drones

MDPI

Article

Safety-Critical Trajectory Tracking Control with Safety-Enhanced Reinforcement Learning for Autonomous Underwater Vehicle

Tianli Li [1,2], Jiaming Tao [1,2], Yu Hu [3], Shiyu Chen [3], Yue Wei [3,*] and Bo Zhang [1,2,3,*]

[1] The College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China; tl.li@szu.edu.cn (T.L.); 2210295044@email.szu.edu.cn (J.T.)
[2] The Shenzhen City Joint Laboratory of Autonomous Unmanned Systems and Intelligent Manipulation, Shenzhen 518060, China
[3] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518107, China; huyu@gml.ac.cn (Y.H.); chenshiyu@gml.ac.cn (S.C.)
* Correspondence: weiyue@gml.ac.cn (Y.W.); zhangbo@szu.edu.cn (B.Z.); Tel.: +86-135-21878067 (Y.W.); +86-755-26929781 (B.Z.)

Abstract: This paper investigates a novel reinforcement learning (RL)-based quadratic programming (QP) method for the safety-critical trajectory tracking control of autonomous underwater vehicles (AUVs). The proposed approach addresses the substantial challenge posed by model uncertainty, which may hinder the safety and performance of AUVs operating in complex underwater environments. The RL framework can learn the inherent model uncertainties that affect the constraints in Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs). These learned uncertainties are subsequently integrated for formulating a novel RL-CBF-CLF Quadratic Programming (RL-CBF-CLF-QP) controller. Corresponding simulations are demonstrated under diverse trajectory tracking scenarios with high levels of model uncertainties. The simulation results show that the proposed RL-CBF-CLF-QP controller can significantly improve the safety and accuracy of the AUV's tracking performance.

Keywords: CBF; CLF; AUV; safety-critical control

check for updates

Academic Editors: Jimin Hwang, Neil Bose and Shuangshuang Fan

Received: 6 December 2024
Revised: 12 January 2025
Accepted: 13 January 2025
Published: 16 January 2025

Citation: Li, T.; Tao, J.; Yu, H.; Chen, S.; Wei, Y.; Zhang, B. Safety-Critical Trajectory Tracking Control with Safety-Enhanced Reinforcement Learning for Autonomous Underwater Vehicle. *Drones* **2025**, *9*, 65. https://doi.org/10.3390/drones9010065

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Over the past few decades, Autonomous Underwater Vehicles (AUVs) have emerged as vital tools for ocean resource exploration, scientific research, and military missions, establishing them as a prominent research domain in the fields of robotics [1] and ocean engineering [2]. With autonomous navigation and control systems, AUVs are capable of executing a diverse array of underwater tasks, including rescue operations, exploration, surveillance, reconnaissance, ocean mapping, and deep-sea archaeology [3–5].

As regards the motion control of AUVs, traditional control strategies—such as PID control, robust control, and adaptive control—have demonstrated effectiveness in specific contexts [6]. However, these methodologies frequently encounter limitations when operating within complex marine environments [7]. As autonomous systems are increasingly deployed in real-world environments, ensuring they adhere to safety constraints is crucial. Unlike general control tasks, safety-critical control approaches provide formal and verifiable guarantees to ensure that autonomous systems operate within predefined safety constraints, preventing catastrophic failures and maintaining public confidence. In light of safety and stability concerns, Control Barrier Functions (CBFs) [8] and Control Lyapunov Functions (CLFs) [9] have garnered significant attention as robust tools in the domain of

nonlinear control, especially within safety-critical systems [10]. The CBF is formulated to prevent the violation of safety constraints by designing a barrier function that ensures that states of the system remain in a special set. The CLF guarantees the stability of the system by optimizing the system's energy function. The combination of CBF and CLF, which ensures both safety and stabilizing control, has been successfully applied in a variety of fields, including self-driving cars, robots, and drones. Galloway et al. [11] and Ames and Powell [12] have demonstrated that constrained CLF-based quadratic programming (CLF-QP) can be solved online to perform motion and manipulation tasks. In [8], the CBF has been combined with CLF-QP as the CBF-CLF-QP method to efficiently handle safety constraints in real time. In [13], a quadratic program integrating Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs) is proposed to ensure the safe navigation of incomplete mobile robots in the presence of moving obstacles.

However, the effectiveness of methodologies based on CLFs and CBFs is dependent on the availability of precise system models. The highly nonlinear dynamics of AUVs and environmental uncertainties complicate the attainment of accurate nominal models, potentially leading to degraded control performance or unsafe operations. Under conditions of limited computational resources, conventional control methods have proposed a number of solutions. Observer-based control methods with advantages such as computational efficiency and ease of deployment have gained the attention of researchers in AUV control [14]. In [15], nonlinear perturbation observers are employed to mitigate the linearized process error and uncertainty in a 6-DOF AUV model. However, these methods often struggle with accurately estimating uncertainty bounds, prompting researchers to explore data-driven techniques to increase the capabilities of conventional control approaches [16,17]. The utilization and enhancement of Deep Reinforcement Learning (DRL) for the purpose of addressing the control issue of trajectory tracking of AUVs was put forward in [18]. Additionally, DRL-based control methods have enabled the control of AUVs in 3D space and dynamic environments that vary over time [19,20]. The effectiveness of these DRL-based algorithms has been validated through AUV tracking control simulations. In [21], a deep learning algorithm called depth deterministic policy gradient (DDPG) is designed for UUV to track the trajectory of the target. Table 1 shows a comparison between the different control methods in AUV motion control.

Table 1. Comparison of AUV motion control methods.

Method	Advantages	Disadvantages	References
Traditional Control (PID)	Well-established, simple to implement	Struggles with complex, nonlinear dynamics	[6]
Nonlinear Control (Observer-Based, Robust, Adaptive)	Addresses uncertainty and model error	Difficulty estimating uncertainty bounds	[14,15]
Data-Driven Control (DRL, Neural Network)	Handles dynamic, complex environments, model-free	High computational cost, need for large amounts of data	[18–21]

In recent years, data-driven approaches have been proposed to address model uncertainty in control problems, with Reinforcement Learning (RL) emerging as an effective method for managing such uncertainties [22]. Ref. [23] introduces a machine learning framework that incorporates CBFs to mitigate the effects of model uncertainty on system safety, optimizing control inputs through an RL framework. Ref. [24] presents a Gaussian Process (GP) model designed to learn high-confidence uncertainty bounds, integrating

it into a multi-agent CBF framework to effectively tackle prediction uncertainties from heterogeneous agent dynamics in real-world environments. Ref. [25] proposes a framework that synergizes RL with CBFs and CLFs, enhancing safety and stability by learning system uncertainties. The GP-CBF-CLF framework addresses model uncertainty by deriving probabilistic bounds on its impact on CBF and CLF dynamics, which are then employed to construct safety and stability probability constraints [26]. Ref. [27] proposes a novel Gaussian CBF for safety control, leveraging GP to flexibly estimate arbitrary hyperparameters and learn the kernel function for constructing the CBF.

In light of the review mentioned above, it is clear that the integration of data-driven and CBF methodologies holds immense research value in the context of safety-critical tracking control of AUVs operating in uncertain environments. Given the strong uncertainties and complex dynamics of AUVs, this paper proposes a safety-critical control framework that integrates Reinforcement Learning with Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs). The contributions of this paper are listed as follows:

(1) A safety-enhanced AUV trajectory tracking problem is investigated in this paper, which considers strong uncertainties of the environments and complex dynamics of AUVs simultaneously. The stability and safety constraints are then designed based on the nominal AUV model to ensure the reliability of the proposed tracking trajectory.

(2) Based on CBF and CLF methods, a novel RL-based controller is proposed for AUV trajectory tracking. The RL network is introduced to learn model uncertainties. The learned uncertainties are then incorporated into the CBF and CLF to create new constraints for optimizing control inputs. Compared with traditional methods [28–31], the proposed safety-enhanced controller demonstrates improved robustness, allowing AUVs to achieve safe and reliable motion control in highly uncertain underwater environments.

The rest of the paper is organized as follows: In Section 2, the kinematic model of the AUV is presented. Section 3 shows the RL-CBF-CLF-QP-based controller in this paper for the AUV trajectory tracking task. In Section 4, simulations show the effectiveness of our proposed safety-enhanced controller in different uncertain underwater environments. Finally, Section 5 concludes this paper.

2. Modeling of Dynamics for AUV

This section is dedicated to the modeling of the kinematics and dynamics of an underwater autonomous vehicle. The robot model and all parameters utilized in this paper are derived from the open-source underwater robot BlueROV2. In order to describe the kinematic model of the AUV, the attachment coordinate system O_B and the inertial reference frame O_I are defined as shown in Figure 1.

The vector $\boldsymbol{\tau} = [X, Y, Z, K, M, N]^T$ represents the system's input force and moment vector. The position state vector $\eta = [x_r, y_r, z_r, \phi_r, \theta_r, \psi_r]^T$ describes the position and orientation of the vehicle in the inertial frame. The velocity state vector $\boldsymbol{v} = [u, v, w, p, q, r]^T$ represents the linear and angular velocities in the body-fixed frame.

