Relaxed Control Barrier Function Based Control for Closest Approach by Underactuated USVs

Karl D. von Ellenrieder, *Senior Member, IEEE*, and Marco Camurri, *Member, IEEE*

***Abstract*—The use of automatic safety-critical control for uncrewed surface vessel (USV) survey, inspection and intervention can provide a computationally lightweight controller which guarantees that a minimum safe standoff distance to a target of interest is always maintained. We propose a trajectory tracking safety-critical controller for the closest safe approach of an underactuated USV with nonholonomic dynamic (acceleration) motion constraints to a target. A backstepping-based control law is designed using a relaxed control barrier function and an analytical convex optimization method. The stability of the controller is proven. Simulations of a USV approaching both stationary and moving targets are used to demonstrate implementation of the method. The performance of the proposed controller is compared with that of a nonlinear model predictive control (MPC) controller in simulation. The simulation results demonstrate that, while the tracking error of the proposed controller is higher than that of an MPC controller, it requires lower computational resources, suggesting it is a good candidate for use on small USVs with low computational power.**

***Index Terms*—Relaxed control barrier functions (CBFs), trajectory tracking, uncrewed surface vessels (USVs), underactuated.**

I. Introduction

MANY survey [1], [2], [3], monitoring [4], [5], [6], and interception [7], [8] applications require an uncrewed surface vessel (USV) to approach a target (stationary or moving) as closely as possible, while still guaranteeing that some minimum safe standoff distance is respected. Safety-critical control using control barrier functions (CBFs) is a relatively recent approach that permits an automatically controlled system to fully utilize the safe operating conditions available, while guaranteeing the stable avoidance of unsafe conditions [9], [10]. When desired, the use of a CBF-based safety-critical controller can permit a system to operate at the boundary between safe and unsafe, ensuring that the system remains on the safe-side of that boundary. For applications that require a USV to perform the closest possible safe approach to an object of interest, CBF-based safety-critical control techniques can be very useful.

A significant advantage of the use of CBFs is that they permit the safety of a system to be incorporated in a flexible, modular, way because the safety objective can be solved separately from the control objective, while at the same time they are reactive, so that knowledge of the system's high-level (deliberative) motion plan is not required. Several recent works have explored the use of CBF-based safety-critical control in the context of USV collision avoidance. For example, [11] proposed the use of a CBF for the International Regulations for the Prevention of Collisions at Sea (COLREG)s-compliant trajectory tracking of fully-actuated marine surface vessels. In [12], the use of a "synergistic CBF" (which uses hybrid control) for underactuated nonholonomic vessels with kinematic (velocity) motion constraints was proposed. The direction in which the obstacle is avoided is biased to overcome the problem of vanishing control inputs during head-on collisions.The author in [13] proposed an optimization-based reactive collision avoidance system that used CBFs integrated into the control allocation system to ensure that the actuator magnitude and rate limits are respected. In [14], control Lyapunov functions (CLFs), CBFs, and quadratic programming were employed for the maneuvering and dynamic positioning of fully-actuated USVs. Integral control is used to counteract disturbances (e.g., from currents) and the effects of control allocation and actuator limitations are included as part of a CLF-CBF based convex optimization problem. A *relaxation term* is included in the CBF so that the safety and stabilization objectives can be separately satisfied.

While all of the works cited above have made significant advances in the use of CBFs for the safety-critical control of USVs, one might ask: "Is collision avoidance is the best application for safety-critical control?" When a planned trajectory intersects an unsafe operating region, a CBF-based safety-critical collision avoidance control system will rapidly transition from the original trajectory to steer the USV along the outside of the boundary of the unsafe region until it is possible to rejoin the original trajectory. The maneuvers generated when using CBFs generally tend to bring the USV as close as possible to an obstacle. Typically, CBFs are used to maximize the safe operating space of a robot in a confined environment by permitting the robot to operate as closely as possible to the boundary between the safe and the unsafe spaces. However, obstacle avoidance does not generally require a robot to move as closely as possible to an obstacle, and sometimes it is likely disadvantageous to do so. For example, having a USV traverse a straight line parallel to the desired trajectory, but shifted away from the unsafe region, would consume less energy and be less dangerous than forcing it to follow a trajectory with a complex shape in order to move along the very edge of the safe region. For these reasons, we

Manuscript received 30 June 2022; revised 1 February 2024; accepted 15 June 2024. *(Corresponding author: Karl von Ellenrieder.)*

Associate Editor: A. Munafo.

The authors are with the Facoltà di Ingegneria, Libera Università di Bolzano, 39100 Bolzano, Italy (e-mail: karl.vonellenrieder@ieee.org; marco.camurri@unibz.it).

Digital Object Identifier 10.1109/JOE.2024.3423869

suggest that the use of safety-critical control (including the methods proposed in [11], [12], [13], and [14]) may be more well-suited for applications requiring the closest safe approach of a USV to an object of interest, such as survey, monitoring, and interception. In such applications one would plan a trajectory that passes through, or very close to the target, intentionally activating the CBF to get as close as possible to the target, while still maintaining a safe stand off distance. Here, we focus on the close approach to a target by a USV using relaxed CBF-based safety critical control.

The use of CBFs for safety critical control generally involves solving an optimization problem to minimize the difference between the control input required in the absence of an unsafe condition and the control effort generated by a barrier function, the magnitude of which rapidly increases as the system approaches the unsafe condition. This optimization problem is usually solved numerically and must be done online and in realtime using the approaches proposed in [11], [12], [13], and [14]. In contrast, the safety-critical control approaches proposed in the series of interrelated papers [15], [16], [17], [18] employ relaxed CBFs (rCBFs) and an analytical solution for the optimization problem described above. The main advantage of the approach taken in these works is that the analytical solution is fairly easy to implement and avoids the need to perform online optimization. In [15], an rCBF, based on the CBF presented in [19], was proposed to provide a collision avoidance control assistance input for a human-controlled two-wheeled mobile robot. The optimization problem is solved analytically when there are multiple stationary obstacles in the operating space of the robot. In a followup study, [16] extended the previous work by introducing a time varying rCBF. To prove the stability of the system, they introduce the notion of a graph space to contend with the time dependent nature of the unsafe region around each obstacle. As a motivating example, they explore the control of a wheel chair with a human-assist control input for collision avoidance. Since the human-assist control input depends on the velocity of the moving obstacles and small speed errors could lead to a collision when the system moves along the boundary of the safe region, [17] build further on the approach in the latter two works by introducing a high-gain observer to estimate the velocities of obstacles. They also show that the rCBF is input-to-state safe when obstacle velocity measurement errors are treated as disturbances. Lastly, [18] modified the approach by replacing the human input signal with an automatic trajectory tracking control input and explore the safety-critical collision avoidance control of a two-wheeled mobile robot in the presence of stationary obstacles. The surge speed control input is taken to always be the same as the trajectory tracking surge speed control input and collision avoidance is achieved by only adding the collision avoidance control input to the steering command. In the representative simulations presented, the robot is able to safely move along the boundary of the safe region, but the control input exhibits a chattering-like behavior.

Here, we build upon the results of [15], [16], [17], and [18], which are formulated for nonholonomic vehicles with kinematic (velocity) motion constraints, by extending the methods to the safety critical control of USVs. The main novel contributions of this article are: 1) extension of the approach to underactuated USVs with nonholonomic dynamic (acceleration) motion constraints; and 2) a modification to the rCBF proposed in [15], which permits the safety critical control input to start acting sooner and more gradually as the USV approaches the boundary of the unsafe region.

The rest of this article is organized as follows. An explanation of how the safe set of states can be defined for trajectory tracking in the presence of moving targets, a mathematical definition of the rCBFs, and the analytical solution for the optimal safety critical control input are provided in Section II. In Section III the equations of motion of the underactuated USV are presented and the control problem is formally stated. The controller is designed in Section IV, illustrative simulations of the controller applied to a USV operating in the presence of both stationary and moving targets are presented in Section V. Finally, Section VII concludes this article.

II. Mathematical Preliminaries

In general, CBFs can be defined for nonlinear systems, which are affine in the control input, of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}_c \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state, the vector functions $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m$ are locally Lipschitz, and $\boldsymbol{u}_c \in \mathbb{R}^m$ is the control input. Here, we specifically consider the use of CBFs for the situation in which $\boldsymbol{x} \in \mathbb{R}^2$ represents the position of a USV and $\boldsymbol{u}_c \in \mathbb{R}^2$ is a set of virtual control inputs.

Take the position of a moving target $\boldsymbol{x}_0(t)$ to be a continuous mapping $\boldsymbol{x}_0 : \mathbb{R} \to \mathbb{R}^2$ and assume that the velocity of the target is smooth and continuous, i.e., $\dot{\boldsymbol{x}}_0 \in C^1$. To ensure the safety of the USV, let the minimum safe approach distance be a circle of radius $\|\tilde{\boldsymbol{x}}_0\|_{\min}$, where $\|\cdot\|$ represents the two-norm of a vector. In safety critical control design, the strategy is to find a control input $\boldsymbol{u}_c$ that renders a given set of safe system states (termed the *safe set*) forward invariant along the solution trajectories of (1). Here, the safe set consists of the points outside of this circle. However, since the target is moving, the points in the safe region around the target change in time as the target moves through the operating space of the system and defining them as a fixed set would be problematic [16]. Thus, for a moving target, the safe region is time dependent and so cannot be formally considered to be a *set*. To circumvent this problem [16] propose the notion of a graph space for safety-critical control in the presence of moving targets, where the graph space is defined as $\mathcal{G} \subset D \times \mathbb{R}$, such that $(\boldsymbol{x}(t), t) \in \mathcal{G}$ for any $x(t) \in \mathcal{C}(t)$. With this definition $\mathcal{G}$ is a fixed subset of $\mathbb{R}^{n+1}$. Here, we instead formulate the safe region in terms of the vector distance between the moving target and the controlled system $\tilde{\boldsymbol{x}}_0 := \boldsymbol{x} - \boldsymbol{x}_0$. With this formulation the notion of a safe set $\tilde{X}_0 \subset \mathbb{R}^2$ and an unsafe set $\tilde{X}_{0u} \subset \mathbb{R}^2 \backslash \tilde{X}_0$ apply, as they are fixed in time with respect to the moving target. An important consequence of this is that a moving CBF defined using $\tilde{X}_0$ is an *implicit*, rather than an *explicit*, function of time.

A relaxed CBF will be used to design the safety-critical controller. We modify the approach proposed in [16] so that the barrier function and resulting virtual control input are based on the vector distance from the USV to the target $\tilde{\boldsymbol{x}}_0$ and on the trajectory tracking error $\tilde{\boldsymbol{x}} := \boldsymbol{x} - \boldsymbol{x}_d$, where $\boldsymbol{x}_d$ is the desired position to be tracked. The objective is to design the virtual control input $\boldsymbol{u}_c$ so that $\tilde{\boldsymbol{x}}_0(t) \in \tilde{X}_0$ for all $\tilde{\boldsymbol{x}}_0(0) \in \tilde{X}_0$ and all $t \geq 0$ when system (1) is operated in the presence of a moving target.

A. Relaxed Control Barrier Functions

Definition 1 (Relaxed CBF): Consider the system (1) and the safe set $\tilde{X}_0$. A class C^1 function $B : D \times D \to \mathbb{R}$, where $D \subset \mathbb{R}^n$, is said to be a relaxed control barrier function if the following three conditions hold:

1) The function $B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) \geq 0$ for all $\tilde{\boldsymbol{x}}_0 \in \tilde{X}_0$.
2) The function B is proper, i.e., $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0 \mid B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) \leq L\}$ is compact for any $L \geq 0$.
3) For any continuous $\boldsymbol{u}_c \in \mathbb{R}^m$, there exist nonnegative constants $\alpha, \beta \geq 0$, so that

$$\inf_{\boldsymbol{u}_c \in \mathbb{R}^m} \dot{B} = \inf_{\boldsymbol{u}_c \in \mathbb{R}^m} \left[\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\tilde{\boldsymbol{x}}}_0 \right] < \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) + \beta. \tag{2}$$

Assumption 1: There exist a safe set $\tilde{X}_0 \subset D$, a CBF $B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0)$ for $\tilde{X}_0$, and a stable trajectory tracking control input $\boldsymbol{u}_t$. The initial position of the system is inside the safe set, $\tilde{\boldsymbol{x}}_0(0) \in \tilde{X}_0$.

B. Trajectory Tracking Safety-Critical Control

The trajectory tracking safety-critical control input is designed by solving the optimization problem

$$\min_{\boldsymbol{u}_c} \|\boldsymbol{u}_c - \boldsymbol{u}_t\|^2, \tag{3a}$$

$$\text{s.t. } \dot{B} - \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) - \beta < 0. \tag{3b}$$

Theorem 1: Consider system (1) under Assumption 1. Let

$$\begin{aligned} I &:= \left(\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot [\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{u}_t] - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\boldsymbol{x}}_d - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\boldsymbol{x}}_0 \\ &= L_f B + L_g B \cdot \boldsymbol{u}_t - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\boldsymbol{x}}_d - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\boldsymbol{x}}_0 \end{aligned} \tag{4}$$

where the terms

$$L_f B := \left(\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot \boldsymbol{f}(\boldsymbol{x}) \tag{5}$$

and

$$L_g B := \left(\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot \boldsymbol{g}(\boldsymbol{x}) \tag{6}$$

are Lie derivatives, and let

$$J := \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) + \beta. \tag{7}$$

Then, the control input

$$\boldsymbol{u}_c = \begin{cases} \boldsymbol{u}_t, & I \leq J \\ \boldsymbol{u}_t - (I - J) \dfrac{L_g B^T}{\|L_g B\|^2}, & I > J \end{cases} \tag{8}$$

is the solution of the optimal control Problem (3).

Proof 1: First, note that using (1) and (4) the time derivative of $B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0)$ can be written as

$$\begin{aligned} \dot{B} &= \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\tilde{\boldsymbol{x}}}_0 \\ &= \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d) + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0) \\ &= \left(\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot \dot{\boldsymbol{x}} - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\boldsymbol{x}}_d - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\boldsymbol{x}}_0 \\ &= \left(\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot [\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{u}_c] - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\boldsymbol{x}}_d - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\boldsymbol{x}}_0 \\ &= L_f B + L_g B \cdot \boldsymbol{u}_c - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\boldsymbol{x}}_d - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\boldsymbol{x}}_0 \\ &= L_f B + L_g B \cdot \boldsymbol{u}_t - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}} \cdot \dot{\boldsymbol{x}}_d - \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \cdot \dot{\boldsymbol{x}}_0 + L_g B \cdot (\boldsymbol{u}_c - \boldsymbol{u}_t) \\ &= I + L_g B \cdot (\boldsymbol{u}_c - \boldsymbol{u}_t). \end{aligned} \tag{9}$$

Then, define the Lagrangian function of Problem (3) as

$$L(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0, \boldsymbol{u}_c, \lambda) := \|\boldsymbol{u}_c - \boldsymbol{u}_t\|^2 + \lambda G(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0, \boldsymbol{u}_c) \tag{10}$$

where

$$G(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0, \boldsymbol{u}_c) := \dot{B} - \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) - \beta \tag{11}$$

and λ is a Lagrange multiplier.

The corresponding Karush–Kuhn–Tucker (KKT) conditions are given by

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{u}_c} = 2(\boldsymbol{u}_c - \boldsymbol{u}_t)^T + \lambda L_g B &= 0 \\ \lambda G(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0, \boldsymbol{u}_c) &= 0. \end{aligned} \tag{12}$$

Consider the following two cases:

1) $\lambda = 0$: In this case the KKT conditions can be satisfied by taking $\boldsymbol{u}_c = \boldsymbol{u}_t$. If this value of $\boldsymbol{u}_c$ is chosen when $I \leq J$, it can be seen from (7) and (9) that the inequality constraint (3b) is satisfied.

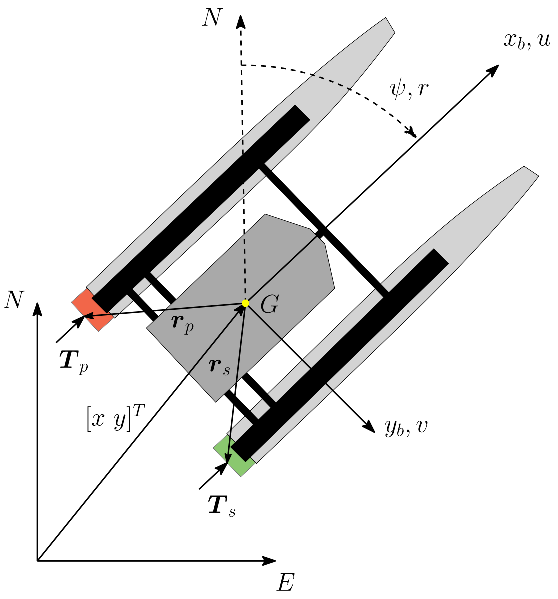


Fig. 1. Three-DOF maneuvering coordinate system definitions.

