

Control Barrier Functions for Sampled-Data Systems with Input Delays

Andrew Singletary, Yuxiao Chen, and Aaron D. Ames

Abstract—This paper considers the general problem of transitioning theoretically safe controllers to hardware. Concretely, we explore the application of control barrier functions (CBFs) to sampled-data systems: systems that evolve continuously but whose control actions are computed in discrete time-steps. While this model formulation is less commonly used than its continuous counterpart, it more accurately models the reality of most control systems in practice, making the safety guarantees more impactful. In this context, we prove robust set invariance with respect to zero-order hold controllers as well as state uncertainty, without the need to explicitly compute any control invariant sets. It is then shown that this formulation can be exploited to address input delays in this system, with the result being CBF constraints that are affine in the input. The results are demonstrated in a high-fidelity simulation of an unstable Segway robotic system in real-time.

I. INTRODUCTION

Control theory in practice is almost always implemented in the form of a digital controller on a physical system that evolves continuously. However, these systems are rarely treated as such due to the difficulties that arise in the formulation of controllers that act optimally for these types of systems. Generally, the system is controlled rapidly enough that the time discretization can be ignored, and the entire system can be treated as continuous, allowing for a much larger class of control techniques. This is especially true for robotic systems, where continuous controllers are often implemented at loop rates faster than 1 kHz.

In the context of safety-critical control, however, it is important to model the system as accurately as possible, in order to extend the guarantees from theory to practice. Moreover, optimization-based controllers tend to be less robust than simple control laws such as PID, therefore call for a more accurate model. As an example, control barrier functions have emerged as a popular technique for guaranteeing safety of continuous-time [1], [2], discrete-time [3], [4], and even sampled-data systems [5]. This technique relies on the knowledge of a control invariant subset of the state-space, and can be formulated as a quadratic program wherein the CBF yields a constraint, affine in the input, that ensures the system stays in that set for all time. However, the computation of such sets is notoriously difficult for nonlinear systems [6], [7].

The method proposed in [8] removes the need for a control invariant set, and instead relies on knowledge of a backup

controller that takes the system to a small safe region of the state space. This method of implicitly defining a control barrier function has been successfully implemented in a variety of complex applications [9]–[11], but it suffers from a lack of robustness.

Another reality of control theory in practice is the presence of input delay. Input delay for myopic, optimization-based controllers (such as Control Lyapunov Functions [12] and control barrier functions), is often handled by making the problem formulation robust to any value of the input delay in some bounded set. However, this treatment degrades performance due to conservatism and complicates the (already difficult) computation of the Lyapunov or barrier functions. In practice, the input delay for a system is relatively easy to identify, so it would be beneficial to formulate the problem with the knowledge of the input delay of the system. While many solutions to handling specific time delays have been proposed, in general they either require either linear systems [13], [14], are applicable only to autonomous systems [15], require difficult construction [16], or rely on frequency-domain analysis that is not applicable to these optimization-based controllers [17].

The contributions of this paper are:

- We propose a formulation of the implicitly defined control barrier function that is applicable to sampled-data systems, while retaining scalability.
- We use the concept of incremental stability [18] to prove robustness of the proposed backup controller-based CBF controller under state uncertainty.
- We are able to guarantee safety under a known input delay with much less conservatism under the proposed framework.

For the remainder of the paper, Section II details the conditions required for the control barrier function to provide safety guarantees for the sampled-data system that arises from a zero-order hold controller. The resulting constraint must be evaluated over a set of states, thus Section III provides a means for tractably evaluating this constraint. Section IV provides a method for handling known input delays, while maintaining the safety guarantee in the form of an affine constraint. The proposed method is then applied in Section V to a high-fidelity Segway simulation, which demonstrates the result. Finally, Section VI provides a summary of the work and details future research directions.

Andrew Singletary, Yuxiao Chen, and Aaron D. Ames are with Department of Mechanical Engineering, California Institute of Technology, Pasadena CA 91125, U.S.A. Email addresses: {asinglet, chenyx, ames}@caltech.edu

II. SAMPLED-DATA CONSIDERATIONS FOR CBFs

A. Background on Control Barrier Functions

In this paper, we consider an affine dynamical system described by the following ordinary differential equation:

$$\dot{x} = f(x) + g(x)u, \quad x \in \mathcal{X} \subseteq \mathbb{R}^n, \quad u \in \mathcal{U} \subseteq \mathbb{R}^m. \quad (1)$$

Here, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz functions. \mathcal{U} is the set of admissible inputs, assumed to be a strict subset of \mathbb{R}^m . For a given dynamic system, given a feedback policy $u: \mathcal{X} \rightarrow \mathcal{U}$, $\phi_t^u(x_0)$ be the state evolution of this closed loop system with initial condition x_0 at time t , i.e., the flow map. With a slight abuse of terminology, we let $\phi_t^{u(\cdot)}$ denote the state flow under a fixed input signal $u(\cdot)$, as opposed to a fixed feedback policy.

Suppose that the system is required to stay within the set described by the function $\mathcal{S} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$ for all time. To guarantee this constraint is satisfied for all time, one way is to find a control invariant set $\mathcal{S}_I \subseteq \mathcal{S}$.

