# Safety-Critical Control using Optimal-decay Control Barrier Function with Guaranteed Point-wise Feasibility

Jun Zeng, Bike Zhang, Zhongyu Li and Koushil Sreenath

*Abstract*— Safety is one of the fundamental problems in robotics. Recently, a quadratic program based control barrier function (CBF) method has emerged as a way to enforce safety-critical constraints. Together with control Lyapunov function (CLF), it forms a safety-critical control strategy, named CLF-CBF-QP, which can mediate between achieving the control objective and ensuring safety, while being executable in real-time. However, once additional constraints such as input constraints are introduced, the CLF-CBF-QP may encounter infeasibility. In order to address the challenge arises due to the infeasibility, we propose an optimal-decay form for safety-critical control wherein the decay rate of the CBF is optimized point-wise in time so as to guarantee point-wise feasibility when the state lies inside the safe set. The proposed control design is numerically validated using an adaptive cruise control example.

## I. INTRODUCTION

### A. Motivation

Safety-critical optimal control and planning is one of the fundamental problems in robotics, *e.g.*, robots need to be able to safely avoid obstacles while using minimal energy. In order to ensure the safety of robotic systems while achieving optimal performance, the tight coupling between potentially conflicting control objectives and safety criteria is usually formulated as an optimization problem. Recent work [1], [2] formulates this problem using control barrier functions as an optimization problem, where the safety criteria is formulated as constraints. Additionally, a robotic system usually has a constrained set for the admissible inputs due to physical limitations, which can be taken as additional constraints in the optimization problem [3]–[5]. However, the input constraint and control barrier function constraint might be in conflict in the optimization and make the optimization infeasible. In this paper, we study this feasibility problem and illustrate a variant form of safety-critical control with control barrier functions which satisfies the input constraint and guarantees point-wise feasibility along the trajectory.

### B. Related Work

CBFs have recently been introduced as a promising way to ensure set invariance by considering the system dynamics. Furthermore, a safety-critical control design for continuous-time systems was proposed by unifying a control Lyapunov function (CLF) and a control barrier function (CBF) through a quadratic program (CLF-CBF-QP) [2], [6]. This method could be deployed as a real-time optimization-based controller with safety-critical constraints, shown in [1], [7]. The adaptive, robust, and stochastic cases of safety-critical control with CBF have been considered in [8]–[10]. CBFs have also been used for high relative degree safety constraints for nonlinear systems [11], [12]. Besides the continuous-time domain, the formulation of CBF was generalized into discrete-time systems in [4], and systems evolving on manifolds in [7]. Recently, CBF constraints were also applied in control problems using a data-driven approach [13]–[18] and optimal control design [19], [20].

However, the input constraint was not considered in the optimization in some of the previous work, such as [6], which means the optimized control input might not be executable due to the physical limitations of the system. Other work, such as [3]–[5], [7]–[9], did consider the input constraint, but the potential conflict between input constraint and CBF constraint was not addressed, potentially resulting in an unsolvable optimization problem. In [2], a valid CBF is specifically designed for adaptive cruise control scenario with explicit integration problem over the system dynamics under input constraint, which could ensure the feasibility in the optimization. However, the feasibility problem for general nonlinear systems remains as an unsolved topic for the safety-critical optimal control.

### C. Contribution

The contributions of this paper are as follows.

- We present the reasons of potential infeasibility in the CBF-QP and CLF-CBF-QP.
- With analysis in both input-space and state-space, we reveal quantitatively and qualitatively that the infeasibility appears potentially due to a small decay rate of the control barrier function constraint.
- We propose an optimal-decay form of CBF-QP and CLF-CBF-QP where the decay rate of the control barrier function is optimized.
- We prove that our optimal-decay formulation is point-wise feasible for any state lying strictly inside the safe set. An adaptive cruise control example is used to numerically verify this.

### D. Paper Structure

The paper is organized as follows: in Sec. II, we present the background about control barrier functions and point out the potential infeasibility in safety-critical control. In Sec. III,

Jun Zeng, Bike Zhang, Zhongyu Li and Koushil Sreenath are with the Department of Mechanical Engineering, University of California, Berkeley, CA, 94720, USA, {zengjunsjtu, bikezhang, zhongyu_li, koushils}@berkeley.edu

Code and simulations are available at https://github.com/HybridRobotics/CBF-Pointwise-Feasibility.

we analyze quantitatively the point-wise feasibility problem in the input-space. In Sec. IV, we show qualitatively the point-wise feasibility problem from the perspective of the state-space. In Sec. V, we propose an optimal-decay form of CBF-QP and CLF-CBF-QP and prove their point-wise feasibility for any state lying strictly inside the safe set. An adaptive cruise control example is used to numerically validate our proposed optimal-decay form in Sec. VI. Finally, Sec. VII provides concluding remarks.

## II. BACKGROUND

We consider a nonlinear affine system of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, with $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ being locally Lipschitz. The system is subject to input constraints

$$\mathbf{u}(t) \in \mathcal{U}_{adm}(\mathbf{x}(t)), \ \forall t \geq 0, \tag{2}$$

where $\mathcal{U}_{adm}(\mathbf{x}(t)) \subset \mathbb{R}^m$ denotes the set of admissible inputs, which could be state dependent.

We consider a set $\mathcal{C} \subset \mathbb{R}^n$ defined as the zero-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, yielding:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\},$$
$$\partial\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) = 0\}, \tag{3}$$
$$\text{Int}(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) > 0\}.$$

Throughout this paper, we refer to $\mathcal{C}$ as a safe set.

