Invariance Guarantees using Continuously Parametrized Control Barrier Functions

Inkyu Jang, *Graduate Student Member, IEEE*, and H. Jin Kim, *Member, IEEE*

***Abstract*— We look into the problem of permanently constraining a controlled dynamical system within a prescribed bound, i.e., constructing an invariant set within a free space. Such problem is often approached by formulating a control barrier function (CBF). However, it is often very difficult to synthesize (and usually impossible to handcraft) a valid CBF when the environment becomes large or complex. In this paper, instead of searching for a single CBF that spans the entire free space, we focus on the fact that the union of control invariant sets is also control invariant. Leveraging the relative ease of synthesizing a CBF with a simple and small invariant set, we introduce a way to utilize a spectrum of differentiably parametrized CBFs (PCBFs) with parameter constraints specified. Enjoying the differentiable structure of the parameter space, control is applied on the augmented state space that comprises the original state and the parameter. A feedback controller based on quadratic programming (QP), namely PCBF-QP, is derived. PCBF-QP is capable of generating invariance-guaranteeing input at a low computational cost. The concept is also extended to cover high-relative-degree CBFs. The proposed approach is validated in three different simulation experiment scenarios, including mobile robot navigation through cluttered space, constrained stabilization of a linear system, and adaptive cruise control.**

***Index Terms*— Safety-critical control, constrained control, nonlinear systems, robotics.**

I. Introduction

Ensuring safety when designing a control law for a controlled system is very important in many real-world applications, and safety-critical control therefore has become one of the most popular topics in the field of control systems engineering. In order to address not only myopic but persistent satisfaction of the safety requirements, safety should be addressed from the perspective of set invariance [1]. A typical safety-critical control methodology therefore aims for creating a control invariant set that is entirely contained within the pregiven set of allowable states. An invariance-preserving control is then applied to keep the system's state within this set.

One of the most widely used approaches to construct a control invariant set is to utilize a control barrier function (CBF) [2]. CBF is a Lyapunov-like scalar function defined on the state space, whose super zero level set defines the control invariant set. Its main strength comes from its simplicity of encoding invariance using a single scalar function. In addition, once synthesized, a valid CBF offers a computationally efficient means of enforcing safety constraints through quadratic programming (QP), namely CBF-QP [3]. CBF-QP consists of a single inequality constraint on the input, and can be solved in real-time by any off-the-shelf convex programming solver.

These advantages offered by CBF and CBF-QP have drawn researchers' significant attention, leading to an extensive body of literature regarding practical applications, especially in robotics [4]–[8] and also in other topics [9], [10]. There also have been works to employ CBFs in a wider range of applications, for example, safety-critical reinforcement learning [11], control of systems with stochasticity [12], [13], parameter-adaptive CBFs [14], time-varying CBFs for satisfaction of signal temporal logic specifications [15]. Recently, the concept of high-order CBF (HOCBF) [16] was introduced to address safety constraints of high relative degree.

For the sake of control performance, it is obviously important to obtain a valid CBF that provides a sufficiently large control invariant set within the prescribed limit. Unfortunately, synthesizing one given the system dynamics and the environment information only is in general not straightforward and remains an open problem, especially in large or complex environments. This is mainly because the condition for a function to qualify as a valid CBF turns out to be in the form of partial differential inequalities, and the environmental conditions (such as obstacle positions) appear as boundary conditions for the inequality. Handcrafting a valid CBF requires great expertise (and is almost impossible even when the environment becomes slightly more complex), and systematically synthesizing one suffers from extremely heavy computational burden. As a result, most of the recent works on CBFs showed to be successful only in very simple environments.

To circumvent this burden, recalling the well-known fact that the union of control invariant sets is also a control invariant set [17, Proposition 4.13], one can contemplate an idea of stitching together multiple simple-shaped control invariant sets to cover a large and complex workspace, instead of searching for a single large one. In this paper, we propose a way to integrate this idea into the CBF framework, allowing the use of simple building-block CBFs to construct a large control invariant set. To elaborate, we consider a differentiably parametrized spectrum of CBFs, which we call parametrized CBF (PCBF). Leveraging the differentiable structure of the parameter space, we devise a CBF-QP-like safety filter based on PCBF (PCBF-QP), which is capable of constraining the

*This work was not supported by any organization
The authors are with the Department of Aerospace Engineering and Automation and Systems Research Institute (ASRI), Seoul National University, 08826 Seoul, Korea. janginkyu.larr@gmail.com, hjinkim@snu.ac.kr

system within the union of small invariant sets derived from each corresponding parameter-specific CBF. It can also address multiple (and possibly infinite) parameter constraints simultaneously while maintaining the QP structure.

In addition to that, following the motivation given by HOCBFs [16], the notion of PCBF is extended to cover high-relative-degree state constraints. Similar to PCBFs, high-order PCBF (HOPCBF) is defined as a differentiably parametrized spectrum of HOCBFs, and we derive the set of input constraints for invariance guarantees using a HOPCBF. A QP-based safety filter using HOPCBF (HOPCBF-QP) is then proposed, which is capable of generating inputs satisfying the HOPCBF constraints at a low computational cost.

The proposed PCBF framework fits well into many practical real-world applications. For example, one can make use of the inherent symmetry of the dynamics model. Such symmetry-induced PCBFs are especially common in the case of mobile robots. It can be shown that if the dynamics exhibits a continuous symmetry, then we can construct a symmetry-induced PCBF out of a single CBF (or PCBF). Another possibility is to consider a class of parametrized differentiable function primitives and specify conditions on the parameters for the resulting function to be a CBF.

We look into three practical control examples where PCBFs can be applied. The first example uses symmetry-induced PCBFs to allow a ground rover navigate through obstacle-cluttered space without running into a collision. In the second example, we consider a stabilization task of a linear system through PCBF constructed using simple quadratic function primitives. Lastly, we consider an adaptive cruise control problem to show the effectiveness of HOPCBF and HOPCBF-QP.

This work is an extension of our recent conference paper [18], which briefly introduced PCBF and PCBF-QP. Upon that, it adds proofs to the key theorems, additional discussions regarding PCBF, the concept of HOPCBF, and some experimental results. The remainder of this paper is organized as follows. In Section II, we give a brief overview on some necessary concepts for understanding PCBF, including CBF and CBF-QP. In Section III, PCBF and PCBF-QP are introduced. The proposed PCBF framework is extended to high-relative-degree forms in Section IV. In Section V, we mention some possible extensions of PCBFs and HOPCBFs, including symmetry-induced and time-varying PCBFs. The simulation results are presented in Section VI, which is followed by the summary and outlook of the work in Section VII.

II. Preliminaries

A. Notation

For positive integers l, m, and n, $\mathbb{R}^l$ and $\mathbb{R}^{m\times n}$ denote the set of l-dimensional real column vectors and matrices of size $m \times n$, respectively. By $\mathbb{S}^n$, we denote the set of symmetric square matrices, and by $\mathbb{S}^n_+$ the set of symmetric positive semidefinite matrices of size $n \times n$. An inequality between two vectors denotes that it is satisfied in an elementwise manner. One between two square matrices represents definiteness conditions, for example, for two symmetric square matrices $A, B \in \mathbb{S}^n$, $A \geq B$ means $A - B \in \mathbb{S}^n_+$.

We use the notation $\partial_\xi \beta(\xi)$ to denote the partial derivative of β with respect to argument ξ. The Lie derivative of a function $V : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is written as $L_f V(x) = \partial_x V(x) \cdot f(x)$. If a function f is r times continuously differentiable, we write $f \in \mathcal{C}^r$.

Throughout this paper, we use the letters x, u, k, and t to denote state, input, CBF parameter, and time, respectively. The roman-font x, u, k are used to emphasize that they are *trajectories*, i.e., functions of time $t \in [0, \infty)$.

B. Dynamics

In this paper, we consider the following nonlinear time-invariant control-affine system dynamics:

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

where $x \in \mathbb{R}^n$ is the state, $u \in U \subseteq \mathbb{R}^m$ is the possibly bounded control input. The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz functions. The control limit U is assumed to be defined by a set of linear inequalities, i.e.,

$$u \in U = \{u \in \mathbb{R}^m : A_u u \leq b_u\}, \tag{2}$$

where A_u and b_u are a matrix and a column vector with appropriate sizes.

C. Set Invariance and Control Barrier Functions

Set invariance is a key concept in ensuring safety of a system. We first begin with the definition of control invariant sets.

Definition 1 (Control Invariance)**.** A set $C \subseteq \mathbb{R}^n$ is control invariant for the system (1) if for every $\mathrm{x}(0) \in C$, there exists an input trajectory $\mathrm{u} : [0, \infty) \to U$ that makes the resulting state trajectory reside permanently in C, i.e., $\mathrm{x}(t) \in C$, for all $t \in [0, \infty)$.

CBF is a powerful tool used to handle control invariance of a set [2]. Suppose the set C is given as a super zero level set of a $\mathcal{C}^1$ function $h : \mathbb{R}^n \to \mathbb{R}$, such that

$$\begin{aligned} C &= \{x \in \mathbb{R}^n : h(x) \geq 0\}, \\ \partial C &= \{x \in \mathbb{R}^n : h(x) = 0\}, \\ \mathrm{Int}\ C &= \{x \in \mathbb{R}^n : h(x) > 0\}. \end{aligned} \tag{3}$$

Definition 2 (Control Barrier Function (CBF))**.** A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is a CBF for the dynamical system (1) if there exists a class $\mathcal{K}$ function[1] $\alpha : [0, \infty) \to \mathbb{R}$ such that

$$\exists u \in U \text{ s.t., } L_f h(x) + L_g h(x) \cdot u + \alpha(x) \geq 0 \tag{4}$$

for all $x \in C$, and $\partial_x h(x) \neq 0$ for all $x \in \partial C$.