Definition 1. A set \mathcal{S}_I is a control invariant set if there exists a control policy that keeps any trajectory starting within \mathcal{S}_I inside \mathcal{S}_I for all time.

If a control invariant subset described as the superlevel-set (see [1]) of a function $h_I(x)$ is known, the CBF condition can be enforced to keep the state within \mathcal{S}_I :

$$\left. \frac{dh_I}{dx} \right|_x (f(x) + g(x)u) + \alpha(h_I(x)) \geq 0, \quad (2)$$

as an affine constraint on the input u . Here, $\alpha(\cdot)$ is an extended class- \mathcal{K} function.

Lemma 1. *If the input satisfies the CBF condition (2) for all time, the system will stay in the set $h_I(x)$ for all time. Specifically, let $u(x)$ be a controller that satisfies (2) and is applied to (1) to yield a closed-loop dynamical system: $\dot{x} = f(x) + g(x)u(x) =: f_{cl}(x)$ (assumed to be forward complete). Then if $x_0 \in \mathcal{S}_I$, $\phi_t^u(x_0) \in \mathcal{S}_I$ for all $t \geq 0$.*

See [1] for proof.

As mentioned in the introduction, the computation of control invariant sets is very difficult in general. Instead, [8] proposes utilizing a fixed backup controller $u_B(x)$ to enlarge a much smaller control invariant set $\mathcal{S}_B = \{x \in \mathbb{R}^n \mid h_B(x) \geq 0\}$ (which is much easier to compute) to the set of all states that can be safely brought to \mathcal{S}_B with the backup controller $u_B: \mathcal{X} \rightarrow \mathcal{U}$. This leads to a control invariant set defined implicitly using the system flow $\phi_t^{u_B}(x_0)$ under u_B as

$$\mathcal{S}_I = \left\{ x \in \mathbb{R}^n \mid \left(\forall \tau \in [0, T], h\left(\phi_\tau^{u_B}\right) \geq 0 \right) \text{ and } \left(h_B\left(\phi_T^{u_B}\right) \geq 0 \right) \right\} \quad (3)$$

The CBF condition (2) then is evaluated at a point x_0 as

$$\begin{aligned} \left. \frac{dh}{dx} \right|_{\phi_\tau^{u_B}} \left. \frac{\partial \phi_\tau^{u_B}}{\partial x} \right|_{x_0} (f(x_0) + g(x_0)u) + \alpha(h(\phi_\tau^{u_B}(x_0))) &\geq 0 \\ \left. \frac{dh_B}{dx} \right|_{\phi_T^{u_B}} \left. \frac{\partial \phi_T^{u_B}}{\partial x} \right|_{x_0} (f(x_0) + g(x_0)u) + \alpha(h_B(\phi_T^{u_B}(x_0))) &\geq 0 \end{aligned} \quad (4)$$

Any input that satisfies (4) for all time $\tau \in [0, T]$ will keep the system in the set \mathcal{S}_I . If this constraint is satisfied for all time, the state is kept in the set \mathcal{S}_I indefinitely. See [8, p. 6].

B. CBFs Along a Zero-Order Hold Backup Controller

While condition (4) can be used to guarantee safety for continuous-time systems, it relies on the input being computed and applied continuously. In the presence of a zero-order hold controller, the description of the invariant set need to be modified. The zero-order hold backup controller, denoted as $\overline{u}_B(x(\cdot), t)$, simply takes on the value of $u_B(x)$ every Δ_t seconds, and holds that value until its next update:

$$\overline{u}_B(x(\cdot), t) = u_B(x(\lfloor t/\Delta_t \rfloor \Delta_t)), \quad (5)$$

where $\lfloor \cdot \rfloor$ is the largest integer not greater than the argument.

Remark 1. The flow of the system under the zero-order backup controller $\phi_t^{\overline{u}_B}$ is well-defined, as unique solutions to sampled-data systems exist so long as the underlying controller is piecewise continuous (c.f. [19, p. 16]).

The updated control invariant set under the zero-order hold backup controller can be written as

$$\overline{\mathcal{S}}_I = \left\{ x \in \mathbb{R}^n \mid \left(\forall \tau \in [0, T], h\left(\phi_\tau^{\overline{u}_B}\right) \geq 0 \right) \text{ and } \left(h_B\left(\phi_T^{\overline{u}_B}\right) \geq 0 \right) \right\} \quad (6)$$

This is a control invariant set for the system under a zero-order controller with the same sampling time Δ_t as \overline{u}_B . The proof follows from [8, Theorem 1].

Remark 2. While the flow of the system under the zero-order hold backup controller is Lipschitz [20], it is nonsmooth. Because of this, the barrier function itself is nonsmooth, and thus \dot{h} cannot be expressed at finitely many points, which correspond to when the controller is updated. Despite this, a nonsmooth barrier function is valid if $\dot{h} \geq -\alpha(h)$ almost everywhere. For proof, see [21, Lemma 2.2].

