IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.
Cite as T-ASE paper.

CBF-Based Hierarchical Quadratic Programs with Guaranteed Feasibility for Safety-Critical Systems

Junjun Xie[1], Liang Hu[1], Yunzhe Tan[1], and Jun Yang[2]

***Abstract*—Control Barrier Function (CBF) based quadratic programs (QPs) have become an effective method for enforcing safety in safety-critical systems and robotics. However, these methods often suffer from infeasibility or overly conservative relaxations when handling multiple constraints, potentially compromising safety. In this paper, we propose a hierarchical framework called "Safety-first" for control design, which simultaneously incorporates performance objectives formulated using Control Lyapunov Functions (CLFs), and safety guarantees via CBFs with input constraints. Unlike existing approaches, the proposed method guarantees solution feasibility while achieving improved performance, and it is scalable to an arbitrary number of CBF constraints. This scalability enables more precise and flexible representation of complex safety requirements using multiple simple CBFs. For application to mobile robot navigation, we employ Constrained Delaunay Triangulation (CDT) to construct multiple CBFs that approximate irregularly-shaped obstacles. Real-world experiments in cluttered and dynamic environments demonstrate that the Safety-first algorithm achieves safe navigation, validating both the theoretical guarantee and practical advantages over existing methods.**

***Note to Practitioners*—This paper is motivated by the need to ensure safety and solution feasibility for safe-critical systems with multiple constraints, such as safe navigation through irregular or dynamic obstacles. Control Barrier Functions (CBFs) have emerged as a promising approach for enforcing safety in optimization-based control, but existing approaches face feasibility issues as constraint complexity grows, especially when considering input limitations and performance goals as well. In practice, a single constraint is often insufficient to represent safety requirements, such as autonomous navigation among irregular or dynamic obstacles. To overcome these issues, this paper provides a hierarchical optimization framework that prioritizes safety yet improving other performance. It guarantees feasibility under any number of constraints, and models safety in complex environments accurately through multiple, easily constructed CBFs. The proposed method is applied to safe navigation tasks involving irregular and dynamic obstacles, showing that our algorithm achieves safe and reliable navigation in challenging scenarios. These results demonstrate the practical value of the framework and its potential for broader application to safety-critical systems operating in uncertain and/or unstructured real-world environments.**

***Index Terms*—Control barrier function, safe navigation, collision avoidance, autonomous systems.**

This work was supported in part by the National Natural Science Foundation of China under Grant 62573157, the Science Center Program of National Natural Science Foundation of China under Grant 62188101, and Shenzhen Science and Innovation Committee under Grant JCYJ20241202123714019.

[1] J. Xie, L. Hu, and Y. Tan are with the School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: 25b965003@stu.hit.edu.cn; 25s165058@stu.hit.edu.cn; l.hu@hit.edu.cn).

[2] J. Yang is with the Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, LE11 3TU, UK (e-mail: j.yang3@lboro.ac.uk).

I. Introduction

As autonomous systems such as drones and self-driving vehicles become more prevalent in our societies, concerns about their safety, e.g., collisions with humans, have also grown. The potential for catastrophic outcomes due to incorrect operations of autonomous systems underscores the urgency of addressing safety issues. As such, significant research efforts have been directed toward developing novel safety-critical control design paradigms that explicitly account for safety requirements alongside traditional performance goals like stability and convergence.

Control barrier functions (CBFs) have recently emerged as a promising method for incorporating safety specifications into controller design. It has become a popular design paradigm deployed across various robotic platforms, including manipulators [1, 2], bipedal robots [3, 4], mobile robots [5, 6], and autonomous vehicles [7, 8]. Typically, CBFs encode safety specifications, such as collision avoidance with obstacles, as constraints in the optimization-based framework. In addition, safety is rarely the sole constraint for autonomous systems, as they must also meet input constraints and accomplish performance objectives. A well-established framework termed CLF-CBF QP [9, 10] was proposed to address these requirements, in which a quadratic programming (QP) formulation that integrates Control Lyapunov Functions (CLFs) and CBFs. Since then, control barrier functions have seen significant advancements. [11–13] introduced high-order CBFs to address systems with high relative degrees, while [14] proposed probabilistic CBFs for stochastic systems, and [15] developed fault-tolerant CBFs to handle model uncertainties.

Despite their promise, two key challenges limit the broader adoption of CBF-based frameworks in complex real-world applications. The first one is CBF design for tasks in complex environments. The effectiveness of CBFs hinges on accurate encoding of the safe set. While existing methods can handle obstacles of simple geometric shapes such as circles [16], ellipses [17, 18], and polygons [19], they often fail to generalize to irregularly shaped or dynamic obstacles. The common approach that approximates the obstacle using one simple geometric shape such as the circumscribed circle is often too conservative to characterize the navigation areas, limiting navigability in cluttered environments. Alternative solutions such as Nonsmooth Control Barrier Functions (NCBFs) [20, 21] combine multiple control barrier functions into a single one to characterize safety against multiple obstacles. However, the nonsmooth nature of NCBFs often results in control oscillations in practice. Another approach neural CBF constructs CBFs

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

in complex environments using deep learning techniques [22, 23]. Compared to traditional methods, neural CBFs tend to be less conservative and can better accommodate complex environments, but they require extensive prior data, limiting their applicability in unknown environments.

The second issue is the potential infeasibility when dealing with multiple constraints. CBFs expressed in the form of inequality constraints are integrated into various optimization frameworks such as quadratic programming (QP) [9, 10, 24] and model predictive control (MPC) [25]. However, multiple constraints can cause conflicts, resulting in the infeasibility of the optimization problem. To address this, a common approach is to formulate some parameters in the constraints as optimization variables for dynamic adaptation. For example, [26] achieved feasibility by adaptively tuning parameters within the constraint formulation; [27] improved over the classical QP method to provide point-wise feasibility, and [16] introduced the parameters as optimization variables into the MPC formulation. However, the parameters optimized in such methods usually depend on the value of the CBF, and thus may fail in certain states. Moreover, in practical applications, the controller is highly sensitive to the weights in the optimization objective, and inappropriate choices may lead to a loss of safety guarantees. [28] introduced an additional CBF to ensure the feasibility of the original CBF constraint, yet the added CBF might itself introduce new infeasibility issues. [29] proposed a bi-level optimization framework considering one CBF and input constraints, and [30] provided a hierarchical structure with different levels of safety, but they ignored performance objectives and input constraints.

To propose a unified solution that tackles both issues discussed above, we introduce a hierarchical optimization framework, termed "Safety-first" that guarantees feasibility under an arbitrary number of CBFs and input constraints. The key idea is to prioritize safety constraints at the top level of a hierarchical QP structure while allowing lower levels to address performance objectives and auxiliary considerations. This approach ensures that safety is never compromised, even in constraint-dense environments. The main contributions of this paper are summarized below:

1) We propose a hierarchical QP framework with CLF and CBF constraints, and theoretically prove its feasibility. Furthermore, the framework is made scalable to accommodate multiple constraints.
2) We analyze the relationship between the proposed framework and typical QP-based methods through a unified formulation, revealing the root causes of infeasibility, safety compromise, and an often overlooked issue—performance degradation in prior work. An Adaptive Cruise Control (ACC) example is used for comparison and validation.
3) We apply our framework to real-world robot navigation among multiple irregularly-shaped obstacles without a prior map. A well-suited geometric approximation method based on Constrained Delaunay Triangulation (CDT) is exploited to construct multiple simple CBFs efficiently, which collectively represent complex, irregular obstacles with reduced conservativeness.

A. Organization and Notation

The remainder of this paper is organized as follows. Section II introduces the mathematical preliminaries and basic concepts. Section III presents the proposed Safety-First CLF-CBF QP method, analyzes its performance and reveals the fundamental limitations of existing methods through a unified framework, and extends it to support multiple constraints. Section IV illustrates the proposed method using an Adaptive Cruise Control (ACC) example, where simulation results demonstrate its improved performance over existing methods. Section V focuses on safe navigation. It presents a novel obstacle approximation technique that represents arbitrarily-shaped obstacles using multiple CBFs. Section VI validates the proposed method in real-world navigation experiments across three challenging scenarios, including narrow corridors, irregular obstacles, and dynamic obstacles. Finally, Section VII summarizes the contributions and concludes the paper.

Notation: $\partial\mathcal{C}$ and $\mathrm{Int}(\mathcal{C})$ stand for the interior and boundary of the set $\mathcal{C}$. $\|\cdot\|$ denotes the Euclidean norm of the vector. $\nabla(\cdot)$ denotes the gradient of the function. A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ belongs to extended $\mathcal{K}_\infty$ functions if $\alpha(0) = 0$, $\lim_{r\to+\infty} \alpha(r) = +\infty$, $\lim_{r\to-\infty} \alpha(r) = -\infty$ and it is strictly increasing. $L_a b(x)$ denotes the Lie-derivative of $b(x)$ along $a(x)$. $\mathcal{A} \times \mathcal{B}$ denotes the Cartesian product of sets $\mathcal{A}$ and $\mathcal{B}$. $\mathbf{0}$ represents a column vector with all elements equal to zero.

II. Preliminaries

In this section, we briefly introduce the concepts of CBFs, CLFs, and two classic control design methods that integrate CBFs and CLFs in a QP framework.

A. Control Lyapunov/Barrier Functions

Consider a control-affine system

$$\dot{x} = f(x) + g(x)u,\ x(t_0) = x_0, u \in \mathcal{U} \subset \mathbb{R}^m \tag{1}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz, and the input constraint $\mathcal{U}$ is a convex polytope.

