Invariance Guarantees using Continuously Parametrized Control Barrier Functions

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

***Abstract*— Constructing a large-enough control invariant set that fits within a given state constraint is a fundamental problem in safety-critical control but is known to be difficult, especially for large or complex spaces. This paper introduces a safe control framework of utilizing PCBF: continuously parametrized high-order control barrier functions (HOCBFs). In PCBF, each choice of parameter corresponds to a control invariant set of relatively simple shape. Invariance-preserving control is done by dynamically selecting a parameter whose corresponding invariant set lies within the safety bound. This eliminates the need for synthesizing a single complex HOCBF that matches the entire free space. It also enables easier adaptation to diverse environments. By assigning a differentiable dynamics on the parameter space, we derive a lightweight feedback controller based on quadratic programming (QP), namely PCBF-QP. We also discuss on how to build a valid PCBF for a class of systems and how to constrain the parameter so that the invariant set does not exceed the safety bound. Finally, simulation experiments are conducted to validate the proposed approach.**

***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. In order to address not 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 pre-given 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 introduces one additional inequality constraint to the input bounds 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], adapting to changing dynamics [14], time-varying CBFs for satisfaction of signal temporal logic specifications [15]. The recent concept of high-order CBF (HOCBF) [16] generalizes the notion of CBF to cover high relative degree.

The common aim of these methods is to construct a barrier certificate corresponding to a invariant set that fits within the given state constraints. Considering the control performance, it is obviously important to obtain a valid barrier function that provides a sufficiently large control invariant set within the prescribed limit. Unfortunately, synthesizing a valid one is often not straightforward and easily becomes computationally burdensome, particularly with large or complex environments.

In this paper, we introduce an invariance-preserving control framework that uses continuously parametrized spectrum of HOCBFs, which we call parametrized CBF (PCBF). PCBF is a Lyapunov-like scalar function that takes not only the state but also a parameter which lives in a continuous parameter space. For each fixed parameter, PCBF is a valid yet relatively simple HOCBF that defines a small building-block control invariant set. Given a set of *safe* parameters (whose corresponding invariant sets lie within the safety bound), a large control invariant set that spans the entire free space is constructed by taking the union of them.

The proposed PCBF framework decouples the problem of synthesizing invariance guarantees into the following two:

Problem 1. Given the system dynamics, construct a PCBF. This includes designing the parameter space and finding the building-block invariant sets.

Problem 2. Given the safety bound and the PCBF, construct the parameter constraint that describes the safe parameter set.

Addressing Problem 1 does not require considering the potentially complex safety bound; and given the building-block invariant sets, Problem 2 is purely geometric (it is a problem of determining set inclusion) and does not explicitly depend on the system dynamics. In many cases, this significantly reduces the computational burden compared to directly searching for a

*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

single valid barrier certificate that spans the entire free space. Moreover, it gives adaptability to various environments, since only the parameter constraint (and not the PCBF) needs to be resynthesized when shifting to a new safety bound.

Finding a general answer to the two problems (i.e., a systematic way of constructing a PCBF and the parameter constraint from the dynamics model only) is hard and remains an open problem. However, in this paper, we find that continuous symmetry of the system dynamics, which is often found in a wide range of real-world dynamical systems, naturally extends a single HOCBF to give a valid PCBF. We also introduce a design process of the building block invariant set using a stabilizing controller and a Lyapunov function. We also briefly discuss on how the parameter constraint should be constructed.

Once a valid PCBF and the parameter constraints are obtained, we are then interested in the invariance-preserving control problem:

Problem 3. Using the PCBF and the parameter constraint, synthesize a computationally lightweight feedback control that renders the resulting safe set (the union of the building blocks) invariant.

To guarantee invariance, we assign a single integrator dynamics on the parameter space and augment it to the system. The control is done with respect to the augmented system in a manner that the parameter always stays within the safe parameter set, and the state stays within the building-block invariant set corresponding to the current parameter value. Leveraging the continuity of the parameter space, we devise a CBF-QP-like safety filter based on PCBF (PCBF-QP), which is capable of constraining the augmented system as mentioned.

The definition of PCBF in this paper is the high-order generalization of our recent conference paper [17], which briefly introduced PCBF and PCBF-QP for the relative-degree-one case. On that, the discussions on the barrier function synthesis, i.e., Problem 1 and Problem 2, are new in this paper. We present two numerical examples, one demonstrates the barrier function design process abovementioned, the other addresses the high-order case. The remainder of this paper is organized as follows. We start by introducing the necessary concepts and assumptions in Section II. The concept of PCBF and its definition is presented in Section III. Then follows Section IV in which we derive PCBF-QP and answer Problem 3. In Section V, we discuss the design process and give a partial answer to Problem 1 and Problem 2. The numerical examples 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. An inequality between two vectors denotes that it is satisfied in an elementwise manner. 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 symbols $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $k \in \mathbb{R}^{n_k}$, and $t \in [0,\infty)$ 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.

B. Dynamics

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

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

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 so that given any initial value $\mathrm{x}(0)$ and a measurable input trajectory $\mathrm{u}(t)$, there exists a unique solution x at least locally. The input is assumed to be bounded by a set of linear inequalities, i.e.,

$$\mathrm{u}(t) \in U = \{u \in \mathbb{R}^m : A_u u \le b_u\}, \quad \forall t \in [0,\infty) \tag{2}$$

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

C. High-Order Control Barrier Functions

Definition 1 (Relative Degree [18, Definition 13.2]). The output $h : \mathbb{R}^n \to \mathbb{R}$ of the system (1) has relative degree r 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 on $\mathbb{R}^n$.