Definition 2 is also known by the name *zeroing* CBF (ZCBF) [3], in the sense that the value of h drops to zero when it approaches the boundary of set C. It should be clearly noted that a CBF can be defined only when the class $\mathcal{K}$ function α is specified.

[1]A function $\alpha : [0, \infty) \to \mathbb{R}$ belongs to class $\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$.

Now, let

$$U_{\text{cbf}}(x) = \{u \in U : L_f h(x) + L_g h(x) \cdot u + \alpha(x) \geq 0\}. \quad (5)$$

It can be easily seen that Definition 2 ensures $U_{\text{cbf}}(x)$ is nonempty for all $x \in C$. Any feedback controller $u(x,t)$ will render the set C invariant if $u(x,t) \in C$ for all $t \in [0,\infty)$. The following theorem is a result from Nagumo's theorem [17, Section 4.2].

Theorem 1 (Invariance Guarantees thorough CBF)**.** *Let $u(x,t)$ be a Lipschitz continuous feedback law such that $u(x,t) \in U_{\text{cbf}}(x)$ for all $x \in C$ and $t \in [0,\infty)$. If a state trajectory $\mathrm{x} : [0,\infty) \to \mathbb{R}^n$ solves $\partial_t \mathrm{x}(t) = f(\mathrm{x}(t)) + g(\mathrm{x}(t))u(\mathrm{x}(t),t)$ and $\mathrm{x}(0) \in C$, then $\mathrm{x}(t) \in C$ for all $t \in [0,\infty)$.*

One widely-used way of synthesizing a control law $u(x,t)$ that satisfies the CBF constraint is to formulate an optimization optimization problem in the following form.

Problem 1 (CBF-QP [3])**.** Given the dynamical system (1), solve

$$u(x,t) = \arg\min_{u \in U} \; J(x,u,t), \quad (6)$$

subject to

$$L_f h(x) + L_g h(x) \cdot u + \alpha(h(x)) \geq 0, \quad (7)$$

where $J(x,u,t)$ is a convex cost function quadratic with respect to u.

The most popular choice of the cost function is

$$J(x,u,t) = \|u - u_{\text{ref}}(x,t)\|^2, \quad (8)$$

where $u_{\text{ref}}(\cdot,\cdot)$ is a given (possibly feedback) reference input signal. Since CBF-QP just adds one inequality constraint to the input constraint (2), it can be solved at a relatively low computational cost.

III. Parametrized CBF

A. Parametrized CBF

Consider a spectrum of CBF candidates $h : \mathbb{R}^n \times \mathbb{R}^{n_k} \to \mathbb{R}$ parametrized by parameter $k \in \mathbb{R}^{n_k}$, and say $h(\cdot,k)$ is a valid CBF for every $k \in K \subseteq \mathbb{R}^{n_k}$. Let the set-valued function $C : \mathbb{R}^{n_k} \to 2^{\mathbb{R}^n}$ be

$$C(k) = \{x \in \mathbb{R}^n : h(x,k) \geq 0\} \quad (9)$$

be the zero superlevel set of $h(\cdot,k)$ for every fixed k. If $k \in K$, $h(\cdot,k)$ is a CBF, and $C(k)$ is a control invariant set. Since the union of control invariant sets is also control invariant,

$$C = \bigcup_{k \in K} C(k) \subseteq \mathbb{R}^n \quad (10)$$

is a control invariant set. The goal of this section is to construct a CBF-like control framework using $h(x,k)$ without searching for a new single CBF that covers C.

Assume the parameter set K is defined as

$$K = \{k \in \mathbb{R}^{n_k} : \rho_i(k) \geq 0, \; \forall i \in I\}, \quad (11)$$

where I is an (possibly infinite) index set, and $\rho_i \in \mathcal{C}^1$, for each i, denotes a parameter constraint. The similar regularity properties are assumed for ρ_i ($i \in I$), i.e.,

$$\partial_k \rho_i(k) \neq 0, \quad \forall k \in \mathbb{R}^{n_k} \text{ s.t. } \rho_i(k) = 0. \quad (12)$$

As mentioned above, $h(\cdot,k)$ for each $k \in K$ being a CBF implies that there exists a class $\mathcal{K}$ function $\alpha(\cdot,k)$ such that

$$\exists u \in U, \text{ s.t. } L_f h(x,k) + L_g h(x,k) \cdot u + \alpha(h(x,k),k) \geq 0. \quad (13)$$

Here, we admitted some abuse of notation $L_f h(x,k) = \partial_x h(x,k) \cdot f(x)$, and similarly for L_g, to denote *partial* Lie derivatives.

With that, we define PCBF as follows.

Definition 3 (Parametrized CBF)**.** A function $h : \mathbb{R}^n \times \mathbb{R}^{n_k} \to \mathbb{R}$ is a parametrized CBF (PCBF) if there exists a continuous function $\alpha : \mathbb{R} \times \mathbb{R}^{n_k} \to \mathbb{R}$ such that, for every fixed $k \in K$,

- $\alpha(\cdot,k)$ is a class $\mathcal{K}$ function,
- $h(\cdot,k)$ is a valid CBF with $\alpha(\cdot,k)$, i.e., (13) holds.

One naive way of utilizing PCBF to keep the state within C is to pick a parameter $k \in K$ which makes $h(x,k) \geq 0$, along with the control input u that satisfies the CBF constraint. The optimization-based controller would take the following form, similar to [7]:

$$\begin{aligned} \min_{u,k} \; & J(x,k,u,t) \\ \text{s.t. } & L_f h(x,k) + L_g h(x,k) \cdot u + \alpha(h(x,k),k) \geq 0 \\ & h(x,k) \geq 0 \\ & \rho_i(k) \geq 0, \quad \forall i \in I \\ & u \in U. \end{aligned} \quad (14)$$

The problem in this form is that (14) is generally a nonlinear and nonconvex problem whose optimization typically requires heavy computation, making it inappropriate for real-time feedback synthesis. Thus, instead of directly optimizing over the parameter space, we will require the parameter k to *evolve continuously* with respect to time by *controlling* it with through its time derivative $\dot{k}$. Consider the following augmented system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ \dot{k} &= v, \end{aligned} \quad (15)$$

where $(x,k) \in \mathbb{R}^n \times \mathbb{R}^{n_k}$ is the augmented state, $(u,v) \in \bar{U} = U \times \mathbb{R}^{n_k}$ is the augmented input. Now, k is no longer an optimization variable but the controller's internal state that is controlled by the virtual input $v \in \mathbb{R}^{n_k}$. With respect to the augmented system (15), consider the disjoint union of $C(k)$, i.e.,

$$\begin{aligned} \bar{C} &= \bigsqcup_{k \in K} C(k) \\ &= \{(x,k) \in \mathbb{R}^n \times \mathbb{R}^{n_k} : h(x,k) \geq 0, \rho_i(k) \geq 0, \; \forall i \in I\}. \end{aligned} \quad (16)$$

We will construct a constraint on the augmented input to render $\bar{C}$ invariant with respect to the augmented system. Since the projection of $\bar{C}$ onto the original state space $\mathbb{R}^n$ is C, invariance of $\bar{C}$ directly relates to invariance of C.

A necessary condition for $\bar{C}$ to be control invariant with respect to the augmented system is the invariance of K with respect to the parameter dynamics (the second line of (15)). Thus, for every $k \in K$, there must exist a virtual input $v \in \mathbb{R}^{n_k}$ such that

$$\rho_i(k) = 0 \;\Rightarrow\; \frac{d}{dt}\rho_i(k) = \partial_k \rho_i(k) \cdot v \geq 0 \qquad (17)$$

for all $i \in I$. Following the motivation of barrier function approaches including CBFs, we *smoothen* this requirement by introducing the inequality constraint

$$\frac{d}{dt}\rho_i(k) + \beta_i(\rho_i(k)) = \partial_k \rho_i(k) \cdot v + \beta_i(\rho_i(k)) \geq 0, \qquad (18)$$

for all $k \in K$ and $i \in I$. Here, β_i, for every i, is a class $\mathcal{K}$ function. Also, for h to be kept nonnegative, we require

$$\begin{aligned}&\frac{d}{dt}h(x,k) + \alpha(h(x,k),k)\\ &= L_f h(x,k) + L_g h(x,k)\cdot u + \partial_k h(x,k)\cdot v + \alpha(h(x,k),k)\\ &\geq 0.\end{aligned} \qquad (19)$$

Notice the difference between (19) and the original CBF constraint (the first constraint of (14)). The added term $\partial_k h(x,k) \cdot v$ allows the controller to select the control u from a wider range, compared to (14). We discuss this in Section III-C.

Combining the two requirements (18) and (19), we can consider the following feasible set on the augmented input space:

$$\begin{aligned}&\bar{U}_{\mathrm{pcbf}}(x,k) =\\ &\left\{(u,v) \in \bar{U} : \begin{array}{l} L_f h + L_g h\cdot u + \partial_k h \cdot v + \alpha(h,k) \geq 0, \\ \partial_k \rho_i \cdot v + \beta_i(\rho_i) \geq 0,\ \forall i \in I \end{array}\right\},\end{aligned} \qquad (20)$$

where (and hereafter when needed) the arguments of h and ρ_i are omitted for brevity.

We now prove that $\bar{C}$ can be made invariant with respect to (15) through a feedback controller that satisfies the PCBF constraint $(u(x,k,t), v(x,k,t)) \in \bar{U}_{\mathrm{pcbf}}(x,k)$.

