Hierarchical Relaxation of Safety-critical Controllers: Mitigating Contradictory Safety Conditions with Application to Quadruped Robots

Jaemin Lee, Jeeseop Kim, and Aaron D. Ames

arXiv:2305.03929v1 [cs.RO] 6 May 2023

Abstract-The safety-critical control of robotic systems often must account for multiple, potentially conflicting, safety constraints. This paper proposes novel relaxation techniques to address safety-critical control problems in the presence of conflicting safety conditions. In particular, Control Barrier Function (CBFs) provide a means to encode safety as constraints in a Quadratic Program (QP), wherein multiple safety conditions yield multiple constraints. However, the QP problem becomes infeasible when the safety conditions cannot be simultaneously satisfied. To resolve this potential infeasibility, we introduce a hierarchy between the safety conditions and employ an additional variable to relax the less important safety conditions (Relaxed-CBF-QP), and formulate a cascaded structure to achieve smaller violations of lower-priority safety conditions (Hierarchical-CBF-QP). The proposed approach, therefore, ensures the existence of at least one solution to the QP problem with the CBFs while dynamically balancing enforcement of additional safety constraints. Importantly, this paper evaluates the impact of different weighting factors in the Hierarchical-CBF-QP and, due to the sensitivity of these weightings in the observed behavior, proposes a method to determine the weighting factors via a sampling-based technique. The validity of the proposed approach is demonstrated through simulations and experiments on a quadrupedal robot navigating to a goal through regions with different levels of danger.

I. INTRODUCTION

Robotic systems are being increasingly deployed to perform a wide-variety tasks in unstructured environments. including applications that necessitate operating in close proximity with people. Therefore, ensuring the safety of these systems while they operate in such environments is crucial. As such, there have been a variety of approaches to safety applied to a range of robot types, including: mobile robots [1]-[3], legged robots [4]-[6], multi-agent systems [7], [8], and even humanoid robots [9]-[11]. As of late, Control Barrier Functions (CBFs) [12] have emerged as a popular method for practically enforcing safety constraints on robotic systems. However, as the environments and tasks become more complicated, safety constraints become contradictory or conflicting. Therefore, it is essential to manage such contradictory safety statements to ensure safe and performant controllers in obstacle-rich environments as illistrated in Fig. 1. With this motivation, this paper addresses the potential infeasibility of Quadratic Programming (QP) that include contradictory CBF constraints and proposes a hierarchical approach for synthesizing feedback controllers and safety filters handling multiple safety constraints.



Fig. 1. **Motivation of our study**: When robots operate in complex environments with numerous obstacles, they are often required to follow trajectories that pass through unsafe regions. Such scenarios give rise to challenges in control and planning to ensure the safety of the robots.

A. Related Work

For robotic applications, CBF-based approaches have proven to be an effective method for generating safe robot motions in the context of planning, control, and learning. To provide an example, for legged robots safety-critical CBF constraints can be enforced to obtain safe foot placements [13], coupled with learning [14], and combed with MPC in a multi-rate fashion [15]. Manipulators can be controlled with CBFs to enforce box constraints in operational space [16], or avoid collisions with the environment [17], [18]. In the whole-body control of humanoid robots, joint limits and self-collision avoidance constraints can be imposed using CBFs [10]. However, these studies assume the existence of a safe set satisfying the safety-critical constraints over the entire time horizon. In reality, many robots operate with contradictory safety statements. This fundamental issue dates back to the control theoretic origins of CBFs.

At their inception [12], [19], CBFs were introduced as a control theoretic tool to ensure safety framed as forward set invariance. A CBF results in an affine inequality constraint in the input that, when enforced, implies safety; since this constraint in affine, it can be enforces in a QP with a cost that minimizes the difference between the desired input and the safe input—the QP nature means it can be solved in real-time. CBFs can also naturally be combined with Control Lyapunov Functions (CLFs), since these again yield an inequality constraint affine in the input—CLFs and CBF can thus be utilized in a single QP to enforce stability and safety under the assumption of feasibility. In the case when the CLF and CBF constraints conflict, the CLF constraint is typically relaxed to strictly fulfill the safety-critical constraints at the

This work is supported by Dow under the project #227027AT.

The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA (e-mail: {jaemin87, jeeseop, ames}@caltech.edu)

cost of goal attainment. This interplay between conflicting CLFs and CBFs was studied in [20] by relaxing CLF constraint and scaling the lower bound of CBF constraint. This naturally leads to the setting of multiple barrier functions.

Multiple barrier functions have been widely studied in the context of multi-agent robotic systems, with Boolean compositions of safety-critical constraints formulated using sums and products of barrier functions [21]. For multiagent systems, non-smooth barrier functions can also be used with the max and min operators for Boolean compositions [22]. Linear Temporal Logic (LTL) tasks are achieved by satisfying multiple time-varying barrier functions [23], and the approach has been extended to multi-agent systems considering conflicts of Temporal Logic tasks [24]. However, none of these approaches consider a hierarchy among the safety-critical constraints. Recently, a decoupling method for multiple CBF constraints has been proposed to compose multiple CBFs while enforcing input constraints [25]. However, this method does not impose a hierarchy of safety-critical constraints either. This paper, therefore, considers potentially conflicting CBFs, and propose a relaxation achieved by imposing a hierarchy based on their relative importance.

B. Contributions

This paper leverages a two-layered control architecture for robotic systems that includes a reduced-order model (ROM) to generate control inputs that result in safe paths, and a full-order model (FOM) that leverage the nonlinear dynamics of the system to track these paths. The proposed approach focuses on principled means of relaxing hierarchical safetycritical constraints, expressed as CBFs in a QP. In particular, we begin with the assumption that multiple safety conditions cannot be simultaneously satisfied and that a hierarchy exists among them. The goal is then to strictly fulfill the top-priority safety condition while minimizing lower-priority violations, via a hierarchical architecture similar to [26]. Additionally, the approach analyzes the weighting factors between the feedback control input error and the relaxation variable and proposes a technique to determine these weighting in the context of achieving a given task.