2) $G(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0, \boldsymbol{u}_c) = 0$: In this case, using (9) in (11), together with (7), gives

$$\begin{aligned} G(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0, \boldsymbol{u}_c) &= \dot{B} - \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) - \beta \\ &= I - J + L_g B \cdot (\boldsymbol{u}_c - \boldsymbol{u}_t) \\ &= 0. \end{aligned} \tag{13}$$

Then, rearranging and solving for $\boldsymbol{u}_c$ gives

$$\boldsymbol{u}_c = \boldsymbol{u}_t - (I - J)\frac{L_g B^T}{\|L_g B\|^2} \tag{14}$$

completing the proof. □

Remark 1: The control input given by (8) for $I > J$ would become unbounded if $\|L_g B\|^2 = 0$. As noted in [16], it is not possible for $\|L_g B\|^2 = 0$ when $I > J$. This can be seen by inspecting (7) and (9). If $\|L_g B\|^2 = 0$, the solution of Problem (3) with the constraint (3b) would give a control input such that $I < J$, i.e., the solution would contradict the condition $I > J$.

III. Problem Formulation

Assume that the origin of the body-fixed coordinate system is located at the center of gravity G of the USV (Fig. 1) and that the vehicle is moving through still water. In a 3-D configuration space, the kinematic and dynamic equations of motion of the USV can be written in a combined vector form as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u\cos\psi - v\sin\psi \\ u\sin\psi + v\cos\psi \\ r \\ f_x \\ f_y \\ f_\psi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{1}{m_{11}} & 0 \\ 0 & \dfrac{a_y}{m_{33}} \\ 0 & \dfrac{a_\psi}{m_{33}} \end{bmatrix} \begin{bmatrix} \tau_x \\ \tau_\psi \end{bmatrix} \tag{15}$$

where ψ is the heading angle of the vehicle, x is the position northward, y is the position eastward, u is the surge speed, v is the sway speed, and r is the yaw rate [20].

Assume that the propellers can only generate a control force along the surge direction and a moment about the yaw axis (using differential thrust). Let τ_x be the total thrust force in the surge direction and τ_ψ be the thruster-produced moment about the yaw axis. The inertial and added mass terms are m_{11}, the USV mass and added mass in the surge direction, m_{22}, the USV mass and added mass in the sway direction, m_{33}, the mass moment of inertia and added mass moment of inertia about the yaw axis, m_{23}, the added mass in the sway direction due to acceleration about the yaw axis, and m_{32}, the added mass moment of inertia about the yaw axis due to acceleration along the sway axis. From (15), it can be seen that the terms

$$a_\psi := \frac{m_{22}m_{33}}{m_{22}m_{33} - m_{23}m_{32}} \quad \text{and} \quad a_y := -\frac{m_{23}}{m_{22}}a_\psi \tag{16}$$

interconnect the motion along the sway and yaw axes of the USV. These terms arise when $m_{23}, m_{32} \neq 0$. The hydrodynamically generated forces, which emanate from the effects of centripetal acceleration, Coriolis acceleration, and the hydrodynamic drag

$$\boldsymbol{d} := [d_x \;\; d_y \;\; d_\psi]^T \tag{17}$$

are given by

$$f_x = \frac{1}{m_{11}}\left[m_{22}vr + \left(\frac{m_{23} + m_{32}}{2}\right)r^2 - d_x\right] \tag{18}$$

$$f_y = a_\psi\left[f'_y - \frac{m_{23}}{m_{22}}f'_\psi\right] \tag{19}$$

and

$$f_\psi = a_\psi\left[f'_\psi - \frac{m_{32}}{m_{33}}f'_y\right] \tag{20}$$

where

$$f'_y = -\frac{1}{m_{22}}\left[m_{11}ur + d_y\right] \tag{21}$$

and

$$f'_\psi = -\frac{1}{m_{33}}\left[(m_{22} - m_{11})uv + \left(\frac{m_{23} + m_{32}}{2}\right)ur + d_\psi\right]. \tag{22}$$

The control objective is to simultaneously achieve stable closed loop trajectory tracking with safety-critical control, where the controlled output of the system is the position of the USV in the NED frame, $\boldsymbol{x} = [x \;\; y]^T$.

Let the pose and body-fixed velocity of the USV be given by $\boldsymbol{\eta} := [x \;\; y \;\; \psi]^T$ and $\boldsymbol{v} := [u \;\; v \;\; r]^T$, respectively.

Assumption 2: The full state of the USV, $\boldsymbol{\eta}$ and $\boldsymbol{v}$, is available for feedback control.

On the water, USV states are often measured using a combination of several sensors including, GPS/GNSS, magnetometers, inertial measurement units, LiDAR, and RADAR. Measurement data must be typically fused and refined using state estimation techniques. For example, the sensor suite developed by [21] includes LiDAR, radar, both stereo and 360° wide field cameras, magnetometers, and a global positioning system (GPS). The authors test both X-band radar and high definition Ka-band radar and find that Ka-band systems work better on moving platforms. LiDAR and the camera systems are used to detect objects at close range (within 100 m), as radar does not work well at short distances and because radar systems tend to have narrow vertical fields of view ±10°. In [22], automatically-generated behaviors resulting from the interaction of multiple USVs, and a manned vessel with USVs, are experimentally tested. On each USV and on the manned vessel, the data measured from a GPS, magnetometer, and IMU are fused to determine the USV state, i.e., position, heading, and speed, using a Kalman filter-based state estimator. The state information from each vehicle is broadcast over a wireless network so that the vehicles are mutually aware of each other's positions. A simultaneous localization and mapping system, which fuses data from a 3-D LiDAR, an IMU, and a GPS was presented in [23]. The system incorporates the measurements from the various sensors using a factor graph model and estimates the vehicle state using a Maximum *a posteriori* approach. Lastly, a vision system was proposed for the precise positioning of USVs during docking operations in [24]. The system uses a monocular camera with a special lens and a set of lights mounted on the docking station to measure the azimuth angle, heading angle, and relative distance between the docking station and the USV. Centimeter-level homing accuracy and a 0.1° angle measurement error are reported.

Assumption 3: The desired trajectory $\boldsymbol{x}_d(t)$ is of differentiability class C^2, i.e., $\boldsymbol{x}_d(t)$, $\dot{\boldsymbol{x}}_d(t)$, and $\ddot{\boldsymbol{x}}_d(t)$ are smooth and bounded.

A variety of trajectory planning approaches that take safety or energy usage into account can be used to design $\boldsymbol{x}_d(t)$. For example, in [25] a lattice-based 5-D trajectory planner was proposed to permit USVs to avoid collisions with other marine vessels using maneuvers that adhere to the COLREGs. The planner estimates the risk of a collision and reasons about contingency maneuvers to counteract possible unpredictable behavior from other vessels, while planning a dynamically feasible trajectory. The computational efficiency of the planner is enhanced by accounting for the distribution and concentration of other marine vessels to dynamically scale the control action primitives. In [26], a wave-aware trajectory planner was proposed to permit a USV to avoid collisions with other marine vessels while opportunistically traversing the wavefields that they generate, with a focus on the minimum time traversal of a marine workspace. The planner performs a search across a 4-D pose-time lattice to generate a collision-free, minimum risk trajectory. A USV path planning algorithm, which takes into account the changes in marine environments resulting from tides, weather, and other environmental restrictions was proposed in [27]. Techniques are proposed for speeding up the A* search on the nodes of visibility graphs so that a marine vehicle can dynamically generate and update the traversal map within a given marine area. A multiobjective optimization method was proposed for real-time USV path planning in [28]. The requirements of collision avoidance are formulated as a set of constraints, taking the COLREGs into account. A hierarchical sorting rule is used to prioritize the optimization objectives so that course/speed changes are preferred over other objectives, such as path length or smoothness. In [29], a two-stage trajectory planning scheme was proposed for minimum-time ship maneuvers in close-range encounters with other surface vessels. The first stage utilizes a directed graph generated taking ocean currents into account. The reachability, cost, and risk of collision are explored on the graph using a wavefront search to find a discrete solution. In the second stage, Bezier curves are used to generate a near-optimal, smooth path using the discrete solution from the first stage. COLREGs compliant collision avoidance for waterjet-propelled USVs was examined in [30]. A hybrid A* planner with motion primitive constraints is used to generate an initial reference path. Obstacle avoidance is achieved using a local threat map based on Apollonius curves and the recovery path planned using the Reeds-Shepp curve. Path planning for USV formations was investigated in [31], where an optimal global path is first planned using a heuristic A* algorithm and an artificial potential field method is then locally applied to plan a dynamically feasible path which avoids obstacles. The Predicted Trajectory Approach for USV global motion planning was proposed in [32]. The method uses a combination of A* and Theta* planners to find a global trajectory, taking USV motion constraints into account. In [33], the Minimum Course Alteration (MCA) Method was proposed for simultaneous collision avoidance with multiple surface vessels and fixed obstacles. The approach uses an A* planner to plan a global route, which is optimized by removing unnecessary waypoints. MCA is applied locally to generate a COLREGs-compliant collision-free path. For applications involving the closest approach of a USV, the accurate detection, classification, and tracking of a waterborne target is needed.

Assumption 4: The target's position $\boldsymbol{x}_0$ and velocity $\dot{\boldsymbol{x}}_0$ are known.

In real-world applications, the relative position and velocity of the target must be determined in real-time using onboard sensors. USV object detection and tracking sensor suites often include RADAR, LiDAR, sonar, cameras, and automatic information systems (AISs). For example, in [34], trajectory mapping and clustering methods were combined to extract information with high accuracy and low computational costs from AIS data. Merge distance is used to measure the similarities between different trajectories and multidimensional scaling is adopted to construct a low-dimensional spatial expression of the similarities between trajectories. An algorithm is proposed to improve the density-based spatial clustering of trajectories with noise. A stereovision-based methodology for tracking the position and orientation of marine surface vessels from a USV was proposed in [35]. The approach combines a stereovision matching algorithm, an Extended Kalman Filter-based predictive-corrective method, and probabilistic models of the motion of the target vessel and stereovision measurements. Both physical, onwater experiments and simulations are used to demonstrate the efficacy of the proposed method. In [36], a method was proposed for detecting and tracking waterborne obstacles from a USV for short-term obstacle avoidance. The data from a stereo camera and IMU are used to generate a point cloud and fit a plane representing the free surface. Outlying points are further processed to detect obstacles, which are tracked using a histogram-like depth model. A real-time, vision-based, system for detecting high-speed USVs using deep learning techniques was proposed in [37]. The system is shown to identify multiple types and sizes of USVs on the water. To detect and track surface vessels from a moving ship [38] propose the use of a convolutional neural network-based data association method for fusing camera and radar measurements. The method does not require a highly accurate alignment and calibration of the camera and radar systems. In [39], Demperster–Shafer Evidence Theory was applied to detect obstacles by fusing data from a LiDAR, radar, and stereovision camera. Field trials demonstrate that the system detects USVs better using the fused data than when using the data from individual sensors.

The control design proceeds in two stages and is based on the use of backstepping. In the first stage, a safety-critical controller is designed for trajectory tracking at the kinematic level. In the second stage, we determine the physical control inputs required to generate the virtual control inputs, while also respecting the USV's dynamic (acceleration) constraints.

IV. Control Design

The underactuated system can apply both a surge force and yaw moment, but cannot directly apply a force in the sway direction. However, as discussed in Section III, the kinetic equations of the USVs motion in the sway and yaw directions are coupled via the added mass terms m_{23} and m_{32}. Using the approach proposed in [40], we take advantage of this coupling to find the yaw moment τ_ψ required to generate a virtual control input $\dot{r}_c$ that relates the time derivative of the yaw rate to the sway acceleration associated with the desired trajectory, such that the commanded control inputs are dynamically feasible.

Here, we propose a control design that includes two backstepping stages, similar to that used for the trajectory tracking of differential drive mobile robots with first order kinematic constraints [41], in which the first stage involves designing a kinematic tracking controller that satisfies velocity constraints and the second stage concerns designing a tracking controller incorporating the dynamics of the system. However, here, the first stage involves designing a kinematic position tracking controller and the nonholonomic second order (acceleration) constraints of the system are taken into account during a second stage of backstepping.

In the first stage, we use backstepping to derive a virtual control law for kinematic position tracking. Then, we propose a relaxed CBF and use the position tracking controller as a basis for the design of a kinematic safety-critical controller. In a second backstepping stage, a dynamic safety-critical tracking controller is then designed taking the nonholonomic acceleration constraints into account to generate the required physical control inputs given the virtual control inputs from the kinematic safety-critical controller. Lastly, since the control inputs include second order time derivatives of the USV states and the target's position, the Dynamic Surface Control Method is used to compute these derivatives, while simultaneously guaranteeing the overall stability of the closed loop system.

A. Kinematic Trajectory Tracking Controller

The first stage in the design of the trajectory tracking controller involves developing a stabilizing controller for the x- and y-position tracking errors. Later, the surge speed u_t and sway speed v_t required for satisfying the tracking requirements are used as virtual control inputs to the second stage, in which the dynamics of the system and its nonholonomic second order acceleration constraint along the sway axis are taken into account.

Define the position error surfaces $\tilde{x} := x - x_d$ and $\tilde{y} := y - y_d$.

Theorem 2: Let $\boldsymbol{x}_d = [x_d \;\; y_d]^T$ be a continuously differentiable reference trajectory, and $\tilde{x}$ and $\tilde{y}$ be a pair of position error surfaces, as defined above. If the virtual control inputs are taken to be

$$\dot{x}_t = -k_x \tilde{x} + \dot{x}_d \tag{23}$$

and

$$\dot{y}_t = -k_y \tilde{y} + \dot{y}_d \tag{24}$$

where $k_x > 0$ and $k_y > 0$ are constants, then the closed loop kinematic tracking control system is uniformly globally exponentially stable.

Proof 2: Consider the candidate Lyapunov function

$$V_1 = \frac{1}{2}\left(\tilde{x}^2 + \tilde{y}^2\right). \tag{25}$$

Taking the derivative of V_1 gives

$$\begin{aligned}\dot{V}_1 &= \tilde{x}\dot{\tilde{x}} + \tilde{y}\dot{\tilde{y}} \\ &= \tilde{x}\left(\dot{x} - \dot{x}_d\right) + \tilde{y}\left(\dot{y} - \dot{y}_d\right).\end{aligned} \tag{26}$$

If we replace $\dot{x}$ in (26) with the virtual control input

$$\dot{x}_c = -k_x \tilde{x} + \dot{x}_d \tag{27}$$

and $\dot{y}$ in (26) with the virtual control input

$$\dot{y}_c = -k_y \tilde{y} + \dot{y}_d \tag{28}$$

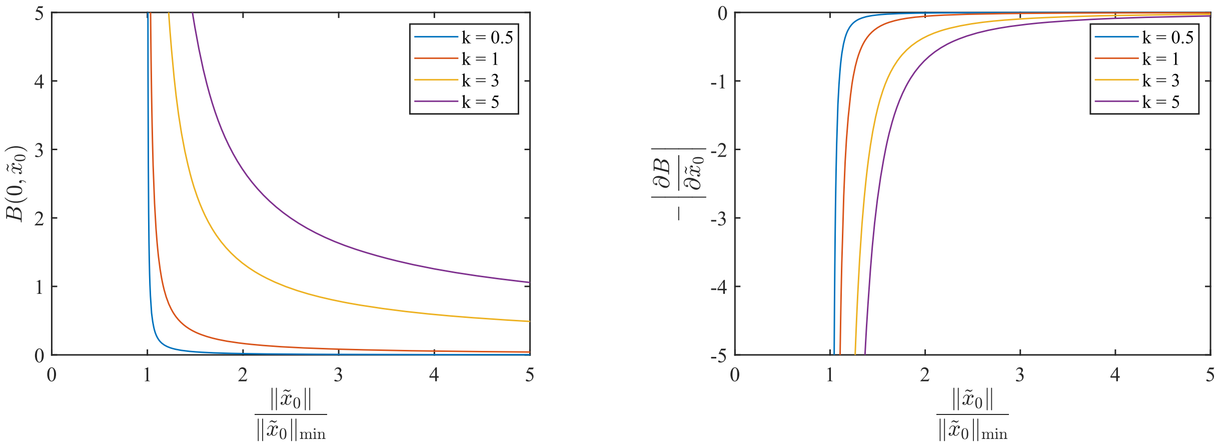
where $k_x > 0$ and $k_y > 0$ are constants, we get

$$\dot{V}_1 = -k_x \tilde{x}^2 - k_y \tilde{y}^2 \tag{29}$$

such that $\dot{V}_1 < 0$ for all $\tilde{x} \neq 0$ and $\tilde{y} \neq 0$.

