

Safe Motion Planning and Control Using Predictive and Adaptive Barrier Methods for Autonomous Surface Vessels

Alejandro Gonzalez-Garcia, Wei Xiao, Wei Wang*, Alejandro Astudillo, Wilm Decré,
Jan Swevers, Carlo Ratti and Daniela Rus

Abstract—Safe motion planning is essential for autonomous vessel operations, especially in challenging spaces such as narrow inland waterways. However, conventional motion planning approaches are often computationally intensive or overly conservative. This paper proposes a safe motion planning strategy combining Model Predictive Control (MPC) and Control Barrier Functions (CBFs). We introduce a time-varying inflated ellipse obstacle representation, where the inflation radius is adjusted depending on the relative position and attitude between the vessel and the obstacle. The proposed adaptive inflation reduces the conservativeness of the controller compared to traditional fixed-ellipsoid obstacle formulations. The MPC solution provides an approximate motion plan, and high-order CBFs ensure the vessel’s safety using the varying inflation radius. Simulation and real-world experiments demonstrate that the proposed strategy enables the fully-actuated autonomous robot vessel to navigate through narrow spaces in real time and resolve potential deadlocks, all while ensuring safety.

I. INTRODUCTION

Autonomous Surface Vessels (ASVs) have gained attention over the past few decades, and the demand is expected to increase as the marine industry continues to expand [1]. Inland waterways, in particular, offer a sustainable mode of transporting goods and people, where ASVs can help increase safety and reduce operational errors [2]. Beyond transportation, ASVs are also helpful in other applications such as hydrographic surveying, water quality monitoring, and waste removal [3]. However, ASVs navigating inland waterways face unique challenges compared to coastal or marine scenarios [4]. The confined nature of narrow canals and rivers, combined with static and dynamic obstacles—such as other vessels—demands precise maneuvering capabilities [5], [6]. Therefore, developing safe motion planning and control systems is critical for successfully deploying ASVs

This work was supported by a grant from the Amsterdam Institute for Advanced Metropolitan Solutions (AMS) in Netherlands, and by the MIT-Belgium - KU Leuven Seed Fund from MIT International Science and Technology Initiatives (MISTI). A. Gonzalez-Garcia was supported by the Flanders Make SBO project ARENA (Agile & Reliable Navigation).

A. Gonzalez-Garcia, A. Astudillo, W. Decré and J. Swevers are with MECO Research Team, Department of Mechanical Engineering, KU Leuven, Belgium and with Flanders Make@KU Leuven, Belgium. {alex.gonzalezgarcia, wilm.decre, jan.swevers}@kuleuven.be

W. Xiao and D. Rus are with the Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139 USA. {weixy, rus}@mit.edu

W. Wang is with the Marine Robotics Lab, Department of Mechanical Engineering, College of Engineering, University of Wisconsin-Madison, Madison, WI 53706 USA. {wwang745}@wisc.edu

C. Ratti is with the SENSEable City Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. {ratti}@mit.edu

*Authors to whom correspondence may be addressed.

in these settings. In this work, we consider a collision-free motion as a safe motion.

Model Predictive Control (MPC) has become a popular strategy for motion planning and control within the robotics community [7]–[9]. The capabilities of handling multi-objective optimization, nonlinear dynamics, and system constraints have made MPC a powerful tool for integrated motion planning and control. However, challenges appear when multiple obstacles and complex shapes are considered. First, as the number of constraints increases, so does the computational complexity. This potentially endangers the real-time capabilities of the algorithm, as the solutions may not be able to meet predefined time requirements. MPC performs better with convex costs and constraint formulations, which lead to conventional ellipsoid representations of robots and obstacles. Although computationally cheap, common fixed ellipsoidal representations are overly conservative [10]. In this work, we consider a conservative representation as one that sacrifices significant free space to achieve an efficient constraint formulation. This can lead to deadlocks, as the robots cannot find a feasible solution. Control Barrier Functions (CBFs) are optimization-based methods that have gathered attention for their safety guarantees and reduced computational needs [11], [12]. CBFs are based on Barrier Functions (BFs), which are Lyapunov-type functions, extended to formulate constraints for control systems. However, classic CBFs can only handle relative-degree-one systems, *i.e.*, systems where the control input appears on the first derivative of the output. High Order CBFs (HOCBFs) [12] were proposed to create constraints for arbitrarily high relative degree systems. CBFs have high computational efficiency, since they can be implemented as a Quadratic Program (QP) when applied independently or with Control Lyapunov Functions [11].

A. Related Work

Prior work for controlling ASVs that require precise maneuvering includes grid-based motion planners, which offer solutions that can account for obstacles and spatial boundaries, but generally do not consider vehicle dynamics or the feasibility of following the calculated path. In [5], a Receding Horizon Planner (RHP) based on lexicographic search was applied to ASVs in urban waterways. Here, the RHP achieved better results than OpenPlanner [13] and the Time Elastic Band (TEB) planner on cluttered environments. Although the RHP could operate online, its obstacle avoidance was overly conservative due to the fixed-radius

inflation of the occupied pixels. Other approaches rely on local reactive planning, such as Velocity Obstacles [14], Artificial Potential Fields [15], or the Constant Avoidance Angle method [16]. Although these types of algorithms are usually computationally efficient, they are also conservative and may lead to deadlock situations in narrow or cluttered scenarios. Model Predictive Path Integral (MPPI) control was proposed in [6] to address multiple vehicles in urban waterways. MPPI leverages model knowledge, sampling techniques, and parallelization to compute a collision-free solution. While MPPI can handle non-convex or non-conservative problem formulations, its performance remains dependent on sampling quality and available computational resources.