The main contributions of this paper are threefold. First, the proposed approach ensures that the formulated Hierarchical-CBF-QP is always feasible while strictly satisfying the top-prioritized safety-critical constraint. In addition, we reduce the violations of the lower-prioritized CBFs in order of their importance. Second, the approach includes a technique to determine the weighting factors between the nonlinear feedback control input error and the relaxation variables. Instead of defining new candidate functions that lower bound the CBF constraints (extended class K functions), we still utilize the same functions and obtain optimal parameters of the weightings in Hierarchical-CBF-QP, minimizing the cost function. Lastly, the proposed method enables to the generation of safer paths without the need for re-planning trajectories, particularly in crowded obstacle-rich environments. We demonstrate this experimentally.

Our paper is organized as follows. In Section II, provide the necessary background on the nonlinear modelling



Fig. 2. **Control Framework:** Our relaxation approaches are crucial components of the control framework described earlier. To ensure a fair comparison of these methods, we have used consistent components and setups in our implementation.

and control of robots, and frame the safety-critical control approach, including the introduction of CBFS and safety filters. Section III outlines the proposed approach for relaxing safety-critical controllers and thereby incorporating contradictory CBF conditions. This section also includes a technique for determining the appropriate weighting factors to optimize performance. The proposed approach is also extended to handle arbitrary CBF constraints in Section III. Finally, Section IV provides both simulation and experiment results of the proposed approach applied to a quadrupedal robot platform (A1) walking in a cluttered environment.

II. PROBLEM STATEMENT

This section covers the fundamental concepts that underpin the proposed control architecture for robotic systems described in Fig. 2: the control-affine system (ROM), the rigid-body dynamic model (FOM), the feedback controllers, and the concept of CBF. With these concepts in place, we define our problem, making proper assumptions to ensure the effectiveness of the proposed control architecture.

A. Reduced-order Model and Feedback control

We introduce control affine systems defined in a state space $\mathcal{X} \subseteq \mathbb{R}^n$ and an admissible input space $\mathcal{U} \subseteq \mathbb{R}^m$:

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

where $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{u} \in \mathcal{U}$ denote the state and control input. In addition, $f : \mathcal{X} \mapsto \mathbb{R}^n$ and $g : \mathcal{X} \mapsto \mathbb{R}^m$ are Lipschitz continuous. In this paper, the above control affine system is considered as ROM such as the linear inverted pendulum or double integrator models. To design a continuous feedback controller, we use a simple PD control law with proportional and derivative gains, \mathbf{K}_p and \mathbf{K}_d , given a state trajectory, $\boldsymbol{x}^d(t), \forall t \in [t_0, t_f]$:

$$\boldsymbol{u}(t) = \mathbf{K}_p \boldsymbol{e}_x(t) + \mathbf{K}_d \dot{\boldsymbol{e}}_x(t)$$
(2)

where $e(t) = x^d(t) - x(t)$. This control input will be the desired input for planning and control based on the full-order

model. The state of ROM is computed and updated based on measurements or the state update of FOM.

B. Full-order Model, Planning, and Task-Space Control

To realize the control input, we employ full-body rigid body dynamics and intermediate planners that connect ROM and FOM. For instance, it is needed to plan the foot placement while implementing the control input u(t) with Raibert's heuristic [27]. Since the planning of footholds is outside the scope of this paper, we assume that the task trajectory $y^d(t) \in \mathbb{R}^{n_y}$ is properly planned in terms of u(t).

The equation of motion for floating-base robots with contacts is typically presented as follows:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^{\top} \boldsymbol{\tau} + \mathbf{J}_{c}(\mathbf{q})^{\top} \mathbf{F}_{c}, \qquad (3)$$

$$\mathbf{J}_{c}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{c}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0}$$
(4)

where $\mathbf{q} \in \mathcal{Q} \subset \mathbb{R}^{n_q}$, $\mathbf{D}(\mathbf{q}) \in \mathbb{S}_{>0}^{n_q}$, $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n_q}$, $\mathbf{S} \in \mathbb{R}^{(n_q-6) \times n}$, and $\boldsymbol{\tau} \in \Gamma \subset \mathbb{R}^{n_q-6}$ denote the joint variable, mass/inertia matrix, the sum of Coriolis/centrifugal and gravitational force, a selection matrix, and a control torque command, respectively. In addition, $\mathbf{F}_c \in \mathbb{R}^{n_c}$ and $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{n_c \times n_q}$ represent the contact wrench and the corresponding Jacobian, respectively. For rigid contacts, we consider an equality constraint (4) in the acceleration level. With \boldsymbol{y}^d , $\dot{\boldsymbol{y}}^d$, \mathbf{q} , and $\dot{\mathbf{q}}$, we formulate a QP problem to obtain the desired control command to minimize the tracking error:

Task-Space Controller:

$$\min_{(\ddot{\mathbf{q}}, \tau, \mathbf{F}_c)} \quad \mathcal{J}_y = \|\mathbf{K}_p(\boldsymbol{y}^a - \boldsymbol{y}) + \mathbf{K}_d(\dot{\boldsymbol{y}}^a - \dot{\boldsymbol{y}}) - \mathbf{J}_y(\mathbf{q})\ddot{\mathbf{q}}\|^2$$
s.t. $\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^\top \mathbf{F}_c,$
 $\mathbf{J}_c(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = 0,$
 $\mathbf{F}_c \in \mathcal{CWC}(\mathbf{q}), \quad \boldsymbol{\tau} \in \Gamma$

 $CWC(\mathbf{q})$ is the set of the contact forces satisfying the contact wrench cone constraint. The control of the legged robots is achieved through a joint-impedance controller, which integrates the joint feedback control with a feedforward torque obtained from the solution to the aforementioned QP.