To enforce the CBF condition, \dot{h} needs to be computed, which requires $\frac{\partial \phi_\tau^{\overline{u}_B}}{\partial x}$ to be evaluated. This expression is continuous over each controller sampling time, and can be computed using finite-differences [22]. To do this, simply integrate forward $n+1$ initial conditions under \overline{u}_B to evaluate $\frac{\partial \phi_{(i+1)\Delta_t}^{u_B}(x)}{\partial \phi_{i\Delta_t}^{u_B}(x)}$ at each time-step. Then, use the chain rule to get

$$\frac{\partial \phi_{i\Delta_t}^{u_B}(x)}{\partial x} = \prod_{n=0}^{i-1} \frac{\partial \phi_{(n+1)\Delta_t}^{u_B}(x)}{\partial \phi_{n\Delta_t}^{u_B}(x)} \quad (7)$$

C. Enforcing the Barrier Condition with Zero-order Hold

The last caveat to consider in the implementation of the CBF condition is the fact that the condition must be met over the entire time horizon of the zero-order hold controller. Note that this consideration must be taken regardless of the method used for expressing the robust control barrier function, and was a subject of prior research of the authors [5].

Consider verifying the barrier function over the horizon of a single time-step of the zero-order hold controller with sample time Δ_t ,

$$h(\phi_\tau^{\bar{u}_B}(x_0)) \geq 0 \quad \forall \tau \in [0, \Delta_t]. \quad (8)$$

The robust satisfaction of the above condition can be verified by checking the stronger condition shown in [5],

$$h(\mathcal{R}(x_0, \Delta_t)) \geq 0, \quad (9)$$

where $\mathcal{R}(x_0, \Delta_t)$ is the set of states reachable from x_0 in time Δ_t with any input $u \in \mathcal{U}$.

Remark 3. This condition adds conservatism to the barrier formulation. While checking only the points $\phi_\tau^u(x_0) \forall \tau \in [0, \Delta_t]$ would result in a more performant condition, this would make the constraint no longer affine, due to its dependence on the decision variable u .

Let $\bar{u}_B(\cdot)$ be the input signal w.r.t. the nominal state flow, the robust CBF condition defined from the set $\bar{\mathcal{S}}_I$,

$$\begin{aligned} \frac{dh}{dx} \Big|_{\phi_\tau^{\bar{u}_B}(x_0)} \frac{\partial \phi_\tau^{\bar{u}_B}(x_0)}{\partial x} (f(x_0) + g(x_0)u) + \alpha(h(\phi_\tau^{\bar{u}_B}(x_0))) &\geq 0 \\ \frac{dh_B}{dx} \Big|_{\phi_\tau^{\bar{u}_B}(x_0)} \frac{\partial \phi_\tau^{\bar{u}_B}(x_0)}{\partial x} (f(x_0) + g(x_0)u) + \alpha(h_B(\phi_\tau^{\bar{u}_B}(x_0))) &\geq 0 \end{aligned} \quad (10)$$

where $x_0 = \mathcal{R}(x_0, \Delta_t)$, and the first inequality must hold for all $\tau \in [0, T]$.

Proposition 1. Let $(\bar{u}_i)_{i=0}^\infty$ be a sequence of inputs that satisfies (10) at the beginning of each time-step and is applied with zero-order hold to the system (1). If $x_0 \in \bar{\mathcal{S}}_I$, then $\phi_t^{\bar{u}_i}(x_0) \in \bar{\mathcal{S}}_I$ for all $t \geq 0$.

Proof. Consider any time interval of a single time-step $\mathcal{T} = [t_0, t_0 + \Delta_t]$, and assume $x_{t_0} \in \bar{\mathcal{S}}_I$.

Denote $\Phi := \{\bigcup_{t \in \mathcal{T}} \phi_t^{\bar{u}_i}(x_{t_0})\}$. Since $x_{t_0} = x_{t_0} + \mathcal{R}(x_{t_0}, \Delta_t)$ covers all states reachable from time t_0 , $\Phi \subset \phi_t^{\bar{u}_i}(x_{t_0})$. Therefore, if \bar{u}_i satisfies (10) for x_{t_0} , then the CBF condition holds for Φ as well. Thus, by the invariance of $\bar{\mathcal{S}}_I$ and Lemma 1, $\phi_t^{\bar{u}_i}(x_{t_0}) \in \bar{\mathcal{S}}_I$ for all $t \in \mathcal{T}$.

Since this condition is met over the entire sequence of time-steps, $\phi_t^{\bar{u}_i}(x_0) \in \bar{\mathcal{S}}_I$ for all $t \geq 0$. \square

Remark 4. It is possible that, for some $x_0 \in \mathcal{S}_I$, that $x_0 \in \mathcal{S}_I$ but $x_0 \notin \bar{\mathcal{S}}_I$. In this case, the system is inside of its control invariant set, but the CBF condition (10) cannot be satisfied. This is due to conservatism mentioned in Remark 3. However, when this occurs, the backup control action can be taken. Thus, the system will stay safe for all time. Furthermore, this occurs on a very small set at the boundary of \mathcal{S}_I , which the strengthening term $\alpha(\cdot)$ makes difficult to reach.