In this subsection, we provide a brief overview of HOCBFs [16]. Let $h : \mathbb{R}^n \to \mathbb{R}$ be a $\mathcal{C}^r$ output for the system (1) with relative degree r. With this, we want to drive the system in a way that $h(\mathrm{x}(t)) \ge 0$ is satisfied throughout the interval $t \in [0,\infty)$. The vector field f appearing in the dynamics (1) is assumed to be at least $\mathcal{C}^{r-1}$, with its $(r-1)$-th order derivative being Lipschitz.

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

$$\begin{aligned} &\psi_0(x) = h(x) \\ &\psi_j(x) = \dot{\psi}_{j-1}(x) + \alpha_j(\psi_{j-1}(x)),\ \forall j \in \{1, \cdots, r-1\} \end{aligned} \tag{3}$$

where $\alpha_j(\cdot)$, $j \in \{1, \cdots, r-1\}$ are class $\mathcal{K}$ functions[1]. 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: $\dot{\psi}_{j-1}(x) = L_f \psi_{j-1}(x)$. The key idea of HOCBF is that

$$\psi_j(x) = \dot{\psi}_{j-1}(x) + \alpha_j(\psi_{j-1}(x)) \ge 0 \tag{4}$$

gives $\psi_{j-1}(x) \ge 0$ 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 exsts an input $u \in U$ for any x such that

$$\begin{aligned} &\dot{\psi}_{r-1}(x,u) + \alpha_r(\psi_{r-1}(x)) \\ &= L_f \psi_{r-1}(x) + L_g \psi_{r-1}(x) \cdot u + \alpha_r(\psi_{r-1}(x)) \ge 0, \end{aligned} \tag{5}$$

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

then it will initiate a chain of nonnegativity certificates and eventually render the set

$$C = \bigcap_{j=1}^{r} C_j \tag{6}$$

invariant, where for each $j \in \{1, \cdots, r\}$ the set C_j is defined as $C_j = \{x \in \mathbb{R}^n : \psi_{j-1}(x) \geq 0\}$.

Definition 2 (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

$$L_f \psi_{r-1}(x) + L_g \psi_{r-1}(x) \cdot u + \alpha_r(\psi_{r-1}(x)) \geq 0, \tag{7}$$

for all $x \in C$, where the functions $\psi_{(\cdot)}$ are defined recursively as $\psi_0(x) = h(x)$, $\psi_j(x) = L_f \psi_{j-1}(x) + \alpha_j(\psi_{j-1}(x))$ for $j \in \{1, \cdots, r-1\}$, and $C = \{x : \psi_{j-1}(x) \geq 0, \ \forall j \in \{1, \cdots, r\}\}$.

Remark 1. CBF is the special case $r = 1$ of Definition 2.

Now, let $U_{\text{hocbf}}(x) = \{u \in U : (7) \text{ holds.}\}$. It can be easily seen that Definition 2 ensures nonemptyness of $U_{\text{hocbf}}(x)$ for all $x \in C$. Any feedback controller $u(x, t)$ will render the set C invariant if $u(x, t) \in U_{\text{hocbf}}(x)$ for all $x \in C$ and $t \in [0, \infty)$. Leveraging this, one can construct a QP-based safety filter which is a generalization of [3]:

$$\begin{aligned} u(t, x) = \arg \min_{u \in \mathbb{R}^m} \ & \|u - u_{\text{ref}}(t, x)\|_W^2 \\ \text{s.t. } & u \in U_{\text{hocbf}}(x), \end{aligned} \tag{8}$$

where $\|\cdot\|_W$ is a weighted two-norm, which can be solved at a very low computational cost using off-the-shelf convex programming solvers.

III. Parametrized CBF

Given the basic definitions and the necessary assumptions, let us begin the discussion by setting up the problem in more detail. Let $A \subset \mathbb{R}^n$ be the set of allowable states. For example, for a mobile robot collision avoidance task, A contains all robot states not overlapping with the obstacles. While A is not necessarily control invariant, not every state in A is actually safe, and we want a large control invariant set C such that $C \subseteq A$. As mentioned in the introduction, directly searching for a HOCBF (and also the corresponding class $\mathcal{K}$ functions) for this purpose often becomes computationally intractable. This is because it requires solving the variational inequality (7) subject to the inequality condition $C \subseteq A$, i.e.,

$$(\psi_{j-1}(x) \geq 0, \ \ \forall j \in \{1, \cdots, r\}) \implies x \in A. \tag{9}$$

On the other hand, constructing a simple HOCBF without considering the constraint $C \subseteq A$ is relatively simpler in many cases. Moreover, it is often easy to obtain a *continuous spectrum* of HOCBFs, rather than a single one. That is, we can often find a single scalar function $h : \mathbb{R}^n \times K_0 \to \mathbb{R}$ such that for any $k \in K_0 \subseteq \mathbb{R}^{n_k}$, $h(\cdot, k)$ is a valid HOCBF that satisfies (7). We call such h a PCBF.