In contrast, MPC can efficiently address convex problems, although representing obstacles as circles or ellipses is conservative [10]. Other approaches attempt to decrease the conservativeness by representing the vessel as a set of circles [17]. However, depending on the size of the boat, this strategy can still be overly conservative, and the computational complexity significantly increases as more constraints are required per obstacle. Non-conservative solutions can be achieved through polytope representations, although they often pose challenges in their formulation and handling, and more computational resources are needed compared to ellipsoidal representations. In [18], a method to formulate polytope-based constraints with discrete CBFs was proposed. The CBFs were defined as the collision constraints within an MPC framework, achieving successful performance even with complex shapes representing the robot and the environment. Still, constraint relaxations were applied to balance feasibility and safety. In [19], an MPC-CBF framework was proposed to control ASVs under disturbances robustly. Robust HOCBFs were designed to steer the boat within a boundary of the desired trajectory, considering an upper bound on the disturbances. The MPC provided an initial control reference, and the HOCBFs modified the control signal to add robustness. However, in contrast to the work in this manuscript, motion planning and obstacle avoidance were not addressed. At the core of this paper, we propose a new adaptive method to construct HOCBFs that maintains the efficiency of ellipsoid solutions but allows for a less conservative representation.

B. Contributions

This paper proposes an algorithm for safe motion planning for fully-actuated ASVs in narrow spaces with static and quasi-static floating obstacles. This work considers narrow spaces as tight spaces for the vessel to maneuver, which can only be tackled under specific orientations. A methodology is proposed to dynamically define obstacle inflation, reducing ellipsoid-based obstacle representations' conservativeness while maintaining computational efficiency. The algorithm leverages the MPC horizon to search for an approximate collision-free path, whereas High Order CBFs (HOCBFs) guarantee safety in tight spaces. The main contributions of this work include:

- An MPC-CBF framework for safe ASV collision avoidance

through narrow spaces;

- HOCBFs with adaptive safety constraint design for obstacle avoidance in narrow passages, and to remain within spatially safe boundaries;
- A safe deadlock recovery mechanism that leverages the full actuation configuration;
- Simulation and real-world experiments with a holonomic ASV, verifying the effectiveness of the MPC-CBF framework against multiple obstacles in narrow spaces.

II. PRELIMINARIES

In this section, the ASV prototype and its dynamics are described, the problem formulation is addressed, and preliminaries on CBF/HOCBF theory are introduced.

A. Quarterscale Roboat

The Quarterscale Roboat [20] is an overactuated ASV with dimensions of 0.90 m in length l , 0.45 m in width w , and 0.15 m in height, and a weight of 15 kg. Four BlueRobotics T200 thrusters actuate the vessel, capable of holonomic 2D motion. The Roboat is equipped with a Velodyne VLP-16 LiDAR and a Microstrain 3DM-GX5-IMU. The onboard computer is an Intel NUC with Linux and Robot Operating System (ROS), and a STM32F103 auxiliary microprocessor. For more details, see [20].

B. ASV Dynamics

The ASV equations of motion are described by:

$$\dot{\eta} = \mathbf{R}(\eta)\mathbf{v}, \quad (1)$$

$$\boldsymbol{\tau} = \mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v}, \quad (2)$$

where $\eta = [x \ y \ \psi]^T \in \mathbb{R}^3$ is the position and orientation in the inertial reference frame, and $\mathbf{v} = [u \ v \ r]^T \in \mathbb{R}^3$ represents the vessel velocity in the body-fixed frame. $\mathbf{R}(\eta) \in \mathbb{R}^{3 \times 3}$ is a transformation matrix, $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ represents the added mass and inertia matrix, $\mathbf{C}(\mathbf{v}) \in \mathbb{R}^{3 \times 3}$ stands for the Coriolis matrix, and $\mathbf{D}(\mathbf{v}) \in \mathbb{R}^{3 \times 3}$ is the drag matrix. $\boldsymbol{\tau} = [\tau_u \ \tau_v \ \tau_r]^T \in \mathbb{R}^3$ contains the force and torque applied by the thrusters, defined by:

$$\boldsymbol{\tau} = \mathbf{B}\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \frac{a_d}{2} & -\frac{a_d}{2} & \frac{b_d}{2} & -\frac{b_d}{2} \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}, \quad (3)$$

where $\mathbf{B} \in \mathbb{R}^{3 \times 4}$ represents the control matrix, defining the arrangement of the thrusters. $\mathbf{u} = [f_1 \ f_2 \ f_3 \ f_4]^T \in \mathbb{R}^4$ is the control vector, where f_1 , f_2 , f_3 , and f_4 correspond to the forces generated by each thruster. a_d is the distance between the port and starboard thrusters, and b_d is the distance between the anterior and rear thrusters. See [20] for more details.

C. Problem Formulation

Consider a scenario where a predefined global path is given in terms of a parameterized curve $\mathbf{p}_d(\omega) = [x_d(\omega), y_d(\omega)]^T$, where $\omega \in \mathbb{R}^+$ is the variable parameter. Then, the path-following goal is to remain close to the path,

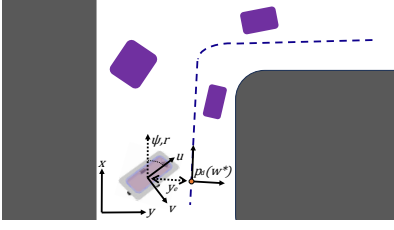


Fig. 1: The ASV with known areas for the waterway (white) and land (gray), a reference path (blue), and unknown floating obstacles (purple).

reducing the cross-track error $y_e = -\sin(\gamma_p)(x - x_d(\omega^*)) + \cos(\gamma_p)(y - y_d(\omega^*))$, and to keep the course of the ASV aligned with the path angle $\gamma_p = \text{atan2}(y'_d(\omega^*), x'_d(\omega^*))$. In these expressions, $\mathbf{p}_d(\omega^*)$ is the path reference point. $\mathbf{p}_d(\omega^*)$ is found by computing ω^* , which minimizes the Euclidean distance between the ASV and the predefined path with $\min_{\omega} f(\omega)$, where $f(\omega) = (x - x_d(\omega))^2 + (y - y_d(\omega))^2$.

The ASV operates in an environment where the boundaries of the waterway (canal, river, etc.) are assumed to be known. Still, unknown floating static or quasi-static obstacles have to be detected by local sensing (e.g., LiDAR). Thus, inequality constraints should be defined to represent collision avoidance constraints for each waterway boundary j and for each detected obstacle i , as $b_{bj} \geq 0$ and $b_{bi} \geq 0$, respectively. The local planning problem is to compute a trajectory that can remain close to the original path but avoids collisions with the spatial boundaries and the floating obstacles. Fig. 1 illustrates the problem.

