A Barrier-Based Scenario Approach to Verify Safety-Critical Systems

Prithvi Akella, Aaron D. Ames[1]

Abstract— In this letter, we detail our randomized approach to safety-critical system verification. Our method requires limited system data to make a strong verification statement. Specifically, our method first randomly samples initial conditions and parameters for a controlled, continuous-time system and records the ensuing state trajectory at discrete intervals. Then, we evaluate these states under a candidate barrier function h to determine the constraints for a randomized linear program. The solution to this program then provides either a probabilistic verification statement or a counterexample. To show the validity of our results, we verify the robotarium simulator and identify counterexamples for its hardware counterpart. We also provide numerical evidence to validate our verification statements in the same setting. Furthermore, we show that our method is system-independent by performing the same verification method on a quadrupedal system in a multi-agent setting as well.

I. Introduction

It is natural to question the validity of controllers for safety-critical systems insofar as safety is of critical importance for these systems. Therefore, there has been a tremendous amount of work in the controls literature concerning both the development and verification of these controllers. On the developmental side, some work aims at learning or modifying existing control theoretic techniques, *e.g.* control barrier and Lyapunov functions, to iteratively develop better controllers that satisfy the desired safety objectives by default [1]–[6]. On a related note, there has also been interest in developing controllers against formal system specifications to ensure satisfactory operation as well [7]–[11]. For the sake of completeness, we have mentioned these works, although this paper will focus more on verification.

As controller verification typically does not restrict the verification analysis to a single control form, there are multiple ways this problem has been approached. One vein of work attempts to determine a Lyapunov or barrier function for the controlled system, to act as a certificate of system stability/safety [3], [12]–[17]. Due to their exploitation of existing control techniques to simplify the verification problem, works in this vein tend to be less sample complex than works in the next paradigm. This second paradigm expresses the verification question as an optimization problem whose solution corresponds to a counterexample or a (probabilistic) verification statement [18]–[23]. These works typically associate satisfactory behavior to positive evaluations of a robustness measure over system trajectories and aim to minimize this measure over a set of parameters of interest.

Each paradigm has its benefits and shortcomings. In the former case, the reduction in sample complexity arises
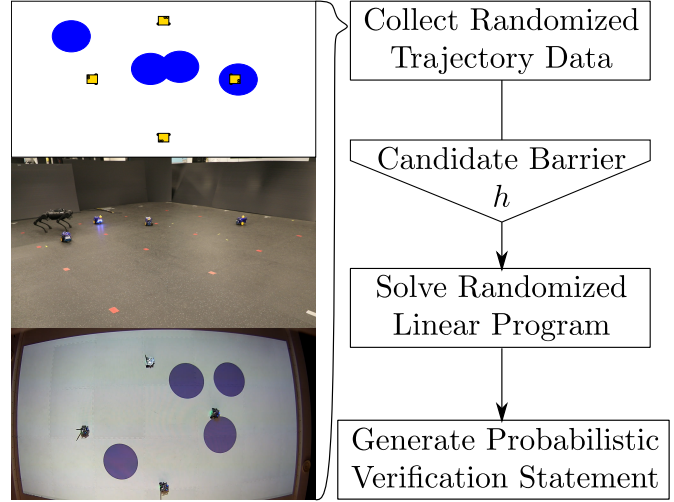
Fig. 1. An overview of the approach detailed within the paper. By collecting randomized state trajectory data of the system-to-be-verified and evaluating all transitions under a candidate barrier function h, we determine the constraints for a randomized linear program. Positivity of the solution to this program corresponds to a probabilistic verification statement regarding this system's ability to render positive the candidate barrier function h over its trajectories.

through *apriori* knowledge of the system dynamics and controller. Except for [16], these dynamics tend to only be functions of the system state and do not take into account extraneous parameters of interest, *e.g.* user-defined control objectives, human input through parameterization, *etc*. Additionally, *apriori* knowledge of the controller may not always be provided, especially if the system's controller is a complex, layered controller, *e.g.* the controller for an autonomous car, a quadruped, a bipedal exoskeleton, *etc*. The latter case permits more flexibility in this vein. Specifically, they only require the capacity to quantify system satisfaction of its objective through a robustness measure with positive robustness indicating objective satisfaction. Then, these methods try to determine the minimum robustness over a given set of parameters. However, they do not make as efficient use of existing control techniques due to their black-box system assumptions. As a result, these optimization problems suffer from poor performance in higher dimensions. Therefore, we aim to address both these shortcomings through our work at the intersection of these paradigms.

