

Safe Control Synthesis via Input Constrained Control Barrier Functions

Devansh R. Agrawal and Dimitra Panagou

Abstract—This paper introduces the notion of an Input Constrained Control Barrier Function (ICCBF), as a method to synthesize safety-critical controllers for non-linear control affine systems with input constraints. The method identifies a subset of the safe set of states, and constructs a controller to render the subset forward invariant. The feedback controller is represented as the solution to a quadratic program, which can be solved efficiently for real-time implementation. Furthermore, we show that ICCBFs are a generalization of Higher Order Control Barrier Functions, and thus are applicable to systems of non-uniform relative degree. Simulation results are presented for the adaptive cruise control problem, and a spacecraft rendezvous problem.

I. INTRODUCTION

Many cyber-physical systems are safety critical, that is, they require guarantees that safety constraints are not violated during operation. Safety is often modeled by defining a safe subset of the state space for a given system, within which the state trajectories must always evolve. Recently, set-theoretic methods, such as Control Barrier Functions (CBFs) have become increasingly popular as a means of constructing and verifying such controllers [1]–[4].

Quadratic Programs (QPs) enforcing CBF conditions have become popular for synthesizing low-level controllers to ensure that system trajectories remain within the safe set. These QPs can be solved in real-time at the current state, enabling safety-critical controllers for a wide range of non-linear systems to be computed efficiently during operation.

Prior work on CBFs has largely focused on systems where a sufficiently large control authority is available to ensure forward invariance of the safe set. However in the presence of input constraints, only a subset of the safe set may be rendered forward invariant. A few methods have been proposed to compute such subsets of the safe set. These include reachability analysis by solving a Hamilton-Jacobi partial differential equations over the state-space [3], [5] and Sum-of-Squares (SOS) based semi-definite programs, which employ the positivstellensatz theorem to provide a certificate of safety [6], [7]. Both methods scale poorly with the dimension of the state-space, and SOS methods assume the dynamics and safety constraints can be expressed as polynomials. Some methods have also been proposed for dynamical systems of specific classes, for instance Euler-Lagrange systems [8].

The authors would like to acknowledge the support of the National Science Foundation under award number 1942907.

Both authors are with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA; {devansh, dpanagou}@umich.edu

In this paper, we introduce the notion of an Input Constrained Control Barrier Function (ICCBF). An ICCBF guarantees that an input constrained controller can render a subset of the safe set forward invariant. We present the construction of these functions, and define the subset of the safe set, and a corresponding controller, such that the subset is guaranteed to be forward invariant under this controller. Furthermore, we show that ICCBFs represent a generalization of Higher Order CBFs (HOCBFs) [9], enabling synthesis of input-constrained controllers for safe sets of higher, and non-uniform relative degree. Finally, the method is applied to an adaptive cruise control problem [10], and a spacecraft rendezvous problem, demonstrating that ICCBFs define a safe controller that respects input constraints.

The paper is structured as follows: In Section 2, some preliminaries on set invariance, as well as the problem statement are presented. Section 3 presents the definition of ICCBFs, and the synthesis of corresponding feedback controllers for ensuring safety. Finally, simulation results are presented in section 4.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. Notation

We denote the set of real numbers \mathbb{R} and non-negative reals \mathbb{R}_+ . A continuous function $\alpha : [0, a) \rightarrow [0, \infty)$ is class- \mathcal{K} if it is strictly increasing on the domain, and $\alpha(0) = 0$. It is class- \mathcal{K}_∞ if $a = \infty$ and $\lim_{r \rightarrow \infty} \alpha(r) = \infty$. A continuous function $\alpha : (-b, a) \rightarrow (-\infty, \infty)$ is an extended class- \mathcal{K} function, if for some $a, b > 0$, if it is strictly increasing and $\alpha(0) = 0$. The Lie derivative of $h(x)$ along $f(x)$ is denoted $L_f h(x) = \frac{dh}{dx} f(x)$. The short hand $\dot{h}(x) = \frac{dh}{dx} \dot{x}$ will also be used to indicate time derivatives of h along flows from a state x . We denote $\text{Int}(\mathcal{C})$ and $\partial\mathcal{C}$ as the interior and boundary of a set \mathcal{C} . The empty set is denoted \emptyset .

B. Problem Setup

Consider a nonlinear, control-affine dynamical system, with state $x \in \mathcal{X} \subset \mathbb{R}^n$ and control input $u \in \mathcal{U} \subset \mathbb{R}^m$ given as

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

where $f : \mathcal{X} \rightarrow \mathbb{R}^n$, $g : \mathcal{X} \rightarrow \mathbb{R}^{n \times m}$ are sufficiently smooth, as discussed in III-B. We will assume these functions are known, and the system state is available to the controller.

We define a state x as *safe* if it lies in a set \mathcal{S} , which is defined as the 0-superlevel set of a continuously differentiable

function $h : \mathcal{X} \rightarrow \mathbb{R}$, as follows:

$$\mathcal{S} \triangleq \{x \in \mathcal{X} : h(x) \geq 0\} \quad (2)$$

$$\partial\mathcal{S} \triangleq \{x \in \mathcal{X} : h(x) = 0\} \quad (3)$$

$$\text{Int}(\mathcal{S}) \triangleq \{x \in \mathcal{X} : h(x) > 0\} \quad (4)$$

The set \mathcal{S} is referred to as the *safe set*. We will assume this set is closed, non-empty and is simply connected.

Definition 1. Under a control input $u \in \mathcal{U}$, the set \mathcal{S} is forward invariant for the closed-loop system (1) if for all $x(0) \in \mathcal{S}$, it holds that $x(t) \in \mathcal{S}$ for all $t \geq 0$.