The definitions of control barrier functions and control Lyapunov functions are summarized as follow, see [6] for detailed explanations. The function $h$ becomes a control barrier function if $\frac{\partial h}{\partial \mathbf{x}} \neq 0$ for all $\mathbf{x} \in \partial\mathcal{C}$, and there exists an extended class $\mathcal{K}_\infty$ function $\alpha$ such that for the control system (1), $h$ satisfies

$$\exists \ \mathbf{u} \text{ s.t. } \dot{h}(\mathbf{x}, \mathbf{u}) \geq -\alpha(h(\mathbf{x})), \ \alpha \in \mathcal{K}_\infty. \tag{4}$$

Besides the system safety, we are also interested in stabilizing the system with a feedback control law $\mathbf{u}$ under a control Lyapunov function $V$ with a class $\mathcal{K}$ function $\gamma$, *i.e.*,

$$\exists \ \mathbf{u} \text{ s.t. } \dot{V}(\mathbf{x}, \mathbf{u}) \leq -\gamma(V(\mathbf{x})), \ \gamma \in \mathcal{K}. \tag{5}$$

Note that we can write down

$$\dot{h}(\mathbf{x}, \mathbf{u}) = L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u}, \tag{6}$$

where $L_f h(\mathbf{x})$ and $L_g h(\mathbf{x})$ are Lie-derivatives of $h(\mathbf{x})$ along $f(\mathbf{x})$ and $g(\mathbf{x})$, respectively. We can also write down

$$\dot{V}(\mathbf{x}, \mathbf{u}) = L_f V(\mathbf{x}) + L_g V(\mathbf{x})\mathbf{u}, \tag{7}$$

where $L_f V(\mathbf{x})$ and $L_g V(\mathbf{x})$ are Lie-derivatives of $V(\mathbf{x})$. The above construction of the CLF and CBF allows us to define safety-critical control for a nonlinear affine system (1).

Given a feedback controller $\mathbf{u} = k(\mathbf{x})$ for the control system (1), we wish to guarantee safety. We consider the following Quadratic Program (QP) based controller that finds the optimal $\mathbf{u}$ in the optimization as follows:

**CBF-QP:**

$$\mathbf{u}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2}||\mathbf{u} - k(\mathbf{x})||^2 \tag{8a}$$
$$\text{s.t. } L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} \geq -\alpha(h(\mathbf{x})), \tag{8b}$$
$$\mathbf{u} \in \mathcal{U}_{adm}(\mathbf{x}). \tag{8c}$$

When the input constraint (8c) is excluded, we have a single inequality constraint, thus the CBF-QP has a closed-form solution per the KKT conditions, and this method was used in [21], [22]. However, when the input constraint is considered, there might not exist any $\mathbf{u}$ satisfying both input constraint and CBF constraint simultaneously. This could lead to a potential infeasible optimization problem.

We could also use a QP based formulation of safety-critical control which unifies safety and stability. Concretely, we consider the following QP based controller:

**CLF-CBF-QP:**

$$\mathbf{u}(\mathbf{x}) = \underset{(\mathbf{u},\delta) \in \mathbb{R}^{m+1}}{\operatorname{argmin}} \frac{1}{2}\mathbf{u}^T H(\mathbf{x})\mathbf{u} + p\delta^2 \tag{9a}$$
$$\text{s.t. } L_f V(\mathbf{x}) + L_g V(\mathbf{x})\mathbf{u} \leq -\gamma(V(\mathbf{x})) + \delta, \tag{9b}$$
$$L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} \geq -\alpha(h(\mathbf{x})), \tag{9c}$$
$$\mathbf{u} \in \mathcal{U}_{adm}(\mathbf{x}). \tag{9d}$$

where $H(\mathbf{x})$ is any positive definite matrix (point-wise in $\mathbf{x}$), and we have a relaxation variable $\delta$ on the CLF constraint (9b) with additional quadratic cost in (9a). When we exclude the input constraint (9d) out of the optimization, the solvability can be guaranteed since the CLF constraint is relaxed and the CBF constraint (9c) is the only hard constraint. This method was applied in [23], [24]. However, when the input constraint (9d) is also considered, we might again encounter an infeasible optimization problem.

## III. POINT-WISE FEASIBILITY IN INPUT-SPACE

Having presented the background of safety-critical control, we will now show how to pick an appropriate $\alpha$ in the CLF-CBF-QP/CBF-QP to guarantee point-wise feasibility from the perspective of input-space, *i.e.*, how to guarantee feasibility of the optimization problem at a given state $\mathbf{x}(t) = \mathbf{x}_t$.

For the state $\mathbf{x}_t$ at time $t$, we define the feasible superlevel set $\mathcal{U}_{cbf}(\mathbf{x}_t)$ as the region satisfying CBF constraint in input-space, *i.e.*,

$$\mathcal{U}_{cbf}(\mathbf{x}_t) := \{\mathbf{u} \in \mathbb{R}^m : L_f h(\mathbf{x}_t) + L_g h(\mathbf{x}_t)\mathbf{u} \geq \\ -\alpha(h(\mathbf{x}_t))\}. \tag{10}$$

Note that $\mathcal{U}_{cbf}(\mathbf{x}_t)$ is a half-space in the input-space $\mathbb{R}^m$ since the CBF constraint is affine in $\mathbf{u}$. The level set of the CBF constraint in input-space is defined as $\partial\mathcal{U}_{cbf}(\mathbf{x}_t)$,

$$\partial\mathcal{U}_{cbf}(\mathbf{x}_t) = \{\mathbf{u} \in \mathbb{R}^m : L_f h(\mathbf{x}_t) + L_g h(\mathbf{x}_t)\mathbf{u} = \\ -\alpha(h(\mathbf{x}_t))\}. \tag{11}$$
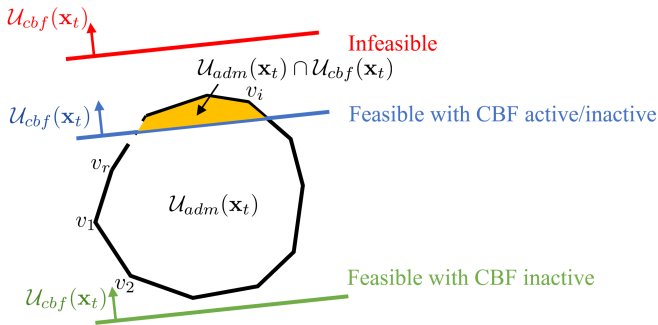
Fig. 1: The set of admissible inputs $\mathcal{U}_{adm}(\mathbf{x}_t)$ is defined as a convex polytope with $r$ vertices. The upper half-spaces, indicated by the red, blue, green arrows, repsent the feasible superlevel set $\mathbf{u}_{cbf}(\mathbf{x}_t)$ corresponding to three different superlevel sets. We can see clearly that at least one vertex of polytope $\mathcal{U}_{adm}(\mathbf{x}_t)$ needs to lie inside or on the surface of the set $\mathcal{U}_{cbf}(\mathbf{x}_t)$ to guarantee the intersection set between $\mathcal{U}_{adm}(\mathbf{x}_t)$ and $\mathcal{U}_{cbf}(\mathbf{x}_t)$ is not empty.

