High-Order Control Barrier Functions

Wei Xiao, *Member, IEEE*, and Calin Belta, *Fellow, IEEE*

***Abstract*—We approach the problem of stabilizing a dynamical system while optimizing a cost and satisfying safety constraints and control limitations. For (nonlinear) affine control systems and quadratic costs, it has been shown that control barrier functions (CBFs) guaranteeing safety and control Lyapunov functions (CLFs) enforcing convergence can be used to (conservatively) reduce the optimal control problem to a sequence of quadratic programs (QPs). Existing works in this category have two main limitations. First, with one exception, they are based on the assumption that the relative degree of the system with respect to a function enforcing a safety constraint is one. Second, the QPs can easily become infeasible, in particular for problems with many safety constraints and tight control limitations. We propose high-order CBFs (HOCBFs), which can accommodate systems of arbitrary relative degrees. For each safety constraint, by using Lyapunov-like conditions, we construct a set of controls that renders the intersection of a set of sets forward invariant, which implies the satisfaction of the original constraint. We formulate optimal control problems with constraints given by HOCBF and CLF, and propose two methods—the penalty method and the parameterization method—to address the feasibility problem. Finally, we show how our methodology can be extended for safe navigation in unknown environments with long-term feasibility. We illustrate the proposed framework on adaptive cruise control and robot control problems.**

***Index Terms*—Lyapunov methods, safety-critical control.**

I. Introduction

The problem of driving a dynamical system to a desired configuration while minimizing its control effort and satisfying safety constraints and control limitations received a lot of attention in recent years [2]–[5]. Recent works propose the use of control barrier functions (CBFs) [2] to enforce safety and control Lyapunov functions (CLFs) [6]–[8] to ensure convergence to desired states.

Barrier functions (BFs) are Lyapunov-like functions [9], whose use can be traced back to optimization problems [10]. More recently, they have been employed in verification and control, e.g., to prove set invariance [11]–[14] and for multiobjective control [15]. CBFs are extensions of BFs for control systems. There are many versions of CBF in the literature. The CBF defined in [2] is allowed to decrease when far away from the boundary of the set. Simpler versions of CBFs, which can approach zero inside the corresponding sets, were proposed in [4] and [5]. Time-varying CBFs were defined and used to enforce the satisfaction of signal temporal logic (STL) formulas in [5].

Most of the works using the CBF-CLF approach are based on the assumption that the (nonlinear) control system is affine in controls and the cost is quadratic in controls. The time domain is discretized, and the state is assumed to be constant in each time interval (at its value at the beginning of the interval). The original optimal control problem (OCP) is reduced to a (possibly large) number of quadratic programs (QPs)—one for each interval [16]. This approach is related to and arguably computationally more efficient than traditional model predictive control (MPC) [17].

While this approach provides a good compromise between the computational effort necessary to compute a solution and its optimality [2], it has two main limitations. First, it is based on the assumption that the relative degree of the system with respect to the function enforcing the safety constraints is one. A backstepping approach was introduced in [18], and it was shown to work for relative degree two. A CBF method for position-based constraints with relative degree two was also proposed in [19]. A more general form, which works for arbitrarily high relative degree constraints, was proposed in [3] and [20]. The method in [3] employs input–output linearization and finds a pole placement controller with negative poles to stabilize the BF to zero. The resulting BF is exponential. The authors in [21] proposed an approach to define another function that is with relative degree one from the original high relative degree constraint. This approach does not include all the states in the definition of a CBF, which may decrease the problem feasibility.

Second, the QPs mentioned earlier could be infeasible, i.e., it is hard to find *valid* CBFs that do not conflict with the control bounds, in particular for problems with many safety constraints. For the ACC problem defined in [2], the minimum braking distance is used to simplify the process of finding a valid CBF, and these results in an additional complex constraint. However, this conflict is hard to address for high-dimensional systems. The approach in [20] tried to address conflict between CBF constraints using control-sharing BFs, without considering the control bounds.

In this article, we define a novel notion of high-order CBF (HOCBF), which is simpler and more general than the one from Nguyen and Sreenath [3]. Our HOCBFs are not restricted to exponential functions, and are determined by a set of class $\mathcal{K}$ functions for high relative degree constraints. As a generalization of the main result from Ames *et al.* [2], a safety set is guaranteed to be forward invariant if the HOCBF constraint is satisfied. In order to find a valid HOCBF, we exploit the definitions of the class $\mathcal{K}$ functions, and develop a methodology, called the *penalty method*, to guarantee the feasibility of the QPs. We also propose a *parameterization method* to deal with the feasibility problem when the penalty method fails. We provide a framework to control a system to safely navigate in an unknown environment while ensuring long-term feasibility.

We illustrate the proposed method on adaptive cruise control (ACC) and robot control problems. The simulations show the effectiveness of the proposed HOCBF method with feasibility guarantee. For the robot problem, we illustrate the feasibility of the solution for unknown environments cluttered with obstacles of similar shape but different sizes.

This article is a significant extension of our recent conference paper [1]. Specifically, in addition to including the technical details related to the general definition of an HOCBF, here we introduce the penalty and the parameterization methods. The penalty method simply

Manuscript received December 15, 2020; revised May 13, 2021; accepted August 9, 2021. Date of publication August 18, 2021; date of current version June 30, 2022. This work was supported in part by National Science Foundation (NSF) under Grant IIS-1723995 and Grant IIS-2024606. *(Corresponding author: Wei Xiao.)*

The authors are with the Division of Systems Engineering, Boston University, Brookline, MA 02446 USA (e-mail: xiaowei@bu.edu; cbelta@bu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2021.3105491.

Digital Object Identifier 10.1109/TAC.2021.3105491

adds weights (penalties) for the class $\mathcal{K}$ functions in the definition of the HOCBF from [1]. We provide conditions that guarantee problem feasibility for this method. When these conditions are not satisfied, we apply the alternative parameterization method, in which the class $\mathcal{K}$ functions are power functions multiplied by penalties. We determine penalties and powers in the HOCBF such that the QPs are always feasible and that minimize the HOCBF value for which the corresponding constraint becomes active. This can improve the problem feasibility in an unknown environment, and we illustrate it in robot navigation.

II. Preliminaries

Definition 1 (Class $\mathcal{K}$ function [22]): A Lipschitz continuous function $\alpha : [0, a) \to [0, \infty), a > 0$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.

Lemma 1 (Lemma 4.4 in [22] and Lemma 2 in [4]): Let $b : [t_0, t_f] \to \mathbb{R}$ be a continuously differentiable function. If $\dot{b}(t) \geq -\alpha(b(t)) \quad \forall t \in [t_0, t_f]$, where α is a class $\mathcal{K}$ function of its argument, and $b(t_0) \geq 0$, then $b(t) \geq 0 \quad \forall t \in [t_0, t_f]$.

Consider a system of the form

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

with $\boldsymbol{x} \in X \in \mathbb{R}^n$ (X denotes a closed-state constraint set) and $f : \mathbb{R}^n \to \mathbb{R}^n$ globally Lipschitz. Solutions $\boldsymbol{x}(t)$ of (1), starting at $\boldsymbol{x}(t_0)$, $t \geq t_0$, are forward complete.

We also consider affine control systems in the form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \tag{2}$$

where $g : \mathbb{R}^n \to \mathbb{R}^{n \times q}$ is globally Lipschitz, and $\boldsymbol{u} \in U \subset \mathbb{R}^q$ (U denotes a closed-control constraint set). Solutions $\boldsymbol{x}(t)$ of (2), starting at $\boldsymbol{x}(t_0)$, $t \geq t_0$, are forward complete.