Note that the condition is evaluated here over a set, rather than a single point. The evaluation of $\phi_\tau^{\bar{u}_B(\cdot)}(x_0)$ poses the most difficulty, as it involves robustly integrating over a set. This makes techniques like interval arithmetic [23], [24] difficult to implement, due to numerical issues.

III. STATE UNCERTAINTY

In this section, the concept of incremental stability will be used to show that for any $\tau \in [0, T]$, a fixed control signal $u(\cdot)$ and an uncertainty set Δ_x , $\phi_\tau^{u(\cdot)}(x_0 + \Delta_x) \subseteq \phi_\tau^{u(\cdot)}(x_0) + \Delta_x$. Thus, the CBF condition (10) can be evaluated over the entire set as an affine condition without requiring robust integration over sets.

A. Incremental stability with Lyapunov functions

To show that safety can be guaranteed for a small neighborhood of initial conditions, we adopt the concept of incremental stability (c.f. [18]).

Definition 2. Given the dynamic system in (1), the system is incrementally stable inside a set $\mathcal{X} \subseteq \mathbb{R}^n$ if $\forall T \geq 0$, $\forall x_1, x_2 \in \mathcal{X}$ and $u(\cdot) : [0, T] \rightarrow \mathbb{R}^m$ such that $\phi_t^{u(\cdot)}(x_1)$ and $\phi_t^{u(\cdot)}(x_2)$ stay inside \mathcal{X} , the evolution of the state satisfies $\|\phi_t^{u(\cdot)}(x_1) - \phi_t^{u(\cdot)}(x_2)\| \leq \beta(\|x_1 - x_2\|, t)$, where $\beta : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ is nonincreasing in t and $\forall t \in [0, T]$, $\beta(\cdot, t)$ is a class- \mathcal{K} function.

Proposition 2. Suppose there exists a Lyapunov function $V : \mathcal{X} \rightarrow \mathbb{R}$ that satisfies $c_1\|x\| \leq V(x) \leq c_2\|x\|$ for some $c_2 \geq c_1 > 0$. For two initial conditions $x_1, x_2 \in \mathcal{X}$ and an input signal $u(\cdot)$ such that $\phi^{u(\cdot)}(x_1), \phi^{u(\cdot)}(x_2)$, and $\phi^{u(\cdot)}(x_1) - \phi^{u(\cdot)}(x_2) \in \mathcal{X}$, let $V(t) = V(\phi_t^{u(\cdot)}(x_1) - \phi_t^{u(\cdot)}(x_2))$. If $\dot{V} \leq 0$, then the system is locally incrementally stable in \mathcal{X} .

Proof. The proof follows from the fact that $V(\cdot)$ and $\|\cdot\|$ are equivalent norms. \square

In the context of control barrier functions with a backup strategy, if the system is incrementally stable, then given a nominal initial condition x_0 and an uncertainty set characterized as a level-set of the Lyapunov function, $\forall x \in \{x | V(x - x_0) \leq \varepsilon\}$, for any input signal $u(\cdot)$, $\phi_t^{u(\cdot)}(x) \in \{x | V(x - \phi_t^{u(\cdot)}(x_0)) \leq \varepsilon\}$. We shall show in Section III-C how this result can simplify the robust CBF condition in (10), which requires that the CBF condition hold for a small set $\mathcal{R}(x_0, \Delta_t)$ around the nominal initial condition.

B. Gaining incremental stability via pre-feedback

The Segway model that considered in Section V is not incrementally stable, but pre-feedback can be used to make it so. Since the error dynamics are being considered, the nonlinear dynamics are linearized to simplify the analysis. Given a set $\mathcal{X} \subseteq \mathbb{R}^n$ of states, multiple linear dynamics models $\dot{x} = A_i x + B_i u$, $i = 1, \dots, N$ can be obtained by considering the extreme points of \mathcal{X} . Given a quadratic Lyapunov function $V = x^T P x$ where P is symmetric and positive definite, and an input set hyperbox defined as $\mathcal{U} = \{-u^{\max} \leq u \leq u^{\max}\} \subseteq \mathbb{R}^m$, we develop the following Linear Matrix Inequality (LMI) to

search for a pre-feedback gain that guarantees incremental stability for the system:

$$\begin{aligned} \min_{K \in \mathbb{R}^{n \times m}} \quad & \|\Lambda P^{-\frac{1}{2}} K^\top\|_\infty \\ \text{s.t.} \quad & \forall i = 1, \dots, N, P(A_i + B_i K) + (A_i + B_i K)^\top P \leq 0, \end{aligned} \quad (11)$$

where $\Lambda = \text{diag}(\frac{1}{u_1^{\max}}, \dots, \frac{1}{u_m^{\max}})$.

The cost function is chosen due to the fact that

$$\left\{ \max_x |K_i x| \text{ s.t. } x^\top P x \leq 1 \right\} = \sqrt{K_i P^{-1} K_i^\top}, \quad (12)$$

which means that the pre-feedback is available within the level set $\{x | x^\top P x \leq \min_{i=1, \dots, N} \frac{u_i^{\max}}{\sqrt{K_i P^{-1} K_i^\top}}\}$. Therefore, minimizing the cost function in (11) is maximizing the size of the level-set of the Lyapunov function in which the pre-feedback is available.