D. High Order Control Barrier Functions

In this subsection, we briefly introduce the concept of high-order CBFs. If interested, the reader is referred to [12] for definitions of relative degree, forward invariance, and class $\mathcal{K}$ functions.

Consider an affine control system:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \quad (4)$$

where $\mathbf{x} \in X \subset \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times q}$ are Lipschitz continuous, and $\mathbf{u} \in U \subset \mathbb{R}^q$ is the control constraint set.

For a constraint $b(\mathbf{x}) \geq 0$ with relative degree m — i.e., we need to differentiate $b(\mathbf{x})$ m times along the dynamics (4) until the control first shows up in the corresponding derivative — $b: \mathbb{R}^n \rightarrow \mathbb{R}$, and $\psi_0(\mathbf{x}) := b(\mathbf{x})$, we first define a sequence of CBFs $\psi_i: \mathbb{R}^n \rightarrow \mathbb{R}, i \in \{1, \dots, m\}$ in the form:

$$\psi_i(\mathbf{x}) := \psi_{i-1}(\mathbf{x}) + \alpha_i(\psi_{i-1}(\mathbf{x})), i \in \{1, \dots, m\}, \quad (5)$$

where $\alpha_i(\cdot), i \in \{1, \dots, m\}$ denotes a class $\mathcal{K}$ function of $(m-i)^{\text{th}}$ order differentiable. We then define a sequence of safe sets $C_i, i \in \{1, \dots, m\}$ corresponding to (5) in the form:

$$C_i := \{\mathbf{x} \in \mathbb{R}^n : \psi_{i-1}(\mathbf{x}) \geq 0\}, i \in \{1, \dots, m\}. \quad (6)$$

Definition 1: (High Order Control Barrier Function (HOCBF)) [12] Let $\psi_i(\mathbf{x}), i \in \{1, \dots, m\}$ be defined by (5) and $C_i, i \in \{1, \dots, m\}$ be defined by (6). A function

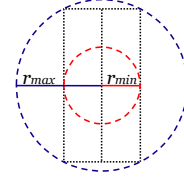


Fig. 2: ASV footprint with enclosing radius $r_{\max}$ (blue), and inner radius $r_{\min}$ (red).

$b: \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as a High Order Control Barrier Function (HOCBF) of relative degree m for system (4) if there exist $(m-i)^{\text{th}}$ order differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1, \dots, m\}$ such that:

$$\sup_{\mathbf{u} \in U} [L_f \psi_{m-1}(\mathbf{x}) + L_g \psi_{m-1}(\mathbf{x})\mathbf{u} + \alpha_m(\psi_{m-1}(\mathbf{x}))] \geq 0, \quad (7)$$

for all $\mathbf{x} \in \cap_{i=1}^m C_i$. In the above, the left part is equivalent to $\psi_m(\mathbf{x})$, and L_f, L_g denote Lie derivatives along f and along g , respectively.

The HOCBF is a general form of a CBF with relative degree one [11]. In other words, setting $m = 1$ reduces the HOCBF to the common CBF form: $L_f b(\mathbf{x}) + L_g b(\mathbf{x})\mathbf{u} + \alpha_1(b(\mathbf{x})) \geq 0$.

Theorem 1: (Safety guarantees with HOCBFs) [12] Given an HOCBF $b(\mathbf{x})$ as in Def. 1 with the associated sets $C_i, i \in \{1, \dots, m\}$ defined by (6), if $\mathbf{x}(0) \in \cap_{i=1}^m C_i$, then any Lipschitz continuous controller $\mathbf{u}(t) \in U$ that satisfies the HOCBF constraint in (7), $\forall t \geq 0$ renders $\cap_{i=1}^m C_i$ forward invariant for system (4), and we have that $b(\mathbf{x}(t)) \geq 0, \forall t \geq 0$.

III. METHODOLOGY

In this section, the proposed methodology for safe motion planning is described.

A. Architecture Overview

The proposed approach employs MPC to generate an initial trajectory while ensuring safety around floating obstacles using HOCBFs and a dynamic obstacle inflation methodology. First, unknown obstacles are detected using the LiDAR perception system, and ellipses are fitted around them. Then, an MPC problem is formulated, treating obstacles and waterway boundaries as constraints, with an inflation of the ASV's inner radius $r_{\min}$ (Fig. 2), defined as half the ASV's width plus a safety distance d_s . The MPC provides an approximate initial trajectory that is not guaranteed to be safe or collision-free.

Next, the proposed adaptive obstacle representation modifies the inflation radius, considering the relative position of each obstacle/boundary and the ASV (see implementation details in Section IV-A.3). After that, the HOCBF obstacle/boundary constraints are inflated by this new computed radius, which guarantees the vessel's safety, assuming it will not rotate on the spot. This assumption can lead to deadlock scenarios in tight spaces. Hence, a deadlock recovery algorithm is proposed to safely maneuver out of these narrow spaces (see Section III-E). Algorithm 1 illustrates the proposed framework at the end of this section.

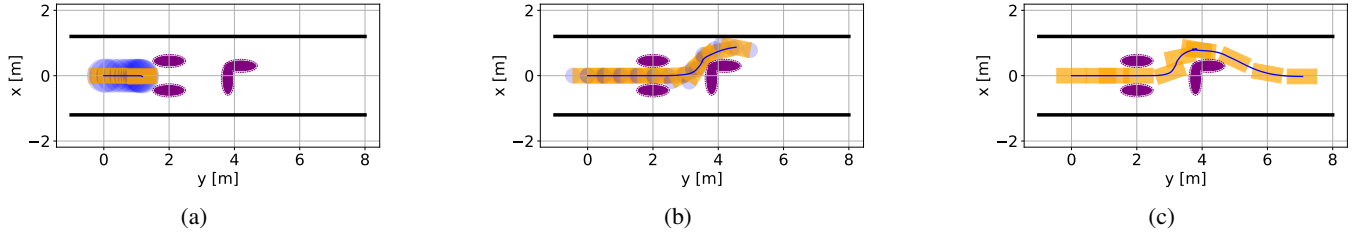


Fig. 3: Comparison between a) MPC under a fixed enclosing circle, b) MPC with a set of circles, c) the proposed MPC-CBF framework. Under narrow circumstances, the fixed inflation radius fails. The set of circles can solve for narrow corridors, but fails when more obstacles require more complex maneuverability. The proposed framework can solve for these cases.