Definition 1. (Control Lyapunov Function) [31] A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is an *exponentially stabilizing control Lyapunov function (CLF)* if there exist positive constants $c_1, c_2, c_3 > 0$ for all $x \in \mathbb{R}^n$ such that

$$c_1\|x\|^2 \le V(x) \le c_2\|x\|^2, \tag{2}$$

$$\inf_{u\in U} \left[L_f V(x) + L_g V(x)u + c_3 V(x)\right] \le 0. \tag{3}$$

Definition 2. (Control Barrier Function) [24] Assume a closed set $\mathcal{C}$ is defined as

$$\begin{aligned} \mathcal{C} &= \{x \in \mathbb{R}^n : h(x) \ge 0\}, \\ \partial\mathcal{C} &= \{x \in \mathbb{R}^n : h(x) = 0\}, \\ \mathrm{Int}(\mathcal{C}) &= \{x \in \mathbb{R}^n : h(x) > 0\}, \end{aligned} \tag{4}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. The function $h(x)$ is called a *control barrier function (CBF)* if there exists an extended $\mathcal{K}_\infty$ function α such that

$$\sup_{u\in U} \left[L_f h(x) + L_g h(x)u + \alpha(h(x))\right] \ge 0 \tag{5}$$

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

and the set $\mathcal{C}$ is called a safe set.

Lemma 1. [24] Given a set $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$ defined by (4) for a continuously differentiable function h, if h is a CBF on $\mathcal{D}$, then any Lipschitz continuous controller $u : \mathcal{D} \to \mathcal{U}$ such that $u(x) \in K_{\text{cbf}}(x)$ where

$$K_{\text{cbf}}(x) = \{u \in \mathcal{U} : L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\} \tag{6}$$

renders the set $\mathcal{C}$ forward invariant, i.e.,

$$\forall x(t_0) \in \mathcal{C} \quad \Rightarrow \quad x(t) \in \mathcal{C}, \forall t \geq t_0. \tag{7}$$

B. CLF-CBF-based QP methods

Based on the concepts of CLFs and CBFs, [24] proposed a quadratic optimization framework **CLF-CBF QP** for safe control design of system (1), which combined CLF and CBF constraints with expected decay rates λ and γ:

$$[u^*, \delta^*] = \arg\min_{[u,\delta]} \frac{1}{2} u^T H u + p\delta^2 \tag{8a}$$

$$\textbf{s.t.} \quad L_f V(x) + L_g V(x) u + \lambda V(x) \leq \delta \tag{8b}$$

$$L_f h(x) + L_g h(x) u + \gamma h(x) \geq 0 \tag{8c}$$

$$u \in \mathcal{U} \subset \mathbb{R}^m \tag{8d}$$

where the slack variable δ makes the CLF constraint (8b) a soft constraint, mitigating its conflict with the CBF (hard) constraint (8c), and the positive definite symmetric matrix H and weight coefficient p are hyperparameters to be tuned beforehand.

The above optimization problem might become conservative or even infeasible if the decay rate γ is not fine-tuned. An improved method called **Optimal-decay CLF-CBF QP** was proposed in [27] as below:

$$[u^*, \delta^*, \omega^*] = \arg\min_{[u,\delta,\omega]} \frac{1}{2} u^T H u + p\delta^2 + p_\omega(\omega - \omega_0)^2 \tag{9a}$$

$$\textbf{s.t.} \quad L_f V(x) + L_g V(x) u + \lambda V(x) \leq \delta \tag{9b}$$

$$L_f h(x) + L_g h(x) u + \omega\gamma_0 h(x) \geq 0 \tag{9c}$$

$$u \in \mathcal{U} \subset \mathbb{R}^m \tag{9d}$$

where two additional parameters p_ω and ω_0 are introduced to ensure the feasibility of the QP. Unfortunately, as pointed out in [27], the safety of system (1) is no longer strictly guaranteed after (8c) is relaxed into (9c).

Both methods above combine multiple objectives into a single objective optimization problem by assigning different weightings to each one. Such an approach allows individual objectives to be sacrificed if it improves the overall weighted objective. However, such a trade-off is not suitable when considering safety, which is a fundamental and non-negotiable requirement for safety-critical systems. To address this limitation, we propose a new hierarchical optimization framework detailed in the next section.

III. Safety-first CLF-CBF QP: A Hierarchical Optimization Framework

In this section, we first present the hierarchical optimization framework Safety-first CLF-CBF QP. To facilitate comparative analysis with existing methods, we then propose a unified formulation which reveals the fundamental connection and differences between existing methods and our proposed one. Furthermore, the proposed method's performance is theoretically analyzed in detail. Finally, the framework is extended to handle multiple constraints.

A. Safety-first CLF-CBF QP

If safety and other performance objectives conflict, safety must be strictly guaranteed, even at the expense of sacrificing system performance. *Inspired by such an observation, we propose a hierarchical optimization framework that prioritizes safety rather than the multi-objective optimization frameworks that trade off safety and other objectives equally, as used in representative methods discussed in Subsection II-B.* We first consider the case with one CLF and one CBF constraint, and defer the case with multiple CBFs to Subsection III-E. As illustrated in Fig. 1, the entire framework is composed of three nested sub-problems: safety $\xrightarrow{\delta_2^*}$ performance $\xrightarrow{\delta_1^*}$ minimal input$\to u^*$, which are presented in detail below:

Sub-problem 1: Safety

$$\begin{aligned} &[\sim, \sim, \delta_2^*] = \arg\min_{[u,\delta_1,\delta_2]} \delta_2^2 \\ \textbf{s.t.} \quad &L_f V(x) + L_g V(x) u + \lambda V(x) \leq \delta_1 \\ &L_f h(x) + L_g h(x) u + \gamma h(x) \geq \delta_2 \\ &u \in \mathcal{U} \subset \mathbb{R}^m \end{aligned} \tag{10}$$

In Sub-problem 1, safety constraints are enforced to the greatest extent as possible subject to input constraints. If a solution exists that strictly satisfies the safety constraint, i.e., $\delta_2^* = 0$, then a control input with guaranteed safety can be obtained later. Otherwise, it indicates that the safety specification is too demanding to achieve under the input constraints, and then a relaxation of the safety specification becomes necessary, which manifests as $\delta_2^* < 0$. Overall, this step characterizes the optimal level of safety achievable under input constraints, regardless of whether the safety constraint is strictly satisfiable or not.

Subsequently, δ_2^* is substituted into the next sub-problem for performance objective optimization.

Sub-problem 2: Performance

$$\begin{aligned} &[\sim, \delta_1^*] = \arg\min_{[u,\delta_1]} \delta_1^2 \\ \textbf{s.t.} \quad &L_f V(x) + L_g V(x) u + \lambda V(x) \leq \delta_1 \\ &L_f h(x) + L_g h(x) u + \gamma h(x) \geq \delta_2^* \\ &u \in \mathcal{U} \subset \mathbb{R}^m \end{aligned} \tag{11}$$

Similarly, this step yields the optimal performance subject to the safety constraint determined by Sub-problem 1. If a solution exists, the performance constraints are satisfied, i.e., $\delta_1^* = 0$, then the CLF will converge with a decay rate no less than the prescribed λ. Otherwise, it will converge as quickly as possible within the feasible set.

The two-step optimization outlined above ensures a lower bound on both safety and performance objectives, but may not uniquely determine the optimal control input u^*, due to the possible existence of multiple u satisfying the safety

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

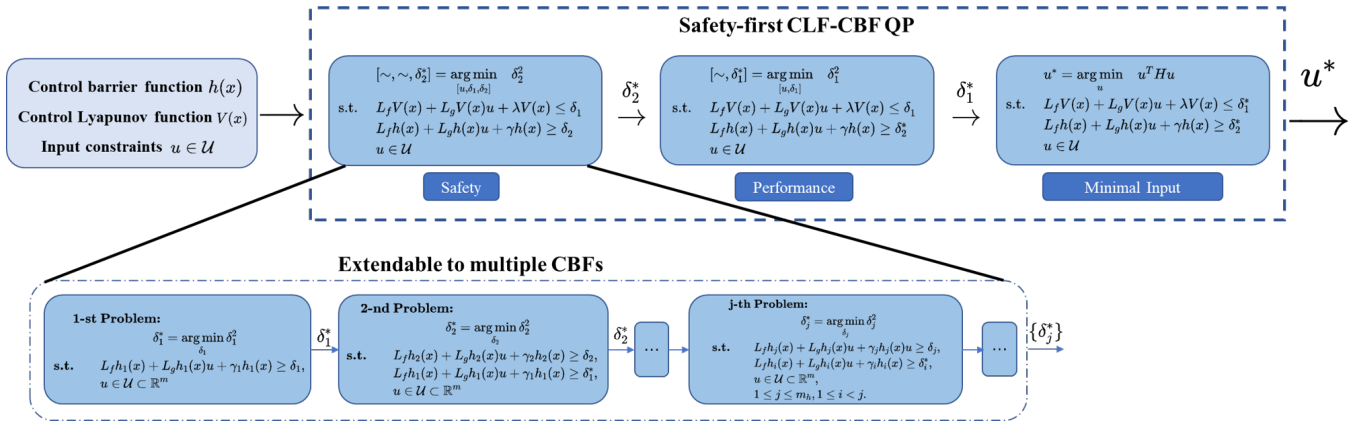


Fig. 1. The framework of Safety-first CLF-CBF QP. Safety, performance objective, and input are considered in a hierarchical order to ensure solution feasibility and safety. Moreover, it can be extended to handle multiple CBFs.