The structural design of BlueROV2 allows for precise regulation of its center of gravity and center of buoyancy, ensuring stability in roll and pitch. This design enables the vehicle to autonomously return to an equilibrium state. Consequently, the state vector can be reduced to four degrees of freedom when deriving the dynamics model. Given the decoupled depth control provided by the thruster configuration of BlueROV2, the state dimension can be further simplified for analytical purposes to three degrees of freedom: the position state vector $\eta = [x_r, y_r, \psi_r]^T$, the velocity vector $\boldsymbol{v} = [u, v, r]^T$, and the corresponding force vector $\boldsymbol{\tau} = [X, Y, N]^T$.

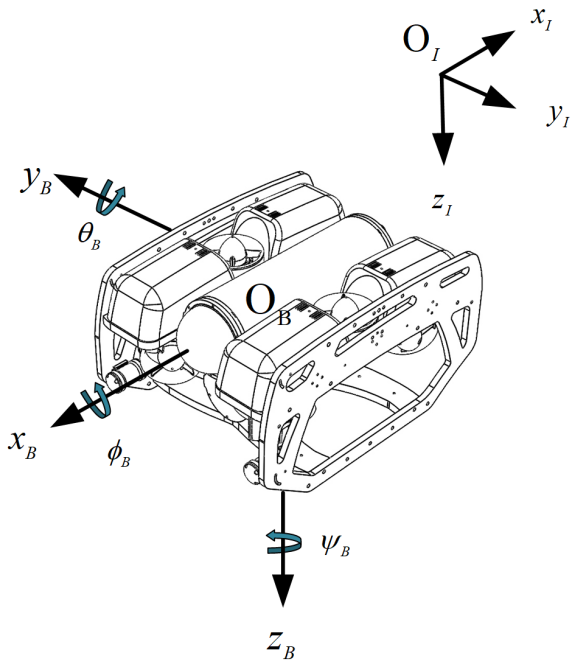


Figure 1. Coordinate frames. A body-fixed frame O_B moves with respect to an inertial frame fixed to the earth O_I .

Moreover, the generalized force vector is a combination of the thrust produced by BlueROV2's six thrusters. The input of the six thrusters is defined as $\boldsymbol{u} = [u_1, u_2, u_3, u_4, u_5, u_6]^T$. It is related to the generalized force vector via the thrust assignment matrix $\mathbf{B}_\tau$, i.e., $\tau = \mathbf{B}_\tau \boldsymbol{u}$. The matrix $\mathbf{B}_\tau$ is a constant matrix determined by the orientation of each thruster related to the chassis, which means that the orientation of BlueROV2's thrusters is fixed.

In the ROV dynamics model, the following forces are considered: hydrodynamic forces, generalized inertial forces, gravity, buoyancy, and forces generated by the thrusters. As proposed in [32], these forces are described as the following formalized kinetic equations:

$$\begin{aligned} M\dot{\boldsymbol{v}} + C(\boldsymbol{v})\boldsymbol{v} + D(\boldsymbol{v})\boldsymbol{v} + g(\eta) &= \mathbf{B}_\tau \boldsymbol{u}, \\ \dot{\eta} &= \mathbf{R}(\eta)\boldsymbol{v}, \end{aligned} \tag{1}$$

where the matrix $\mathbf{R}(\eta)$ is the kinematic transformation matrix between the body-fixed frame and the inertial frame with

$$\mathbf{R}(\eta) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The matrix M in Equation (1) represents the inertia matrix including the added mass, which can be approximated as a diagonal: $M = \mathrm{diag}(M_x, M_y, M_\psi)$. The matrices $M_x = m - X_{\dot{u}}$, $M_y = m - Y_{\dot{v}}$, and $M_\psi = I_z - N_{\dot{r}}$ denote the inertial terms including the added mass. The matrix $C(v)$ represents the Coriolis force matrix, which reflects the Coriolis forces induced by the motion of the rigid body and additional mass effects as

$$C(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & -M_y v \\ 0 & 0 & M_x u \\ M_y v & -M_x u & 0 \end{bmatrix}. \tag{3}$$

The matrix $D(\boldsymbol{v})$ represents the hydrodynamic damping matrix with $D(\boldsymbol{v}) = \text{diag}(X_u, Y_v, N_r) + \text{diag}(D_u|u|, D_v|v|, D_r|r|)$, where X_u,Y_v, and N_r represent the linear damping terms, and D_u, D_v, and D_r are corresponding quadratic drag terms. The symbol $g(\eta)$ denotes the constant force vector, primarily accounting for the combined buoyancy and gravitational forces acting on the vehicle.

This above formalism provides a comprehensive representation of the dynamic behavior of underwater vehicles, integrating both the linear and nonlinear hydrodynamic effects, external forces, and transformation between coordinate systems. Define the state of the system $x = [\eta^T, \boldsymbol{v}^T]^T$. Consequently, the system state transition equation can be expressed as

$$\dot{x} = \begin{bmatrix} R(\eta)\boldsymbol{v} \\ M^{-1}(C(v)\boldsymbol{v} - D(v)\boldsymbol{v} - g) \end{bmatrix} + \begin{bmatrix} 0_{3\times} \\ M^{-1} \end{bmatrix} \mathbf{B}_\tau \boldsymbol{u}. \tag{4}$$

In accordance with the aforementioned state transfer equations, a concise analysis of the sources of uncertainty in the system model can be conducted. The primary source of uncertainty pertains to the mass matrix, denoted by M. While the mass and moment of inertia are relatively straightforward to ascertain for an AUV, the additional mass of the system often hinges on the form factor of the robot, a dimension that is often challenging to ascertain with precision. Formally, it will be coupled with other uncertainties, thus exerting a more complex effect on the controller. Another significant source of uncertainty is the water drag coefficient of the AUV, which is primarily represented by the linear and quadratic damping coefficients. Errors in quadratic damping coefficients will cause substantial disturbances in the control system when the AUV is operating at high speeds.

Based on the aforementioned equation and model, the underwater motion behavior of BlueROV2 can be systematically characterized, providing a solid theoretical foundation for its autonomous navigation capabilities.

3. The Safety-Enhanced Trajectory Tracking Control Framework

The objective of this section is to present the construction methods associated with the CBF and CLF, with an analysis of the influence of model uncertainty on these methods, and to define the learning objectives and learning strategies of an RL framework. These are then combined to construct a complete controller for AUV safety-critical trajectory tracking tasks with model uncertainty.

3.1. CBF-CLF-QP Problem Formulation

In this subsection, the construction of controller constraints in terms of stability and security is considered. For the concern regarding the stability of the system, the CLF-based method, which is related to cost functions in optimal control problems [33], is introduced in [34] and generalized in [35]. In this part, the method of controlling the tracking of an underwater vehicle constrained by the CLF, based on quadratic programming, is mainly introduced. Following the definition in Section 2, the lateral dynamics of the underwater vehicle's trajectory tracking control system are constructed here. The output $h(x)$ of the system is the tracking error of the system compared with the desired trajectory. Combined with Equation (4), the system model can be described as an affine control system as follows:

$$\begin{cases} \dot{x} = \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{v}} \end{bmatrix} = f(\eta, \boldsymbol{v}) + g(\eta, \boldsymbol{v})\boldsymbol{u} \\ y = (x - x_d)^2 = h(x) \end{cases}. \tag{5}$$

Then the definition of the CLF is presented as follows.

Definition 1 ([34]). *For a given affine control system (5), there is a continuous differentiable function $V(x) : \mathbb{R}^n \to \mathbb{R}$. If there is a control input $\boldsymbol{u} \in \boldsymbol{U}$, it is called an effective CLF. For any $x \in \boldsymbol{X}$ and $\forall t \in I(x_0)$, there exists*

$$\begin{aligned} c_1\|x\|^2 \le V(x) \le c_2\|x\|^2, \\ \inf_{\boldsymbol{u}\in\boldsymbol{U}}\Big[L_fV(x) + L_gV(x)\boldsymbol{u} + \alpha(V(x))\Big] \le 0, \end{aligned} \tag{6}$$

where $c_1 > 0$, $c_2 > c_1$, and α is an extended K-class function.

$L_fV(x) \triangleq \frac{\partial V(x)}{\partial x}f(x)$ and $L_gV(x) \triangleq \frac{\partial V(x)}{\partial x}g(x)$ denote the partial derivatives of the function $V(x)$ along (5). Note that $L_gV(x) \neq 0$ must be satisfied; otherwise, the control input u in (6) disappears. Therefore, the input and output of the system should be linearized. Specifically, a virtual input ν is defined as the second derivative of the system output y as follows:

$$\nu = \ddot{y} = L_f^2h(x) + L_gL_fh(x)\boldsymbol{u}. \tag{7}$$

Moreover, define a state transformation $\Phi : x \to \xi(x)$ with

$$\xi(x) = \left[h(x)^\top, L_fh(x)^\top\right]^\top; \tag{8}$$

then a new linear system is obtained as

$$\dot{\xi} = \begin{bmatrix} 0_{2\times2} & \mathbf{I}_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix}\xi + \begin{bmatrix} 0_{2\times2} \\ \mathbf{I}_{2\times2} \end{bmatrix}\nu = F\xi + G\nu. \tag{9}$$

Ames et al. proposed a control method in [36] to guarantee the exponential stability of the transverse dynamics η with a sufficiently fast convergence rate. It introduces the fast exponential stability control Lyapunov function (RES-CLF), which is defined as follows.