Theorem 2. *The PCBF (augmented) input constraint set $\bar{U}_{\mathrm{pcbf}}(x,k)$ is nonempty for all $(x,k) \in \bar{C}$.*

Proof. Since h satisfies (13), for each $(x,k) \in \bar{C}$, there exists an input $u^* \in U$ such that $L_f h(x,k) + L_g h(x,k) \cdot u^* + \alpha(h(x,k),k) \geq 0$. Then, $(u = u^*, v = 0) \in \bar{U}_{\mathrm{pcbf}}(x,k)$. □

Theorem 3. *Let $(u(x,k,t), v(x,k,t))$ be a feedback strategy for the augmented system (15) such that it is locally Lipschitz with respect to x and k, measurable with respect to t, and $(u(x,k,t), v(x,k,t)) \in \bar{U}_{\mathrm{pcbf}}(x,k)$ for all $(x,k) \in \bar{C}$, $t \in [0,\infty)$. If a trajectory of augmented state $(\mathrm{x}(\cdot), \mathrm{k}(\cdot)) : [0,\infty) \to \mathbb{R}^n \times \mathbb{R}^{n_k}$ solves the closed-loop dynamics*

$$\begin{aligned}\dot{\mathrm{x}}(t) &= f(\mathrm{x}(t)) + g(\mathrm{x}(t))u(\mathrm{x}(t),\mathrm{k}(t),t)\\ \dot{\mathrm{k}}(t) &= v(\mathrm{x}(t),\mathrm{k}(t),t)\end{aligned} \qquad (21)$$

starting from an initial condition $(\mathrm{x}(0), \mathrm{k}(0)) \in \bar{C}$, then $(\mathrm{x}(t), \mathrm{k}(t)) \in \bar{C}$ for all $t \in [0,\infty)$.

Proof. Similar to Theorem 1, the proof directly follows from Nagumo's theorem. Whenever $(x,k) \in \partial\bar{C}$, i.e., either $h(x,k) = 0$ or $\rho_i(k) = 0$ for some i, the velocity vector of the augmented dynamics (15) $(\dot{x}, \dot{k}) = (f(x) + g(x)u(x,k,t), v(x,k,t))$ will point into the direction where all the active constraints are nondecreasing. □

From the above two theorems, we conclude that (20) works as a valid barrier certificate for $\bar{C}$. It is also notable that $\bar{U}_{\mathrm{pcbf}}(x,k)$ consists of linear inequalities only, given the assumption (2) on the input set. We make the best use of these properties to construct a CBF-QP-like safety filter utilizing PCBF, namely PCBF-QP.

B. PCBF-based QPs for Invariance Guarantees

Consider the following optimization-based controller, which we call PCBF-QP.

Problem 2 (PCBF-QP)**.** Given the augmented dynamical system (15) and PCBF h, solve

$$(u(x,k,t), v(x,k,t)) = \arg\min_{(u,v)\in\bar{U}} J(x,k,u,v,t), \qquad (22)$$

subject to

$$L_f h(x,k) + L_g h(x,k)\cdot u + \partial_k h(x,k)\cdot v + \alpha(h(x,k),k) \geq 0 \qquad (23)$$

and

$$\partial_k \rho_i(k) \cdot v + \beta_i(\rho_i(k)) \geq 0, \ \ \forall i \in I, \qquad (24)$$

where $J(\cdots)$ is a cost function that is (jointly) convex quadratic with respect to u and v.

A decent choice of the cost function J that works fine for many cases is to let J take the similar form as CBF-QP:

$$J(x,k,u,v,t) = \|u - u_{\mathrm{ref}}(x,t)\|^2 + \mu\|v\|^2, \qquad (25)$$

where u_{ref} is the reference input signal, $\mu > 0$ is a tunable parameter employed to ensure strict convexity of (22). Positive μ provides existence and uniqueness of the QP solution and enhances numerical stability of the controller.

C. Expressivity of Input

One remarkable advantage offered by PCBF is that it provides a broader feasible region for the optimization-based controller. Let

$$U_{\mathrm{pcbf}}(x,k) = \{u \in U : \exists v \in \mathbb{R}^{n_k},\ \text{s.t. } (u,v) \in \bar{U}_{\mathrm{pcbf}}(x,k)\} \qquad (26)$$

be the image of the PCBF augmented input constraint set $\bar{U}_{\mathrm{pcbf}}(x,k)$ projected onto the original input space U. Compared to the feasible input set by CBF $h(\cdot,k)$ for fixed k

$$\begin{aligned}&U_{\mathrm{cbf}}(x,k)\\ &= \{u \in U : L_f h(x,k) + L_g h(x,k)\cdot u + \alpha(h(x,k),k) \geq 0\},\end{aligned} \qquad (27)$$

it can be found that $U_{\mathrm{pcbf}}(x,k) \supseteq U_{\mathrm{cbf}}(x,k)$ for all (x,k) where the equality holds if and only if $v = 0$ is the only solution for

$$\partial_k \rho_i(k) \cdot v \geq 0, \quad \forall i \in I, \qquad (28)$$

that is, K is a single-element set. It can be further shown that with bounded U and sufficiently large $\beta_i(\rho_i(k), k)$ (i.e., k is far enough from the boundary of K), if $\partial_k h(x,k) \neq 0$, then $U_{\mathrm{pcbf}}(x,k) = U$. For any $u \in U$, take

$$v = -\frac{L_f h + L_g h \cdot u + \alpha(h,k)}{\|\partial_k h\|^2} \cdot (\partial_k h)^\top . \tag{29}$$

This result implies that PCBF-based controllers are less conservative compared to solving CBF-based optimization in a pointwise manner like (14).

IV. PCBF WITH HIGH RELATIVE DEGREE

Definition 4 (Relative Degree [19, Definition 13.2])**.** The output $h : \mathbb{R}^n \to \mathbb{R}$ of the system (1) has relative degree r ($1 \leq r \leq n$) if $L_g L_f^j h(x) = 0$ for all $x \in \mathbb{R}^n$ and $j \in \{0, \cdots, r-2\}$, and $L_g L_f^{r-1} h(x) \neq 0$ almost everywhere (a.e.) on $\mathbb{R}^n$.

Recently, HOCBFs [16] showed to be powerful when dealing with safety constraints of high (greater than one) relative degree. In this section, we extend the concept of PCBF to continuously parametrized safety constraints with high relative degree, which we call HOPCBF. Then, a control strategy to generate invariance-guaranteeing input at low computational cost based on QP, namely HOPCBF-QP, is derived.

A. Review of HOCBF

We first begin with a brief overview of the motivation and theoretical details of HOCBF. Let $h : \mathbb{R}^n \to \mathbb{R}$ be a $\mathcal{C}^r$ output for the system (1) of relative degree $r > 1$. Similar to the $r = 1$ case (CBF and CBF-QP), we want to drive the system state such that $h(x) \geq 0$ is satisfied. To cope with the high relative degree, the vector field f appearing in the dynamics (1) should be at least $\mathcal{C}^{r-1}$, and its $(r-1)$-th order derivative Lipschitz.

First, define a sequence of functions $\psi_{(\cdot)} : \mathbb{R}^n \to \mathbb{R}$ as follows:

$$\begin{aligned} &\psi_0(x) = h(x) \\ &\psi_j(x) = \dot{\psi}_{j-1}(x) + \alpha_j(\psi_{j-1}(x)), \ \forall j \in \{1, \cdots, r-1\} \end{aligned} \tag{30}$$

where $\alpha_j(\cdot)$, $j \in \{1, \cdots, r-1\}$ are class $\mathcal{K}$ functions. For the sake of well-definedness of ψ_j, α_j is assumed to be at least $\mathcal{C}^{r-j}$. Note that the relative degree of ψ_j is at least $r-j$ and thus the term $\dot{\psi}_{j-1}$ in the second line can be written as a function of x only. The key idea of HOCBF is that

$$\psi_j(x) = \dot{\psi}_{j-1}(x) + \alpha_j(\psi_{j-1}(x)) \geq 0 \tag{31}$$

gives $\psi_{j-1}(x) \geq 0$ if $\psi_{j-1}(\cdot)$ value starts at a nonnegative initial condition. If there exists a class $\mathcal{K}$ function α_r such that there exists $u \in U$ for all x that satisfies

$$\begin{aligned} &\frac{d}{dt}\psi_{r-1}(x) + \alpha_r(\psi_{r-1}(x)) \\ &= L_f \psi_{r-1}(x) + L_g \psi_{r-1}(x) \cdot u + \alpha_r(\psi_{r-1}(x)) \geq 0, \end{aligned} \tag{32}$$

then it will initiate a chain of nonnegativity certificates and eventually render the set $C = \bigcap_{j=1}^r C_j$ invariant, where for each $j \in \{1, \cdots, r\}$ the set C_j is defined as

$$C_j = \{x \in \mathbb{R}^n : \psi_{j-1}(x) \geq 0\}. \tag{33}$$

Definition 5 (High-Order CBF [16])**.** A function $h : \mathbb{R}^n \to \mathbb{R}$ having relative degree r with respect to the system (1) is a HOCBF, if there exist class $\mathcal{K}$ functions $\alpha_j \in \mathcal{C}^{r-j}$, $j \in \{1, \cdots, r\}$ such that there exists an input $u \in U$ (depending on x) satisfying

$$L_f \psi_{r-1}(x) + L_g \psi_{r-1}(x) \cdot u + \alpha_r(\psi_{r-1}(x)) \geq 0, \tag{34}$$

for all $x \in C = C_1 \cap \cdots \cap C_r$.