C. Control Barrier Functions (CBFs)

This section provides a brief overview of the fundamental concepts and definitions related to CBFs. We introduce a safe set $C \subset \mathbb{R}^n$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \mapsto \mathbb{R}$, its boundary and interior sets:

$$\mathcal{C} \coloneqq \{ \boldsymbol{x} \in \mathbb{R}^{n} : h(\boldsymbol{x}) \ge 0 \}, \\ \partial \mathcal{C} \coloneqq \{ \boldsymbol{x} \in \mathbb{R}^{n} : h(\boldsymbol{x}) = 0 \}, \\ \text{int}(\mathcal{C}) \coloneqq \{ \boldsymbol{x} \in \mathbb{R}^{n} : h(\boldsymbol{x}) > 0 \}.$$
(6)

The nonlinear system is safe with respect to the safe set C if C is forward invariant. To recall the basic concepts and definitions related to CBFs, a continuous function α : $(a,b) \mapsto \mathbb{R}$ is an *extended class* \mathcal{K} (\mathcal{K}^e) function if it is strictly increasing with $\alpha(0) = 0$. In particular, if the open interval is $(-\infty, \infty)$ and α is strictly increasing with $\alpha(0) = 0$, $\lim_{r \to -\infty} \alpha(r) = -\infty$, and $\lim_{r \to +\infty} \alpha(r) = +\infty$, α is an extended class \mathcal{K}_{∞} (\mathcal{K}_{∞}^{e}) function. Using the function α , CBFs are defined according to [12].

Definition 1. Let $C \subset \mathbb{R}^n$ be the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \mapsto \mathbb{R}$ as in (6). h is a **Control Barrier Function (CBF)** if there exists a function $\alpha \in \mathcal{K}_{\infty}^e$ such that for all¹ $x \in \mathbb{R}^n$:

$$\sup_{\boldsymbol{u}\in\mathbb{R}^m}\dot{h}(\boldsymbol{x},\boldsymbol{u})=\sup_{\boldsymbol{u}\in\mathbb{R}^m}\left[\mathcal{L}_fh(\boldsymbol{x})+\mathcal{L}_gh(\boldsymbol{x})\boldsymbol{u}\right]\geq-\alpha(h(\boldsymbol{x})).$$

where $\mathcal{L}_f h(\boldsymbol{x}) = \frac{\partial h}{\partial \boldsymbol{x}}(\boldsymbol{x}) f(\boldsymbol{x})$ and $\mathcal{L}_g h(\boldsymbol{x}) = \frac{\partial h}{\partial \boldsymbol{x}}(\boldsymbol{x}) g(\boldsymbol{x})$ are Lie derivatives. Given a CBF, we define a set of control input values for all $\boldsymbol{x} \in \mathbb{R}^n$ as follows:

$$\mathcal{K}_{\mathrm{cbf}}(\boldsymbol{x}) = \left\{ \boldsymbol{u} \in \mathbb{R}^m : \mathcal{L}_f h(\boldsymbol{x}) + \mathcal{L}_g h(\boldsymbol{x}) \boldsymbol{u} \geq -\alpha(h(\boldsymbol{x})) \right\}.$$

Let u^* be the desired control value. We can find the (pointwise) closes safe control value through the use of a CBF framed in the context of the following QP:

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u} \in \mathcal{U}}{\operatorname{arg\,min}} \|\boldsymbol{u}^{\star} - \boldsymbol{u}\|^{2},$$

s.t. $\dot{h}(\boldsymbol{x}, \boldsymbol{u}) \ge -\alpha(h(\boldsymbol{x})).$ (7)

The key result of [12] is that \hat{u} renders the set C forward invariant (that is, safe), i.e., for all $x_0 \in C$ it follows that $x_0 \in C$ for all time when \hat{u} is applied to (1).

D. Problem Definition

In this paper, let us consider two barrier functions h_1 : $\mathbb{R}^n \mapsto \mathbb{R}$ and h_2 : $\mathbb{R}^n \mapsto \mathbb{R}$, in turn the corresponding safe sets are defined as C_1 and C_2 . Based on the defined CBFs, we formulate a standard CBF-QP given the feedback control input u^* .

CBF-QP (Two CBFs):

$$egin{aligned} oldsymbol{u}^ullet &= rgmin_{oldsymbol{u}\in\mathcal{U}} & \|oldsymbol{u}^\star-oldsymbol{u}\|^2, \ ext{s.t.} & \dot{h}_1(oldsymbol{x},oldsymbol{u}) \geq -lpha_1(h_1(oldsymbol{x})), & ext{(8a)} \ & \dot{h}_2(oldsymbol{x},oldsymbol{u}) \geq -lpha_2(h_2(oldsymbol{x})) & ext{(8b)} \end{aligned}$$

When the feedback control command u^* satisfies all constraints without conflicting each other, u^{\bullet} becomes original feedback control input u^* . However, the constraints frequently conflict with each other so we address specific assumptions for our problem, which is potentially infeasible CBF-QP.

To define our problem, we first define the following sets using C_1 and C_2 :

$$\mathcal{C}_{\cap} = \left\{ \boldsymbol{x} \in \mathbb{R}^{n} : h_{1}(\boldsymbol{x}) \ge 0 \land h_{2}(\boldsymbol{x}) \ge 0 \right\}, \\ \mathcal{C}_{\cup} = \left\{ \boldsymbol{x} \in \mathbb{R}^{n} : h_{1}(\boldsymbol{x}) \ge 0 \lor h_{2}(\boldsymbol{x}) \ge 0 \right\}$$
(9)

where \land and \lor denote logical operators AND and OR, respectively. It is true that $C_{\cap} = C_1 \bigcap C_2$ and $C_{\cup} = C_1 \bigcup C_2$. For proof of this, see Lemma 3.1 in [21]. One of the critical assumptions in our problem definition is as follows:

¹Here we consider all points in \mathbb{R}^n for notational simplicity. Technically, one could consider a subset E containing C and restrict our attention to this set E. One would only need to check the CBF conditions on this set E.



Fig. 3. Sets in our problem definition: Given safety-critical constraints, the sets defined in this paper, C_1 , C_2 , C_{\cap} , and $C_1 - C_{\cap}$, are described visually.

Assumption 1. Consider two properly defined class \mathcal{K}^e_{∞} functions α_1 and α_2 . We assume that there exists a state \boldsymbol{x} along the trajectory such that $\boldsymbol{x} \in C_1 - C_{\cap}$, which means $h_1(\boldsymbol{x}) > 0$ and $h_2(\boldsymbol{x}) < 0$, as shown in Fig. 3.