TABLE I
COMPARISONS OF VARIOUS QP-BASED METHODS

Methods	Corresponding Variables in Unified CLF-CBF QP				Guarantee		
	δ_1	δ_2	$\mathcal{U}_\delta$	$[p_1, p_2]$	Feasibility	Safety	Performance
CLF-CBF QP	δ	0	$\mathbb{R} \times \{0\}$	$[p, 1]$	No	Yes if feasible	No
Optimal-decay CLF-CBF QP	δ	$(\omega_0 - \omega)\gamma_0 h(x)$	$\begin{cases} \mathbb{R}^2 & \text{if } h(x) \neq 0 \\ \mathbb{R} \times \{0\} & \text{if } h(x) = 0 \end{cases}$	$[p, \frac{p\omega}{(\gamma_0 h(x))^2}]$	Yes if $h(x) \neq 0$	No	No
Safety-first CLF-CBF QP	δ_1	δ_2	$\mathbb{R}^2$	$[q, q^2], q \to +\infty$	Yes	Yes if possible*	Yes if possible*

* Safety or performance is strictly guaranteed when the corresponding CBF/CLF constraints can be satisfied under input constraints; otherwise, they are relaxed to the minimal extent.

constraints and performance requirements. Consequently, a further optimization step is necessary. To minimize the control input, δ_1^* and δ_2^* are substituted into the next problem:

Sub-problem 3: Minimal Input

$$\begin{aligned} & u^* = \arg\min_u u^T Hu \\ \textbf{s.t.} \quad & L_fV(x) + L_gV(x)u + \lambda V(x) \le \delta_1^* \\ & L_fh(x) + L_gh(x)u + \gamma h(x) \ge \delta_2^* \\ & u \in \mathcal{U} \subset \mathbb{R}^m \end{aligned} \tag{12}$$

where H is a positive definite symmetric matrix. Intuitively, u^* is a safe and stabilizing control input if $\delta_1^* = 0$ and $\delta_2^* = 0$ are always satisfied. When the input constraints are stringent, u^* sequentially satisfies the safety specifications and then the performance objectives as closely as possible. A detailed analysis of the relevant properties will be elaborated upon in the following subsections.

B. Unified CLF-CBF QP

The existing two methods, CLF-CBF QP and its Optimal-decay variant, suffer from either infeasibility or safety compromise. To ease the comparison between our proposed method and the existing ones, we unify the existing two methods in the following form (called **Unified CLF-CBF QP** thereafter):

$$[u^*, \delta_1^*, \delta_2^*] = \arg\min_{[u,\delta_1,\delta_2]} \frac{1}{2} u^T Hu + p_1\delta_1^2 + p_2\delta_2^2 \tag{13a}$$

$$\textbf{s.t.} \quad L_fV(x) + L_gV(x)u + \lambda V(x) \le \delta_1 \tag{13b}$$

$$L_fh(x) + L_gh(x)u + \gamma h(x) \ge \delta_2 \tag{13c}$$

$$u \in \mathcal{U} \subset \mathbb{R}^m \tag{13d}$$

$$[\delta_1, \delta_2] \in \mathcal{U}_\delta \tag{13e}$$

where $p_1, p_2 \ge 0$, and $\mathcal{U}_\delta \subset \mathbb{R}^2$. The correspondence parameters and variables between the two methods and the unified form are summarized in Table I.

Based on the unified form, the differences between the methods are easily captured by the feasible space of slack variables $\mathcal{U}_\delta$. Specifically, $\mathcal{U}_\delta = \mathbb{R} \times \{0\}$ for CLF-CBF QP, and $\mathbb{R}^2$ when $h(x) \neq 0$, $\mathbb{R} \times \{0\}$ when $h(x) = 0$ for Optimal-decay variant. Moreover, the sufficient conditions of feasibility, safety, and performance guarantee in terms of the feasible spaces are apparent:

- *Feasibility guarantee:* $\mathcal{U}_\delta = \mathbb{R}^2$;
- *Performance guarantee:* $\mathcal{U}_\delta = \{0\} \times \mathcal{A}$ (hard CLF constraint);
- *Safety guarantee:* $\mathcal{U}_\delta = \mathcal{A} \times \{0\}$ (hard CBF constraint);

where $\mathcal{A} \subset \mathbb{R}$ is an arbitrary set.

From this perspective, the advantages and disadvantages of different methods are revealed: CLF-CBF QP guarantees safety but suffers from issues of infeasibility and performance degradation; while Optimal-decay CLF-CBF QP is feasible when $h(x) \neq 0$, but it does not ensure guarantees for both safety and performance. Another crucial point is that the

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

condition for feasibility conflicts with those for safety and performance. Thus, the main question arises: *How should we balance feasibility, safety, and performance in the face of their inherent conflicts?* It will reveal that our proposed Safety-first hierarchical optimization provides the solution.

C. Limit-weighted Form of Unified CLF-CBF QP

Based on the above analysis, fixing the slack variables δ_1 and δ_1 cannot satisfy all three conditions simultaneously. Therefore, we need to set $\mathcal{U}_\delta = \mathbb{R}^2$ explicitly to guarantee feasibility and use other methods to make the slack variables always approach 0 as closely as possible. In trade-off-based multi-objective optimization, it is common to increase the weight of a particular objective, which makes the optimal solution more biased towards that objective alone.

In the following theorem, we prove that a limit-weighted form of the Unified CLF-CBF QP is equivalent to our proposed Safety-first CLF-CBF QP in Subsection III-A.

Theorem 1. Consider the following Unified CLF-CBF QP with a positive definite symmetric matrix H and weight coefficient q

$$\begin{aligned} \min \quad & \frac{1}{2}u^T H u + q\delta_1^2 + q^2\delta_2^2 \\ \textbf{s.t.} \quad & L_f V(x) + L_g V(x)u + \lambda V(x) \leq \delta_1 \\ & L_f h(x) + L_g h(x)u + \gamma h(x) \geq \delta_2 \\ & u \in \mathcal{U} \subset \mathbb{R}^m \end{aligned} \tag{14}$$

then

1) (14) is always feasible with any $q \geq 0$;
2) when the coefficient $q \to +\infty$, the optimal solution of (14) $[u', \delta_1', \delta_2']$ converges to $[u^*, \delta_1^*, \delta_2^*]$ obtained by the proposed Safety-first CLF-CBF QP (10)-(12);
3) when the coefficient $q \to +\infty$, $\delta_2' \to 0$ if $\delta_2 = 0$ is feasible, and $\delta_2', \to 0, \delta_1' \to 0$ if $\delta_2 = \delta_1 = 0$ is feasible.

The proof is provided in Appendix A.

Theorem 1 establishes the connections between Safety-first CLF-CBF QP and the unified form through a special structure, and explains how our method maintains feasibility while considering constraints from an innovative perspective.

D. Feasibility, Safety and Performance Analysis

This subsection presents a comprehensive analysis of the proposed Safety-first CLF-CBF QP method in terms of feasibility, safety, and performance.

1) Feasibility Analysis: The feasibility of CLF-CBF QP and Optimal-decay CLF-CBF QP has been analyzed in [27]. To summarize briefly, CLF-CBF QP suffers from infeasibility when constraints conflict, while Optimal-decay CLF-CBF QP is feasibility-guaranteed except when $h(x) = 0$. In comparison, our proposed Safety-first CLF-CBF QP is feasible all the time, as summarized in the following theorem.

Theorem 2. Given the system (1), the Safety-first CLF-CBF QP (10)-(12) is always feasible as long as $\mathcal{U} \neq \emptyset$.

Proof. Since $\mathcal{U} \neq \emptyset$, for any $u_a \in \mathcal{U}$, there always exists $\delta_{1a} = L_f V(x) + L_g V(x)u_a + \lambda V(x)$ and $\delta_{2a} = L_f h(x) + L_g h(x)u_a + \gamma h(x)$ satisfying the constraints. Obviously, $[u_a, \delta_{1a}, \delta_{2a}]$ is a feasible solution of the three sub-problems (10)-(12), i.e., the three sub-problems are feasible and the solutions $\delta_2^*, \delta_1^*, u^*$ always exist. ■

2) Safety and Performance Analysis: With the concept of forward invariance, *Lemma* 1 shows the safety-guaranteed ability of CBFs, which is inherited by CLF-CBF QP [9, 24]. However, Optimal-decay CLF-CBF QP is not forward complete, manifesting that the system is unsafe in some cases [27].

Another commonly overlooked issue is that, similar to the compromise on safety in the optimal-decay variant, the relaxation of the CLF constraint in CLF-CBF QP and its optimal-decay variant may lead to unnecessary performance degradation, even if all constraints can be satisfied strictly [9, 24]. In contrast, our proposed Safety-first method avoids compromising safety and performance if not needed, as summarized below.

Corollary 1. Consider the system (1), and define

$$K_{\text{cbf}}^{\gamma}(x) = \{u | L_f h(x) + L_g h(x)u + \gamma h(x) \geq 0\}, \tag{15}$$

and

$$K_{\text{clf}}^{\lambda}(x) = \{u | L_f V(x) + L_g V(x)u + \lambda V(x) \leq 0\}, \tag{16}$$

then the controller $u^*(x)$ derived from Safety-first CLF-CBF QP (10)-(12)

- guarantees safety, i.e., renders $\mathcal{C}$ forward invariant, if $\mathcal{U} \cap K_{\text{cbf}}^{\gamma}(x) \neq \emptyset$;
- guarantees safety and performance, i.e., renders $\mathcal{C}$ forward invariant and makes the system exponentially stable, if $\mathcal{U} \cap K_{\text{cbf}}^{\gamma}(x) \cap K_{\text{clf}}^{\lambda}(x) \neq \emptyset$.