Definition 2 (Forward invariant set): A set $C \subset \mathbb{R}^n$ is forward invariant for system (1) [or (2)] if its solutions starting at any $\boldsymbol{x}(t_0) \in C$ satisfy $\boldsymbol{x}(t) \in C$ for $\forall t \geq t_0$.

For a continuously differentiable function $b : \mathbb{R}^n \to \mathbb{R}$, let

$$C := \{\boldsymbol{x} \in \mathbb{R}^n : b(\boldsymbol{x}) \geq 0\}. \tag{3}$$

Definition 3 (BF [2], [4], [5]): The function $b : \mathbb{R}^n \to \mathbb{R}$ is a candidate BF for system (1) if there exists a class $\mathcal{K}$ function α such that

$$\dot{b}(\boldsymbol{x}) + \alpha(b(\boldsymbol{x})) \geq 0 \quad \forall \boldsymbol{x} \in C. \tag{4}$$

Theorem 1: (see[4] and [5]) Given a set C as in (3), if there exist a BF $b : C \to \mathbb{R}$, then C is forward invariant for system (1).

Definition 4 (CBF [2], [4], [5]): Given a set C as in (3), $b(\boldsymbol{x})$ is a candidate CBF for system (2) if there exists a class $\mathcal{K}$ function α s.t.

$$\sup_{\boldsymbol{u} \in U} [L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})\boldsymbol{u} + \alpha(b(\boldsymbol{x}))] \geq 0 \quad \forall \boldsymbol{x} \in C \tag{5}$$

where L_f and L_g denote the Lie derivatives[1] along f and g, respectively.

We call the CBF in Definition 4 a *candidate* because $\alpha(\cdot)$ is not fixed. In this article, we show how to define a class $\mathcal{K}$ function $\alpha(\cdot)$ such that there exists a control $\boldsymbol{u} \in U$ that satisfies (5). A CBF is completely defined when $\alpha(\cdot)$ is specified. This applies to Definition 3 as well.

Theorem 2 (see[4] and [5]): Given a CBF b with the associated set C from (3), any Lipschitz continuous controller $\boldsymbol{u}(t) \quad \forall t \geq t_0$ that satisfies (5) renders the set C forward invariant for (2).

Definition 5 (CLF [8]): A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is a globally and exponentially stabilizing CLF for system (2) if there exist constants $c_1 > 0, c_2 > 0, c_3 > 0$, and $c_1||\boldsymbol{x}||^2 \leq V(\boldsymbol{x}) \leq c_2||\boldsymbol{x}||^2$ such that, for $\forall \boldsymbol{x} \in \mathbb{R}^n$

$$\inf_{\boldsymbol{u} \in U} [L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x})] \leq 0. \tag{6}$$

Definition 6 (Relative degree [22]): The relative degree of a (sufficiently) differentiable function $b : \mathbb{R}^n \to \mathbb{R}$ with respect to system (2) is the number of times we need to differentiate it along the dynamics of (2) until the control $\boldsymbol{u}$ explicitly shows.

In this article, since function b is used to define a constraint $b(\boldsymbol{x}) \geq 0$, we will also refer to the relative degree of b as the relative degree of the constraint.

Many existing works [2], [3], [5] combine CBF and CLF with quadratic costs to form optimization problems. The CLF constraint is always slacked (i.e., a slack variable is added to relax the constraint, and minimized by adding it to the cost) when combined with CBF to make the problem feasible; however, state convergence may not be guaranteed. Time is discretized, and an optimization problem with constraints given by CBF and CLF is solved at each time step. The resulting problem is a sequence of QPs. The control from solving the QP is held constant and is applied at the current time step. The dynamics (2) is updated, and the procedure is repeated. It is important to note that this method works conditioned upon the fact that the control input shows up in (5), i.e., $L_g b(\boldsymbol{x}) \neq 0, \exists \boldsymbol{x} \in X$ and the QPs are all feasible.

III. High-Order CBFs

In this section, we define high-order BFs (HOBFs) and HOCBFs.

Example: ACC: Consider the ACC problem [2] with vehicle dynamics

$$\dot{v}(t) = u(t), \ \dot{z}(t) = v_0 - v(t) \tag{7}$$

where $v(t)$ denotes the velocity of the ego vehicle along its lane, $z(t)$ denotes the distance between the ego and the preceding vehicles, $v_0 > 0$ denotes the speed of the preceding vehicle, and $u(t)$ is the control input of the ego vehicle.

We require that the distance $z(t)$ between the ego vehicle and its immediately preceding vehicle be greater than a constant $\delta > 0$ for all the times, i.e.,

$$z(t) \geq \delta \quad \forall t \geq t_0. \tag{8}$$

Let $\boldsymbol{x}(t) := (v(t), z(t))$ and $b(\boldsymbol{x}(t)) = z(t) - \delta$. With $\alpha(\cdot)$ in Definition 4 chosen as the identity function, according to (5), in order to ensure safety, we need to have

$$\underbrace{v_0 - v(t)}_{L_f b(\boldsymbol{x}(t))} + \underbrace{0}_{L_g b(\boldsymbol{x}(t))} \times u(t) + \underbrace{z(t) - \delta}_{b(\boldsymbol{x}(t))} \geq 0. \tag{9}$$

Note that $L_g b(\boldsymbol{x}(t)) = 0$ in (9), i.e., $u(t)$ does not show up. Therefore, we cannot use this CBF to formulate an optimization problem, as described at the end of Section II.

A. High-Order BF

As in [5], we consider a time-varying function to define an invariant set for system (1). For a mth-order differentiable function $b : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}$, we define a sequence of functions $\psi_i : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}, i \in \{1, \ldots, m\}$ in the form

$$\psi_i(\boldsymbol{x}, t) = \dot{\psi}_{i-1}(\boldsymbol{x}, t) + \alpha_i(\psi_{i-1}(\boldsymbol{x}, t)), i \in \{1, \ldots, m\} \tag{10}$$

where $\alpha_i(\cdot), i \in \{1, \ldots, m\}$ denote class $\mathcal{K}$ functions of their argument and $\psi_0(\boldsymbol{x}, t) = b(\boldsymbol{x}, t)$.

[1]The Lie derivative of a function along a vector field captures the change in the value of the function along the vector field (see, e.g., [22]).

We further define a sequence of sets $C_i(t), i \in \{1,\ldots,m\}$ associated with (10) in the form

$$C_i(t) = \{\boldsymbol{x} \in \mathbb{R}^n : \psi_{i-1}(\boldsymbol{x},t) \geq 0\}, i \in \{1,\ldots,m\}. \quad (11)$$

Definition 7: Let $C_i(t), i \in \{1,\ldots,m\}$ be defined by (11) and $\psi_i(\boldsymbol{x},t), i \in \{1,\ldots,m\}$ be defined by (10). A function $b : \mathbb{R}^n \times [t_0,\infty) \to \mathbb{R}$ is a candidate HOBF for (1) if it is mth-order differentiable and there exist differentiable class $\mathcal{K}$ functions $\alpha_i, i\in\{1,\ldots m\}$ s.t.

$$\psi_m(\boldsymbol{x},t) \geq 0 \quad (12)$$

for all $(\boldsymbol{x},t) \in C_1(t)\cap,\ldots,\cap C_m(t) \times [t_0,\infty)$.

Similar to Definition 4, an HOBF is defined when $\alpha_i(\cdot), i \in \{1,\ldots,m\}$ are found.