Definition 2 ([36]). *If there exists $\gamma, c1, c2 > 0$, such that $\forall 0 < \varepsilon < 1$ and $\forall \xi \in \mathbb{R}^{mr}$, then the one-parameter family $V_\varepsilon : \mathbb{R}^{mn} \to \mathbb{R}$ of continuous differentiable functions is called the RES-CLF of system (6) with*

$$\begin{aligned} c_1\|\xi\|^2 \le V_\varepsilon(\xi) \le \frac{c_2}{\varepsilon^2}\|\xi\|^2, \\ \dot{V}_\varepsilon(\xi, v) + \frac{\lambda}{\varepsilon}V_\varepsilon(\xi) \le 0. \end{aligned} \tag{10}$$

Based on the input–output linearization treatment of (7) and (9), the original higher-order system (5) can be transformed to a linear system (9) for analyzing the stability of this system. The Lyapunov function $V_\varepsilon = \xi^\top P\xi$ is then constructed, where P is a positive definite matrix obtained by the Riccati equation.

By defining $\bar{f} = F\xi$ and $\bar{g} = G$, the derivative of the RES-CLF of system (6) can be rewritten as

$$\dot{V}_\varepsilon(\xi, \nu) = L_{\bar{f}}V_\varepsilon(\xi) + L_{\bar{g}}V_\varepsilon(\xi)\nu \tag{11}$$

where $L_{\bar{f}}V_\varepsilon(\xi) = \xi^T\left(F^TP_\varepsilon + P_\varepsilon F\right)\xi$ and $L_{\bar{g}}V_\varepsilon(\xi) = 2\xi^TP_\varepsilon G$.

It is worth noting that, in actual control, the feasible virtual control quantity ν should be converted into the system input $\boldsymbol{u}$ as

$$\boldsymbol{u}(x, v) = \boldsymbol{u}^* + \left(L_gL_f^{r-1}h(x)\right)^{-1}\nu, \tag{12}$$

where

$$\boldsymbol{u}^*(x) = -\left(L_gL_f^{r-1}h(x)\right)^{-1}L_f^rh(x). \tag{13}$$

Based on Equations (7) and (11), we subsequently rewrite the derivative of the RES-CLF as

$$\dot{V}_\varepsilon(\xi, \boldsymbol{u}) = L_{\bar{f}} V_\varepsilon(\xi) + L_{\bar{g}} V_\varepsilon(\xi) L_f^2 h(x) + L_{\bar{g}} V_\varepsilon(\xi) L_g L_f h(x) \boldsymbol{u}. \tag{14}$$

Based on the above analysis, a complete CLF-QP problem is obtained as

$$\begin{aligned} \boldsymbol{u}^*(x) &= \arg\min_{\boldsymbol{u} \in \boldsymbol{U}} \tfrac{1}{2} \boldsymbol{u}^T H(x) \boldsymbol{u} \\ \text{s.t.} \quad & \dot{V}_\varepsilon(\xi, \boldsymbol{u}) + \tfrac{\lambda}{\varepsilon} V_\varepsilon(\xi) \leq 0 \end{aligned} \tag{15}$$

Remark 1. *The optimization problem is formulated using the full dynamics of the underwater vehicle, as opposed to the input–output linearized dynamics. One key reason for adopting this approach is that, when the system exhibits model uncertainty, performing an inverse operation such as* (9) *may reintroduce uncertainty into the control process. This situation may undermine the effectiveness of the analysis of model uncertainties within the system. A detailed uncertainty analysis is presented later in the following subsection. Furthermore, by utilizing the full dynamics, the control input* $\boldsymbol{u}$ *can be explicitly incorporated into the cost function of the QP problem. As a comparison, the physical interpretation of the virtual input* v *is ambiguous in the cost function.*

Building upon the previous discussion, the inclusion of security constraints for the controller is considered here. System security can be defined as the frontline invariance of the system state in the security concentration [8].

In the context of underwater robot trajectory tracking systems, the safety of the system is directly related to the distance between the robot and the obstacle and indirectly related to its relative velocity. In this regard, the barrier function may be defined in accordance with the Euclidean distance of the robot from the obstacle. The safe distance, represented by r_s, is dependent upon the size of the obstacle, the dimensions of the robot, and other factors. The variables $x_o(t)$ and $y_o(t)$ represent the obstacle positions. The variable i denotes the number of obstacles in the event of multiple obstructions. Therefore, the barrier function h_{ci} for i is defined as

$$h_{ci} = \sqrt{(x_r - x_o(t))^2 + (y_r - y_o(t))^2} - r_s \geq 0. \tag{16}$$

In this paper, we employ the reciprocal barrier function, as defined in reference [37] with $B(x) = \frac{\gamma}{h_c(x)}$ and $\gamma > 0$. Ensuring the security of system (4) was proposed by Peter Wieland and Frank Allgöwer[38], where $L_g B(x) \neq 0$ is required in CBF conditions. According to the state transition equation of (4), the first-order time derivative of $B(x)$ can be obtained by

$$\dot{B}(x) = L_f B(x) + L_g B(x) \boldsymbol{u}, \tag{17}$$

where $L_f B(x)$ and $L_g B(x)$ are lie derivatives as follows:

$$\begin{aligned} L_f B(x) &= \frac{\partial B}{\partial x} \begin{bmatrix} \mathbf{R}(\psi)\boldsymbol{v} \\ M^{-1}(C\boldsymbol{v} - D\boldsymbol{v} - g) \end{bmatrix}, \\ L_g B(x) &= \frac{\partial B}{\partial x} \begin{bmatrix} 0_{3\times 3} \\ M^{-1} \end{bmatrix} \mathbf{B}_\tau. \end{aligned} \tag{18}$$

Since the inertia matrix M and the thrust distribution matrix B_τ are both constant matrices, according to (17) and (18), it is evident that $L_g B(x) = 0$, which indicates that the relative degree of the system with respect to B is greater than 1. Furthermore, the traditional CBF constraint does not incorporate the control input $\boldsymbol{u}$, rendering the conventional CBF approach ineffective. Therefore, it is necessary to design a CBF that is suitable for systems

with higher relative degrees. The exponential control barrier function (ECBF) introduced in [39] is particularly well suited for more general cases. Specifically, the function $B : \mathbb{R}^m \rightarrow \mathbb{R}$ serves as the ECBF with the relative degree r for the affine system. If there exists a $K \in \mathbb{R}^{1\times r}$ that satisfies

$$\sup_{\boldsymbol{u}\in\boldsymbol{U}}\left[L_f^n B(x) + L_g L_f^{n-1} B(x)\boldsymbol{u} + K\zeta(x)\right] \geq 0 \tag{19}$$

for $\forall x \in \{x \in \mathbb{R}^n \mid B(x) \geq 0\}$ with

$$\zeta = \begin{bmatrix} B(x) \\ \dot{B}(x) \\ \ddot{B}(x) \\ \cdots \\ B^{(r-1)}(x) \end{bmatrix} = \begin{bmatrix} B(x) \\ L_f B(x) \\ L_f^2 B(x) \\ \cdots \\ L_f^{r-1} B(x) \end{bmatrix}, \tag{20}$$

then it guarantees that $B(x_0) \geq 0 \Longrightarrow B(x(t)) \geq 0, \forall t \geq 0$.

The definition and proof of the ECBF is similar to that of the CLF in that it employs an input–output linearization transformation in [39]. The choice of K depends on the stability of the control system after input–output linearization. The pole configuration is often used to compute K, which is not repeated here. Based on the discussion in this subsection and the previous subsection, the following CBF-CLF-QP problem is defined as follows.

$$\begin{array}{ll} & \mathbf{u}^*(x) = \underset{\mathbf{u}=(\boldsymbol{u},\delta)}{\arg\min} \frac{1}{2}\boldsymbol{u}^T H(x)\boldsymbol{u} + p\delta^2 \\ \text{s.t.} & \dot{V}_\varepsilon(\xi,\boldsymbol{u}) + \frac{\lambda}{\varepsilon} V_\varepsilon(\xi) \leq \delta \\ for\ i = 1\cdots n_o & L_f^2 B_i(x) + L_g L_f B_i(x)\boldsymbol{u} \geq -K\zeta \end{array}, \tag{21}$$

where δ is added to the cost function as a slack variable to ensure that safety is prioritized over tracking stability. i and n_o represent the obstacle serial number and the number of obstacles, respectively.