The proof of this corollary is straightforward from *Theorem* 1 and *Lemma* 1.

For systems with input constraints, achieving safety in every possible state is not always feasible. Consider a vehicle driving at high speed: if it detects an obstacle far ahead, safety can be ensured through deceleration or full braking. However, if the obstacle is detected at a very close distance, avoiding a collision becomes impossible regardless of the control command. In such cases, maintaining full braking remains the most appropriate action, even if safety cannot be guaranteed.

In essence, our proposed Safety-first method follows the same manner. It computes safety-guaranteed control actions if the safe constraints can be satisfied strictly; otherwise, if constraints are impossible to be satisfied by any actions (as full braking in the above example), it will choose an action with minimal relaxation of constraints.

E. Extended Form for Multiple Constraints

In this subsection, we extend the proposed Safety-first CBF-CLF QP framework to accommodate multiple constraints.

The main idea is to assign a priority order to different objectives and constraints from highest to lowest: input constraints, safety constraints, performance objectives, and minimal input. This prioritization aligns well with practical application. For scenarios involving multiple constraints, such as multiple CBFs, this priority-based framework remains valid.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

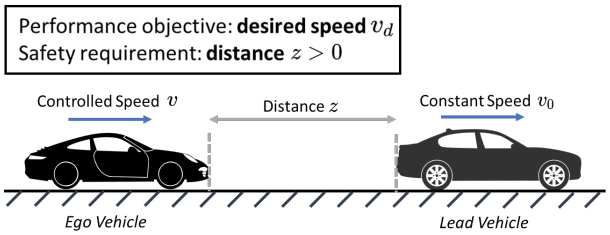


Fig. 2. The schematic diagram of the ACC case.

Assume that there are m_h CBFs $\{h_j(x)\}, 1 \leq j \leq m_h$, where $h_j(x)$ represents the CBF with the j-th highest priority. For each CBF $h_j(x)$, the corresponding constraint can be formulated as

$$L_f h_j(x) + L_g h_j(x) u + \gamma_j h_j(x) u \geq \delta_j \tag{17}$$

with the decay rate γ_j and slack variable δ_j. Following the defined priority order, $h_j(x)$ is considered in the j-th optimization problem, constructed as follows:

j-th Problem:

$$\delta_j^* = \arg\min_{\delta_j} \delta_j^2 \tag{18a}$$

s.t.
$$L_f h_j(x) + L_g h_j(x) u + \gamma_j h_j(x) u \geq \delta_j, \tag{18b}$$
$$L_f h_i(x) + L_g h_i(x) u + \gamma_i h_i(x) \geq \delta_i^*, \tag{18c}$$
$$u \in \mathcal{U} \subset \mathbb{R}^m, \tag{18d}$$
$$1 \leq j \leq m_h, 1 \leq i < j. \tag{18e}$$

where (18b) is the constraint to be considered in the current problem, while (18c) are higher-priority constraints that have already been considered in previous problems. The process is illustrated in Fig. 1.

The slack variables δ_j^* is derived from (18), and substituted into the $\{k+1\}$-th step. After all CBF constraints are addressed, then optimize the performance and input as (11) and (12) with all optimized CBF constraints substituted.

A natural way to prioritize CBF constraints is to assign higher priority to those with smaller CBF values, as they correspond to the most critical and risky conditions the system faces. Moreover, the choice of priority order does not affect the feasibility of the method, but it may influence the performance of safety. When multiple safety constraints conflict, lower-priority safety constraints may be compromised to satisfy higher-priority ones. Notice that we can still retain a trade-off approach within a given priority level, where multiple objectives or constraints are considered in the same priority level simultaneously through the use of weighting terms, making it a flexible framework.

IV. A Case Study: Adaptive Cruise Control

In this section, we use a simple case study, Adaptive Cruise Control (ACC), to show the issues of existing methods and the superiority of our proposed method.

TABLE II
Part of parameters in ACC simulations

Parameters	Value
m (kg)	1650
g (m/s^2)	9.81
T_h (s)	1.8
v_0 (m/s)	14
f_0, f_1, f_2	0.1, 5, 0.25
c_a, c_d	0.3, 0.3
λ	5
γ	5
H	$\frac{2}{1650^2}$
p	2×10^{-3}
ω_0	1
p_ω	0.2
Control frequency (Hz)	50

A. Problem Setup

Consider the scene where an ego vehicle moves along a one-lane road with a desired velocity v_d, and a lead vehicle drives at a speed of v_0 in front, as depicted in Fig. 2. The system model is formulated as below:

$$\dot{s} = \begin{bmatrix} v \\ -\frac{1}{m} F_r(v) \\ v_0 - v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} u \tag{19}$$

where $s = \begin{bmatrix} p & v & z \end{bmatrix}^T$. p, v are the position and velocity of the ego vehicle, respectively; z is the distance between the lead vehicle and the ego vehicle, m is the mass of the ego vehicle, and the friction $F_r(v) = f_0 + f_1 v + f_2 v^2$. The input of the ego vehicle u, which represents the accelerator or brake, is subject to

$$-m c_d g \leq u \leq m c_a g \tag{20}$$

Given a nominal controller $k(x) = F_r(v)$, the ego vehicle aims to achieve the desired velocity v_d and avoid collision from the lead vehicle. The CLF and CBF functions are designed the same as in [32]:

$$V(s) = (v - v_d)^2 \tag{21}$$

as a CLF and

$$h(s) = z - T_h v - \frac{1}{2} \frac{(v - v_0)^2}{c_d g} \tag{22}$$

as a CBF, where T_h is the look-ahead time.

We compare the results obtained from CLF-CBF QP, Optimal-decay CLF-CBF QP, and Safety-first CLF-CBF QP. Part of the fixed hyperparameters are as summarized in Table II. The solution feasibility and safety are influenced by the initial states and whether the performance objective conflicts with the safety requirements. To evaluate this, four scenarios with different initial states and target velocities are designed, and the corresponding settings are summarized in Table III.

B. Results Analysis

As shown in Fig. 3(a)-Fig. 3(b), in cases 1 and 2, the results of CLF-CBF QP and its optimal-decay variant are nearly identical (with overlapping curves). Since the initial state is safe in both cases, CBF constraints are relatively weak, making

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

TABLE III
SETTINGS OF FOUR SCENARIOS

Parameters	Case 1	Case 2	Case 3	Case 4
s_0	$[0, 20, 100]^T$	$[0, 20, 100]^T$	$[0, 20, 20]^T$	$[0, 20, 20]^T$
v_d (m/s)	10	24	10	24
Remark	Initial safe and $v_d < v_0$	Initial safe but $v_d > v_0$	Initial unsafe but $v_d < v_0$	Initial unsafe and $v_d > v_0$

all methods feasible and ensuring system safety. However, as observed from the speed and CLF curves, none of the methods achieve the desired decay rate (the curves exceed the set boundary), though the Safety-first CLF-CBF QP exhibits the fastest convergence. This validates that our proposed Safety-first method mitigates the performance degradation issues in existing methods.

In cases 3 and 4, the initial state is unsafe, which makes CLF-CBF QP infeasible. Similar to the previous cases, the Optimal-decay CLF-CBF QP exhibits performance degradation in case 3, while our proposed Safety-first method converges more quickly. More importantly, in case 4, the compromise in safety by the Optimal-decay CLF-CBF QP results in a collision (distance <0), whereas our method avoids the collision by applying maximum braking, even when the original CBF constraint is not satisfied. These results show our proposed Safety-first method mitigates the infeasibility and safety compromise issues in existing methods.

Furthermore, to verify that the Safety-first CLF-CBF QP is the limit-weighted form of the Unified CLF-CBF QP, we gradually increase the weighting coefficient p for the CLF slack variable in CLF-CBF QP, and compare it with our proposed method. As shown in Fig. 4, as the coefficient p increases, the velocity and CLF curves of CLF-CBF QP (from red to green) gradually approach that of our Safety-first method (blue). This validates our discussion on the limit-weighted form in Subsection III-C.

Remark 1. In ACC cases, our method behaves similarly to adopting minimal control directly, which may suffice as a fallback in control systems with low-dimensional control input where the relationship between the input and safety is monotonic and easily interpretable. However, in general multi-dimensional settings, this relationship becomes nontrivial, and minimal control no longer guarantees safety. In contrast, our Safety-first hierarchical approach provides a more principled and robust solution across diverse high-dimensional control scenarios.

V. Application to Safe Navigation

In this section, we tailor our Safety-first CLF-CBF QP to safe navigation tasks of mobile robots. Safety-first CLF-CBF QP is utilized to plan linear and angular velocities, referred to as the "Safety-first planner". Then, a novel geometric approximation technique based on constrained Delaunay Triangulation (CDT) is proposed to approximate irregularly-shaped obstacles via multiple circles, which is well-suitable for our Safety-first approach and applicable to unknown environments without requiring prior data. Finally, a ubiquitous issue of QP-based methods, undesired equilibria, is mitigated.