From (25) and (29) it can be seen that

$$\dot{V}_1 \leq -2\mu_1 V_1 \tag{30}$$



(a)

(b)

Fig. 2. Effects of k on the variation of (a) the relaxed control barrier function B, and (b) the negative magnitude of its gradient $-|\partial B/\partial \tilde{\boldsymbol{x}}_0|$, as a function of the normalized distance between the USV and target $\|\tilde{\boldsymbol{x}}_0\|/\|\tilde{\boldsymbol{x}}_0\|_{\min}$. To focus on the variation of B with $\tilde{\boldsymbol{x}}_0$ only, we take $\tilde{\boldsymbol{x}} = 0$ and examine $B(0, \tilde{\boldsymbol{x}}_0)$.

such that

$$V_1 \le V_{10}\mathrm{e}^{-2\mu_1(t-t_0)} \quad \forall t \ge t_0 \tag{31}$$

where V_{10} is the value of V_1 at the initial time $t = t_0$ and $\mu_1 = \min\{k_x,\ k_y\}$. Let $\tilde{\boldsymbol{x}} := [\tilde{x}\ \ \tilde{y}]^T$. Using (25) gives

$$\|\tilde{\boldsymbol{x}}\| \le \sqrt{2V_{10}\mathrm{e}^{-2\mu_1(t-t_0)}}. \tag{32}$$

Thus, the position tracking errors exponentially decrease as t increases for times $t > t_0$. Since this result holds for any t_0 and V_1 is radially unbounded, i.e., $V_1 \to \infty$ for any $\tilde{x}, \tilde{y} \to \infty$, the controller renders the closed loop kinematic tracking system uniformly globally exponentially stable. □

Finally, the velocity components in the body-fixed reference frame corresponding to the virtual control inputs $\dot{x}_t$ and $\dot{y}_t$ are determined using the transformation

$$\begin{aligned} u_t &= \dot{x}_t \cos\psi + \dot{y}_t \sin\psi, \\ v_t &= -\dot{x}_t \sin\psi + \dot{y}_t \cos\psi. \end{aligned} \tag{33}$$

B. Kinematic Safety-Critical Controller

We propose the use of a modified form of the relaxed control barrier function proposed by [15]. Here, we take the relaxed CBF to be

$$B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) = \frac{1}{2}\left\{\frac{1}{\left[\|\tilde{\boldsymbol{x}}_0\|^{1/k} - \|\tilde{\boldsymbol{x}}_0\|_{\min}^{1/k}\right]} + \tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{x}}\right\} \tag{34}$$

where k can be selected to modify how agressively the safety critical controller responds as the USV approaches the safety "barrier" located at $\|\tilde{\boldsymbol{x}}_0\| = \|\tilde{\boldsymbol{x}}_0\|_{\min}$. To see the effects of varying k, consider the gradient of $B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0)$ taken with respect to $\tilde{\boldsymbol{x}}_0$

$$\frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} = -\frac{\tilde{\boldsymbol{x}}_0\|\tilde{\boldsymbol{x}}_0\|^{(1-2\,k)/k}}{2k\left[\|\tilde{\boldsymbol{x}}_0\|^{1/k} - \|\tilde{\boldsymbol{x}}_0\|_{\min}^{1/k}\right]^2}. \tag{35}$$

As shown in Fig. 2, as k increases the magnitudes of both $B(0, \tilde{\boldsymbol{x}}_0)$ and $\partial B/\partial \tilde{\boldsymbol{x}}_0$ are larger for all $\tilde{\boldsymbol{x}}_0$ (we set $\tilde{\boldsymbol{x}} = 0$ in $B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0)$ here to focus on the effects of the distance between the USV and target). However, the magnitudes of both functions increase more slowly as $\|\tilde{\boldsymbol{x}}_0\|$ approaches $\|\tilde{\boldsymbol{x}}_0\|_{\min}$. As a result, when k is larger, the safety critical control input starts to act sooner and more gradually as the USV approaches the boundary between the safe and unsafe operating region. While the safety critical control input is determined at the kinematic level, when it changes impulsively the associated control forces and moments required at the dynamic level (i.e., for τ_x and τ_ψ) will be large. Therefore, selecting higher values of k will help to reduce situations in which the USV actuators become saturated as it approaches the safe/unsafe boundary. Lastly, note that the rCBF proposed in [15] is recovered when $k = 1/2$.

From the first two components of (15), it can be seen that the kinematic equation of motion for the virtual control inputs u_c and v_c is

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u_c \\ v_c \end{bmatrix}. \tag{36}$$

Thus, the virtual tracking control inputs are in the control affine form of (1) with $\boldsymbol{f}(\boldsymbol{x}) = 0$ and

$$\boldsymbol{g}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \tag{37}$$

From (34), we have

$$\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{x}}. \tag{38}$$

This latter expression can be used together with (4), (7), (34), (35), and (37) in (8) to obtain the kinematic virtual control input signal $\boldsymbol{u}_c := [u_c\ \ v_c]^T$.

Remark 2: For convex optimization problems in which the KKT conditions are satisfied, the sensitivity of the optimum control input $\boldsymbol{u}_c$ to perturbations of the inequality constraint

$\dot{B} - \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) - \beta < 0$ can be examined using the magnitude of the Lagrange multiplier λ.

Consider the perturbed version of the original optimization problem

$$\min_{\boldsymbol{u}_c} \|\boldsymbol{u}_c - \boldsymbol{u}_t\|^2,$$

$$\text{s.t.}$$

$$\dot{B} - \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) - \beta < \delta\beta \tag{39}$$

where $\delta\beta$ is a perturbation of the constraint. When $\delta\beta > 0$ the constraint is loosened (relaxed) and when $\delta\beta < 0$ the constraint is tightened. Let $\boldsymbol{u}_c^*$ be the optimal solution for the unperturbed problem and λ^* be the optimal value of the Lagrange multiplier for the unperturbed problem. As shown in [42], the optimal solution to the perturbed problem $\boldsymbol{u}_c$ is lower bounded as

$$\boldsymbol{u}_c \geq \boldsymbol{u}_c^* - \lambda^* \delta\beta. \tag{40}$$

Since the difference between the optimal solutions to the perturbed and unperturbed problems depends on the optimal value of the Lagrange multiplier for the unperturbed problem, the magnitude of λ^* can be used to investigate the sensitivity of the optimum control input $\boldsymbol{u}_c$ to perturbations of the inequality constraint $\dot{B} - \alpha B(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_0) - \beta < 0$. Specifically, when the magnitude of λ^* is large, small perturbations $\delta\beta$ cause large changes in the value of the optimal control input $\boldsymbol{u}_c$, and when the magnitude of λ^* is small the changes in $\boldsymbol{u}_c$ are more gradual.

The safety critical controller is activated when $I > J$. In this case, the first of the two KKT Conditions in (12) can be solved to obtain

$$\lambda^* = \frac{2(I - J)}{\|L_g B\|^2}. \tag{41}$$

From the second KKT condition in (12) we have

$$(I - J) = -L_g B \cdot (\boldsymbol{u}_c^* - \boldsymbol{u}_t) \tag{42}$$

so that

$$\lambda^* = -\frac{2 L_g B \cdot (\boldsymbol{u}_c^* - \boldsymbol{u}_t)}{\|L_g B\|^2}. \tag{43}$$

Inserting this latter result into inequality (40) gives

$$\boldsymbol{u}_c \geq \boldsymbol{u}_c^* + \frac{2 L_g B \cdot (\boldsymbol{u}_c^* - \boldsymbol{u}_t)}{\|L_g B\|^2} \delta\beta. \tag{44}$$

Therefore, λ^* varies as

$$\lambda^* \sim \frac{1}{\|L_g B\|} \tag{45}$$

where from (6), (35), and (38) we have

$$\begin{aligned} \|L_g B\| &= \left\| \left(\frac{\partial B}{\partial \tilde{\boldsymbol{x}}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot \boldsymbol{g}(\psi) \right\| \\ &= \left\| \left(\tilde{\boldsymbol{x}} + \frac{\partial B}{\partial \tilde{\boldsymbol{x}}_0} \right) \cdot \boldsymbol{g}(\psi) \right\|. \end{aligned} \tag{46}$$

Consider the case when the system first approaches an target from afar and the tracking error $\tilde{\boldsymbol{x}}$ is small. For a given value of $\boldsymbol{g}(\psi)$, $\partial B / \partial \tilde{\boldsymbol{x}}_0$ dominates the value of $\|L_g B\|$. As shown in Fig. 2, when k is small the value of $|\partial B / \partial \tilde{\boldsymbol{x}}_0|$ remains small until very close to the boundary of the safe set $\|\tilde{\boldsymbol{x}}_0\| = \|\tilde{\boldsymbol{x}}_0\|_{\text{min}}$. Since λ^* is inversely proportional to $\|L_g B\|$, the optimal control input is more sensitive to perturbations in the constraint for lower values of k, so that the control input varies rapidly near the boundary of the safe set, resulting in reduced stability.

C. Dynamic Safety-Critical Tracking Control

To determine the physical control inputs τ_x and τ_ψ that generate the virtual control input $\boldsymbol{u}_c$, we define the velocity error surfaces $\tilde{u} := u - u_c$, $\tilde{v} := v - v_c$ and $\tilde{r} := r - r_c$. As will be shown in this section, the time derivative of r_c will be used as a virtual control input that drives the sway acceleration error $\dot{\tilde{v}}$ to zero while preserving the sway-yaw coupling of the system's dynamics. This ensures that the resulting physical control inputs τ_x and τ_ψ produce a dynamically feasible motion, i.e., a motion that satisfies the nonholonomic acceleration constraints of the vehicle. The term r_c is obtained by integrating the virtual control input $\dot{r}_c$.

Theorem 3: Let $\boldsymbol{x}_d = [x_d \quad y_d]^T$ be a twice continuously differentiable reference trajectory, and $\tilde{u}$, $\tilde{v}$, and $\tilde{r}$ be a set of velocity error surfaces, as defined above. If the virtual control input $\dot{r}_c$ and physical control inputs τ_x and τ_ψ are taken to be

$$\begin{aligned} \dot{r}_c &= \frac{a_\psi}{a_y} \left[-k_v \tilde{v} - f_y + \dot{v}_c \right] + f_\psi \\ \tau_x &= m_{11} \left[-k_u \tilde{u} - f_x + \dot{u}_c \right] \\ \tau_\psi &= \frac{m_{33}}{a_\psi} \left[-k_r \tilde{r} - f_\psi + \dot{r}_c \right] \end{aligned} \tag{47}$$

where $k_u > 0$, $k_v > 0$ and $k_\psi > 0$ are constants, then the closed loop dynamic tracking control system is uniformly globally exponentially stable.

Proof 3: Consider the candidate Lyapunov function

$$V_2 = \frac{1}{2} \left(\tilde{u}^2 + \tilde{v}^2 + \tilde{r}^2 \right). \tag{48}$$

Using (15) and the definitions of the velocity error surfaces, the time derivative of V_2 can be written as

$$\begin{aligned} \dot{V}_2 &= \tilde{u}\dot{\tilde{u}} + \tilde{v}\dot{\tilde{v}} + \tilde{r}\dot{\tilde{r}} \\ &= \tilde{u} \left(f_x + \frac{1}{m_{11}} \tau_x - \dot{u}_c \right) + \tilde{v} \left(f_y + a_y \frac{\tau_\psi}{m_{33}} - \dot{v}_c \right) \\ &\quad + \tilde{r} \left(f_\psi + \frac{a_\psi}{m_{33}} \tau_\psi - \dot{r}_c \right). \end{aligned} \tag{49}$$

Note that the last two terms of (49) are coupled by the control input τ_ψ. Using (20), we can eliminate τ_ψ in (19) to get

$$\begin{aligned} \dot{v} &= f_y + a_y \frac{\tau_\psi}{m_{33}} \\ &= f_y + \frac{a_y}{a_\psi} \left[\dot{r} - f_\psi \right]. \end{aligned} \tag{50}$$

Replacing $\dot r$ in (50) with the virtual control input $\dot r_c$ (49) can be rewritten as

$$\dot V_2 = \tilde u\left(f_x + \frac{1}{m_{11}}\tau_x - \dot u_c\right) + \tilde v\left(f_y + \frac{a_y}{a_\psi}\left[\dot r_c - f_\psi\right] - \dot v_c\right) + \tilde r\left(f_\psi + \frac{a_\psi}{m_{33}}\tau_\psi - \dot r_c\right). \tag{51}$$

Using (47) in (51) gives

$$\dot V_2 = -k_u\tilde u^2 - k_v\tilde v^2 - k_\psi\tilde r^2 \tag{52}$$

such that $\dot V_2 < 0$ for all $\tilde u \neq 0$, $\tilde v \neq 0$ and $\tilde r \neq 0$.

From (48) and (52), it can be seen that

$$\dot V_2 \leq -2\mu_2 V_2 \tag{53}$$

such that

$$V_2 \leq V_{20}\mathrm{e}^{-2\mu_2(t-t_0)}, \quad \forall t \geq t_0 \tag{54}$$

where V_{20} is the value of V_2 at the initial time $t = t_0$ and $\mu_2 = \min\{k_u,\ k_v,\ k_\psi\}$. Let $\tilde{\boldsymbol{u}} := [\tilde u\ \ \tilde v\ \ \tilde r]^T$. Using (48) gives

$$\|\tilde{\boldsymbol{u}}\| \leq \sqrt{2V_{20}\mathrm{e}^{-2\mu_2(t-t_0)}}. \tag{55}$$

Thus, the velocity tracking errors exponentially decrease as t increases for times $t > t_0$. Since this result holds for any t_0 and V_2 is radially unbounded, i.e., $V_2 \to \infty$ for any $\tilde u, \tilde v, \tilde r \to \infty$, the dynamic tracking controller renders the closed loop system uniformly globally exponentially stable. □

Remark 3: From (49), (51), and (52), it can be seen that selecting the virtual control input $\dot r_c$ to be that given in (47) yields $\dot{\tilde v} = -k_v\tilde v$. Since the controller drives $\tilde v$ to zero, and hence also drives the sway acceleration error $\dot{\tilde v}$ to zero while preserving the sway-yaw coupling of the system's dynamics, the resulting motion satisfies the nonholonomic acceleration constraints of the vehicle.

The virtual control input $\dot r_c$ and the control inputs τ_x and τ_ψ are determined using (31), which requires knowledge of the time derivatives of u_c and v_c. Here, the computation of $\dot u_c$ and $\dot v_c$ is simplified using the Dynamic Surface Control (DSC) Method. This method was originally proposed in [43] and [44] as a means to circumvent the *explosion of complexity* problem that arises when using the traditional Backstepping Method in which a large number of terms must be included in the computation of the time derivatives of the virtual control inputs. With DSC the time derivatives are essentially computed by implementing a set of first order filters, which are shown to converge to the true time derivatives using a Lyapunov stability approach. The DSC Method was refined in [45] using singular perturbation theory to show that the conservativeness required from the original DSC approach can be relaxed via the selection of a suitably small filter parameter and that the use of high control gains is not required for achieving small steady-state tracking errors. In [46], the DSC Method was used to design a controller for a fully-actuated underwater vehicle with actuator magnitude constraints. The results are compared to the use of a traditional backstepping controller.

D. *DSC-Based Safety-Critical Tracking Controller*

Let $\hat u_c$ and $\hat v_c$ be filtered estimates of u_c and v_c, respectively. These estimates will be used to approximate $\dot u_c$ and $\dot v_c$. Redefine the velocity tracking error surfaces for the surge speed and sway speed, respectively, as $\tilde u := u - \hat u_c$ and $\tilde v := v - \hat v_c$, and define the set of estimation errors as $\tilde u_c := \hat u_c - u_c$ and $\tilde v_c := \hat v_c - v_c$. Since we already have an exact expression for $\dot r_c$ its estimate is not needed, so define $\tilde r := r - r_c$, as before.