The QP controller is capable of achieving safety-critical tracking control of the AUV by solving the aforementioned QP problem and obtaining a feasible solution within each control step. Subsequently, the feasible solution is employed as a control input to the actuator.

Remark 2. *As shown in* (21) *and the above discussion, a comprehensive QP controller based on the AUV model has been proposed, ensuring the safe execution of the AUV trajectory tracking task. The advantage of the QP-based controller lies in its flexibility to incorporate various constraints to meet practical control requirements. Additionally, it enables the uncertainty analysis of the constraints and the design of appropriate compensation strategies, which is discussed in the next subsection.*

3.2. RL-CBF-CLF-QP Controller Design

In this subsection, a comprehensive examination of the impact of system uncertainty on the designed controller is conducted, along with a detailed explanation of how reinforcement learning can be utilized to compensate for these uncertainties. An in-depth analysis of system uncertainty is presented here first, where it is assumed that the uncertainties inherent in both the real AUV dynamic model and the nominal model can be expressed as follows:

$$\dot{x} = \hat{F}(x) + \Delta F(x) + (\hat{G} + \Delta G)\boldsymbol{u}, \tag{22}$$

where $\hat{F}(x)$ and $\hat{G}$ are nominal model estimates of the system, and $\Delta F(x) = F(x) - \hat{F}(x)$ and $\Delta G = G - \hat{G}$ denote the uncertain terms.

The impact of model uncertainty on the CBF is analyzed here for designing the RL-CBF-CLF-QP problem. Since the model uncertainty affects only the second-order derivative of $B(x)$, i.e., $B(x)$ and $\dot{B}(x)$ do not contain kinetic parameter terms, from (17) and (22), it is obtained that

$$L_f^2 B(x) = \frac{\partial^2 B}{\partial x^2} \cdot (\hat{f} + \Delta f) + \frac{\partial B}{\partial x} \cdot \frac{\partial (\hat{f} + \Delta f)}{\partial x}, \tag{23}$$

$$L_g L_f B(x) = \frac{\partial}{\partial x} \left(\frac{\partial B}{\partial x} \cdot (\hat{f} + \Delta f) \right) \cdot (\hat{g} + \Delta g). \tag{24}$$

By simplifying (23) and (24) as a nominal term and an uncertainty term, the expression of $\ddot{B}(x)$ can be depicted as

$$\ddot{B}(x) = L_{\hat{f}}^2 B(x) + \Delta_1^b(x) + (L_{\hat{g}} L_{\hat{f}} B(x) + \Delta_2^b(x))\boldsymbol{u}, \tag{25}$$

where $\Delta_1^b(x) = \frac{\partial^2 B}{\partial x^2} \cdot \Delta f + \frac{\partial B}{\partial x} \cdot \frac{\partial \Delta f}{\partial x}$ and $\Delta_2^b(x) = \frac{\partial}{\partial x} \left(\frac{\partial B}{\partial x} \cdot \Delta f \right) \cdot (\hat{g} + \Delta g) + \frac{\partial}{\partial x} \left(\frac{\partial B}{\partial x} \cdot \hat{f} \right) \cdot \Delta g$.

An RL framework can be employed to learn these uncertainty terms by constructing an estimator of $\ddot{B}$ as

$$\widetilde{\ddot{B}}(x, \boldsymbol{u}, \theta) = \widehat{\ddot{B}}(x, \boldsymbol{u}) + \alpha_\theta^b + \beta_\theta^b \boldsymbol{u}. \tag{26}$$

Clearly, the objective of the RL framework proposed in this paper is to develop policies α_θ^b and β_θ^b in (26) that mitigate the errors arising from model uncertainty, thereby ensuring that the estimated constraints align with the actual model. They will be used as the output of reinforcement learning, that is, action to achieve an estimate of the above uncertainty, so they have the same dimension as $\Delta_1^b(x)$ and $\Delta_2^b(x)$.

Based on analysis of the influence of uncertainty on CBF constraints, a similar method is also presented to analyze the CLF. Since high-order CLF dynamics depend only on the linear dynamics F and G, the virtual input v in the constraint can be described as

$$v = \widehat{\dot{y}} = L_{\hat{f}}^2 h(x) + L_{\hat{g}} L_{\hat{f}} h(x) \boldsymbol{u} + \Delta_1^y(x) + \Delta_2^y(x) \boldsymbol{u}, \tag{27}$$

where $\Delta_1^y(x)$ and $\Delta_2^y(x)$ can be derived similar to $\Delta_1^b(x)$ and $\Delta_2^b(x)$. Note that $\Delta_1^y(x)$ and $\Delta_2^y(x)$ are not uniform in dimension. In order to unify the dimensions of the vectors, let (27) combine with (11) as

$$\begin{aligned} \dot{V}_\varepsilon(\xi, \boldsymbol{u}) =& L_{\bar{f}} V_\varepsilon(\xi) + L_{\bar{g}} V_\varepsilon(\xi) L_{\hat{f}}^2 h(x) + L_{\bar{g}} V_\varepsilon(\xi) L_{\hat{g}} L_{\hat{f}} h(x) \boldsymbol{u} \\ &+ \underbrace{L_{\bar{g}} V_\varepsilon(\xi) \Delta_1^y(x)}_{=:\Delta_1^v} + \underbrace{L_{\bar{g}} V_\varepsilon(\xi) \Delta_2^y(x)}_{=:\Delta_2^v} \end{aligned} \tag{28}$$

Then an estimator of $\dot{V}_\varepsilon$ is presented as

$$\widetilde{\dot{V}}_\varepsilon(\xi, \boldsymbol{u}, \theta) = \widehat{\dot{V}}_\varepsilon(\xi, \boldsymbol{u}) + \alpha_\theta^v + \beta_\theta^v \boldsymbol{u}. \tag{29}$$

Similarly, α_θ^v and β_θ^v will be used as the outputs of the reinforcement learning and will have the same form as Δ_1^v and Δ_2^v for their estimation.

Based on the analysis above, the final safety-enhanced optimization problem, which includes the learning-based estimator of all uncertainties, is the RL-CBF-CLF-QP problem, which is presented as

$$\begin{aligned} & & \mathbf{u}^*(x) = \underset{\mathbf{u}=(\boldsymbol{u},\delta)}{\arg\min} \tfrac{1}{2} \boldsymbol{u}^T H(x) \boldsymbol{u} + p\delta^2 \\ & \text{s.t.} & \widetilde{\dot{V}}_\varepsilon(\xi, \boldsymbol{u}, \theta) + \tfrac{\lambda}{\varepsilon} V_\varepsilon(\xi) \leq \delta \\ & for\ i = 1 \cdots n_o & \widetilde{\ddot{B}}_i(x, \boldsymbol{u}, \theta) \geq -K\zeta. \end{aligned} \tag{30}$$

The RL framework is applied here to learn the uncertain terms within the CBF and CLF constraints in the QP problem. The reward function of the learning problem is designed with the objective of minimizing the discrepancy between the actual and estimated values of the constraints, which are based on the nominal model in the QP problem (30). Furthermore, an error loss function is defined for the tracking task to ensure that the controller does not deviate excessively from the desired point, thereby avoiding any adverse effects on the training process. Consequently, the time loss function is defined as follows:

$$\begin{aligned} l_{B,\theta} &= \|\ddot{B} - \widetilde{\ddot{B}}(x, \boldsymbol{u}, \theta)\|^2, \\ l_{V,\theta} &= \|\dot{V}_\varepsilon - \widetilde{\dot{V}}_\varepsilon(x, \boldsymbol{u}, \theta)\|^2, \\ l_{E,\eta} &= \|\eta - \eta_d\|^2, \end{aligned} \tag{31}$$

where $\ddot{B}$ and $\dot{V}_\varepsilon$ are the second-order derivatives of the obstacle function designed from the real model and the derivatives of the Lyapunov function of the linear system (16), and $\widetilde{\ddot{B}}$ and $\widetilde{\dot{V}}_\varepsilon$ are the estimates obtained by combining the reinforcement learning to estimate the uncertainty and the nominal model. η and η_d represent the pose and expected pose of AUV, respectively.