A. Safety-first Planner

Considering the kinematic model of mobile robots expressed as:

$$s = \begin{bmatrix} x & y & \theta \end{bmatrix}^T,\ \dot{s} = \begin{bmatrix} v\cos\theta & v\sin\theta & \Omega \end{bmatrix}^T \tag{23}$$

where the system state x, y, θ represents the 2D position and yaw angle, while the inputs v and Ω correspond to the linear and angular velocities, respectively. The constraints for velocities are represented as

$$v_{\min} \le v \le v_{\max}, \Omega_{\min} \le \Omega \le \Omega_{\max} \tag{24}$$

Given the goal position $p_g = (x_g, y_g)$, a CLF is designed for goal reaching as proposed in [33]

$$V(s) = \|p - p_g\|^2 - k_1[\mathcal{S}(p - p_g, \theta) - 1] \tag{25}$$

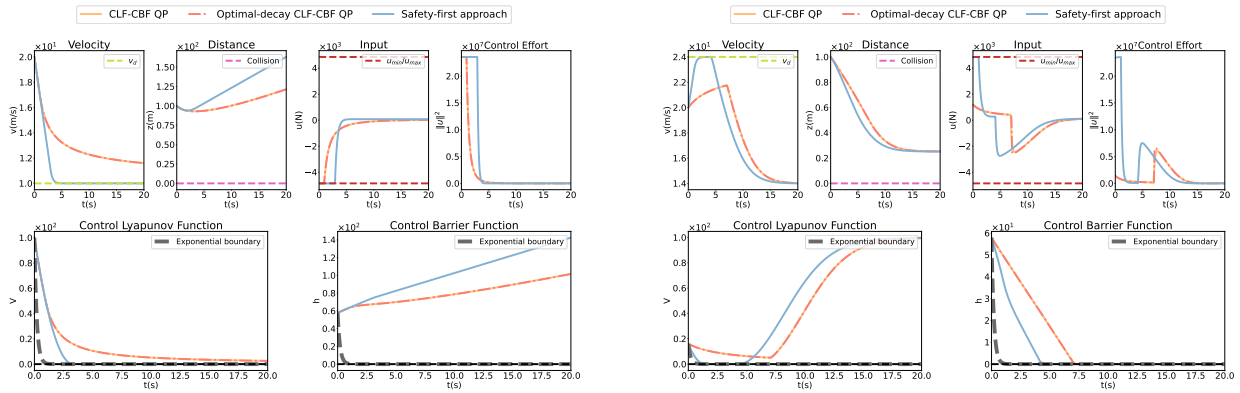
where $\mathcal{S}(p - p_g, \theta) = \frac{[\cos\theta \quad \sin\theta](p - p_g)}{\|p - p_g\|}$ represents the similarity between θ and the angle toward the target point, and k_1 is a factor for tuning.

B. CDT-based Circular Approximation

To define safety specifications for navigation, it is crucial yet challenging to design appropriate CBFs given detected obstacles. A common approach in safe navigation is to approximate obstacles with the smallest circumscribed circle or ellipse [18], and then design CBFs based on the distance of robots from these simple geometric shapes. However, as illustrated in Fig. 5(a) and Fig. 5(b), such approximations are imprecise and overly conservative for representing obstacles with complex shapes, which reduces the effectiveness of safety guarantees in cluttered environments.

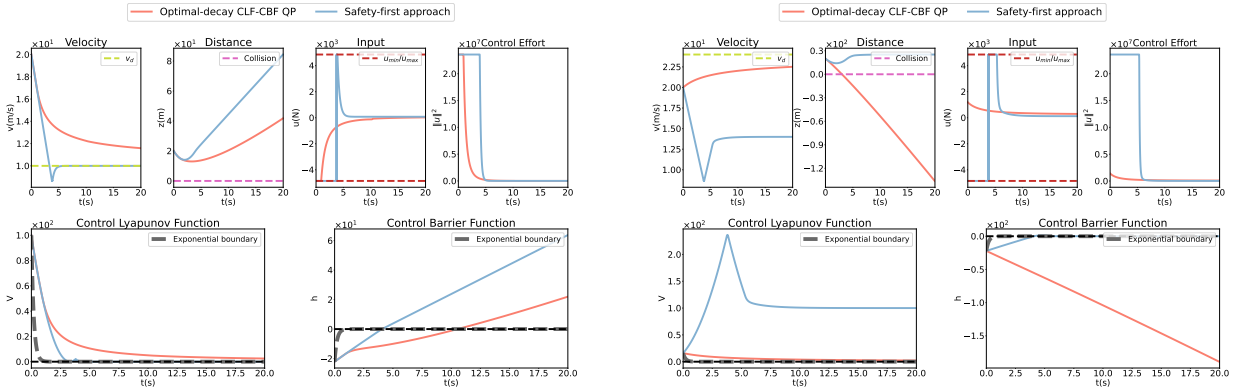
To balance the simplicity of circular approximation and the need of accurate obstacle representation, we propose a novel circular approximation approach based on constrained Delaunay triangulation (CDT). It uses multiple circles to represent obstacles with complex shapes, improving precision without being overly conservative. While the method increases the number of CBFs, and consequently the chance of constraint conflicts and the possibility of solution infeasibility in traditional CBF-based QP methods, it is particularly suitable for our proposed hierarchical optimization framework which ensures feasibility regardless of the number of constraints.

CDT is an extended form of Delaunay Triangulation for approximating non-convex structures. It performs triangulation on a given point set while satisfying the Delaunay properties as much as possible [34]. Our overall approach involves sampling the boundary points of obstacles, generating a triangular mesh using CDT, then computing the circumcircle of each triangle and clustering them to reduce their number. Finally, a CBF is designed for each circle to ensure obstacle avoidance. The algorithm process is summarized in **Algorithm** 1.



(a) Case 1: Initial safe and $v_d < v_0$

(b) Case 2: Initial safe but $v_d > v_0$



(c) Case 3: Initial unsafe but $v_d < v_0$

(d) Case 4: Initial unsafe and $v_d > v_0$

Fig. 3. Simulation results in ACC example.

It is important to emphasize that the Delaunay properties are highly effective for obstacle approximation. First, Delaunay triangulation divides the convex hull formed by the point set, so the generated triangulation can strictly cover the obstacle region by using only a subset of the boundary points. For non-convex shapes, constraints are defined to handle local non-convex structures. Additionally, Delaunay triangulation satisfies the *maximum minimum angle property* and the *empty circle property*, ensuring that the circumcircles obtained are very close to the original mesh, resulting in minimal conservatism, and that there is nearly no inclusion relationship between the circumcircles, meaning there are no redundant circles.

The number of circumcircles directly determines the number of CBFs, and an excessive number of CBF constraints increases computation time. Therefore, we merge the circumcircles involved in the approximation. Specifically, we arrange all the circles in descending order of their radii and sequentially check whether each circle "approximately" contains the remaining circles. The condition for this can be expressed as

$$\sqrt{(x_l - x_s)^2 + (y_l - y_s)^2} < r_l - r_s + r_{\text{thres}} \tag{26}$$

where $r_{\text{thres}} \geq 0$ is a predefined threshold, and (x_l, y_l, r_l), (x_s, y_s, r_s) represent the center coordinates and radius of the larger and smaller circles, respectively. If the larger circle approximately contains another, we remove all smaller contained circles and increase the radius of the larger one by r_{thres}. This process is repeated for each subsequent circle until all are considered. It effectively reduces the number of circles without significantly compromising the approximation accuracy. Furthermore, it can be proven that this method does not require iterative execution and can yield the final result in a single process.

After obtaining the approximate circles by CDT, we then design a CBF for each circle. For each circle represented by position $p_i = (x_i, y_i)$ and radius r_i, the CBF is set as

$$h_i(s) = \|p - p_i\| - r_i^2 + k_2[\mathcal{S}(p - p_i, \theta) - 1] \tag{27}$$

With CLF $V(s)$ and multiple CBFs $h_i(s)$, robot velocity $[v^*, \Omega^*]$ is planned using Safety-first CLF-CBF QP with multiple constraints as (18).

Remark 2. The threshold r_{thres} for merge balances the trade-off between approximation accuracy and the number of CBF constraints. A larger r_{thres} results in fewer constraints and lower computational complexity, but at the cost of reduced accuracy. A recommended choice is $r_{\text{thres}} \leq 2\min\{r_s\}$, where $\min\{r_s\}$ denotes the minimum radius among all circles. Under this choice, two circles are merged into a larger one only if they intersect. This prevents blockage of feasible paths caused by over-expansion due to merging, as illustrated in Fig. 6.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

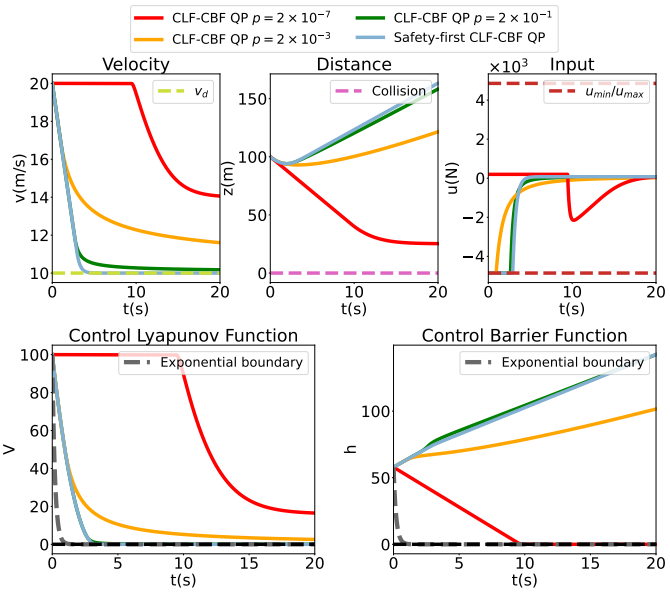


Fig. 4. Verification experiment on the relationship between Safety-first CLF-CBF QP and the limit-weighted form of Unified CLF-CBF QP. Increasing the weight p gradually (red, yellow, and green), the velocity and CLF curves converge toward, but never exactly equivalent with that of Safety-first CLF-CBF QP (blue).

