Recurrent Control Barrier Functions: A Path Towards Nonparametric Safety Verification

Jixian Liu and Enrique Mallada

Abstract—Ensuring the safety of complex dynamical systems often relies on Hamilton-Jacobi (HJ) Reachability Analysis or Control Barrier Functions (CBFs). Both methods require computing a function that characterizes a safe set that can be made (control) invariant. However, the computational burden of solving high-dimensional partial differential equations (for HJ Reachability) or large-scale semidefinite programs (for CBFs) makes finding such functions challenging. In this paper, we introduce the notion of Recurrent Control Barrier Functions (RCBFs), a novel class of CBFs that leverages a recurrent property of the trajectories, i.e., coming back to a safe set, for safety verification. Under mild assumptions, we show that the RCBF condition holds for the signed-distance function, turning function design into set identification. Notably, the resulting set need not be invariant to certify safety. We further propose a data-driven nonparametric method to compute safe sets that is massively parellizable, and trades off conservativeness against computational cost.

I. INTRODUCTION

Safety is a fundamental requirement in the control of dynamical systems, particularly in safety-critical applications such as robotics, autonomous vehicles, etc. Safety of the system is typically enforced via Hamilton-Jacobi (HJ) reachability analysis [1] or Control Barrier Functions (CBFs) [2], both of which build a function whose superlevel set is a control invariant safe set. Unfortunately, despite the popularity of these methods, their application relies on the computation of the value function or CBF, which presents significant challenges. HJ-reachability analysis requires solving partial differential equations, which suffers from the curse of dimensionality [3]. The synthesis of valid CBFs often requires solving a Sum-of-Squares (SOS) optimization problem, which is also computationally demanding when applied to high-dimensional systems [4], [5].

To reduce the computational burden, some data-driven methods have been proposed. DeepReach greatly improves computational efficiency for high-dimensional HJ reachability by using neural PDE solvers, but its learning-based approximation limits interpretability despite strong empirical performance [6]. [7] accelerates the synthesis of CBF by utilizing Koopman-based matrix multiplications, though at the expense of losing strict guarantees due to operator approximation. [8] constructs control-invariant safe sets from hard constraints via data-driven CBFs, offering efficiency but

with safety guarantees limited by uneven or sparse sampling quality.

In this paper, we build a framework to trade off the computational complexity of finding safe control sets with the level of conservativeness of the solution, which has theoretical safety guarantees. A key insight of the proposed approach is to substitute the invariance property that Reachability and CBF methods aim to guarantee with a more flexible notion called recurrence [9], [10]. A set is $(\tau$ -) recurrent if every trajectory that leaves the set comes back to it (within τ units of time) infinitely many times. Recurrence has emerged as a practical surrogate for invariance in analysis and verification—e.g., for regions of attraction [11], stability [9], and safety verification [10]. Information-theoretically, enforcing (control) recurrence demands lower data rates than invariance [12] and can often be achieved from finite trajectories [13].

Building on this literature, we extend the notion of Recurrent Barrier Functions proposed in [10] to account for the addition of controls, thus introducing Recurrent Control Barrier Functions (RCBFs). RCBFs relax strict invariance by requiring a finite-time (τ) return to a safe set—conditions met by signed distance functions of given sets—while preserving safety as long as the set excludes the τ -backward reachable tube of the unsafe region. We devise a nonparametric, sampling-based procedure to synthesize RCBFs and verify safety quickly and at scale. To do so, we introduce a robust RCBF condition that uses trajectory data to certify a neighborhood of the initial state; an adaptive sampling method and data-driven exploration remove the need for large optimization programs. The method is GPU-friendly and lets practitioners trade conservativeness for computation without compromising safety.

The remainder of this paper is organized as follows. Section II reviews preliminaries on HJ reachability analysis and CBFs. Section III introduces the definition of RCBFs, extending classical CBFs through recurrence-based safety conditions. Section IV develops the robust conditions that allow for data-driven verification of the RCBF property on a neighborhood of trajectory samples. Section V integrates the robust conditions into a sampling-based method for nonparametric safety verification that actively chooses where to sample based on prior outcomes. Section VI provides numerical validations demonstrating the effectiveness of the proposed approach. Section VII concludes the paper and outlines directions for future work.

Notation: $\|\cdot\|$ is an arbitrary norm on \mathbb{R}^n . For $x \in \mathbb{R}^n$ and r > 0, the closed ball of radius r centered at x is defined

J. Liu and E. Mallada are with the Department of Electrical and Computer Engineering, Johns Hopkins University, MD 21218, U.S.A. jliu376@jh.edu, mallada@jhu.edu.

This work was supported by NSF through grant Global Center 2330450, and Johns Hopkins University Institute for Assured Autonomy.

as $\mathcal{B}_r(x) := \{ y \in \mathbb{R}^n \mid ||y - x|| \le r \}$. Given a set $S \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, the signed distance from x to S is

$$\mathrm{sd}(x,S) := \begin{cases} \inf_{y \in \partial S} \|y - x\|, & \text{if } x \notin S, \\ -\inf_{y \in \partial S} \|y - x\|, & \text{if } x \in S. \end{cases}$$

II. PRELIMINARIES AND RELATED WORK

A. Problem Statement

Consider a continuous-time control system:

$$\dot{x} = F(x, u),\tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the system's state in the state space \mathcal{X} , $u \in U \subseteq \mathbb{R}^m$ is the control input. We define $\mathcal{U}^{(a,b]} := \{u : (a,b] \to U | u$ is measurable}, as the set of control signals on the time interval (a,b], and $\mathcal{U} := \mathcal{U}^{(0,+\infty)}$. Given $u_0 \in \mathcal{U}^{(0,a]}$ and $u_1 \in \mathcal{U}^{(0,b]}$, their concatenation $u_0u_1 \in \mathcal{U}^{(0,a+b]}$ is defined as

$$(u_0u_1)(t) = \begin{cases} u_0(t), & t \in (0, a], \\ u_1(t), & t \in (a, a + b]. \end{cases}$$

Similarly, for $u \in \mathcal{U}^{(a,b]}$ and $(c,d] \subset (a,b]$ we will use $u|_{(c,d]}$ to denote the restriction of u to the interval (c,d].

In a more general setting, consider a sequence of control inputs $u_n \in \mathcal{U}^{(0,\tau_n]}$, where $\tau_n > 0$ for every $n \in \mathbb{N}$. We define $u_{[n]} := u_0 u_1 \cdots u_n$, and $u_{[\infty]} := \lim_{n \to \infty} u_{[n]}$. At times, we adopt a slight abuse of notation by writing u both for instantaneous inputs in U and for signals in $\mathcal{U}^{(a,b]}$; the intended interpretation will always be clear from context.