B. Model Predictive Control

For the MPC formulation, the ASV dynamic model is first rearranged by merging (1) and (2), reformulating the dynamic model as (4), where $f(\mathbf{x}) = [\mathbf{0}_{3 \times 3} \ \mathbf{R}(\boldsymbol{\eta}); \mathbf{0}_{3 \times 3} \ -\mathbf{M}^{-1}(\mathbf{C}(\mathbf{v}) + \mathbf{D}(\mathbf{v}))]$, $g(\mathbf{x}) = [\mathbf{0}_{3 \times 4}; \mathbf{M}^{-1}\mathbf{B}]$, the state vector $\mathbf{x}(t) = [x \ y \ \psi \ u \ v \ r]^T \in \mathbb{R}^6$ and the control vector $\mathbf{u}(t) = [f_1 \ f_2 \ f_3 \ f_4]^T$. Rotated ellipse constraints describe the floating obstacles as:

$$\frac{(x'_i)^2}{a_{mi}^2} + \frac{(y'_i)^2}{b_{mi}^2} \leq 1, \quad (8)$$

Here, $x'_i = (x - x_{oi})\cos\theta_{oi} + (y - y_{oi})\sin\theta_{oi}$, $y'_i = (x - x_{oi})\sin\theta_{oi} - (y - y_{oi})\cos\theta_{oi}$, x_{oi}, y_{oi} is the position of the i -th obstacle, and θ_{oi} is its orientation. $a_{mi} = a_{oi} + r_{\min}$ is the length of the inflated semi-major axis, $b_{mi} = b_{oi} + r_{\min}$ is the length of the inflated semi-minor axis, and a_{oi}, b_{oi} are the lengths of the obstacle semi-major and semi-minor axes (the result of the obstacle fitter plus the safety distance d_s), respectively. The constraints for the waterway limits depend on the chosen spatial representation, as describing either a continuous tube as in [19], or rectangular corridors as in [21]. Then, a Nonlinear Program (NLP) is formulated as:

$$\min_{\substack{\mathbf{x}_0, \dots, \mathbf{x}_N, \\ \mathbf{u}_0, \dots, \mathbf{u}_{N-1}}} J(\mathbf{x}, \mathbf{u}) = \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k) + M(\mathbf{x}_N) \quad (9a)$$

$$\text{s.t. } \mathbf{x}_0 = \hat{\mathbf{x}}_0, \quad (9b)$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + g(\mathbf{x}_k)\mathbf{u}_k, k = 0, \dots, N-1, \quad (9c)$$

$$\mathbf{x}_{k,\min} \leq \mathbf{x}_k \leq \mathbf{x}_{k,\max}, k = 0, \dots, N, \quad (9d)$$

$$\mathbf{u}_{k,\min} \leq \mathbf{u}_k \leq \mathbf{u}_{k,\max}, k = 0, \dots, N-1, \quad (9e)$$

$$\text{obstacle } i \text{ constraint (8)}, k = 0, \dots, N, \quad (9f)$$

$$\text{waterway boundary constraints}, k = 0, \dots, N. \quad (9g)$$

Here, $\mathbf{x}_k \in \mathbb{R}^{n_q}$ denotes the vessel state, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ denotes the control input, $\hat{\mathbf{x}}_0 \in \mathbb{R}^{n_q}$ denotes the current state estimate, $L(\mathbf{x}_k, \mathbf{u}_k)$ denotes the Lagrange objective term, and $M(\mathbf{x}_N)$ the Mayer objective term. $L(\mathbf{x}_k, \mathbf{u}_k)$ is designed to take into account the path-following objectives of minimizing the cross-track error and the deviation from the angle path. Additionally, the cost function is extended to follow a desired speed u_{ref} , which can be a waterway limit or an operation preference, and to reduce rotational motion aggressiveness

and control effort. Thus, it is defined as:

$$L(\mathbf{x}_k, \mathbf{u}_k) = Q_y y_{e,k}^2 + Q_r r_k^2 + Q_\psi ((\sin(\psi_k) - \sin(\gamma_{p,k}))^2 + (\cos(\psi_k) \cos(\gamma_{p,k}))^2) + Q_u (u_k - u_{\text{ref}})^2 + \|\mathbf{u}_k\|_{\mathbf{W}}^2, \quad (10)$$

where Q_y, Q_ψ, Q_u, Q_r are penalty weights, and $\mathbf{W}$ is the control input weighting matrix. Similarly, the Mayer objective term is defined as:

$$M(\mathbf{x}_N) = Q_y y_{e,N}^2 + Q_r r_N^2 + Q_\psi ((\sin(\psi_N) - \sin(\gamma_{p,N}))^2 + (\cos(\psi_N) \cos(\gamma_{p,N}))^2) + Q_u (u_N - u_{\text{ref}})^2. \quad (11)$$

The MPC formulation was prototyped in Python using the `Rockit` toolbox [22] and the trajectory optimization solver `FATROP` [23], with its C-code generation capability.

C. Adaptive Obstacle Representation

The proposed MPC formulation provides a solution that does not account for the complete ASV footprint, risking collisions. Nevertheless, a conventional solution would be conservative, inflating the obstacles using the enclosing circle with radius $r_{\max}$ (Fig. 2). Therefore, a methodology is proposed to change the inflation radius depending on the relative position of the obstacle and the current ASV orientation. First, an assumption is established to leverage the overactuation of the vessel:

Assumption 1: As the ASV approaches an obstacle initially, there will be no significant or aggressive rotations.

