles Symposium*. IEEE, 6 2017, pp. 174–179.

[10] Rijksoverheid, "wetten.nl - Regeling - Binnenvaartpolitiereglement - BWBR0003628," 2017. [Online]. Available: https://wetten.overheid.nl/BWBR0003628/2017-01-01

[11] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 10 2019.

[12] N. E. Du Toit and J. W. Burdick, "Robot Motion Planning in Dynamic, Uncertain Environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2 2012.

[13] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 9 2016.

[14] A. Turnwald and D. Wollherr, "Human-Like Motion Planning Based on Game Theoretic Decision Making," *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, 1 2019.

[15] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *International Conference on Intelligent Robots and Systems*, 2010, pp. 797–803.

[16] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[17] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 7 2017, pp. 285–292.

[18] Marine Advanced Robotics, "WAM-V 16 ASV." [Online]. Available: https://www.wam-v.com/wam-v-16-asv

[19] Surfbee, "Marine Robots." [Online]. Available: https://www.surfbee.io/

[20] B. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "The branching-course model predictive control algorithm for maritime collision avoidance," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 10 2019.

[21] D. K. M. Kufoalor, T. A. Johansen, E. F. Brekke, A. Hepsø, and K. Trnka, "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios," *Journal of Field Robotics*, vol. 37, no. 3, pp. 387–403, 4 2020.

[22] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and Decision-Making for Autonomous Vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 5 2018.

[23] B. Paden, M. Cap, S. Z. Yong, D. Yershov, E. Frazzoli, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 4 2016.

[24] C. Katrakazas, M. Quddus, W. H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 11 2015.

[25] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern, and M. Webster, "Odin: Team VictorTango's entry in the DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 467–492, 8 2008.

[26] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 3 2009.

[27] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 10 2008.

[28] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," 4 2016.

[29] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 50, pp. 2492–24 978, 2019.

[30] M. Everett, Y. F. Chen, and J. P. How, "Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3052–3059, 5 2018.

[31] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially Aware Motion Planning with Deep Reinforcement Learning," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 1343–1350, 3 2017.

[32] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *IEEE International Conference on Robotics and Automation*, vol. 2016-June, 6 2016, pp. 1433–1440.

[33] E. Alcalá, V. Puig, and J. Quevedo, "LPV-MP planning for autonomous racing vehicles considering obstacles," *Robotics and Autonomous Systems*, vol. 124, p. 103392, 2 2020.

[34] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 9 2015.

[35] H. Zhu and J. Alonso-Mora, "Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[36] E. F. Camacho and C. Bordons, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing. London: Springer London, 1999.

[37] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, 12 2016.

[38] G. Flores, S. Zhou, R. Lozano, and P. Castillo, "A Vision and GPS-Based Real-Time Trajectory Planning for a MAV in Unknown and Low-Sunlight Environments," *Journal of Intelligent & Robotic Systems*, vol. 74, pp. 59–67, 2014.

[39] Publicatieplatform UitvoeringsContent Nederlandse Overheid, "COLREG Convention on the international regulations for preventing collisions at sea," 2009. [Online]. Available: https://puc.overheid.nl/nsi/doc/PUC_2381_14/3/

[40] H.-T. L. Chiang and L. Tapia, "COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2024–2031, 7 2018.

[41] B. Brito, H. Zhu, W. Pan, and J. Alonso-Mora, "Social-VRNN: One-Shot Multi-modal Trajectory Prediction for Interacting Pedestrians," in *Conference on Robot Learning*, 2020.

[42] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle," in *IEEE International Conference on Robotics and Automation*, 5 2018, pp. 6189–6196.

[43] M. Spahn, B. Brito, and J. Alonso-mora, "Coupled Mobile Manipulation via Model Predictive Control with Free Space Decomposition," *IEEE International Conference on Robotics and Automation*, 2021.

[44] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[45] "Automatic Identification System (AIS) Overview." [Online]. Available: https://www.navcen.uscg.gov/?pageName=AISmain

[46] J. Tordesillas, B. T. Lopez, and J. P. How, "FASTER: Fast and Safe Trajectory Planner for Flights in Unknown Environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2019.

[47] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 10 2000.

[48] S. Park, M. Cap, J. Alonso-Mora, C. Ratti, and D. Rus, "Social Trajectory Planning for Urban Autonomous Surface Vessels," *IEEE Transactions on Robotics*, pp. 1–14, 2020.

[49] Google Maps, "Amstel," 2021. [Online]. Available: https://www.google.nl/maps/place/Amstel/@52.3677388,4.8972296,496m/data=!3m1!1e3!4m5!3m4!1s0x47c60a2e56c2abdb:0xfcf9103108e321e7!8m2!3d52.2791264!4d4.8784106

[50] M. Sukel, "Video to traffic flow information," 2019. [Online]. Available: https://github.com/maartensukel/video-to-traffic-flow-information-1

[51] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 4 2018.

[52] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *International Conference on Image Processing*, 2016.