Given an initial state $x \in \mathbb{R}^n$ and a control signal $u \in \mathcal{U}^{(0,a]}$, we denote by $\phi(t,x,u)$ the trajectory solving (1) for all $t \in (0,a]$. Throughout, we impose the following regularity assumptions on (1).

Assumption 1 (Forward Completeness). The control system (1) is **forward complete**, that is, for any initial condition $x \in \mathbb{R}^n$ and any input $u \in \mathcal{U}$, the solution $\phi(\cdot, x, u)$ exists and is unique on $[0, \infty)$.

Assumption 2 (Uniform Local Lipschitz Continuity). The vector field F(x, u) in (1) is locally Lipschitz in x, uniformly with respect to u. More precisely, for every compact set $S \subseteq \mathbb{R}^n$, there exists a constant $L \geq 0$ such that

$$||F(y,u) - F(x,u)|| \le L||y - x||, \quad \forall x, y \in S, \ \forall u \in U.$$

B. Safety Assessment

Our goal is finding input signals $u(\cdot) \in \mathcal{U}$ such that the solution $\phi(t,x,u)$ to (1) can avoid an unsafe region $\mathcal{X}_u \subset \mathcal{X}$ for all time $t \geq 0$. To that end, we aim to design an algorithm that can quickly find a strict subset of $\mathcal{X} \backslash \mathcal{X}_u$ that achieves this goal. We will therefore say that a state x is considered to be safe if one can find a control $u \in \mathcal{U}$ such that the state trajectory $\phi(t,x,u)$ does not visit the unsafe region for all future time.

Definition 1 (Safe State). A state $x \in \mathcal{X}$ is said to be **safe** w.r.t. the system (1) if there exists a control $u \in \mathcal{U}$ such that the trajectory $\phi(t, x, u)$ never visits the unsafe region \mathcal{X}_u , i.e., $\exists u \in \mathcal{U}$, s.t. $\forall t \geq 0$, $\phi(t, x, u) \notin \mathcal{X}_u$.

A common approach to ensure safety according to Definition 1 is to find some set $\mathcal C$ that does not intersect with $\mathcal X_u$ and has the additional property that trajectories that start in $\mathcal C$ can be kept in $\mathcal C$. That is, $\mathcal C$ is control invariant.

Definition 2 (Control Invariant Set). A set $C \subseteq \mathcal{X}$ is **control** invariant w.r.t. (1) if for every $x \in C$, there exists a control $u \in \mathcal{U}$ such that $\phi(t, x, u) \in C$ for all t > 0.

C. Reachability Analysis

As mentioned before, a widely adopted method to verify safety is the Hamilton–Jacobi (HJ) reachability analysis. In this framework, one aims to compute the collection of all initial states from which, no matter what control one chooses, the trajectory will eventually end in the unsafe set \mathcal{X}_u . We provide a formal definition next.

Definition 3 (Backward Reachable Tube). For a set S, and constant T > 0, the T-Backward Reachable Tube (T-BRT) is defined as:

$$\mathcal{R}_T(S) := \{ x \mid \forall u \in \mathcal{U}^{(0,T]}, \exists t \in (0,T], \text{s.t.} \phi(t,x,u) \in S \}.$$

When $T = \infty$, we refer to it simply as the Backward Reachable Tube (BRT) and denote by $\mathcal{R}(S)$.

To construct the BRT, the HJ reachability procedure casts the safety verification task as an optimal control problem. Here, the controller's objective is to avoid the unsafe set \mathcal{X}_u . This is quantitatively expressed through a value function $V(x,t) := \min_{s \in [-t,0]} l(\phi(s,x,u))$, which measures the minimum cost or the distance to entry \mathcal{X}_u . In the absence of disturbances, the evolution of V(x,t) is governed by a Hamilton-Jacobi-Isaacs Variational Inequality that takes the form of a Hamilton-Jacobi-Bellman equation [3]:

$$\min\{D_t V(x,t) + H(x,t,\nabla V(x,t)), l(x) - V(x,t)\} = 0,$$

where l(x) is the terminal condition where V(x,0) = l(x), and $H(x,t,\nabla V(x,t)) := \max_{u \in U} D_x V(x,t) \cdot f(x,u)$.

Once V(x,T) is computed, the T-BRT is given by the sublevel set

$$\mathcal{R}_T(\mathcal{X}_u) = \{ x \mid V(x,T) \le 0, x \in \mathcal{X}_u \},\$$

which implies that any state within this set will eventually lead to \mathcal{X}_u under any control $u(\cdot)$ within less than T units of time. HJ reachability gives rigorous safety guarantees when $x \in \mathcal{R}^c_{+\infty}(\mathcal{X}_u)$, which is the largest safe control invariant set, but is computational costly in high dimensions. Efficient solvers for the HJ PDE mitigate this [3], improving practicality. Our work tackles safety from a complementary angle.

D. Control Barrier Functions

CBFs offer another conservative alternative to HJ reachability. By bounding \dot{h} with an extended class- \mathcal{K} function, they render a chosen set \mathcal{C} control invariant and thus ensure safety. To formally introduce CBFs we are required to introduce the notion of extended class \mathcal{K} functions.

Definition 4 (Extended Class K Function). A function κ : $\mathbb{R} \to \mathbb{R}$ is an extended class K function if it is continuous, strictly increasing, and satisfies $\kappa(0) = 0$.

We are now ready to formally introduce CBFs.

Definition 5 (Control Barrier Function [2]). A continuously differentiable function h(x) is a CBF for the system (1) if there exists an extended class K function κ such that,

$$\max_{u \in U} L_F h(x) + \kappa(h(x)) \ge 0, \tag{2}$$

for all $x \in \mathcal{X}$, and where $L_F h(x) = \frac{\partial h}{\partial x}^{\top} F(x, u)$, are first-order Lie derivatives.

Theorem 1 ([2]). An immediate consequence of Definition 5 is that any Lipschitz-continuous controller k(x) satisfying

$$k(x) \in \{ u \in U \mid L_F h(x) + \kappa(h(x)) \ge 0 \},$$

renders the set $h_{\geq 0} := \{x : h(x) \geq 0\}$ invariant. Thus, $h_{\geq 0}$ is, by definition, control invariant.

Thus, if such a CBF h exists and $h_{\geq 0} \cap \mathcal{X}_u = \emptyset$, all states in $h_{\geq 0}$ can find a control signal $u \in \mathcal{U}$, whose signal at any moment is in the set of k(x). That is to say for any intial state $x \in h_{\geq 0}$, there exists $u \in \mathcal{U}$ such that $\phi(t, x, u) \in h_{\geq 0}, \forall t \geq 0$, which means the states in $h_{\geq 0}$ are safe [2].