If the current state x satisfies Assumption 1, the potential for infeasibility arises due to a conflict between the constraints (8a) and (8b), simultaneously.

Proposition 1. Under Assumption 1, if a state x is given, there exist α_1 and α_2 such that no control input from $\mathcal{K}_{cbf,1}(x)$ belongs to $\mathcal{K}_{cbf,2}(x)$.

Proof. Let us consider $h_2(x) = -\beta_1(x)h_1(x) + \beta_2(x)$ where $\beta_1(x) > 0$ and $\beta_2(x) < \beta_1(x)h_1(x)$. In addition, it is noted β_1 and β_2 are the class C^1 functions. The time derivatives of $\beta_1(x)$ and $\beta_2(x)$ becomes as follows:

$$\frac{\partial \beta_{1,2}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \frac{\partial \beta_{1,2}}{\partial \boldsymbol{x}} \left(f(\boldsymbol{x}) + g(\boldsymbol{x}) \boldsymbol{u} \right) = \dot{\beta}_{1,2}(\boldsymbol{x}, \boldsymbol{u}).$$

Then, the time derivative of $h_2(x)$ becomes

$$egin{aligned} \dot{h}_2(oldsymbol{x},oldsymbol{u}) &= -\dot{eta}_1(oldsymbol{x},oldsymbol{u})h_1(oldsymbol{x}) - eta_1(oldsymbol{x})\dot{h}_1(oldsymbol{x},oldsymbol{u}) &+ \dot{eta}_2(oldsymbol{x},oldsymbol{u}) \ &\leq -\dot{eta}_1(oldsymbol{x},oldsymbol{u})h_1(oldsymbol{x}) + eta_1(oldsymbol{x})lpha_1(oldsymbol{h}_1(oldsymbol{x})) &+ \dot{eta}_2(oldsymbol{x},oldsymbol{u}) \ &\leq -\dot{eta}_1(oldsymbol{x},oldsymbol{u})h_1(oldsymbol{x}) + eta_1(oldsymbol{x})lpha_1(oldsymbol{x})) &+ \dot{eta}_2(oldsymbol{x},oldsymbol{u}) \ &\leq -\dot{eta}_1(oldsymbol{x},oldsymbol{u})h_1(oldsymbol{x}) + eta_1(oldsymbol{x})lpha_1(oldsymbol{x})) &+ \dot{eta}_2(oldsymbol{x},oldsymbol{u}). \end{aligned}$$

If the defined α_2 satisfies the following inequality for all control inputs $\boldsymbol{u} \in \mathcal{K}_{cbf,1}$,

$$\dot{eta}_1({m{x}},{m{u}})h_1({m{x}}) - eta_1({m{x}})lpha_1(h_1({m{x}})) - \dot{eta}_2({m{x}},{m{u}}) > lpha_2(h_2({m{x}}))$$

there is no \boldsymbol{u} satisfying $\dot{h}_2(\boldsymbol{x}, \boldsymbol{u}) \geq -\alpha_2(h_2(\boldsymbol{x}))$ in $\mathcal{K}_{\mathrm{cbf},1}(\boldsymbol{x})$. Therefore, any control inputs from $\mathcal{K}_{\mathrm{cbf},1}(\boldsymbol{x})$ do not belong to $\mathcal{K}_{\mathrm{cbf},2}(\boldsymbol{x})$.

As noted in Proposition 1, the CBF-QP (Two-CBFs) problem is infeasible, which means that u^{\bullet} does not exist. To address this issue, we need to relax the QP problem, In the relaxation process, we introduce a new assumption:

Assumption 2. Under Assumption 1, the first safety condition is more critical than the other, so it must be strictly satisfied while tracking the trajectory, $\mathbf{x}(t) \in C_1$ for all time.

Assuming that the conditions specified in Assumptions 1 and 2 are met, we present methods to obtain control inputs that prioritize the satisfaction of safety-critical constraints.



Fig. 4. Intuitions for the relaxation approaches: Relaxed-CBF-QP (R-CBF-QP) aims to reduce the size of the safe set $C_1 - C_{\cap}$ by adjusting the set C_2 . On the other hand, Hierarchical-CBF-QP (H-CBF-QP) combines the optimized control input u^* with $\dot{h}_2(x, \hat{u}_2)$ to generate the appropriate control input.

III. RELAXATION OF CBF-QP

In this section, we present our proposed relaxation approaches for a scenario with two CBFs, Relaxed-CBF-QP and Hierarchical-CBF-QP. Then, we examine the impact of the weighting factors of QP formulations on the robot's behavior. In addition, we discuss a potential approach for determining the weighting factor of safety relaxation. We then generalize the approach to accommodate arbitrary CBFs.

A. Two CBFs Case

To ensure the feasibility of the CBF-QP problem, we modify the CBF constraint (8b) under Assumptions 1 and 2. Specially, we define a minimum constant value ϵ for the secondary safety-critical constraint to ensure that $x \in C_{\cap}$.

Definition 2. For a state $x \in C_1 - C_{\cap}$, let $\mathbf{C}(x)$ be the set of constant values as follows:

$$\mathbf{C}(\boldsymbol{x}) \coloneqq \{ c \in \mathbb{R}_{\geq 0} : h_1(\boldsymbol{x}) \ge 0 \land h_2(\boldsymbol{x}) \ge -c \}.$$
(10)

Then, the minimum value among the elements of $\mathbf{C}(\mathbf{x})$ is defined as $\epsilon \coloneqq \min(\mathbf{C}(\mathbf{x}))$ as depicted in Fig. 4.