Under input constraints, however, it may be impossible to render the safe set forward invariant, as in Example 1:

Example 1. Consider the unstable single-input, scalar dynamical system with input constraints:

$$\dot{x} = x + u, \quad u \in \mathcal{U} = [-1, 1]$$

Let the safety set be

$$\mathcal{S} = \{x \in \mathbb{R} : x \leq 2\}, \quad h(x) = 2 - x$$

Now consider the boundary state $x = 2$. In this case,

$$\dot{h}(x, u) = -2 - u \leq -1$$

and therefore under any $u \in \mathcal{U}$, closed-loop trajectories will leave safe set. Thus \mathcal{S} cannot be rendered forward invariant under the input constraints. \triangle

To address the problem when the safe set cannot be rendered forward invariant, we define an *inner safe set* as follows:

Definition 2. A non-empty closed set \mathcal{C}^* is an inner safe set of the safe set \mathcal{S} for the dynamical system (1), if $\mathcal{C}^* \subseteq \mathcal{S}$ and there exists a feedback controller $u : \mathcal{C}^* \rightarrow \mathcal{U}$ such that for the closed-loop system (1), for all $x(0) \in \mathcal{C}^*$, it holds that $x(t) \in \mathcal{C}^*$ for all $t \geq 0$.

Now, we can state the main objective of this paper:

Problem 1. Given the system (1), find a closed set $\mathcal{C}^* \subseteq \mathcal{S}$ and a feedback controller $u : \mathcal{C}^* \rightarrow \mathcal{U}$, such that for any $x(0) \in \mathcal{C}^*$, the closed-loop trajectories of (1) satisfy $x(t) \in \mathcal{C}^*$ for all $t \geq 0$.

In words, the objective is to find a subset of the safe set, and a corresponding feedback controller that renders the subset forward invariant.

C. Set Invariance

Nagumo's theorem provides a necessary and sufficient condition for the forward invariance of a set \mathcal{S} .

Lemma 1. [11, Thm. 3.1] Consider the system $\dot{x} = f(x)$, and assume that for each initial condition $x \in \mathcal{X}$, it admits a globally unique solution. Then, a closed set $\mathcal{S} \subseteq \mathcal{X}$ is forward invariant for the system, if and only if

$$f(x) \in \hat{T}_{\mathcal{S}}(x), \quad \forall x \in \partial\mathcal{S}, \quad (5)$$

where $\hat{T}_{\mathcal{S}}(x)$ is the tangent cone of \mathcal{S} at x .

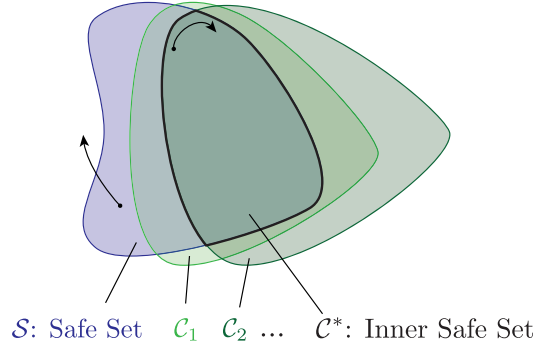


Fig. 1. Visual representation of ICCBF method. The safe set \mathcal{S} and two intermediate sets \mathcal{C}_1 and \mathcal{C}_2 are drawn. The final inner safe set \mathcal{C}^* is the intersection of each of these sets, and can be rendered forward invariant.

In this work, since \mathcal{S} is defined by $h(x) \geq 0$, Nagumo's theorem simplifies to the following statement.

Lemma 2. Consider the system (1). Let the set \mathcal{S} be defined by a continuously differentiable function $h : \mathcal{X} \rightarrow \mathbb{R}$, as per (2-4). Consider a controller $u : \mathcal{S} \rightarrow \mathcal{U}$, such that for any initial condition $x \in \mathcal{S}$, the closed-loop system (1) admits a globally unique solution. Then set \mathcal{S} is forward invariant if and only if

$$L_f h(x) + L_g h(x)u(x) \geq 0, \quad \forall x \in \partial\mathcal{S}. \quad (6)$$

If h is a Zeroing Control Barrier Function (ZCBF), a controller that renders \mathcal{S} forward invariant is guaranteed to exist [1], [2]. The focus of this paper is on scenarios where h is not a ZCBF. In the following section, we propose Input Constrained Control Barrier Functions (ICCBFs) to address these cases.

III. INPUT CONSTRAINED CONTROL BARRIER FUNCTIONS

In this section we define Input Constrained Control Barrier Functions (ICCBFs). A brief motivation for the construction is provided, followed by the formal definitions. Finally a method to find ICCBFs is proposed.

A. Motivation

Suppose the safe set \mathcal{S} associated with h cannot be rendered forward invariant by any controller $u \in \mathcal{U}$. Then we define a function $b_1 : \mathcal{X} \rightarrow \mathbb{R}$ and a set \mathcal{C}_1 (visualised in Figure 1) as follows

$$b_1(x) = \inf_{u \in \mathcal{U}} [\dot{h}(x, u) + \alpha_0(h(x))] \quad (7)$$

$$\mathcal{C}_1 = \{x \in \mathcal{X} : b_1(x) \geq 0\} \quad (8)$$

where α_0 is some user specified extended class- \mathcal{K} function and the shorthand $\dot{h}(x, u) = L_f h(x) + L_g h(x)u$ is used. Notice that since an infimum over \mathcal{U} is taken, b_1 is only dependent on the state x , and not the input u .

