2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682938

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 nonlinear 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 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].

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, which we term the inner safe set. A few methods have been proposed to find the inner safe set. These include reachability analysis by solving a Hamilton-Jacobi equations [3], [5] and Sum-of-Squares (SOS), which employ the positivstellensatz theorem to provide a certificate of safety [6], [7]. Both methods scale poorly with the dimension of the state-space. Some methods have also been proposed for specific classes of systems, e.g. Euler-Lagrange systems [8] or mechanical systems in a manifold [9].

In this paper, we introduce the notion of an Input Constrained Control Barrier Function (ICCBF). We show that an ICCBF guarantees that an input constrained controller can render the super-level set of the ICCBF forward invariant. Furthermore, we show that ICCBFs represent a generalization of Higher Order CBFs (HOCBFs) [10], enabling synthesis of input-constrained controllers for safety functions of non-uniform relative degree. Finally, the method is applied to an adaptive cruise control problem [11], and a spacecraft rendezvous problem, demonstrating that ICCBFs define a safe controller that respects input constraints.

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

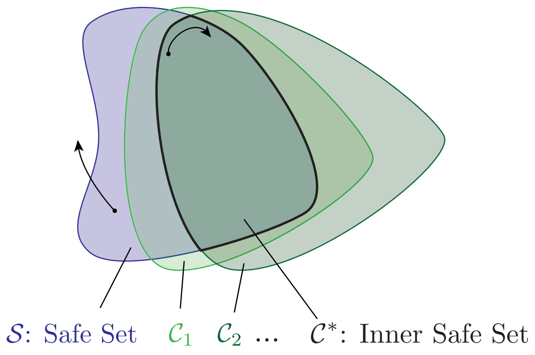

Fig. 1. Visual representation of ICCBF method. The safe set $\mathcal{S}$ and two intermediate sets $\mathcal{C}_1$ and $\mathcal{C}_2$ are drawn. The final inner safe set $\mathcal{C}^*$ is the intersection of each of these sets, and can be rendered forward invariant.

II. Problem Formulation and Preliminaries

Notation: We denote the set of real numbers $\mathbb{R}$ and non-negative reals $\mathbb{R}_+$. A continuous function $\alpha : [0, a) \to [0, \infty)$ is class-$\mathcal{K}$ if it is strictly increasing, and $\alpha(0) = 0$. The Lie derivative of $h(x)$ along $f(x)$ is denoted $L_f h(x) = \frac{dh}{dx} f(x)$. Int($\mathcal{C}$) and $\partial\mathcal{C}$ denote the interior and boundary of a set $\mathcal{C}$.

A. Problem Setup

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

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

where $f : \mathcal{X} \to \mathbb{R}^n$, $g : \mathcal{X} \to \mathbb{R}^{n\times m}$ are sufficiently smooth, as will be discussed in III-B. We assume these functions are known, and the system state is measured exactly. Under a Lipschitz continuous feedback law $u = \pi(x)$, the closed-loop system is

$$\dot{x} = f(x) + g(x)\pi(x). \tag{2}$$

We define a state x as *safe*, if it lies in a set $\mathcal{S}$, the 0-superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$:

$$\mathcal{S} \triangleq \{x \in \mathcal{X} : h(x) \geq 0\} \tag{3}$$

$$\partial\mathcal{S} \triangleq \{x \in \mathcal{X} : h(x) = 0\} \tag{4}$$

$$\text{Int}(\mathcal{S}) \triangleq \{x \in \mathcal{X} : h(x) > 0\} \tag{5}$$

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

Definition 1. *A set $\mathcal{S}$ is rendered* forward invariant *by a feedback controller $\pi : \mathcal{S} \to \mathcal{U}$, if for the closed-loop system* (2), *$x(0) \in \mathcal{S}$ implies $x(t) \in \mathcal{S}$ for all $t \geq 0$.*

978-1-6654-3659-5/21/$31.00 ©2021 IEEE

Due to input constraints however, there may not exist a controller which renders the safe set forward invariant (Example 1). We propose the definition of an *inner safe set*.