In other words, if $h_2(x)$ is greater than or equal to $-\epsilon$, then the state x satisfies both safety conditions and is in the safe set $C \cap$. Instead of using $h_2(x)$ in the CBF constraint, we introduce new barrier function with an offset as $h'_2(x) =$ $h_2(x)+c$ where $C'_2 = \{x \in \mathbb{R}^n : h'_2(x) \ge 0\}$. The secondary safety-critical constraints can then be expressed as following inequality constraints

$$\begin{aligned} h_2'(\boldsymbol{x}, \boldsymbol{u}) &= \mathcal{L}_f(h_2(\boldsymbol{x}) + c) + \mathcal{L}_g(h_2(\boldsymbol{x}) + c)\boldsymbol{u} \\ &= \dot{h}_2(\boldsymbol{x}, \boldsymbol{u}) \geq -\alpha_2(h_2(\boldsymbol{x}) + c) \end{aligned}$$

where $\mathcal{L}_f c = 0$ and $\mathcal{L}_g c = 0$, respectively. We introduce an additional relaxation variable δ to make the CBF-QP problem feasible, which satisfies $\alpha_2(h_2(\boldsymbol{x}) + c) = \alpha_2(h_2(\boldsymbol{x})) + \delta$, which means

$$c = \alpha_2^{-1}(\alpha_2(h_2(\boldsymbol{x})) + \delta) - h_2(\boldsymbol{x}) \ge \epsilon.$$

If we already know ϵ , then the relaxation variable is bounded as $\delta \ge \alpha_2(h_2(\boldsymbol{x}) + \epsilon) - \alpha_2(h_2(\boldsymbol{x}))$. Otherwise, we can simply set $\delta \ge 0$ in the optimization problem. Now, we formulate the first relaxed approach called Relaxed-CBF-QP as follows:

Relaxed-CBF-QP (R-CBF-QP):

$$oldsymbol{u} = rgmin_{oldsymbol{u}\in \mathcal{U},\;\delta\in\mathbb{R}_{\geq 0}} \quad \mathcal{J}_r = \lambda_u \|oldsymbol{u}^\star - oldsymbol{u}\|^2 + \lambda_\delta\delta^2,$$

s.t.
$$\dot{h}_1(x, u) \ge -\alpha_1(h_1(x)),$$
 (11a)

$$h_2(\boldsymbol{x}, \boldsymbol{u}) \ge -\alpha_2(h_2(\boldsymbol{x})) - \delta,$$
 (11b)

where $\lambda_{\delta} \geq 0$ and $\lambda_u \geq 0$ are the weighting factors. The solution to the above QP problem depends on the relative weighting factors in the cost function.

Remark 1. Let us consider $\lambda_u = 0$ and $\lambda_\delta > 0$. As per Definition 2 and the modified CBF constraint (11b), the solution to the *R*-CBF-QP problem satisfies (11b) with the lower bound $\alpha_2(h_2(\mathbf{x}) + \epsilon)$. Therefore, we have $\delta^* =$ $\alpha_2(h_2(\mathbf{x}) + \epsilon) - \alpha_2(h_2(\mathbf{x}))$, which does not incorporate any feedback control effort.

Corollary 1. When $\lambda_u > 0$ and $\lambda_{\delta} = 0$, the constraint (11b) vanishes. Thus, the control input from the solution to *R*-*CBF*-*QP* is identical to one obtained by solving *CBF*-*QP* formulated in (7) considering $h_1(\mathbf{x})$.

As mentioned in the problem definition, our objective is to partially satisfy the secondary safety-critical constraints while strictly enforcing the primary constraint. So, we obtain \hat{u}_k for each safety-critical constraint by solving (7). Then, we employ the solution \hat{u}_k to compute $\dot{h}_k(x, \hat{u}_k)$. Next, we formulate a QP problem with an equality constraint that enforces the hierarchy between the safety-critical constraints under Assumption 2.

Hierarchical-CBF-QP (H-CBF-QP):

$$\begin{aligned} \boldsymbol{u}^{*} &= \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U},\ \delta\in\mathbb{R}} \quad \mathcal{J}_{h} = \lambda_{u} \|\boldsymbol{u}^{*} - \boldsymbol{u}\|^{2} + \lambda_{\delta}\delta^{2}, \\ \text{s.t.} \quad \dot{h}_{1}(\boldsymbol{x},\boldsymbol{u}) \geq -\alpha_{1}(h_{1}(\boldsymbol{x})), \qquad (12a) \\ \dot{h}_{2}(\boldsymbol{x},\boldsymbol{u}) + \delta = \dot{h}_{2}(\boldsymbol{x},\hat{\boldsymbol{u}}_{2}), \qquad (12b) \end{aligned}$$

Because of Assumptions 1 and 2, δ cannot be zero in the H-CBF-QP. So, we enforce the constraint (12b) to minimize the difference between $\dot{h}_2(x, u)$ and $\dot{h}_2(x, \hat{u}_2)$ as shown in Fig. 4.

Corollary 2. Like Corollary 1, the constraint (12b) disappears when $\lambda_u > 0$ and $\lambda_{\delta} = 0$.

Proposition 2. When $\lambda_u = 0$ and $\lambda_\delta > 0$, *H-CBF-QP* becomes (7) with a cost $\mathcal{J}_j = \| \boldsymbol{u}^* - \boldsymbol{u} \|_{\mathbf{M}}^2 + \boldsymbol{p}^\top \boldsymbol{u}$.

Proof. By substituting δ , the cost function of H-CBF-QP becomes $\mathcal{J}_h = \|\dot{h}_2(\boldsymbol{x}, \hat{\boldsymbol{u}}_2) - \dot{h}_2(\boldsymbol{x}, \boldsymbol{u})\|^2$. Let us consider $\hat{\boldsymbol{u}}_2 = \boldsymbol{u}^* + \Delta \boldsymbol{u}$ from (7). Then, the cost function is specified as follows:

$$\mathcal{J}_h = (\boldsymbol{u}^* + \Delta \boldsymbol{u} - \boldsymbol{u})^\top \mathbf{M} (\boldsymbol{u}^* + \Delta \boldsymbol{u} - \boldsymbol{u}) \\ = \|\boldsymbol{u}^* - \boldsymbol{u}\|_{\mathbf{M}}^2 + \boldsymbol{p}^\top \boldsymbol{u} + \text{const}$$

where $\boldsymbol{p}^{\top} = -2\Delta \boldsymbol{u}^{\top} \mathbf{M}$ and $\mathbf{M} = (\mathcal{L}_g h_2(\boldsymbol{x}))^{\top} \mathcal{L}_g h_2(\boldsymbol{x})$, which is positive definite. Different from R-CBF-QP, H-CBF-QP implicitly considers the feedback control input although $\lambda_u = 0$ due to Proposition 2.