Sum-of-Squares (SOS) programming is widely used to synthesize/verify polynomial CBFs, but its cost grows rapidly with system dimension [14], [15], and polynomials may poorly capture complex safety sets. Neural network CBFs improve expressivity [16], [17], [18], yet their validity is harder to certify due to limited interpretability [18].

III. RECURRENT CONTROL BARRIER FUNCTION

The core idea behind ensuring safety using traditional CBFs is to construct a scalar function that makes $h_{\geq 0}$ control invariant. Such sets can be as computationally expensive as a BRT, making CBF synthesis difficult. Leveraging recurrence, we show this explicit invariant set is unnecessary: valid RCBFs can be built from **control recurrent sets**, which relax invariance while keeping safety guarantees.

A. Control Recurrent Sets

In this section, we briefly cover the definition of recurrent sets in a control systems setting, which broadly allow trajectories to leave a set, provided they come back to it. The presentation follows [10], [11], [12], particularly [12].

Definition 6 (Control Recurrent Sets). A compact set $S \subseteq \mathbb{R}^n$ is called **control recurrent** w.r.t. (1) if, for all $x \in S$, $\exists u \in \mathcal{U}$, such that for any $t \geq 0$,

$$\exists t' > t \text{ with } \phi(t', x, u) \in S.$$
 (3)

Likewise, a set $S \subseteq \mathbb{R}^n$ *is called* **control** τ -recurrent $(\tau > 0)$ w.r.t. (1) if, for all $x \in S$, $\exists u \in \mathcal{U}$, such that for any $t \geq 0$,

$$\exists t' > t$$
, with $t' - t \in (0, \tau]$, and $\phi(t', x, u) \in S$. (4)

We refer to such $\phi(t, x, u)$ as a $(\tau$ -)recurrent trajectory.

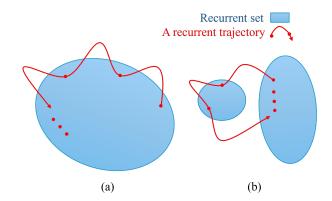


Fig. 1: Illustration of Recurrent Sets and Recurrent Trajectories

As Figure 1 shows, although a τ -recurrent set is not necessarily invariant, it ensures that trajectories starting in S will revisit it within at most τ -time units infinite times. Notably, based on the Definition 6, an invariant set is always τ -recurrent for any $\tau>0$. Additionally, a 0-recurrent set is equivalent to an invariant set. Thus, Definition 6 generalizes invariance by allowing the trajectory $\phi(t,x,u)$ to leave the set S before returning [10]. Compared with the invariant sets, recurrent sets show a more flexible shape; it does not need the region to be connected, and it does not require the system (1) to point inwards (or at least not outwards) on all the boundary ∂S .

B. Recurrent Control Barrier Function

We now move towards introducing the proposed Recurrent Control Barrier Functions. In fact, similar to [10], simply requiring trajectories to return to the set within a finite time, infinitely many times can guarantee the safety for the dynamical system.

Definition 7 (Recurrent Control Barrier Function). *Consider* the control system (1). A continuous function $h : \mathbb{R}^n \to \mathbb{R}$ is a **Recurrent Control Barrier Function (RCBF)** if for all $x \in D_0 := h_{\geq -c}$, with c > 0, $\exists u \in \mathcal{U}^{(0,\tau]}$ s.t.

$$\max_{t \in (0,\tau]} e^{\gamma(h(\phi(t,x,u)))t} h(\phi(t,x,u)) \ge h(x), \tag{5}$$

where the function $\gamma: \mathbb{R} \to \mathbb{R}_{>0}$.

In (5) we follow the standard convention that when the sup is not achieved within the set $(0,\tau]$ the max is $-\infty$. Thus, for the max to be lower bounded, it implies that it is achieved within $(0,\tau]$. A particular choice of γ that will be of use throughout this paper is

$$\gamma_{\alpha,\beta}(s) = \Big\{\alpha, \quad \text{if } s \geq 0\,, \quad \text{ and } \quad \beta, \quad \text{if } s < 0\,, \qquad \ (6)$$

where α and β are positive parameters. This will be particularly useful in our converse results in Section III-C.

The following theorem describes how to use RCBFs to assess safety.

Theorem 2 (Safety Assessment via RCBFs). Let h be an RCBF as in Definition 7. Then:

(i) The superlevel set $h_{\geq 0}$ is control τ -recurrent, i.e., for any $x \in h_{\geq 0}$ there exists $u \in \mathcal{U}$ such that the trajectory $\phi(t, x, u)$ always returns to $h_{\geq 0}$ within time τ .

Moreover, if $h_{>0} \cap \mathcal{R}_{\tau}(\mathcal{X}_u) = \emptyset$ *, then:*

(ii) For any $x \in h_{\geq 0}$, every $u \in \mathcal{U}$ that renders $\phi(t, x, u)$ τ -recurrent also ensures that $\phi(t, x, u) \notin \mathcal{X}_u$ for all $t \geq 0$.

In particular, under the condition $h_{\geq 0} \cap \mathcal{R}_{\tau}(\mathcal{X}_u) = \emptyset$, the set $h_{\geq 0}$ is safe.

Proof. (i) Control τ -recurrence of $h_{\geq 0}$. Given $x \in h_{\geq 0}$, fix $x_0 := x$ and $t_0 := 0$. By the RCBF condition (5), there exists $u_0 \in \mathcal{U}^{(0,\tau]}$ and a time

$$\tau_0 := \max \left\{ \arg \max_{t \in (0,\tau]} e^{\gamma(h(\phi(t,x_0,u))) t} h(\phi(t,x_0,u)) \right\},$$
(7)

such that $x_1 := \phi(\tau_0, x_0, u_0) \in h_{\geq 0}$ and $t_1 := \tau_0 + t_0$. Proceed inductively: given $x_n \in h_{\geq 0}$ and t_n , use (5) to select $u_n \in \mathcal{U}^{(0,\tau]}$ and $\tau_n \in (0,\tau]$ as in (7), leading to $x_{n+1} := \phi(\tau_n, x_n, u_n) \in h_{\geq 0}$, and $t_{n+1} = \tau_n + t_n$.