Then the feasibility problem becomes whether the intersection between $\mathcal{U}_{cbf}(\mathbf{x}_t)$ and $\mathcal{U}_{adm}(\mathbf{x}_t)$ as defined in (2) is empty or not. If the intersection is not empty, the optimization in CBF-QP or CLF-CBF-QP is feasible at $\mathbf{x}_t$.

In order to provide a quantitative way of explaining point-wise feasibility from the perspective of input-space, We suppose that the set of admissible inputs $\mathcal{U}_{adm}(\mathbf{x}_t)$ could be described as a convex polytope defined with $r$ vertices, presented in Fig. 1, where each vertex is noted as $v_i(\mathbf{x}_t) \in \mathbb{R}^m$ and $i \in \{1, 2, ..., r(\mathbf{x}_t)\}$. Then we have $\mathcal{U}_{adm}(\mathbf{x}_t)$ could be written as

$$
\mathcal{U}_{adm}(\mathbf{x}_t) = \{\mathbf{u} \in \mathbb{R}^m : \mathbf{u} = \sum_{i=1}^{r(\mathbf{x}_t)} \lambda_i(\mathbf{x}_t) v_i(\mathbf{x}_t),
$$
$$
\sum_{i=1}^{r(\mathbf{x}_t)} \lambda_i(\mathbf{x}_t) = 1, \lambda_i(\mathbf{x}_t) \geq 0\}. \tag{12}
$$

Then the necesary and sufficient condition of having the intersection between $\mathcal{U}_{cbf}(\mathbf{x}_t)$ defined in (10) and $\mathcal{U}_{adm}(\mathbf{x}_t)$ defined in (12) being not empty is that, at least one vertex of polytope $\mathcal{U}_{adm}(\mathbf{x}_t)$ lies inside or on the surface of the set $\mathcal{U}_{cbf}(x)$, shown as follows,

$$
\mathcal{U}_{adm}(\mathbf{x}_t) \cap \mathcal{U}_{cbf}(\mathbf{x}_t) \neq \emptyset \Leftrightarrow \exists i, v_i \in \mathcal{U}_{cbf}(\mathbf{x}_t). \tag{13}
$$

Hence, the set of candidate $\alpha$ functions in the CBF constraint (4), denoted as $\mathcal{K}_{fea}^{\alpha}(\mathbf{x}_t)$, that guarantees the feasibility of the optimization, becomes the union of the set of functions that guarantees any vertex in $\mathcal{U}_{adm}(\mathbf{x}_t)$ lies inside or on the surface of the set $\mathcal{U}_{cbf}(\mathbf{x}_t)$, which could be written as follows

$$
\mathcal{K}_{fea}^{\alpha}(\mathbf{x}_t) = \bigcup_{i=1}^{r(\mathbf{x}_t)} \{\alpha \in \mathcal{K}_\infty : \alpha(h(\mathbf{x}_t)) \geq -L_f h(\mathbf{x}_t) - L_g h(\mathbf{x}_t) v_i(\mathbf{x}_t)\},
$$

and it could be reformulated with a minimum operator

$$
\mathcal{K}_{fea}^{\alpha}(\mathbf{x}_t) = \{\alpha \in \mathcal{K}_\infty : \alpha(h(\mathbf{x}_t)) \geq \min_i(-L_f h(\mathbf{x}_t) - L_g h(\mathbf{x}_t) v_i(\mathbf{x}_t))\}. \tag{14}
$$

Therefore, given a function $\alpha$, the CBF-QP/CLF-CBF-QP in (8) and (9) are point-wise feasible if $\exists \alpha \in \mathcal{K}_\infty$ s.t.,

$$
\alpha(h(\mathbf{x}_t)) \geq \min_i(-L_f h(\mathbf{x}_t) - L_g h(\mathbf{x}_t) v_i(\mathbf{x}_t))
$$

is satisfied at state $\mathbf{x}_t$.

Moreover, given a state $\mathbf{x}_t$, from (14), we can see when

$$
\min_i(-L_f h(\mathbf{x}_t) - L_g h(\mathbf{x}_t) v_i(\mathbf{x}_t)) \leq 0, \tag{15}
$$

we have $\mathcal{K}_{fea}^{\alpha}(\mathbf{x}_t) = \mathcal{K}_\infty$, i.e., we have the point-wise feasibility, for any $\alpha \in \mathcal{K}_\infty$. However, when (15) is not satisfied, $\mathcal{K}_{fea}^{\alpha}(\mathbf{x}_t)$ becomes a proper subset of $\mathcal{K}_\infty$ and the function $\alpha$ has a lower bound at state $\mathbf{x}_t$ to guarantee the point-wise feasibility.

From above, we could pick an appropriate $\alpha$ function to guarantee point-wise feasibility at a given state $\mathbf{x}_t$. Beside feasibility, we are also interested in whether the CBF constraint is activated during the optimization.

**Remark 1.** We have $\forall i, v_i(\mathbf{x}_t) \in \mathcal{U}_{cbf}(\mathbf{x}_t)$ when

$$
\alpha(h(\mathbf{x}_t)) \geq \max_i(-L_f h(\mathbf{x}_t) - L_g h(\mathbf{x}_t) v_i(\mathbf{x}_t)), \tag{16}
$$

which means $\mathcal{U}_{adm}(\mathbf{x}_t) \subset \mathcal{U}_{cbf}(\mathbf{x}_t)$, therefore in this case, the CBF constraint does not confine the input constraint during the optimization. This case is illustrated by the green $\mathcal{U}_{cbf}(\mathbf{x}_t)$ in Fig. 1. When (16) is not satisfied, we have the CBF constraint confine the input constraint, i.e., the intersection between $\mathcal{U}_{adm}(\mathbf{x}_t)$ and $\mathcal{U}_{cbf}(\mathbf{x}_t)$ becomes a proper set of $\mathcal{U}_{adm}(\mathbf{x}_t)$, indicated by the blue $\mathcal{U}_{cbf}(\mathbf{x}_t)$ in Fig. 1. When the intersection is empty, illustrated by the red $\mathcal{U}_{cbf}(\mathbf{x}_t)$ in Fig. 1, the optimization problem is infeasible.