Theorem 3: The set $C_1(t)\cap,\ldots,\cap C_m(t)$ is forward invariant for system (1) if $b(\boldsymbol{x},t)$ is an HOBF.

Proof: If $b(\boldsymbol{x}(t),t)$ is an HOBF, then $\psi_m(\boldsymbol{x}(t),t) \geq 0$ for $\forall t \in [t_0,\infty)$, i.e., $\dot{\psi}_{m-1}(\boldsymbol{x}(t),t) + \alpha_m(\psi_{m-1}(\boldsymbol{x}(t),t)) \geq 0$. By Lemma 1, since $\boldsymbol{x}(t_0) \in C_m(t_0)$ (i.e., $\psi_{m-1}(\boldsymbol{x}(t_0),t_0)) \geq 0$, and $\psi_{m-1}(\boldsymbol{x}(t),t)$ is an explicit form of $\psi_{m-1}(t)$), then $\psi_{m-1}(\boldsymbol{x}(t),t)) \geq 0 \ \ \forall t \in [t_0,\infty)$, i.e., $\dot{\psi}_{m-2}(\boldsymbol{x}(t),t) + \alpha_{m-1}(\psi_{m-2}(\boldsymbol{x}(t),t)) \geq 0$. Again, by Lemma 1, since $\boldsymbol{x}(t_0) \in C_{m-1}(t_0)$, we also have $\psi_{m-2}(\boldsymbol{x}(t),t)) \geq 0 \ \ \forall t \in [t_0,\infty)$. Iteratively, we can get $\boldsymbol{x}(t) \in C_i(t) \ \ \forall i \in \{1,\ldots,m\} \ \ \forall t \in [t_0,\infty)$. Therefore, the set $C_1(t)\cap,\ldots,\cap C_m(t)$ is forward invariant. ■

Remark 1: The sets $C_i(t), i \in \{1,\ldots,m\}$ should have a nonempty intersection at t_0 in order to satisfy the forward invariance condition starting from t_0 in Theorem 3. If $b(\boldsymbol{x}(t_0),t_0) \geq 0$, we can always choose proper class $\mathcal{K}$ functions $\alpha_i(\cdot), i \in \{1,\ldots,m\}$ to make $\psi_i(\boldsymbol{x}(t_0),t_0) \geq 0 \ \ \forall i \in \{1,\ldots,m-1\}$. There are some extreme cases, however, when this is not possible. For example, if $\psi_0(\boldsymbol{x}(t_0),t_0) = 0$ and $\dot{\psi}_0(\boldsymbol{x}(t_0),t_0) < 0$, then $\psi_1(\boldsymbol{x}(t_0),t_0)$ is always negative no matter how we choose $\alpha_1(\cdot)$. Similarly, if $\psi_0(\boldsymbol{x}(t_0),t_0) = 0$, $\dot{\psi}_0(\boldsymbol{x}(t_0),t_0) = 0$, and $\dot{\psi}_1(\boldsymbol{x}(t_0),t_0) < 0$, $\psi_2(\boldsymbol{x}(t_0),t_0)$ is also always negative, etc. To deal with such extreme cases (as with the case when $b(\boldsymbol{x}(t_0),t_0) < 0$), we would need a feasibility enforcement method, which is beyond the scope of this article.

B. High-Order CBF

Definition 8: Let $C_i(t), i \in \{1,\ldots,m\}$ be defined by (11) and $\psi_i(\boldsymbol{x},t), i \in \{1,\ldots,m\}$ be defined by (10). A function $b : \mathbb{R}^n \times [t_0,\infty) \to \mathbb{R}$ is a candidate HOCBF of relative degree m for system (2) if there exist differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1,\ldots,m\}$ s.t.

$$\sup_{\boldsymbol{u}\in U}[\mathcal{L}_f^m b(\boldsymbol{x},t) + \mathcal{L}_g\mathcal{L}_f^{m-1}b(\boldsymbol{x},t)\boldsymbol{u} + \frac{\partial^m b(\boldsymbol{x},t)}{\partial t^m}$$
$$+O(b(\boldsymbol{x},t)) + \alpha_m(\psi_{m-1}(\boldsymbol{x},t))] \geq 0 \quad (13)$$

for all $(\boldsymbol{x},t) \in C_1(t)\cap,\ldots,\cap C_m(t) \times [t_0,\infty)$. $\mathcal{L}_f$ and $\mathcal{L}_g$ denote the partial Lie derivatives w.r.t. $\boldsymbol{x}$ along f and g, respectively.

Similar to Definition 4, an HOCBF is defined when $\alpha_i(\cdot), i \in \{1,\ldots,m\}$ are specified. In this article, we show how to find such functions. In the aforementioned equation, $O(\cdot)$ is given by

$$O(b(\boldsymbol{x},t)) = \sum_{i=1}^{m-1}\mathcal{L}_f^i(\alpha_{m-i}\circ\psi_{m-i-1})(\boldsymbol{x},t)$$
$$+ \frac{\partial^i(\alpha_{m-i}\circ\psi_{m-i-1})(\boldsymbol{x},t)}{\partial t^i}.$$

Given an HOCBF b, we define the set of control that satisfies

$$K_{\text{hocbf}}(\boldsymbol{x},t) = \{\boldsymbol{u} \in U : \mathcal{L}_f^m b(\boldsymbol{x},t) + \mathcal{L}_g\mathcal{L}_f^{m-1}b(\boldsymbol{x},t)\boldsymbol{u}$$
$$+ \frac{\partial^m b(\boldsymbol{x},t)}{\partial t^m} + O(b(\boldsymbol{x},t)) + \alpha_m(\psi_{m-1}(\boldsymbol{x},t)) \geq 0\}. \quad (14)$$

Theorem 4: Given an HOCBF $b(\boldsymbol{x},t)$ from Definition 8 with the associated sets $C_i(t), i \in \{1,\ldots,m\}$ defined by (11), if $\boldsymbol{x}(t_0) \in C_1(t_0)\cap,\ldots,\cap C_m(t_0)$, then any Lipschitz continuous controller $\boldsymbol{u}(t) \in K_{\text{hocbf}}(\boldsymbol{x}(t),t) \ \ \forall t \geq t_0$ renders the set $C_1(t)\cap,\ldots,\cap C_m(t)$ forward invariant for system (2).

Proof: Since $\boldsymbol{u}(t)$ is Lipschitz continuous and $\boldsymbol{u}(t)$ only shows up in the last equation of (10) when we take Lie derivative on (10), we have that $\psi_m(\boldsymbol{x},t)$ is also Lipschitz continuous. The system states in (2) are all continuously differentiable, so $\psi_i(\boldsymbol{x},t), i \in \{1,\ldots,m\}$ are also continuously differentiable. Therefore, the HOCBF has the same property as the HOBF in Definition 7, and the proof is the same as Theorem 3. ■

Remark 2: The general, time-varying HOCBF introduced in Definition 8, can be used for general, time-varying constraints (e.g., STL specifications [5]) and systems. However, many problems, such as the ACC and robot control problems that we consider in this article, have time-invariant system dynamics and constraints. Therefore, in the rest of this article, we focus on time-invariant versions for simplicity.

Remark 3 (Relationship between time-invariant HOCBF and exponential CBF in [3]): In Definition 7, if we set class $\mathcal{K}$ functions $\alpha_1, \alpha_2 \ldots \alpha_m$ to be linear functions with positive coefficients, then we can get exactly the same formulation as in [3] that is obtained through input–output linearization, i.e.,

$$\psi_i(\boldsymbol{x}) = \dot{\psi}_{i-1}(\boldsymbol{x}) + k_i\psi_{i-1}(\boldsymbol{x}), i \in \{1,\ldots,m\} \quad (15)$$

where $k_i > 0, i \in \{1,\ldots,m\}$. Therefore, the time-invariant version HOCBF defined in this article is a generalization of the exponential CBF introduced in [3].