Proposition 3. *Given a dynamic system as described in (1) with $\mathcal{U} = \{-u^{\max} \leq u \leq u^{\max}\}$, a set $\mathcal{X} \subseteq \mathbb{R}^n$, and a Lyapunov function $V(x) = x^\top P x$, $P \geq 0$, assume that $\forall x_1, x_2 \in \mathcal{X}$, $\forall u \in \mathcal{U}$, $f(x_1) + g(x_1)u - f(x_2) - g(x_2)u \in \text{Conv}(A_i)(x_1 - x_2)$. Then, with a K solved with (12), the system with pre-feedback $\dot{x} = f(x) + g(x)(u + Kx)$ is incrementally stable within $\mathcal{X} \cap \{x | x^\top P x \leq \min_{i=1, \dots, N} \frac{u_i^{\max}}{\sqrt{K_i P^{-1} K_i^\top}}\}$.*

Proof. Since the A and B matrix enters linearly into the Lyapunov condition in (11), by the assumption that $f(x_1) + g(x_1)u - f(x_2) - g(x_2)u \in \text{Conv}(A_i)(x_1 - x_2)$, convexity shows that $\dot{V}(x_1 - x_2) \leq 0$, which shows incremental stability. \square

C. Generalizing State Uncertainty

The CBF condition shown in Equation (10) is shown for a specific uncertainty set $\mathbf{x}_0 = R(x_0, \Delta_t)$ that arises from the sampled-data nature of the system. For an incrementally stable dynamic system, the CBF condition is rewritten as

$$\begin{aligned} \frac{dh}{dx} \Big|_{\phi_\tau^{\bar{u}_B^{(\cdot)}}(x_0) \Big|_{\mathbf{x}_0}} \frac{\partial \phi_\tau^{\bar{u}_B^{(\cdot)}}}{\partial x} \Big|_{\mathbf{x}_0} (f(\mathbf{x}_0) + g(\mathbf{x}_0)u) + \alpha(h(\phi_\tau^{\bar{u}_B^{(\cdot)}}(x) \Big|_{\mathbf{x}_0})) \\ \frac{dh_B}{dx} \Big|_{\phi_T^{\bar{u}_B^{(\cdot)}}(x_0) \Big|_{\mathbf{x}_0}} \frac{\partial \phi_T^{\bar{u}_B^{(\cdot)}}}{\partial x} \Big|_{\mathbf{x}_0} (f(\mathbf{x}_0) + g(\mathbf{x}_0)u) + \alpha(h_B(\phi_T^{\bar{u}_B^{(\cdot)}}(x) \Big|_{\mathbf{x}_0})) \end{aligned} \quad (13)$$

The new set in which the constraint is being evaluated is $\mathbf{x}_0 := \mathcal{R}(x_0, \Delta_t) + \Delta_x$, where $\Delta_x \subset \mathbb{R}^n$ is the state uncertainty set such that the estimated value of the state $\tilde{x} \in x + \Delta_x$, with x being the true state. The other major difference from Equation (10) is that the flow over the backup trajectory is now being computed for the nominal value of x_0 , and it is simply being evaluated over the set $\phi^\mu(x_0) + \Delta_x$. This greatly simplifies the computation, and makes the constraint tractable in real-time.

Theorem 1. *Let $\bar{u}(\cdot)$ be an input signal with zero-order hold that satisfies (13). If the system is incrementally stable in $\bar{\mathcal{S}}_I$, then $\phi_t^{\bar{u}(\cdot)}(x_0) \in \bar{\mathcal{S}}_I$ for all $t \geq 0$.*

Proof. From incremental stability, we have $\forall x_1, x_2 \in \bar{\mathcal{S}}_I$, and for $\beta : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ nonincreasing in t , and $\forall t_1, t_2 \in \mathbb{R}_+$ with $t_2 > t_1$,

$$\begin{aligned} \left\| \phi_t^{u(\cdot)}(x_1) - \phi_t^{u(\cdot)}(x_2) \right\| &\leq \beta(\|x_1 - x_2\|, t) \\ \Downarrow \\ \left\| \phi_{t_2}^{u(\cdot)}(x_1) - \phi_{t_2}^{u(\cdot)}(x_2) \right\| &\leq \left\| \phi_{t_1}^{u(\cdot)}(x_1) - \phi_{t_1}^{u(\cdot)}(x_2) \right\| \end{aligned} \quad (14)$$

Therefore,

$$\begin{aligned} \left\| \phi_0^{u(\cdot)}(x_1) - \phi_0^{u(\cdot)}(x_2) \right\| &= \|x_1 - x_2\| \\ \Downarrow \\ \phi_t^{\bar{u}_B^{(\cdot)}}(\mathbf{x}_0) \subset \phi_t^{\bar{u}_B^{(\cdot)}}(x_0) \Big|_{\mathbf{x}_0} \end{aligned} \quad (15)$$