The desired control $u \in \mathcal{U}$ is thus defined by concatenating the restrictions of u_n to the intervals $(0, \tau_n]$, i.e.,

$$u_{[n]} = u_0 \big|_{(0,\tau_0]} u_1 \big|_{(0,\tau_1]} \dots u_n \big|_{(0,\tau_n]} \in \mathcal{U}^{(0,t_n]}$$

and letting $u=\lim_{n\to\infty}u_{[n]}\in\mathcal{U}^{(0,t^*]}$, where $t^*=\lim_{n\to\infty}t_n$. An argument similar to [9, Lemma 1] shows that $t^*=\infty$. Moreover, it follows from the construction that for all $n\geq 0$,

$$x_{n+1} = \phi(\tau_n, x_n, u_n |_{(0, \tau_n]}) = \phi(t_n, x, u),$$
 (8)

and therefore $\phi(t_n,x,u)\in h_{\geq 0}$. It follows then from the fact that for all $n\geq 0,\ x_n\in h_{\geq 0},\ t_{n+1}-t_n\in (0,\tau]$ and $t_n\to\infty$, that the trajectory $\phi(t,x,u)$ is τ -recurrent w.r.t. $h_{\geq 0}$. Since $x\in h_{\geq 0}$ was chosen arbitrarily, (i) follows.

(ii) Safety under $h_{\geq 0} \cap \mathcal{R}_{\tau}(\mathcal{X}_u) = \emptyset$. Assume $h_{\geq 0} \cap \mathcal{R}_{\tau}(\mathcal{X}_u) = \emptyset$. Take any $x \in h_{\geq 0}$ and any control $u \in \mathcal{U}$ that renders $\phi(t,x,u)$ τ -recurrent w.r.t. $h_{\geq 0}$. Suppose, towards a contradiction, that the trajectory is unsafe: there exists t' > 0 with $\phi(t',x,u) \in \mathcal{X}_u$. Since $h(x) \geq 0$ and h < 0 on \mathcal{X}_u , by continuity there exists a last exit time $t'' \in [0,t']$ with $h(\phi(t'',x,u)) = 0$ and $h(\phi(t,x,u)) < 0$ for all $t \in (t'',t']$. Because $h_{\geq 0} \cap \mathcal{R}_{\tau}(\mathcal{X}_u) = \emptyset$, the state $\phi(t'',x,u)$ cannot reach \mathcal{X}_u within time τ , hence $t' - t'' > \tau$ and

$$h(\phi(t''+t,x,u)) < 0 \quad \forall t \in (0,\tau],$$

which contradicts with the fact that u renders $\phi(t,x,u)$ τ -recurrent w.r.t. $h_{\geq 0}$.

Besides ensuring safety, RCBFs also share similar properties like standard CBFs. In particular, it is possible to show that whenever $x \in D_0 \backslash h_{\geq 0}$, there is always some $u \in \mathcal{U}$ such that

$$\lim \inf_{t \to \infty} d(h_{\geq 0}, \phi(t, x, u)) = 0,$$

with $d(S,x) := \min_{y \in S} \|y - x\|$, thus ensuring that trajectories come back to $h_{\geq 0}$ under simplified condition. We do not make these claims formal here, and refer the reader to [10] for similar arguments.

C. Signed Distance Function: a Valid RCBF

In this section, we present a striking result. The existence of a CBF h satisfying some regularity conditions is sufficient for synthesizing a simple sign distance function that satisfies our RCBF condition. Firstly, we require h to be sector contained.

Definition 8 (Sector Containment). Let $h: D \subseteq \mathbb{R}^n \to \mathbb{R}$ be continuous. If $\exists a_1, a_2 > 0$ such that

$$(h(x) - a_1 \operatorname{sd}(x, h_{<0}))(h(x) - a_2 \operatorname{sd}(x, h_{<0})) \le 0$$
 (9)

for all $x \in D$, we say that h is sector contained.

The second condition refers to the particular choice of extended class $\mathcal K$ function. In particular, we will consider the sub-class:

$$\kappa_{\alpha,\beta}(s) := \gamma_{\alpha,\beta}(s) \, s. \tag{10}$$

Theorem 3 (Validity of Signed Distance Function as RCBF). Let h be a CBF satisfying (2) and (9) over $D_0 := h_{\geq -c}$ with parameters c > 0 and $a_2 > a_1 > 0$, and extended class \mathcal{K} function $\kappa_{\alpha,\beta}$ as in (10), with parameters $\alpha > 0$ and $\beta > 0$. Then, for any closed set S satisfying $h_{\geq 0} \subseteq S \subseteq h_{\geq -c}$, with $\partial S \cap h_{=0} = \emptyset$, the function

$$\hat{h}(\cdot) = -\mathrm{sd}(\cdot, S)$$

is an RCBF over $\hat{D}_0 := \hat{h}_{\geq -\hat{c}}$ where $\hat{c} \geq 0$ is the largest constant satisfying $\hat{h}_{>-\hat{c}} \subseteq h_{>-c}$.

Precisely, for all $x \in \hat{h}_{>-\hat{c}}$, there exists $u \in \mathcal{U}^{(0,\tau]}$ s.t.

$$\max_{t \in (0,\hat{\tau}]} e^{\hat{\gamma}(\hat{h}(\phi(t,x,u)))t} \hat{h}(\phi(t,x,u)) \ge \hat{h}(x)$$
 (11)

where $\hat{\gamma} := \gamma_{\hat{\alpha},\hat{\beta}}$, with $\hat{\alpha},\hat{\beta} > 0$ satisfying $\hat{\alpha} > \alpha,\hat{\beta} < \beta$, and $\hat{\tau} \ge \max\{\frac{\log(a_2/a_1)}{\hat{\alpha}-\alpha},\frac{\log(a_2/a_1)}{\beta-\hat{\beta}}\} + \frac{\log(\overline{\delta}/\underline{\delta})}{\min\{\hat{\alpha},\hat{\beta}\}}$ where

$$\overline{\delta} := \sup_{x \in D_0} \left(\operatorname{sd}(x, S) - \operatorname{sd}(x, h_{\geq 0}) \right),$$

$$\underline{\delta} := \inf_{x \in D_0} \left(\operatorname{sd}(x, S) - \operatorname{sd}(x, h_{\geq 0}) \right).$$

Proof. The proof follows closely similar results for the noncontrol case [10, Theorem 11] and it is omitted due to space constraints. \Box

IV. SAFETY ENFORCEMENT USING RECURRENCE

In this section, we aim to develop robust conditions that leverage trajectory samples to certify the satisfaction of the RCBF condition on a neighborhood of the trajectory. The proposed approach reduced the problem of checking the RCBF condition on uncountably many points, to checking it on finitely many states, possibly in parallel.