Remark 4 (Comparison between HOCBF and MPC): In an MPC approach [17], the optimization is defined over a receding horizon. Compared to the myopic HOCBF method considered in this article (i.e., the optimization is over one step), the MPC optimization is more likely to be feasible. However, it is more difficult, as it is, in general a nonlinear program. On the other hand, the HOCBF approach can handle nonlinear (affine) dynamics, and the corresponding optimization problems are easy to solve. The myopia of an HOCBF is significantly improved if a valid HOCBF is found offline, as shown in Section IV-A. This offline computation is hard to be performed in an MPC approach.

Example revisited: For the ACC problem introduced at the beginning of Section III, the relative degree of the constraint from (8) is 2. Therefore, we need an HOCBF with $m = 2$. We choose quadratic class $\mathcal{K}$ functions for both $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$, i.e., $\alpha_1(b(\boldsymbol{x}(t))) = b^2(\boldsymbol{x}(t))$ and $\alpha_2(\psi_1(\boldsymbol{x}(t))) = \psi_1^2(\boldsymbol{x}(t))$. In order for $b(\boldsymbol{x}(t)) := z(t) - \delta$ to be an HOCBF for (7), a control input $u(t)$ should satisfy

$$L_f^2 b(\boldsymbol{x}(t)) + L_g L_f b(\boldsymbol{x}(t))u(t) + 2b(\boldsymbol{x}(t))L_f b(\boldsymbol{x}(t))$$
$$+(L_f b(\boldsymbol{x}(t)))^2 + 2b^2(\boldsymbol{x}(t))L_f b(\boldsymbol{x}(t)) + b^4(\boldsymbol{x}(t)) \geq 0. \quad (16)$$

Note that $L_g L_f b(\boldsymbol{x}(t)) \neq 0$ in (16) and the initial conditions are $b(\boldsymbol{x}(t_0)) \geq 0$ and $\dot{b}(\boldsymbol{x}(t_0)) + b^2(\boldsymbol{x}(t_0)) \geq 0$.

IV. Optimal Control With Time-Invariant HOCBF

In this section, we show how to find a valid HOCBF. Consider an OCP for system (2) with the cost defined as

$$J(\boldsymbol{u}(t)) = \int_{t_0}^{t_f} \mathcal{C}(||\boldsymbol{u}(t)||)dt \tag{17}$$

where $||\cdot||$ denotes the two-norm of a vector. t_0 and t_f denote the initial and final times, respectively, and $\mathcal{C}(\cdot)$ is a strictly increasing function of its argument (such as the energy consumption function $\mathcal{C}(||\boldsymbol{u}(t)||) = ||\boldsymbol{u}(t)||^2$). Assume a time-invariant (safety) constraint $b(\boldsymbol{x}) \geq 0$ with relative degree m has to be satisfied by system (2). Then, the control input $\boldsymbol{u}$ should satisfy the time-invariant HOCBF version of the constraint from (13)

$$L_f^m b(\boldsymbol{x}) + L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} + O(b(\boldsymbol{x})) + \alpha_m(\psi_{m-1}(\boldsymbol{x})) \geq 0 \tag{18}$$

for all $\boldsymbol{x} \in C_1 \cap, \ldots, \cap C_m$ ($C_i, i \in \{1,\ldots,m\}$ denotes the time-invariant version of $C_i(t)$), where $O(b(\boldsymbol{x})) = \sum_{i=1}^{m-1} L_f^i(\alpha_{m-i} \circ \psi_{m-i-1})(\boldsymbol{x})$.

If convergence to a given state is required in addition to optimality and safety, then, as in [2], HOCBF can be combined with CLF. Suppose the control bound U for (2) is defined as

$$U = \{\boldsymbol{u} \in \mathbb{R}^q : \boldsymbol{u}_{\min} \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_{\max} \quad \forall t \in [t_0, t_f]\} \tag{19}$$

where $\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max} \in \mathbb{R}^q$. In order to solve this optimization problem, we use the QP-based approach (suppose $\mathcal{C}(||\boldsymbol{u}(t)||) = ||\boldsymbol{u}(t)||^2$) introduced at the end of Section II, i.e., we partition the time interval $[t_0, t_f]$ into a set of equal time intervals $\{[t_0, t_0+\Delta t), [t_0+\Delta t, t_0+2\Delta t), \ldots\}$, where $\Delta t > 0$. In each interval $[t_0+\omega\Delta t, t_0+(\omega+1)\Delta t)$ ($\omega = 0,1,2,\ldots$), we keep the state constant at its value at the beginning of the interval and also assume the control is constant, and reformulate the optimization problem as a sequence of QPs. Specifically, at $t = t_0 + \omega\Delta t$ ($\omega = 0,1,2,\ldots$), we solve

$$(\boldsymbol{u}^*(t), \delta^*(t)) = \arg\min_{\boldsymbol{u}(t),\delta(t)} ||\boldsymbol{u}(t)||^2 + p\delta^2(t)$$
$$\text{s.t.}(18), (19) \text{ and}$$
$$L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x}) \leq \delta \tag{20}$$

where δ is a slack variable used to relax (soften) the CLF constraint and $p > 0$ is a weight. After solving (20), we integrate (2) with control $\boldsymbol{u}^*(t)$ kept constant during $[t_0+\omega\Delta t, t_0+(\omega+1)\Delta t)$. This QP-based method is suboptimal compared with the original OCP (17), as the optimizations are performed pointwise.

Constraint (18) may conflict with (19), in which case we cannot find a valid HOCBF. If this happens, the OCP becomes infeasible. In the rest of this section, we propose a two-stage methodology to find a valid HOCBF, which is based on offline computations of solutions to (20). We first apply the *penalty method* (see Section IV-A,) assuming the class $\mathcal{K}$ functions are given. If this fails, i.e., the conditions under which the penalty method works are not satisfied, we turn to the *parameterization method* (see Section IV-B). We find a valid HOCBF based on the worst-case initial state for some symmetric unsafe sets (such as circular unsafe sets). For such sets, the problem feasibility does not heavily depend on the initial state and the worst-case initial condition is also easy to find. For example, for a spherical obstacle, the worst-case initial state corresponds to maximum velocity directed at the center of the sphere. With some conservatism, other geometries can also be dealt with by covering them with symmetric sets—in our recent work [23], we used disks. Note that a valid HOCBF might be hard to find for nonconvex unsafe sets [24], in which case the proposed approximation method in [23] can still work.

A. Penalty Method

In (10), we multiply the class $\mathcal{K}$ function $\alpha_i(\cdot)$ with penalties (weights) $p_i \geq 0, i \in \{1,\ldots,m\}$ in the form

$$\psi_i(\boldsymbol{x}) = \dot{\psi}_{i-1}(\boldsymbol{x}) + p_i\alpha_i(\psi_{i-1}(\boldsymbol{x})), i \in \{1,\ldots,m\}. \tag{21}$$

The sets X and U are closed. Let

$$U_{\min} := \inf_{\boldsymbol{x}\in X, \boldsymbol{u}\in U} [-L_g L_f^{m-1} b(\boldsymbol{x}(t))\boldsymbol{u}]$$

$$U_{\max} := \sup_{\boldsymbol{x}\in X, \boldsymbol{u}\in U} [-L_g L_f^{m-1} b(\boldsymbol{x}(t))\boldsymbol{u}]$$

$$F_{\min} := \inf_{\boldsymbol{x}\in X} [L_f^m b(\boldsymbol{x})].$$

The following theorem provides conditions for the feasibility guarantee of the QP (20).