Definition 3 (PCBF)**.** A function $h : \mathbb{R}^n \times \mathbb{R}^{n_k} \to \mathbb{R}$ is a PCBF 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 $k \in K_0 \subseteq \mathbb{R}^{n_k}$.

Remark 2. PCBF is a generalization of HOCBF: Consider the case where K_0 is a set with only one element. It also generalizes the special case $r = 1$ presented in [17].

Similarly to HOCBFs, we define a sequence of multivariate functions $\phi_j(x, k)$ as follows.

$$\begin{aligned} &\phi_0(x, k) = h(x, k) \\ &\phi_j(x, k) = L_f \phi_{j-1}(x, k) + \alpha_j(\phi_{j-1}(x, k), k), \end{aligned} \tag{10}$$

and the sets

$$\begin{aligned} &C_j(k) = \{x \in \mathbb{R}^n : \phi_{j-1}(x, k) \geq 0\}, \ j \in \{1, \cdots, r\}, \\ &C(k) = \bigcap_{j=1}^{r} C_j(k). \end{aligned} \tag{11}$$

The definitions for $\phi_{(\cdot)}$ are very similar to $\psi_{(\cdot)}$ from (3), and their values are actually identical given stationary (i.e., fixed) k. Still, it is important to make distinctions between the two definitions (we used different notations here), and with time-varying parameter $\mathrm{k}(t)$, the value of $\phi_j(\mathrm{x}(t), \mathrm{k}(t))$ might not be equal to $\frac{d}{dt}\phi_{j-1}(\mathrm{x}(t), \mathrm{k}(t)) + \alpha_j(\phi_{j-1}(\mathrm{x}(t), \mathrm{k}(t)), \mathrm{k}(t))$.

Observe that for every $k \in K_0$, the set $C(k)$ is control invariant since $h(\cdot, k)$ is a valid HOCBF. Further, since the union of control invariant sets is also control invariant [19, Proposition 4.13], for any $K \subseteq K_0$,

$$C = \bigcup_{k \in K} C(k) \subseteq \mathbb{R}^n \tag{12}$$

is also control invariant. While the sets $C(k)$ are not necessarily safe (i.e., $C(k) \in A$), given A, we can properly select a subset K such that C is not only control invariant but also $C \subseteq A$, without the need for directly addressing the set inclusion condition (9).

Notice that finding a valid PCBF does not depend on the given environment A. Additionally, given a PCBF h, constructing a valid parameter constraint K that matches the safety bound A does not require explicit consideration of the dynamics. This enables decoupling of the barrier synthesis problem, as delineated in Problem 1 and Problem 2.

IV. Invariance-Preserving Control using PCBF

In Section II, we have seen that we can derive a computationally lightweight QP-based safety filter which takes a reference input signal and gives a safe input which renders the HOCBF invariant set C (from (6)) invariant, once a valid HOCBF is given. In this section, we derive PCBF-QP, a QP-based computationally light safety filter that renders the PCBF invariant set C (from (12)) invariant.

A. Constructing Input Constraints using Parameter-Augmented Dynamics

Rendering C (the PCBF invariant set) invariant is equivalent to driving the state trajectory $\mathrm{x}(t)$ in a way that there exists at least one choice of parameter $k \in K$ such that $\mathrm{x}(t) \in C(k)$. A simple but naive approach for this is to pick a parameter $k \in K$ which makes $\phi_{j-1}(x,k) \geq 0$ for all $j \in \{1, \cdots, r\}$, along with the control input u that satisfies the HOCBF constraint. That is, similarly to [7], select $\mathrm{u}(t)$ and $\mathrm{k}(t)$ that solve

$$\begin{aligned}\min_{u,k}\ & J(x,k,u,t)\\ \text{s.t. } & L_f\phi_{r-1}(x,k) + L_g\phi_{r-1}(x,k)u + \alpha_r(\phi_{r-1}(x,k),k) \geq 0\\ & \phi_{j-1}(x,k) \geq 0\ \forall j \in \{1,\cdots,r\},\ k \in K,\ u \in U\end{aligned} \tag{13}$$

with $x = \mathrm{x}(t)$, where J is an appropriate cost function. The problem in this form is that (13) is generally a nonlinear and nonconvex problem whose optimization typically requires heavy computation, making it inappropriate for real-time feedback synthesis. Moreover, it is very likely that the solution is discontinuous with respect to $\mathrm{x}(t)$, which may result in a severe chattering phenomenon.

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. Consider the following augmented system

$$\begin{aligned}\dot{\mathrm{x}}(t) &= f(\mathrm{x}(t)) + g(\mathrm{x}(t))\mathrm{u}(t)\\ \dot{\mathrm{k}}(t) &= \mathrm{v}(t),\end{aligned} \tag{14}$$

where $(\mathrm{x}(t), \mathrm{k}(t)) \in \mathbb{R}^n \times \mathbb{R}^{n_k}$ is the augmented state, $(\mathrm{u}(t), \mathrm{v}(t)) \in \bar{U} = U \times \mathbb{R}^{n_k}$ is the augmented input. Now, the parameter k is no longer an optimization variable but the controller's internal state that has to be controlled by the virtual input $v \in \mathbb{R}^{n_k}$.