B. High-Order PCBF

Now, we introduce the high-relative-degree version of PCBF and PCBF-QP similar to HOCBF, namely high-order PCBF (HOPCBF) and HOPCBF-QP. We begin with the follwing definition.

Definition 6 (High-Order PCBF)**.** A function $h : \mathbb{R}^n \times \mathbb{R}^{n_k} \to \mathbb{R}$ is a HOPCBF of relative degree r if there exist functions α_j $(\in \mathcal{C}^{r-j}) : \mathbb{R} \times \mathbb{R}^{n_k} \to \mathbb{R}$ ($j \in \{1, \cdots, r-1\}$) such that $\alpha_j(\cdot, k)$ are class $\mathcal{K}$ functions that make $h(\cdot, k)$ a valid HOCBF of relative degree r for every fixed $k \in K$.

With this definition, one might attempt to obtain the function sequence (30). Unfortunately, considering the augmented system (15) with nonzero parameter change rate $\dot{k} \neq 0$, the HOPCBF Definition 6 does not in general give well-defined function sequence $\psi_{(\cdot)}$ like the HOCBF's case, since there no longer is a guarantee that the time derivatives of $\psi_{(\cdot)}$ exist. This is because the parameter trajectory satisfying (24) might not be r times differentiable. This is due to the mismatch in the relative degrees of the HOPCBF with respect to $\dot{k}$ (which is one) and to u (which is r by assumption). Instead, we define another sequence of functions $\phi_{j-1}(x,k)$ ($j \in \{1, \cdots, r\}$) as

$$\begin{aligned} &\phi_0(x,k) = h(x,k) \\ &\phi_j(x,k) = L_f \phi_{j-1}(x,k) + \alpha_j(\phi_{j-1}(x,k), k), \end{aligned} \tag{35}$$

and the sets

$$\begin{aligned} &C_j(k) = \{x \in \mathbb{R}^n : \phi_{j-1}(x,k) \geq 0\}, \ j \in \{1, \cdots, r\}, \\ &C(k) = \bigcap_{j=1}^r C_j(k). \end{aligned} \tag{36}$$

The definitions for $\phi_{(\cdot)}$ are very similar to $\psi_{(\cdot)}$ from (30), and they are actually identical given zero parameter speed (i.e., $\dot{k} = 0$). Observe that for every $k \in K$, the set $C(k)$ is control invariant since $h(\cdot, k)$ is a valid HOCBF. Therefore, we set the control objective as to drive the system's state x and the parameter k such that $x \in C(k)$ and $k \in K$. That is, we want to constrain the augmented system (15) within the set $\bar{C} = \bigsqcup_{k \in K} C(k)$ by keeping $\phi_{j-1}(\cdot)$ values nonnegative. For $j \in \{1, \cdots, r-1\}$, $\phi_j(\cdot) \geq 0$ can be achieved via requiring for all $j \in \{1, \cdots, r-1\}$,

$$\begin{aligned} &\frac{d}{dt}\phi_{j-1}(x,k) + \alpha_j(\phi_{j-1}(x,k), k) \\ &= L_f \phi_{j-1}(x,k) + L_g \phi_{j-1}(x,k) \cdot u + \partial_k \phi_{j-1}(x,k) \cdot v \\ &\quad + \alpha_j(\phi_{j-1}(x,k), k) \\ &= \phi_j(x,k) + \partial_k \phi_{j-1}(x,k) \cdot v \geq 0. \end{aligned} \tag{37}$$

To obtain the last inequality, we used the fact that $\phi_j(x,k) = L_f\phi_{j-1}(x,k) + \alpha_j(\phi_{j-1}(x,k))$, and the term $L_g\phi_{j-1}(x,k)$ reduces to zero since the relative degree of $\phi_{j-1}(\cdot,k)$ with respect to the original dynamics (1) is $r-j+1>1$. And for $j=r$, we achieve $\phi_r(\cdot)\geq 0$ through

$$\begin{aligned}&\frac{d}{dt}\phi_{r-1}(x,k) + \alpha_r(\phi_{r-1}(x,k))\\ &= L_f\phi_{r-1}(x,k) + L_g\phi_{r-1}(x,k) + \partial_k\phi_{r-1}(x,k)\cdot v\\ &\quad + \alpha_r(\phi_{r-1}(x,k),k) \geq 0.\end{aligned} \tag{38}$$

With that, we can define the HOPCBF (augmented) input constraint set

$$\begin{aligned}&\bar{U}_{\text{hopcbf}}(x,k) =\\ &\left\{(u,v)\in\bar{U} : \begin{array}{l}(37)\ \forall j\in\{1,\cdots,r-1\},\ (38),\\ \partial_k\rho_i(k)\cdot v + \beta_i(\rho_i(k))\geq 0,\ \forall i\in I\end{array}\right\}\end{aligned} \tag{39}$$

and obtain the following result similar to Theorem 2 and Theorem 3.

Theorem 4. *The HOPCBF (augmented) input constraint set $\bar{U}_{\text{hopcbf}}(x,k)$ is nonempty for all $(x,k)\in\bar{C}$. Moreover, Theorem 3 also holds for controllers satisfying the HOPCBF constraint $(u(x,k,t), v(x,k,t))\in\bar{U}_{\text{hopcbf}}(x,k)$, $\forall(x,k)\in\bar{C},\ t\in[0,\infty)$.*

Proof. The proof for the former part is the same as Theorem 2, consider $v=0$. The latter part can be shown using Nagumo's theorem, similar to Theorem 3. □

C. HOPCBF-QP

Similar to PCBF-QP, a good way of synthesizing a feedback law that satisfies the HOPCBF constraints is to formulate an optimization problem as follows.

Problem 3 (HOPCBF-QP)**.** Given the augmented dynamical system (15) and a HOPCBF h, solve

$$(u(x,k,t), v(x,k,t)) = \arg\min_{u,v\in\bar{U}} J(x,k,u,v,t), \tag{40}$$

subject to

$$\begin{aligned}&\phi_j(x,k) + \partial_k\phi_{j-1}(x,k)\cdot v\geq 0\quad \forall j\in\{1,\cdots,r-1\},\\ &L_f\phi_{r-1}(x,k) + L_g\phi_{r-1}(x,k)\cdot u\\ &\quad + \partial_k\phi_{r-1}(x,k)\cdot v + \alpha_r(\phi_{r-1}(x,k),k)\geq 0,\\ &\partial_k\rho_i(k)\cdot v + \beta_i\rho_i(k)\geq 0\quad \forall i\in I,\end{aligned} \tag{41}$$

where $J(\cdots)$ is a cost function that is strictly convex quadratic with respect to u and v.

The cost function same with with PCBF-QP (25) can also be used here. The same relationship between CBF and HOCBF also applies to PCBF and HOPCBF: PCBF is a specialization of HOPCBF where the relative degree is one. Furthermore, similar to the the relationship between PCBF and CBF, HOPCBF is a generalization of HOCBF: HOCBF is a special case of HOPCBF when the parameter space K is a single-element set.

V. Extension of PCBF and HOPCBF

Before we present the simulation experiment results, we point out some possible extensions of PCBF and HOPCBF.

A. PCBF Construction using Continuous Symmetry of the Dynamics Model

In this subsection, we introduce a convenient yet powerful method to construct a PCBF out of a building-block CBF (or PCBF) by making use of the inherent symmetry of the system dynamics.

A system model found in real-world often (and almost always for a mobile robot) exhibits a continuous symmetry. This means that the dynamics is invariant under a continuous spectrum of coordinate changes. For example, the kinematics and dynamics of a planar mobile robot can be written in the same form regardless of which $SE(2)$ coordinate we choose.

Mathematically, continuous symmetries can be understood using Lie group actions. If the dynamics (1) exhibits a continuous symmetry, then there exists a Lie group G acting on the state space $\mathbb{R}^n$ such that the dynamics remains invariant with respect to the coordinate change given by the Lie group action. That is, for any $q\in G$, if the state and input trajectory pair $(\mathrm{x}:[0,\infty)\to\mathbb{R}^n, \mathrm{u}:[0,\infty)\to U)$ solves the ODE (1), then also does the pair $(q\circ\mathrm{x},\mathrm{u})$. Equivalently,

$$f(q(x)) + g(q(x))u = \partial_x q(x)\cdot(f(x)+g(x)u) \tag{42}$$

for all $x\in\mathbb{R}^n$ and $u\in U$. Here, q as a function denotes the group action.