Remark 3. The CDT-based approximation can be extended to 3D space using 3D constrained Delaunay triangulation [35]. Similar to 2D scenarios, 3D obstacles are approximated by a set of circumspheres corresponding to the tetrahedral elements of the triangulation. It still retains the advantages of facilitating CBF design with low approximation conservatism.

C. Undesirable Equilibria

One issue of QP-based methods is undesirable equilibria. Since the constraint generated from the CLF $V(x)$ used to ensure safe navigation is relaxed when solving the constrained optimization problem, it could cause the robot to stop before reaching the goal position. [36–38] studied and analyzed this issue, making adjustments to the CBFs and CLFs to mitigate the issue. For safe navigation, we also apply a simple improvement method as below.

According to the kinematic model (23), mobile robots will stop when $v = 0, \Omega = 0$ holds. As long as we ensure the velocity does not reach zero before it arrives at the goal position, it will not stop prematurely. Therefore, we modify the lower bound $v_{\min}$ to prevent the velocity from dropping to zero at positions other than the goal. On the other hand, if the robot enters a dangerous area, it becomes necessary to remove all control constraints, enabling the robot to reach safe areas quickly. Hence, we set

$$v_{\min} = \begin{cases} 0 & \exists h(x) < 0 \\ (1 - e^{-kV(x)}) v_{\max} & \text{otherwise} \end{cases} \tag{28}$$

to prevent potential undesirable equilibria.

Algorithm 1 CDT-based Circular Approximation

Input: Obstacles set $\{\mathcal{O}_i\}$
$\mathcal{R} = \{\}$
% Constrained Delaunay Triangulation
for each obstacle $\mathcal{O}_i$ **do**
  $\mathcal{R}_i = \{\}$
  Sample edge points to obtain the set $\mathcal{P}_i$
  Define edge and other constraints for CDT
  Get the triangulation $\mathcal{T}_i = \text{CDT}(\mathcal{P}_i)$
  for each triangle in $\mathcal{T}_i$ **do**
    Get the circumcircle represented by the location (x_o, y_o) and radius r_o
    $\mathcal{R}_i \leftarrow \mathcal{R}_i \cup \{(x_o, y_o, r_o)\}$
  end for
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_i$
end for
% Merging circles
$\mathcal{R} \leftarrow \textit{RadiusDescendingSort}(\mathcal{R})$
Set a merging threshold r_{thres}
$\mathcal{R}_{\text{merged}} = \{\}$
for each circle (x_l, y_l, r_l) in $\mathcal{R}$ **do**
  MERGED←Fasle
  for each remaining circle (x_s, y_s, r_s) **do**
    if $\sqrt{(x_l - x_s)^2 + (y_l - y_s)^2} < r_l - r_s + r_{\text{thres}}$ **then**
      Delete (x_s, y_s, r_s) from $\mathcal{R}$
      MERGED←True
    end if
  end for
  if MERGED then
    $r_l \leftarrow r_l + r_{\text{thres}}$
  end if
  $\mathcal{R}_{\text{merged}} \leftarrow \mathcal{R}_{\text{merged}} \cup (x_l, y_l, r_l)$
end for
Output: $\mathcal{R}_{\text{merged}}$

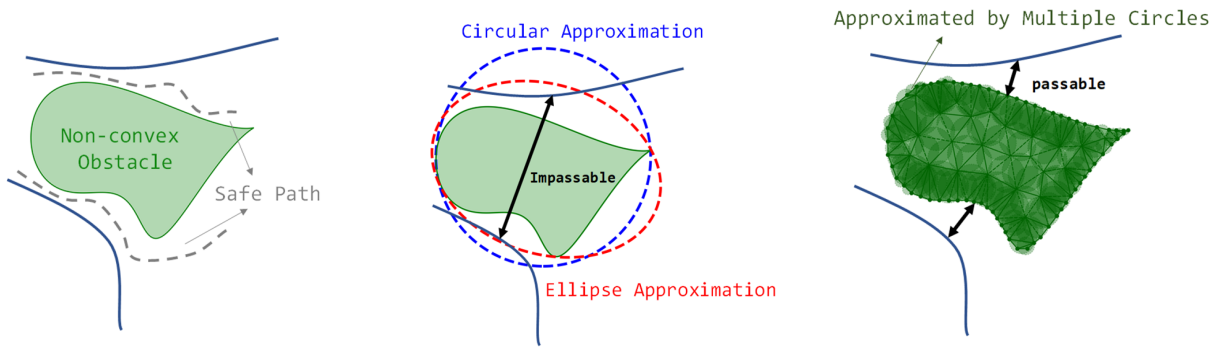
VI. Experimental Results

In this section, we first compare the computation time with varying numbers of obstacles to assess scalability in complex environments. Then, we validate our proposed safety-first method through real-world navigation experiments in three challenging environments: narrow corridors, multiple irregular obstacles, and dynamic obstacles, all under limited sensing range and without a prior map.

A. Computation Time and Scalability

Our Safety-first methods solve multiple QP problems hierarchically to obtain the final result. Meanwhile, the CDT-based approximation divides an obstacle into multiple circles, which increases the number of constraints in the optimization problem and leads to heavier computation load. Fortunately, since QP has relatively low solving complexity (common methods such as interior-point methods have polynomial-time computational complexity), it can still maintain a relatively fast speed within a certain number of obstacles. We evaluate the single-step computation time with different numbers of circular obstacles considered simultaneously using `cvxopt.solvers.qp` in

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.



(a) Original Shape (b) Common Approximation (c) CDT-based Circular Approximation

Fig. 5. The diagram of methods for approximating obstacles. a) The original obstacle is non-convex, making the passable area narrow, but there are still safe paths; b) common methods approximate obstacles using a circle or ellipse, resulting in no passable area due to excessive conservation; c) our method approximates the obstacle using multiple circles based on constrained Delaunay triangulation, providing a more precise result.

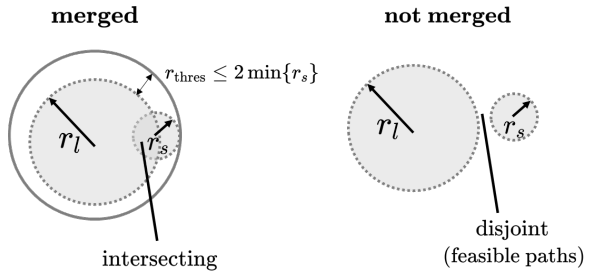


Fig. 6. An illustration of recommended choice of r_{thres}. Under the choice that $r_{\text{thres}} \leq 2\min\{r_s\}$, only intersecting circles may be merged, which prevents blockage of feasible paths caused by over-expansion.

TABLE IV
COMPUTATION TIME UNDER DIFFERENT NUMBERS OF CIRCLES

# of circles	1	5	10	20
CLF-CBF QP	1.7ms	2.1ms	2.4ms	5.4ms
Optimal-decay CLF-CBF QP	1.9ms	2.2ms	2.5ms	5.8ms
Safety-first CLF-CBF QP	3.7ms	7.5ms	12.9ms	23.5ms

Python on a PC with an Intel® i9-12900K CPU, and the results are presented in Table IV. Although the proposed method consumes slightly longer time than other methods with dense obstacles, it remains sufficient for real-time navigation (∼80Hz with 10 circular obstacles). In practice, obstacle avoidance is a short-range behavior and a limited number of obstacles exist in the limited sensing range of robots; thus our method has potential for robot navigation in complex environments.

B. Real-world Experiments

We design three representative scenarios, including *narrow corridors*, *multiple irregular obstacles*, and *dynamic obstacles*, to verify the effectiveness of our method. We use a motion capture system to locate the robot and obstacles. An Intel® NUC13 receives data from the motion capture system and plans the velocity based on our algorithm, and the robot itself is equipped with a Raspberry Pi 3B for velocity control. It is important to emphasize again that the robot navigates without a prior map, and only the positions of obstacle boundary points within the robot's sensing range (0.2m in our experiments),

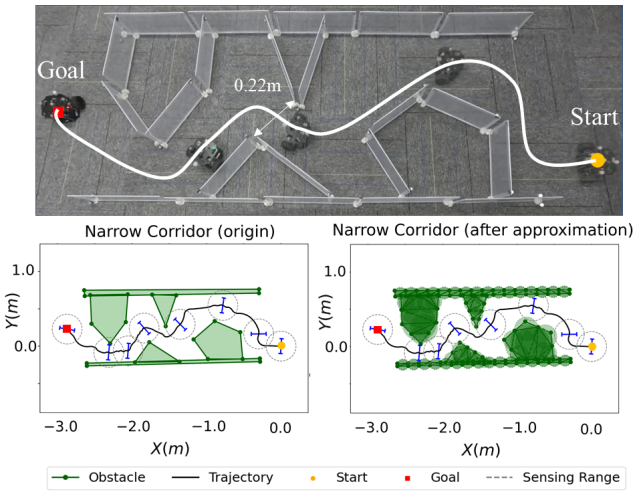
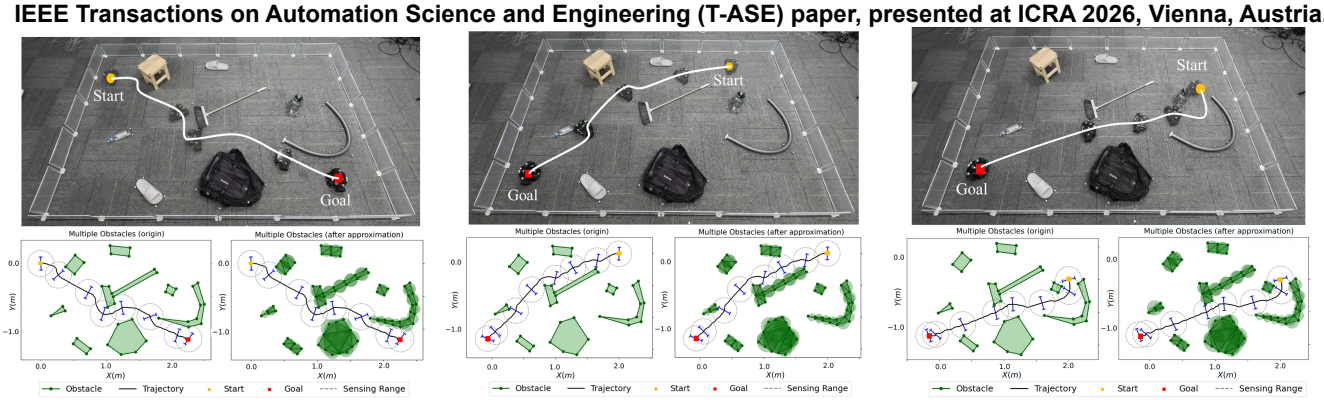