At this point, we introduce an additional (yet not restrictive) assumption that K can be represented using differentiable inequality constraints, i.e., $K = \{k \in \mathbb{R}^{n_k} : \rho_i(k) \geq 0,\ \forall i \in I\}$, where ρ_i are differentiable functions, I is a finite index set, with the regularity condition $\partial_k\rho_i(k) \neq 0$ if $\rho_i(k) = 0$. With respect to the augmented system (14), consider the disjoint union of $C(k)$, i.e.,

$$\begin{aligned}\bar{C} &= \bigsqcup_{k\in K} C(k)\\ &= \{(x,k) : k \in K,\ \phi_{j-1}(x,k) \geq 0\ \forall j \in \{1,\cdots,r\}\}.\end{aligned} \tag{15}$$

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 $\bar{C}$ under the augmented dynamics directly relates to invariance of C under the original dynamics.

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 single-integrator parameter dynamics. Thus, if $\mathrm{k}(t) \in K$, there must exist a virtual input $v \in \mathbb{R}^{n_k}$ such that

$$\rho_i(\mathrm{k}(t)) = 0 \ \Rightarrow\ \frac{d}{dt}\rho_i(\mathrm{k}(t)) = \partial_k\rho_i(\mathrm{k}(t)) \cdot v \geq 0 \tag{16}$$

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

$$\partial_k\rho_i(k) \cdot v + \beta_i(\rho_i(k)) \geq 0, \tag{17}$$

for all $k \in K$ and $i \in I$. Here, β_i is a class $\mathcal{K}$ function. Similarly, for $\phi_{j-1}(x,k)$ to be kept nonnegative, we require

$$\begin{aligned}&\dot{\phi}_{j-1}(x,k,u,v) + \alpha_j(\phi_{j-1}(x,k),k)\\ &= L_f\phi_{j-1}(x,k) + L_g\phi_{j-1}(x,k)\cdot u + \partial_k\phi_{j-1}(x,k)\cdot v\\ &\quad + \alpha_j(\phi_{j-1}(x,k),k)\\ &= \phi_j(x,k) + \partial_k\phi_{j-1}(x,k)\cdot v \geq 0\end{aligned} \tag{18}$$

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

$$\begin{aligned}&\dot{\phi}_{r-1}(x,k,u,v) + \alpha_r(\phi_{r-1}(x,k))\\ &= L_f\phi_{r-1}(x,k) + L_g\phi_{r-1}(x,k)\cdot u + \partial_k\phi_{r-1}(x,k)\cdot v\\ &\quad + \alpha_r(\phi_{r-1}(x,k),k) \geq 0.\end{aligned} \tag{19}$$

To obtain the last equality of (18), 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 $L_g\phi_{j-1}(x,k)\cdot u$ reduces to zero since the relative degree of $\phi_{j-1}(\cdot,k)$ with respect to the original dynamics (1) is $r-j+1$, which is greater than 1.

Combining (17), (18), and (19), we can consider the following feasible set on the augmented input space:

$$\bar{U}_{\mathrm{pcbf}}(x,k) = \left\{(u,v) \in \bar{U} : \begin{array}{l}(18)\ \forall j \in \{1,\cdots,r-1\},\ (19),\\ \partial_k\rho_i(k)\cdot v + \beta_i(\rho_i(k)) \geq 0,\ \forall i \in I\end{array}\right\}, \tag{20}$$

which is nonempty for any $(x,k) \in \bar{C}$. This is because for any u such that $L_f\phi_{r-1}(x,k) + L_g\phi_{r-1}(x,k)u + \alpha_r(\phi_{r-1}(x,k),k) \geq 0$, it can be easily seen that $(u,0) \in \bar{U}_{\mathrm{pcbf}}(x,k)$. The existence of such u is guaranteed for all $(x,k) \in \bar{C}$ since $h(\cdot,k)$ is a HOCBF.

Remark 3. PCBF can also handle time-varying parameter constraints *relaxing* with respect to time, for example, an exploration task. A sufficient yet not conservative condition for this is the existence of a class $\mathcal{K}$ function β_i for every $i \in I$ such that $\partial_t\rho_i(k,t) + \beta_i(\rho_i(k,t)) \geq 0$. Then, replacing (17) with $\partial_t\rho_i + \partial_k\rho_i\cdot v + \beta_i(\rho_i) \geq 0$ does not break invariance, since $v = 0$ is still a feasible solution.

Remark 4. PCBF does not have relative degree r with respect to the augmented dynamics. That is why (20) has an inequality constraint for every j. Another perspective of viewing this is that with respect to the augmented dynamics, we have constructed $r + |I|$ barrier-like functions, all with relative degree 1, the intersection of whose super zero level sets defines $\bar{C}$, in a way that they are all compatible within $\bar{C}$, i.e., $\bar{U}_{\mathrm{pcbf}}(x,k)$ is nonempty for all $(x,k) \in \bar{C}$. This compatibility is not general for naive composition of multiple (HO)CBFs, as there might not exist a control input that satisfies all the input constraints: The intersection of multiple control invariant sets is in general not invariant.