B. Design of Weighing Factors

The ideal approach would be to formulate and solve a large-scale nonlinear optimization problem that determines the weighting factors, but this is challenging since the entire control framework comprises multiple optimization problems in a cascaded structure. Therefore, in this section, we propose a straightforward method to find suitable parameters for the weightings. To simplify the problem, we fix the weighting factor for the feedback control input error as a constant $\lambda_u = 1$, and express the other weighting factor as a function of h_2 . Specifically, we define $\lambda_{\delta}(h_2(\boldsymbol{x}))$ as:

$$\lambda_{\delta}(h_2(\boldsymbol{x})) = \begin{cases} \gamma_0 & \text{if } h_2(\boldsymbol{x}) > 0, \\ \gamma_0(1 + \Delta\gamma |h_2(\boldsymbol{x})|) & \text{else} \end{cases}$$
(13)

where $\gamma_0 > 0$ is a baseline weighting factor for relaxing $h_2(\mathbf{x})$, and $\Delta \gamma > 0$ is a gradient factor for preventing the violation of $h_2(\mathbf{x}) \geq 0$. This weighting factor reduces δ more as $h_2(\mathbf{x})$ decreases below 0. As discussed in Corollary 2 and Proposition 2, the proper selection of the parameters $(\gamma_0, \Delta \gamma)$ is crucial to balance the performance of the feedback controller and the satisfaction of the lower-prioritized safety condition.

One straightforward approach for searching the optimal parameter is to use a sampling-based technique. First, we define a cost function that depends on the task trajectory, which is discretized with a time interval Δt . The cost function is given as follows:

$$r = \sum_{i=1}^{N-1} \|\boldsymbol{e}_x(i\Delta t)\|_{w_i} + \|\boldsymbol{e}_x(T)\|_{w_f} + w_\delta \sum_{i=1}^N \delta(i\Delta t)$$
(14)

where w_i and w_f are the weightings for the tracking trajectory and achieving the final goal, respectively, and $w_f \gg w_i > 0$. $w_{\delta} > 0$ is a discount factor that penalizes safety relaxation of the secondary condition. Second, we randomly draw samples from the Normal distribution $(\gamma_0, \Delta \gamma) \sim \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and covariance matrix, respectively. We iterate the simulations with the random samples to find the optimal pair $(\gamma_0, \Delta \gamma)$ that minimizes the cost function (14). When designing the weightings by evaluating the cost, we eliminate the offset as $r_o(\Delta\gamma, \gamma_0) = r(\Delta\gamma, \gamma_0) - \min(r)$ where $r(\Delta\gamma, \gamma_0)$ denotes the cost with $\Delta\gamma$ and γ_0 .

C. Generalization

We generalize the proposed approach for handling general cases to incorporate m CBF constraints. To achieve this, we rewrite Assumptions 1 and 2 as follows for the extensive case:

Assumption 3. Given m CBFs, h_1, \dots, h_m , we assume that a state x belongs to $C_1 - C_{\cap}$ where $C_{\cap} = \{x \in \mathbb{R}^n : h_1(x) \ge 0 \land \dots \land h_m(x) \ge 0\}$. The importance of the CBFs is determined in lexicographic order.



Fig. 5. Simulation results: (a) x and y positions of the robot, (b) CBF values with respect to different values of γ_0 , (c) tracking errors with respect to different values of γ_0 , (d) color map of the cost values in terms of the sampled parameters, (e) robotic simulation results demonstrated using a quadruped robot (A1) in 2D operational space. In figure (a), we present the entire results of H-CBF-QP given the sampled γ_0 and $\Delta\gamma$. Figures (b) and (c) include the results with the fixed $\Delta\gamma$ for showing a clear tendency of γ_0 (1, 2, 3, 4, 5).

Under the above assumption, it is possible to extend our approach in both implicity and explicit ways. First, we formulate a single QP problem that considers the feedback control input and all safety-critical constraints implicitly. This is the Implicit-H-CBF-QP (IH-CBF-QP) and is given as follows:

Implicit-H-CBF-QP (IH-CBF-QP):

$$\begin{split} \min_{\boldsymbol{u}\in\mathcal{U},\ \delta_{2},\cdots,\delta_{m}\in\mathbb{R}} \quad \mathcal{J}_{imp} &= \lambda_{u} \|\boldsymbol{u}^{\star}-\boldsymbol{u}\|^{2} + \sum_{k=2}^{m} \lambda_{\delta_{k}} \delta_{k}^{2}, \\ \text{s.t.} \quad \dot{h}_{1}(\boldsymbol{x},\boldsymbol{u}) \geq -\alpha_{1}(h_{1}(\boldsymbol{x})), \\ \dot{h}_{k}(\boldsymbol{x},\boldsymbol{u}) + \delta_{k} &= \dot{h}_{i}(\boldsymbol{x}, \hat{\boldsymbol{u}}_{k}), \quad \forall i \in \{2,\cdots,m\} \end{split}$$

We obtain a feasible control input by solving m QP problems, which are (7) for k = 2 to k = m and the above IH-CBF-QP. However, the solution to the problem is too sensitive and ambiguous to the weighting parameters in the cost function. Since all relaxation variables are coupled, it is hard to impose the hierarchy among them in this implicit way, even if heuristics are used. Improper weightings may result in reducing the lower-prioritized constraint and increasing the error of the higher-prioritized one.

To address this issue, we introduce a sequential optimization method, Explicit-H-CBF-QP (EH-CBF-QP). This formulation is based on a recursive extension of the H-CBF-QP and defined as follows for $k \ge 3$:

Explicit-H-CBF-QP (EH-CBF-QP):

$$\begin{split} \min_{\boldsymbol{u}\in\mathcal{U},\ \delta_k\in\mathbb{R}} \quad & \mathcal{J}_{exp} = \lambda_u \|\boldsymbol{u}^* - \boldsymbol{u}\|^2 + \lambda_{\delta_k} \delta_k^2, \\ \text{s.t.} \quad & \dot{h}_1(\boldsymbol{x},\boldsymbol{u}) \geq -\alpha_1(h_1(\boldsymbol{x})), \\ & \dot{h}_i(\boldsymbol{x},\boldsymbol{u}) + \delta_i^* = \dot{h}_i(\boldsymbol{x},\hat{\boldsymbol{u}}_i), \quad \forall i \in \{2,\cdots,k-1\} \\ & \dot{h}_k(\boldsymbol{x},\boldsymbol{u}) + \delta_k = \dot{h}_k(\boldsymbol{x},\hat{\boldsymbol{u}}_k), \end{split}$$

Here, δ_i^* denotes the optimal relaxation variable obtained from the optimization of the *i*-th H-CBF-QP. By recursively solving the EH-CBF-QP problems from k = 2 to k = m, we obtain the optimal control input and relaxation variables for all safety-critical constraints.