The set \mathcal{C}_1 has a useful property: Suppose there exists a point $x \in \mathcal{X}$ such that $x \in \partial\mathcal{S}$ and $x \in \mathcal{C}_1$, i.e., $h(x) = 0$

and $b_1(x) \geq 0$. Then, from (7) it follows, $\inf_{u \in \mathcal{U}} [\dot{h}] \geq 0$, and therefore $\dot{h} \geq 0$ for any $u \in \mathcal{U}$, i.e.,

$$x \in \partial \mathcal{S} \cap \mathcal{C}_1 \implies \dot{h}(x, u) \geq 0, \forall u \in \mathcal{U}. \quad (9)$$

Thus the closed-loop vector field $f(x) + g(x)u$ lies in the tangent cone of \mathcal{S} at x , i.e., no closed-loop trajectory can leave \mathcal{S} through such an x .

Now suppose there exists a Lipschitz continuous controller $u \in \mathcal{U}$ such that \mathcal{C}_1 is forward invariant. Then the set $\mathcal{S} \cap \mathcal{C}_1$ is also forward invariant: Consider an initial condition $x(0) \in \mathcal{S} \cap \mathcal{C}_1$. Suppose there exists a $t > 0$ where $h(x(t)) < 0$. By continuity, there must be some $\tau \in [0, t)$ where $h(x(\tau)) = 0$ and $\dot{h}(x(\tau), u) < 0$. However, since the controller renders \mathcal{C}_1 forward invariant, we have $x(\tau) \in \partial \mathcal{S}$ and $x(\tau) \in \mathcal{C}_1$. By (9), $\dot{h}(x(\tau), u) \geq 0$, which is a contradiction. Therefore, trajectories cannot reach a state $x \notin \mathcal{S}$.

To summarise, in systems where \mathcal{S} cannot be rendered forward invariant, the subset set $\mathcal{S} \cap \mathcal{C}_1$ is forward invariant under any controller that renders \mathcal{C}_1 forward invariant.

In defining b_1 , we allowed the user to specify the extended class- \mathcal{K} function α_0 . However not all class- \mathcal{K} functions will admit a controller that renders \mathcal{C}_1 forward invariant. Either a different class- \mathcal{K} function could be used, or we can repeat the same steps: define $b_2(x) = \inf_{u \in \mathcal{U}} [\dot{b}_1(x, u) + \alpha_1(b_1(x))]$ and $\mathcal{C}_2 = \{x \in \mathcal{X} : b_2(x) \geq 0\}$. Now any controller u that renders \mathcal{C}_2 forward invariant also renders $\mathcal{C}_1 \cap \mathcal{C}_2$ forward invariant, and therefore the set $\mathcal{S} \cap \mathcal{C}_1 \cap \mathcal{C}_2$ is also forward invariant by the same controller. This idea is formalised in the next subsection.

B. ICCBFs

Consider the dynamical system (1) with bounded control inputs $u \in \mathcal{U}$ and a safe set \mathcal{S} defined by a function $h : \mathcal{X} \rightarrow \mathbb{R}$, as per (2-4). Assume h is not a ZCBF. We define the following sequence of functions:

$$b_0(x) = h(x) \quad (10)$$

$$b_1(x) = \inf_{u \in \mathcal{U}} [L_f b_0(x) + L_g b_0(x)u + \alpha_0(b_0(x))] \quad (11)$$

$$b_2(x) = \inf_{u \in \mathcal{U}} [L_f b_1(x) + L_g b_1(x)u + \alpha_1(b_1(x))] \quad (12)$$

\vdots

$$b_{i+1}(x) = \inf_{u \in \mathcal{U}} [L_f b_i(x) + L_g b_i(x)u + \alpha_i(b_i(x))] \quad (13)$$

\vdots

$$b_N(x) = \inf_{u \in \mathcal{U}} [L_f b_{N-1}(x) + L_g b_{N-1}(x)u + \alpha_{N-1}(b_{N-1}(x))] \quad (14)$$

where each α_i is some user-specified class- \mathcal{K} function, and N is a positive integer. We assume the functions f, g, h are sufficiently smooth such that b_N and its derivative are defined. Each function $b_i : \mathcal{X} \rightarrow \mathbb{R}$ is a scalar function that only depends on the state. The time derivative $\dot{b}_i = \frac{db_i}{dx} \dot{x} = L_f b_i(x) + L_g b_i(x)u$ is still affine in u .

Remark 1. Higher Order CBFs, as in [9], are a special case of ICCBFs. For instance, in systems of relative degree 2, $L_g h(x) = 0$ for all $x \in \mathcal{S}$. In this case, in the construction of ICCBFs we have

$$b_1(x) = \inf_{u \in \mathcal{U}} [L_f h(x) + L_g h(x)u + \alpha_0(h(x))] \quad (15)$$

$$= \inf_{u \in \mathcal{U}} [L_f h(x) + \alpha_0(h(x))] \quad (16)$$

$$= L_f h(x) + \alpha_0(h(x)) \quad (17)$$

which is exactly the function defined in [9]. This repeats for any relative degree greater than 2, and thus for a system with relative degree ρ , the first ρ expressions of ICCBFs are identical to those of HOCBFs. Moreover, ICCBFs can handle systems with non-uniform relative degree, by choosing N greater or equal to the largest relative degree of the system in \mathcal{S} .

Next, we define a family of sets,

$$\mathcal{C}_0 = \{x \in \mathcal{X} : b_0(x) \geq 0\} = \mathcal{S} \quad (18)$$

$$\mathcal{C}_1 = \{x \in \mathcal{X} : b_1(x) \geq 0\} \quad (19)$$

\vdots

$$\mathcal{C}_i = \{x \in \mathcal{X} : b_i(x) \geq 0\} \quad (20)$$

\vdots

$$\mathcal{C}_N = \{x \in \mathcal{X} : b_N(x) \geq 0\} \quad (21)$$

and the intersection of these sets

$$\mathcal{C}^* = \mathcal{C}_0 \cap \mathcal{C}_1 \cap \dots \cap \mathcal{C}_N \quad (22)$$

We assume the set \mathcal{C}^* is closed, non-empty and has no isolated points.