**Remark 2.** We say a constraint is active in the optimization when the optimal solution lies on the constraint line [25]. When the CBF constraint does not confine the input constraint, the CBF constraint becomes inactive during the optimization. However, when the CBF constraint confines the input constraint, it does not necessarily guarantee the CBF constraint activation, as the value of optimal solution also depends on the design of cost function in the optimization. Therefore, the choice of $\alpha$ function together with the design of cost function determine the activation of CBF constraint in the safety-critical optimal control, shown in Fig. 1.

**Remark 3.** In this section, the admissible set is assumed as a bounded convex polytope. In fact, these discussions could be easily generalized for an unbounded convex set, as any unbounded set could be regarded as a limit of a sequence of bounded sets [26, Chap. 11]. Moreover, our proposed optimal-decay approach in this paper that guarantees point-wise feasibility will only rely on the admissible set being convex while not strictly assuming it as a bounded convex polytope, see Theorem 1.

We have seen point-wise feasibility in input-space in this section. Next, we will look at point-wise feasibility in the state-space in Sec. IV. After that, we provide a formulation which allows us to guarantee the point-wise feasibility without parameter tuning in Sec. V.

## IV. POINT-WISE FEASIBILITY IN STATE-SPACE

The point-wise feasibility problem could also be understood through qualitative illustration in the state-space. Given a state $\mathbf{x}_t$ at time $t$, we will define $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ to represent the set of reachable states after infinitesimal time $\delta t$ while satisfying system dynamics (1) and input constraint $\mathcal{U}_{adm}(\mathbf{x}_t)$ starting from state $\mathbf{x}(t) = \mathbf{x}_t$, *i.e.*,

$$\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) = \{\mathbf{x}(t + \delta t) \in \mathbb{R}^n : \forall \bar{t} \in [t, t + \delta],$$
$$\dot{\mathbf{x}}(\bar{t}) = f(\mathbf{x}(\bar{t})) + g(\mathbf{x}(\bar{t}))\mathbf{u}(\bar{t}), \quad (17)$$
$$\mathbf{u}(\bar{t}) \in \mathcal{U}_{adm}(\mathbf{x}(\bar{t})), \mathbf{x}(t) = \mathbf{x}_t\}.$$

The evolution of the system dynamics also needs to be safe and thus satisfy the definition of control barrier function (4) during time segment $[t, t + \delta t]$, thus we have

$$h(\mathbf{x}(t + \delta t)) \geq h(\mathbf{x}(t)) - \int_t^{t+\delta t} \alpha(h(\mathbf{x}(\bar{t})))d\bar{t}. \quad (18)$$

This allows us to define the superlevel set in state-space for $\mathbf{x}(t + \delta t)$ satisfying the CBF constraints,

$$\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t) = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq h(\mathbf{x}_t)$$
$$- \int_t^{t+\delta t} \alpha(h(\mathbf{x}(\bar{t})))d\bar{t}, \mathbf{x}(t) = \mathbf{x}_t\}. \quad (19)$$

We also define

$$\mathcal{S}_{cbf}(\mathbf{x}_t, 0) = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq h(\mathbf{x}_t)\}, \quad (20)$$

motivated by $\int_t^{t+\delta t} \alpha(h(\mathbf{x}(\bar{t})))d\bar{t} = 0$ when $\delta t = 0$. The set $\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ corresponds to the set of all possible $\mathbf{x}$ for which $h(\mathbf{x}) \geq h(\mathbf{x}_t)$.

Since the state $\mathbf{x}(t + \delta t)$ should satisfy the system dynamics and control barrier function constraint, the optimization problem is then point-wise feasible at state $\mathbf{x}_t$ when the intersection between $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ and $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ is not empty. We are interested in whether the intersection between $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ and $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ is empty or not under different circumstances. However, it is numerically complicated or even impossible to calculate $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ for a general nonlinear affine system. Thus, we provide an intuition of understanding point-wise feasibility problem through geometry in state-space.

**Remark 4.** Notice that we always have $\mathcal{S}_{cbf}(\mathbf{x}_t, 0) \subset \mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ as $\alpha$ is class $\mathcal{K}_\infty$ function and $h(.)$ is positive.

In practice, we usually define the safety set $\mathcal{C}$ in (3) corresponding to the free space outside the obstacle, illustrated in Fig. 2. $\partial\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ and $\partial\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ are illustrated with black solid and colorful dashed curves respectively, and $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$, $\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ are illustrated as the regions on the top-right side of them. Since we always have $\mathcal{S}_{cbf}(\mathbf{x}_t, 0) \subset \mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$, $\partial\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ is always closer to the obstacle, lying on the bottom-left side of $\partial\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ for any choice of $\alpha$ function. We classify the point-wise feasibility problem into three scenarios as follows.

### A. Moving away from obstacles

When $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) \subset \mathcal{S}_{cbf}(\mathbf{x}_t, 0)$, *i.e.*, the system is moving away from obstacles. The scenario is illustrated in Fig. 2a. In this scenario, we have $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) \subset \mathcal{S}_{cbf}(\mathbf{x}_t, 0) \subset \mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$. This means that for any class $\mathcal{K}_\infty$ function $\alpha$, the optimization will always be point-wise feasible at state $\mathbf{x}_t$.

### B. Moving around obstacles

When $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ intersects with $\partial\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$, we have the reachable state-space lies partly on the top-right side of $\partial\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$, shown in Fig. 2b. In this scenario, the optimization is always feasible for any class $\mathcal{K}_\infty$ function $\alpha$, as $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) \cap \mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ is not empty and is always a subset of $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$.