A. Verification of a Cell

To proceed, we first analyze how trajectories deviate from one another. This step lays the foundation for constructing a stronger verification criterion that ensures local safety in a neighborhood $\mathcal{B}_r(x)$ of each sampled point x, which we refer here as a cell.

Lemma 1. Suppose that two trajectories $\phi(t, x, u)$, $\phi(t, y, u)$ starts from x and y respectively and share the same control input u all the times, where $||x - y|| \le r$. Then we have:

$$|\operatorname{sd}(\phi(t, y, u), S) - \operatorname{sd}(\phi(t, x, u), S)| \le re^{Lt}, \forall t \ge 0, \quad (12)$$

where L is a uniform bound on the Lipschitz constant of (1) on the x variable.

We leverage Lemma 1 to verify different properties of a given cell $\mathcal{B}_r(\cdot)$. In particular, Theorem 4 below gives us a condition that verifies whether a cell $\mathcal{B}_r(x)$ is completely inside $\mathcal{R}_\tau(\mathcal{X}_u)$, completely outside $\mathcal{R}_\tau(\mathcal{X}_u)$, or partially inside $\mathcal{R}_\tau(\mathcal{X}_u)$. This will be critical to over approximate $\mathcal{R}_\tau(\mathcal{X}_u)$.

Theorem 4. Consider a state $x \in \mathcal{X}$, and let $\mathcal{B}_r(x) := \{y | \|y - x\| \le r\}$ be a neighborhood of x. Then, we have:

(i) Given $x \in \mathcal{X}$, if there exists $u \in \mathcal{U}^{(0,\tau]}$ s.t.

$$\forall t \in [0, \tau], \ \operatorname{sd}(\phi(t, x, u), \mathcal{X}_u) > re^{Lt},$$
 (13)

for some r > 0, then $\mathcal{B}_r(x) \cap \mathcal{R}_{\tau}(\mathcal{X}_u) = \emptyset$,

(ii) Conversely, given $x \in \mathcal{X}$, if for all $u \in \mathcal{U}^{(0,\tau]}$,

$$\exists t \in (0, \tau], \text{ s.t. } \operatorname{sd}(\phi(t, x, u), \mathcal{X}_u) < -re^{Lt},$$
 (14)

for some r > 0, then $\mathcal{B}_r(x) \subseteq \mathcal{R}_\tau(\mathcal{X}_u)$

Proof. (i) If the initial states satisfy the condition (13), then by Lemma 1, we have:

$$\operatorname{sd}(\phi(t, y, u), \mathcal{X}_u) \ge \operatorname{sd}(\phi(t, x, u), \mathcal{X}_u) - re^{-Lt} > 0,$$

for all $t \in [0, \tau]$ and all $y \in \mathcal{B}_r(x)$. Hence,

$$\mathcal{B}_r(x) \cap \mathcal{R}_\tau(\mathcal{X}_u) = \emptyset.$$

(ii) If instead the initial states satisfy the condition (14), let $t^* < \tau$ be the time at which

$$\operatorname{sd}(\phi(t^*, x, u), \mathcal{X}_u) < -re^{Lt^*}.$$

Again, by Lemma 1, we have:

$$\operatorname{sd}(\phi(t^*, y, u), \mathcal{X}_u) \le \operatorname{sd}(\phi(t^*, x, u), \mathcal{X}_u) + re^{Lt^*} < 0,$$

for all $y \in \mathcal{B}_r(x)$. Consequently,

$$\mathcal{B}_r(x) \subseteq R_{t^*}(\mathcal{X}_u).$$

The following theorem verifies whether the states of a cell all satisfy the RCBF condition (5) or all such states are guaranteed not to satisfy such a condition.

Theorem 5. Given a closed set S, a candidate RCBF $h(\cdot) := -\operatorname{sd}(\cdot, S)$, and function $\gamma := \gamma_{\alpha,\beta}$, with $\alpha, \beta > 0$.

(i) Let

$$\hat{h}_r^-(x, u, t) := h(\phi(t, x, u)) - re^{Lt} \tag{15}$$

and assume that $\exists u \in \mathcal{U}^{(0,\tau]}$ s.t. the following holds

$$\max_{t \in (0,\tau]} e^{\gamma(\hat{h}_r^-(x,u,t))t} \, \hat{h}_r^-(x,u,t) \ge h(x) + r, \quad (16)$$

for some r > 0. Then for all $y \in \mathcal{B}_r(x)$, the RCBF condition is satisfied, i.e.,

$$\max_{t \in (0,\tau]} e^{\gamma(h(y,u,t))t} h(\phi(t,y,u)) \ge h(y), \tag{17}$$

(ii) Let

$$\hat{h}_r^+(x, u, t) := h(\phi(t, x, u)) + re^{Lt}$$
(18)

and assume that $\forall u \in \mathcal{U}^{(0,\tau]}$ s.t. the following holds

$$\max_{t \in (0,\tau]} e^{\gamma(\hat{h}_r^+(x,u,t))t} \, \hat{h}_r^+(x,u,t) < h(x) - r \tag{19}$$

for some r > 0. Then, for all $y \in \mathcal{B}_r(x)$, the RCBF condition is not satisfied, i.e.,

$$\max_{t \in (0,\tau]} e^{\gamma (h(\phi(t,y,u)))t} h(\phi(t,y,u)) < h(y). \tag{20}$$

Proof. (i) Let t^* and u^* be the time that maximizes the left-hand side of (16), i.e.

$$t^* = \arg\max_{t \in (0,\tau]} \max_{u \in \mathcal{U}^{(0,\tau]}} e^{\gamma(\hat{h}_r^-(x,u,t))t} \, \hat{h}_r^-(x,u,t),$$

$$u^* = \arg\max_{u \in \mathcal{U}^{(0,\tau]}} \max_{t \in (0,\tau]} e^{\gamma(\hat{h}_r^-(x,u,t))t} \, \hat{h}_r^-(x,u,t).$$

At this maximizing time $t^* \in (0, \tau]$, it follows that

$$\max_{u \in \mathcal{U}, t \in (0, \tau]} e^{\gamma (h(y, u, t))t} h(\phi(t, y, u))$$

$$\geq e^{\gamma (h(y, u^*, t^*))t^*} h(y, u^*, t^*)$$

$$\geq e^{\gamma (\hat{h}_r^-(x, u, t^*))t^*} \hat{h}_r^-(x, u, t^*)$$

$$\geq h(x) + r$$

$$\geq h(y),$$

where the first inequality follows from the definition of maximum, the second and fourth inequalities are derived from Lemma 1 and the third inequality is derived from the conditions of (17).