Definition 3. For the above construction, if there exists a class- \mathcal{K} function α_N such that

$$\sup_{u \in \mathcal{U}} [L_f b_N(x) + L_g b_N(x)u + \alpha_N(b_N(x))] \geq 0 \quad \forall x \in \mathcal{C}^*, \quad (23)$$

then b_N is an Input Constrained Control Barrier Function (ICCBF).

Note, this does not require b_N to be a ZCBF on \mathcal{C}_N . The definition only requires condition (23) to hold for all $x \in \mathcal{C}^*$ which is a subset of \mathcal{C}_N . The main result of this paper can now be stated:

Theorem 1 (Main Result). Given the input constrained dynamical system (1), if b_N , defined by (10 - 14), is an ICCBF, then any Lipschitz continuous controller $u : \mathcal{C}^* \rightarrow \mathcal{U}$ such that $u(x) \in K_{ICCBF}(x)$, where

$$K_{ICCBF}(x) = \{u \in \mathcal{U} : L_f b_N(x) + L_g b_N(x)u \geq -\alpha_N(b_N(x))\} \quad (24)$$

renders the set $\mathcal{C}^* \subseteq \mathcal{S}$ (22) forward invariant.

Proof. Since u is a Lipschitz continuous controller, the closed-loop system (1) is also Lipschitz continuous. To show

forward invariance of \mathcal{C}^* , we use Nagumo's theorem on the closed-loop system. In particular, we show

$$\begin{aligned} [f(x) + g(x)u(x)] &\in \hat{T}_{\mathcal{C}^*}(x), \\ \forall u(x) &\in K_{\text{ICCBF}}(x), \text{ and } \forall x \in \partial\mathcal{C}^*, \end{aligned} \quad (25)$$

where $\hat{T}_{\mathcal{C}^*}(x)$ is the tangent cone of \mathcal{C}^* at x . At each $x \in \partial\mathcal{C}^*$, we have $b_i(x) = 0$ for some set, possibly all, of i 's in $\{0, \dots, N\}$. Let $\mathcal{I}(x) = \{i : b_i(x) = 0\}$. Condition (25) is satisfied if, $b_i(x, u(x)) \geq 0$, for all $i \in \mathcal{I}(x)$, for all $x \in \mathcal{C}^*$, provided $u(x) \in K_{\text{ICCBF}}(x)$. In other words, we show that for all $x \in \mathcal{C}^*$,

$$b_i(x) = 0 \implies \dot{b}_i(x, u(x)) \geq 0, \forall u(x) \in K_{\text{ICCBF}}(x) \quad (26)$$

for all $i \in \{0, \dots, N\}$.

Cases $i \in \{0, \dots, N-1\}$: Consider any $x \in \mathcal{C}^* \cap \partial\mathcal{C}_i$. Since $\mathcal{C}^* \subseteq \mathcal{C}_{i+1}$, $x \in \partial\mathcal{C}_i \cap \mathcal{C}_{i+1}$. By (13, 20),

$$\begin{aligned} \inf_{u \in \mathcal{U}} [L_f b_i(x) + L_g b_i(x)] &\geq 0 \\ \therefore L_f b_i(x) + L_g b_i(x) &\geq 0, \quad \forall u \in \mathcal{U} \end{aligned}$$

and therefore $\dot{b}_i(x, u) \geq 0$, for all $u \in \mathcal{U}$. Since $K_{\text{ICCBF}}(x) \subseteq \mathcal{U}$, condition (26) is satisfied.

Case $i = N$: Consider $x \in \mathcal{C}^* \cap \partial\mathcal{C}_N$. Since b_N is an ICCBF and $b_N(x) = 0$, by (21, 24),

$$L_f b_N(x) + L_g b_N(x)u(x) \geq 0, \quad \forall u(x) \in K_{\text{ICCBF}}(x)$$

and therefore $\dot{b}_N(x, u) \geq 0$ for any $u(x) \in K_{\text{ICCBF}}(x)$.

In conclusion, we have shown that condition (26) is satisfied for all i , and therefore the conditions of Nagumo's theorem are satisfied, which completes the proof. \square

Remark 2. The practical value of this construction is that for a given system, a set \mathcal{S} of safe states of practical importance can be specified, which may not be rendered forward invariant under the given system dynamics. By using ICCBFs, we remove states of the set \mathcal{S} , and construct an inner set for which we can find a controller that renders it forward invariant.

We would like to note a useful special case, the simple ICCBF:

Definition 4. In the above construction, if \mathcal{C}^* is a strict subset of \mathcal{C}_N , i.e., $\mathcal{C}^* \subset \mathcal{C}_N$, then b_N is a simple ICCBF.

Theorem 2. For the dynamical system (1), if b_N is a simple ICCBF, all Lipschitz continuous controllers $u(x) \in \mathcal{U}$ render the set \mathcal{C}^* forward invariant.