Theorem 5. *Let $h_0:\mathbb{R}^n\times\mathbb{R}^{\hat{n}_k}\to\mathbb{R}$ be a PCBF for a system with continuous symmetry, i.e., (42) holds. Then, $h(x,k) = h_0(q^{-1}(x),\hat{k})$ is a PCBF with the new parameter $k=(q,\hat{k})\in G\times\mathbb{R}^{\hat{n}_k}$.*

Proof. Suppose $\alpha_0:[0,\infty)\times\mathbb{R}^{\hat{n}_k}\to\mathbb{R}$ is the parametrized class $\mathcal{K}$ function for the original PCBF h_0. We will show that h is a PCBF with respect to the parametrized class $\mathcal{K}$ function $\alpha(\cdot,k)=\alpha_0(\cdot,\hat{k})$. Suppose $\xi\in C(k)$, i.e., $h(\xi,k) = h_0(q^{-1}(\xi),\hat{k})\geq 0$. Then, there exists an input $u\in U$ such that

$$\begin{aligned}&L_f h(\xi,k) + L_g h(\xi,k)\cdot u + \alpha(h(\xi,k),k)\\ &= L_f(h_{0,\hat{k}}\circ q^{-1})(\xi) + L_g(h_{0,\hat{k}}\circ q^{-1})(\xi)\cdot u\\ &\quad + \alpha(h_{0,\hat{k}}(q^{-1}(\xi)),k)\\ &= \partial_x(h_{0,\hat{k}}\circ q^{-1})(\xi)\cdot(f(\xi)+g(\xi)u) + \alpha_{0,\hat{k}}(h_{0,\hat{k}}(q^{-1}(\xi)))\\ &= \partial_x h_{0,\hat{k}}(q^{-1}(\xi))\cdot\partial_x q^{-1}(\xi)\cdot(f(\xi)+g(\xi)u)\\ &\quad + \alpha_{0,\hat{k}}(h_{0,\hat{k}}(q^{-1}(\xi)))\\ &= \partial_x h_{0,\hat{k}}(q^{-1}(\xi))\cdot\big(f(q^{-1}(\xi)) + g(q^{-1}(\xi))u\big)\\ &\quad + \alpha_{0,\hat{k}}(h_{0,\hat{k}}(q^{-1}(\xi)))\\ &= L_f h_{0,\hat{k}}(q^{-1}(\xi)) + L_g h_{0,\hat{k}}(q^{-1}(\xi))\cdot u\\ &\quad + \alpha_{0,\hat{k}}(h_{0,\hat{k}}(q^{-1}(\xi)))\\ &\geq 0,\end{aligned} \tag{43}$$

where we used h_0, α_0 and their *curried* forms $h_{0,\hat{k}}(\cdot) = h_0(\cdot,\hat{k})$, $\alpha_{0,\hat{k}}(\cdot) = \alpha_0(\cdot,\hat{k})$ interchangeably for the sake of notational convenience. The chain rule is used to obtain the third equality, and the PCBF condition for h_0 at state $x = q^{-1}(\xi)$ was used to get the last inequality. □

In Section VI-A, we demonstrate one possible application of a symmetry-induced PCBF to a wheeled mobile robot, whose dynamics is symmetric under the Lie group action by $SE(2)$. A similar construction can also be done in the case of HOPCBFs. If $h_0(x, \hat{k})$ is a HOPCBF, then also is $h(x, k) = h_0(q^{-1}(x), \hat{k})$.

B. Time-Varying Constraints

In some cases where the safety requirements appear to be time-varying, one can think of time-varying CBF and parameter constraints. Many prior works such as [16], [20] dealt with time-varying CBFs, i.e., h is a function of not only x but also time $t \in [0, \infty)$. The similar formulation also applies to PCBFs, both time-varying PCBFs and time-varying parameter constraints. Let $h(x, k, t)$ now be a function of state x, parameter k, and time t. Also consider time-varying parameter constraints $K(t) = \{k \in \mathbb{R}^{n_k} : \rho_i(k, t) \geq 0, \forall i \in I\}$, and let $\bar{C}(t) = \{(x, k) : k \in K(t), h(x, k, t) \geq 0\}$. One might attempt to call h a *time-varying PCBF* if for all $t \in [0, \infty)$ and for every fixed $k \in K(t)$, there exists an input $u \in U$ such that

$$\partial_t h + L_f h + L_g h \cdot u + \alpha(h, k) \geq 0. \tag{44}$$

However, this definition does not help constraining the augmented state within $C(t)$, since the condition

$$\frac{d}{dt} h(x, k, t) + \alpha(h(x, k, t), k) \geq 0 \tag{45}$$

may now be infeasible. To that end, we introduce an additional assumption that $K(t)$ always expands, i.e., $K(t_1) \subseteq K(t_2)$ if $t_1 \leq t_2$. A sufficient (yet not conservative) condition for expanding $K(t)$ is the existence of a parametrized class $\mathcal{K}$ function β_i for every $i \in I$ such that

$$\partial_t \rho_i(k, t) + \beta_i(\rho_i(k, t), k) \geq 0 \tag{46}$$

for all $k \in K(t)$. These β_i-s can be used in constructing the time-varying version of $\bar{U}_{\mathrm{pcbf}}$ as follows.

$$\bar{U}_{\mathrm{pcbf}}(x, k, t) = \left\{ (u, v) : \begin{array}{l} \partial_t h + L_f h + L_g h \cdot u + \partial_k h \cdot v + \alpha(h, k) \geq 0, \\ \partial_t \rho_i + \partial_k \rho_i \cdot v + \beta_i(\rho_i, k) \geq 0, \quad \forall i \in I \end{array} \right\} \tag{47}$$

For all $t \in [0, \infty)$, this $\bar{U}_{\mathrm{pcbf}}(x, k, t)$ is always nonempty for all $(x, k) \in \bar{C}(t)$.[2]

The similar approach can also be applied to HOPCBFs. If $h(x, k, t - \tau)$ is a time-varying HOCBF (according to [16]) for every fixed $\tau \in [0, \infty)$ and $k \in K(\tau)$, then we call h a time-varying HOPCBF. Under the same condition on $K(t)$, we can construct the time-varying version of $\bar{U}_{\mathrm{hopcbf}}$ as

$$\bar{U}_{\mathrm{hopcbf}}(x, k, t) = \left\{ (u, v) : \begin{array}{l} \dfrac{d\phi_{j-1}}{dt} + \alpha_j(\phi_{j-1}, k) \geq 0, \quad \forall j \in \{1, \cdots, r\} \\ \partial_t \rho_i + \partial_k \rho_i \cdot v + \beta_i(\rho_i, k) \geq 0, \quad \forall i \in I \end{array} \right\}, \tag{48}$$

where the time-varying versions of $\phi_{(\cdot)}$-s are defined as

$$\begin{aligned} \phi_0 &= h \\ \phi_j &= \partial_t \phi_{j-1} + L_f \phi_{j-1} + \alpha_j(\phi_{j-1}, k), \quad \forall j \in \{1, \cdots, r\}. \end{aligned} \tag{49}$$

For the function sequence to be well-defined, h should be r times continuously differentiable with respect to not only (x, k) but also (x, k, t). The first inequality constraint of (48) can be further expanded in a similar manner to (37) and (38) as

$$\begin{aligned} &\frac{d\phi_{j-1}}{dt} + \alpha_j(\phi_{j-1}, k) \\ &\quad = \partial_t \phi_{j-1} + L_f \phi_{j-1} + \partial_k \phi_{j-1} \cdot v + \alpha_j(\phi_{j-1}, k) \\ &\quad = \phi_j + \partial_k \phi_{j-1} \cdot v \end{aligned} \tag{50}$$

for $j \in \{1, \cdots, r-1\}$, and

$$\begin{aligned} &\frac{d\phi_{r-1}}{dt} + \alpha_r(\phi_{r-1}, k) \\ &\quad = \partial_t \phi_{r-1} + L_f \phi_{r-1} + L_g \phi_{r-1} \cdot u + \partial_k \phi_{r-1} \cdot v \\ &\qquad + \alpha_r(\phi_{r-1}, k). \end{aligned} \tag{51}$$

C. PCBF-QP with Infinitely Many Parameter Constraints

It is interesting that the optimization problem (22) remains convex and the same feasibility and invariance properties hold even with infinite number of parameter constraints, i.e., I is not a finite set. This suggests the potential of PCBF-QP to encompass a broader range of applications beyond simple QPs. In Section VI-B, we show a case where infinitely many parameter constraints are employed and the resulting PCBF-QP is formulated into a semidefinite program (SDP) given a mild additional assumption on the class $\mathcal{K}$ function $\beta_{(\cdot)}$. A similar extension can also be made in the case of HOPCBF-QP.

VI. Case Study

In this section, we present three simulation results that well exemplify practical applications of the proposed method comprising PCBF-QP and HOPCBF-QP. In the first scenario, we make use of the continuous symmetry of the dynamics to construct a symmetry-induced PCBF as introduced in Section V-A. The second scenario showcases Section V-C, i.e., how to cope with infinite number of parameter constraints. In the third scenario, we demonstrate HOPCBF-QP using an adaptive cruise control task, where a time-varying parameter constraint (see Section V-B) is applied.

[2]Consider $v = 0$.

A. Collision-Free Mobile Robot Navigation

In this case study, we will construct a PCBF for a wheeled ground rover navigating in obstacle-cluttered space obeying the following simplified bicycle-like dynamics:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = f(x) + g(x)u = \begin{bmatrix} x_3 \cos x_4 \\ x_3 \sin x_4 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & x_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{52}$$

where the components of $x = [x_1, x_2, x_3, x_4]^\top \in \mathbb{R}^4$ denote horizontal and vertical positions, forward velocity, and heading angle of the robot, respectively, which are controlled through acceleration ($u_1 \in \mathbb{R}$) and steering ($u_2 \in \mathbb{R}$) inputs. We assume that the inputs are bounded by a box constraint $u \in U = [-1, 1] \times [-1, 1]$.

The mission for this example is to track the reference input given by the user, while avoiding multiple circular shaped obstacles. The number of obstacles is N, and for each $i \in \{1, \cdots, N\}$, the i-th obstacle is located at position $(z_{i,1}, z_{i,2})$ on the x_1-x_2 plane and has radius $R_i > 0$. The robot is also modeled as a circular shape on the x_1-x_2 plane, having radius $r > 0$.