B. PCBF-based QPs for Invariance Guarantees

Consider the following optimization-based controller, which we call PCBF-based QP (PCBF-QP): Given the augmented dynamical system (14) and PCBF h, solve

$$\begin{aligned}(\mathrm{u}(t),\mathrm{v}(t)) = \arg &\min_{(u,v)\in\bar{U}} J(\mathrm{x}(t),\mathrm{k}(t),u,v,t)\\ &\ \text{s.t. } (u,v) \in \bar{U}_{\mathrm{pcbf}}(\mathrm{x}(t),\mathrm{k}(t)),\end{aligned} \tag{21}$$

where $J(\cdots)$ is a cost function that is strictly convex quadratic with respect to (u, v). This QP always has a unique global minimizer for any $(\mathrm{x}(t), \mathrm{k}(t)) \in \bar{C}$, since $\bar{U}_{\mathrm{pcbf}}(\mathrm{x}(t), \mathrm{k}(t))$ is nonempty (as mentioned above) and J is strictly convex with respect to the decision variables. A decent choice of the cost function J that works fine for many cases is to let J take the CBF-QP-like form:

$$J(x, k, u, v, t) = \|u - u_{\mathrm{ref}}(x, t)\|_W^2 + \mu \|v\|_V^2, \tag{22}$$

where $\|\cdot\|_W$ and $\|\cdot\|_V$ are weighted two-norms, u_{ref} is the reference input signal, $\mu > 0$ is a tunable and usually very small parameter which provides strict convexity and enhances numerical stability of the controller.

V. PCBF and Parameter Set Design

A general framework to construct the PCBF and the parameter constraints is in general not straightforward still remains an open problem. However, in this section, we introduce some design techniques that apply to a class of practical dynamical systems.

A. Symmetry-Induced PCBFs

Definition 4 (Continuous Symmetry of Dynamics)**.** The dynamics (1) is said to have continuous symmetry if there exists a Lie group G acting on the state space $\mathbb{R}^n$ such that for any $q \in G$, if the state and input trajectory pair (x, u) solves the ODE (1), then also does the pair $(q \circ \mathrm{x}, \mathrm{u})$. Here, $q \in G$ as a function ($q : \mathbb{R}^n \to \mathbb{R}^n$) denotes the Lie group action.

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. As delineated in Definition 4, continuous symmetries can be mathematically understood using Lie group actions. Continuous symmetry provides a simple yet powerful way of constructing PCBFs, which we call symmetry-induced PCBFs.

Theorem 1 (Symmetry-Induced PCBF)**.** *Let $\hat{h} : \mathbb{R}^n \times \hat{K}_0 \to \mathbb{R}$ be a PCBF (with relative degree r, $\hat{K}_0 \subseteq \mathbb{R}^{\hat{n}_k}$) for a system with continuous symmetry with the corresponding Lie group of symmetry being G. Then, $h(x, k) = \hat{h}(q^{-1}(x), \hat{k})$ is a PCBF with the new parameter $k = (q, \hat{k}) \in K_0 = G \times \hat{K}_0$.*

Proof. We will prove the equivalent statement: If $h : \mathbb{R}^n \to \mathbb{R}$ is a HOCBF, then $h' = h \circ q^{-1}$ also is.

Let $(\mathrm{x}, \mathrm{u}) : [0, T) \to \mathbb{R}^n \times U$ be a dynamically feasible state-input trajectory pair where T is a positive time horizon, i.e., x is the unique solution to the ODE $\dot{\mathrm{x}}(t) = f(\mathrm{x}(t)) + g(\mathrm{x}(t))\mathrm{u}(t)$ given initial condition $\mathrm{x}(0)$. Given the continuous symmetry and the same u, if $\mathrm{z} : [0, T)$ solves the initial value problem $\dot{\mathrm{z}}(t) = f(\mathrm{z}(t)) + g(\mathrm{z}(t))\mathrm{u}(t)$ and $\mathrm{z}(0) = q(\mathrm{x}(0))$, then for all $t \in [0, T)$, $\mathrm{z}(t) = q(\mathrm{x}(t))$.

Thus, for any testing function $\beta : \mathbb{R}^n \to \mathbb{R}$, $\beta(\mathrm{x}(t)) = (\beta \circ q^{-1} \circ q \circ \mathrm{x})(t) = (\beta \circ q^{-1})(\mathrm{z}(t))$. This means that under the same input signal $\mathrm{u}(t)$, $\left(\frac{d}{dt}\right)^j \beta(\mathrm{x}(t)) = \left(\frac{d}{dt}\right)^j (\beta \circ q^{-1})(\mathrm{z}(t))$ up to any order $j \in \{0, 1, \cdots\}$ as long as they exist. Letting β be the $\psi_{(\cdot)}$ functions concludes the proof. □

Symmetry-induced PCBFs are also powerful in terms of parameter constraint construction, since the shape of the invariant sets $C(k)$ remain unchanged under the orbit of the Lie group action.

Theorem 2. *Suppose h is a symmetry-induced PCBF with the Lie group of symmetry and the parameter space being G and $G \times \hat{K}_0$, respectively. Let $C_e(\hat{k}) = C((1_G, \hat{k}))$, where 1_G is the identity element of G. Then, $C(k) = q(C_e(\hat{k})) = \{q(x) : x \in C_e(\hat{k})\}$ ($k = (q, \hat{k})$).*

Proof. The result follows directly from the definition of symmetry-induced PCBF $h(x, k) = \hat{h}(q^{-1}(x), \hat{k})$. □

Theorem 2 tells that $C(k)$-s are *transformed copies* of $C_e(\hat{k})$, which reduces the search space for constructing the parameter constraint. For example, if A is expressed as

$$A = \{x \in \mathbb{R}^n : l_i(x) \geq 0, \ \forall i \in I\}, \tag{23}$$

then any ρ_i such that

$$\rho_i(k) \leq \min_{x \in C_e(\hat{k})} (l_i \circ q)(x) \tag{24}$$

would make $C \subseteq A$. Notice that the search space on the right hand side is reduced to $C_e(\hat{k})$.