Corollary 3. In the EH-CBF-QP formulation, the hierarchy is strictly maintained, i.e., it is impossible to decrease δ_k^2 without increasing δ_i^2 with higher priority (i < k).

IV. IMPLEMENTATION

In this section, we present simulation and experiment results that demonstrate the effectiveness of our approach using a quadruped robot system (A1).

A. Simulation Scenario

The simulations were performed on a laptop equipped with a 3.4 GHz Intel Core i7 processor using MATLAB, Pybullet, and OSQP software tools. In addition, we consider a double integrator (ROM) and PD controller to validate our approach in the high-level architecture, then implement a full-body task space controller to realize the control input of ROM in the simulation. The desired orientation of the robot's base is computed based on the ratio of velocity components in the *x* and *y* directions. The state of ROM is defined as $\boldsymbol{x} = [\boldsymbol{p}^{\top}, \ \boldsymbol{p}^{\top}]^{\top}$ where $\boldsymbol{p} \in \mathbb{R}^2$ represents the position in a 2dimensional space, and the control input $\boldsymbol{u} \in \mathcal{U}$ is bounded by an input set $\mathcal{U} = \{\boldsymbol{z} \in \mathbb{R}^2 : |z_i| \leq 5, i = 1, 2\}.$

In this paper, we aim to solve a tracking problem where the task trajectory, $p^d(t)$, is given for [0, 20] seconds with $p^d(0) = [0, 0]^{\top}$ and $p^d(20) = [-12, 10]^{\top}$. To ensure safety, we enforce inequality constraints such that $h_1(x) = p_x + 0.2 \ge 0$, $h_2(x) = p_y + 0.2 \ge 0$, and

$$h_k = \|\boldsymbol{p} - \boldsymbol{p}_k\| - r_k \ge 0$$



Fig. 6. **Snapshots of experimental validation**: (a) and (b) show the snapshots of experiments implemented by using R-CBF-QP and H-CBF-QP, respectively. (c) and (d) represent CBF value variations in R-CBF-QP and H-CBF-QP, respectively. The robot successfully reaches the goal destination using both approaches. However, there is a collision between the robot and one of the yellow boxes, which is located near the center of Region 2, in the third snapshot of (a). In contrast, H-CBF-QP effectively avoids the collision.

where $p_3 = [-1, 5.5]^{\top}$, $p_4 = [-1, 4.5]^{\top}$, $r_3 = 4$, and $r_4 = 5$. In this simulation, we assume that the region represented by $h_3(x) \ge 0$ (region 1) is the more dangerous region than the region represented by $h_4(x) \ge 0$ (region 2), and hence, a hierarchy among these safety constraints is imposed in lexicographic order. For the class \mathcal{K} functions, we utilize the following function:

$$\alpha(h) = \begin{cases} \varphi_1 \log(\varphi_2 h + \varphi_3) - \varphi_1 \log \varphi_3 & h \ge 0, \\ -\varphi_1 \log(-\varphi_2 h + \varphi_3) + \varphi_1 \log \varphi_3 & h < 0 \end{cases}$$

where $\varphi_1, \varphi_2, \varphi_3 > 0$. In this section, we set $\varphi_1 = 1$, $\varphi_2 = 0.5$, and $\varphi_3 = 0.1$. We draw random samples for δ_2 with $\boldsymbol{\mu} = [3, 1.5]$ and $\boldsymbol{\Sigma} = \text{diag}(1, 1)$. Next, we consider simulation results with the samples in $\gamma_0 \in [0, 5]$ and $\Delta \gamma \in [0, 2.5]$ for simplicity. For the cost (14), the weightings are set as $w_i = 1$, $w_f = 1000$, $w_\delta = 0.5$, respectively.

B. Simulation Results

We simulated the R-CBF-QP and H-CBF-QP controllers with different parameters for the weighting factor, and the results are depicted in Fig. 5. Fig. 5(a) shows the desired and actual base positions of the robot with different controllers. While R-CBF-QP prioritizes safety-critical constraints for region 1, the actual path deviates slightly from the desired trajectory. In contrast, the robot controlled by H-CBF-QP deviates more far from the desired trajectory than the one controlled by R-CBF-QP because the robot tries to become safer by escaping region 2. Fig 5(b) shows that the barrier function value $h_4(x)$ becomes closer to zero while maintaining $h_3(x)$ above zero, indicating that our approach is capable of reducing the violations of secondary safety conditions while strictly satisfying those with the highest priority. However, as shown in Fig. 5(c), the tracking error of the feedback controller increases as we attempt to make the robot safer in terms of the secondary safety condition. To balance the feedback control performance and safety, we deploy the proposed a sampling-based technique to find proper parameters and analyze them in a cost map of Fig. 5(d).

Fig. 5(e) shows the simulation results implemented by using Pybullet. The black path indicating the result driven by R-CBF-QP goes through the center of region 2. However, the robot controlled by H-CBF-QP tries to avoid region 2, even if it incurs some tracking errors. Interestingly, if we set the weighting parameters too large, the robot moves far from the desired trajectory, and it becomes challenging to return to the original path. Therefore, we need to find the proper balance between performance and safety parameters.