Proof. By definition, since b_N is a simple ICCBF, \mathcal{C}^* is a strict subset of \mathcal{C}_N . Then $\mathcal{C}^* \cap \partial\mathcal{C}_N = \emptyset$, i.e., there does not exist a $x \in \mathcal{C}^*$ such that $b_N(x) = 0$. Following Theorem 1, we do not need to consider case where $i = N$ in condition (26). The remaining cases, with $i \in \{0, \dots, N-1\}$ satisfy condition (26) for all $u(x) \in K_{\text{ICCBF}}(x)$. Therefore, any Lipschitz continuous $u \in \mathcal{U}$ admits globally unique solutions and satisfies condition (25), completing the proof. \square

Remark 3. Intuitively, the existence of a simple ICCBF represents a system where the dynamics at the boundaries of

\mathcal{C}^* are such that the unforced dynamics $f(x)$ dominate the forcing term $g(x)u$ in driving the system towards safety. If a simple ICCBF is found, no safety critical controller is needed for the system to ensure state trajectories remain within the safe set, provided the system is initialised within \mathcal{C}^* .

C. QP-based Control Synthesis

Finally, similar to [2], [3], we define an optimization-based controller that will render $\mathcal{C}^* \subseteq \mathcal{S}$ forward invariant. In this section, suppose the set \mathcal{U} can be expressed as a convex polytope, i.e., $\mathcal{U} = \{u \in \mathbb{R}^m : Au \leq B\}$ where $A \in \mathbb{R}^{p \times m}$ and $B \in \mathbb{R}^p$ for some positive integer p . Then a controller based on the solution to the following Quadratic Program (QP) renders the set \mathcal{C}^* forward invariant.

Theorem 3. Consider the dynamical system (1). Suppose b_N is an ICCBF associated with the class- \mathcal{K} function α_N and inner safe set \mathcal{C}^* . Let $H : \mathcal{X} \rightarrow \mathbb{R}_+^{m \times m}$, where $\mathbb{R}_+^{m \times m}$ is the set of real $m \times m$ positive definite matrices, and $F : \mathcal{X} \rightarrow \mathbb{R}^m$ be Lipschitz continuous cost functions. The solution to the following quadratic optimization problem

$$\begin{aligned} u^*(x) &= \underset{u \in \mathbb{R}^m}{\operatorname{argmin}} \quad \frac{1}{2}u^T H(x)u + F(x)^T u \\ \text{subject to} \quad &L_f b_N(x) + L_g b_N(x)u \geq -\alpha_N(b_N(x)) \\ &u \in \mathcal{U} \end{aligned} \quad (27)$$

yields an input-constrained feedback controller $u^* : \mathcal{C}^* \rightarrow \mathcal{U}$ that renders \mathcal{C}^* forward invariant. Thus, $h(x(t)) \geq 0$ for all $t \geq t_0$, i.e. the system trajectories always evolve within the safe set \mathcal{S} .

Proof. The constraints in (27) specify that $u^*(x) \in K_{\text{ICCBF}}(x)$. Since the $u^*(x)$ is a Lipschitz continuous controller, (following the same arguments as [12, Thm. 8]), by Theorem 1, the controller $u^*(x)$ obtained as the solution to the optimization problem (27) renders \mathcal{C}^* forward invariant. \square

Remark 4. Notice the final optimization-based controller only requires b_N and its derivatives. This means the computational complexity of the controller is almost identical to the CBF-QP based controllers of [2], [10].¹

To conclude, we have introduced the notion of an ICCBF. Once an ICCBF is found, the inner safe set $\mathcal{C}^* \subseteq \mathcal{S}$ given by (22) and the quadratic-program based feedback controller $u^* : \mathcal{C}^* \rightarrow \mathcal{U}$ given by (27) renders the \mathcal{C}^* forward invariant. In this way, ICCBFs address Problem 1.

IV. SIMULATIONS

A. Adaptive Cruise Control

As a demonstration, we apply ICCBFs to the Adaptive Cruise Control (ACC) problem of [10]. Consider a point-mass model of a vehicle moving in a straight line. The vehicle is following a vehicle in-front, which moves at

¹Computing b_N and its derivative can involve long expressions. In our implementations, we used Mathematica [13] for symbolic manipulation, for the ACC case study, and Julia's auto-differentiation package ForwardDiff.jl [14], [15], for the autonomous rendezvous case study.