Under this assumption, the ASV can be treated geometrically as a rectangle with a fixed heading angle. Then, the next step is to compute the closest point between the ASV and the obstacle ellipse, as this would be the first point to create a collision. Finding the closest points between a rectangle and an ellipse is an optimization problem that can be efficiently solved numerically using Newton's method or gradient descent. Once the closest points $[x^*, y^*]^T, [x_o^*, y_o^*]^T$ are found, the relative angle α_o between the ASV and the obstacle closest contact points is computed with $\phi_o = \text{atan2}((y_o^* - y^*), (x_o^* - x^*))$ and $\alpha_o = \psi - \phi_o$. Finally, the projection of the ASV width and length is used to compute the ASV span in the direction toward the obstacle, which divided by two gives the dynamic inflation radius $r_o(\alpha_o)$:

$$r_o(\alpha_o) = (w|\sin\alpha_o| + l|\cos\alpha_o|)/2. \quad (12)$$

Remark 1: The adaptive obstacle representation is designed around Assumption 1, which limits the solution to

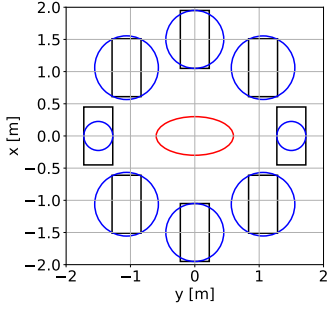


Fig. 4: Examples of the dynamic radius inflation. The ASV footprint is shown in black, the red ellipse represents the obstacle plus a safety distance, and the blue circles represent the considered inflation.

fully-actuated systems (as many inland waterways vessels are [3], [24]). The no-rotation assumption allows the vessel to move longitudinally and laterally to counteract obstacles and guarantee safety. The boat can control its orientation; thus, the assumption can become a system constraint. However, understanding that maneuverability is needed in some tight spaces, the deadlock recovery algorithm in Section III-E is presented to perform safe rotations.

Remark 2: The adaptive inflation (12) establishes that if the ASV is facing directly towards an obstacle, then the distance between the center of the vessel and the obstacle circumference should be at least half the vessel length. If the obstacle is at a 90-degree angle with respect to the boat, then the obstacle should be at a distance of at least half the ASV's width. This equation modifies the constraint dimension such that the ASV can navigate through narrow corridors, in contrast to overly-conservative representations (see Fig. 3). Fig. 4 illustrates the equation.

D. High Order Control Barrier Functions

Since the MPC provides an approximate solution with no collision-free guarantees for the full dimension of the ASV, HOCBFs are proposed to ensure safety. Following the MPC constraint (8), a barrier function $b_{oi}(\mathbf{x})$ is designed for each obstacle i as an ellipse:

$$b_{oi}(\mathbf{x}) = \frac{(x'_i)^2}{a_{bi}^2} + \frac{(y'_i)^2}{b_{bi}^2} - 1, \quad (13)$$

where $a_{mi} = a_{oi} + r_{oi}(\alpha_{oi})$, $b_{mi} = b_{oi} + r_{oi}(\alpha_{oi})$ are the length of the inflated semi-major and semi-minor axes, respectively. In the case of waterway boundaries, similar to the MPC constraints, they can be defined in terms of tubes or corridors. For demonstration purposes, this work considers scenarios that simulate straight, narrow canals. Thus, the system should satisfy the constraints $-(h_c)/2 + r_{ot}(\alpha_{ot}) \leq x + x_c \leq (h_c)/2 - r_{ob}(\alpha_{ob})$, $-(w_c)/2 + r_{ol}(\alpha_{ol}) \leq y + y_c \leq (w_c)/2 - r_{or}(\alpha_{or})$, where (x_c, y_c) is the center of the canal, h_c, w_c are the canal height and width, and $r_{oj}(\alpha_{oj}), r_{oj}(\alpha_{oj}), \forall j = t, b, l, r$ are computed with $\phi_{oj} = \{0, \pi, -\pi/2, \pi/2\}$ rad, respectively.

Then, four barrier functions are defined as:

$$\begin{aligned} b_{bb} &= (h_c)/2 - r_{ob}(\alpha_{ob}) + \Delta_x, & b_{bt} &= (h_c)/2 - r_{ot}(\alpha_{ot}) - \Delta_x, \\ b_{bl} &= (w_c)/2 - r_{ol}(\alpha_{ol}) + \Delta_y, & b_{br} &= (w_c)/2 - r_{or}(\alpha_{or}) - \Delta_y. \end{aligned} \quad (14)$$

where $\Delta_x = x + x_c, \Delta_y = y + y_c$. The relative degrees of the safety constraints (13)-(14) are all 2, i.e., $m = 2$ in the HOCBF (7). The corresponding HOCBF constraints can then be defined as:

$$L_f^m b_{oi}(\mathbf{x}) + L_g L_f^{m-1} b_{oi}(\mathbf{x}) \mathbf{u} + O(b_{oi}(\mathbf{x})) \quad (15a)$$

$$+ \alpha_m(\psi_{bi, m-1}(\mathbf{x})) \geq 0,$$

$$L_f^m b_{bj}(\mathbf{x}) + L_g L_f^{m-1} b_{bj}(\mathbf{x}) \mathbf{u} + O(b_{bj}(\mathbf{x})) \quad (15b)$$

$$+ \alpha_m(\psi_{2, m-1}(\mathbf{x})) \geq 0,$$

where $\psi_{k, m-1}, k \in \{oi, bj\}$ are defined as in (5) for $b_{oi}(\mathbf{x})$ and $b_{bj}(\mathbf{x})$, respectively. Any control input applied to the ASV should satisfy (15). At each time step, the control input solution from the MPC is used as reference $\mathbf{u}_R$, for the following optimization:

$$\min_{\mathbf{u}} \|\mathbf{u} - \mathbf{u}_R\|^2, \quad \text{s.t., (15)}. \quad (16)$$

Hence, the HOCBFs compute a new control input $\mathbf{u}$. The optimization problem (16) is a QP problem, which was implemented in this work in C++ using the software package qpOASES [25].