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 $\pi : \mathcal{C}^* \to \mathcal{U}$ such that $\mathcal{C}^*$ is rendered forward invariant by π.*

Example 1. Consider the following scalar dynamical system with input and safety constraints:

$$\dot{x} = x + u, \quad \mathcal{U} = [-1, 1], \quad \mathcal{S} = \{x \in \mathbb{R} : x \leq 2\}$$

i.e. $h(x) = 2 - x$. Now consider the boundary state $x = 2 \in \mathcal{S}$. Since

$$\dot{h} = -2 - u \implies \dot{h} \leq -1 \ \forall u \in \mathcal{U},$$

i.e., all closed-loop trajectories starting at $x(0) = 2$ leave the safe set. Thus $\mathcal{S}$ cannot be rendered forward invariant. The set $\mathcal{C}^* = \{x : x \leq 1\}$ is an inner safe set. △

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 $\pi : \mathcal{C}^* \to \mathcal{U}$, such that for any $x(0) \in \mathcal{C}^*$, the closed-loop trajectories of (2) 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.

B. Set Invariance

Nagumo's theorem provides a necessary and sufficient condition for the forward invariance of a set $\mathcal{S}$. In this work, Nagumo's theorem simplifies to:

Lemma 1. *Consider the system (1). Let the set $\mathcal{S}$ be defined by a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$, as per (3-5). Consider a Lipschitz continuous feedback controller $\pi : \mathcal{S} \to \mathcal{U}$, such that for any initial condition $x(0) \in \mathcal{S}$, the closed-loop system (2) admits a globally unique solution. Then set $\mathcal{S}$ is forward invariant if and only if*

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

In [1], [2], a stronger notion of the control barrier function is introduced:

Definition 3 (Control Barrier Function [1])**.** *Let $S \subset \mathcal{X} \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$. h is a* Control Barrier Function (CBF) *if there exists an extended class-$\mathcal{K}_\infty$ function α such that for the control system* (1)*:*

$$\sup_{u \in \mathcal{U}} [L_f h(x) + L_g h(x) u] \geq -\alpha(h(x)) \tag{7}$$

for all $x \in \mathcal{X}$.

Lemma 2 ([1], Theorem 2)**.** *Let $\mathcal{S} \subset \mathbb{R}^n$ be a set defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$. If h is a CBF on $\mathcal{X}$, and $dh/dx(x) \neq 0$ for all $x \in \partial\mathcal{C}$, then any Lipschitz continuous controller $\pi(x) \in K_{CBF}$, where*

$$K_{CBF}(x) = \{u \in \mathcal{U} : L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\}, \tag{8}$$

for the control system (2)*, renders the set $\mathcal{S}$ safe.*

In this paper, we focus on cases where $h(x)$ defining the safe set $\mathcal{S}$ is not a valid control barrier function.

III. Input Constrained Control Barrier Functions

In this section we define Input Constrained Control Barrier Functions (ICCBFs). To aid the reader, first the method is explained conceptually, and formal definitions are presented second.

A. Motivation

Suppose the safe set $\mathcal{S}$ associated with h cannot be rendered forward invariant by any feedback controller $\pi(x)$, since there exist some states where it would require $u \notin \mathcal{U}$ to render safe. We wish to remove these states from $\mathcal{S}$. We define a function $b_1 : \mathcal{X} \to \mathbb{R}$ and a set $\mathcal{C}_1$ (visualized in Figure 1) as follows

$$b_1(x) = \inf_{u \in \mathcal{U}} [L_f h(x) + L_g h(x) u + \alpha_0(h(x))] \tag{9}$$

$$\mathcal{C}_1 = \{x \in \mathcal{X} : b_1(x) \geq 0\} \tag{10}$$

where α_0 is some user specified class-$\mathcal{K}$ function. Since an infimum over $\mathcal{U}$ is taken, b_1 only depends on x, and not u.