B. PCBF Construction using Stabilizing Control

Let $x_0 \in \mathbb{R}^n$ be a point in state space to which the system is stabilizable. That is, there exists a Lyapunov function $V(\cdot)$ such that $V(x) \geq 0$ for all x, $V(x_0) = 0$ if and only if $x = x_0$, and

$$\min_{u \in U} L_f V(x) + L_g V(x) u \leq 0. \tag{25}$$

Then, for any $b \geq 0$, $b - V(x)$ is a CBF (i.e., a HOCBF with $r = 1$) [7], and therefore $h(x, b) = b - V(x)$ is a PCBF with parameter $b \in K_0 = \{b \in \mathbb{R} : b \geq 0\}$. In many cases, finding a valid Lyapunov function is easier than searching directly for a (HO)CBF, not only because Lyapunov functions are sometimes handcraftable, but also because there are readily a handful of existing nonlinear control methodologies designed specifically for stabilization of nonlinear systems. These include but not are limited to backstepping control, sliding mode control, neural Lyapunov control [20], all coming with valid Lyapunov functions. With appropriate setting, Hamilton-Jacobi reachability [21], [22] method can also be used to find a Lyapunov function.

Combined with the symmetry-based technique of the previous subsection, this Lyapunov-based method serves as an especially powerful tool in tasks such as collision avoidance of mobile robots. If the system exhibits a continuous symmetry, the system being stabilizable to $x_0 \in \mathbb{R}^n$ automatically implies it is also stabilizable to $q(x_0)$, for any q from the group of symmetry G, and as such, $h(x, k) = b - V(q^{-1}(x))$ is a valid PCBF with the parameter k being $k = (b \geq 0, q \in G)$. In other words, the PCBF framework allows synthesizing a safety filter for complex environments using only a single stabilizing controller.

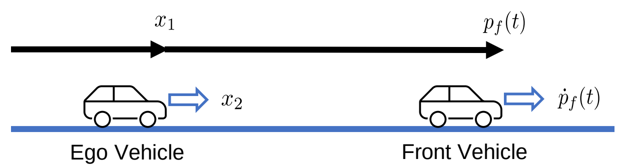

Fig. 1. The inter-vehicle distance maintenance scenario (Section VI-A). The ego vehicle is required to maintain a safe distance of $\boldsymbol{\delta}$ from the front vehicle. The position of the ego vehicle and the front car are $\boldsymbol{x_1}$ and $\boldsymbol{p_f}$, respectively. Their velocities are $\boldsymbol{x_2}$ and $\boldsymbol{\dot{p}_f}$.

C. Parameter Augmentation using Auxiliary Variables

Another advantage offered by the PCBF framework is that it allows augmenting the parameter space using *auxiliary* variables. That is, given a PCBF h, the user can build a new PCBF $\bar{h}(x, \bar{k}) = h(x, k)$ with the augmented parameter $\bar{k} = (k, \eta) \in \bar{K} = K_0 \times E$, where $\eta \in E \subseteq \mathbb{R}^\eta$ is the auxiliary variable. One can adopt many constraint engineering techniques from mathematical optimization [23] to mitigate the burden of building the parameter constraints. For example, if the shape of A is complex to handle, one can define a simple-shaped (e.g., polygonal or ellipsoidal) set-valued function $D(\eta) \subseteq \mathbb{R}^n$, and specify the parameter constraints to ensure $D(\eta) \subseteq A$ and $C(k) \subseteq D(\eta)$.

VI. Case Study

In this section, we present two simulation results that well exemplify practical applications of the proposed framework. In the first scenario, we demonstrate PCBF-QP with $r = 2$ using a simplified vehicle dynamics model, where a time-varying parameter constraint (see Remark 3) is employed. The second scenario addresses mobile robot collision avoidance problem, where the design techniques introduced in Section V are used.

A. Inter-Vehicle Distance Maintenance

Consider the following simplified vehicle dynamics:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f(x) + g(x)u = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad (26)$$

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. As shown in Fig. 1, 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 $\dot{p}_f(t)$. We assume that the front car never moves backwards, i.e., $\dot{p}_f(t)$ is nonnegative.

The PCBF h is designed as follows:

$$h(x, k) = k - x_1 \quad (27)$$

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 $\alpha_1(z, k) = \sqrt{az + \epsilon^2} - \epsilon$ and $\alpha_2(z, k) = \gamma z$ ($z \geq 0$, $\epsilon > 0$, $\gamma > 0$), this h satisfies the condition Definition 3 if $a \geq -2/u_L > 0$. With them, $\phi_{(\cdot)}$ are given as follows:

$$\begin{aligned} \phi_0(x, k) &= h(x, k) = k - x_1 \\ \phi_1(x, k) &= -x_2 + \sqrt{a(k - x_1) + \epsilon^2} - \epsilon. \end{aligned} \quad (28)$$

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

$$\rho(k, t) = p_f(t) - k - \delta. \quad (29)$$

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 the conditions in Remark 3 holds with any class $\mathcal{K}$ function β.