Theorem 5: If $U_{\max} \leq F_{\min}$, then there exist (small enough) $p_i \geq 0, i \in \{1,\ldots,m\}$ such that the control limitations (19) do not conflict with the HOCBF constraint (18) $\forall \boldsymbol{x}(t_0) \in C_1 \cap \cdots \cap C_m$.

Proof: It follows from the sequence of equations in (21) that $p_1, p_2, \ldots, p_{m-1}$ will appear in all terms of $O(b(\boldsymbol{x}))$ in (18), i.e., $O(b(\boldsymbol{x})) = 0$ if $p_i = 0 \quad \forall i \in 1,2,\ldots,m-1$. Since p_m shows up in the last equation of (21), we have that

$$-L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} \leq L_f^m b(\boldsymbol{x})$$

if $p_i = 0 \quad \forall i \in 1,2,\ldots,m$. Since $L_f^m b(\boldsymbol{x}) \geq F_{\min}$, if $-L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} \leq F_{\min}$, then the last constraint is satisfied.

The control bound on $\boldsymbol{u}$ in (19) always satisfies

$$U_{\min} \leq -L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} \leq U_{\max}.$$

Since $F_{\min} \geq U_{\max}$, the intersection of the sets determined by the last two inequalities is always nonempty, i.e., the intersection of the control bounds (19) and the HOCBF constraint (18) is always nonempty. We conclude that there exist small enough penalties $p_1 \geq 0, p_2 \geq 0, \ldots, p_m \geq 0$ such that the control limitations (19) will not conflict with the HOCBF constraint (18). ■

The following corollary provides simpler conditions for systems [such as (7)] that satisfy extra properties.

Corollary 1: If $\mathbf{0} \in U$ and $L_f^m b(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x} \in X$, then there exist (small enough) $p_i \geq 0, i \in \{1,\ldots,m\}$ such that the control limitations (19) do not conflict with the HOCBF constraint (18) $\forall \boldsymbol{x}(t_0) \in C_1 \cap \cdots \cap C_m$.

Proof: Similar to the proof of the last theorem, we have

$$-L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} \leq L_f^m b(\boldsymbol{x})$$

if $p_i = 0 \quad \forall i \in 1,2,\ldots,m$. Since $L_f^m b(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x} \in X$, if $-L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} \leq 0$, then the last constraint is satisfied. The $\mathbf{0}$ vector is included in the last equation, and $\mathbf{0} \in U$. Therefore, there exist small enough penalties $p_1 \geq 0, p_2 \geq 0, \ldots, p_m \geq 0$ such that the control limitations (19) do not conflict with the HOCBF constraint (18). ■

Example revisited: For the ACC problem introduced in Section III, $L_f^2 b(\boldsymbol{x}) = 0$. If 0 is included in the control bound, then from Corollary 1, it follows that HOCBF constraints do not conflict with the control bound when we choose small enough penalties p_1, p_2 for $\alpha_1(\cdot), \alpha_2(\cdot)$.

Remark 5 (Applying the penalty method): Given the class $\mathcal{K}$ functions $\alpha_i(\cdot), i \in \{1, \ldots, m\}$ in an HOCBF $b(\boldsymbol{x})$, if the QP (20) becomes infeasible for some $\boldsymbol{x} \in C_1 \cap \cdots \cap C_m$, or it becomes infeasible at some time $t \in [t_0, t_f]$, we restart from time t_0 and add penalties to the class $\mathcal{K}$ functions as in (21). Note that, when penalties are applied, the sets $C_i, i = 2, \ldots, m$ will be affected. By random selection, we try to find values for the penalties such that $\boldsymbol{x}(t_0) \in C_1 \cap \cdots \cap C_m$. If the optimization problem becomes feasible, then we are done. Otherwise, we decrease the value of p_1, as p_1 shows up in all the $\psi_i(\cdot)$ functions in (21), and thus decreasing p_1 is the most efficient way among all the penalties to make the problem feasible. However, decreasing p_1 can significantly shrink C_2, which might result in $\boldsymbol{x}(t_0) \notin C_2$, as shown in (11). In order to avoid this, we can proceed with decreasing p_2, and recursively try to find penalties such that $\boldsymbol{x}(t_0) \in C_1 \cap \cdots \cap C_m$. If we are not successful, then we will turn to the parameterization method described next.

B. Parameterization Method

When the conditions in Theorem 5 or Corollary 1 are not satisfied or no penalties can be found because of $\boldsymbol{x}(t_0)$ as in Remark 5, we use the parameterization method, in which we also determine the class $\mathcal{K}$ functions. Since power functions are mostly used as class $\mathcal{K}$ functions, we can explicitly write (21) as

$$\psi_i(\boldsymbol{x}) = \dot{\psi}_{i-1}(\boldsymbol{x}) + p_i \psi_{i-1}^{q_i}(\boldsymbol{x}), i \in \{1, \ldots, m\} \tag{22}$$

where $q_i \geq 1 \quad \forall i \in \{1, \ldots, m\}$. The penalties $p_i, i \in \{1, \ldots, m\}$ and powers $q_i, i \in \{1, \ldots, m\}$ are parameters of the HOCBF, and they determine at what time the HOCBF constraint (18) becomes active (i.e., it is satisfied as an equality). If the HOCBF constraint becomes active when system (2) is close to the obstacle (i.e., $b(\boldsymbol{x})$ is close to 0), system (2) may require a large control input such that the safety constraint $b(\boldsymbol{x}) \geq 0$ could be enforced by the HOCBF, which could possibly conflict with the control bound U. Thus, the OCP can become infeasible. Ideally, we would like to choose the parameters such that (20) is feasible in $[t_0, t_f]$. We may just randomly sample the penalties and powers such that the problem (20) is feasible [assume there exists a feasible solution for the problem and there exist such parameters that can make the problem feasible; otherwise, we need to consider all possible class $\mathcal{K}$ functions instead of just power functions in (22)]. However, we do not want the HOCBF constraint (18) to be active when system (2) is far from the corresponding obstacle (i.e., $b(\boldsymbol{x})$ is large) as the obstacles may not be detected before the HOCBF constraint (18) becomes active in an unknown environment. If this happens, the initial conditions of an HOCBF in Theorem 4 may not be satisfied or the HOCBF constraint may conflict with the control bound U, and thus, the safety is not guaranteed. Therefore, we want to choose the parameters such that the value of the HOCBF $b(\boldsymbol{x})$ when the constraint first becomes active is minimized.

Remark 6 (Applying the parameterization method): We can use a gradient-descent method to find the optimal penalties and powers such that the problem is feasible while also minimizing $b(\boldsymbol{x}(t_a))$, where $t_a \in [t_0, t_f]$ is the time at which the HOCBF constraint first becomes active under the worst-initial condition (such as maximum approaching speed). We randomly sample for p_i, q_i, and start with any $p_i, q_i, i \in \{1, \ldots, m\}$ such that the QPs are all feasible in $[t_0, t_f]$ and take $b(\boldsymbol{x}(t_a))$ as the objective function to minimize with $p_i, q_i, i \in \{1, \ldots, m\}$ as decision variables. We evaluate the gradient of $b(\boldsymbol{x}(t_a))$ with respect to p_i, q_i, then minus p_i, q_i by the gradient times a learning rate $\gamma > 0$ and solve the QP (20) again $\forall t \in [t_0, t_f]$ to find a smaller possible $b(\boldsymbol{x}(t_a))$. We repeat this process for each feasible sample (see Algorithm 1).