Now, take the virtual control input $\dot r_c$ and physical control inputs τ_x and τ_ψ to be

$$\begin{aligned}\dot r_c &= \frac{a_\psi}{a_y}\left[-k_v\left(v - v_c\right) - f_y + \dot{\hat v}_c\right] + f_\psi \\ \tau_x &= m_{11}\left[-k_u\left(u - u_c\right) - f_x + \dot{\hat u}_c\right] \\ \tau_\psi &= \frac{m_{33}}{a_\psi}\left[-k_r\left(r - r_c\right) - f_\psi + \dot r_c\right].\end{aligned} \tag{56}$$

Then, the closed loop error system for the dynamics of the system is

$$\begin{aligned}\dot{\tilde u} &= -\ k_u\tilde u - k_u\tilde u_c \\ T_d\dot{\hat u}_c &= -\ \tilde u_c \\ \dot{\tilde v} &= -\ k_v\tilde v - k_v\tilde v_c \\ T_d\dot{\hat v}_c &= -\ \tilde v_c \\ \dot{\tilde r} &= -\ k_r\tilde r\end{aligned} \tag{57}$$

where $T_d \in (0, 1)$ is the time coefficient of the filter and the initial values of the estimates are taken to be $\hat u_c(0) = 0$ and $\hat v_c(0) = 0$. Note that the first and second equations in (57) are interconnected, and the third and fourth equations in (57) are interconnected. In each of the two interconnected systems, the filter time constant T_d is much shorter than the time constant of the physical system, so that it acts like a small parameter. Thus, the stability of each of the two interconnected systems can be analyzed as a singular perturbation problem.

Theorem 4: Consider the dynamic portion of system (15), which is given by (18)–(20) and assume that the terms f_x, f_y, and f_ψ in these equations are Lipschitz continuous. Assume that the trajectory $\boldsymbol{x}_d(t)$, its first derivative $\dot{\boldsymbol{x}}_d(t)$, the position of the target $\boldsymbol{x}_0(t)$, and its first derivative $\dot{\boldsymbol{x}}_0(t)$, are smooth and bounded. If the control inputs and virtual control input are given by (47), then there exist suitably large control gains k_u, k_v, and k_r, and a suitably small filter parameter $T_d \in (0, 1)$ that render the solution trajectories of the closed loop error tracking dynamics (57) semiglobally uniformly ultimately bounded, in the sense that $\dot V_2 \leq -k_s V_2 + \delta_V$, where V_2 is a Lyapunov function, $k_s \in \mathbb{R}^+$ is a constant related to k_u, k_v and k_r, and $\delta_V \in \mathbb{R}$ is a constant that is not necessarily small.

TABLE I
COMPOSITION OF THE INSPECTION TRAJECTORY

Segment	Type	Start point/center		End point/arclength	
		N [m]	E [m]	N [m]	E [m]
1	line	$-25-0.5/\sqrt{2}$	$-25+0.5/\sqrt{2}$	$25-0.5/\sqrt{2}$	$25+0.5/\sqrt{2}$
2	arc	$25-(r_t+0.5)/\sqrt{2}$	$25+(r_t+0.5)/\sqrt{2}$	$\pi/2$	
3	arc	$25+(r_t-0.5)/\sqrt{2}$	$25+(3r_t+0.5)/\sqrt{2}$	π	
4	line	$25+(2r_t-0.5)/\sqrt{2}$	$25+(4r_t+0.5)/\sqrt{2}$	$25+(4r_t+0.5)/\sqrt{2}$	$25+(2r_t-0.5)/\sqrt{2}$
5	arc	$25+(3r_t+0.5)/\sqrt{2}$	$25+(r_t-0.5)/\sqrt{2}$	π	
6	arc	$25+(r_t+0.5)/\sqrt{2}$	$25-(r_t+0.5)/\sqrt{2}$	$\pi/2$	
7	line	$25+0.5/\sqrt{2}$	$25-0.5/\sqrt{2}$	$-25+0.5/\sqrt{2}$	$-25-0.5/\sqrt{2}$

Arclengths are given in radians and r_t is the radius of each circular arc.

Proof 4: The proof follows from [45, Th. 2]. □

To determine the physical control inputs τ_x and τ_ψ, and the virtual control input $\dot{r}_c$, $\dot{\hat{u}}_c$, and $\dot{\hat{v}}_c$ are approximated from the second and fourth subequations of (57) and substituted into (56).

V. Illustrative Simulations

We demonstrate an implementation of the safety-critical trajectory-tracking controller by simulating the approach of an underactuated, differential thrust driven, USV (the WAMV-USV16, see Fig. 1) to both stationary and moving targets. Imagine that we would like to have the USV briefly approach a target for inspection, but that there is some minimum safe distance that must be respected. To accomplish our safe approach, we plan a simple straight-line trajectory that crosses through the unsafe space surrounding the target and rely on the reactive safety-critical controller to instead move the USV along the outside of the boundary of the unsafe set. Here, we focus on a 50 m × 50 m operating space in a North-East-Down reference frame. The origin of the reference frame is located at the center of the operating space.

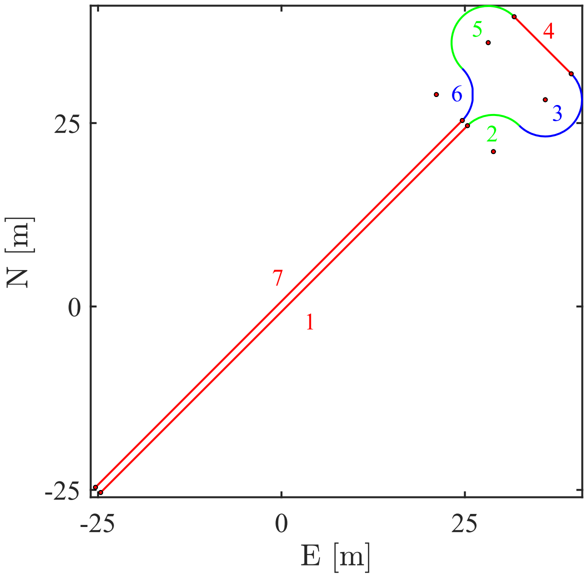


Fig. 3. Composition of the inspection trajectory. Each segment is labelled with the number corresponding to the list of segments in Table I.

A. Simulation Cases

Three simulation scenarios are considered as follows.

1) A stationary target is located at the point $(x_0, y_0) = (0.5, -0.5)$ m. The desired trajectory of the USV $\boldsymbol{x}_d(t)$ crosses through the unsafe region of the target in order to perform a 180° inspection of the target. This simulation scenario is commonly used in the literature for the evaluation of CBF-based controllers, see, e.g., [12], [14], [47], and [48].
2) A moving target crosses the operating space with a constant speed of $u_0 = 1.0$ m/s along the line given by $x_0(t) = 0$ m and $y_0(t) = u_0 t - 25$ m. Without the safety-critical controller, the USV would collide with the target at the point $(x, y) = (0, 0)$ m. The simulation demonstrates how a target can be inspected while in motion.

   In both simulation Cases A and B the desired trajectory of the USV is

$$x_d(t) = u_0 t - 25 \text{ m}$$

$$y_d(t) = u_0 t - 25 \text{ m} \tag{58}$$

giving the USV a desired surge speed of $u_d = \sqrt{2}u_0$. Thus, $\boldsymbol{x}_d(t)$ diagonally crosses the operating space from its southwest corner at time $t = 0$ s and reaches its northeast corner at time $t = 50$ s.

3) A stationary target is located at the point $(x_0, y_0) = (0, 0)$ m. In order to demonstrate a 360° inspection of the target, the desired trajectory $\boldsymbol{x}_d(t)$ is designed to cross the unsafe region around the target from two opposing sides. The desired trajectory is continuous and composed of three straight lines joined by four circular arcs, with a radius of $r_t = 5$ m each. The endpoints of the line segments and centers of the circular arcs are presented in Table I and shown in Fig. 3. The time dependence of $\boldsymbol{x}_d(t)$ is formulated so that the desired speed along the trajectory is $\|\dot{\boldsymbol{x}}_d(t)\| = \sqrt{2}$ m/s. The continuous trajectory was selected for ease of construction and so that our single controller could be used in the simulations. In practice, one would likely use a set of two parallel, straight-line trajectories with the vehicle first crossing from one side, stopping, turning around 180° in place, and then crossing from the opposite side to complete the inspection.

TABLE II
CONTROL PARAMETERS FOR SIMULATION CASES A) AND B)

α	β	k	k_x	k_y	k_u	k_v	k_r	$\|\tilde{\boldsymbol{x}}_0\|_{\min}$ **[m]**	T_d **[s]**
0.75	1.0	5	0.5	0.5	0.075	0.075	0.075	5.0	0.01

Implementing such a system would require either some sort of high level planner that switches between a set of low level controllers, for example one controller for the straight line paths and a second controller for rotating in place, or the use of a low-level switching controller, e,g, see [49].

B. Vehicle Properties

The inertial and added mass terms of the USV are taken to be $m_{11} = 189.0$ kg, $m_{22} = 1036.4$ kg, $m_{33} = 2411.1$ kg-m^2, and $m_{23} = m_{32} = -543.5$ kg-m. Using these values in (16) gives $a_y = 0.595$ and $a_\psi = 1.134$.

Using a hydrodynamic model similar to the one developed in [50], the drag of the USV is taken to be

$$\boldsymbol{d} = \begin{bmatrix} 50\,u + 70\,u|u| \\ 948.2\,v + 385.4\,r \\ 385.4\,v + 1926.9\,r \end{bmatrix}. \tag{59}$$

Note that the fore-aft asymmetry of the USV couples the drag terms for the sway and yaw motion so that a yaw moment arises from a sway motion, and a sway force arises from a yawing motion because the submerged portion of the bow is curved upwards (with the draft decreasing slightly towards the bow) and the bow is finer than the stern.

The magnitude of the maximum thrust available from each of the two propellers is taken to be 2250 N, giving a maximum total thrust of $|\tau_x|_{\max} = 4500$ N. This is high for the propulsion systems typically used on 16 ft USVs, but achievable using currently available commercial electric outboards. It is assumed the propellers can generate the same maximum magnitude of thrust in both the forward and reverse directions. This assumption is an idealization and would be most accurate for a propeller operating in an *open water* test condition. Since the flow around the transom affects the flow around the propeller, the exact thrust profile will be speed- and orientation-dependent. As the main focus of this work is the design of a controller, the development of a more accurate hydrodynamic model of the propulsion system is beyond its scope.

The centerline-to-centerline separation of the USV demihulls is 2.0 m, so that the magnitude of the maximum yaw moment that can be generated about G (see Fig. 1) is $|\tau_\psi|_{\max} = 4500$ N-m.

C. Controller Tuning and Control Parameters

The controller was manually tuned. The control parameters used for all three simulation cases are shown in Table II.

The control parameters were tuned by hand using a trial and error approach. The procedure implemented is as follows.

1) The tracking controller was first tuned with no safety controller terms included. Straight-line and circular desired trajectories were generated. The tracking controller was tuned online by plotting the tracked trajectory superimposed on the desired trajectory, and revising the controller parameters until almost no error between the desired and tracked trajectories could be visually observed, apart from initial transients. To start, all tracking controller gains were set to 1.0. The simulations appeared to be most sensitive to changes in the speed controller gains k_u, k_v, and k_r, so these terms were tuned first. The values were first doubled or halved to see if the tracking errors decreased or increased, and then the delta in the values of the controller gains between simulations runs were either doubled or halved to achieve the best qualitative performance. The position tracking gains k_x and k_y were then similarly tuned.
2) Next, the CBF safety controller terms were implemented in the simulation. The radius of the safety “barrier” was set to $\|\tilde{\boldsymbol{x}}_0\|_{\min} = 5.0$ m. This number was selected because it is near the length of the WAMV USV16 used in the simulation model so that, relative to the size of the USV, the resulting maneuver requires a tight turn around the target. The time coefficient of the DSC filter was set to $T_d = 0.01$ s, as this is about an order of magnitude smaller than the time constants of the physical system. For example, a common sampling rate on USV platforms is 4 Hz (see [50]), which corresponds to a 0.25-s sampling period. During tuning it was found that the closed loop system required larger control gains and was less stable when smaller values of k were used. The system was more robust to changes in the desired trajectory and the motion of the target when a value of $k = 5$ was used, so k was set to this value. During initial tests with the safety controller implemented, α and β were both set to 1.0. The value of α determines how aggressively the safety controller reacts as the USV approaches the boundary of the safe set. The value of α was tuned so that the resulting safety maneuver was stable, but the trajectory of the USV still hugged the boundary of the safe set. For the simulation conditions tested, changes to the value of β did not noticeably affect the closed loop behavior of the system. Thus, this parameter was left at its initial value of $\beta = 1.0$. The tracking controller gains k_x, k_y, k_u, k_v, and k_r were then retuned slightly from their preliminary values in the first step above to yield the best qualitative performance.

D. Simulation Results

Case A: The results of the Case A simulations are presented in Figs. 4–10. The desired and actual trajectories of the USV are shown in Fig. 4. The USV follows the desired trajectory until just before reaching the boundary of the unsafe set (red dashed circle), it then turns sharply and transits along the boundary until passing the target and rejoining $\boldsymbol{x}_d$. The section of the USV trajectory between the two blue dots corresponds to the locations at which the safety-critical controller is active, i.e., when $I > J$ in (8). From Fig. 7, it can be seen that the safety-critical controller is activated at about $t = 20$ s and remains ON

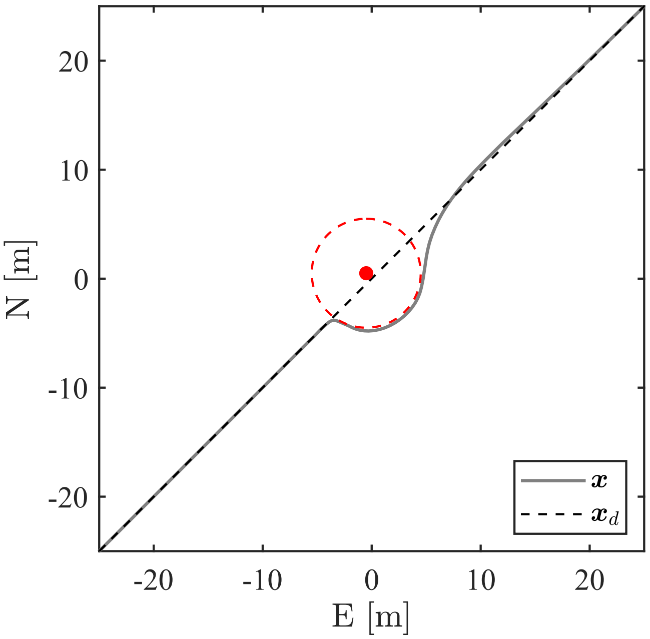


Fig. 4. Case A: Desired $\boldsymbol{x}_d$ and actual $\boldsymbol{x}$ trajectories of the USV as it moves around a stationary target, which is indicated by the red dot at $\boldsymbol{x}_0 = (0.5, -0.5)$ m. The blue dots denote the points along $\boldsymbol{x}$ at which the safety-critical controller is activated/deactivated. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is designated by a red dashed circle.

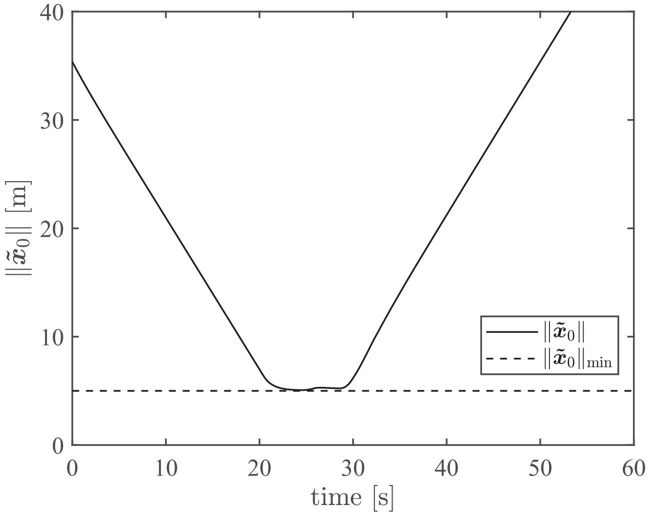


Fig. 5. Case A: Distance between USV and target $\|\tilde{\boldsymbol{x}}_0\|$ as a function of time. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is indicated with a dashed line.