Fig. 7. Safe navigation in narrow corridor scenario. The minimum width of the corridor is 0.22m, slightly larger than the robot's width (about 0.2m). The CDT algorithm approximates obstacles relatively accurately, enabling our robot to navigate successfully.

together with the robot's pose, are provided by the motion capture system for real-time local planning. In such a way, the experiment tests the potential of our method for safe navigation in unknown new environments.

The main parameters set in the experiment are $\lambda = 2, \gamma_i = 5, H = \text{diag}[1 \quad 1], k_1 = 0.5, k_2 = 0.1, v_{\min} = 0, v_{\max} = 0.15\text{m/s}, \Omega_{\min} = -0.7, \Omega_{\max} = 0.7, k = 0.3$. For an unknown and varying number of CBFs and the CLF used for navigation, we set the priority order as follows: consider all CBFs from low to high priority first, then the CLF, and finally the minimization of velocity.

1) Narrow Corridors: As mentioned in Subsection V-B, the errors caused by classical approximation methods are too conservative to miss traversable areas and thus lead to complete failure in path planning. Hence, we design a challenging narrow corridor scenario with a **minimum width of only 0.22m, slightly larger than the robot's width (about 0.2m)**, as illustrated in Fig. 7. With our proposed CDT approximation method, obstacles are more accurately represented by multiple circles, allowing the robot to pass through the corridor safely



(a) (b) (c)

Fig. 8. Safe navigation in multiple irregular obstacles scenarios. For multiple obstacles that contain non-convex structures, our Safety-first planner still navigates the robot safely and successfully in three runs of different starting and goal points.

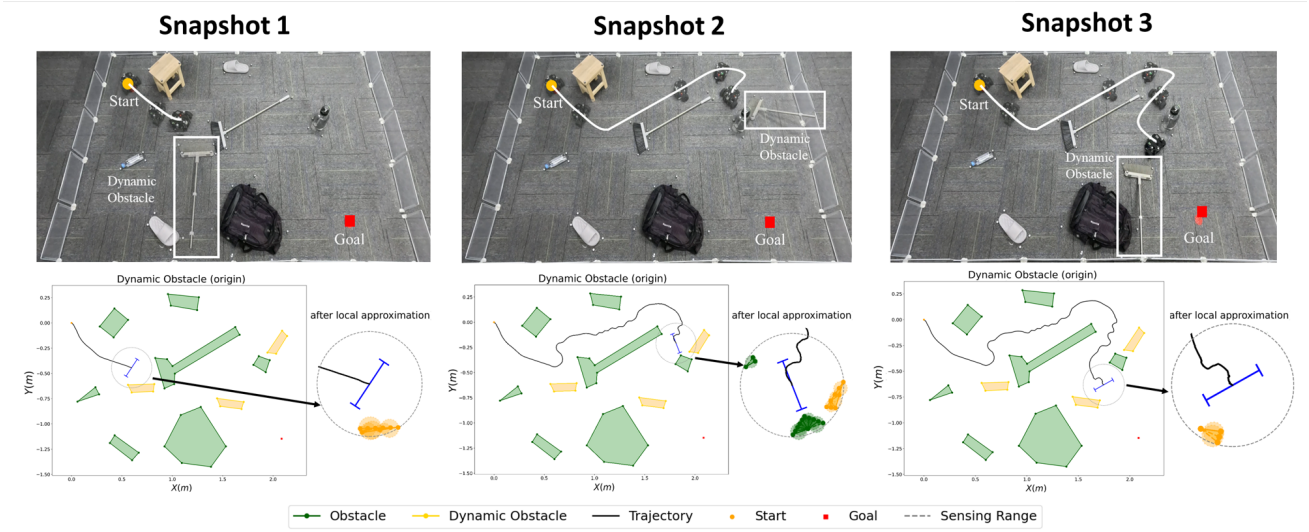


Fig. 9. Snapshots of safe navigation in dynamic obstacle scenarios. The dynamic obstacle (yellow) is moved to block the robot three times, but the Safety-first planner is able to avoid dynamic obstacles in time and replan the path until completing the navigation task.

and reach the goal.

2) Multiple Irregular Obstacles: We design a scenario with multiple irregular obstacles, including non-convex geometric shapes. To show the effectiveness of our proposed method, we run the experiments three times with different starting and ending positions, as shown in Fig. 8. All three runs are successful, validating the superiority of our method for safe navigation through irregular obstacles.

3) Dynamic Obstacles: We design an experiment scenario with dynamic obstacles, where the dynamic obstacles are enclosed within white boxes. During the navigation, we move the obstacle three times quickly to block its original path. The snapshots of navigation trajectories are depicted in Fig. 9, which shows that the robot can promptly change its direction to avoid the dynamic obstacles and eventually reach the goal. It is important to note that the robot does not receive any dynamic information; the ability to avoid dynamic obstacles is due to its real-time, efficient planning capability.

VII. Conclusion

In this paper, we propose a hierarchical optimization framework integrating CLFs and CBFs with guaranteed feasibility. The main idea is to prioritize safety over other performance objectives. We analyze the connections of our proposed method to existing CLF-CBF-based QPs and explain the reasons behind the related advantages and limitations with a verification through ACC simulations. Then we tailor the proposed framework to mobile robot navigation based on a novel approximation technique for irregularly-shaped obstacles, which is well-suitable for our method. Real-world experiments demonstrate the effectiveness of our proposed method in safe navigation through complex unknown environments with multiple and dynamic irregularly-shaped obstacles. This work focuses on model-based settings; however, the potential in model-free settings is also recognized, for example, in combination with reinforcement learning [39]. In future work, we will further investigate model-free cases to extend the framework to diverse and complex robotic tasks and applications.

Appendix A
Proof of Theorem 1

Proof. For 1), by choosing any $u_{\text{temp}} \in \mathcal{U}$, there exists $\delta_1 = L_f V(x) + L_g V(x) u_{\text{temp}} + \lambda V(x)$, $\delta_2 = L_f h(x) + L_g h(x) u_{\text{temp}} + \gamma h(x)$ satisfying all constraints in (14), i.e., (14) is feasible.

For 2), since (14) and all sub-problems of Safety-first CLF-CBF QP (10)-(12) are quadratic programs with linear constraints, the Karush-Kuhn-Tucker (KKT) conditions are the necessary and sufficient conditions for their optimal solution.

First, we prove the optimal solution of (14) satisfies the KKT conditions of Sub-problem 1 in (10). Noticing that the only distinction between (10) and (14) is the objective function, we divide the objective function of (14) by q^2 and then compare the gradient of Lagrange functions,

$$\begin{aligned} L_{\text{lw1}} - L_{\text{sf1}} &= \frac{1}{q}\delta_1 + \frac{1}{2q^2}u^T H u \\ \Rightarrow \nabla L_{\text{lw1}} - \nabla L_{\text{sf1}} &= \begin{bmatrix} \frac{1}{q^2}Hu \\ \frac{1}{q} \\ 0 \end{bmatrix} \end{aligned} \tag{29}$$

where $L_{\text{lw1}}, L_{\text{sf1}}$ are the Lagrange functions of (14) and (10) respectively. Since it is proved that (14) is feasible, then the optimal solution of (14) denoted as $[u', \delta_1', \delta_2']$ satisfies the KKT conditions $\nabla L_{\text{lw1}}(u', \delta_1', \delta_2') = \mathbf{0}$. When $q \to +\infty$, we have

$$\begin{aligned} \nabla L_{\text{sf1}}(u', \delta_1', \delta_2') &= \nabla L_{\text{lw1}}(u', \delta_1', \delta_2') - \lim_{q\to+\infty} \begin{bmatrix} \frac{1}{q^2}Hu' \\ \frac{1}{q} \\ 0 \end{bmatrix} \\ &= \mathbf{0}. \end{aligned} \tag{30}$$

It shows that the optimal solution of (14) converges to that of (10). However, only δ_2^* is passed to Sub-problem 2 in (11), so we need to prove why δ_1' is equivalent to the optimal solution δ_1^* of Sub-problem 2 in (11).