The set $\mathcal{C}_1$ has a useful property: Suppose there exists a point $x \in \partial\mathcal{S}$ and $x \in \mathcal{C}_1$, i.e., $h(x) = 0$ and $b_1(x) \geq 0$. Then, from (9),

$$x \in \partial\mathcal{S} \cap \mathcal{C}_1 \implies \inf_{u \in \mathcal{U}} [L_f h(x) + L_g h(x) u] \geq 0 \tag{11}$$

$$\implies L_f h(x) + L_g h(x) u \geq 0, \ \forall u \in \mathcal{U}. \tag{12}$$

Thus the closed-loop trajectory cannot leave $\mathcal{S}$ through x. Notice that if there exists a Lipschitz continuous controller π which renders $\mathcal{C}_1$ forward invariant, it is immediate $\mathcal{S} \cap \mathcal{C}_1$ is also forward invariant: any $x(t)$ that reaches the boundary $\partial\mathcal{S}$ must lie in $\mathcal{C}_1$ (by assumption on π), and thus by (12), $x(t)$ also cannot leave $\mathcal{S}$.

The problem now is to find the controller that renders $\mathcal{C}_1$ forward invariant. If this cannot be done, the steps can be repeated: 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 that renders $\mathcal{C}_2$ forward invariant also renders $\mathcal{C}_1 \cap \mathcal{C}_2$ forward invariant, and therefore the set $\mathcal{C}^* = \mathcal{S} \cap \mathcal{C}_1 \cap \mathcal{C}_2$ is also forward invariant *by the same controller*. This idea is formalized 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

$h : \mathcal{X} \to \mathbb{R}$, as per (3-5). We define the following sequence of functions:

$$b_0(x) = h(x) \tag{13a}$$
$$b_1(x) = \inf_{u \in \mathcal{U}} [L_f b_0(x) + L_g b_0(x) u + \alpha_0(b_0(x))] \tag{13b}$$
$$\vdots$$
$$b_N(x) = \inf_{u \in \mathcal{U}} [L_f b_{N-1}(x) + L_g b_{N-1}(x) u + \alpha_{N-1}(b_{N-1}(x))], \tag{13c}$$

where each α_i is a 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. The time derivative $\dot{b}_i = L_f b_i(x) + L_g b_i(x) u$ is still affine in u. Next, we define a family of sets,

$$\mathcal{C}_0 = \{x \in \mathcal{X} : b_0(x) \geq 0\} = \mathcal{S} \tag{14a}$$
$$\mathcal{C}_1 = \{x \in \mathcal{X} : b_1(x) \geq 0\} \tag{14b}$$
$$\vdots$$
$$\mathcal{C}_N = \{x \in \mathcal{X} : b_N(x) \geq 0\}. \tag{14c}$$

The intersection of these sets is $\mathcal{C}^*$, assumed closed, non-empty and without isolated points:

$$\mathcal{C}^* = \mathcal{C}_0 \cap \mathcal{C}_1 \cap ... \cap \mathcal{C}_N. \tag{15}$$

Definition 4. *For the dynamical system* (2) *with safe set $\mathcal{S}$ and continuously differentiable class-$\mathcal{K}$ functions $\alpha_0, ..., \alpha_{N-1}$, if there exists a class-$\mathcal{K}$ function α_N such that*

$$\sup_{u \in \mathcal{U}} [L_f b_N(x) + L_g b_N(x) u + \alpha_N(b_N(x))] \geq 0 \ \forall x \in \mathcal{C}^*, \tag{16}$$

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

Note, this does not require b_N to be a CBF on $\mathcal{C}_N$. The definition only requires condition (16) to hold for $x \in \mathcal{C}^*$, a subset of $\mathcal{C}_N$.

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

$$K_{ICCBF}(x) = \{u \in \mathcal{U} : L_f b_N(x) + L_g b_N(x) u \geq -\alpha_N(b_N(x))\} \tag{17}$$

renders the set $\mathcal{C}^* \subseteq \mathcal{S}$ (15) forward invariant.