### C. Moving close to obstacles

When $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) \cap \mathcal{S}_{cbf}(\mathbf{x}_t, 0) = \emptyset$, this usually happens when the system is moving close to obstacles, shown in Fig. 2c and 2d. In this scenario, when $\alpha$ becomes too small (CBF level set in red), the intersection between $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ and $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ is empty and the optimization problem becomes infeasible at state $\mathbf{x}_t$. This indicates that $\alpha(h(\mathbf{x}_t))$ needs to be greater than a lower bound to make the optimization point-wise feasible at state $\mathbf{x}_t$.

**Remark 5.** When $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) \subset \mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$, the CBF constraint does not confine the reachable set which means the CBF constraint is inactive in the optimization, shown with green level sets $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ in Fig. 2. When the intersection between $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ and $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ is non-empty and becomes as a proper subset of $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ shown with blue level sets $\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ in Fig. 2, the CBF constraint does confine the reachable set. This does not necessarily guarantee CBF constraint activation, as the constraint activation also depends on the design of cost function, which is similar to what we discussed in Remark 2. For the red level sets $\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ in Fig. 2c and 2d, the optimization is infeasible.

## V. POINT-WISE FEASIBLE FORMULATION

### A. Formulation and Point-wise Feasibility

In Sec. III and IV, we have seen that the optimization in safety-critical control might become point-wise infeasible at a given state $\mathbf{x}_t$, if either $\mathcal{U}_{cbf}(\mathbf{x}_t) \cap \mathcal{U}_{adm}(\mathbf{x}_t) = \emptyset$ or $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t) \cap \mathcal{S}_{cbf}(\mathbf{x}_t, \delta t) = \emptyset$ which means the convergence of control barrier function is less than a lower bound. In other words, when the decay rate of the lower bound of $h(\mathbf{x})$ is not large enough, we might encounter infeasibility in the optimization problem. To solve this problem, we could manually tune the form of $\alpha$ function to make the optimization feasible, however, this tuning process becomes relatively difficult when the system dynamics becomes complicated. This motivates us to introduce an optimal-decay form of CBF constraint to guarantee the point-wise feasibility for any $\mathbf{x}$ with $h(\mathbf{x}) > 0$.

With the same notation as CBF-QP in (8) and CLF-CBF-QP in (9), we introduce an optimal-decay form of CBF-QP
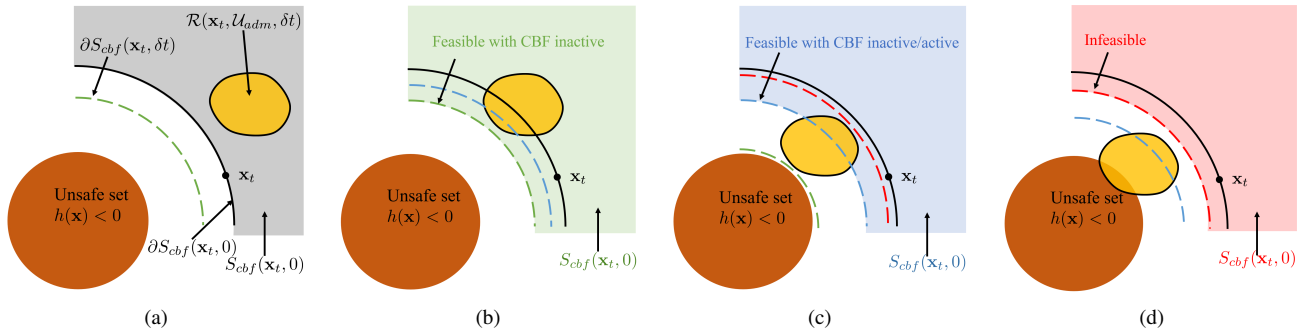
Fig. 2: Point-wise feasibility problem in different scenarios. Given a state $\mathbf{x}_t$, we illustrate $\mathcal{R}(\mathbf{x}_t, \mathcal{U}_{adm}, \delta t)$ as a closed region of yellow color and $\partial \mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ as level sets in black. $\partial \mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ are plotted in red (infeasible), blue (feasible with active/inactive CBF constraint) and green (feasible with inactive CBF constraint) for different scenarios. $\mathcal{S}_{cbf}(\mathbf{x}_t, 0)$ (grey) and $\mathcal{S}_{cbf}(\mathbf{x}_t, \delta t)$ (green, blue, red) are regions on the top-right side of their corresponding level sets.

and CLF-CBF-QP in this section. The optimal decay CBF-QP is formulated as follows

---

**Optimal-decay CBF-QP:**

$$\mathbf{u}(\mathbf{x}) = \operatorname*{argmin}_{(\mathbf{u}, \omega) \in \mathbf{R}^{m+1}} \frac{1}{2} ||\mathbf{u} - k(\mathbf{x})||^2 + p_\omega (\omega - \omega_0)^2 \quad (21a)$$

$$\text{s.t. } L_f h(\mathbf{x}) + L_g h(\mathbf{x}) \mathbf{u} \geq -\omega \alpha(h(\mathbf{x})), \quad (21b)$$

$$\mathbf{u} \in \mathcal{U}_{adm}(\mathbf{x}). \quad (21c)$$

---

Compared with CBF-QP, we optimize the decay rate of the CBF constraint with a new variable $\omega$ in (21b) and add a quadratic cost in (21a). $p_\omega$ is a positive scalar and a scalar $\omega_0$ could usually be chosen to tune the performance of the controller. Similarly, we develop the optimal form of CLF-CBF-QP as follows

---

**Optimal-decay CLF-CBF-QP:**

$$\mathbf{u}(\mathbf{x}) = \operatorname*{argmin}_{(\mathbf{u}, \delta, \omega) \in \mathbf{R}^{m+2}} \frac{1}{2} \mathbf{u}^T H(\mathbf{x}) \mathbf{u} + p\delta^2 + p_\omega (\omega - \omega_0)^2$$

$$\quad (22a)$$

$$\text{s.t. } L_f V(\mathbf{x}) + L_g V(\mathbf{x}) \mathbf{u} \leq -\gamma(V(\mathbf{x})) + \delta, \quad (22b)$$

$$L_f h(\mathbf{x}) + L_g h(\mathbf{x}) \mathbf{u} \geq -\omega \alpha(h(\mathbf{x})), \quad (22c)$$

$$\mathbf{u} \in \mathcal{U}_{adm}(\mathbf{x}). \quad (22d)$$

---

We call (21) and (22) as optimal-decay form of CBF-QP and CLF-CBF-QP, since it actually optimizes the decay rate of lower bound of the control barrier function with variable $\omega$ in the optimization. The point-wise feasibility of optimization for any $\mathbf{x} \in \text{Int}(\mathcal{C})$ in (21), (22) is illustrated with the following theorem.