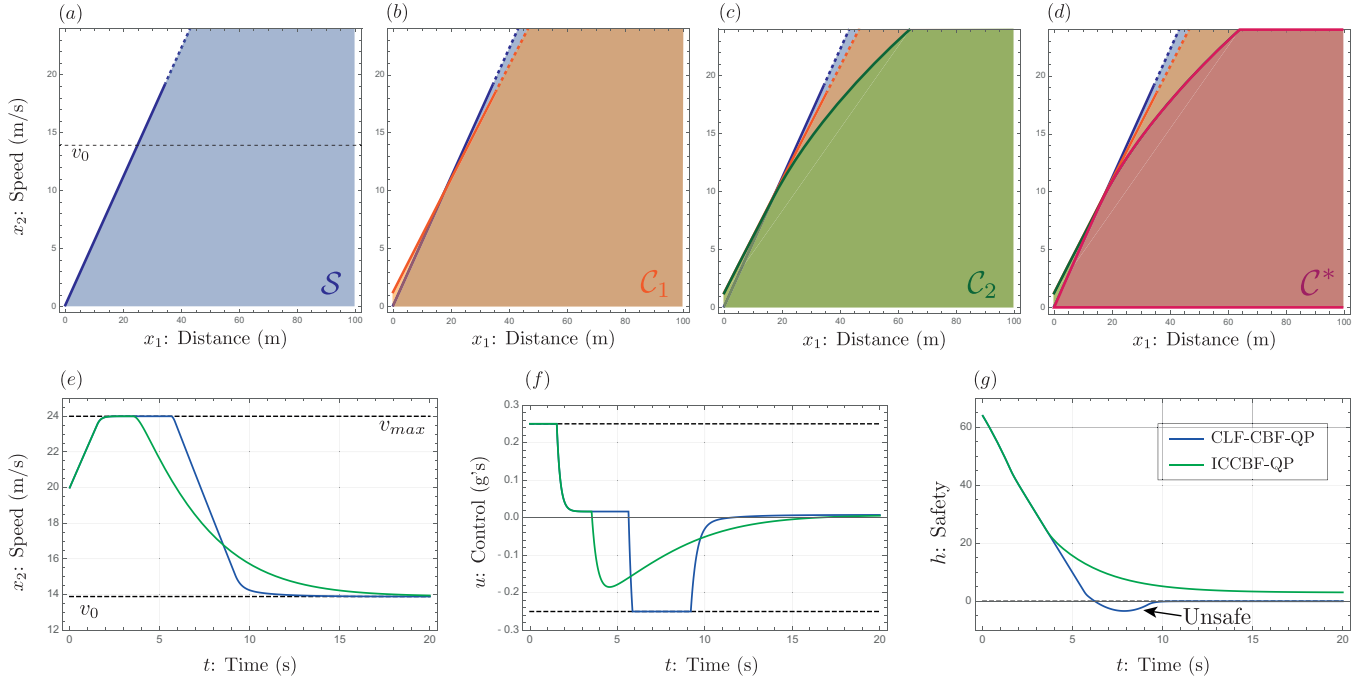


Fig. 2. Figures (a-d): State-space diagrams indicating the sets (a) \mathcal{S} , (b) \mathcal{C}_1 , (c) \mathcal{C}_2 and (d) \mathcal{C}^* . The horizontal dashed line in (a) indicates v_0 , the speed of the car in-front. Figure (d) represents the inner safe set \mathcal{C}^* that is rendered forward invariant. Figures (e-g): Simulation results for speed, control input and safety under the CLF-CBF-QP controller [10] and the ICCBF-QP.

a known constant speed v_0 . The objective is to design a controller which will prevent the vehicle from colliding with the vehicle in-front, but will allow it to accelerate to the speed limit when it is safe to do so.

Using the model defined in [10], the safety constraint is specified as $d \geq 1.8v$, where d is the distance to the vehicle in-front, and v is the speed of the vehicle. Defining the state of the system $x = [d, v]^T$, the dynamical model is

$$\begin{bmatrix} \dot{d} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v_0 - v \\ -F(v)/m \end{bmatrix} + \begin{bmatrix} 0 \\ g_0 \end{bmatrix} u$$

where $F(v) = f_0 + f_1v + f_2v^2$ models resistive forces on the vehicle, m is the mass of the vehicle, g_0 is acceleration due to the gravity. The control input u is the acceleration of the vehicle, and we suppose it is constrained to

$$\mathcal{U} = \{u : -0.25 \leq u \leq 0.25\}$$

which represents a maximum deceleration of $0.25g$'s. The safe set \mathcal{S} is defined by the function h ,

$$\mathcal{S} = \{x \in \mathcal{X} : h(x) \geq 0\}$$

$$h(x) = x_1 - 1.8x_2$$

and we can verify that \mathcal{S} is not forward invariant under the input constraints. Thus, h is not a ZCBF, and we will apply ICCBFs to find an inner safe set.

We choose, arbitrarily, $N = 2$ and the class- \mathcal{K} functions

$$\alpha_0(h) = 4h, \quad \alpha_1(h) = 7\sqrt{h}, \quad \alpha_2(h) = 2h,$$

to define the functions b_1, b_2 and sets $\mathcal{C}_1, \mathcal{C}_2$.

The sets are visualized in Figure 2. The interior of a set is shaded, and the boundary of the set is indicated with a thick line. Each boundary is partitioned into two subsets: the subset where there exists a feasible control input to keep trajectories within the set are indicated with a solid line, and the subset where no feasible control input will keep trajectories within the set are indicated with a dashed line.

To verify that b_2 is an ICCBF, the following optimization problem was used:

$$\begin{aligned} \gamma = \underset{x \in \mathcal{X}}{\text{minimise}} \quad & \sup_{u \in \mathcal{U}} [\dot{b}_N(x, u) + \alpha_N(b_N(x))] \\ \text{subject to} \quad & x \in \mathcal{C}^* \end{aligned}$$

By the definition of ICCBFs, b_N is an ICCBF if and only if the optimization problem is feasible, with solution $\gamma \geq 0$. While this optimization problem is, in general, non-linear and non-convex, a standard non-linear optimization software can be used to solve it. Applied to the ACC problem, we determined $\gamma = 2.33 \geq 0$, and therefore b_2 is a valid ICCBF.

Finally, we define \mathcal{C}^* as the intersection of $\mathcal{S}, \mathcal{C}_1, \mathcal{C}_2$. This set is visualised in Figure 2(d). The following controller renders the set forward invariant:

$$\begin{aligned} u^*(x) = \underset{u \in \mathcal{R}}{\text{argmin}} \quad & \frac{1}{2}(u - u_d(x))^2 \\ \text{subject to} \quad & L_f b_2(x) + L_g b_2(x)u \geq -2b_2(x) \\ & u \in \mathcal{U} \end{aligned}$$

where $u_d(x)$ is the desired acceleration. The desired acceleration is computed using the Control Lyapunov Function $V(x) = (x_2 - v_{max})^2$, where $v_{max} = 24$ is the speed limit.

Thus, $u_d(x)$:

$$L_f V(x) + L_g V(x)u_d(x) = -10V(x)$$

The inner safe set \mathcal{C}^* and the input-constrained feedback controller u^* represent the solution to Problem 1 for the ACC problem.