Proof. Since u is a Lipschitz continuous controller, the closed-loop system (2) 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 that

$$x \in \mathcal{C}^*, \pi(x) \in K_{\text{ICCBF}}(x) \text{ and } b_i(x) = 0 \implies \frac{db_i}{dx}[f(x) + g(x)\pi(x)] \geq 0, \tag{18}$$

We show (18) holds for each $i \in \mathcal{I}(x) = \{i : b_i(x) = 0\}$:

Cases $i \in \{0, ..., N-1\}$: 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, 14),

$$\inf_{u \in \mathcal{U}} [L_f b_i(x) + L_g b_i(x) u] \geq 0$$
$$\therefore L_f b_i(x) + L_g b_i(x) u \geq 0, \quad \forall u \in \mathcal{U} \tag{19}$$

and since $\pi(x) \in K_{\text{ICCBF}}(x) \subseteq \mathcal{U}$, (18) 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 (14c, 17),

$$L_f b_N(x) + L_g b_N(x) \pi(x) \geq 0 \ \forall \pi(x) \in K_{\text{ICCBF}}(x), \tag{20}$$

thus satisfying (18).

In conclusion, we have shown that condition (18) is satisfied for all $i \in \mathcal{I}(x)$, and therefore the conditions of Nagumo's theorem are satisfied, completing the proof. □

Remark 1. 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 some states from the set $\mathcal{S}$, and construct an inner set for which we can find a controller that renders it forward invariant.

Remark 2. A quadratic program based feedback controller can be used for polytopic input constraints, $\mathcal{U} = \{u : Pu \leq q\}$:

$$\begin{aligned} \pi(x) = \ & \text{argmin } u^T u \\ & \text{subject to } L_f b_N(x) + L_g b_N(x) u \geq -\alpha_N(b_N(x)) \\ & \qquad Pu \leq q, \end{aligned}$$

provided suitable regularity conditions hold, for instance $L_g b_N(x)$ is linearly independent of the rows of P [12], [13]. Note, this QP is always guaranteed to be feasible.

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

Definition 5. *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 $\pi : \mathcal{C}^* \to \mathcal{U}$ 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 $\mathcal{C}^* \cap \partial\mathcal{C}_N = \emptyset$, the null set, 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 (18). The remaining cases, with $i \in \{0, ..., N-1\}$ satisfy condition (18) for all $\pi(x) \in \mathcal{U}$. Therefore, any Lipschitz continuous $\pi : \mathcal{C}^* \to \mathcal{U}$ admits globally unique solutions and satisfies condition (18), completing the proof. □

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 $g(x)\pi(x)$ 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 initialized within $\mathcal{C}^*$.

Remark 3. Higher Order CBFs, as in [10], 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

$$\begin{aligned} b_1(x) &= \inf_{u\in\mathcal{U}} [L_f h(x) + L_g h(x) u + \alpha_0(h(x))] \\ &= \inf_{u\in\mathcal{U}} [L_f h(x) + \alpha_0(h(x))] \\ &= L_f h(x) + \alpha_0(h(x)) \qquad (21) \end{aligned}$$

which is exactly the function defined in [10]. This repeats for higher relative degrees. 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 $\mathcal{S}$.

Remark 4. The search (over integers N and class-$\mathcal{K}$ functions α_i) and validation for ICCBFs (i.e., verifying (16)) can be complicated, as is the case with Lyapunov functions in general. For practical implementation, we can solve the following optimization problem:

$$\gamma = \underset{x\in\mathcal{C}^*}{\text{minimize}} \quad \sup_{u\in\mathcal{U}} [\dot{b}_N(x,u) + \alpha_N(b_N(x))] \qquad (22)$$

By the definition, b_N is an ICCBF if and only if the optimization problem is feasible, with solution $\gamma \geq 0$. Since this optimization is nonlinear, unless a guaranteed global optimizer is used, this can only be used to invalidate b_N as a ICCBF. For our experiments, we manually checked a few (approx. 6) N and α_i until $\gamma \geq 0$. Whether a finite N exists for a given dynamical system such that b_N is an ICCBF remains an open question.

IV. Simulations

A. Adaptive Cruise Control

As a demonstration, we apply ICCBFs to the Adaptive Cruise Control (ACC) problem of [11]. Consider a point-mass model of a vehicle moving in a straight line. The vehicle is following a vehicle d distance in-front, moving at a known constant speed v_0. The objective is to design a controller to accelerate to the speed limit but prevent the vehicles from colliding.