C. Experimental Validation

We demonstrate the effectiveness and efficiency of our proposed approaches using the real quadruped robot (A1). We maintain consistency between the simulation scenario and the experimental setup. To do experiments in a realistic environment, we construct an experimental structure using boxes (region 1) and artificial turf plates (region 2) shown in Fig. 6. At the center of the region 2, we put a box which is lighter than ones in the region 1. Given the assumption that the boxes in the region 1 are too heavy for the robot to move, we prioritize avoiding region 1 over region 2. Due to space limitations, we scale down the problem setup and set the centers of the regions at $\boldsymbol{p}_3 = [-1.7, \ 1.5]^{ op}$ and $p_4 = [-0.35, 1.8]^{\top}$, with radii of $r_3 = 1.7$ and $r_4 = 1.34$, respectively. The goal destination is set at $[-4, 3.34]^{\top}$, and we generate a continuous trajectory toward the destination during a 20-second interval. The detailed specifications for the CBFs are the same as ones in the simulations. By demonstrating simulations with the sampled parameters, then, we obtain the pair for H-CBF-QP as (5, 1.3).

In these experiments, both relaxation approaches, R-CBF-QP and H-CBF-QP, are able to generate feasible control inputs for the robot. Despite colliding with the box located at the center of region 2, R-CBF-QP is able to steer the robot towards the goal. However, if the box becomes much heavier than our setup, the robot may not be able to reach the goal destination. In contrast, H-CBF-QP generates a safer behavior and enables the robot to avoid the yellow box at the center of region 2, as shown in Fig. 7. Fig. 6(c) and



Fig. 7. **Experimental results**: The movements of the robot in the operational space (left) and the position over the defined time horizon (right). There exists a box at the center of Region 2 depicted by a yellow square.

(d) confirm that H-CBF-QP results in less violations of the safety condition associated with h_4 compared to R-CBF-QP.

V. CONCLUSION

This paper presents a novel and intuitive approach to address contradictory safety conditions in robotic systems. By imposing a hierarchy among these conditions, we formulate feasible QP problems that consider the hierarchical safety conditions while generating dynamic locomotion. R-CBF-QP generates a feasible control input, taking into account only the highest priority safety constraint. On the other hand, the proposed H-CBF-QP is capable of strictly holding the safety condition with the highest priority, while handling others with lower priority. We also leverage a sampling-based method to determine the weighting factors in the H-CBF-QP formulation. The effectiveness of our approach is demonstrated through simulations in achieving safer locomotion of the quadruped robot (A1).

In the future, we aim to expand our proposed approach to handle more intricate physical interactions between robots and their environments. Our focus is on incorporating physical reaction forces when walking on contact-rich terrain or interacting with other agents in obstacle-rich environments. Furthermore, we are exploring the theoretical aspect of Inputto-State safety verification of our method and assessing the impact of uncertainties arising from estimations.

REFERENCES

- M. Igarashi and H. Nakamura, "Collision avoidance assist control for two-wheel vehicle robots by control barrier function," in *Proceedings* of the International Automatic Control Conference. IEEE, 2018, pp. 1–6.
- [2] M. Desai and A. Ghaffari, "Clf-cbf based quadratic programs for safe motion control of nonholonomic mobile robots in presence of moving obstacles," in *Proceedings of the International Conference on Advanced Intelligent Mechatronics*. IEEE, 2022, pp. 16–21.
- [3] A. Manjunath and Q. Nguyen, "Safe and robust motion planning for dynamic robotics via control barrier functions," in *Proceedings of the IEEE Conference on Decision and Control.* IEEE, 2021, pp. 2122– 2128.
- [4] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safetyaware informative motion planning for legged robots," *arXiv preprint* arXiv:2103.14252, 2021.
- [5] J. Liu, M. Li, J.-K. Huang, and J. W. Grizzle, "Realtime safety control for bipedal robots to avoid multiple obstacles via clf-cbf constraints," *arXiv preprint arXiv:2301.01906*, 2023.
- [6] J. Lee, E. Bakolas, and L. Sentis, "An efficient and direct method for trajectory optimization of robots constrained by contact kinematics and forces," *Autonomous Robots*, vol. 45, pp. 135–153, 2021.

- [7] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [8] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.
- [9] D. Kim, S. J. Jorgensen, J. Lee, J. Ahn, J. Luo, and L. Sentis, "Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 936–956, 2020.
- [10] C. Khazoom, D. Gonzalez-Diaz, Y. Ding, and S. Kim, "Humanoid self-collision avoidance using whole-body control with control barrier functions," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*. IEEE, 2022, pp. 558–565.
 [11] M. Koptev, N. Figueroa, and A. Billard, "Real-time self-collision
- [11] M. Koptev, N. Figueroa, and A. Billard, "Real-time self-collision avoidance in joint space for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1240–1247, 2021.
- [12] 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, 2017.
- [13] 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.
- [14] N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 1041–1053.
- [15] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multilayered safety for legged robots via control barrier functions and model predictive control," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 8352–8358.
- [16] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained robot control using control barrier functions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 279–285.
- [17] F. Ferraguti, C. T. Landi, A. Singletary, H.-C. Lin, A. Ames, C. Secchi, and M. Bonfè, "Safety and efficiency in robotics: the control barrier functions approach," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 139–151, 2022.
- [18] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10954– 10961, 2022.
- [19] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proceedings of the IEEE Conference on Decision and Control.* IEEE, 2014, pp. 6271–6278.
- [20] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed pointwise feasibility," in *Proceedings of the American Control Conference*. IEEE, 2021, pp. 3856–3863.
- [21] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *Proceedings of the IEEE Conference on Decision and Control*, 2016, pp. 2659–2664.
- [22] 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.
- [23] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 1, pp. 96–101, 2018.
- [24] ——, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 3, pp. 757–762, 2019.
- [25] J. Breeden and D. Panagou, "Compositions of multiple control barrier functions under input constraints," arXiv preprint arXiv:2210.01354, 2022.
- [26] J. Lee, E. Bakolas, and L. Sentis, "Hierarchical task-space optimal covariance control with chance constraints," *IEEE Control Systems Letters*, vol. 6, pp. 2359–2364, 2022.
- [27] M. H. Raibert, H. B. Brown Jr, and M. Chepponis, "Experiments in balance with a 3d one-legged hopping machine," *International Journal* of Robotics Research, vol. 3, no. 2, pp. 75–92, 1984.