In order to encourage the ego vehicle to move at a speed close to v_{ref}, we make use of PCBF-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 a simple linear speed feedback $u_{\text{ref}}(x, t) = L(v_{\text{ref}} - x_2)$. 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 $u_L = -1$, $u_U = 1$, $\delta = 0.5$, $\epsilon = 0.1$, $a = 2$, $\gamma = 2$, $\beta(y) = 2y$, $\mu = 0.01$ 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. 2. As shown in the plots, the ego vehicle successfully keeps the safe distance from the preceding vehicle and the input limits simultaneously through PCBF-QP.

B. Collision-Free Mobile Robot Navigation

In this example, following the design techniques explained in Section V, we will construct a PCBF for a wheeled ground rover navigating in obstacle-cluttered space obeying the following simplified bicycle-like dynamics:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = f(x) + g(x)u = \begin{bmatrix} x_3 \cos x_4 \\ x_3 \sin x_4 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & x_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (30)$$

where the components of $x = [x_1, x_2, x_3, x_4]^\top \in \mathbb{R}^4$ denote the 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 $(z_{i,1}, z_{i,2})$

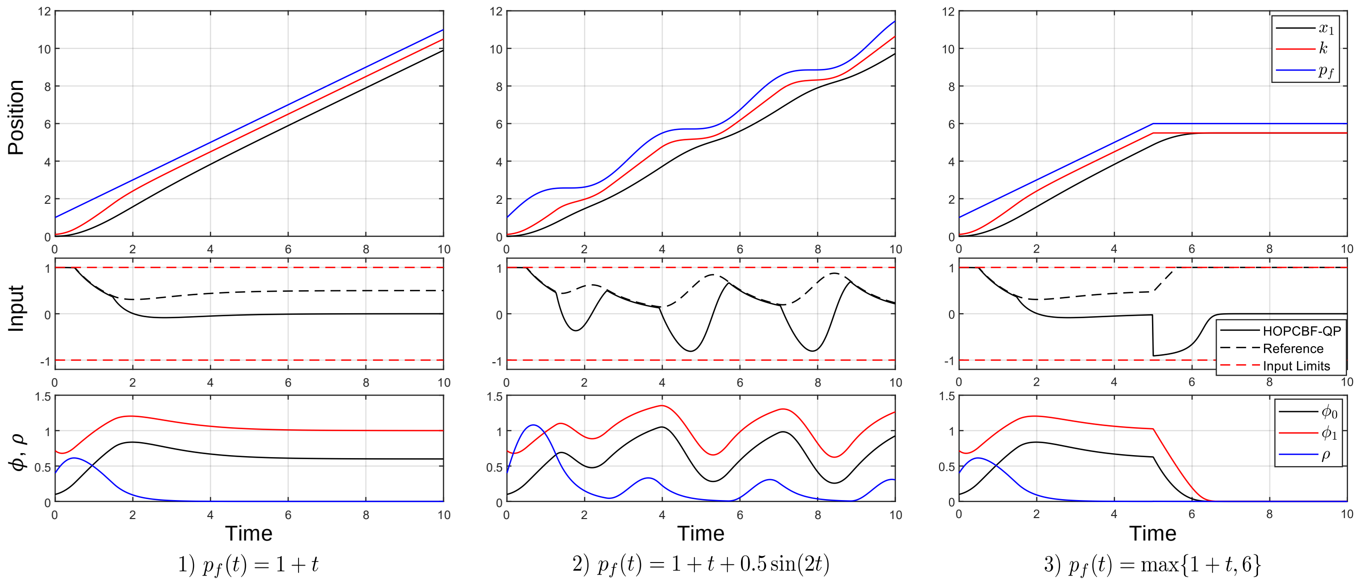

Fig. 2. Simulation result for Section VI-A. Regardless of the front vehicle behavior (as long as it does not reverse), PCBF-QP is capable of keeping the ego vehicle's position $\boldsymbol{x_1}$ at least $\boldsymbol{\delta = 0.5}$ apart from the front vehicle's position $\boldsymbol{p_f}$. The values of $\boldsymbol{\phi_{(\cdot)}}$ and $\boldsymbol{\rho}$ are simultaneously kept nonnegative through PCBF-QP.