Our Contribution: We aim to utilize the benefits of both paradigms to address the shortcomings of the other. Our approach will focus on those safety-critical systems whose controllers vary with respect to a parameterized input, *e.g.* varying goal locations, obstacle locations, different control objectives, *etc*. For these systems, we will provide either a counterexample or probabilistic verification guarantee. More

specifically, our approach will use control barrier functions evaluated over randomly sampled system trajectories to inform the constraints for a randomized linear program. The solution to this program will identify a counterexample or provide for that probabilistic guarantee. Finally, we show the efficacy of validating our verification statements for a system simulator and its hardware counterpart. To preface our contribution, however, we will first introduce some necessary background information in the next section.

II. PROBLEM FORMULATION

This section will be split into two parts. The first part will detail some mathematical background information - discrete control barrier functions and scenario optimization. The second part will formally state the problem under study. To preface both parts, however, we will define some notation.

Notation: $\mathbb{Z}_+$ is the set of all positive integers, and $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$. $|A|$ is the cardinality of the set A.

A. Mathematical Preliminaries

Discrete Control Barrier Functions: First introduced by Ames et al. in [24] and built upon by Agrawal et al. in [25], discrete control barrier functions are a novel control tool designed to enforce forward invariance of their 0-superlevel sets for nonlinear discrete-time systems such as:

$$x_{k+1} = f(x_k, u_k), \ x \in \mathcal{X} \subset \mathbb{R}^n, \ u \in \mathcal{U} \subset \mathbb{R}^m.$$

Then, a discrete exponential control barrier function (DE-CBF) $h : \mathbb{R}^n \to \mathbb{R}$ is designed to classify those control inputs that maintain positivity of the same function in a specific manner. More accurately, if we define the 0-superlevel set of a candidate DE-CBF h as

$$\mathcal{C} = \{x \in \mathcal{X} \mid h(x) \geq 0\},$$

then the definition of a DE-CBF is as follows:

Definition 1 (Adapted from Definition 4 in [25])**.** *A function* $h : \mathbb{R}^n \to \mathbb{R}$ *is a discrete exponential control barrier function if the following inequality holds for some* $\gamma \in [0,1)$:

$$\forall \ x_k \in \mathcal{C}, \ \exists \ u \in \mathcal{U} \ \text{s.t.} \ h\left(f(x_k, u_k)\right) \geq \gamma h(x_k).$$

Then if we define the set of inputs that satisfy the CBF inequality in Definition 1 as

$$K(x) = \{u \in \mathcal{U} \mid h(f(x,u)) \geq \gamma h(x)\},$$

we have the following Theorem regarding forward invariance of the 0-superlevel set $\mathcal{C}$ of the DE-CBF h.

Theorem 1 (Adapted from Proposition 3 in [25])**.** *For a discrete exponential control barrier function* $h : \mathbb{R}^n \to \mathbb{R}$ *and it's 0-superlevel set* $\mathcal{C}$, *if* $x_0 \in \mathcal{C}$ *and all inputs* $u_k \in K(x_k) \ \forall \ k \in \mathbb{Z}_+$, *then* $x_k \in \mathcal{C} \ \forall \ k \in \mathbb{Z}_+$.

In what will follow, we will only reference discrete exponential control barrier functions and as such will simply refer to such functions as control barrier functions. Then, the overarching idea as to how we will utilize these functions for verification is to construct a linear program whose constraints

are the inequalities mentioned in Definition 1. The decision variable will be γ and the constraints will be randomly sampled from robot trajectories. While solving such an optimization problem will prove rather easy, guaranteeing that the solution has meaning over all trajectories is the subject of scenario optimization which will be detailed next.