E. Deadlock Recovery Algorithm

The proposed methodology takes advantage of the capability of ASV holonomic motion. However, not utilizing rotations to avoid obstacles may lead to deadlock situations. Henceforth, a deadlock recovery algorithm is introduced. First, a stuck-detection algorithm is used. This algorithm keeps track of a number of the vessel's previous positions and calculates the average of such positions. If the maximum distance between the average and the rest of the positions is lower than a threshold, then the ASV is assumed to be stuck, and a flag is raised that the vessel is in a narrow passage. This heuristic should be tuned according to the expected motion of the ASV. The deadlock recovery maneuver is activated if the vessel is inside a narrow passage. The maneuver first computes an approximate angle that may liberate the boat. This computation assumes that the ASV can only be stuck in two scenarios: 1) between two obstacles, or 2) between an obstacle and one of the waterway boundaries. Thus, the stuck scenario is assessed using the closest points between the ASV to each obstacle and each waterway boundary. According to the scenario, the two closest points $\mathbf{ps1}, \mathbf{ps2}$ between the closest elliptical obstacle and the second object (second closest obstacle or closest waterway boundary) are computed, i.e., the nearest points between two ellipses or between an ellipse and a straight line. Next, the normal of

the angle between them is computed:

$$\phi_s = \text{atan2}((y_{s1} - y_{s2}), (x_{s1} - x_{s2})), \quad (17a)$$

$$\phi_+ = \phi_s + \pi/2 - \psi, \quad (17b)$$

$$\phi_- = \phi_s - \pi/2 - \psi, \quad (17c)$$

$$\psi_d = \begin{cases} \phi_s + \pi/2, & \text{if } |\phi_+| \geq |\phi_-|, \\ \phi_s - \pi/2, & \text{if } |\phi_+| < |\phi_-|, \end{cases} \quad (17d)$$

and a Control Lyapunov Function (CLF), used in conjunction with CBFs [11] [12], is defined by choosing $V = (r + k_v(\psi - \psi_d))^2$, with parameter $k_v > 0$, resulting in a modification of (16) as:

$$\min_{\mathbf{u}} \|\mathbf{u} - \mathbf{u}_R\|^2 \quad (18a)$$

$$\text{s.t. (15),} \quad (18b)$$

$$L_v V(\mathbf{x}) + L_g V(\mathbf{x})\mathbf{u} + c_3 V(\mathbf{x}) \leq 0. \quad (18c)$$

Finally, by verifying the distance between the vessel and the objects during the recovery procedure, the ASV is monitored to determine if it is still in the narrow passage or if it has managed to navigate it.

Algorithm 1 Proposed safe motion planning algorithm.

- 1: Project LiDAR points into a short-term memory grid.
 - 2: Fit ellipses on clustered points.
 - 3: Solve an MPC problem formulation with the ASV inner radius inflation (9).
 - 4: Compute the non-conservative radius for each obstacle i with (12).
 - 5: Solve the Quadratic Programming (QP) problem (16) with the updated HOCBF constraints, using the MPC solution as reference.
 - 6: Determine if the robot is stuck.
 - 7: **if** Robot is stuck **then**
 - 8: Compute approximate heading for deadlock recovery with (17).
 - 9: Raise the flag that the ASV is in a narrow passage.
 - 10: **end if**
 - 11: **while** ASV in a narrow passage **do**
 - 12: Check if the robot is out of the narrow passage.
 - 13: Perform steps 1-4, and solve the QP problem (18) to rotate the vessel to the computed desired heading safely.
 - 14: **end while**
-

IV. RESULTS

In this section, the simulation and experimental setup are described, and the results are discussed.

A. Setup

1) *Simulation Setup*: The simulation environment uses ROS, Rviz, and Gazebo to simulate LiDAR sensor information and for visualization, whereas the dynamic model (1)-(2) is programmed into a C++ ROS node. A Gazebo world was built with a lake visualization, representing an intersection scenario with canal boundaries. Different environments were designed considering various obstacle types such as boxes,

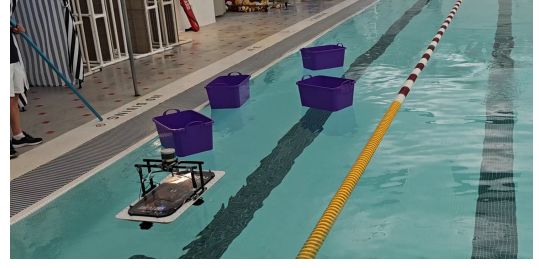


Fig. 5: Experimental setup illustration.

poles, and buoys. The simulation environment was designed to use the same ROS codebase architecture as with the experimental platform. Simulations were carried out using an Intel(R) Core(TM) i7-12800H @ 4.8GHz, running the control loop at a frequency of 10 Hz,

2) *Experimental Setup*: The Quarterscale Roboat [20] was deployed on a swimming pool. A swimming lane with a width of 2.5 m was selected to test, mimicking a narrow straight waterway. Four plastic boxes were placed inside the lane, creating narrow spaces between the boxes, and between boxes and the lane boundaries, as seen in Fig. 5. In addition, the boxes were not anchored, which formed a quasi-static but dynamic environment. Localization is achieved using a LiDAR-Inertial-Odometry algorithm [26].

3) *Obstacle Detection System*: An online obstacle detection system using LiDAR data was implemented to test the motion planning algorithm in simulations and experiments, based on the perception system used in [5]. The LiDAR data is filtered within a desired xyz range, and then projected into a 2D grid. The grid map is then inflated by $r_{\min} + d_s$, creating clusters of what could represent a single obstacle (as no assumptions about the shape or size of the obstacles are made). Next, ellipses are fitted around contours using OpenCV [27]. Finally, the ellipses are deflated by $r_{\min}$, resulting in what is assumed to be each ellipse that contains a detected obstacle plus a safety distance.

B. Simulation Results

Through all simulation runs, the vessel was tasked to follow a curve starting at coordinates (0,0) and ending at (6,6). In all environments, obstacles are positioned to create narrow spaces between them and between obstacles and the canal boundaries. Fig. 6 shows the results on three different obstacle configurations. Here, the ASV is represented as the orange rectangle, the LiDAR point cloud is shown in purple, and the trajectory is given in blue. For all scenarios, the ASV maneuvered the obstacle courses without collisions. Table I includes computational performance data.