The following RL problem is formulated, where the reward for a given state x is defined as the weighted sum of the negative loss function in (31). Additionally, a penalty term $-l_e : \mathbb{R}^n \to \mathbb{R}$ is introduced to penalize task failure (e.g., collisions or timeouts) as

$$R(x, \theta) = -w_v l_{V,\theta} - \sum_{i=1}^{n_o} w_{b,i} l_{B_i,\theta} - w_e l_{E,\eta} - l_e(x). \tag{32}$$

Then the uncertainty learning problem for the RL framework is defined as

$$\begin{aligned} &\max_{\theta} \mathbb{E}_{x_0 \sim X_0} \int_0^T R(x(\tau), \theta) d\tau \\ &\text{s.t. } \dot{x} = f(x) + g(x)\boldsymbol{u}(x) \end{aligned}. \tag{33}$$

Based on the all above analysis, problem formulation, and RL framework construction, the complete framework of the proposed RL-based controller is proposed as Algorithm 1, with the detailed training process of the RL framework. In this algorithm, π_θ represents the actor network with the parameters θ that generates actions a, specifically containing α_θ^v, β_θ^v, α_θ^b, and β_θ^b. The critic networks, with the parameters ϕ_1 and ϕ_2, are denoted as Q_{ϕ_1} and Q_{ϕ_2}. The target networks $\pi_{\theta'}$, $Q_{\phi_1'}$, and $Q_{\phi_2'}$ are soft-updated versions of the main networks, helping to stabilize learning. This proposed algorithm iterates for $\mathcal{M}$ episodes, each containing $\mathcal{K}$ steps. The state of the autonomous underwater vehicle (AUV) is represented by $\mathbf{x}$, with the reference trajectory $\mathbf{x}_d(t)$ guiding its motion. The replay buffer $\mathcal{D}$ stores transitions for experience replay, and delayed updates are applied to the actor network every d steps, following two-delay deep deterministic policy gradient algorithm (TD3) framework [40].

Remark 3. *The full process of the safety-enhanced RL-based AUV trajectory tracking control is depicted in Figure 2. In the framework of this proposed controller, the inputs to the RL network include variables such as the state of the AUVs and the state of surrounding obstacles. The outputs consist of the uncertain parameters within the constraints, such as α_θ^v, β_θ^v, α_θ^b, and β_θ^b. These parameters are integrated with the original constraint terms to compensate for the effects of uncertainty. These parameters are incorporated into the QP constraints derived from the nominal model to estimate the real system's constraint values, ensuring that the generated control input consistently satisfies these constraints.*

Algorithm 1 Training Process of RL-CBF-CLF-QP Controller for AUV Trajectory Tracking

Initialize actor π_θ, critics Q_{ϕ_1}, Q_{ϕ_2}, and their target networks $\pi_{\theta'}$, $Q_{\phi_1'}$, $Q_{\phi_2'}$.
Initialize replay buffer $\mathcal{D}$.
for $i = 1$ to $\mathcal{M}$ **do**
  Randomly select reference trajectory $x_d(t)$ and initialize obstacle parameters Obs.
  Initialize AUV state x.
  for $t = 1$ to $\mathcal{K}$ **do**
    Generate action $a(\alpha_\theta^v, \beta_\theta^v, \alpha_\theta^b, \beta_\theta^b)$ using RL policy π_θ with exploration noise.
    Incorporate a into CBF and CLF constraints according to (26) and (29).
    Solve QP problem (30) to obtain control input u^*.
    Execute control input u^* and update AUV state according to (4).
    Observe new state x'
    Calculate reward r as defined by (32).
    Store transition (x, a, r, x') in replay buffer $\mathcal{D}$.
    Sample a mini-batch from $\mathcal{D}$.
    Update critics Q_{ϕ_1}, Q_{ϕ_2}.
    if t mod $d = 0$ **then**
      Update actor π_θ by the deterministic policy gradient.
      Update target networks: $\pi_{\theta'} \leftarrow \lambda\pi_\theta + (1-\lambda)\pi_{\theta'}$, $Q_{\phi_i'} \leftarrow \lambda Q_{\phi_i} + (1-\lambda)Q_{\phi_i'}$.
    end if
  end for
end for

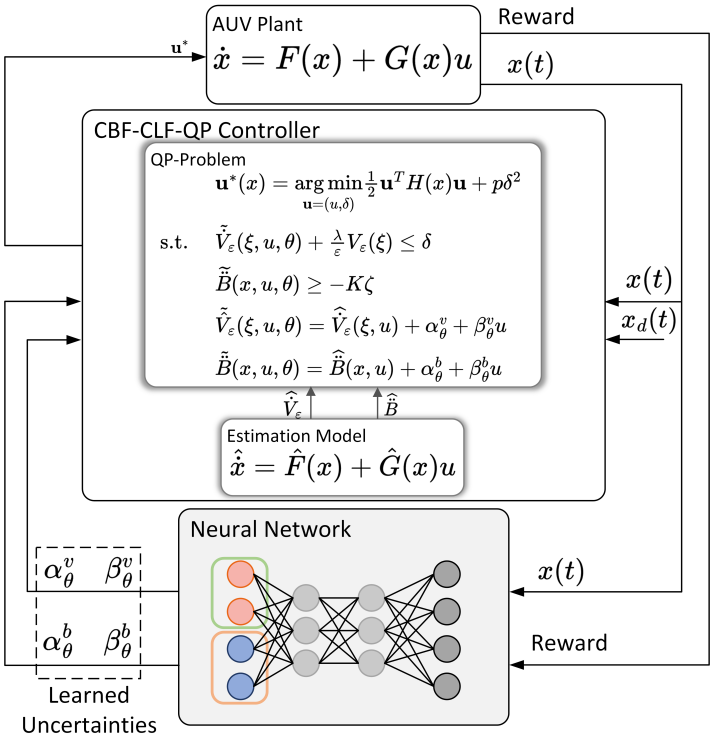


Figure 2. RL-CBF-CLF-QP control framework.

4. Simulation and Discussions

This section is presented to verify the effectiveness of the safety-enhanced RL-CBF-CLF-QP-based controller in the task of AUV trajectory tracking with model uncertainties. Based on the AUV dynamic model described in Section 2, a corresponding reinforcement learning environment for training agents is constructed here.

4.1. 3D Tracking Control Demo

We first performed a simulation demonstration in a 3D scene to show the complete degree of freedom motion of the AUV and the implementation of basic trajectory tracking and safe obstacle avoidance. We built a simple scene and set up three obstacles. In Figure 3, we can see that the QP controller based on the CBF and CLF can complete the obstacle avoidance tracking control in 3D, and from the three views, we can also see that the method achieves good control when model uncertainty is not added.

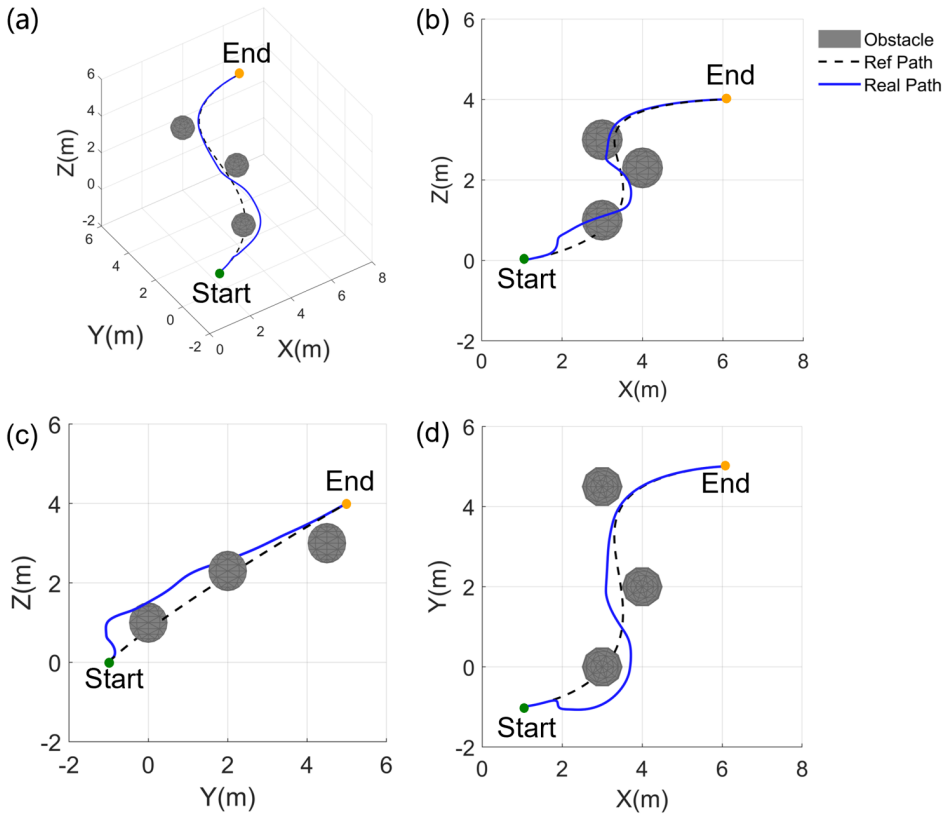


Figure 3. 3D obstacle avoidance tracking control scene simulation: (**a**) 3D trajectory, (**b**) front view, (**c**) side view, and (**d**) top view.

Further, we add uncertainty to the nominal model to test the control effect of the proposed method under uncertainty conditions. It is worth noting that for computational and presentation purposes, we focus on the obstacle avoidance and tracking behaviors of the AUV in the 2D plane and analyze the results in the subsequent methods.

4.2. Application to Autonomous Underwater Vehicles

4.2.1. Simulation Settings

The proposed control framework is applied to three different scenarios of AUV tracking control and compared with previous methods. The open-source nature of BlueROV2 facilitates convenient access to the diverse range of model parameters associated with BlueROV2, as illustrated in Table 2, with specific figures drawn from the literature [41].