As in [11], the safety constraint is specified as $d \geq 1.8v$. Defining the state $x = [d, v]^T$, the dynamical model is

$$\begin{bmatrix} \dot{d} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v_0 - v \\ -F(v)/m \end{bmatrix} + \begin{bmatrix} 0 \\ g_0 \end{bmatrix} u, \ \mathcal{U} = \{u : |u| \leq 0.25\}$$

where u is the control input, $F(v) = f_0 + f_1 v + f_2 v^2$ models resistive forces on the vehicle, m is the mass of the vehicle, g_0 is acceleration due to the gravity. The safe set $\mathcal{S}$ is

$$\mathcal{S} = \{x \in \mathcal{X} : h(x) = x_1 - 1.8x_2 \geq 0\}$$

and we can verify that $\mathcal{S}$ is not forward invariant under the input constraints. Thus, h is not a CBF, 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 $\mathcal{C}_1, \mathcal{C}_2$. To (approximately) verify that b_2 is an ICCBF, the optimization (22) was used, and $\gamma = 2.33$ was found.

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. Where there exists a feasible control input to keep trajectories within the set, the line is solid, and where no feasible control input will keep trajectories within the set, the line is dashed. $\mathcal{C}^*$, the intersection of $\mathcal{S}, \mathcal{C}_1, \mathcal{C}_2$, is visualized in Figure 2(d). The following controller is used:

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

where $\pi_d(x)$ 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, $\pi_d(x)$:

$$L_f V(x) + L_g V(x) \pi_d(x) = -10V(x)$$

We compare our controller to the CLF-CBF-QP [11]:

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

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). 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, by explicitly accounting for input constraints ICCBF-QP controller keeps the input-constrained system safe, where the CLF-CBF-QP doesn't.

B. Autonomous Rendezvous

In this section, the ICCBF method is applied to an autonomous rendezvous operation (adapted from [14]) between a chaser spacecraft modelled as a point mass, and a target body, e.g. the International Space Station (ISS) (Figure 3). The target is modelled as a point on a disk of radius $\rho = 2.4$ m rotating with a constant angular velocity $\omega = 0.6°$/sec 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 $\gamma = 10°$ cone of the docking axis. The system state $x \in \mathbb{R}^5$ is the relative position (p_x, p_y) and velocity (v_x, v_y) and angle of the docking port ψ. Instead