C. Experimental Results

Physical experiments were conducted in a swimming pool lane. The Roboat was tasked to follow a straight-line path through the center of the lane. Four plastic boxes were randomly positioned at each experimental run to obstruct the path. As the boxes were not anchored, the experiments showcase the performance of the proposed approach in

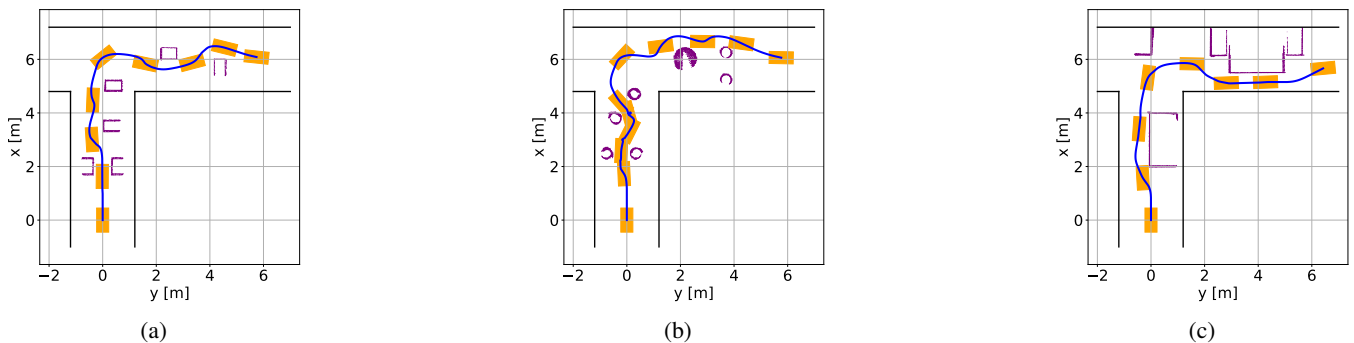


Fig. 6: Simulation results.

Computation	Fixed MPC	Multi. Circles	MPC-CBF	HOCBF QP	Closest points	Closest points (per obs)
Median (ms)	2.281	8.284	18.187	0.054	15.852	3.963
Max (ms)	26.601	66.794	39.008	0.18	12.227 (max case)	3.056 (max case)

TABLE I: Solution times for different computations on simulation, including: MPC with a fixed enclosing circle; MPC with a set of circles; full MPC-CBF framework; adaptive HOCBF QP; computation of the closest points between the ASV and all obstacles; average computation of the closest points between the ASV and one obstacle.

dynamic environments. The experiment was repeated 10 times. Fig. 7 depicts three different experimental runs. In addition to the ASV, its trajectory, and the LiDAR point cloud data, gray ellipses show the obstacle detection result at the given time step.

V. DISCUSSION

The simulation results show the diverse sets of obstacles and scenarios that the proposed method can resolve without collision. In Fig. 6a-b, the closest obstacles to the origin act as a narrow gate, where the vessel can only navigate through by driving straight. In Fig. 6c, the large boxes simulate docks, containers, or docked vessels, which may obstruct large canal parts. Observing the solution times in Table I, the proposed approach implementation is capable of meeting real-time requirements (10-20Hz). However, the closest-points computation is the most expensive step, which could be implemented in parallel and further decrease the computation times. In this sense, solving the fixed-circle MPC and the QP together takes on average 2.281 ms. Moreover, the MPC with a set of circles can require longer computations, while still failing to solve for different tight scenarios, as in Fig. 3. Furthermore, although the method was designed for static obstacles, the unstructured nature of aquatic environments extends the results to quasi-static obstacles. The obstacles in the pool were slowly moving (speeds below 0.5 m/s). Still, the active perception system and the reactive nature of the proposed adaptive HOCBFs allowed the system to navigate without collisions. We report a 100% success rate out of 10 physical experiments. However, it should be clarified that there were instances in which the ASV got stuck for a few seconds due to insufficient space to pass through. However, the natural motion of the obstacles eventually created feasible gaps for the vessel to navigate in between. As seen in Fig. 7, the ASV consistently drove through narrow spaces between obstacles or between an obstacle and a lane. The accompa-

nying video shows numerous successful experimental runs. Finally, although the method reduces conservativeness in contrast to representations with a singular large circle or a set of circles, there is still reliance on ellipsoidal shapes to represent obstacles. Hence, the practical performance of the proposed strategy depends on the obstacle detection system. If the obstacle ellipsoid covers much free space, this method will not guarantee solving for tight spaces. Nonetheless, if the perception system can provide solutions tailored to avoid the loss of free space, then the method has a higher chance of success. For instance, the method performs better if long rectangles are not treated as a single ellipsoid, but if each line within the field of view is represented as an independent ellipsoid.

VI. CONCLUSION

In this paper, we propose a safe motion planning strategy for fully actuated ASVs that integrates Model Predictive Control (MPC) with Control Barrier Functions (CBFs). Our approach introduces an adaptive obstacle-representation methodology that retains the efficiency of ellipsoidal representations while reducing the conservativeness of traditional formulations. The proposed strategy was validated through both real-time simulations and real-world experiments. The results demonstrate that our method effectively navigates ASVs through narrow spaces and resolves deadlock situations in real-time, ensuring safe and reliable operation.

REFERENCES

- [1] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles i: a review," *Journal of Marine Science and Technology*, 2021.
- [2] C. Domenighini, "Autonomous inland navigation: a literature review and extracontractual liability issues," *Journal of Shipping and Trade*, 2024.