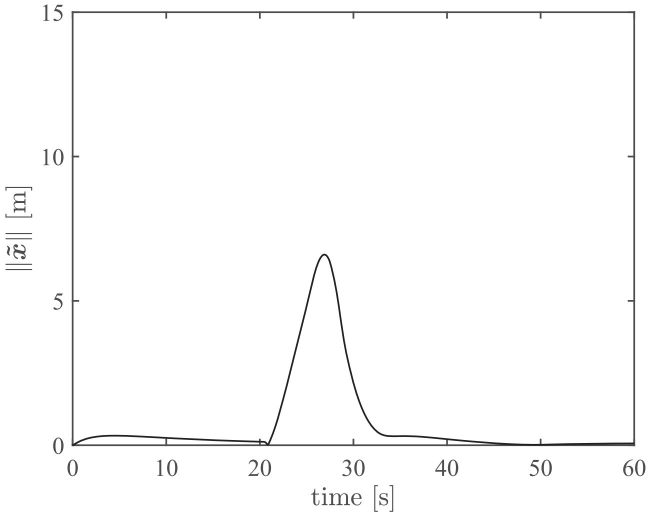


Fig. 6. Case A: Tracking error.

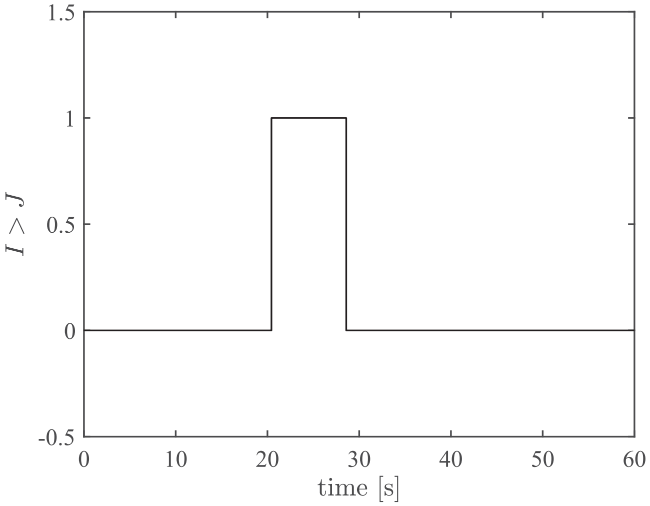


Fig. 7. Case A: Dimensionless, binary switching signal for the safety-critical controller. Signal has a value of 1 when $I > J$ and the safety-critical controller is active, otherwise its value is 0 and the control input is the same as that of the trajectory tracking controller $\boldsymbol{u}_t$.

until about $t = 30$ s. As can be seen in Fig. 5, the USV "hugs" the outer boundary of the unsafe set while the safety-critical controller is active. By comparing Figs. 4, 5, and 8, it can be seen that, even though the change in heading is larger when the USV approaches the boundary of the safe set $\|\tilde{\boldsymbol{x}}_0\| = \|\tilde{\boldsymbol{x}}_{\min}\|$ than the change in heading is when it moves away from the safe set (Fig. 4), the approach to the boundary is smoother, in terms of the distance to the target $\|\tilde{\boldsymbol{x}}_0\|$ versus time (see the changes in the slope of $\|\tilde{\boldsymbol{x}}_0\|$ just after 20 s just before 30 s in Fig. 5). This is because the USV decelerates more slowly when it approaches the target than it accelerates after passing the target. From looking at the surge force τ_x plotted in Fig. 8, it can be seen that the magnitude of the reverse thrust occurring just after 20 s is smaller that the magnitude of the forward thrust occurring just before 30 s.

The reason for this is that the trajectory tracking error grows during the time the USV is constrained to move along the boundary of the unsafe set, as shown in Fig. 6. When the tracking controller is reactivated after passing the target, i.e., when $I \leq J$ in (8), the large tracking errors result in large actuator efforts, which cause the USV to turn rapidly. This can be seen in Fig. 8, where the surge force and yaw moments generated by the controller are plotted. When the safety-critical controller is first activated, the controller commands reverse thrust and a positive yaw moment to rapidly slow the USV and turn it to

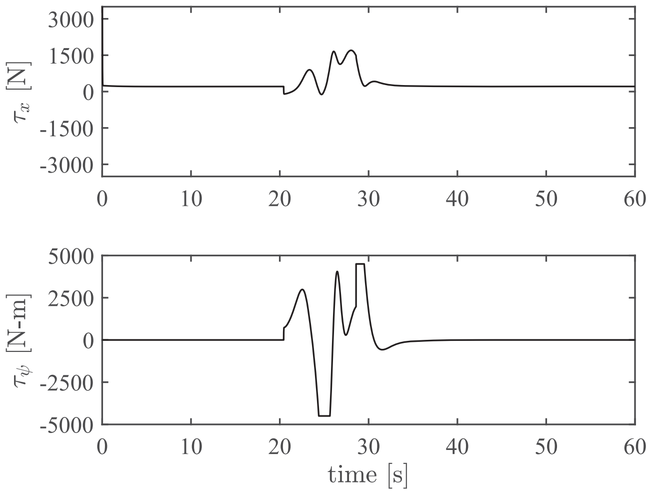


Fig. 8. Case A: Surge force τ_x and yaw moment τ_ψ.

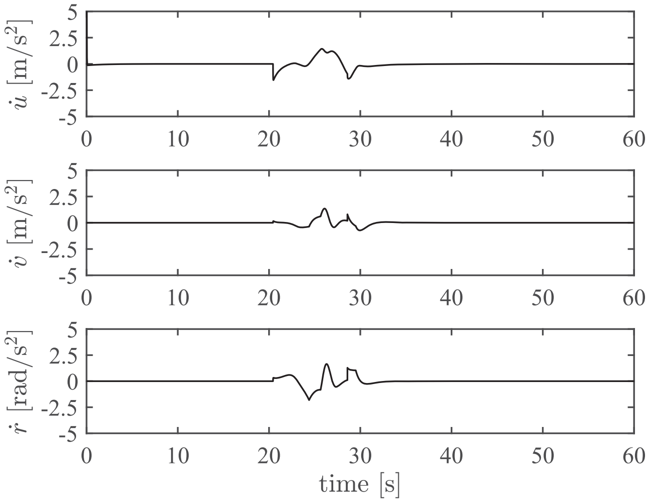


Fig. 9. Case A: Surge acceleration $\dot{u}$, sway acceleration $\dot{v}$, and yaw acceleration $\dot{r}$.

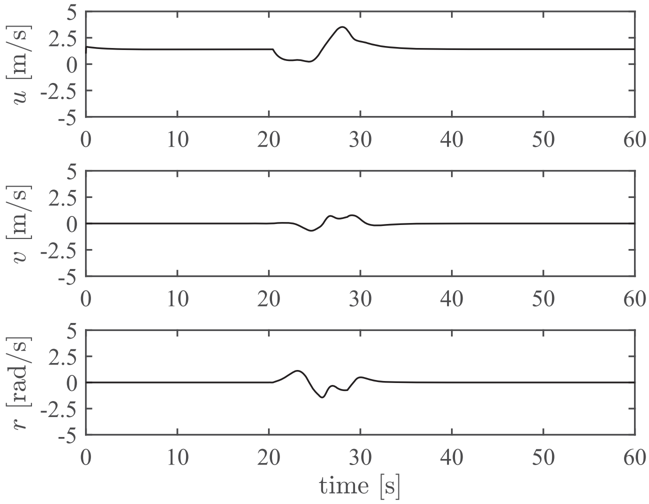


Fig. 10. Case A: Surge u speed, sway v speed, and yaw r speed.

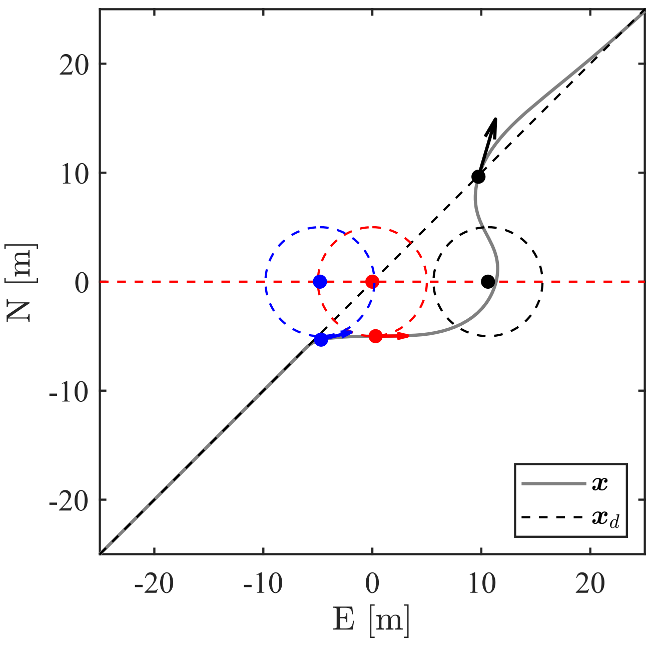


Fig. 11. Case B: Desired $\boldsymbol{x}_d$ and actual $\boldsymbol{x}$ trajectories of the USV as it approaches a moving target, whose trajectory $\boldsymbol{x}_0$ intersects the desired trajectory of the USV at time $t = 25$ s. The blue, red, and block dots enclosed by dashed circles of corresponding colors plotted along $\boldsymbol{x}_0$ indicate the positions of the target and unsafe set at times $t = 20$ s, $t = 25$ s, and $t = 35$ s, respectively. The dots of corresponding color plotted along $\boldsymbol{x}$ represent the position of the USV at the same times. The orientation and length of the arrows indicate the heading angle and relative speed of the USV.

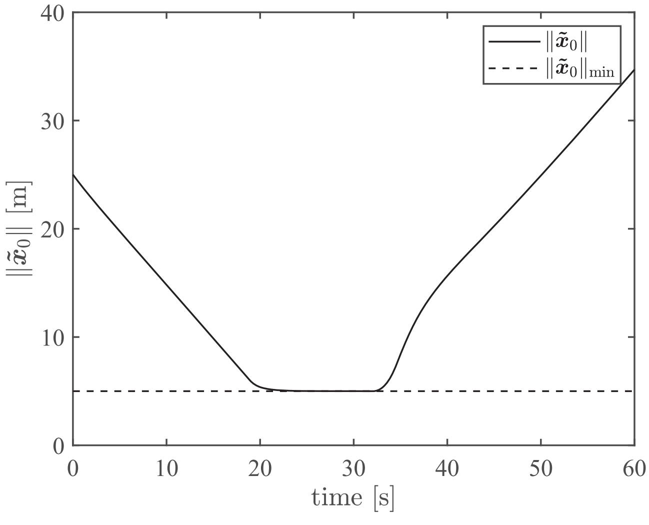


Fig. 12. Case B: Distance between USV and target $\|\tilde{\boldsymbol{x}}_0\|$ as a function of time. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is indicated with a dashed line.

starboard so that it can transit the boundary of the unsafe set. When the trajectory tracking controller is reactivated, a large, briefly saturated, yaw moment is produced.

Case B: The results of the Case B simulations are presented in Figs. 11–17. The trajectory of the target, and the desired and actual trajectories of the USV are shown in Fig. 11. The USV follows the desired trajectory until just before reaching

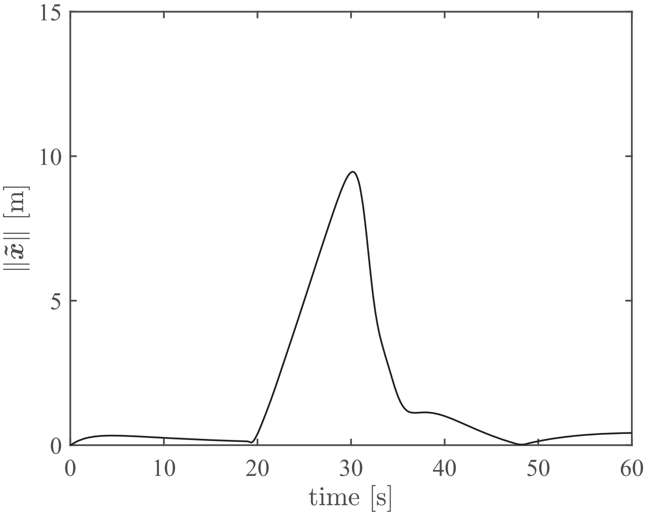


Fig. 13. Case B: Tracking error.

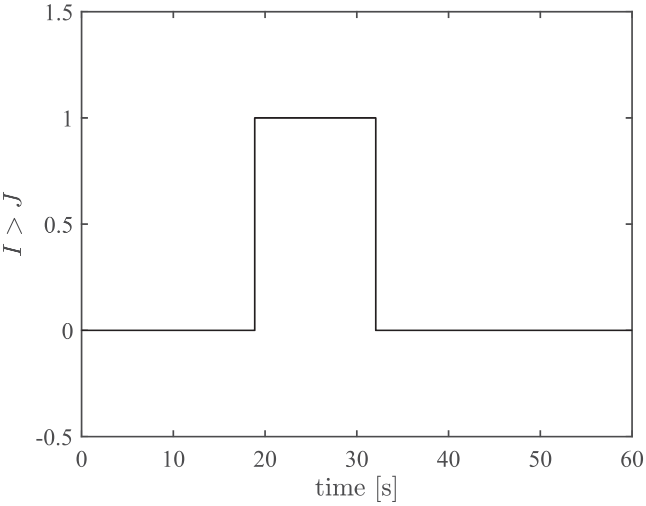


Fig. 14. Case B: Dimensionless, binary switching signal for the safety-critical controller. Signal has a value of 1 when $I > J$ and the safety-critical controller is active, otherwise its value is 0 and the control input is the same as that of the trajectory tracking controller $\boldsymbol{u}_t$.

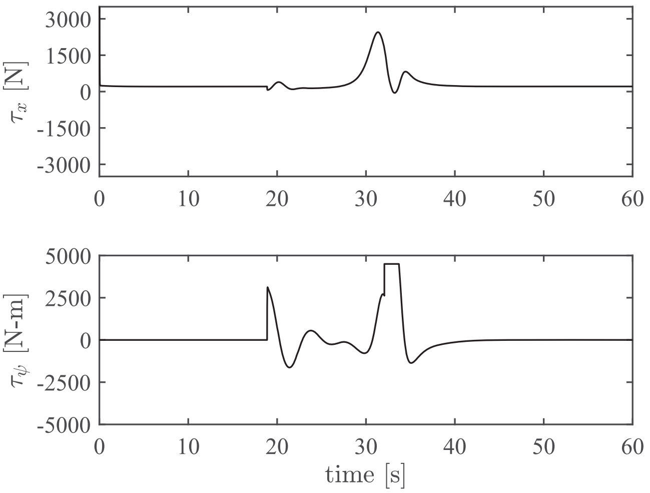


Fig. 15. Case B: Surge force τ_x and yaw moment τ_ψ.

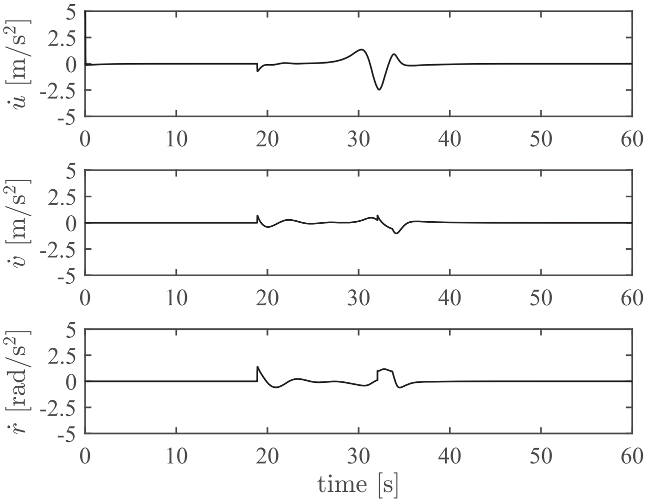


Fig. 16. Case B: Surge acceleration $\dot{u}$, sway acceleration $\dot{v}$, and yaw acceleration $\dot{r}$.