We compare our controller to the CLF-CBF-QP controller of [10]:

$$\begin{aligned} & \underset{u \in R, \delta \in R_+}{\text{argmin}} && \frac{1}{2}u^2 + 0.1\delta^2 \\ & \text{subject to} && L_f V(x) + L_g V(x)u \leq -10V(x) + \delta \\ & && L_f h(x) + L_g h(x)u \geq -2h(x) \end{aligned}$$

and clip of the solutions of the QP such that $u^*(x)$ lies in the range of feasible control inputs.

In Figures 2 (e-g), the proposed controller (green) is compared to the CLF-CBF-QP controller (blue), proposed in [10]. We can see the CLF-CBF-QP reaches the input-constraint at $t = 5.9$ seconds. The input limits force the system to leave the safe set. The ICCBF-QP remains feasible and safe for the entire duration, by applying brakes early, at $t = 2.9$ seconds, instead of $t = 5.0$ seconds. Thus, the ICCBF-QP controller is able to keep the input-constrained system safe, where the CLF-CBF-QP method fails, because it does not consider input constraints.

B. Autonomous Rendezvous

In this section, the ICCBF method is applied to the case study of an autonomous rendezvous operation between a chaser spacecraft modelled as a point mass, and a target body, for instance the International Space Station (ISS), see Figure 3. The target spacecraft is rotating with a constant angular velocity relative to the Local-Vertical Local-Horizontal (LVLH) frame. The objective is to determine the appropriate propulsive forces to bring the chaser spacecraft from a range of 100 m to 3 m. The safety constraint is to maintain a line-of-sight (LOS) constraint: the spacecraft's position must remain within a $\gamma = 10^\circ$ cone of the docking axis. This model is adapted from [16], where linearized Model-Predictive-Controllers (LMPC) were used.

The target is modelled as a disk of radius $\rho = 2.4$ m, with a constant angular velocity $\omega = 0.6^\circ/\text{sec}$ relative to the local-vertical local-horizontal (LVLH) frame, and with a target docking port attached to the edge of the disk. The system state $x \in \mathbb{R}^5$ is 5 dimensional consisting of the relative position (p_x, p_y) and velocity (v_x, v_y) of the spacecraft in the LVLH frame as the first four states, and angle of the ISS with respect to the LVLH frame ψ as the last state.

Instead of using the linearised Clohessy-Wiltshire equations (as in [16]), we use the full nonlinear equations of relative motion. In this work, only gravitational forces due to the Earth and propulsive forces are modelled, but other nonlinear effects like solar radiation pressure or air resistance

can also be included. The dynamical model is then given by

$$\frac{d}{dt} \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \\ \psi \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ n^2 p_x + 2nv_y + \frac{\mu}{r^2} - \frac{\mu(r+p_x)}{r_c^3} \\ n^2 p_y - 2nv_x - \frac{\mu p_y}{r_c^3} \\ \omega \end{bmatrix} + \frac{1}{m_c} \begin{bmatrix} 0 \\ 0 \\ u_x \\ u_y \\ 0 \end{bmatrix} \quad (28)$$

where $r_c = \sqrt{x^2 + y^2}$ is the relative distance to the chaser, $r = 6771$ km is the radius of orbit of the ISS, $\mu = 398,600 \text{ km}^3/\text{s}^2$ is the gravitational parameter of Earth, $n = \sqrt{\mu/r^3}$ is the mean motion of the target satellite around the Earth, $\omega = 0.6^\circ/\text{s}$ is the angular velocity of the target relative to the LVLH frame, and $m_c = 1000$ kg is the mass of the chaser vehicle, assumed constant during the rendezvous. The control inputs (u_x, u_y) are the propulsive forces in the local vertical and local horizontal directions, respectively. We suppose that these forces are 1-norm bounded, $|u_x| + |u_y| \leq 0.25$ kN.

The Line-of-Sight constraint is also non-linear, and can be encoded exactly using θ , the line of sight angle,

$$\begin{aligned} h(x) &= \cos \theta - \cos \gamma \\ &= \frac{\vec{r}_{c-p} \cdot \hat{e}}{\|\vec{r}_{c-p}\|} - \cos(\gamma), \end{aligned}$$

where $\vec{r}_{c-p} = [(p_x - \rho \cos \psi), (p_y - \rho \sin \psi)]^T$ is the position vector of the chaser relative to the docking port, and $\hat{e} = [\cos \psi, \sin \psi]^T$ is the direction vector along the docking axis.

The desired behaviour of the spacecraft is encoded using the Control Lyapunov function:

$$V(x) = \left(v_x + \frac{p_x - \rho \cos \psi}{10} \right)^2 + \left(v_y + \frac{p_y - \rho \sin \psi}{10} \right)^2$$

which specifies that the desired velocity v_d is along negative \vec{r}_{c-p} .

To construct the ICCBF, again $N = 2$ was chosen. The following class- \mathcal{K} functions were used:

$$\alpha_0(h) = 0.25h, \quad \alpha_1(h) = 0.85h, \quad \alpha_2(h) = (0.05 + k)h$$

where $k > 0$ is a parameter we allow the Quadratic Program to minimize, as in [4]. Thus, the controller u^* is the solution to u in the following quadratic optimization problem

$$\begin{aligned} & \underset{u \in R^2; \delta, k \in R_+}{\text{argmin}} && \frac{1}{2}(u_x^2 + u_y^2) + 10\delta + 50k \\ & \text{subject to} && L_f V(x) + L_g V(x)u \leq -0.1V(x) + \delta \\ & && L_f b_2(x) + L_g b_2(x)u \geq -(0.05 + k)b_2(x) \\ & && |u_x| + |u_y| \leq 0.25 \end{aligned}$$

Figure 3(c-g) show simulation results of the rendezvous operation. The chaser is initialized at (100, -10) meters from the target spacecraft, and follows the trajectories drawn in (c-e), demonstrating a successful transfer. The 1-norm of the computed thrust force is indicated in (f), and (g) shows that the LOS constraint is satisfied at all times during the transfer. 3D animations, videos and source code for both case studies are available at [17].