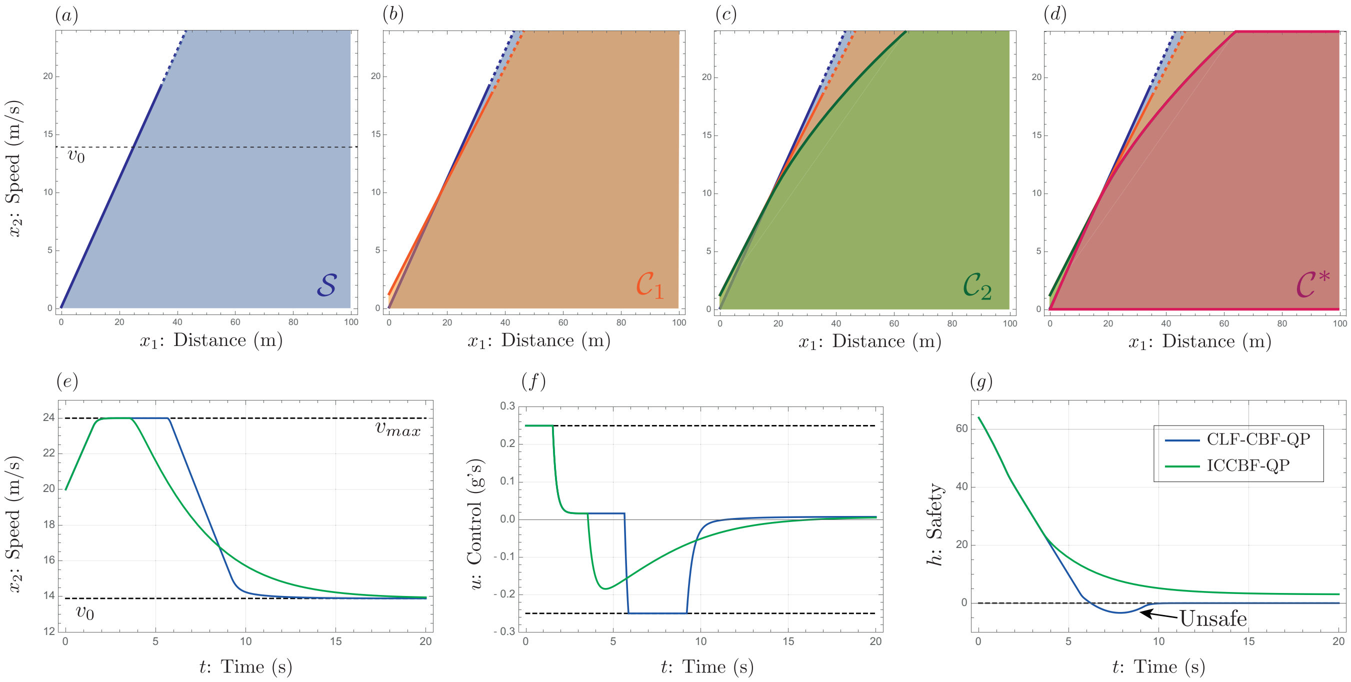

Fig. 2. Figures (a-d): State-space diagrams indicating the sets (a) $\mathcal{S}$, (b) $\mathcal{C}_1$, (c) $\mathcal{C}_2$ and (d) $\mathcal{C}^*$. The horizontal dashed line in (a) indicates v_0, the speed of the car in-front. Figure (d) represents the inner safe set $\mathcal{C}^*$ that is rendered forward invariant. Figures (e-g): Simulation results for speed, control input and safety under the CLF-CBF-QP controller [11] and the ICCBF-QP.