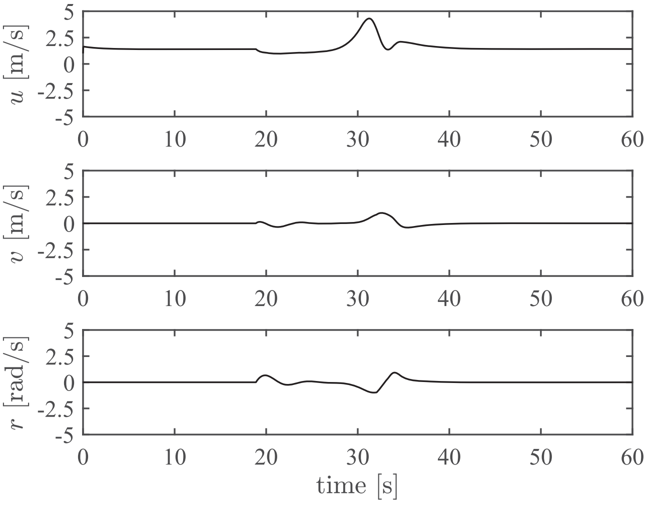


Fig. 17. Case B: Surge u speed, sway v speed, and yaw r speed.

the boundary of the unsafe set of the moving target at $t = 20$ s, when, as can be seen in Fig. 14, the safety-critical controller is activated. The USV then turns sharply and moves in parallel with the target for about 11 m and then turns to port crossing in front of the target, always remaining at the boundary of the moving unsafe set (Fig. 12), until the safety-critical controller is deactivated (at around $t = 31$ s, see Fig. 14). After this time the tracking controller causes the USV to turn sharply to starboard and back to the desired trajectory (Fig. 13).

In this simulation scenario the USV moves parallel to the target before crossing in front of it. Because of this the USV would have a clear view of the starboard, fore and port sides of the target. By employing a high level trajectory planner to plan multiple crossings of the target's trajectory, slightly before and

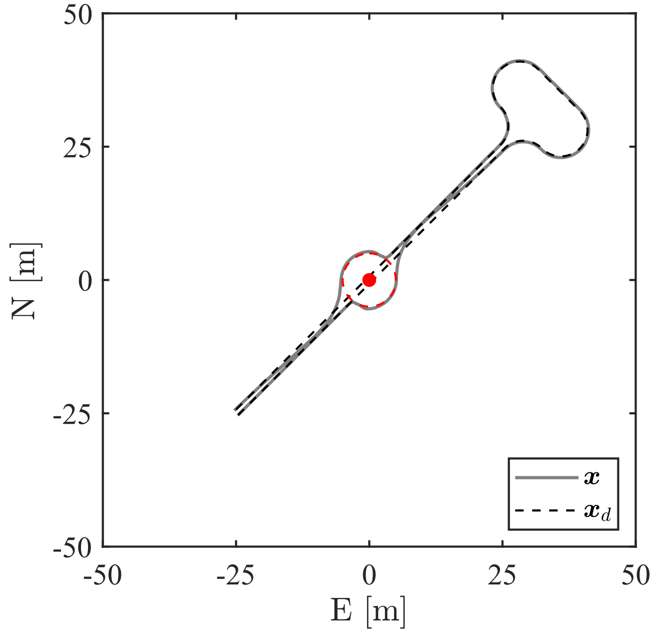


Fig. 18. Case C: Desired $\boldsymbol{x}_d$ and actual $\boldsymbol{x}$ trajectories of the USV as it moves around a stationary target, which is indicated by the red dot at $\boldsymbol{x}_0 = (0, 0)$ m. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is designated by a red dashed circle.

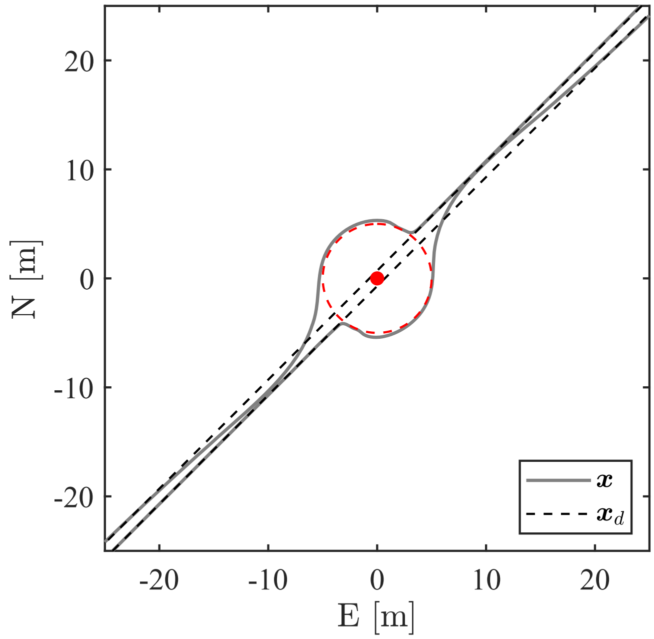


Fig. 19. Case C: Close up of the desired $\boldsymbol{x}_d$ and actual $\boldsymbol{x}$ trajectories of the USV shown in Fig. 18. The stationary target, which is indicated by the red dot at $\boldsymbol{x}_0 = (0, 0)$m. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is designated by a red dashed circle.

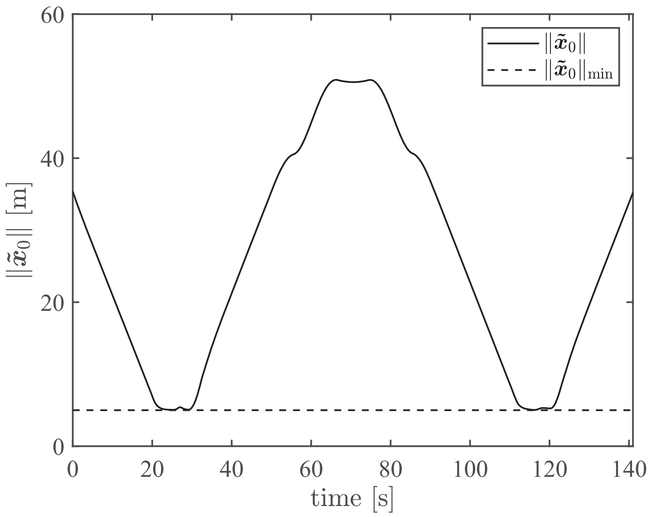


Fig. 20. Case C: Distance between USV and target $\|\tilde{\boldsymbol{x}}_0\|$ as a function of time. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is indicated with a dashed line.

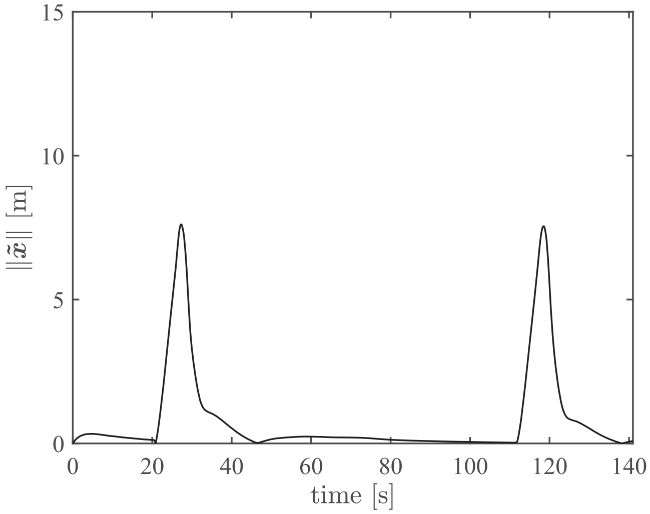


Fig. 21. Case C: Tracking error.

after the target's anticipated position, it may be possible to build a complete 360° view of the target.

Case C: The results of the Case C simulations are presented in Figs. 18–25. This simulation scenario is selected to provide an explicit demonstration of how one could use the controller to perform a 360° inspection of a stationary target. The straight-line portions of this trajectory are effectively the same as the trajectory used in the Case A simulations. As expected, the tracking errors, switching signal, actuator forces, accelerations, and body-fixed velocities for the corresponding portions of the trajectories in the two simulation cases are the same.

VI. Comparison With MPC

We compare our proposed approach with a nonlinear Model Predictive Control (MPC) implementation. In particular, we used the MPC-CBF formulation defined by Zeng et al. [48], which combines the theory of CBF and MPC in one formulation. In their work, the authors showed two example applications: a 2-D double integrator, which is a linear system, and a competitive car racing problem, which involves a highly nonlinear system. In

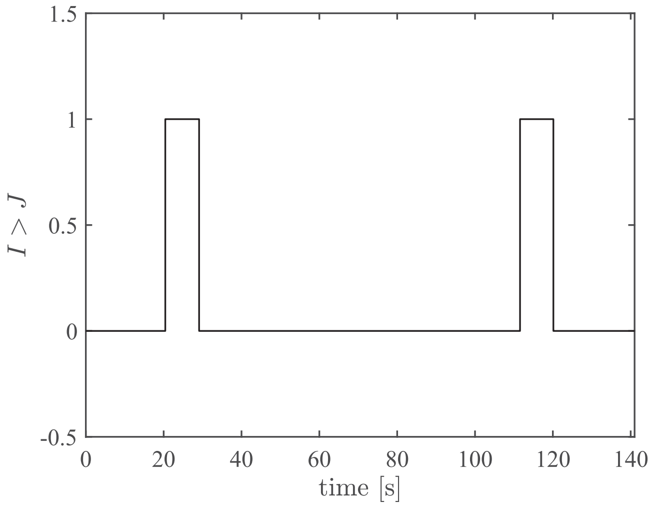


Fig. 22. Case C: Dimensionless, binary switching signal for the safety-critical controller. Signal has a value of 1 when $I > J$ and the safety-critical controller is active, otherwise its value is 0 and the control input is the same as that of the trajectory tracking controller $\boldsymbol{u}_t$.

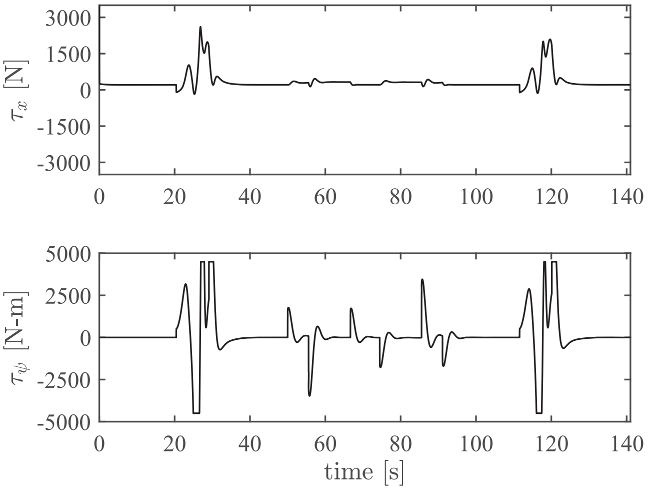


Fig. 23. Case C: Surge force τ_x and yaw moment τ_ψ.

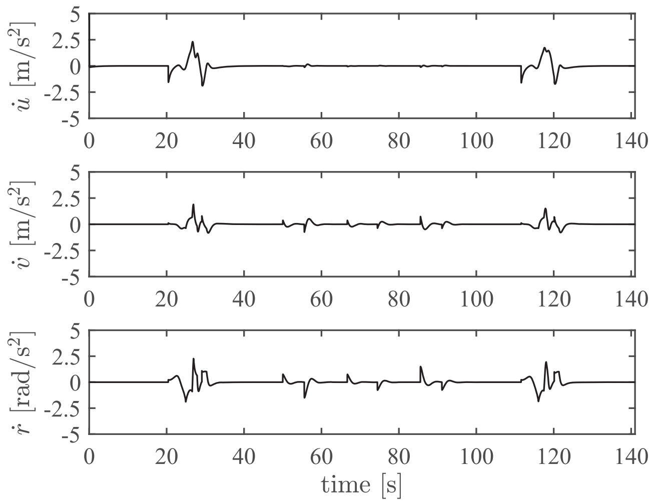


Fig. 24. Case C: Surge acceleration $\dot{u}$, sway acceleration $\dot{v}$, and yaw acceleration $\dot{r}$.

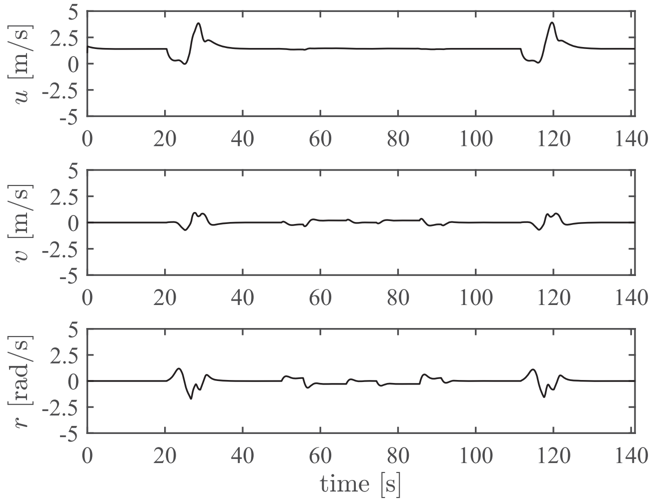


Fig. 25. Case C: Surge u speed, sway v speed, and yaw r speed.

this second case, they used linear regression to compute a time-invariant linear version of system starting from a precomputed trajectory to be passed to the MPC. In contrast, we discretize the continuous nonlinear system (15) using a timestep $T_s = 0.2\,\text{s}$, such that the state from (15) can be defined as

$$\boldsymbol{x}_k = \boldsymbol{x}(t + kT_s) \tag{60}$$

and formulate the full nonlinear MPC problem with an horizon $N = 10$, as follows:

$$J_k^\star(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_{k:k+N-1|k}} \sum_{n=0}^{N} q(\boldsymbol{x}_{k+n|k}, \boldsymbol{u}_{k+n|k}) \tag{61a}$$

$$\text{s.t.}\quad \boldsymbol{x}_{k+n+1|k} = \hat{\boldsymbol{f}}(\boldsymbol{x}_{k+n|k}) + \hat{\boldsymbol{g}}(\boldsymbol{x}_{k+n|k})\boldsymbol{u}_{k+n|k}, \quad n = 0, \ldots, N-1 \tag{61b}$$

$$\boldsymbol{x}_{k+n|k} \in \mathcal{X}, \quad \boldsymbol{u}_{k+n|k} \in \mathcal{U}, \quad n = 0, \ldots, N-1 \tag{61c}$$

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_k \tag{61d}$$

$$\boldsymbol{x}_{k+N|k} \in \mathcal{X}_f \tag{61e}$$

$$\Delta h(\boldsymbol{x}_{k+n|k}, \boldsymbol{u}_{k+n|k}) \geq -\gamma h(\boldsymbol{x}_{k+n|k}), \quad n = 0, \ldots, N-1 \tag{61f}$$

where $\hat{\boldsymbol{f}}(\cdot), \hat{\boldsymbol{g}}(\cdot)$ are the discrete versions of the functions describing the full dynamic system defined in (15); the $\boldsymbol{x}_{a|b}$ represents the state at step a as predicted from step b; and $J_k^\star$ is the optimal cost at time k. In contrast to [48], the stage cost and terminal cost are collapsed into $q(\boldsymbol{x}_k, \boldsymbol{u}_k)$, as the terminal goal at the end of the horizon is already part of the reference trajectory and does not require a different weight than the rest of the trajectory

$$q(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k. \tag{62}$$

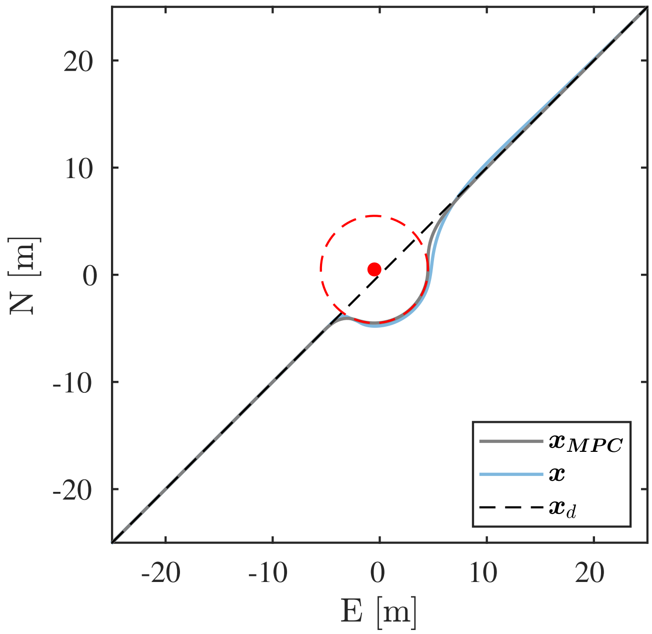


Fig. 26. MPC Case A: Comparison between the desired trajectory $\boldsymbol{x}_d$ in dashed black, the actual trajectory $\boldsymbol{x}_{\text{MPC}}$ computed by the nonlinear MPC in grey, and the actual trajectory computed by the CBF controller in light blue (see Fig. 4) of the USV as it moves around a stationary target. The target is indicated by the red dot at $\boldsymbol{x}_0 = (0.5, -0.5)$ m. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is designated by a red dashed circle.