Given the inaccuracy of model parameters in real-world scenarios, a simulation task is conducted to evaluate the impact of model uncertainty on the controller design. Specifically, errors are introduced to key system parameters, including the inertia matrix, quadratic water resistance terms, and linear water resistance coefficients. The output function, $h(x)$, is defined according to Equation (5), with the system required to avoid collisions with obstacles while maintaining effective tracking performance.

- In Scenario 1, a basic environment with a single static obstacle is considered. Various degrees and types of uncertainty are introduced to preliminarily verify the feasibility of the proposed control framework.
- In Scenario 2, multiple obstacles are introduced into the environment to evaluate the feasibility and stability of the control framework in more complex scenarios.
- In Scenario 3, a dynamic obstacle is employed to verify the feasibility of the control framework in real-time control, and performance of this controller is tested in environments with obstacles moving at different speeds.

4.2.2. Reinforcement Learning Setting

The standard two-delay deep deterministic policy gradient algorithm (TD3) [40,42] is employed for agent training. The delayed update and target value smoothing design of TD3 ensures more stable learning in continuous control tasks, and it has been tested by us for its good training results. The actor neural network input dimension is 15, comprising the complete state of the AUVs ($_{6\times1}x$), the reference state of the trajectories ($_{6\times1}x_d$), and the obstacle information ($_{3\times1}Obs$), including obstacle dimensions, locations, etc. The dimensions of the output are 8, corresponding to $_{3\times1}\beta_\theta^b$, β_θ^v and $_{1\times1}\alpha_\theta^b$, α_θ^v, respectively. The actor neural network and the criterion neural network each have three hidden layers, with a width of 512 for each layer. We build the training environment based on the AUV model described in Chapter 2 and design the training tasks for different task scenarios. In each control step, the QP controller will compute feasible control instructions based on the nominal model and execute them. The resulting states of the simulation will be used to estimate the error magnitude of the CBF and CLF constraints and fed into the neural network as a penalty term to help the neural network output the constraint error estimates, resulting in a more accurate and safer trajectory tracking controller after enough tasks have been executed. The agent was trained on simulations of tracking tasks comprising up to approximately 300 steps per episode. The discrete time step was set to $T_s = 0.1$ seconds, and the agent was trained to avoid obstacles and to complete trajectory tracking. The final agent was obtained after training over 1 million steps.

Table 2. Parameters of BlueROV2.

Notation	Value	Unit
m	13.5	kg
I_z	0.37	kg·m^2
X_u	13.7	Ns/m
$D_{u\|u\|}$	141.0	Ns^2/m^2
Y_v	0	Ns/m
$D_{v\|v\|}$	217.0	Ns^2/m^2
N_r	0	Ns
$N_{r\|r\|}$	1.5	Ns^2
$X_{\dot{u}}$	6.36	kg
$Y_{\dot{v}}$	7.12	kg
$N_{\dot{r}}$	0.222	kg·m^2

4.3. Results and Discussions

To evaluate the tracking performance of the trained agent in a trajectory tracking task with a single obstacle, a simulation of Scenario 1 was conducted in this subsection. The nominal model was augmented with varying degrees of model error. In Condition 1, an additional mass of 7 kg was introduced to the AUV, representing approximately 50% of the robot's original mass. Additionally, a scaling factor of 2 was applied to the quadratic damping coefficient, resulting in a value twice the true coefficient.

Under these conditions, the RL-CBF-CLF-QP controller was shown to avoid the implementation of an overly aggressive control strategy. This is illustrated in Figure 4a, where the model uncertainty led to instances of collision. In Condition 2, the nominal model was assumed to underestimate both the mass and water damping coefficient of the actual system. The nominal mass was set to approximately 50% of the real mass, and the quadratic damping coefficient was scaled by a factor of 0.5. As shown in Figure 4a, the conventional CBF-CLF-QP method can complete the tracking task with comparable efficacy. However, the model uncertainty led to an overly conservative control strategy. The maximum tracking error for both methods was evaluated, with CBF-CLF-QP exhibiting a maximum error of 4.85 m during control, while the RL-CBF-CLF-QP method showed a reduced error of 3.42 m.

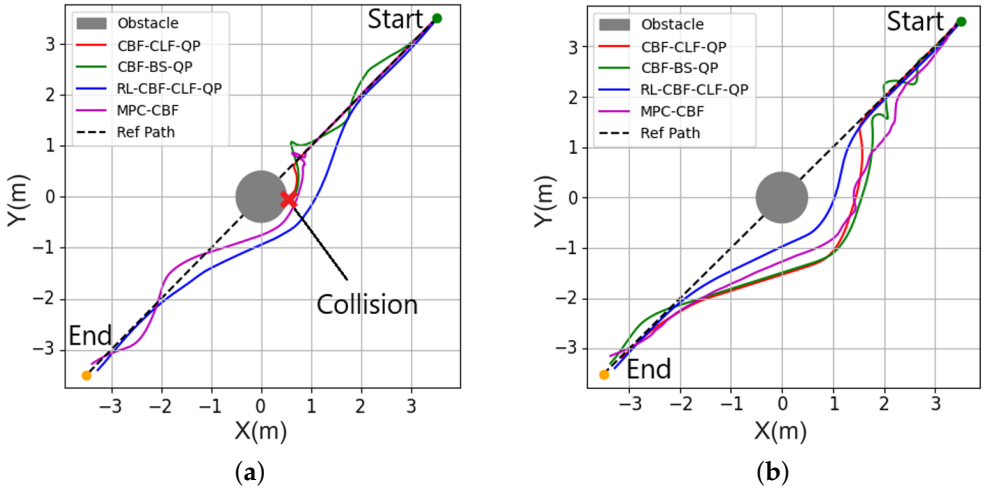


Figure 4. Obstacle avoidance tracking trajectories of the AUV under different model uncertainties: (**a**) tracking trajectories under condition 1; (**b**) tracking trajectories under condition 2.

In addition to the original CBF-CLF-QP approach, a minimally invasive QP controller based on backstepping control (CBF-BS-QP) [43] is also tested in scenario 1, which is usually considered robust to model uncertainties and perturbations. However, the CBF-BS-QP method resulted in significant errors and oscillations in the states of the AUV system in the tracking task (as shown in Figure 4). The fundamental cause of this issue may be attributed to the control structure of the CBF-BS-QP framework. In the CBF-BS-QP method, the desired control inputs are first generated, and the QP problem is subsequently constructed to optimally solve the problem by minimally modifying these control inputs. Consequently, the effects of uncertainty on the CBF constraints and the backstepping controllers are coupled and superimposed, which may undermine the effectiveness of the CBF-BS-QP method. We also tested the method of combining MPC (Model Predictive Control) with the CBF. The MPC-CBF method iteratively considers the constraints of the CBF in each prediction step so that MPC contains the corresponding security constraints that ensure system security [44]. However, we can see from the figure that since the MPC-based control method is highly dependent on an accurate model, it also performs poorly in the face of model uncertainty. Additionally, due to the nature of the principle of its control method, the effect of model inaccuracy on the MPC controller will be magnified as the prediction window N increases. Given the poor performance of the CBF-BS-QP and MPC-CBF method in Scenario 1, it will no longer be used as a comparison method in subsequent scenario tests.

In the simulation of Scenario 2, multiple obstacles were introduced to assess the generalization capability of the proposed method. The model errors applied in Scenario

1 were incorporated. As illustrated in the trajectory Figure 5, the presence of model uncertainty led to a miscalculation in the CBF evaluation, resulting in the system adopting an overly aggressive control strategy. This, in turn, extended the task completion time. After simulating the multi-obstacle scenario, the completion time was measured. The results demonstrate that the RL-based control method completed the tracking task in 25.3 s (with a reference time of 20 s), while the original method required 29.2 s. These findings suggest that the increased obstacle avoidance time due to model uncertainty is mitigated through the use of the RL method. Further analysis revealed that the performance discrepancy stemmed from an inaccurate estimation of the CBF. This inaccuracy caused the safety constraints to activate only when the AUV was in close proximity to the obstacles, delaying the activation of the obstacle avoidance mechanism. As a result, the AUV was compelled to brake abruptly or, in some instances, reverse, which significantly increased the time required for obstacle avoidance.

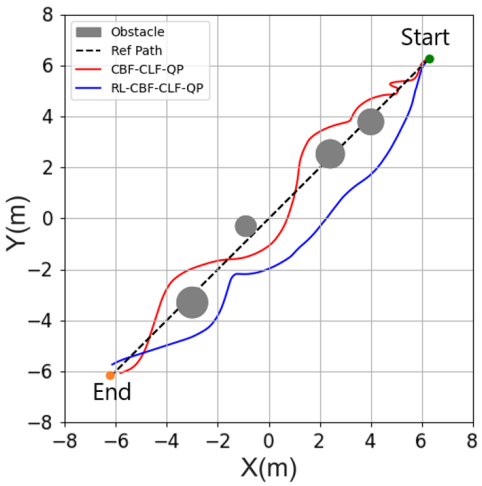


Figure 5. Trajectory comparison of two methods in response to multiple obstacles.