(ii) Let t^* and u^* be the time and control signal that maximize the left-hand side of (20), i.e.

$$t^* = \arg\max_{t \in (0,\tau]} \max_{u \in \mathcal{U}^{(0,\tau]}} e^{\gamma(h(y,u,t))t} h(y,u,t),$$

$$u^* = \arg\max_{u \in \mathcal{U}^{(0,\tau]}} \max_{t \in (0,\tau]} e^{\gamma(h(y,u,t))t} \, h(y,u,t).$$

Again, at this maximizing time $t^* \in (0, \tau]$, we get

$$\max_{u \in \mathcal{U}^{(0,\tau]}, t \in (0,\tau]} e^{\gamma(h(y,u,t))t} h(y,u,t)$$

$$\leq e^{\gamma(\hat{h}_r^+(x,u^*,t^*))t^*} \hat{h}_r^+(x,u^*,t^*)$$

$$\leq \max_{u \in \mathcal{U}^{(0,\tau]}, t \in (0,\tau]} e^{\gamma(\hat{h}_r^+(x,u,t))t} \hat{h}_r^+(x,u,t)$$

$$< h(x) - r$$

$$< h(y),$$

where the second inequality is derived from the definition of the maximum, the first and fourth inequalities are derived from Lemma 1, and the third inequality is derived from the conditions of (19).

V. Numerical Methods

Building on Theorem 4 and Theorem 5, we propose a safety verification algorithm aimed at finding a set $S \subset \mathcal{X}$ such that $h = -\operatorname{sd}(x, S)$ satisfies all the necessary properties for safety assessment described in Theorem 2. The proposed method adaptively partitions \mathcal{X} into cells $\mathcal{G} := \{g_i :=$ $\mathcal{B}_{r_i}(x_i)\}_{i=1}^{|\mathcal{G}|}$, such that $g_i \cap g_j = \emptyset, i \neq j$ and $\cup \mathcal{G} := \emptyset$ $\bigcup_{i=1}^{|\mathcal{G}|} g_i = \mathcal{X}$, when running our algorithms, two disjoint lists, \mathcal{G}_s (tentative safe cells) and \mathcal{G}_u (verified unsafe cells) are maintained and refined, and progressively, cells from \mathcal{G}_s are assigned to \mathcal{G}_u (while keeping $\mathcal{X} = (\cup \mathcal{G}_u) \cup (\cup \mathcal{G}_s)$) until one is able to guarantee that the safe set $S = \cup \mathcal{G}_s$ and RCBF $h = -\operatorname{sd}(x, S)$ satisfy the robust conditions of Theorem 2.

This verification process is carried out in three stages, as illustrated in Algorithm 1, where lines 2, 3, and 4 represent stages 1, 2, and 3, respectively.

Algorithm 1 VerifyRegion($\mathcal{X}, \tau, \alpha, \beta$)

```
1: Input: State Space \mathcal{X}, Parameters \tau, \alpha, and \beta > 0.
2: \mathcal{G}_s, \mathcal{G}_u = \text{VerifyCells}(\mathcal{X}, \mathcal{X}_u, 0, (13), (14))
3: \mathcal{G}_s, \mathcal{G}_u = \text{VerifyCells}(\mathcal{G}_s, \mathcal{G}_u, \tau, (13), (14))
4: \mathcal{G}_s, \mathcal{G}_u = \text{VerifyCells}(\mathcal{G}_s, \mathcal{G}_u, \tau, (17), (20)) \triangleright \alpha \text{ and } \beta
     are used in the conditions (17) and (20).
```

5: **return** \mathcal{G}_s , \mathcal{G}_u

Each stage aims to sequentially get a better approximation of a region $S \subseteq \mathcal{X}$ for $h = -\operatorname{sd}(x, S)$ to be a valid RCBF. Stage 1 first finds a sufficiently fine outer approximation of \mathcal{X}_u . Stage 2 finds an outer approximation of $\mathcal{R}_{\tau}(\cup \mathcal{G}_u)$, with \mathcal{G}_u being the output of Stage 1. Finally, Stage 3 further uses $S = \cup \mathcal{G}_s$ in order to find such a h satisfy the RCBF condition.

All stages are implemented by calling a VerifyCells routine, Algorithm 2, with the current estimates of \mathcal{G}_s and \mathcal{G}_u and the assignment conditions C_s and C_u , corresponding to conditions (13) and (17), and (14) and 20, respectively. Note that we initially start with one cell (the full set \mathcal{X}), and each pass progressively finds finer and more accurate approximations for \mathcal{G}_s and \mathcal{G}_u .

Algorithm 2 VerifyCells($\mathcal{G}, \mathcal{G}_u, \tau, \mathcal{C}_s, \mathcal{C}_u$)

```
1: Input: Grid \mathcal{G} and \mathcal{G}_u, Parameter \tau \geq 0, Robust Safe
     Condition C_s, and Robust Unsafe Condition C_u.
    while \mathcal{G} \neq \emptyset do
           for \forall g_i = \mathcal{B}_r(x) \in \mathcal{G}, i \in \mathbb{N} for some r and x do
3:
                  \mathcal{G} \leftarrow \mathcal{G} - \{g_i\}
4:
              \mathcal{G}, \mathcal{G}_s, \mathcal{G}_u = \widetilde{\operatorname{SafetyCheck}}(g_i, \mathcal{G}, \mathcal{G}_u, \tau, \mathcal{C}_s, \mathcal{C}_u)
5:
           end for
7: end while
8: return \mathcal{G}_s, \mathcal{G}_u
```

The verification process in Algorithm 2 can be done in parallel for all cells in the input \mathcal{G} and ends when this set is empty. This framework facilitates high parallelism through concurrent cell verification while ensuring rigorous safety guarantees. That is to say, each cell in Algorithm 2 is eventually verified to be safe or declared to be unsafe by employing the safe and unsafe assignment conditions, C_s and C_u , corresponding to each stage. The specific verification of each cell is implemented by routines SafetyCheck (Algorithm 3).