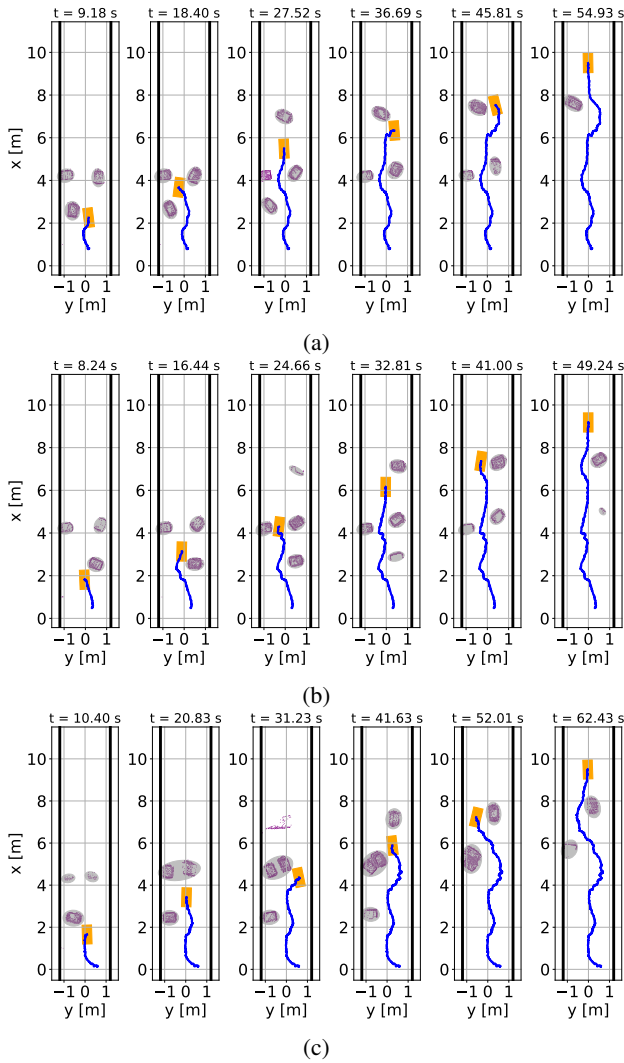


Fig. 7: Experimental results.

[3] W. Wang, D. Fernández-Gutiérrez, R. Doornbusch, J. Jordan, T. Shan, P. Leoni, N. Hagemann, J. K. Schiphorst, F. Duarte, C. Ratti, and D. Rus, "Roboat III: An autonomous surface vessel for urban transportation," *Journal of Field Robotics*, vol. 40, no. 8, pp. 1996–2009, 2023.

[4] Y. Cheng, M. Jiang, J. Zhu, and Y. Liu, "Are we ready for unmanned surface vehicles in inland waterways? the usv inland multisensor dataset and benchmark," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3964–3970, 2021.

[5] T. Shan, W. Wang, B. Englot, C. Ratti, and D. Rus, "A receding horizon multi-objective planner for autonomous surface vehicles in urban waterways," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 4085–4092.

[6] L. Streichenberg, E. Trevisan, J. J. Chung, R. Siegwart, and J. Alonso-Mora, "Multi-agent path integral control for interaction-aware motion planning in urban canals," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1379–1385.

[7] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3340–3356, 2022.

[8] M. Muehlebach and R. D'Andrea, "A method for reducing the complexity of model predictive control in robotics applications," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2516–2523, 2019.

[9] J. Ubbink, R. Viljoen, E. Aertbeliën, W. Decré, and J. De Schutter, "From instantaneous to predictive control: A more intuitive and tunable mpc formulation for robot manipulators," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 748–755, 2025.

[10] A. Gonzalez-Garcia, I. Collado-Gonzalez, R. Cuan-Urquiza, C. Sotelo, D. Sotelo, and H. Castañeda, "Path-following and lidar-based obstacle avoidance via nmpe for an autonomous surface vehicle," *Ocean Engineering*, vol. 266, p. 112900, 2022.

[11] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. of 53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.

[12] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2021.

[13] H. Darweesh, E. Takeuchi, K. Takeda, Y. Ninomiya, A. Sujiwo, L. Y. Morales, N. Akai, T. Tomizawa, and S. Kato, "Open source integrated planner for autonomous navigation in highly dynamic environments," *Journal of Robotics and Mechatronics*, vol. 29, no. 4, pp. 668–684, 2017.

[14] Y. Huang, L. Chen, and P. Gelder, "Generalized velocity obstacle algorithm for preventing ship collisions at sea," *Ocean Engineering*, vol. 173, pp. 142–156, 2019.

[15] B. Shi, Y. Su, C. Wang, L. Wan, and Y. Luo, "Study on intelligent collision avoidance and recovery path planning system for the waterjet-propelled unmanned surface vehicle," *Ocean Engineering*, vol. 182, pp. 489–498, 2019.

[16] M. S. Wiig, K. Y. Pettersen, and T. R. Krogstad, "Collision avoidance for underactuated marine vehicles using the constant avoidance angle algorithm," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 951–966, 2020.

[17] J. De Vries, E. Trevisan, J. Van Der Toorn, T. Das, B. Brito, and J. Alonso-Mora, "Regulations Aware Motion Planning for Autonomous Surface Vessels in Urban Canals," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 2022, pp. 3291–3297.

[18] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 286–292.

[19] W. Wang, W. Xiao, A. Gonzalez-Garcia, J. Swevers, C. Ratti, and D. Rus, "Robust model predictive control with control barrier functions for autonomous surface vessels," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 6089–6095.

[20] W. Wang, L. 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 *Proc. 2018 IEEE Int. Conf. Robot. Autom.*, 2018, pp. 6189–6196.

[21] M. Bos, B. Vandewal, W. Decré, and J. Swevers, "Mpc-based motion planning for autonomous truck-trailer maneuvering," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 4877–4882, 2023, 22nd IFAC World Congress.

[22] J. Gillis, B. Vandewal, G. Pipeleers, and J. Swevers, "Effortless modeling of optimal control problems with rocket," in *39th Benelux Meeting on Systems and Control*, vol. 138. Elspeet, The Netherlands, 2020.

[23] L. Vanroye, A. Sathya, J. De Schutter, and W. Decré, "Fatrop: A fast constrained optimal control problem solver for robot trajectory optimization and control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 036–10 043.

[24] Y.-Y. Zhang, J. Billet, and P. Slaets, "Experimental identification of decoupled ship dynamic models for an autonomous catamaran urban cargo vessel," *IFAC-PapersOnLine*, vol. 58, no. 20, pp. 229–234, 2024.

[25] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[26] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

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