In Scenario 3, dynamic obstacles, representative of those encountered in real-world tasks, are simulated. Using the model uncertainty observed in prior scenarios, the performance of the RL-based controller is evaluated across varying obstacle sizes and velocities. The discrete control strategy enables these controllers to address dynamic obstacles effectively. However, model uncertainty may lead to inaccuracies in the estimation of safety constraints at each control step, thereby increasing the potential for collisions in dynamic environments. Simulations were conducted under conditions with obstacles of differing radii (0.1 m–0.6 m) and velocities (0.1 m/s–0.3 m/s). Results indicate that system safety was preserved in several more challenging cases, including those with high-speed or large obstacles. In contrast, the non-RL approach demonstrated a higher incidence of collisions, as shown in Figure 6.

In order to better quantify the performance advantages of the method, some indicators are defined as follows:

- Maximum Tracking Error E_{traj}: This metric quantifies the Euclidean distance between the robot's real-time position and the target coordinates on the reference trajectory. A smaller maximum tracking error signifies superior tracking performance, with lower values indicating a more precise adherence to the trajectory.
- Minimum Barrier Function Value h_{cmin}: Defined in Equation (16), the barrier function h_c represents the system's safety criteria. The safety radius r_s—the designed obstacle avoidance threshold for the AUV within the CBF constraints—consists of the obstacle's radius, the robot's radius, and an added safety margin. This threshold serves as a

baseline for safe operation: when the actual obstacle distance falls below r_s, the AUV enters a high-risk zone. Conversely, if the barrier value is significantly greater than 0, the control strategy may be overly conservative. Thus, an optimal value for h_{cmin} would be positive and close to zero.

- Task Completion Time T_c: Due to the added complexity of obstacle avoidance, the actual task time generally exceeds the expected time. Hence, shorter task completion times reflect the controller's efficiency in terms of expedience and time management.

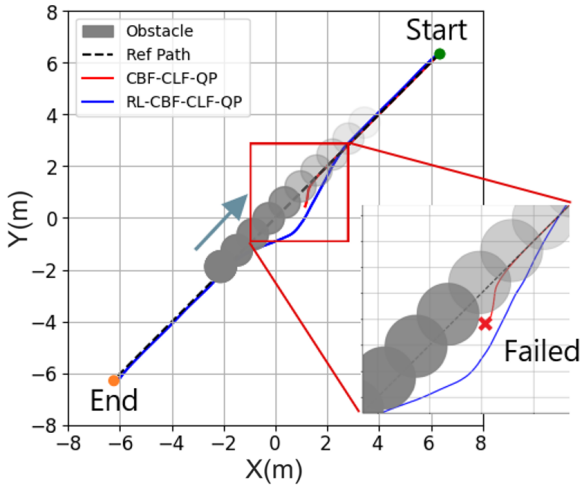


Figure 6. Trajectory comparison of the two methods in response to dynamic obstacles.

Subsequently, based on the simulation results, a statistical comparison of the method's performance across the aforementioned indicators is presented in Table 3. It is important to note that, to quantitatively assess the advantages of the proposed method, the obstacle avoidance requirements in the failed scenarios were relaxed to facilitate the acquisition of relevant data. Specifically, the obstacle radius in Scenario 1, Condition 1, was reduced to 0.4 m, and the expected trajectory speed was lowered to mitigate the risk of collisions. Additionally, a lower obstacle speed of 0.15 m/s was selected in Scenario 3 to ensure the collection of comparable quantitative data.

According to the data presented in Table 3, it is evident that the RL-based method achieves faster and safer completion of the tracking task across various scenarios. While the CBF-CLF-QP method successfully completes the task in several instances, it generates control commands that are deemed dangerous based on the defined safety distance, as too closely to obstacles. Notably, in Scenario 1, Condition 2, the RL-based method accomplishes the task with a minimum barrier function value that is both relatively safe and closer to zero. This indicates that the RL-based method effectively mitigates the overly conservative control issues associated with model uncertainty.

Table 3. Comparison results in different scenarios. The three parameter units are E_{traj} (m), h_{cmin} (m), and T_c (s).

Methods	Scenario 1						Scenario 2			Scenario 3		
	Condition 1			Condition 2								
	E_{traj}	h_{cmin}	T_c	E_{traj}	h_{cmin}	T_c	E_{traj}	h_{cmin}	T_c	E_{traj}	h_{cmin}	T_c
CBF-CLF-QP	2.899	−0.146	30.8	4.582	0.82	20.6	7.586	−0.154	29.5	3.22	−0.169	25.6
RL-CBF-CLF-QP	1.394	0.001	30.4	3.419	0.29	19.5	2.841	0.289	21.9	2.15	0.023	23.8

To gain deeper insights into the role of RL in this task, the generalized force generated by the robot thruster under Condition 1 of Scenario 1 (corresponding to the data in the second to fourth columns of Table 3) was recorded. A comparison of the time variation of the generalized force reveals that the RL-based method initiates lateral obstacle avoidance control around 8 to 10 s, as depicted in Figure 7a. The enhanced accuracy of the CBF enables the AUV to respond to the obstacle at an earlier stage, thereby ensuring safety and further underscoring the superiority of the RL-based method. Similarly, in Figure 7b, the system achieves safety-critical tracking control with a significantly reduced control effort. In contrast, the CBF-CLF-QP method suffers from model inaccuracies, leading to a delayed response in the CBF function and the control inputs to reach the thrust boundaries, which does not match the safety-critical control requirements.

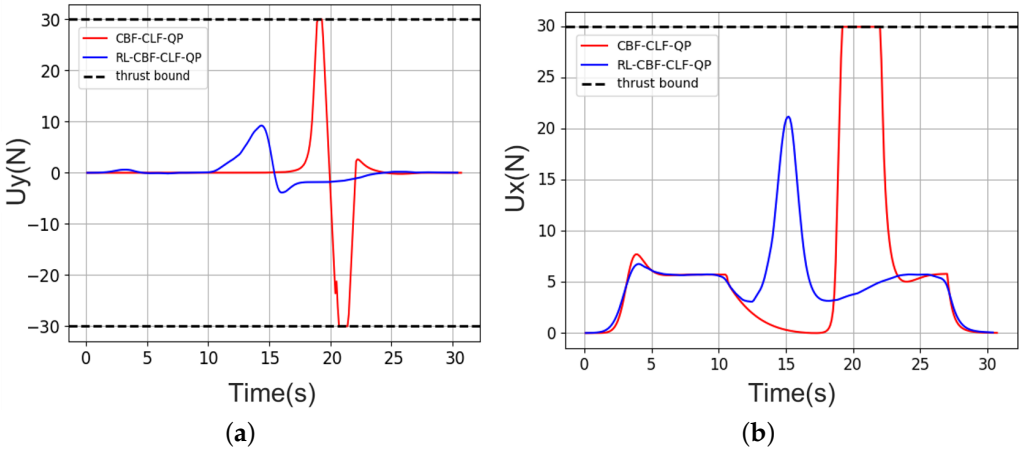


Figure 7. Curve of AUV generalized control over time: (**a**) lateral generalized force variation in the O_b reference frame; (**b**) forward generalized force variation in the O_b reference frame.

5. Conclusions

This paper presented an RL-CBF-CLF-QP-based trajectory tracking controller for AUVs, along with the establishment of corresponding safety and stability constraints. Additionally, an RL method was integrated with the CLF and CBF to effectively address model uncertainty, leading to more robust trajectory tracking strategies for AUVs. The proposed safety-enhanced controller was simulated and validated under various obstacle scenarios, with the results demonstrating that the RL-based method significantly enhanced AUV safety in uncertain conditions and markedly increased task success rates. These findings substantiate the efficacy and superiority of the proposed controller. While a comprehensive quantitative analysis of the RL-CBF-CLF-QP controller's algorithmic complexity may require further investigation, we have given careful consideration to its practical feasibility. The RL component requires significant computational resources during training, but once the policy is learned, real-time inference is computationally efficient. The CBF and CLF components introduce additional constraints, which are enforced via a quadratic programming (QP) solver. The solver is generally fast for low-dimensional systems, but the computational burden increases with system complexity and the number of safety constraints. We believe that the RL-CBF-CLF-QP controller is feasible in many practical scenarios, particularly after the training phase. For more complex, high-dimensional systems, optimizations such as efficient QP solvers or reducing safety constraints could be considered to ensure real-time performance. We plan to explore these aspects in future work.

Author Contributions: All authors contributed to the study conception and design. Conceptualization: T.L. and J.T.; methodology: J.T.; formal analysis and investigation: T.L., S.C. and Y.W.; writing—original draft: J.T.; writing—review and editing: B.Z. and Y.W.; supervision: T.L., B.Z. and Y.W.; funding acquisition: Y.H. and B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) (No. 62173234), the Shenzhen Science and Technology Program (No. JCYJ20220818095816035), and Key-Area Research and Development Program of Guangdong Province (No. 2023B1111050010).