As the first step, we find that the dynamics (52) is continuously symmetric under the Lie group action of $SE(2)$. This symmetry is very natural in that the dynamics of a ground robot (52) can be written in the same form regardless of the choice of coordinate. Let (p_1, p_2, θ) be the coordinates of $SE(2)$ such that the group operation $\cdot : SE(2) \times SE(2) \to SE(2)$ is realized as

$$\begin{aligned} q_1 \cdot q_2 &= (p_{11}, p_{12}, \theta_1) \cdot (p_{21}, p_{22}, \theta_2) \\ &= (p_{11} + p_{21} \cos\theta_1 - p_{22} \sin\theta_1, \\ &\quad p_{12} + p_{21} \sin\theta_1 + p_{22} \cos\theta_1, \\ &\quad \theta_1 + \theta_2). \end{aligned} \tag{53}$$

Then, $q = (p_1, p_2, \theta) \in SE(2)$ as a bijective map $q : \mathbb{R}^4 \to \mathbb{R}^4$ defined as

$$q\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}\right) = \begin{bmatrix} p_1 + x_1 \cos\theta - x_2 \sin\theta \\ p_2 + x_1 \sin\theta + x_2 \cos\theta \\ x_3 \\ \theta + x_4 \end{bmatrix} \tag{54}$$

is a left Lie group action that satisfies (42).

Secondly, we derive a symmetry-induced PCBF h for the system and extend it using the continuous symmetry of the dynamics. The function h_0 defined as

$$h_0(x, \hat{k}) = b - \sqrt{x_1^2 + 4x_2^2 + \epsilon^2} + \epsilon - \frac{1}{2}x_3^2 - 1 + \cos x_4, \tag{55}$$

where $\epsilon > 0$ is a constant used to make h_0 continuously differentiable, is a PCBF with parameter $\hat{k} = (b, \eta_1, \cdots, \eta_N) \in \mathbb{R}^{N+1}$ with any class $\mathcal{K}$ function α_0, since $\frac{d}{dt}h_0$ can be made zero with

$$u = \frac{1}{\sqrt{x_1^2 + 4x_2^2 + \epsilon^2}} \begin{bmatrix} -x_1 \cos x_4 - 2x_2 \sin x_4 \\ -2x_2 \end{bmatrix} \in U. \tag{56}$$

The parameters $\eta_{(\cdot)}$ are *slack* variables that will be used when constructing collision avoidance constraints. Composing with (54), we get

$$\begin{aligned} h(x, k) &= h_0(q^{-1}(x), \hat{k}) \\ &= b - \sqrt{\delta_1^2(1 + 3s^2) + \delta_2^2(1 + 3c^2) - 6\delta_1\delta_2 cs + \epsilon^2} \\ &\quad + \epsilon - \frac{1}{2}x_3^2 - 1 + \cos(x_4 - \theta), \end{aligned} \tag{57}$$

where δ_1, δ_2, c, and s are shorthands for $x_1 - p_1$, $x_2 - p_2$, $\cos\theta$, and $\sin\theta$, respectively.

Finally, we set up the parameter constraints for collision avoidance. To guarantee collision avoidance over an infinite time horizon, we require the projection of set $C(k)$ onto the x_1-x_2 plane and each obstacle to be placed at least r apart. For every $i \in \{1, \cdots, N\}$, a sufficient yet not very conservative condition for $C(k)$ and the i-th obstacle being at least r away is

$$\begin{aligned} &\rho_i(k) = \\ &\hat{p}_1 \cos\eta_i + 2\hat{p}_2 \sin\eta_i - b - \epsilon - (R_i + r)\sqrt{1 + 3\sin^2\eta_i} \geq 0, \end{aligned} \tag{58}$$

where

$$\begin{aligned} \hat{p}_1 &= (z_{i,1} - p_1)\cos\theta + (z_{i,2} - p_2)\sin\theta, \\ \hat{p}_2 &= -(z_{i,1} - p_1)\sin\theta + (z_{i,2} - p_2)\cos\theta. \end{aligned} \tag{59}$$

The basic idea to arrive at (58) is that η_i that satisfies (58) defines a separating hyperplane on the state space

$$\begin{aligned} &(x_1 - z_{i,1})(\cos\theta\cos\eta_i - 2\sin\theta\sin\eta_i) \\ &+ (x_2 - z_{i,2})(\sin\theta\cos\eta_i + 2\cos\theta\sin\eta_i) \\ &+ R_i\sqrt{1 + 3\sin^2\eta_i} = 0 \end{aligned} \tag{60}$$

between $C(k)$ (buffered by the robot's size r) and the i-th obstacle. We omit the details of the derivation, since it is a tedious series of basic hand-doable calculations.

Simulation experiment was conducted using $\epsilon = 0.01$, $r = 0.3$ and $N = 15$ randomly placed obstacles of random sizes. Note that handcrafting a single CBF that covers this workspace is almost impossible. We used $\alpha(y, k) = 2y$ and $\beta_{(\cdot)}(y, k) = 2y$ for the class $\mathcal{K}$ functions. The reference input u_{ref} is given through manual control by a human operator, who is instructed to give aggressive inputs towards the obstacles, so the overall closed-loop system should rely on the PCBF to avoid any collision. Fig. 1 shows four snapshots taken from the simulation. Regardless of the aggressiveness of the manual reference input, the robot always stays within the set $C(k)$ which is placed collision-free due to the parameter constraints. In Fig. 2, it can be seen that the values of PCBF h and the parameter constraint functions $\rho_{(\cdot)}$ are kept nonnegative throughout the simulation.

B. Constrained Stabilization of a Linear System

In the second example, we consider the stabilization task of the following single-input single-output (SISO) linear time-invariant (LTI) system:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= c^\top x, \end{aligned} \tag{61}$$

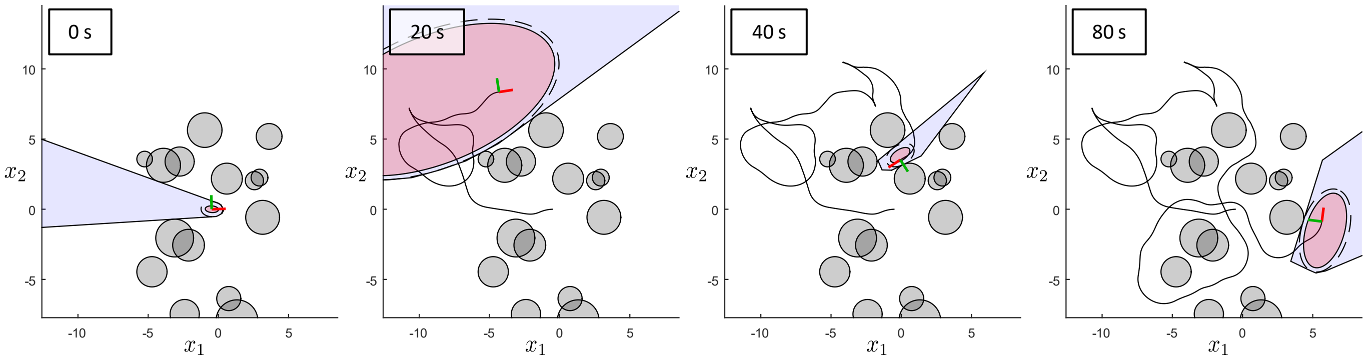


Fig. 1. Four snapshots taken from the simulation experiment on collision-free mobile robot navigation (Section VI-A). The obstacle configuration (position and size) is randomly chosen, and the reference input $\boldsymbol{u}_{\mathbf{ref}}$ is manually given by a human operator who is instructed to transmit aggressive inputs. In each subfigure, the red ellipse denotes $\boldsymbol{C(k)}$, the dotted ellipse is $\boldsymbol{C(k)}$ buffered by the robot's size, black solid line is the robot's trajectory on the $\boldsymbol{x_1}$-$\boldsymbol{x_2}$ plane, gray shaded regions are the obstacles, and the blue polygonal region is the collision-free space defined by the hyperplanes in (60). The boxes on the top left of each snapshot denotes the time the snapshot is taken. The attitude of the robot is depicted as red and green axes. PCBF-QP ensures the robot to stay away from any collision, regardless of the aggressiveness of the input.

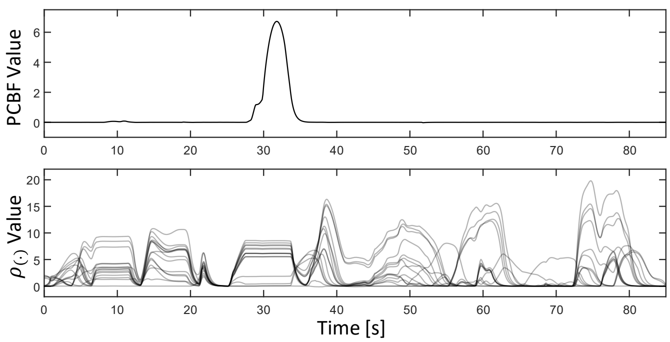


Fig. 2. The values of PCBF $\boldsymbol{h}$ and parameter constraint $\boldsymbol{\rho_{(\cdot)}}$ in the robot navigation example, plotted as a function of time. It can be seen that PCBF-QP is capable of keeping the values nonnegative at all times.

where $x \in \mathbb{R}^n$ is the state, $u, y \in \mathbb{R}$ are the single-dimensional input and output, A, B, c are constant matrices and a column vector of appropriate sizes. With the assumption that the given system is stabilizable, we want to stabilize it to the origin while keeping the output y of the system within a prescribed bound near the origin with bounded input. Since this system is LTI, without loss of generality, we can let the input and output constraints be $-1 \le u \le 1$ and $-1 \le y \le 1$, respectively.