Fix any \mathbf{x}_0, \bar{u}, t . For brevity, let $\Phi_1 := \phi_t^{\bar{u}_B^{(\cdot)}}(\mathbf{x}_0)$ and $\Phi_2 := \phi_t^{\bar{u}_B^{(\cdot)}}(x_0) \Big|_{\mathbf{x}_0}$, and let $\dot{\mathbf{x}} := \frac{\partial \phi_t^{\bar{u}_B^{(\cdot)}}}{\partial x} \Big|_{\mathbf{x}_0} (f(\mathbf{x}_0) + g(\mathbf{x}_0)u)$.

$$\Phi_1 \subset \Phi_2 \Rightarrow \frac{dh}{dx} \Big|_{\Phi_1} \subset \frac{dh}{dx} \Big|_{\Phi_2} \Rightarrow \frac{dh}{dx} \Big|_{\Phi_1} \dot{\mathbf{x}} \subset \frac{dh}{dx} \Big|_{\Phi_2} \dot{\mathbf{x}}.$$

Following the same logic, we have

$$\Phi_1 \subset \Phi_2 \Rightarrow \alpha(h(\Phi_1)) \subset \alpha(h(\Phi_2))$$

Thus,

$$\frac{dh}{dx} \Big|_{\Phi_1} \dot{\mathbf{x}} + \alpha(h(\Phi_1)) \subset \frac{dh}{dx} \Big|_{\Phi_2} \dot{\mathbf{x}} + \alpha(h(\Phi_2)).$$

Therefore, any $\mathbf{x}_0, \bar{u}(\cdot)$ that meets condition (13) will also meet condition (10), and by Proposition 1, $\phi_t^{\bar{u}(\cdot)}(x_0) \in \bar{\mathcal{S}}_I$ for all $t \geq 0$ \square

IV. INPUT DELAY

In the previous sections, it is shown that safety can be guaranteed for sampled-data systems with an affine constraint using control barrier functions. This section will extend the safety guarantees to systems with known time-delay, without simply making the barrier robust to a set of possible input delays, which would degrade system performance. Note that the analysis in this section can be done for any robust control invariant set, not limited to the version discussed in previous sections.

A. Preliminaries

Assumption 1. Suppose that the system has a time delay equal to some integer n multiple of the controller period Δ_t . Therefore, the system evolves with dynamics

$$\dot{x} = f(x) + g(x)\bar{u}(x, t - n\Delta_t) \quad (16)$$

for zero-order hold controller \bar{u} .

This is a reasonable assumption, especially for the time delay caused by the numerical computation of the digital controller. Moreover, rounding of the time-delay can always be made robust with an addition to the state uncertainty.

Since it is not possible to provide any input to the system before time $t = n\Delta_t$, one more assumption is required.

Assumption 2. From any initial set of states $\mathbf{x}_0 = x_0 + \Delta_x \subset \overline{\mathcal{S}}_I$, we require

$$\phi_{n\Delta_t}^0(\mathbf{x}_0) \subset \overline{\mathcal{S}}_I. \quad (17)$$

Here, the 0 in $\phi_{n\Delta_t}^0$ denotes the fact that a control input of zero is applied to the system during this time.

In practice, this is not a restrictive assumption since the initial condition can be set well within the safe set. Moreover, if this is not met, there is no hope to keep the system safe whatsoever.

In order to obtain the state at which the control input will be applied, the most recent $n\Delta_t$ inputs must be stored in a vector \bar{u}_H . Since no input can be applied during time $t \in [0, n\Delta_t]$, the input vector is initialized to all zeros. With this, the state at which the i th computed control action will be applied can be expressed as

$$x_{(i+n)\Delta_t} = \phi_{n\Delta_t}^{\bar{u}_H}(x_{i\Delta_t}) \quad (18)$$

The input history vector \bar{u}_H is executed under zero-order hold, just as the inputs are applied to the system. Starting with the initial state, the algorithm for handling input delay is now described.

B. Algorithm Overview

At initial time-step t_0 , the control action to be implemented at time $t = n\Delta_t$ is computed. To keep the system safe, the barrier conditions must be evaluated at state $x_{n\Delta_t}$, which is computed using Equation (18). Note that the constraint itself does not need to be altered, and is still affine. The only extra step is the integration from x_0 to $x_{n\Delta_t}$.

The input chosen by the quadratic program at time t_0 is then placed at the head of the \bar{u}_H buffer, after each previous value is shifted backwards. Thus, the oldest value in the input buffer is lost, as it has already taken effect on the system.

The computation for all future time-steps is outlined in Algorithm 1.

Algorithm 1: CBF with Input Delay of $n\Delta_t$

```

double  $u_H[n] = \{0\}$ ;
 $i \leftarrow 0$ ;
while true do
     $x_{(i+n)\Delta_t} = \phi_{n\Delta_t}^{\bar{u}_H}(x_{i\Delta_t})$ ;
    Compute safe action with  $u_{des}, x_{(i+n)\Delta_t}$  using (13);
    update  $u_H$ ;
     $i = i + 1$ ;
end

```

While this algorithm may seem trivial, it is only made possible by treating the system as a sampled-data system. The continuous case of this algorithm would be much more complex, as there is no finite time-history of inputs to integrate over. Safety under this algorithm is summarized with the following theorem.