Algorithm 1: The Parameterization Method.

Input: An OCP as in (17) with safety requirement $b(\boldsymbol{x}) \geq 0$, a terminal condition $N > 0$

Output: $\boldsymbol{p}^*, \boldsymbol{q}^*, \mathcal{D}_{min}$

1. Sample $\boldsymbol{p}, \boldsymbol{q}$ in $(0, c)$ and $(1, d), c > 0, d \geq 1$, resp.;
2. Solve (20) for each sample $\forall t \in [t_0, t_f]$;
3. Pick a feasible $\boldsymbol{p}_0, \boldsymbol{q}_0$, $\mathcal{D}_{min} = b(\boldsymbol{x}(t_a))$, iter. $= 1$;

while *iter.*++ $\leq N$ **do**

  Evaluate $\frac{\partial b(\boldsymbol{x}(t_a))}{\partial p_i}, \frac{\partial b(\boldsymbol{x}(t_a))}{\partial q_i}, i \in \{1 \ldots m\}$ at $\boldsymbol{p}_0, \boldsymbol{q}_0$;
  Set $\frac{\partial b(\boldsymbol{x}(t_a))}{\partial p_i} = 0, \frac{\partial b(\boldsymbol{x}(t_a))}{\partial q_i} = 0$ if they cannot be evaluated due to the infeasibility of (20);
  $(\boldsymbol{p}, \boldsymbol{q}) \leftarrow (\boldsymbol{p}, \boldsymbol{q}) - \gamma \frac{\partial b(\boldsymbol{x}(t_a))}{\partial (\boldsymbol{p}, \boldsymbol{q})}, \gamma > 0$;
  Solve (20), $\forall t \in [t_0, t_f]$, with $\boldsymbol{p}, \boldsymbol{q}$;
  If (20) is feasible $\forall t \in [t_0, t_f]$ and $\mathcal{D}_{min} \geq b(\boldsymbol{x}(t_a))$, then $\mathcal{D}_{min} = \mathcal{D}_j(\boldsymbol{p}, \boldsymbol{q})$, $\boldsymbol{p}_0 = \boldsymbol{p}, \boldsymbol{q}_0 = \boldsymbol{q}$; Otherwise, break;

end

Repeat above for all feasible samples, and set $\boldsymbol{p}^* = \boldsymbol{p}_0, \boldsymbol{q}^* = \boldsymbol{q}_0$ corresponding to the minimum $\mathcal{D}_{min}$;

Remark 7 (Comparison between the penalty and parameterization methods): The penalty method is simpler and more intuitive than the parameterization method as we can just choose small penalties to make the problem feasible (see Remark 5). The parameterization method does not necessarily choose small p_i's, but searches through the whole parameter space, and thus is less conservative and more computationally expensive. The parameterization method works well for unknown environments as we make the HOCBF constraints active as late as possible to allow the system to detect obstacles.

Remark 8 (The effect of time discretization): In order to implement the HOCBF method in real systems, we need to discretize the time, as described at the end of Section II. The forward invariance of the safety sets might not be guaranteed in between the discretization time instants. The work in [21] addresses this issued by trying to find closed-form solutions to the problem. The sampling approach from Cortez *et al.* [25] is used to ensure the constraint satisfaction in the time intervals between the discretization instants. The self-triggered method from Yang *et al.* [26] can also be used to determine discretization times ensuring invariance in continuous time.

Remark 9 (The use of extended class $\mathcal{K}$ functions): We can define $\alpha_i(\cdot), i \in \{1, \ldots, m\}$ in Definition 8 as extended class $\mathcal{K}$ functions ($\alpha : [-a, a] \to [-\infty, \infty]$ as in Definition 1) to ensure robustness of an HOCBF to perturbations [21]. However, the use of extended class $\mathcal{K}$ functions cannot ensure a constraint to be satisfied if it is initially violated for a relative degree one CBF, which can also cause a similar problem in an HOCBF since $\psi_i(\boldsymbol{x})$ in (10) is recursively defined.

V. Case Studies and Results

In this section, we complete the ACC case study and introduce a robot control problem. All the computations and simulations were conducted in MATLAB.

A. Adaptive Cruise Control

For the dynamics given by (7), we consider a cost $J(u) = \int_{t_0}^{t_f} u^2(t)dt$, and we require the vehicle to achieve a desired speed $v_d = 24$ m/s. For this, we define a CLF $V(\boldsymbol{x}) = (v - v_d)^2$ (see Definition 5).

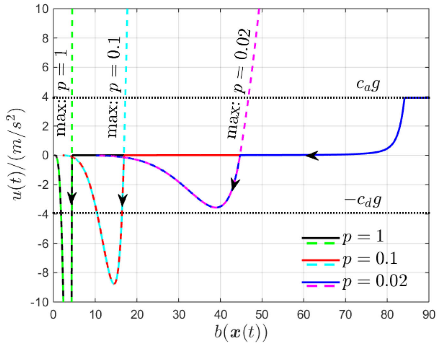

Fig. 1. Control input $u(t)$ as $b(\boldsymbol{x}(t)) \to 0$ for different p when using quadratic class $\mathcal{K}$ function. All the solid lines (black, red, blue) start from $b(x) = 90$. They coincide before the corresponding HOCBF constraint becomes active (e.g., the red solid line can only be seen after $b(x) <= 45$), when the solid line starts overlapping with its associated dashed line. The arrows denote the changing trend for $b(\boldsymbol{x}(t))$ with respect to time.

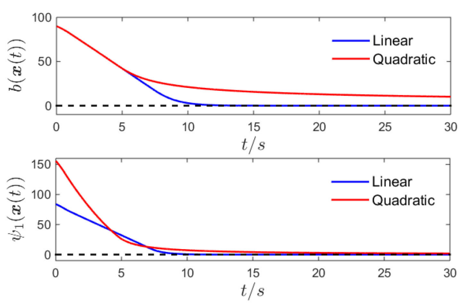

Fig. 2. Variations of functions $b(\boldsymbol{x}(t))$ and $\psi_1(\boldsymbol{x}(t))$ for linear ($p = 1$) and quadratic ($p = 0.02$) class $\mathcal{K}$ functions, respectively. $b(\boldsymbol{x}(t)) \geq 0$ and $\psi_1(\boldsymbol{x}(t)) \geq 0$ imply the forward invariance of $C_1 \cap C_2$.

We consider a control constraint $-0.4g \leq u(t) \leq 0.4\,g, g = 9.81\,\text{m/s}^2$. The relative degree of (8) is two, and we define three different HOCBFs as in Definition 8 by choosing square root, linear and quadratic class $\mathcal{K}$ functions for both $\alpha_1(\cdot), \alpha_2(\cdot)$ in (21) with penalties $p_1 = p_2 = p > 0$. The other parameters are the same as in [1].

We define $b(\boldsymbol{x}) = z - \delta$ as an HOCBF with $m = 2$. Since $L_f^2 b(\boldsymbol{x}) = 0$, the conditions in Corollary 1 are satisfied and we can find small enough p_1, p_2 such that the problem is feasible. We present the penalty case study for quadratic class $\mathcal{K}$ functions in Fig. 1. The dashed lines denote the values of the right-hand side of the HOCBF constraint (i.e., $\frac{L_f^m b(\boldsymbol{x}) + O(b(\boldsymbol{x})) + \alpha_m(\psi_{m-1}(\boldsymbol{x}))}{-L_g L_f^{m-1} b(\boldsymbol{x})}$), and the solid lines are the optimal controls. When the dashed lines and solid lines coincide, the HOCBF constraint for $b(\boldsymbol{x})$ is active.