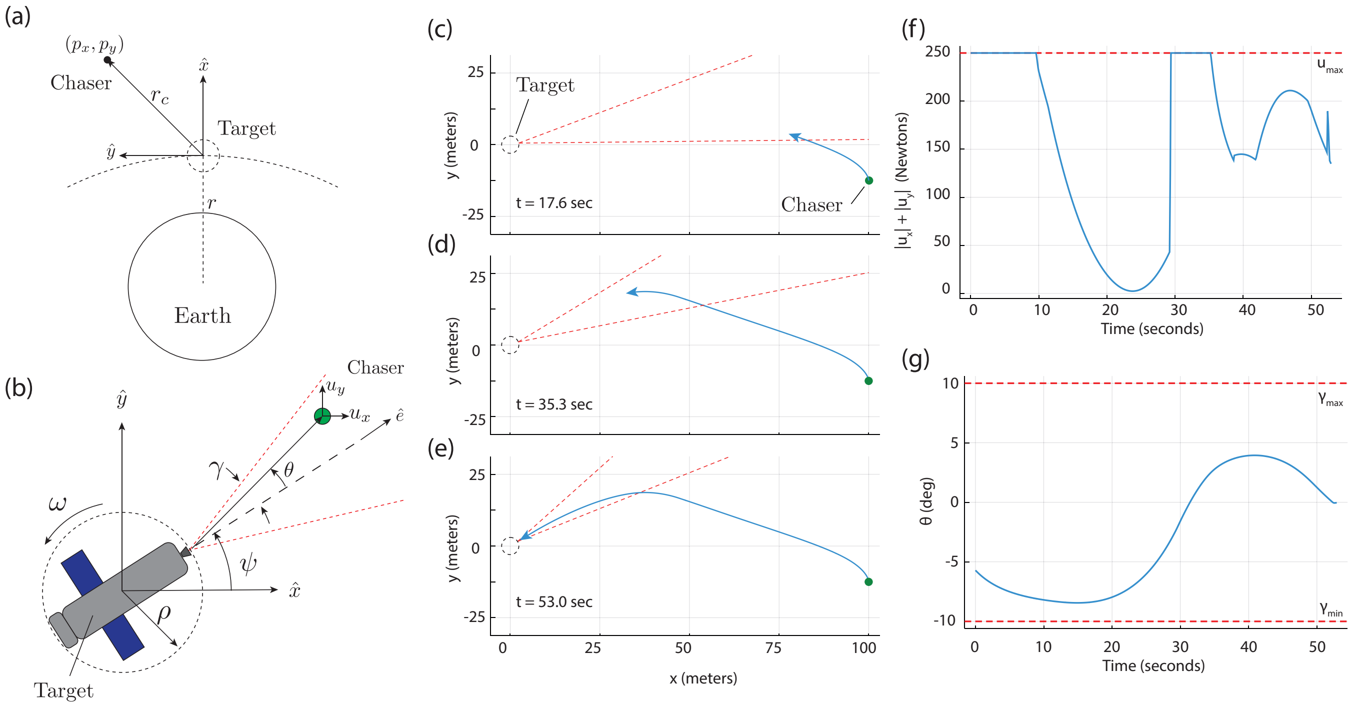

Fig. 3. (a, b) Schematic of the rendezvous problem. (a) represents the Local-Vertical Local-Horizontal frame. (b) details the target and chaser spacecrafts. The target spacecraft is rotating with constant angular velocity ω. The red dashed lines indicate the Line-of-Sight cone, which the chaser spacecraft 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.

of using the (linearized) Clohessy-Wiltshire equations (as in [14]), we use the exact equations of relative motion[1]:

$$\frac{d}{dt}\begin{bmatrix}p_x\\p_y\\v_x\\v_y\\\psi\end{bmatrix} = \begin{bmatrix} v_x\\ v_y\\ n^2p_x+2nv_y+\frac{\mu}{r^2}-\frac{\mu(r+p_x)}{r_c^3}\\ n^2p_y-2nv_x-\frac{\mu p_y}{r_c^3}\\ \omega\end{bmatrix}+\frac{1}{m_c}\begin{bmatrix}0\\0\\u_x\\u_y\\0\end{bmatrix} \tag{23}$$

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, $\mu = 398,600$ km^3/s^2 is the gravitational parameter of Earth, $n = \sqrt{\mu/r^3}$ is the mean motion of the target satellite around the Earth, $\omega = 0.6°$/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. Suppose the forces are 1-norm bounded, $|u_x|+|u_y|\leq 0.25$ kN. The LOS constraint is $h(x) \geq 0$, where

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

and $\vec{r}_{c-p} = [(p_x - \rho\cos\psi), (p_y - \rho\sin\psi)]^T$ is the position vector of the chaser relative to the docking port, and $\hat{e} = [\cos\psi, \sin\psi]^T$ is the docking axis vector. We use a CLF to guide the chase to the the docking port:

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

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], and verified approximately. Thus, the controller u^* is the solution to u in the following quadratic optimization problem

$$\begin{aligned} \underset{u\in\mathbb{R}^2;\delta,k\in\mathbb{R}_+}{\text{argmin}} \quad & \tfrac{1}{2}(u_x^2+u_y^2) + 10\delta + 50k \\ \text{subject to} \quad & L_fV(x) + L_gV(x)u \leq -0.1V(x) + \delta \\ & L_fb_2(x) + L_gb_2(x)u \geq -(0.05+k)b_2(x) \\ & |u_x|+|u_y|\leq 0.25 \end{aligned}$$

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 [15].

[1] In this work, only gravitational forces due to the Earth and propulsive forces are modelled, but other non-linear effects like solar radiation pressure or air resistance can also be included.

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. We demonstrated the method on an adaptive cruise control problem and a spacecraft rendezvous 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 (ICCPS)*, 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] 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 *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2038–2044.

[10] 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.

[11] 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.

[12] W. W. Hager, "Lipschitz continuity for constrained processes," *SIAM Journal on Control and Optimization*, vol. 17, no. 3, pp. 321–338, 1979.

[13] B. Morris, M. J. Powell, and A. D. Ames, "Sufficient conditions for the lipschitz continuity of qp-based multi-objective control of humanoid robots," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 2920–2926.

[14] 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.

[15] Source code and animations are available at https://github.com/dev10110/Input-Constrained-Control-Barrier-Functions.