Theorem 2. Given a control invariant set $\overline{\mathcal{S}}_I$, if inputs are chosen with Algorithm 1, and the system model is accurate,

then the system (16) remains safe, i.e. $\phi_t^{\bar{u}}(x_0) \in \overline{\mathcal{S}}_I$ for all $t \geq 0$.

Proof. Assume by contradiction that for some time $t = m\Delta_t$, $x_{m\Delta_t} \notin \overline{\mathcal{S}}_I$. Let this be the first time in which $x \notin \overline{\mathcal{S}}_I$, thus $x_{k\Delta_t} \in \overline{\mathcal{S}}_I \forall k < m$.

Because system integration is accurate,

$$x_{m\Delta_t} = \phi_{n\Delta_t}^{\bar{u}_H^m}(x_{(m-n)\Delta_t}) \quad \bar{u}_H^m = [u_{m-2n+1}, \dots, u_{m-n}]$$

$$x_{(m-1)\Delta_t} = \phi_{n\Delta_t}^{\bar{u}_H^{m-1}}(x_{(m-n-1)\Delta_t}) \quad \bar{u}_H^{m-1} = [u_{m-2n}, \dots, u_{m-n-1}]$$

Since $x_{(m-1)\Delta_t} \in \overline{\mathcal{S}}_I$, the control inputs chosen up until time t_{m-n-1} keep the system in $\overline{\mathcal{S}}_I$. Therefore, the control action u_{m-n} must cause the system to exit $\overline{\mathcal{S}}_I$. The CBF condition at $t = (m-n-1)\Delta_t$ is based on $\phi_{n\Delta_t}^{\bar{u}_H^{m-1}}(x_{(m-n-1)\Delta_t})$, the estimated $x_{(m-1)\Delta_t}$, but since the model is assumed to be correct, it is equal to the actual state. However, if u_{m-n} was computed via CBF condition, then $\phi_{n\Delta_t}^{\bar{u}_H^m}(x_{(m-n)\Delta_t}) \in \overline{\mathcal{S}}_I$ by Theorem 1. This implies that $x_{m\Delta_t} \neq \phi_{n\Delta_t}^{\bar{u}_H^m}(x_{(m-n)\Delta_t})$, which contradicts the assumption that the model is accurate. \square

The set $\overline{\mathcal{S}}_I$ described in Section II is an example of one such control invariant set robust to zero-order hold, but this theorem holds for any other such set.

It is important to recognize the fact that, under this algorithm, one is effectively performing open-loop control over the time-horizon of the input delay. However, due to the state uncertainty result from Section III, it is possible to guarantee safety for a range of possible initial conditions that the system is expected to lay within at the time of the control input being enacted. Thus, we have the following extension:

Corollary 1. Given an invariant set $\overline{\mathcal{S}}_I$, robust to state uncertainty Δ_x , if inputs are chosen with Algorithm 1, and $\phi_{n\Delta_t}^{\bar{u}_H}(x_0) \in x_{n\Delta_t} + \Delta_x$ for any $x_0 \in \overline{\mathcal{S}}_I$ (i.e. the system integration is accurate up to the set uncertainty set Δ_x), then the system (16) remains safe, i.e. $x(t) \in \overline{\mathcal{S}}_I$ for all $t \geq 0$.

Proof. The proof follows directly from the Theorem 1, and simply utilizes the guarantees over state uncertainty from Section III, or from [5] for more general control barrier functions. \square

V. SIMULATION OF RESULTS ON SEGWAY ROBOT

The simulation is done in a ROS-based simulation environment. The full, nonlinear dynamics are integrated under zero-order hold at a variable sampling time Δ_t . The true state of the system is not known to the controller, only the state estimate from an extended Kalman filter. This state observer receives noisy sensor data based on the true state of the system. A pre-feedback gain was computed following Section III-B.

The Segway has 4 states $x = [p, \dot{p}, \theta, \dot{\theta}]^T$. The safe set is described by $\mathcal{S} = \{x | 1 - 4p^2 \geq 0\}$, which enforces the robot position p to stay within a 0.5 m range from the origin. The robust barrier condition (13) is evaluated over sets using the interval arithmetic library libaffa [25]. The constraint is imposed at the 10 closest points to the boundary of the safe set along the backup trajectory.

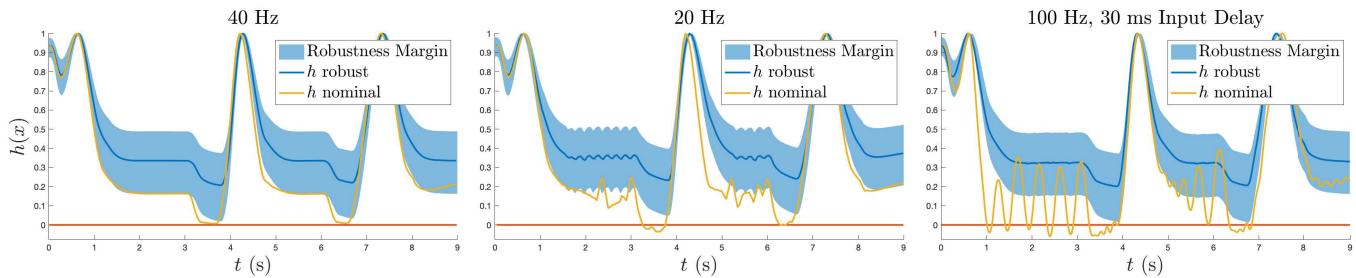


Fig. 1. Results from simulations with three different controller frequencies, and one with input delay.