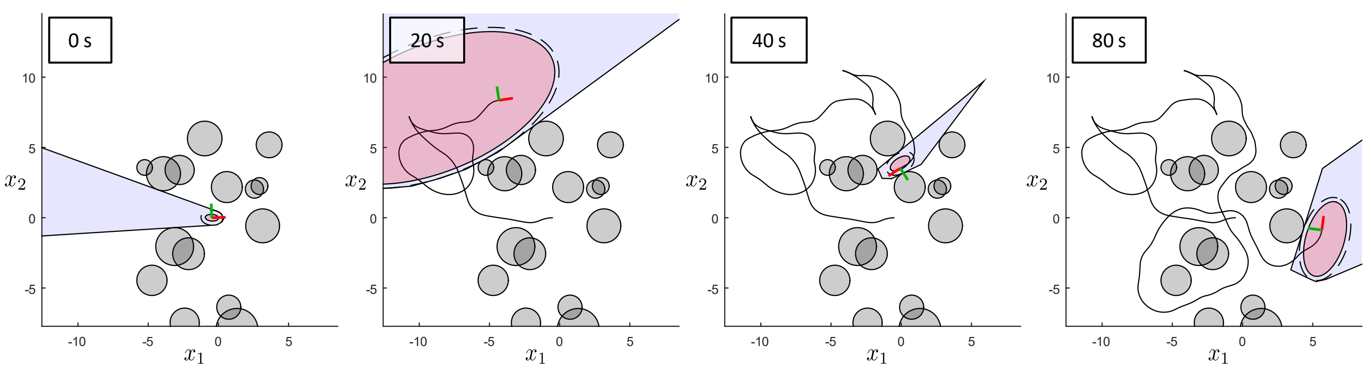

Fig. 3. Four snapshots taken from the simulation experiment on collision-free mobile robot navigation (Section VI-B). The obstacle configuration (position and size) is randomly chosen, and the reference input $\boldsymbol{u_{ref}}$ is manually given by a human operator who is instructed to transmit aggressive 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 separating hyperplanes described by the auxiliary variable $\boldsymbol{\eta}$. 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.

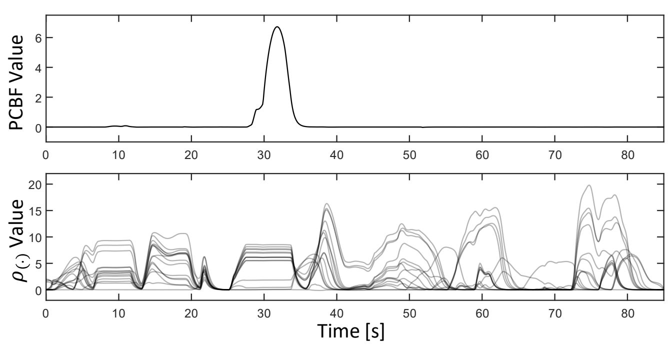

Fig. 4. The values of PCBF $\boldsymbol{h}$ and parameter constraint $\boldsymbol{\rho_{(\cdot)}}$ in the robot navigation example, plotted as a function of time. It can be seen that PCBF-QP is capable of keeping the values nonnegative at all times.

on the x_1-x_2 plane and has radius $R_i > 0$. The robot is modeled as a circle on the x_1-x_2 plane, having radius $R > 0$.

As the first step, we find that the dynamics (30) is continuously symmetric under the Lie group action of $SE(2)$. This symmetry is very natural in that the dynamics of a ground robot (30) can be written in the same form regardless of the choice of coordinate. For a $q \in SE(2)$, the Lie group action $q(x)$ is defined as *accordingly translating and rotating* the pose-related elements (x_1, x_2, x_4) with the velocity x_3 being unchanged. We also find that the robot is stabilizable to the origin by utilizing the following handcrated control Lyapunov function.

$$V(x) = \sqrt{x_1^2 + 4x_2^2 + \epsilon^2} - \epsilon + \frac{1}{2}x_3^2 + 1 - \cos x_4 \qquad (31)$$

This V is a valid Lyapunov function for any $\epsilon > 0$ because

$$u = \frac{1}{\sqrt{x_1^2 + 4x_2^2 + \epsilon^2}} \begin{bmatrix} -x_1 \cos x_4 - 2x_2 \sin x_4 \\ 0 \end{bmatrix} \in U \quad (32)$$

is a feasible input that makes the $V(x)$ value nonincreasing. With the two observations given, we follow the steps delineated in Section V to obtain a symmetry-induced PCBF

$$h(x, k) = b - V(q^{-1}(x)), \; k = (b, q) \in K_0 = [0, \infty) \times SE(2). \quad (33)$$

Notice that the x_1-x_2 projection of the b-level set of $V(x)$, i.e., $C(k)$ with q being the identity element, is always an ellipse with the two semiaxis diameters being $2\sqrt{b^2 + 2b\epsilon}$ and $\sqrt{b^2 + 2b\epsilon}$ and the major semiaxis pointing to the positive x_1 direction. Utilizing this and Theorem 2, and following Section V-C, we augment the parameter space using an N-dimensional auxiliary variable $\eta \in \mathbb{R}^N$. This defines N hyperplanes (i.e., lines) on the x_1-x_2 space, resulting in polygonal $D(\eta) \subseteq \mathbb{R}^2$. For each element of η, we employ a parameter constraint which constraints each hyperplane to strictly pass between each obstacle and the ellipse (buffered by the robot's size R), as shown in Fig. 3. This ensures $C(k)$ and the obstacles do not overlap, and thus $C \subseteq A$. We omit the details of the derivation due to limited space and 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. 3 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. 4, it can be seen that the values of PCBF h and the parameter constraint functions $\rho_{(\cdot)}$ are kept nonnegative throughout the simulation.

VII. Conclusion

In this work, we introduced the concept of PCBF, a differentiably parametrized spectrum of HOCBFs, 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 control invariant sets. We also introduced some design techniques that can be used for a class of systems to design a valid PCBF and the parameter constraints for invariance guarantees within a given safe region. Simulation experiments were conducted to validate the proposed methodology.

While the proposed design technique was successful in the shown simulation environments, systematic synthesis of these certificates for general nonlinear systems remains an open area requiring further investigation. In addition, fusing with stochastic control methods to enable PCBFs to cover uncertain or stochastic dynamics models is another 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] 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

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

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

[20] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.

[21] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[22] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1817–1824.

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