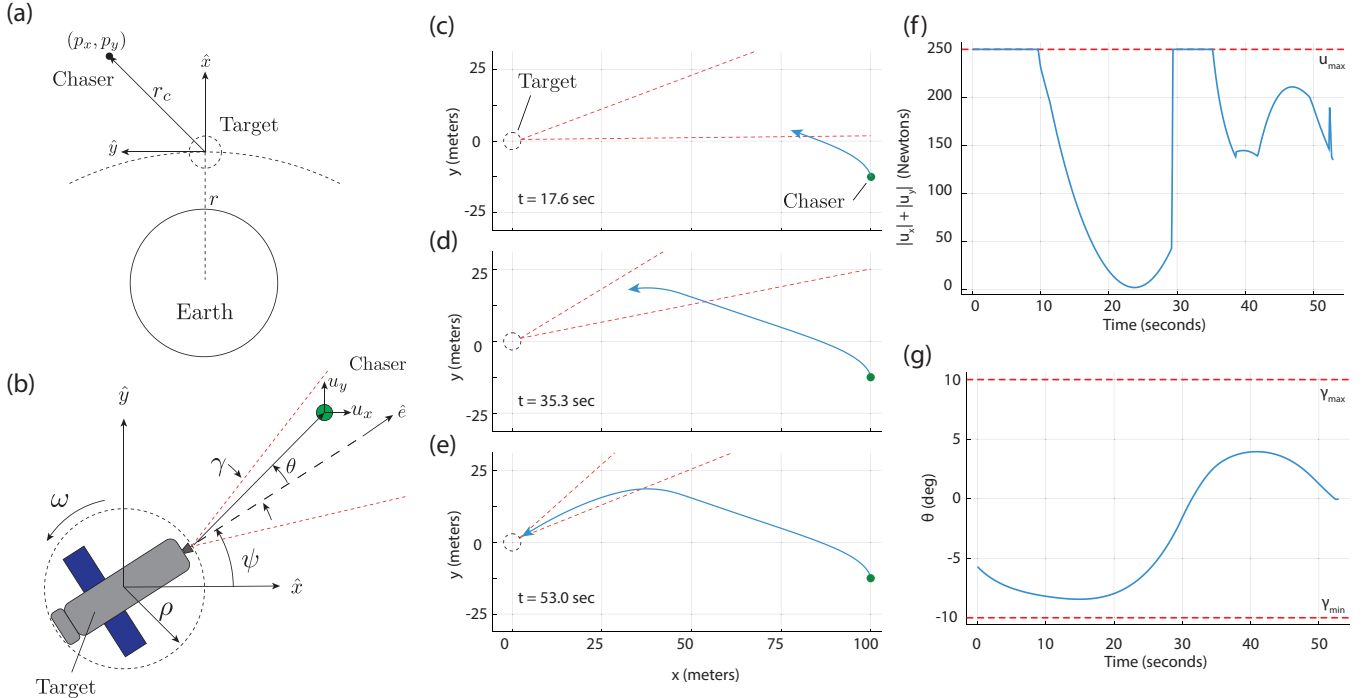


Fig. 3. (a, b) Schematic of the rendezvous problem. (a) represents the Local-Vertical Local-Horizontal frame. (b) details the target and chaser spacecrafts. The target spacecraft is rotating with constant angular velocity ω . The red dashed lines indicate the Line-of-Sight cone, which the chaser spacecraft must remain within. (c-e) show snapshots of the trajectory at three instances. The green dot represents the initial condition. (f) shows the 1-norm of the propulsive force and (g) indicates the line of sight angle θ .

V. CONCLUSION

In this paper, we have presented a framework that allows input constraints to be explicitly included in the construction of control barrier functions and to guarantee that safety is maintained with an input-constrained controller. The construction identifies an inner safe set and a feedback controller to render the subset safe. The method was demonstrated in two cases, an adaptive cruise control problem and a spacecraft rendezvous control problem. An optimization based method was used to verify the conditions of the ICCBF. Directions for future work include investigating numerically efficient methods to automate the search of ICCBFs, and to compare the complexity with other reachability methods, in particular for systems with high-dimensional states. Finally, the robustness of this controller to noise and model mismatch could also be investigated.

REFERENCES

- [1] 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)*, 2019, pp. 3420–3431.
- [2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [3] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCCPS)*, 2018, pp. 98–106.
- [4] K. Garg and D. Panagou, "Control-lyapunov and control-barrier functions based quadratic program for spatio-temporal specifications," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 1422–1429.
- [5] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [6] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.
- [7] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2018.
- [8] W. S. Cortez and D. V. Dimarogonas, "Correct-by-design control barrier functions for euler-lagrange systems with input constraints," in *2020 American Control Conference (ACC)*, 2020, pp. 950–955.
- [9] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 474–479.
- [10] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.
- [11] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
- [12] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015, analysis and Design of Hybrid Systems ADHS.
- [13] W. R. Inc., "Mathematica, Version 12.2," champaign, IL, 2020. [Online]. Available: <https://www.wolfram.com/mathematica>
- [14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.
- [15] J. Revels, M. Lubin, and T. Papamarkou, "Forward-mode automatic differentiation in Julia," *arXiv:1607.07892 [cs.MS]*, 2016. [Online]. Available: <https://arxiv.org/abs/1607.07892>

- [16] H. Park, S. Di Cairano, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 1922–1927.
- [17] Source code and animations are available at <https://github.com/dev10110/Input-Constrained-Control-Barrier-Functions>.