To achieve this control objective, we set up the PCBF candidate function h as

$$h(x, k) = b - \frac{1}{2} x^\top P x, \tag{62}$$

where $k = (b, L, P)$ is the parameter consisting of the Lyapunov function bound $b \in \mathbb{R}$, feedback gain $L \in \mathbb{R}^{1 \times n}$, and the Lyapunov function candidate $P \in \mathbb{S}^n$. Similar to the role of $\eta_{(\cdot)}$ in the previous example, although the second component L does not explicitly appear in (62), it serves as a slack variable when formulating the parameter constraints.

For (62) to be a PCBF, similar to Lyapunov-based CBFs [7], we require $V_P(x) = \frac{1}{2} x^\top P x$ be a Lyapunov function for the closed-loop system $\dot{x} = (A - BL)x$, i.e.,

$$\dot{V}_P(x) = \frac{1}{2} x^\top \left(P(A - BL) + (A - BL)^\top P \right) x \le 0 \tag{63}$$

for all x, and therefore we set the first parameter constraint as

$$\rho_1(k) (\in \mathbb{S}^n) = -P(A - BL) - (A - BL)^\top P \ge 0. \tag{64}$$

Secondly, for the PCBF condition to be satisfied with bounded input, we require the feedback input $u = -Lx$ to meet the specified input bounds, i.e., $-1 \le Lx \le 1$ for all $x \in C(k) = \{x : \frac{1}{2} x^\top P x \le b\}$. This is equivalent to

$$\rho_2(k) (\in \mathbb{S}^n) = P - 2bL^\top L \ge 0, \tag{65}$$

which serves as the second parameter constraint.

Similarly, to satisfy the output constraint ($-1 \le c^\top x \le 1$ for all $x \in C(k)$), we let

$$\rho_3(k) (\in \mathbb{S}^n) = P - 2bcc^\top \ge 0 \tag{66}$$

be the third parameter constraint.

Note that the parameter constraints (64), (65), and (66) are expressed as semidefinite constraints, rather than scalar inequalities. A semidefinite constraint is equivalent to having a spectrum of infinite number of scalar inequality constraints that reads

$$\xi^\top \rho_{(\cdot)}(k) \xi \ge 0, \quad \forall \xi \in \mathbb{R}^p, \tag{67}$$

where p is the appropriate dimension. Thus, the parameter velocity v should satisfy

$$\frac{d}{dt} \left(\xi^\top \rho_i(k) \xi \right) + \beta_i \left(\xi^\top \rho_i(k) \xi \right) \ge 0, \quad \forall \xi \in \mathbb{R}^p \tag{68}$$

for all i. To deal with this, we take linear class $\mathcal{K}$ functions $\beta_{(\cdot)}(z) = \gamma_{(\cdot)} \cdot z$ ($\gamma_{(\cdot)} > 0$, $z \in \mathbb{R}$), and then (68) again takes the semidefinite form:

$$\frac{d}{dt} \rho_i(k) + \gamma_i \cdot \rho_i(k) \ge 0, \tag{69}$$

which yields semidefinite PCBF-QP.

To stabilize the system to the origin, we take a control Lyapunov function

$$V(x) = \frac{1}{2} x^\top S x \quad (S \in \mathbb{S}^n_+) \tag{70}$$

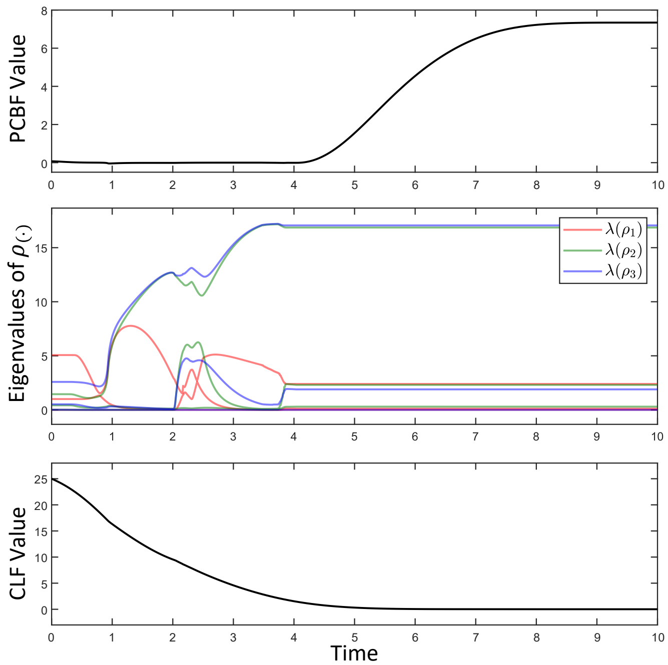


Fig. 3. The values of PCBF $\boldsymbol{h}$, parameter constraints $\boldsymbol{\rho}_{(\cdot)}$, and CLF $\boldsymbol{V}$ in the linear system control scenario (Section VI-B), plotted as a function of time. In the second plot, the eigenvalues of the parameter constraints are drawn. The parameter constraints are satisfied if and only if all the eigenvalues are nonnegative. It can be seen that the CLF value successfully decays towards zero while PCBF and $\boldsymbol{\rho}_{(\cdot)}$ values being kept nonnegative through CLF-PCBF-QP. Note that $\boldsymbol{\rho}_{(\cdot)}$ are guaranteed to have real eigenvalues since they are symmetric matrices.

and then the following CLF-CBF-QP-style [3] PCBF-QP, namely CLF-PCBF-QP, is employed.

$$\begin{aligned} \min_{u\in U, v, \delta} \quad & \|u\|^2 + \mu\|v\|^2 + \nu\delta^2 \\ \text{s.t.} \quad & L_f V(x) + L_g V(x)\cdot u + \alpha_{\text{clf}}(V(x)) \le \delta \\ & L_f h(x,k) + L_g h(x,k)\cdot u + \partial_k h(x,k)\cdot v \\ & \quad + \alpha(h(x,k)) \ge 0 \\ & \frac{d}{dt}\rho_i(k) + \gamma_i \cdot \rho_i(k) \ge 0, \quad \forall i \in \{1,2,3\} \end{aligned} \tag{71}$$

Here, μ and ν are positive weights, α_{clf} is a class $\mathcal{K}$ function, $\delta \in \mathbb{R}$ is an optimization variable that penalizes insufficient decaying speed of the CLF value without affecting the overall feasibility of the optimization.

Here, we consider the triple integrator model with system matrices

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{72}$$

where the components of the three-dimensional state are written as $x = [x_1, x_2, x_3]^\top$. The objective here is to bound its velocity component $x_2 = c^\top x \in [-1, 1]$ ($c = [0, 1, 0]^\top$).

The simulation results using $S = [2, 2, 1; 2, 3, 2; 1, 2, 2]$, $\mu = 0.1$, $\nu = 10$, $\alpha_{\text{clf}}(y) = 5y$, $\gamma_{(\cdot)} = 5$ starting from the initial state $\mathrm{x}(0) = [5, 0, 0]^\top$ are shown in Fig. 3. For the parameter constraints, their eigenvalues are depicted as they are semidefinite constraints. It can be seen that $\rho_{(\cdot)}$ values are kept semidefinite, and $h(x, k)$ nonnegative throughout the simulation. Fig. 4 shows the closed-loop trajectory projected onto the x_1-x_2 plane. We compare with linear quadratic regulator (LQR) controller which gives the same Lyapunov function $\frac{1}{2}x^\top Sx$. It shows how the velocity x_2 is well bounded through the deployment of CLF-PCBF-QP.

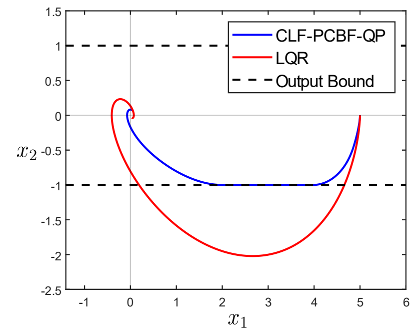


Fig. 4. The trajectory portrait of the linear system control scenario, projected onto the $\boldsymbol{x_1}$-$\boldsymbol{x_2}$ plane. While the LQR-based controller results in violation of the output constraint, the proposed PCBF-based controller (CLF-PCBF-QP) successfully stabilizes the target system while satisfying the requirements.

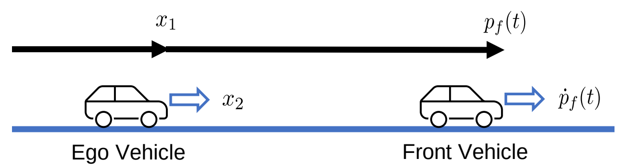


Fig. 5. The adaptive cruise control scenario (Section VI-C). The ego vehicle is required to maintain a safe distance of $\boldsymbol{\delta}$ from the front vehicle. The position of the ego vehicle and the front car are $\boldsymbol{x_1}$ and $\boldsymbol{p_f}$, respectively. Their velocities are $\boldsymbol{x_2}$ and $\dot{\boldsymbol{p}}_{\boldsymbol{f}}$.