**Theorem 1.** When $\mathcal{U}_{adm}(\mathbf{x})$ is convex, the optimizations in optimal-decay CBF-QP and CLF-CBF-QP are point-wise feasible for any $\mathbf{x}$ lying inside $\mathcal{C}$, *i.e.*, the optimizations are solvable with a unique solution for any $\mathbf{x}$ when $h(\mathbf{x}) > 0$.

*Proof.* For optimal-decay CBF-QP, since $\omega \in \mathbb{R}$ is a variable to optimize and $h(\mathbf{x}) > 0$, $-\omega \alpha(h(\mathbf{x}))$ could vary between $-\infty$ and $+\infty$ for any $\mathbf{x}$. Then, for any $\mathbf{x}$. there always exists

a value $\omega$ such that the feasible region between (21b) and (21c) is non-empty.

Moreover, we notice that the CBF constraints and input constraints are convex with respect to optimization variables $\mathbf{u}$ and $\omega$. Hence, the feasible region in the optimal-decay CBF-QP is convex and not empty. Additionally, our cost function (21a) is a quadratic form and positive-definite, which is strictly convex. Therefore, the minimization in (21) is convex (convex cost and constraints) with non-empty feasible region, which is solvable and holds a unique solution [25, Chap. 5].

Compared to optimal-decay CBF-QP, the optimal-decay CLF-CBF-QP is nothing different except there is a CLF constraint in (22b), which is also convex with respect to optimization variables and is optimized with the relaxation variable $\delta$. Therefore, the optimization in optimal CLF-CBF-QP is also convex with a non-empty feasible region, which is feasible with an unique solution. $\square$

In fact, we could generalize the point-wise feasibility in Thm. 1 for any $\mathbf{x}$ when $h(\mathbf{x}) \neq 0$, as $\omega \alpha(h(\mathbf{x}))$ could vary between $-\infty$ and $\infty$ when $h(\mathbf{x}) \neq 0$ with $\omega \in \mathbb{R}$. However, this is not interesting since the system is already unsafe when $h(\mathbf{x}) < 0$.

**Remark 6.** The optimization variable $\omega$ is not required to be positive. In fact, when the optimized value of $\omega$ is negative, this implies that the control barrier function is increasing with the optimized control input, correponding to the safety with set invariance. In the scenario of $\omega$ as negative, the system is keeping further away from the obstacles.

### B. Convergence of CBF constraint

The variable $\omega$ optimizes the convergence rate of the CBF constraint, with $\omega \alpha(.)$ still being a class $\mathcal{K}_\infty$ function. Therefore, with optimal-decay CBF-QP and CLF-CBF-QP, we equivalently use a state-dependent rate for the convergence of the CBF constraint, since the optimized value of $\omega$ will be calculated differently at different state $\mathbf{x}$.

Moreover, smaller $\omega_0$ and larger $p_\omega$ would make the control barrier function decay slower. This makes sense from a mathematical perspective, since the larger $\omega_0$ optimizes the

nominal CBF constraint with larger decay rate. Similarly, for smaller $p_\omega$, we will have optimized value of $\omega$ deviating more from the value $\omega_0$, which also brings a larger decay rate on the control barrier function.

### C. Persistent Feasibility and Safety

Note that point-wise feasibility for all $\mathbf{x}$ with $h(\mathbf{x}) > 0$ does not guarantee that our system is persistently feasible, *i.e.*, the system might become unsafe even if the system is initialized at $\mathbf{x}(0) = \mathbf{x}_0$ with $h(\mathbf{x}_0) > 0$. This is due to the fact that when we reach $h(\mathbf{x}) = 0$, we can't change $\omega$ to make the optimization problem feasible while simultaneously enforcing the optimal-decay CBF constraint (22c) and the input constraint. This makes the control barrier function invalid with respect to the safe set defined in (3) due to the input constraint. In this case, the CBF-based control policy isn't safe, i.e., it is not forward complete, see [6, Def. 1]. However, given any invariant set that is a subset of the safe set $\mathcal{C}$, our optimal-decay CBF-QP/CLF-CBF-QP will be persistently feasible if the initial state lies inside this safe invariant set.

The maximum invariant set for an optimal-decay controller could be different with respect to hyperparameters, such as $p_\omega$ and $\omega_0$, which will be illustrated in Sec. VI-D. Notice that solving an explicit representation of this control invariant set for a nonlinear system is usually intractable [27]–[29] and this exceeds the scope of this paper. We will discuss the invariant safety and related performance in future work.

**Remark 7.** When $p_\omega$ tends to $+\infty$ in (21) and (22), then given a state $\mathbf{x}$, if the original CBF constraint is satisfied, *i.e.*,

$$\exists \mathbf{u} \in \mathcal{U}_{adm}(\mathbf{x}), \ L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} \geq -\alpha(h(\mathbf{x})),$$

we will have the optimized value of $\omega$ as $\omega^*(\mathbf{x}) = \omega_0$. Specifically, when $\omega_0 = 1$, it will make the optimal-decay CBF-QP and CLF-CBF-QP as the original CBF-QP/CLF-CBF-QP. However, when the original CBF-QP and CLF-CBF-QP become infeasible, *i.e.*,

$$\forall \mathbf{u} \in \mathcal{U}_{adm}(\mathbf{x}), \ L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} < -\alpha(h(\mathbf{x})),$$

our optimal-decay form is still feasible and the optimal value of $\omega$ is as follows

$$\omega^*(\mathbf{x}) = \sup_{\mathbf{u} \in \mathcal{U}_{adm}} \frac{L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u}}{-\alpha(h(\mathbf{x}))}.$$

To sum up, when $p_\omega = +\infty$ and $\omega_0 = 1$, the optimal-decay CBF-QP and optimal-decay CLF-CBF-QP are equivalent to original CBF-QP and CLF-CBF-QP, if the original ones are already feasible. When the original constraint is infeasible, the optimization variable $\omega$ makes the proposed one still feasible.