The weight parameters for (62) are

$$Q = \text{diag}(1000, 1000, 12.5, 14000, 0.01, 0.01) \tag{63a}$$

$$R = 2.9 \times 10^{-4} \cdot I_2. \tag{63b}$$

The system is subject to the state constraint $\mathcal{X}$ and the input constraints $\mathcal{U}$

$$\mathcal{X} = \{\boldsymbol{x}_k \in \mathbb{R}^n : \boldsymbol{x}_{\min} \leq \boldsymbol{x}_k \leq \boldsymbol{x}_{\max}\} \tag{64a}$$

$$\mathcal{U} = \{\boldsymbol{u}_k \in \mathbb{R}^m : \boldsymbol{u}_{\min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max}\} \tag{64b}$$

with lower and upper bounds set to

$$\begin{aligned}\boldsymbol{x}_{\max} &= [100, 100, 3\pi/4, 7, 7, 4]^T, \quad \boldsymbol{x}_{\min} \\ &= [-100, -100, -\pi/4, -7, -7, -4]^T \end{aligned}\tag{65a}$$

$$\boldsymbol{u}_{\max} = [4500, 4500]^T, \quad \boldsymbol{u}_{\min} = [-4500, -4500] \tag{65b}$$

which satisfy the actuator magnitude constraints of the USV's thrusters (see Section V).

As in [48], we choose a quadratic barrier function for the constraint (61f), with $\gamma = 0.4$ and

$$h(\boldsymbol{x}_k) = (x_k - x_0)^2 + (y_k - y_0)^2 - \|\tilde{\boldsymbol{x}}_0\|^2_{\min} \tag{66a}$$

$$\Delta h(\boldsymbol{x}_{k+n|k}) = h(\boldsymbol{x}_{k+n+1|k}) - h(\boldsymbol{x}_{k+n|k}). \tag{66b}$$

We execute the Case A and the Case B simulations using the MPC-CBF formulation (61), which was implemented using the Nonlinear MPC library in MATLAB.

Case A: The results of the simulations for Case A are shown in Figs. 26–30. As shown in Fig. 26, the MPC tends to stay as

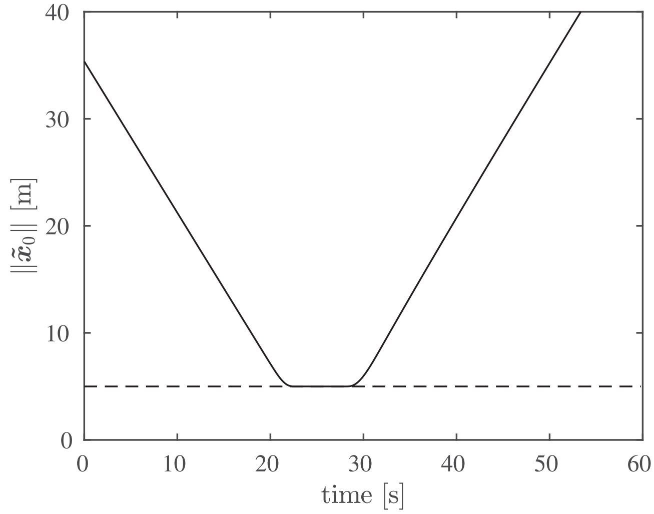


Fig. 27. MPC Case A: Distance between USV and target $\|\tilde{\boldsymbol{x}}_0\|$ as a function of time. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is indicated with a dashed line.

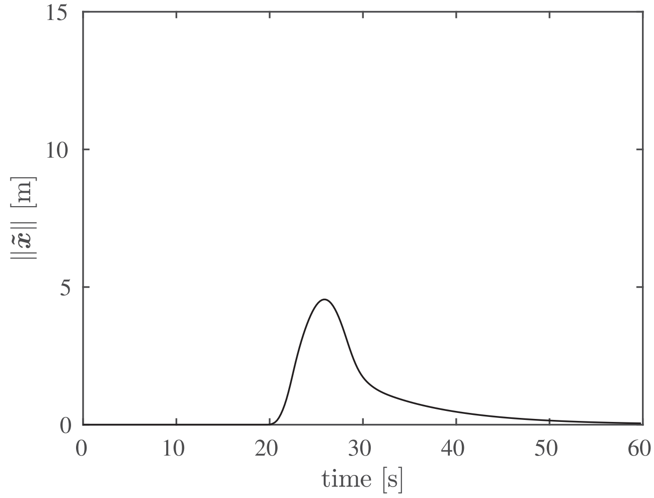


Fig. 28. MPC Case A: Tracking error.

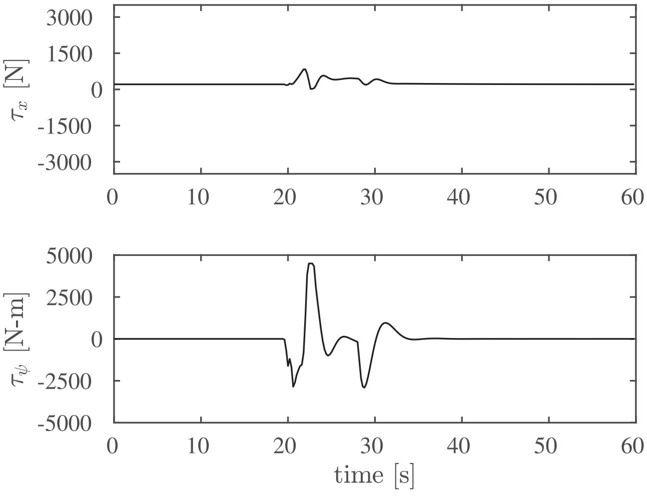


Fig. 29. MPC Case A: Surge force τ_x and yaw moment τ_ψ.

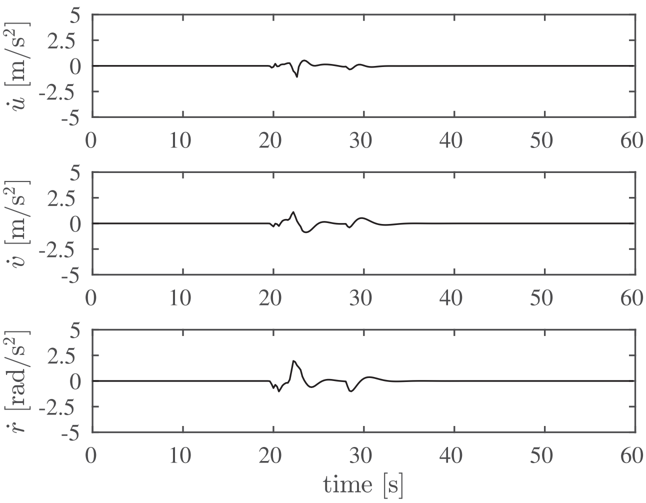


Fig. 30. MPC Case A: Surge acceleration $\dot{u}$, sway acceleration $\dot{v}$, and yaw acceleration $\dot{r}$.

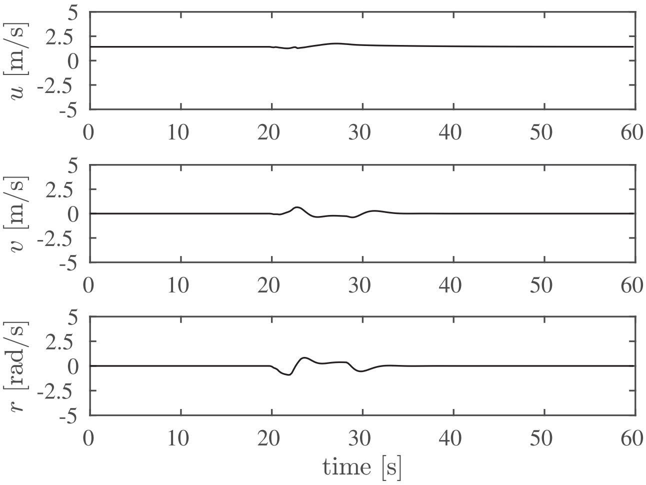


Fig. 31. MPC Case A: Surge u speed, sway v speed, and yaw r speed.

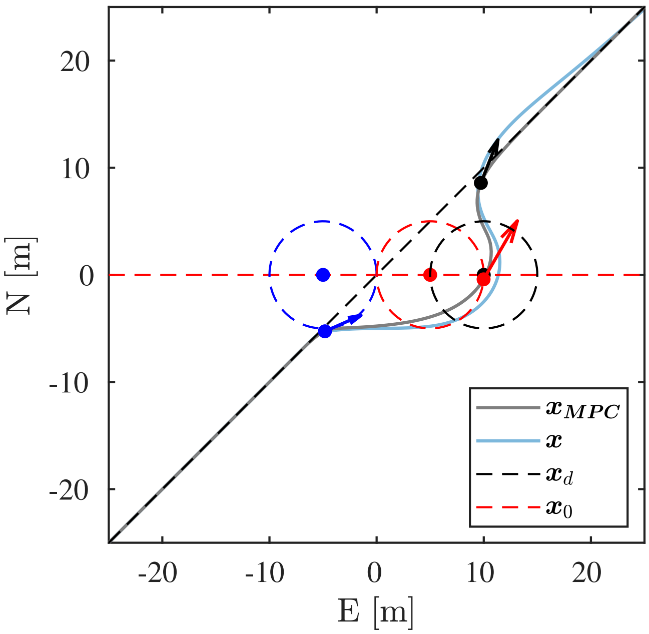


Fig. 32. MPC Case B: Comparison between the desired trajectory $\boldsymbol{x}_d$ in dashed black, the actual trajectory $\boldsymbol{x}_{\mathrm{MPC}}$ from the nonlinear MPC in grey, and the actual trajectory computed by the CBF controller (light blue) of the USV. The USV approaches a moving target, whose trajectory $\boldsymbol{x}_0$ intersects the desired trajectory of the USV at time $t = 25$ s, when both the USV and target would be located at $(x, y) = (0, 0)$ m without activation of the safety-critical controller. The blue, red, and block dots enclosed by dashed circles of corresponding colors plotted along $\boldsymbol{x}_0$ indicate the positions of the target and unsafe set at times $t = 20$ s, $t = 25$ s, and $t = 35$ s, respectively. The dots of corresponding color plotted along $\boldsymbol{x}$ represent the position of the USV at the same times. The orientation and length of the arrows indicate the heading angle and relative speed of the USV.

close as possible to the unsafe boundary to minimize the tracking error with the reference trajectory. This is reflected at the more aggressive yaw moment applied at time 28.8 s to approach the trajectory line at approx 31 s and the longer period spent on the safety margin, as shown in Fig. 27.

The tracking error from Fig. 28 shows a maximum error equal or below 5 m (equivalent the radius of the unsafe area). Even though after 31 s the vessel is already on the trajectory line, the error $\|\tilde{\boldsymbol{x}}\|$ is nonzero and decays to 0.1 m in 22.6 s, slightly slower than the proposed CBF controller. This is because the error it is measured as the difference between the current position $\boldsymbol{x}$ and the desired trajectory position $\boldsymbol{x}_d$, which lies on a straight line. Therefore, the MPC controller tries to compensate for the time previously lost while circumnavigating the unsafe area, but the high weight on the desired forward velocity u (compared to the one on the positions x, y) limits accelerations after the USV is past the unsafe area (see Fig. 30), so as to keep a close tracking of the desired value for u (see Fig. 31).

It is worth noting that with a lower weight on u, the MPC controller would find solutions that stop before the unsafe boundary (because initially the cost would be lower than turning), thereby accumulating a position error that eventually makes the problem unfeasible. Instead, a high cost on the velocity error forces the controller to keep the USV moving, at the cost of making the tracking error converge slower after the unsafe area is passed.

Case B: The simulation results for Case B using the MPC are shown in Figs. 32–36. As for the static case, the strategy adopted by the MPC to minimize the position error makes the USV turn left as early as possible to stay closer to the desired trajectory. This is visible in Fig. 32 at time 25 s (the red dot and arrow), when the USV is at the boundary of the unsafe area while moving North at the same time. The permanence on the boundary to minimize the tracking error is also visible in Fig. 33 where the USV is at 5 m from the center of the target for more than 10 s. This is made possible by the horizon of 2 s given to the MPC, which can therefore predict the locations of the unsafe area at the cost of higher computational complexity. In contrast, the proposed CBF controller commands the USV to move eastwards slowly together with the unsafe area and turn North only after the unsafe area has completely moved past the USV. In this case, the maximum error from the MPC is 7.62 m (see Fig. 34) which is approximately 20% less compared to the CBF approach, at the expense of a higher computational complexity for the MPC. This

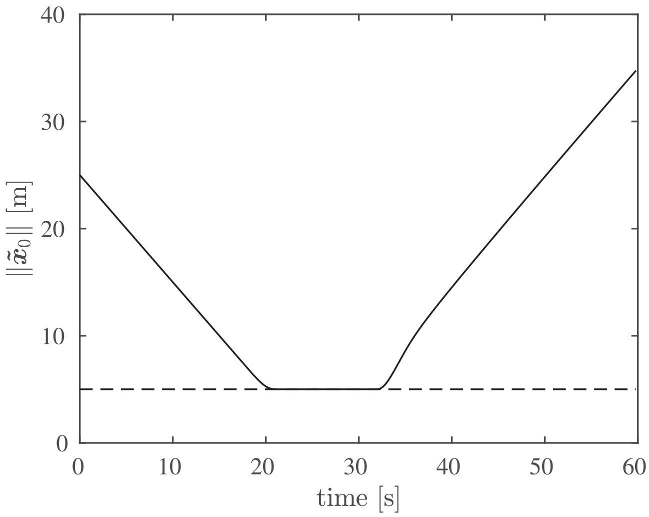


Fig. 33. MPC Case B: Distance between USV and target $\|\tilde{\boldsymbol{x}}_0\|$ as a function of time. The boundary of the unsafe set $\|\tilde{\boldsymbol{x}}_0\|_{\min}$ is indicated with a dashed line.

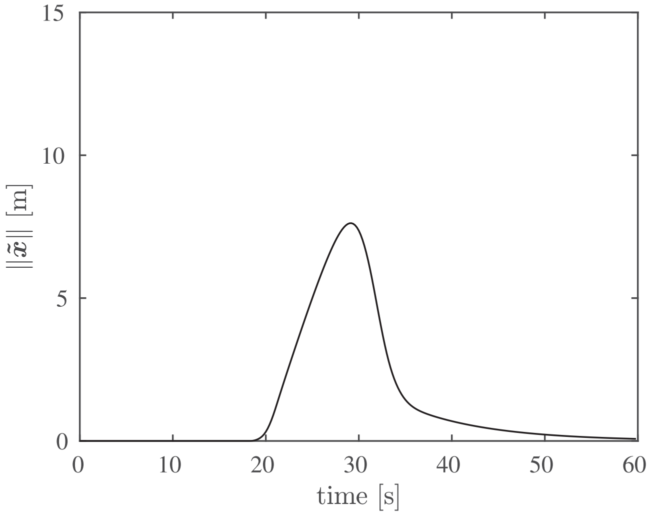


Fig. 34. MPC Case B: Tracking error.

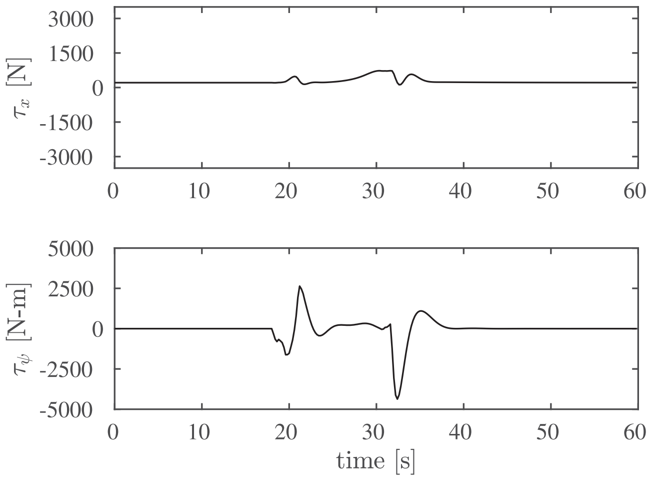


Fig. 35. MPC Case B: Surge force τ_x and yaw moment τ_ψ.