C. Adaptive Cruise Control using HOPCBF

Consider the following simplified vehicle dynamics:

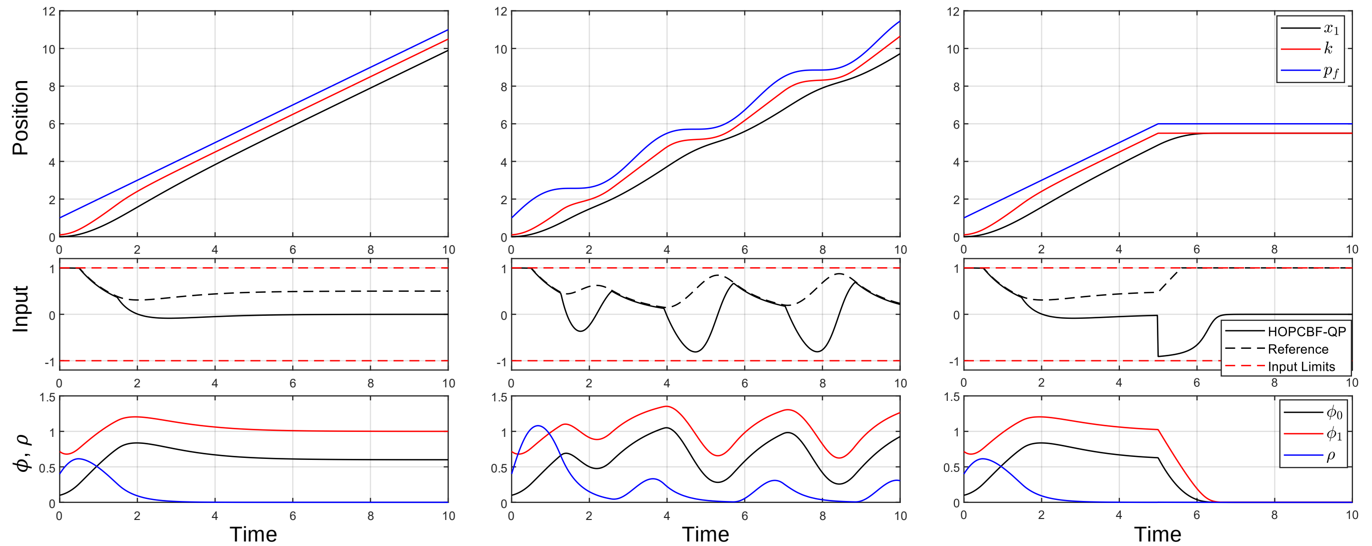
$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f(x) + g(x)u = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{73}$$

where each component of the state $x = [x_1\ x_2]^\top \in \mathbb{R}^2$ denote the position and the velocity of the vehicle, $u \in [u_L, u_U] \subset \mathbb{R}$ denotes the acceleration input. The two parameters $u_L < 0$ and $u_U > 0$ denote the control bounds. In this example, we want to perform collision-free adaptive cruise control for this vehicle using a HOPCBF with relative degree $r = 2$. As shown in Fig. 5, the goal of the ego vehicle is to move forward at a prescribed speed $v_{\text{ref}} > 0$ while maintaining a safe distance of $\delta > 0$ with the preceding vehicle at position $p_f(t)$ and velocity $\dot{p}_f(t)$, where p_f is given as a continuous and piecewise C^1 function and $\dot{p}_f$ a piecewise continuous function of time t. We assume that the front car never moves backwards, i.e., $p_f(t)$ is monotone increasing with respect to t.

The HOPCBF candidate h is as follows:

$$h(x, k) = k - x_1 \tag{74}$$

where $k \in \mathbb{R}$ is the parameter whose physical interpretation is the position on the road before which the vehicle is able to come to a complete stop. With $\alpha_1(z, k) = \sqrt{az + \epsilon^2} - \epsilon$



1) $p_f(t) = 1 + t$ 2) $p_f(t) = 1 + t + 0.5\sin(2t)$ 3) $p_f(t) = \max\{1 + t, 6\}$

Fig. 6. Simulation result for the adaptive cruise control scenario. Regardless of the front vehicle behavior (as long as it does not reverse), HOPCBF-QP is capable of keeping the ego vehicle's position $\boldsymbol{x_1}$ at least $\boldsymbol{\delta = 0.5}$ apart from the front vehicle's position $\boldsymbol{p_f}$. The values of $\boldsymbol{\phi_{(\cdot)}}$ and $\boldsymbol{\rho}$ are simultaneously kept nonnegative through HOPCBF-QP.

and $\alpha_2(z, k) = \gamma z$ ($z \geq 0$, $\epsilon > 0$, $\gamma > 0$), h from (74) is a HOPCBF, given the parameter $a > 0$ satisfies $a \geq -2/u_L$. Thus, the function sequence $\phi_{(\cdot)}$ is given as follows:

$$\begin{aligned} \phi_0(x, k) &= h(x, k) = k - x_1 \\ \phi_1(x, k) &= -x_2 + \sqrt{a(k - x_1) + \epsilon^2} - \epsilon. \end{aligned} \tag{75}$$

To avoid collision with the front vehicle, we introduce one time-varying parameter constraint

$$\rho(k, t) = p_f(t) - k - \delta. \tag{76}$$

It is straightforward to check that $\rho(k, t) \geq 0$ if and only if $p_f(t) - p \geq \delta$ for all $x = (p, v) \in C(k)$, and $\rho(k, t)$ always increases with respect to time and thus satisfies (46) with any class $\mathcal{K}$ function β.

In order for the ego vehicle to move at a speed close to v_{ref}, we make use of HOPCBF-QP with cost $J(x, k, u, v, t) = (\text{sat}(u_{\text{ref}}(x, t)) - u)^2 + \mu v^2$ where the reference input u_{ref} is given as the following feedback law.

$$u_{\text{ref}}(x, t) = L(v_{\text{ref}} - x_2) \tag{77}$$

Here, $\text{sat}(\cdot)$ is the saturation function that clips off the excessive input to fit the bound $u \in [u_L, u_U]$, and μ and L are constant positive reals.

Simulation was conducted using $u_L = -1$, $u_U = 1$, $\delta = 0.5$, $\epsilon = 0.1$, $a = 2$, $\gamma = 2$, $\beta(y, k) = 2y$, $\mu = 0.01$ and $L = 1$. The ego vehicle starts at $x = 0$ and $k = 0.1$, and its reference speed is $v_{\text{ref}} = 1.5$. For the leading vehicle behavior, we consider three different scenarios.

1) $p_f(t) = 1 + t$: The front vehicle moves at a constant speed which is slower than v_{ref}.
2) $p_f(t) = 1 + t + 0.5\sin(2t)$: The front vehicle repeatedly accelerates and decelerates.
3) $p_f(t) = \max\{1 + t, 6\}$: The front vehicle first moves at a constant speed, and then suddenly stops at $t = 5$.

The results for three scenarios can be found in Fig. 6. As shown in the plots, the ego vehicle successfully keeps the safe distance from the preceding vehicle and the input limits simultaneously through HOPCBF-QP.

VII. Conclusion

In this work, we introduced the concept of PCBF, a differentiably parametrized spectrum of CBFs, along with PCBF-QP, a QP-based feedback controller that uses a PCBF. Multiple parameter constraints can be addressed using a PCBF, allowing it to cover a relatively large and complex subset of the workspace using simple building-block function primitives. The concept was extended to cover state constraints of high relative degree, namely HOPCBFs. Multiple simulation experiments were conducted to validate the proposed methodology.

While we successfully used simple building-block CBFs to construct a PCBF by specifying valid ones using parameter constraints, the synthesis of such building blocks remains an area requiring further investigation. Additionally, fusing with stochastic control methods to enable PCBFs to cover uncertain dynamics models is also a possible future work.

References

[1] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.
[3] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
[4] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8352–8358.

[5] 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.

[6] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2897–2904, 2022.

[7] I. Jang and H. J. Kim, "Safe control for navigation in cluttered space using multiple Lyapunov-based control barrier functions," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2056–2063, 2024.

[8] D. D. Oh, D. Lee, and H. J. Kim, "Safety-critical control under multiple state and input constraints and application to fixed-wing UAV," in *62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023.

[9] A. D. Ames, T. G. Molnár, A. W. Singletary, and G. Orosz, "Safety-critical control of active interventions for COVID-19 mitigation," *IEEE Access*, vol. 8, pp. 188 454–188 474, 2020.

[10] S. Feng, R. de Castro, and I. Ebrahimi, "Fast charging of batteries using cascade-control-barrier functions," in *2023 American Control Conference (ACC)*, 2023, pp. 2481–2486.

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

[12] A. Clark, "Control barrier functions for complete and incomplete information stochastic systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2928–2935.

[13] C. Santoyo, M. Dutreix, and S. Coogan, "A barrier function approach to finite-time stochastic system verification and control," *Automatica*, vol. 125, p. 109439, 2021.

[14] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2267–2281, 2021.

[15] 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.

[16] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2021.

[17] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[18] I. Jang and H. J. Kim, "Invariance guarantees using continuously parametrized control barrier functions," in *2023 The 23rd International Conference on Control, Automation and Systems (ICCAS)*. ICROS, 2023, pp. 70–75. [Online]. Available: https://janginkyu.github.io/files/iccas2023-paper.pdf

[19] H. K. Khalil, *Nonlinear systems*. Prentice Hall, 2002.

[20] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6814–6821.

Inkyu Jang (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from Seoul National University, Seoul, Korea, in 2020. Starting from 2020, he is currently working towards a Ph.D. degree in aerospace engineering at Seoul National University, Seoul, Korea.

His research interests include safety-critical control and motion planning for mobile robots.

H. Jin Kim (Member, IEEE) received the B.S. degree from Korea Advanced Institute of Technology, Daejeon, Korea, in 1995, and the M.S. and Ph.D. degrees from University of California, Berkeley, Berkeley, CA, USA, in 1999 and 2001, respectively, all in mechanical engineering.

From 2002 to 2004, she was a postdoctoral researcher with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. In 2004, she joined the Department of Aerospace Engineering, Seoul National University, Seoul, Korea, where she is currently a professor and the head of the department. Her research interests include control and navigation of autonomous robotic systems, robot learning, and computer vision for robotics.