Data Availability Statement: Data will be made available on reasonable request.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Blidberg, D.R. The development of autonomous underwater vehicles (AUV); a brief summary. *Proc. IEEE ICRA* **2001**, *4*, 122–129.
2. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: A review. *IEEE J. Ocean. Eng.* **2013**, *39*, 131–149. [CrossRef]
3. Yuh, J. Design and control of autonomous underwater robots: A survey. *Auton. Robot.* **2000**, *8*, 7–24. [CrossRef]
4. Zeng, Z.; Lian, L.; Sammut, K.; He, F.; Tang, Y.; Lammas, A. A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Eng.* **2015**, *110*, 303–313. [CrossRef]
5. Shi, Y.; Shen, C.; Fang, H.; Li, H. Advanced control in marine mechatronic systems: A survey. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1121–1131. [CrossRef]
6. Hammad, M.M.; Elshenawy, A.K.; El Singaby, M. Trajectory following and stabilization control of fully actuated AUV using inverse kinematics and self-tuning fuzzy PID. *PLoS ONE* **2017**, *12*, e0179611. [CrossRef]
7. Li, D.; Du, L. Auv trajectory tracking models and control strategies: A review. *J. Mar. Sci. Eng.* **2021**, *9*, 1020. [CrossRef]
8. Ames, A.D.; Grizzle, J.W.; Tabuada, P. Control barrier function based quadratic programs with application to adaptive cruise control. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; pp. 6271–6278.
9. Sontag, E.D. A Lyapunov-like characterization of asymptotic controllability. *SIAM J. Control Optim.* **1983**, *21*, 462–471. [CrossRef]
10. Li, B.; Wen, S.; Yan, Z.; Wen, G.; Huang, T. A survey on the control lyapunov function and control barrier function for nonlinear-affine control systems. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 584–602. [CrossRef]
11. Galloway, K.; Sreenath, K.; Ames, A.D.; Grizzle, J.W. Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs. *IEEE Access* **2015**, *3*, 323–332. [CrossRef]
12. Ames, A.D.; Powell, M. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. In *Proceedings of the Control of Cyber-Physical Systems: Workshop Held at Johns Hopkins University, March 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 219–240.
13. Desai, M.; Ghaffari, A. Clf-cbf based quadratic programs for safe motion control of nonholonomic mobile robots in presence of moving obstacles. In Proceedings of the 2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Sapporo, Japan, 11–15 July 2022; pp. 16–21.
14. He, L.; Zhang, Y.; Fan, G.; Liu, Y.; Wang, X.; Yuan, Z. Three-Dimensional Path Following Control of Underactuated AUV Based on Nonlinear Disturbance Observer and Adaptive Line-of-Sight Guidance. *IEEE Access* **2024**, *12*, 83911–83924. [CrossRef]
15. Thanh, P.N.N.; Tam, P.M.; Anh, H.P.H. A new approach for three-dimensional trajectory tracking control of under-actuated AUVs with model uncertainties. *Ocean Eng.* **2021**, *228*, 108951. [CrossRef]
16. Yan, J.; Li, X.; Yang, X.; Luo, X.; Hua, C.; Guan, X. Integrated localization and tracking for AUV with model uncertainties via scalable sampling-based reinforcement learning approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 6952–6967. [CrossRef]
17. Yan, Z.; Wang, M.; Xu, J. Robust adaptive sliding mode control of underactuated autonomous underwater vehicles with uncertain dynamics. *Ocean Eng.* **2019**, *173*, 802–809. [CrossRef]
18. Yu, R.; Shi, Z.; Huang, C.; Li, T.; Ma, Q. Deep reinforcement learning based optimal trajectory tracking control of autonomous underwater vehicle. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 4958–4965.
19. Jiang, P.; Song, S.; Huang, G. Attention-based meta-reinforcement learning for tracking control of AUV with time-varying dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6388–6401. [CrossRef]
20. Fang, Y.; Huang, Z.; Pu, J.; Zhang, J. AUV position tracking and trajectory control based on fast-deployed deep reinforcement learning method. *Ocean Eng.* **2022**, *245*, 110452. [CrossRef]

21. Yan, J.; Lin, J.; Yang, X.; Chen, C.; Guan, X. Cooperation Detection and Tracking of Underwater Target via Aerial-Surface-Underwater Vehicles. *IEEE Trans. Autom. Control* **2024**, early access, 1–16. [CrossRef]
22. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
23. Taylor, A.; Singletary, A.; Yue, Y.; Ames, A. Learning for safety-critical control with control barrier functions. In Proceedings of the Learning for Dynamics and Control, PMLR, Berkeley, CA, USA, 11–12 June 2020; pp. 708–717.
24. Cheng, R.; Khojasteh, M.J.; Ames, A.D.; Burdick, J.W. Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC), Jeju, Republic of Korea, 14–18 December 2020; pp. 777–783.
25. Choi, J.J.; Castañeda, F.; Tomlin, C.J.; Sreenath, K. Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions. *arXiv* **2020**, arXiv:2004.07584.
26. Castaneda, F.; Choi, J.J.; Zhang, B.; Tomlin, C.J.; Sreenath, K. Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 14–17 December 2021; pp. 6762–6769.
27. Khan, M.; Chatterjee, A. Gaussian Control Barrier Functions: Safe Learning and Control. In Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC), Jeju, Republic of Korea, 14–18 December 2020; pp. 3316–3322.
28. Joe, H.; Kim, M.; Yu, S.c. Second-order sliding-mode controller for autonomous underwater vehicle in the presence of unknown disturbances. *Nonlinear Dyn.* **2014**, *78*, 183–196. [CrossRef]
29. Wu, H.M.; Karkoub, M. Hierarchical backstepping control for trajectory-tracking of autonomous underwater vehicles subject to uncertainties. In Proceedings of the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Gyeonggi-do, Republic of Korea, 22–25 October 2014; pp. 1191–1196.
30. Sahu, B.K.; Subudhi, B. Adaptive tracking control of an autonomous underwater vehicle. *Int. J. Autom. Comput.* **2014**, *11*, 299–307. [CrossRef]
31. González, J.; Gomáriz, S.; Batlle, C.; Galarza, C. Fuzzy controller for the yaw and velocity control of the Guanay II AUV. *IFAC-PapersOnLine* **2015**, *48*, 268–273. [CrossRef]
32. Fossen, T.I. *Marine Control Systems—Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*; Org. Number NO 985 195 005 MVA; Marine Cybernetics: Trondheim, Norway, 2002. Available online: www.marinecybernetics.com (accessed on 18 October 2022).
33. Freeman, R.A.; Kokotovic, P.V. Inverse optimality in robust stabilization. *SIAM J. Control Optim.* **1996**, *34*, 1365–1391. [CrossRef]
34. Artstein, Z. Stabilization with relaxed controls. *Nonlinear Anal. Theory Methods Appl.* **1983**, *7*, 1163–1173. [CrossRef]
35. Sontag, E.D. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Syst. Control Lett.* **1989**, *13*, 117–123. [CrossRef]
36. Ames, A.D.; Galloway, K.; Sreenath, K.; Grizzle, J.W. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Trans. Autom. Control* **2014**, *59*, 876–891. [CrossRef]
37. Tee, K.P.; Ge, S.S.; Tay, E.H. Barrier Lyapunov functions for the control of output-constrained nonlinear systems. *Automatica* **2009**, *45*, 918–927. [CrossRef]
38. Wieland, P.; Allgöwer, F. Constructive safety using control barrier functions. *IFAC Proc. Vol.* **2007**, *40*, 462–467. [CrossRef]
39. Nguyen, Q.; Sreenath, K. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 322–328.
40. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
41. Li, Q.; Cao, Y.; Li, B.; Ingram, D.M.; Kiprakis, A. Numerical modelling and experimental testing of the hydrodynamic characteristics for an open-frame remotely operated vehicle. *J. Mar. Sci. Eng.* **2020**, *8*, 688. [CrossRef]
42. Huang, S.; Dossa, R.F.J.; Ye, C.; Braga, J.; Chakraborty, D.; Mehta, K.; Araújo, J.G. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *J. Mach. Learn. Res.* **2022**, *23*, 1–18.
43. Wang, C.; Yu, W.; Zhu, S.; Song, L.; Guan, X. Safety-critical trajectory generation and tracking control of autonomous underwater vehicles. *IEEE J. Ocean. Eng.* **2022**, *48*, 93–111. [CrossRef]
44. Liu, S.; Zeng, J.; Sreenath, K.; Belta, C.A. Iterative Convex Optimization for Model Predictive Control with Discrete-Time High-Order Control Barrier Functions. In Proceedings of the 2023 American Control Conference (ACC), San Diego, CA, USA, 31 May–2 June 2023; pp. 3368–3375. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.