Algorithm 3 SafetyCheck $(g_i, \mathcal{G}, \mathcal{G}_u, \tau, \mathcal{C}_s, \mathcal{C}_u)$

```
1: Input: Cell g_i to check; Grids \mathcal{G}, \mathcal{G}_s, \mathcal{G}_u; Parameter \tau >
   0; Robust Safe Condition C_s; Robust Unsafe Condition
```

2: $\mathcal{X}_u = \mathcal{G}_u$

3: $\mathcal{G}_s = \emptyset$

4: Sample n_s trajectories of length τ from a representative point in g_i

5: if all sampled trajectories satisfy C_u then

 $\mathcal{G}_u \leftarrow \mathcal{G}_u \cup \{g_i\}$

7: else if at least one sampled trajectory satisfies C_s then

 $\mathcal{G}_s \leftarrow \mathcal{G}_s \cup \{g_i\}$ 8:

9: else

 $\mathcal{G} \leftarrow \mathcal{G} \cup \operatorname{SplitCell}(q_i)$ 10:

11: **end if**

12: **return** $\mathcal{G}, \mathcal{G}_s, \mathcal{G}_u$

Notably, to verify stages 2 and 3, each cell is required to sample n_s trajectories of length τ and check whether all satisfy an unsafe condition or at least one satisfies the safe condition. Finally, in cases where neither safe nor unsafe conditions can be verified, one is required to either increase the resolution via the SplitCell routine (Algorithm 4) or eventually declare the cell to be unsafe when the resolution is met.

Algorithm 4 SplitCell(g)

```
1: Input: Grid cell g = \mathcal{B}_r(x) \in \mathcal{G}
```

2: Let $(x_1, x_2, \dots, x_n) = x$ 3: $P := \left\{ x + \frac{2r}{3} \cdot \boldsymbol{\delta} | \boldsymbol{\delta} \in \{-1, 0, 1\}^n \right\}$ 4: **return** $\mathcal{G}_{split} := \left\{ \mathcal{B}_{\frac{r}{3}}(p) | p \in P \right\}$

VI. NUMERICAL SIMULATIONS

In this section, we validate the performance and the safety of our algorithm using a 3D evasion problem:

$$\dot{x} = \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v + v\cos x_3 + ux_2 \\ v\sin x_3 - ux_1 \\ -u \end{bmatrix},$$

with $[x_1, x_2]^T \in \mathbb{R}^2$ representing the relative planar location and $x_3 \in [0, 2\pi]$ the relative direction. $v \ge 0$ is the aircraft velocity and $u \in [-1,1]$ is the evader's angular velocity. A collision occurs if $\sqrt{x_1^2 + x_2^2} \le 1$, which defines a cylindrical collision set of radius 1 along the x_3 -axis. Our goal is to determine the set of initial states that inevitably lead to a collision, regardless of the evader's actions.

A. Results Comparison

We use $\tau=1$ s and a total number of control samples per cell $n_s=3000$. V_{BRT} denotes the unsafe region volume computed via HJ reachability at $r_{\min}=0.041$ (the grid resolution). We calculate the intersection ratio $(V_{BRT\cap h'\leq 0}/V_{BRT})$ between our method ($\beta=\alpha=0.05$) and HJ solutions at different precisions. As shown in Table I and Figure 2, low-precision HJ analysis underestimates unsafe regions, risking safety misjudgment. Our method guarantees complete containment of true unsafe regions at all precisions.

TABLE I: Comparison of the Fraction of the Unsafe Zone Volume Captured by Different Methods at Different Precision

Methods $\setminus r_{\min}$	1.111	0.370	0.123	0.041
HJ BRT [19]	0.510 (X)	0.900 (X)	0.953 (X)	1 (1)
Recurrent Set	1 (🗸)	1 (🗸)	1 (🗸)	1 (1)

Beyond safety guarantees, Table II demonstrates our method's faster computation time at high precision through parallelization.

TABLE II: Comparison of Computation Time between HJ reachability and Recurrent Set Approximation under Different Precision, $\tau=1$ s, $\alpha=1$, $n_{\rm s}=3000$

Method $\setminus r_{\min}$	1.111	0.370	0.123	0.041
HJ BRT [19]	0.02 s (X)	0.19 s (X)	2.33 s (X)	83.73 s (✓)
Recurrent Set	0.13 s (✓)	0.61 s (✓)	3.19 s (✓)	19.75 s (✓)

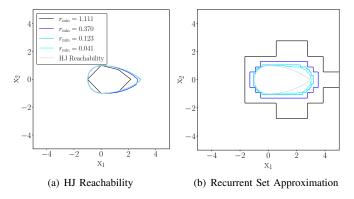


Fig. 2: Contour Plot of the Boundary of the Unsafe Region with Different Precision and Different Methods when $x_3 = \pi$

B. Ablation Study

Definition 6 shows the recurrent set converges to the invariant set as $\tau \to 0$. To gauge parameter effects on RCBF performance, we ran a sweep over two metrics: the normalized volume gap $(V_{\tau} - V_{\rm BRT})/V_{\rm BRT}$ and computation time t. Figure 3 indicates an inverse τ -accuracy trade-off: smaller τ reduces the volume gap but drives computation time up (roughly exponentially). Nonetheless, runtimes remain practical and safety is preserved for all tested parameters.

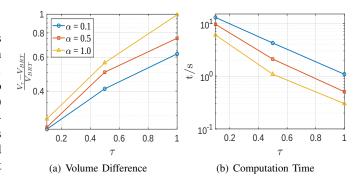


Fig. 3: Volume gap and computation time versus τ (and α); $n_{\rm s}{=}3000, r_{\rm min}{=}0.370.$

VII. CONCLUSION AND DISCUSSION

We introduced Recurrent Control Barrier Functions (RCBFs), generalizing CBFs by enforcing finite-time (τ) return rather than strict invariance. We proved that the signed distance to a τ -recurrent set is a valid RCBF, yielding rigorous safety guarantees. A sampling-based algorithm approximates the safe region. Simulations demonstrate provably safe, though over-approximated, sets with competitive computational performance.

An approximation gap persists between computed and true safe sets. While denser sampling improves accuracy, the precise link between sampling parameters and error remains open. Future work will quantify this relationship and develop corresponding models to guide adaptive sampling for tighter guarantees.

REFERENCES

- 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.
- [2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in 2019 18th European control conference (ECC), IEEE, 2019, pp. 3420–3431.
- [3] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamiltonjacobi reachability: A brief overview and recent advances," in 2017 IEEE 56th Annual Conference on Decision and Control (CDC), IEEE, 2017, pp. 2242–2253.
- [4] H. Dai and F. Permenter, "Convex synthesis and verification of control-lyapunov and barrier functions with input constraints," in 2023 American Control Conference (ACC), IEEE, 2023, pp. 4116–4123.
- [5] A. Clark, "Verification and synthesis of control barrier functions," in 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 6105–6112.
- [6] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 1817–1824.
- [7] C. Folkestad, Y. Chen, A. D. Ames, and J. W. Burdick, "Data-driven safety-critical control: Synthesizing control barrier functions with koopman operators," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2012–2017, 2020.