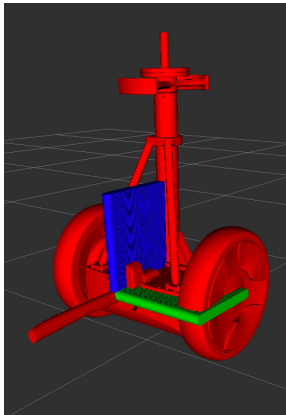


Fig. 2. Segway in simulation. The sequence of axes show the system along the backup trajectory.

Figure 1 shows the result of three simulations with the nominal CBF conditions (4), and the robust condition (13). The robust condition is set to handle a state uncertainty set based on the uncertainty caused by the zero-order hold and the Kalman filter. At 40 Hz, the Segway is able to stay within the set with the nominal controller, but it is unable to maintain invariance at 20 Hz, or in the presence of an input delay of 30 ms. The robust barrier is able to maintain safety for not just the nominal trajectory, but over the entire robustness margin.

VI. CONCLUSION

In this paper, the authors first introduced a robust variation of the backup-controller-based control barrier function that guarantees safety in the presence of a zero-order controller, as well as state uncertainty. The barrier can be used without computing any robust control invariant sets, and can be evaluated in real-time as an affine constraint. An extension of control barrier functions for systems with known input delay was then introduced, made possible only by treating the system as sampled-data as opposed to continuous. Finally, the theory was validated in a real-time high-fidelity simulation environment of a Segway robot. For future work, the authors would like to explore solving this problem using discrete-time control barrier functions in a nonlinear program.

REFERENCES

- [1] 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.
- [2] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [3] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation."
- [4] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Safe policy synthesis in multi-agent pomdps via discrete-time barrier functions," *arXiv preprint arXiv:1903.07823*, 2019.
- [5] T. Gurriet, P. Nilsson, A. Singletary, and A. D. Ames, "Realizable set invariance conditions for cyber-physical systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 3642–3649.
- [6] M. Fiacchini, T. Alamo, and E. F. Camacho, "On the computation of convex robust control invariant sets for nonlinear systems," *Automatica*, vol. 46, no. 8, pp. 1334–1338, 2010.
- [7] I. Mitchell, "A summary of recent progress on efficient parametric approximations of viability and discriminating kernels." in *SNR@ CAV*, 2015, pp. 23–31.
- [8] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3592–3599.
- [9] A. Singletary, P. Nilsson, T. Gurriet, and A. D. Ames, "Online active safety for robotic manipulators," in *2019 International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, p. Final Version Submitted.
- [10] A. Singletary, T. Gurriet, P. Nilsson, and A. Ames, "Safety-critical rapid aerial exploration of unknown environments," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [11] A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "A control barrier perspective on episodic learning via projection-to-state safety," *arXiv preprint arXiv:2003.08028*, 2020.
- [12] P. Ogren, M. Egerstedt, and X. Hu, "A control lyapunov function approach to multi-agent coordination," in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, vol. 2. IEEE, 2001, pp. 1150–1155.
- [13] M. Jankovic, "Control barrier functions for constrained control of linear systems with input delay," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 3316–3321.
- [14] E. Fridman, "A refined input delay approach to sampled-data control," *Automatica*, vol. 46, no. 2, pp. 421–427, 2010.
- [15] G. Orosz and A. D. Ames, "Safety functionals for time delay systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4374–4379.
- [16] M. Jankovic, "Control lyapunov-razumikhin functions and robust stabilization of time delay systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 7, pp. 1048–1060, 2001.
- [17] Q.-C. Zhong, *Robust control of time-delay systems*. Springer Science & Business Media, 2006.
- [18] D. Angeli, "A lyapunov approach to incremental stability properties," *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.
- [19] G. Lars and P. Jürgen, "Nonlinear model predictive control theory and algorithms," 2011.
- [20] M. Abbaszadeh, "Is lipschitz continuity preserved under sampled-data discretization?" *arXiv preprint arXiv:1612.08469*, 2016.
- [21] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [22] R. J. LeVeque, "Finite difference methods for differential equations," *Draft version for use in AMath*, vol. 585, no. 6, p. 112, 1998.
- [23] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, "Interval analysis," in *Applied interval analysis*. Springer, 2001, pp. 11–43.
- [24] 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)*. IEEE, 2018, pp. 98–106.
- [25] O. Gay, D. Coeurjolly, and N. Hurst, "Libaffa-c++ affine arithmetic library for gnu/linux," 2006.