## VI. CASE STUDY: ADAPTIVE CRUISE CONTROL

Having presented our optimal-decay CBF-QP/CLF-CBF-QP formulation, we proceed to validate the proposed strategy using an adaptive cruise control (ACC) example, which has been commonly used to validate safety-critical control strategies [1], [12].

### A. Simulation Setup

Consider a point-mass model of an ego vehicle moving along a straight line to follow a lead vehicle. The dynamics are given as follows

$$\dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ -\dfrac{1}{m}F_r(\mathbf{x}) \\ v_l - x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{m} \\ 0 \end{bmatrix} \mathbf{u}, \tag{23}$$

where $(x_1, x_2)$ are the position and velocity $(x_2 = \dot{x}_1)$ of the ego vehicle, $m$ is the mass of the vehicle, and $x_3$ is the distance between the vehicle and the lead vehicle traveling at a velocity of $v_l$. $F_r$ represent the aerodynamic drag and we use the empirical form given as below

$$F_r = f_0 + f_1 x_2 + f_2 x_2^2, \tag{24}$$

where $f_0$, $f_1$, and $f_2$ are empirical constants.

A CLF-CBF-QP controller is designed to solve this ACC problem. In order to regulate speed, we pick a control Lyapunov function as follows

$$V = (x_2 - v_d)^2. \tag{25}$$

A control barrier function is used to guarantee that the ego vehicle will not collide with the leading vehicle. We consider a control barrier function as

$$h = x_3 - 1.8x_2, \tag{26}$$

where the factor of 1.8 is a result of converting to SI units. Besides speed regulation and safe distance maintenance, we also consider input constraints as follows

$$c_d mg \leq \mathbf{u} \leq c_a mg, \tag{27}$$

where $c_a$ and $c_d$ are coefficients for maximum and minimum wheel forces.

Using the control Lyapunov function (25) and the control barrier function (26), we design an original CLF-CBF-QP controller and an optimal-decay CLF-CBF-QP controller with formulation in (9) and (22), respectively. The system is simulated with the two controllers with a frequency at 100Hz, and the values of parameters in the simulations are shown in Tab. I. Notice that in the simulations, we consider $\alpha(.)$ and $\gamma(.)$ in (9) and (22) as linear functions with constant coefficients $\alpha$ and $\gamma$.

### B. Point-wise Feasibility

To compare the performance between our optimal-decay CLF-CBF-QP and the original form, we simulate the system with different initial conditions and illustrate them together in Fig. 3, where solid and dashed lines represents the results from our optimal-decay CLF-CBF-QP and the original form, respectively. For our optimal-decay form, we firstly pick the hyperparameters $\omega_0$ and $p_\omega$ as 1 and $10^8$, which corresponds to the choice we discussed in Sec. V-B.

Precisely, the ego vehicle is initialized at origin $x_1(0) = 0$ m, the initial distance between the ego car and the leading one is set as $x_3(0) = 100$ m and initial speed $x_2(0)$ ranges from 26 m/s to 32 m/s, shown in Fig. 3b. We can

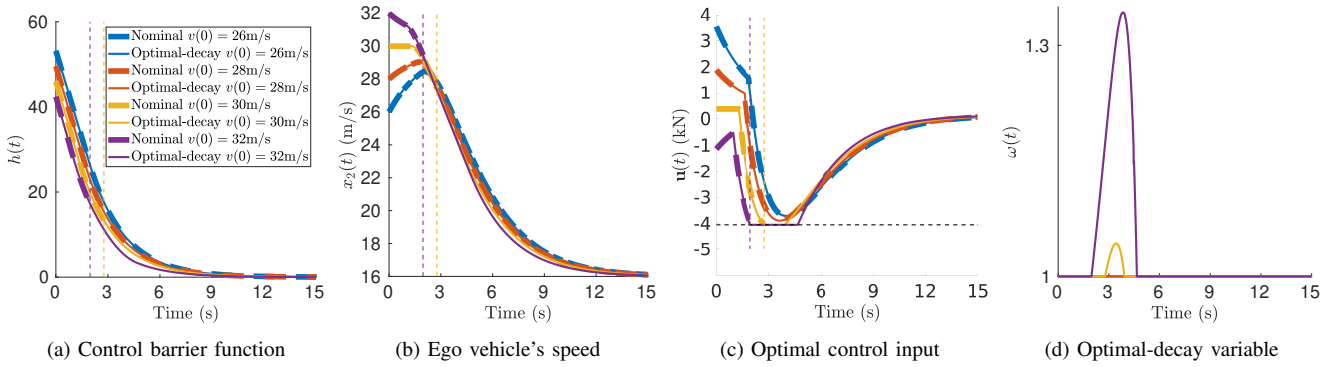(a) Control barrier function  (b) Ego vehicle's speed  (c) Optimal control input  (d) Optimal-decay variable

Fig. 3: Simulation results of adaptive cruise control using original/optimal-decay form of CLF-CBF-QP with different initial velocities. The solid and dashed lines correspond to simulation with optimal-decay CLF-CBF-QP and original CLF-CBF-QP, respectively. Different colors represent different initial conditions. The colorful vertical dashed lines in Fig. 3b represents when the original CLF-CBF-QP becomes infeasible and we can clearly see that the original CLF-CBF-QP becomes infeasible for purple ($x_2(0) = 32$m/s) and yellow ($x_2(0) = 30$m/s) initial conditions.
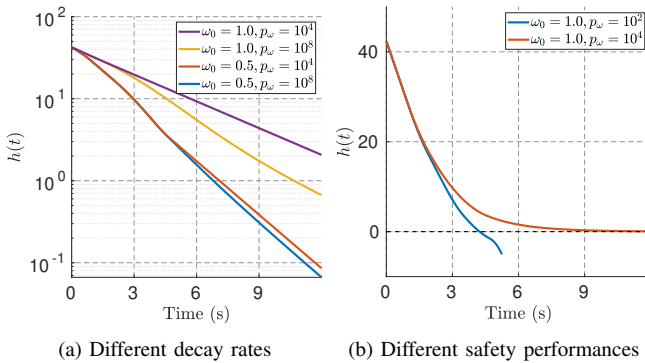


(a) Different decay rates  (b) Different safety performances

Fig. 4: Illustration of the influence of hyperparameters $\omega_0$ and $p_\omega$ on CBF decay rates and system safety performance. The left plot is in log scale on the y-axis to show the difference in the decay rate.