- [8] J. Lee, J. Kim, and A. D. Ames, "A data-driven method for safety-critical control: Designing control barrier functions from state constraints," in 2024 American Control Conference (ACC), IEEE, 2024, pp. 394–401.
- [9] R. Siegelmann, Y. Shen, F. Paganini, and E. Mallada, "A recurrence-based direct method for stability analysis and gpu-based verification of non-monotonic lyapunov functions," in 62nd IEEE Conference on Decision and Control (CDC), IEEE, Dec. 2023, pp. 6665–6672.
- [10] Y. Shen, H. Sibai, and E. Mallada, "Generalized barrier functions: Integral conditions & recurrent relaxations," in 60th Allerton Conference on Communication, Control, and Computing, Sep. 2024, pp. 1–8.
- [11] Y. Shen, M. Bichuch, and E. Mallada, "Model-free learning of regions of attraction via recurrent sets," in 61st IEEE Conference on Decision and Control (CDC), Dec. 2022, pp. 4714–4719.
- [12] H. Sibai and E. Mallada, "Recurrence of nonlinear control systems: Entropy and bit rates," in *Proceedings of the 27th ACM International Conference on Hybrid Systems: Com*putation and Control (HSCC), ser. HSCC '24, New York, NY, USA: Association for Computing Machinery, May 2024, pp. 1–9.
- [13] H. Sibai and E. Mallada, "Recurrence of nonlinear control systems: Entropy, bit rates, and finite alphabets," in *Nonlinear Analysis: Hybrid Systems*, Feb. 2025, pp. 1–16, submitted.
- [14] A. Clark, "A semi-algebraic framework for verification and synthesis of control barrier functions," *IEEE Transactions on Automatic Control*, 2024.
- [15] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop* on Hybrid Systems: Computation and Control, Springer, 2004, pp. 477–492.
- [16] W. Xiao et al., "Barriernet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, 2023.
- [17] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," in *Conference* on Robot Learning, PMLR, 2023, pp. 1970–1980.
- [18] O. So et al., "How to train your neural control barrier function: Learning safety filters for complex input-constrained systems," in 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2024, pp. 11532–11539.
- [19] StanfordASL, Hj_reachability: Hamilton-jacobi reachability analysis in jax, https://github.com/ StanfordASL/hj_reachability, 2024.
- [20] F. Bullo, Contraction Theory for Dynamical Systems, 1.2. Kindle Direct Publishing, 2024, ISBN: 979-8836646806.

APPENDIX

A. Proof of Lemma 1

Proof. According to the assumption of system (1), since the system (1) is uniformly continuous in u, thus $L_u = \max_{t \in (0,\tau]} \|g(\phi(t,x,u))\| := \max_{\|v\|=1,t \in (0,\tau]} \|g(\phi(t,x,u))v\| > 0$ exists. And the system (1) is uniformly continuous in u, and Lipschitz continuous in x for fixed control u, with a little abuse of the notation, for all the states x' and y' in trajectories $\phi(t,x,u)$ and $\phi(t,y,u)$, $\forall t \in (0,\tau]$ we have:

$$||F(x,u) - F(y,u)|| \le L||x - y||$$

$$||F(x,u) - F(x,v)|| = ||g(x)(u - v)|| \le L_u||u - v||$$

Thus, according to Corollary 3.17 and Grönwall Comparison Lemma in [20], we have:

$$\begin{aligned} &\|\phi(t,x,u) - \phi(t,y,u)\| \\ &\leq e^{Lt} \|x - y\| + L_u \int_0^t e^{L(t-s)} \|u(s) - u(s)\| ds \\ &= e^{Lt} \|x - y\| \\ &\leq re^{Lt}, \end{aligned}$$

where equality is held because two trajectories have the same input trajectory.

Suppose $x^* = \arg\min_{x^* \in \partial S} \operatorname{sd}(\phi(t, x, u), \mathcal{S}), y^* = \arg\min_{y^* \in \partial S} \operatorname{sd}(\phi(t, y, u), \mathcal{S})$. The three cases are analyzed as follows:

Case 1: $\phi(t,x,u)$ and $\phi(t,y,u)$ are both in S, and then we have

$$\begin{aligned} &|\mathrm{sd}(\phi(t,x,u),S) - \mathrm{sd}(\phi(t,y,u),S)| \\ = &|\|\phi(t,x,u) - x^*\| - \|\phi(t,y,u) - y^*\|| \\ \leq &|\|\phi(t,x,u) - x^*\| - \|\phi(t,y,u) - x^*\|| \\ \leq &\|\phi(t,x,u) - \phi(t,y,u)\| \\ \leq &re^{Lt}, \end{aligned}$$

where the first equality follows from the definition, and the first inequality follows from the triangle inequality.

Case 2: $\phi(t, x, u)$ and $\phi(t, y, u)$ are both not in S, and then with the same reason, similarly, we have

$$\begin{aligned} &|\mathrm{sd}(\phi(t,x,u),S) - \mathrm{sd}(\phi(t,y,u),S)| \\ = &|\|\phi(t,x,u) - x^*\| - \|\phi(t,y,u) - y^*\|| \\ \leq &|\|\phi(t,x,u) - x^*\| - \|\phi(t,y,u) - x^*\|| \\ \leq &\|\phi(t,x,u) - \phi(t,y,u)\| \\ \leq &re^{Lt}, \end{aligned}$$

Case 3: One of $\phi(t,x,u)$ and $\phi(t,y,u)$ is in S and the other not in S. Without loss of generality, we can assume that $\phi(t,x,u)$ is in S and $\phi(t,y,u)$ is not in S. Then there at least exists a $\lambda \in [0,1]$ such that $\lambda \phi(t,x,u) + (1-\lambda)\phi(t,y,u) \in \partial S$ and we denote $p^* := \lambda \phi(t,x,u) + (1-\lambda)\phi(t,y,u) \in \partial S$, thus, we have

$$\begin{aligned} &|\operatorname{sd}(\phi(t,x,u),S) - \operatorname{sd}(\phi(t,y,u),S)| \\ = &\|\phi(t,y,u) - y^*\| + \|\phi(t,x,u) - x^*\| \\ \leq &\|\phi(t,x,u) - p^*\| + \|\phi(t,y,u) - p^*\|| \\ = &\|\phi(t,x,u) - \phi(t,y,u)\| \\ \leq &re^{Lt}, \end{aligned}$$

where the first equality and the first inequality follow from the definition of the signed distance function, and the second equality follows from the definition of p^* . In all cases, we obtain $|\mathrm{sd}(\phi(t,x,u),S)-\mathrm{sd}(\phi(t,y,u),S)| \leq re^{Lt}$ as required. \square