In Fig. 1, the HOCBF constraint does not conflict with the braking limitation $-c_d g$ when $p = 0.02$ for a quadratic class $\mathcal{K}$ function. The minimum control input (negative) increases as p decreases. Then, we set p to be $1, 0.02$ for linear and quadratic class $\mathcal{K}$ functions, respectively. We present the forward invariance of the set $C_1 \cap C_2$, where $C_1 := \{\boldsymbol{x}(t) : b(\boldsymbol{x}(t)) \geq 0\}$ and $C_2 := \{\boldsymbol{x}(t) : \psi_1(\boldsymbol{x}(t)) \geq 0\}$ in Fig. 2.

B. Robot Control

Consider the unicycle model for a wheeled mobile robot $\dot{x} = v\cos\theta, \dot{y} = v\sin\theta, \dot{v} = u_2, \dot{\theta} = u_1, x, y$ denote the location, θ is the heading angle, v denotes the linear speed, and u_1, u_2 are the two control inputs (turning speed and forward acceleration). Note that the dynamics are in the form (2), with $\boldsymbol{x} = (x, y, \theta, v)^T$, $\boldsymbol{u} = (u_1, u_2)^T$, $f = (v\cos\theta, v\sin\theta, 0, 0)^T$, and $g = (0, 0; 0, 0; 1, 0; 0, 1)$.

In this problem, we have two *objectives*: (o1) *minimize energy consumption* $J(\boldsymbol{u}(t)) = \int_{t_0}^{t_f} (u_1^2(t) + u_2^2(t))dt$, and (o2) *reach destination* $(x_d, y_d) \in \mathbb{R}^2$, during time interval $[t_1, t_2], t_0 \leq t_1 \leq t_2 \leq t_f$, and two *constraints*: (c1) *safety* $(x(t) - x_o)^2 + (y(t) - y_o)^2 \geq r^2$, where $(x_o, y_o) \in \mathbb{R}^2$ denotes the location of a circular obstacle and $r = 7\,\text{m}$ is its size (a little larger than the actual size, which is 6 m), and (c2) *robot limitations*$v_{\min} \leq v(t) \leq v_{\max}$, $u_{1,\min} \leq u_1(t) \leq u_{1,\max}$, $u_{2,\min} \leq u_2(t) \leq u_{2,\max}$, where $v_{\min} = 0\,\text{m/s}, v_{\max} = 2\,\text{m/s}$, $u_{1,\max} = -u_{1,\min} = 0.2\,\text{rad/s}$, and $u_{2,\max} = -u_{2,\min} = 0.5\,\text{m/s}^2$.

We use HOCBFs to (strictly) impose constraints (c1) and (c2) and two CLFs $V_1(\boldsymbol{x}) = (\theta - \text{atan}(\frac{y_d - y}{x_d - x}))^2, V_2(\boldsymbol{x}) = (v - v_d)^2, v_d = 2\,\text{m/s}$ to achieve objective (o2). We capture objective (o1) in the cost of the optimization problem. For constraint (c1), we define an HOCBF $b(\boldsymbol{x}) = (x(t) - x_o)^2 + (y(t) - y_o)^2 - r^2$ with $m = 2$. Since $L_f^2 b(\boldsymbol{x}) = 2v^2$ is guaranteed to be nonnegative, the penalty method always works given proper $\boldsymbol{x}(t_0)$. However, we use the parameterization method for (c1), since we also wish to minimize the HOCBF value when the HOCBF constraint first becomes active. This approach gives good results in an unknown environment with obstacles, as shown below. We use two HOCBFs to impose the speed part of (c2): $b_{\max}(\boldsymbol{x}) = v_{\max} - v$ and $b_{\min}(\boldsymbol{x}) = v - v_{\min}$. Both have relative degree 1, and $L_f b_{\max}(\boldsymbol{x}) = L_f b_{\min}(\boldsymbol{x}) = 0$. If we set $\boldsymbol{u} = \boldsymbol{0}$, the speed will not change. Therefore, these HOCBF constraints do not conflict with the control bound, and we do not need to use the penalty or parameterization methods.

We find penalties p_1, p_2 and powers q_1, q_2 for a worst-case scenario. We consider maximum initial speed $v(t_0) = v_{\max}$ and initial position $(x(t_0), y(t_0)) = (5,\ 25\,\text{m})$. We assume the obstacle center $(x_o, y_o) = (32,\ 25\,\text{m})$ and the destination $(x_d, y_d) = (45,\ 25\,\text{m})$ are aligned, and the initial heading $\theta(t_0) = 0$ is also parallel to this line [see Fig. 3(a)]. We study the feasibility robustness of the solution (i.e., how feasibility is affected by changes in state and/or environment).

When the destination component y_d is exactly 25 m, the robot stops before the obstacle (if p_1, p_2, q_1, q_2 are feasible for the QP), i.e., it cannot arrive at the destination. We call this stop point an equilibrium point, as shown in Fig. 3(a). However, if y_d has a positive offset (arbitrary small), the robot can overpass the obstacle and arrive at the destination following the left trajectories shown in Fig. 3(a). Otherwise, the robot will produce right trajectories also shown in Fig. 3(a). Therefore, we choose a small offset for y_d (i.e., $y_d = 25.0000001$ m) when trying to find the optimal p_1, p_2, q_1, q_2.

We randomly sample p_1, p_2, q_1, q_2 ($p_1, p_2 \in (0, 3]$, $q_1, q_2 \in [1, 3]$) to get 2000 points, and run simulations for 30 s. We use the algorithm in Remark 6 to optimize each sample and get the optimal $(p_1^*, p_2^*, q_1^*, q_2^*) = (0.7535, 0.6664, 1.0046, 1.0267)$ such that the QPs are feasible and the HOCBF value is minimized when the HOCBF constraint first becomes active. The HOCBF constraint is active when $b(\boldsymbol{x}) = \mathcal{D}_{\min}$ ($\mathcal{D}_{\min} = 5.4\,\text{m}^2$, a distance metric instead of the real distance). In the rest of this section, we study the feasibility robustness of the proposed method, assuming the optimal $(p_1^*, p_2^*, q_1^*, q_2^*)$ given above. Note that the QP feasibility does not depend on the specific initial condition as long as the robot initially has a distance (in terms of $b(\boldsymbol{x})$) of at least $\mathcal{D}_{\min}$ from the obstacle, and the initial state is within the predefined bound (see [27]).