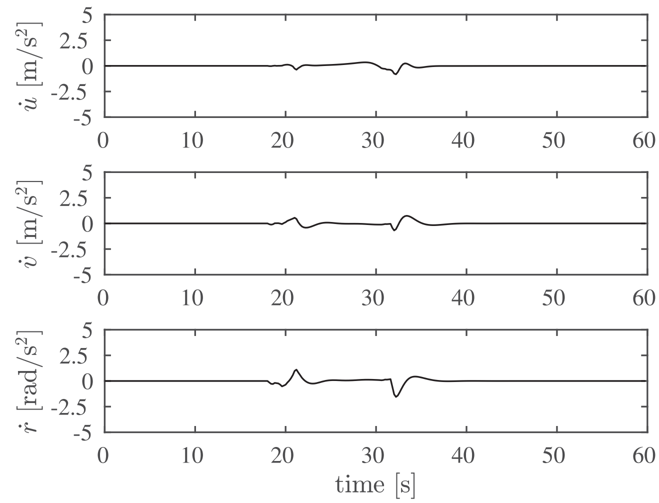


Fig. 36. MPC Case B: Surge acceleration $\dot{u}$, sway acceleration $\dot{v}$, and yaw acceleration $\dot{r}$.

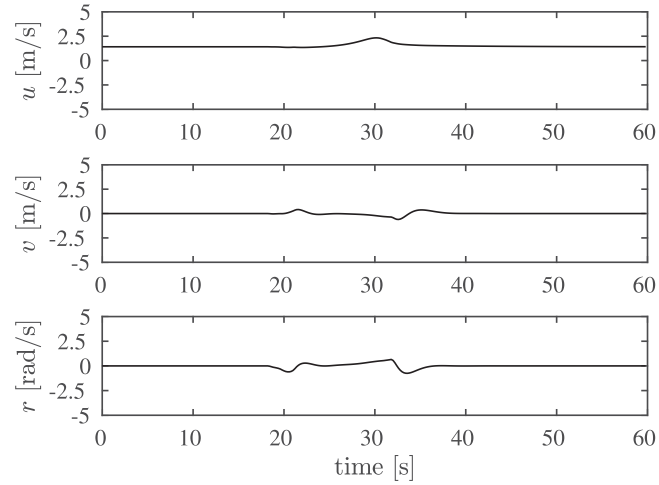


Fig. 37. MPC Case B: Surge u speed, sway v speed, and yaw r speed.

translates in a total execution time of 163 s for a 30 s simulated experiment for the MPC, against an execution time of less than 1 s for the CBF controller. Both simulations were run on the same customer grade laptop computer. As shown in Fig. 35, the control actions are generally smooth, with only two peaks in the yaw moment at time 20 s when the USV encounters the boundary of the unsafe set for the first time, and at time 32 s when the target has moved enough to let the USV recover the trajectory line (cf. Fig. 33). This is also reflected by the accelerations in Fig. 36. As mentioned previously for Case A, the high weight on the desired surge speed makes the USV track u_0 very closely up to the first encounter with the unsafe boundary, after which it increases up to 2.32 ms^{-1} at time 30 s when the USV is about to cross the $\mathrm{N} = 0$ position, and then slowly converges to u_0 to make up for desired positions ahead in time along the trajectory (see Fig. 37).

In summary, the MPC implementation shows the potential of lower tracking error, especially in the moving target scenario, but the higher computational complexity renders the MPC approach unfeasible in case of low computational power requirements.

VII. Conclusion

In this article, we propose a trajectory tracking safety-critical controller for the closest safe approach of an underactuated USV with nonholonomic dynamic (acceleration) motion constraints to a target of interest. In general, safety-critical control methods require the online, realtime, computation of the control input that satisfies an optimization problem. Here, the control law is based on the use of a relaxed control barrier function and an analytical solution to the optimization problem. The analytical solution is fairly easy to implement and avoids the need to perform online optimization. Further, we propose a modification to the relaxed CBF in [15], which permits the safety critical control input to start acting sooner and more gradually as the USV approaches the boundary of an unsafe region. Simulation results show that when a desired trajectory is intentionally planned so that it passes through the unsafe set surrounding a stationary or moving target, the USV will approach the target and move along the boundary of the unsafe set, before passing the target and returning to follow the parts of the desired trajectory outside of the unsafe area. In the simulations, it is observed that the yaw moment actuator effort generated when the USV resumes trajectory tracking after passing the target is saturated. This occurs because the tracking error builds as the USV moves along the boundary of the unsafe set and can become quite large.

The Backstepping Method employed in this work leads to a nonlinear proportional-derivative controller. Such controllers can be *robustified*, as required for specific applications, by adding *nonlinear damping* terms, or augmenting the the system with an integral control term [51]. However, designing a controller to handle a very large range of operating conditions without retuning generally requires the use of robust control techniques, such as higher order sliding mode control, or adaptive control. Since backstepping is typically used as the first step in the design of such controllers, the present work opens up the possibility of using the analytical, and computationally lightweight, CBF-based approach proposed here in combination with more advanced robust trajectory tracking controllers, as well as with robust input-to-state safe CBF approaches [52], [53], which can generally guarantee stability in a greater variety of operating conditions.

Future work will explore how to reduce actuator saturation when the safety-critical controller is deactivated, as well as how to extend the proposed approach to handle model uncertainty, exogenous disturbances, and multiple targets.

Acknowledgment

The authors would like to thank Dr. Michele Focchi for his kind input and suggestions on the implementation of the MPC comparison controller.

References

[1] M. Kurowski, J. Thal, R. Damerius, H. Korte, and T. Jeinsch, “Automated survey in very shallow water using an unmanned surface vehicle,” *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 146–151, 2019.

[2] M. Bibuli, R. Ferretti, A. Odetti, and T. Cosso, “River survey evolution by means of autonomous surface vehicles,” in *Proc. Int. Workshop Metrol. Sea Learn. Measure Sea Health Parameters*, 2021, pp. 412–417.

[3] K. D. Von Ellenrieder, H. C. Henninger, and S. Licht, “Dynamic modelling and control of a portable USV for bathymetric survey,” in *Global Oceans* 2020: Singapore: IEEE, 2020, pp. 1–7.

[4] D. Madeo, A. Pozzebon, C. Mocenni, and D. Bertoni, “A low-cost unmanned surface vehicle for pervasive water quality monitoring,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1433–1444, Apr. 2020.

[5] W. T. Beshah et al., “IoT based real-time water quality monitoring and visualization system using an autonomous surface vehicle,” in *OCEANS 2021: San Diego–Porto*. Piscataway, NJ, USA: IEEE, 2021, pp. 1–4.

[6] Y. Huang et al., “An autonomous probing system for collecting measurements at depth from small surface vehicles,” in *OCEANS 2021: San Diego–Porto*. Piscataway, NJ, USA: IEEE, 2021, pp. 1–6.

[7] J. Fan, Y. Li, Y. Liao, T. Ma, Y. Ge, and Z. Wang, “A formation reconfiguration method for multiple unmanned surface vehicles executing target interception missions,” *Appl. Ocean Res.*, vol. 104, 2020, Art. no. 102359.

[8] B. Du, B. Lin, C. Zhang, B. Dong, and W. Zhang, “Safe deep reinforcement learning-based adaptive control for USV interception mission,” *Ocean Eng.*, vol. 246, 2022, Art. no. 110477.

[9] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” *IFAC Proc. Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[10] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *Proc. 18th Eur. Control Conf.*, 2019, pp. 3420–3431.

[11] E. H. Thyri, E. A. Basso, M. Breivik, K. Y. Pettersen, R. Skjetne, and A. M. Lekkas, “Reactive collision avoidance for ASVs based on control barrier functions,” in *Proc. IEEE Conf. Control Technol. Appl.*, 2020, pp. 380–387.

[12] M. Marley, R. Skjetne, and A. R. Teel, “Synergistic control barrier functions with application to obstacle avoidance for nonholonomic vehicles,” in *Proc. Amer. Control Conf.*, 2021, pp. 243–249.

[13] J. Matouš, E. A. Basso, E. H. Thyri, and K. Y. Pettersen, “Unifying reactive collision avoidance and control allocation for multi-vehicle systems,” in *Proc. IEEE Conf. Control Technol. Appl.*, 2021, pp. 76–81.

[14] E. A. Basso, E. H. Thyri, K. Y. Pettersen, M. Breivik, and R. Skjetne, “Safety-critical control of autonomous surface vehicles in the presence of ocean currents,” in *Proc. IEEE Conf. Control Technol. Appl.*, 2020, pp. 396–403.

[15] M. Igarashi and H. Nakamura, “Collision avoidance assist control for two-wheel vehicle robots by control barrier function,” in *Proc. Int. Autom. Control Conf.*, 2018, pp. 1–6.

[16] M. Igarashi, I. Tezuka, and H. Nakamura, “Time-varying control barrier function and its application to environment-adaptive human assist control,” *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 735–740, 2019.

[17] I. Tezuka and H. Nakamura, “Time-varying obstacle avoidance by using high-gain observer and input-to-state constraint safe control barrier function,” *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 391–396, 2020.

[18] S. Fukuda, Y. Satoh, and O. Sakata, “Trajectory-tracking control considering obstacle avoidance by using control barrier function,” in *Proc. Int. Autom. Control Conf.*, 2020, pp. 1–6.

[19] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[20] K. D. von Ellenrieder, *Control of Marine Vehicles*. Berlin, Germany: Springer, 2021.

[21] L. Elkins, D. Sellers, and W. R. Monach, “The autonomous maritime navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles,” *J. Field Robot.*, vol. 27, no. 6, pp. 790–818, 2010.

[22] I. R. Bertaska et al., “Experimental evaluation of automatically-generated behaviors for USV operations,” *Ocean Eng.*, vol. 106, pp. 496–514, 2015.

[23] W. Wang et al., “Roboat II: A novel autonomous surface vessel for urban environments,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 1740–1747.

[24] W. Yu, J. Wang, G. Tang, M. Li, and Y. Qiao, “Dual-attention-based optical terminal guidance for the recovery of unmanned surface vehicles,” *Ocean Eng.*, vol. 239, 2021, Art. no. 109852.

[25] B. C. Shah et al., “Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic,” *Auton. Robots*, vol. 40, no. 7, pp. 1139–1163, 2016.

[26] P. Rajendran, T. Moscicki, J. Wampler, B. C. Shah, K. von Ellenrieder, and S. K. Gupta, “Wave-aware trajectory planning for unmanned surface vehicles operating in congested environments,” in *Proc. IEEE Int. Symp. Safety, Secur. Rescue Robot.*, 2018, pp. 1–7.

[27] B. C. Shah and S. K. Gupta, “Long-distance path planning for unmanned surface vehicles in complex marine environment,” *IEEE J. Ocean. Eng.*, vol. 45, no. 3, pp. 813–830, Jul. 2019.

[28] L. Hu et al., “A multiobjective optimization approach for COLREGs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1167–1179, Mar. 2020.

[29] G. Li, H. P. Hildre, and H. Zhang, “Toward time-optimal trajectory planning for autonomous ship maneuvering in close-range encounters,” *IEEE J. Ocean. Eng.*, vol. 45, no. 4, pp. 1219–1234, Oct. 2020.

[30] 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 Eng.*, vol. 182, pp. 489–498, 2019.

[31] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, “The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations,”,” *Ocean Eng.*, vol. 223, 2021, Art. no. 108709.

[32] S. Han, L. Wang, Y. Wang, and H. He, “An efficient motion planning based on grid map: Predicted trajectory approach with global path guiding,” *Ocean Eng.*, vol. 238, 2021, Art. no. 109696.

[33] C. Liang, X. Zhang, Y. Watanabe, and Y. Deng, “Autonomous collision avoidance of unmanned surface vehicles based on improved a star and minimum course alteration algorithms,” *Appl. Ocean Res.*, vol. 113, 2021, Art. no. 102755.

[34] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, “Spatio-temporal vessel trajectory clustering based on data mapping and density,” *IEEE Access*, vol. 6, pp. 58939–58954, 2018.

[35] A. J. Sinisterra, M. R. Dhanak, and K. von Ellenrieder, “Stereovision-based target tracking system for USV operations,” *Ocean Eng.*, vol. 133, pp. 197–214, 2017.

[36] J. Muhovič, B. Bovcon, M. Kristan, and J. Perš, “Obstacle tracking for unmanned surface vessels using 3-D point cloud,” *IEEE J. Ocean. Eng.*, vol. 45, no. 3, pp. 786–798, Jul. 2020.

[37] Z. Zhou, S. Yu, and K. Liu, “A real-time algorithm for visual detection of high-speed unmanned surface vehicle based on deep learning,” in *Proc. IEEE Int. Conf. Signal Inf. Data Process.*, 2019, pp. 1–5.

[38] K. Kim, J. Kim, and J. Kim, “Robust data association for multi-object detection in maritime environments using camera and radar measurements,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5865–5872, Jul. 2021.

[39] D. Liu, J. Zhang, J. Jin, Y. Dai, and L. Li, “A new approach of obstacle fusion detection for unmanned surface vehicle using Dempster-Shafer evidence theory,” *Appl. Ocean Res.*, vol. 119, 2022, Art. no. 103016.

[40] H. Ashrafiuon, S. Nersesov, and G. Clayton, “Trajectory tracking control of planar underactuated vehicles,” *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1959–1965, Apr. 2017.

[41] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Amsterdam, Netherlands: Elsevier, 2013.

[42] S. Boyd and L. Vandenberghe, *Convex Optimization*, 7th Ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[43] D. Swaroop, J. Gerdes, P. P. Yip, and J. K. Hedrick, “Dynamic surface control of nonlinear systems,” in *Proc. Amer. Control Conf.*, 1997, pp. 3028–3034.

[44] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, “Dynamic surface control for a class of nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 45, no. 10, pp. 1893–1899, Oct. 2000.

[45] Y. Pan and H. Yu, “Dynamic surface control via singular perturbation analysis,” *Automatica*, vol. 57, pp. 29–33, 2015.

[46] K. D. von Ellenrieder, “Dynamic surface control of trajectory tracking marine vehicles with actuator magnitude and rate limits,” *Automatica*, vol. 105, pp. 433–442, 2019.

[47] M. Marley, R. Skjetne, E. Basso, and A. R. Teel, “Maneuvering with safety guarantees using control barrier functions,” *IFAC-PapersOnLine*, vol. 54, no. 16, pp. 370–377, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896321015202

[48] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *Proc. Amer. Control Conf.*, 2021, pp. 3882–3889.

[49] I. Bertaska and K. von Ellenrieder, “Experimental evaluation of supervisory switching control for unmanned surface vehicles,” *IEEE J. Ocean. Eng.*, vol. 44, no. 1, pp. 7–28, Jan. 2019.

[50] E. I. Sarda, H. Qu, I. R. Bertaska, and K. D. von Ellenrieder, “Station-keeping control of an unmanned surface vehicle exposed to current and wind disturbances,” *Ocean Eng.*, vol. 127, pp. 305–324, 2016.

[51] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ, USA: Wiley, 2011.

[52] S. Kolathaya and A. D. Ames, “Input-to-state safety with control barrier functions,” *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 108–113, Jan. 2019.

[53] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames, “Safe controller synthesis with tunable input-to-state safe control barrier functions,” *IEEE Control Syst. Lett.*, vol. 6, pp. 908–913, 2021.

Karl D. von Ellenrieder (Senior Member, IEEE) received the B.S. degree in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1990, and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 1992 and 1998, respectively.

From 1998 to 2002, he was a Research Fellow with Monash University, Melbourne, VIC Australia, and from 2003 to 2016, he was a Professor of ocean engineering with Florida Atlantic University, Boca Raton, FL, USA.

Since 2016, he has been a Professor of automation with the Faculty of Engineering, with the Libera Universitá di Bolzano, BZ, Italy. His research interests include topics related to the control of uncrewed vehicle systems.

Dr. von Ellenrieder is a Member of the IEEE Oceanic Engineering Society and the IEEE Control Systems Society.

Marco Camurri (Member, IEEE) received the B.Eng. and M.Eng. degrees in computer engineering from the University of Modena and Reggio Emilia, Modena, Italy, in 2009 and 2012, respectively, and the Ph.D. degree in bioengineering and robotics from the University of Genoa and the Istituto Italiano di Tecnologia (IIT), Genoa, Italy, in 2017.

After one year as Postdoc with IIT, from 2018 to 2022, he was Postdoc/Senior Research Associate with the Oxford Robotics Institute, University of Oxford, Oxford, U.K. Since 2023, he is an RTD (equivalent to Associate Professor) with the Faculty of Engineering, with the Libera Universitá di Bolzano, BZ, Italy. His research interests include field robotics, mobile robot perception, model predictive control, autonomous navigation, sensor fusion, and legged robotics.

Open Access funding provided by ‘Libera Università di Bolzano’ within the CRUI CARE Agreement