### System Parameters

| | | | | | |
|---|---|---|---|---|---|
| $m$ | 1650 kg | $v_l$ | 16 m/s | $v_d$ | 30 m/s |
| $f_0$ | 0.1 N | $f_1$ | 5 Ns/m | $f_2$ | 0.25 Ns$^2$/m$^2$ |
| $c_d$ | -0.25 | $c_a$ | 0.25 | $g$ | 9.81 m/s$^2$ |

### Controller Parameters

| | | | | | |
|---|---|---|---|---|---|
| $\gamma$ | 1.0 | $p$ | 1.0 | $\alpha$ | 0.5 |

TABLE I: Parameter values used in simulation.

different values of hyperparameters $\omega_0$ and $p_\omega$. We keep the same Lyapunov function and control barrier function with the same parameters in Table. I and the simulation results are shown in Fig. 4. In Fig. 4a, we can observe that smaller $\omega_0$ and larger $p_\omega$ would make the control barrier function decay slower, which verified exactly what was stated in Sec. V-B.

### D. Safety with hyperparameters

The simulations in Sec. VI-A and VI-C that we have seen are all safe with set invariance on $\mathcal{C}$ defined in (3). However, for the same initial condition $x_2(0) = 32$ m/s, if we decrease $p_\omega$, the system could be unsafe after a while. Specifically, we show that when $\omega_0 = 1, p_\omega = 10^2$, our system actually becomes unsafe after around 4s. This happens since the initial condition no longer lies inside the invariant set for the optimal-decay CLF-CBF-QP with this configuration of hyperparameters, which was previously discussed in Sec. V-C. It must be noted that while a formulation guarantees point-wise feasibility for $h(\mathbf{x}) > 0$, this does not guarantee persistent feasibility for $h(\mathbf{x}) > 0$ in a sufficiently long-time trajectory. This is seen in Fig. 4b, where point-wise feasibility for all states with $h(\mathbf{x}) > 0$ still leads to the system unsafe after a while.

### VII. CONCLUSION

In this paper, we have analyzed the point-wise feasibility problem of safety critical control from the perspective of input-space and state-space. Then, in order to deal with the conflict between CBF constraint and input constraint, we

see that CLF-CBF-QP with initial speed as 30 m/s and 32 m/s will become infeasible during the simulations. This infeasibility comes from the conflict between input constraint and original CBF constraint and we can see clearly that the CLF-CBF-QP fails with input saturation and our optimal-decay form handles the problem properly, shown in Fig. 3c. The simulation with initial speed as 26 m/s and 28 m/s does not encounter any infeasible issues where the CLF-CBF-QP and our optimal-decay form shared the same control performance. Moreover, we notice $\omega$ becomes not equal to one only when there is a conflict between the input constraint and the original CBF constraint, shown in Fig. 3d. The evolution of control barrier function for two controllers are shown in Fig. 3a. To sum up, the optimal-decay CLF-CBF-QP is equivalent to the original one when $\omega_0 = 1$ if the original one is already feasible, while still being as a feasible problem even when the original CLF-CBF-QP is infeasible.

### C. Convergence rate with hyperparameters

As we have stated in Sec. V-C, the system might become unsafe if it is initialized outside its invariant set. To illustrate this property, we simulate the controller from an initial speed $x_2(0) = 32$ m/s and leading distance $x_3(0) = 100$ m with

proposed an optimal-decay form for CBF-QP and CLF-CBF-QP to guarantee point-wise feasibility inside the safe set. We numerically verified this control design in an ACC example.

## REFERENCES

[1] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE International Conference on Decision and Control*, 2014.

[2] 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*, 2016.

[3] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *The International Journal of Robotics Research*, 2018.

[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*, 2017.

[5] J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," *Robotics: Science and Systems*, 2020.

[6] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, 2019.

[7] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds," in *American Control Conference*, 2015, pp. 2038–2044.

[8] Q. T. Nguyen, "Robust and Adaptive Dynamic Walking of Bipedal Robots," Ph.D. dissertation, Carnegie Mellon University, 2017.

[9] 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.

[10] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *American Control Conference (ACC)*, 2020.

[11] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016, pp. 322–328.

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

[13] X. Li and C. Belta, "Temporal logic guided safe reinforcement learning using control barrier functions," *arXiv preprint arXiv:1903.09885*, 2019.

[14] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3387–3395.

[15] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 708–717.

[16] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," *arXiv preprint arXiv:2004.03315*, 2020.

[17] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning hybrid control barrier functions from data," *arXiv preprint arXiv:2011.04112*, 2020.

[18] A. Robey, L. Lindemann, S. Tu, and N. Matni, "Learning robust hybrid control barrier functions for uncertain systems," *arXiv preprint arXiv:2101.06492*, 2021.

[19] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," *arXiv preprint arXiv:2007.11718*, 2020.

[20] M. Cohen and C. Belta, "Approximate optimal control for safety-critical systems with control barrier functions," *arXiv preprint arXiv:2008.04122*, 2020.

[21] R. Freeman and P. V. Kokotovic, *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media, 2008.

[22] 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*, 2014.

[23] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs," in *Control of Cyber-Physical Systems*. Springer, 2013, pp. 219–240.

[24] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs," *IEEE Access*, vol. 3, pp. 323–332, 2015.

[25] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] R. G. Bartle and D. R. Sherbert, *Introduction to real analysis*. Wiley New York, 2000, vol. 2.

[27] S. V. Rakovic and M. Baric, "Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1599–1614, 2010.

[28] N. Athanasopoulos, G. Bitsoris, and M. Lazar, "Construction of invariant polytopic sets with specified complexity," *International Journal of Control*, vol. 87, no. 8, pp. 1681–1693, 2014.

[29] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3665–3670, 2017.