1) Feasibility Robustness to the Heading Angle: In this case, we only change the value of destination component y_d. Based on $y_d = 25.0000001$ m and $y_d = 24.9999999$ m, we further offset y_d

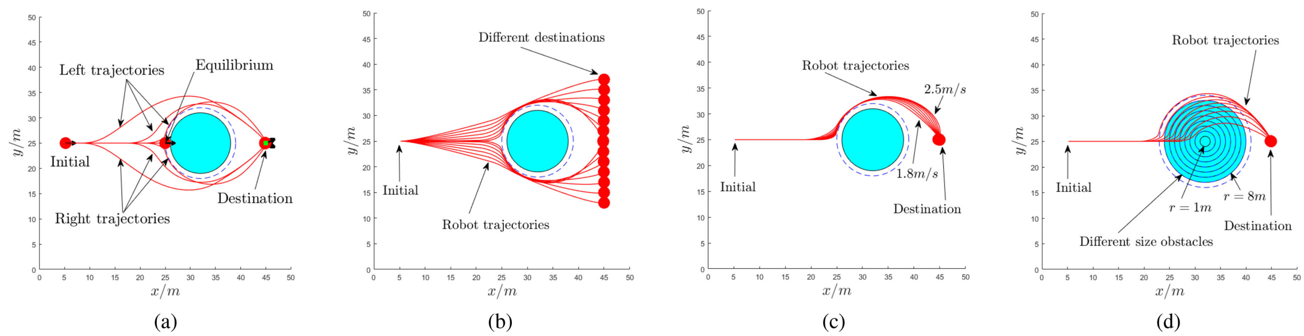

Fig. 3. Robot control problem using HOCBFs and the parameterization method. (a) Trajectories under different parameters. (b) Trajectories under different obstacle-approaching angles. (c) Trajectories under different obstacle-approaching speeds. (d) Trajectories for obstacles of different sizes.

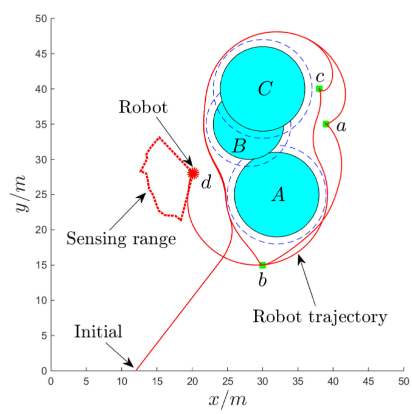

Fig. 4. Safe exploration in an unknown environment.

by $+2$ m for $y_d = 25.0000001$ m (-2 m for $y_d = 24.9999999$ m), and generate 7 destinations for both cases, respectively. These 14 destinations are all feasible, which shows good feasibility robustness of the penalty method to changes in the heading angle when approaching the obstacle, as shown in Fig. 3(b). The HOCBF values when the HOCBF constraint becomes active are all smaller than $\mathcal{D}_{\min}$.

2) Feasibility Robustness to the Approaching Speed: We vary the approaching speed to the obstacle between 1.8 and 2.5 m/s (2 m/s was the value for the original problem). All these values are all feasible, which shows good feasibility robustness of the penalty method to the change in speed when approaching the obstacle. The HOCBF values when the HOCBF constraint becomes active increase as the approaching speed increases, as shown in Fig. 3(c).

3) Feasibility Robustness to the Obstacle Size: Here, we only change the obstacle size from the predefined value $r = 7$ m. We consider a range of r between 2 and 9 m. The results show good feasibility robustness to the change of obstacle size as the QPs are always feasible and the robot can safely arrive at its destination, as shown in Fig. 3(d). The HOCBF values when the HOCBF constraint becomes active do not change under different-size obstacles.

Finally, in order to show the feasibility robustness is independent of the location of the obstacles, we present an application of robot safe exploration in an unknown environment. Suppose the robot is equipped with a sensor [$\frac{2}{3}\pi$ field of view and 7 m (greater than the one corresponding to $\mathcal{D}_{\min}$) sensing distance with $1 -$ m sensing uncertainty] to detect the obstacles, and there are three unknown obstacles (to the robot) whose center locations are $(32, 25\,\text{m}), (28, 35\,\text{m})$, and $(30, 40\,\text{m})$ with radius $6, 5$, and 6 m, respectively. The robot is required to arrive sequentially at points $a := (39, 35\,\text{m}), b := (30, 15\,\text{m}), c := (38, 40\,\text{m})$, and $d := (20, 28\,\text{m})$. The robot can safely arrive at these four destinations with the penalties and powers $(p_1, p_2, q_1, q_2) = (0.7535, 0.6664, 1.0046, 1.0267)$ as calculated earlier, which shows good feasibility robustness. The robot trajectory is shown in Fig. 4. The computation time to solve the QP at each time step is less than 0.01 s (Intel(R) Core(TM) i7-8700 CPU @ 3.2 GHz$\times$2).

VI. Conclusion

We extended BFs and CBFs to HOBFs and HOCBFs, and showed how they can be used to solve OCPs with safety requirements and control limitations for systems with high relative degree. We showed how the new definitions can be used to significantly increase the feasibility of the OCPs. We applied the proposed framework to an ACC problem and to a robot navigating in an unknown environment facing real-time safety constraints. In the future, we will investigate the use of machine learning techniques to improve system performance.

References

[1] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. 58th IEEE Conf. Decis. Control*, Nice, France, 2019, pp. 474–479.

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

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

[4] P. Glotfelter, J. Cortes, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Syst. Lett.*, vol. 1, no. 2, pp. 310–315, Oct. 2017.

[5] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 96–101, Jan. 2019.

[6] E. Sontag, "A Lyapunov-like stabilization of asymptotic controllability," *SIAM J. Control Optim.*, vol. 21, no. 3, pp. 462–471, 1983.

[7] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Anal.: Theory, Methods, Appl.*, vol. 7, no. 11, pp. 1163–1173, 1983.

[8] A. D. Ames, K. Galloway, and J. W. Grizzle, "Control Lyapunov functions and hybrid zero dynamics," in *Proc. 51st IEEE Conf. Decis. Control*, 2012, pp. 6837–6842.

[9] P. Wieland and F. Allgower, "Constructive safety using control barrier functions," in *Proc. 7th IFAC Symp. Nonlinear Control Syst.*, 2007, pp. 473–478.

[10] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[11] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[12] J. P. Aubin, *Viability Theory*. Berlin, Germany: Springer, 2009.
[13] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1415–1428, Aug. 2007.
[14] R. Wisniewski and C. Sloth, "Converse barrier certificate theorem," in *Proc. 52nd IEEE Conf. Decis. Control*, Florence, Italy, 2013, pp. 4713–4718.
[15] D. Panagou, D. M. Stipanovic, and P. G. Voulgaris, "Multi-objective control for multi-agent systems using Lyapunov-like barrier functions," in *Proc. 52nd IEEE Conf. Decis. Control*, Florence, Italy, 2013, pp. 1478–1483.
[16] K. Galloway, K. Sreenath, A. D. Ames, and J. Grizzle, "Torque saturation in bipedal robotic walking through control Lyapunov function based quadratic programs," *IEEE*, vol. 3, pp. 323–332, 2015.
[17] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Santa Barbara, CA, USA: Nob Hill Publishing, 2017.
[18] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *Proc. Amer. Control Conf.*, 2015, pp. 4542–4548.
[19] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control Lyapunov and control barrier functions for systems evolving on manifolds," in *Proc. Amer. Control Conf.*, 2015, pp. 2038–2044.
[20] X. Xu, "Constrained control of input-output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018.
[21] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.
[22] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[23] W. Xiao *et al.*, "Rule-based optimal control for autonomous driving," in *Proc. ACM/IEEE 12th Int. Conf. Cyber-Phys. Syst.*, 2021, pp. 143–154.
[24] A. Robey *et al.*, "Learning control barrier functions from expert demonstrations," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 3717–3724.
[25] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 530–545, Mar. 2021.
[26] G. Yang, C. Belta, and R. Tron, "Self-triggered control for safety critical systems using control barrier functions," in *Proc. Amer. Control Conf.*, 2019, pp. 4454–4459.
[27] W. Xiao, C. Belta, and C. G. Cassandras, "Feasibility guided learning for constrained optimal control problems," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 1896–1901.