Due to $\delta_2^* = \delta_2'$ being the optimal solution for both two problems, we fix it and remove it from the objectives to be optimized, then (14) becomes the following equivalent form:

$$\begin{aligned} \min \quad & \frac{1}{2}u^T H u + q\delta_1^2 \\ \textbf{s.t.} \quad & L_f V(x) + L_g V(x) u + \lambda V(x) \le \delta_1 \\ & L_f h(x) + L_g h(x) u + \gamma h(x) \ge \delta_2^* \\ & u \in \mathcal{U} \end{aligned} \tag{31}$$

In the same manner, divide the objective function of (31) by q and denote Lagrange functions of (31) and Sub-problem 2 in (11) as L_{lw2} and L_{sf2}, respectively. When $q \to +\infty$, the difference between their gradients at $[u', \delta_1']$ is

$$\begin{aligned} \nabla L_{\text{lw2}}(u', \delta_1') - \nabla L_{\text{sf2}}(u', \delta_1') &= \lim_{q\to+\infty} \begin{bmatrix} \frac{1}{q}Hu' \\ 0 \end{bmatrix} \\ &= \mathbf{0} \end{aligned} \tag{32}$$

which shows δ_1' converges to δ_1^*. Similarly, u' converges to u^* can be proved as well after fixing $\delta_1 = \delta_1^*$ in (31) and comparing it with Sub-problem 3 in (12).

For 3), if $\delta_2 = 0$ is feasible with some solution $[u_0, \delta_{10}, 0]$, then for any feasible solution $[u, \delta_1, \delta_2]$ with $\delta_2 \neq 0$, there exists a constant $\kappa = \sqrt{\frac{u_0^T H u_0 + 2q\delta_{10}^2}{2\delta_2^2}}$ such that if $q > \kappa$, the value of the objective function, $\frac{1}{2}u^T H u + q\delta_1^2 + q^2\delta_2^2$, satisfies the inequality

$$\frac{1}{2}u^T H u + q\delta_1^2 + q^2\delta_2^2 \ge q^2\delta_2^2 > \frac{1}{2}u_0^T H u_0 + q\delta_{10}^2$$

which is the value of the objective function for the solution $[u_0, \delta_{10}, 0]$, i.e., $\delta_2' \to 0$.

Similarly, if $\delta_2 = \delta_1 = 0$ is feasible with some solution $[u_1, 0, 0]$, then for any solution $[u, \delta_1, \delta_2]$ with $\delta_1 \neq 0$ or $\delta_2 \neq 0$, there exists a constant $\kappa = \sqrt{\frac{u_1^T H u_1}{2\delta_2^2}}$ if $\delta_2 \neq 0$ or $\kappa = \frac{u_1^T H u_1}{\delta_1^2}$ if $\delta_1 \neq 0$, such that if $q > \kappa$, the value of the objective function, $\frac{1}{2}u^T H u + q\delta_1^2 + q^2\delta_2^2$, satisfies the inequality

$$\frac{1}{2}u^T H u + q\delta_1^2 + q^2\delta_2^2 \ge q\delta_1^2 + q^2\delta_2^2 > \frac{1}{2}u_1^T H u_1$$

which is the value of the objective function for the solution $[u_1, 0, 0]$, i.e., $\delta_2' \to 0$ and $\delta_1' \to 0$. ■

REFERENCES

[1] Y. Xiong, D.-H. Zhai, and Y. Xia, "Robust whole-body safety-critical control for sampled-data robotic manipulators via control barrier functions," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 16 050–16 061, 2025.

[2] M. Davoodi, A. Iqbal, J. M. Cloud, W. J. Beksi, and N. R. Gans, "Rule-based safe probabilistic movement primitive control via control barrier functions," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 3, pp. 1500–1514, 2023.

[3] J. Liu, M. Li, J. W. Grizzle, and J.-K. Huang, "Clf-cbf constraints for real-time avoidance of multiple obstacles in bipedal locomotion and navigation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 497–10 504.

[4] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." in *Robotics: Science and Systems*, vol. 13. Cambridge, MA, USA, 2017, pp. 1–10.

[5] M. Desai and A. Ghaffari, "Clf-cbf based quadratic programs for safe motion control of nonholonomic mobile robots in presence of moving obstacles," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2022, pp. 16–21.

[6] M. Srinivasan and S. Coogan, "Control of mobile robots using barrier functions under temporal logic specifications," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 363–374, 2021.

[7] L. Zhao, L. Wang, Y. Cao, Y. Yang, and S. Wen, "Learning-based fault-tolerant control with high-order control barrier functions," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 14 689–14 698, 2025.

[8] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "Barriernet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, 2023.

[9] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

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

[11] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 474–479.

[12] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety, and performance-critical control," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3021–3028, 2021.

[13] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[14] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, vol. 130, p. 109688, 2021.

[15] L. Zhao, L. Wang, Y. Cao, Y. Yang, and S. Wen, "Learning-based fault-tolerant control with high-order control barrier functions," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 14 689–14 698, 2025.

[16] J. Liu, J. Yang, J. Mao, T. Zhu, Q. Xie, Y. Li, X. Wang, and S. Li, "Flexible active safety motion control for robotic obstacle avoidance: A cbf-guided mpc approach," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2686–2693, 2025.

[17] Y. Tsai and T. Hsiao, "Local path planning of an autonomous mobile robot with nonholonomic constraints based on control barrier functions

and elliptical bounding box," in *2024 International Automatic Control Conference (CACS)*, 2024, pp. 1–6.

[18] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3679–3685.

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

[20] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 310–315, 2017.

[21] R. Funada, M. Santos, R. Maniwa, J. Yamauchi, M. Fujita, M. Sampei, and M. Egerstedt, "Distributed coverage hole prevention for visual environmental monitoring with quadcopters via nonsmooth control barrier functions," *IEEE Transactions on Robotics*, vol. 40, pp. 1546–1565, 2024.

[22] A. E. Chriat and C. Sun, "On the optimality, stability, and feasibility of control barrier functions: An adaptive learning-based approach," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7865–7872, 2023.

[23] O. So, Z. Serlin, M. Mann, J. Gonzales, K. Rutledge, N. Roy, and C. Fan, "How to train your neural control barrier function: Learning safety filters for complex input-constrained systems," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 532–11 539.

[24] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[25] K. Liu, L. Dong, X. Tan, W. Zhang, and L. Zhu, "Optimization-based flocking control and mpc-based gait synchronization control for multiple quadruped robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1929–1936, 2024.

[26] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2267–2281, 2022.

[27] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3856–3863.

[28] W. Xiao, C. Belta, and C. G. Cassandras, "Sufficient conditions for feasibility of optimal control problems using control barrier functions," *Automatica*, vol. 135, p. 109960, 2022.

[29] Y. Lyu, W. Luo, and J. M. Dolan, "Probabilistic safety-assured adaptive merging control for autonomous vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 764–10 770.

[30] J. Lee, J. Kim, and A. D. Ames, "Hierarchical relaxation of safety-critical controllers: Mitigating contradictory safety conditions with application to quadruped robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2384–2391.

[31] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[32] J. Choi, P. Kotaru, and B. Zhang, "Cbf-clf-helper," https://github.com/HybridRobotics/CBF-CLF-Helper.

[33] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *2016 American Control Conference (ACC)*, 2016, pp. 2252–2258.

[34] J. R. Shewchuk, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," in *Workshop on applied computational geometry*. Springer, 1996, pp. 203–222.

[35] P. R. Cavalcanti and U. T. Mello, "Three-dimensional constrained delaunay triangulation: a minimalist approach." in *IMR*, 1999, pp. 119–129.

[36] M. F. Reis, A. P. Aguiar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 731–736, 2021.

[37] M. Desai and A. Ghaffari, "Auxiliary control to avoid undesirable equilibria in constrained quadratic programs for trajectory tracking applications," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2078–2092, 2023.

[38] X. Tan and D. V. Dimarogonas, "On the undesired equilibria induced by control barrier function based quadratic programs," *Automatica*, vol. 159, p. 111359, 2024.

[39] J. Xie, S. Zhao, L. Hu, and H. Gao, "Certificated actor-critic: Hierarchical reinforcement learning with control barrier functions for safe navigation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 5482–5488.

Junjun Xie received the B.E. degree in automation from the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China, in 2023, where he is currently working toward the Ph.D. degree with the School of Intelligence Science and Engineering. His research interests include control theory and reinforcement learning for robot safety, with applications in mobile and legged robots.

Liang Hu received both the B.E. and M.E. degrees in Control Science and Engineering from Harbin Institute of Technology, China, in 2008 and 2010, respectively, and the Ph.D. degree in Computer Science from Brunel University London, UK, in 2016.

He is a professor with the School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen, China. His research interests include localisation, navigation, control and their applications in robotics and autonomous systems.

Yunzhe Tan received the B.E. degree in automation from the School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen, China, in 2025, where he is currently working toward the M.E. degree in control engineering. His research interests include safety-critical control and legged robot locomotion.

Jun Yang received the B.Sc. degree in automation from the Department of Automatic Control, Northeastern University, Shenyang, China, in 2006, and the Ph.D. degree in control theory and control engineering from the School of Automation, Southeast University, Nanjing, China, in 2011. He is currently a Reader in Autonomous and Electric Vehicles with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, UK. His research interests include disturbance observer, motion control, mechatronics, robotics and automation. Dr. Yang is also a fellow of IET and AAIA. He serves as an Associate Editor or a Technical Editor for IEEE Transactions on Industrial Electronics, IEEE/ASME Transactions on Mechatronics, and IEEE Open Journal of Industrial Electronics Society. He was a recipient of the EPSRC New Investigator Award.