Scenario Optimization: The brief description of scenario optimization in this section will stem primarily from the work done by Campi and Garrati in [26], [27]. Scenario optimization tries to identify robust solutions to uncertain convex optimization problems of the following form:

$$\begin{aligned} z^* = \underset{z \in \mathbb{Z} \subset \mathbb{R}^d}{\operatorname{argmin}} \quad & c^T z, \\ \text{subject to} \quad & z \in \mathbb{Z}_\delta, \ \delta \in \Delta. \end{aligned} \quad \text{(UP)}$$

Here, (UP) is the uncertain program as $\delta \in \Delta$ is an uncertain parameter with probability measure $\mathbb{P}$. Convexity is assured via assumed convexity in the spaces $\mathbb{Z}$ and $\mathbb{Z}_\delta$, and typically, $|\Delta| = \infty$. Hence, direct identification of a robust solution z^* such that $z^* \in \mathbb{Z}_\delta \ \forall \ \delta \in \Delta$ is usually infeasible.

To resolve this issue, the study of scenario optimization solves a related optimization problem formed from an N-sized sample of the constraints δ and provides a probabilistic guarantee on the robustness of the corresponding solution z_N^*. Specifically, if we were to take an N-sized sample of δ, $\{\delta_i\}_{i=1}^N$ - termed scenarios in the scenario optimization literature - we could construct the following scenario program:

$$\begin{aligned} z_N^* = \underset{z \in \mathbb{Z} \subset \mathbb{R}^d}{\operatorname{argmin}} \quad & c^T z, \\ \text{subject to} \quad & z \in \bigcap_{i=1,2,\ldots,N} \mathbb{Z}_{\delta_i}. \end{aligned} \quad \text{(RP-N)}$$

Then, we require the following assumption.

Assumption 1. The scenario program (RP-N) is solvable for any N-sample set $\{\delta_i\}_{i=1}^N$ and has a unique solution z_N^*.

For more information on why this assumption is made, we direct the reader to [26], [27]. Assumption 1 then guarantees existence of a scenario solution z_N^* for (RP-N). As such, we can define a set containing those constraints $\delta \in \Delta$ to which the scenario solution z_N^* is not robust, *i.e.*

$$F(z) = \{\delta \in \Delta \mid z \notin \mathbb{Z}_\delta\}.$$

With this set definition we can formally define the *violation probability* of our solution.

Definition 2. The *violation probability* $V(z)$ of a given $z \in \mathbb{Z}$ is defined as the probability of sampling a constraint δ to which z is not robust, *i.e.* $V(z) = \mathbb{P}[\delta \in F(z)]$.

Then, the main theorem is as follows:

Theorem 2 (Adapted from Theorem 1 in [26])**.** *Let Assumption 1 hold. The following inequality is true:*

$$\mathbb{P}^N[V(z_N^*) > \epsilon] \leq \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}.$$

In Theorem 2 above, N is the number of sampled constraints δ for the scenario program (RP-N); z_N^* is the scenario

solution to the corresponding scenario program; $V(z_N^*)$ is the violation probability of that solution as per Definition 2; d is the dimension in which z lies, *i.e.* $z \in \mathbb{R}^d$; and $\mathbb{P}^N$ is the induced probability measure over sets of N-samples of δ given the probability measure $\mathbb{P}$ for δ. For more information on the intersection of scenario optimization and control more generally, we direct the readers to [28], [29] and the citations within. With this information, we can now formally state the problem under study in the next section.

B. Problem Statement

We wish to verify safety properties for a given system without knowledge of the controller and/or dynamics. As such, we will assume our system is a continuous control system whose dynamics f and controller U are unknown, though U may depend on extraneous parameters θ.

$$\dot{x} = f(x, u), \ x \in \mathcal{X} \subset \mathbb{R}^n, \ u \in \mathcal{U} \subset \mathbb{R}^m,$$
$$u = U(x, \theta), \ \theta \in \Theta \subset \mathbb{R}^p, \quad (1)$$
$$\dot{x}(t, \theta) = f\left(x(t, \theta), U(x(t, \theta), \theta)\right), \ (x_0, \theta) \in \mathcal{X} \times \Theta.$$

As is common, $\mathcal{X}$ is the state space, $\mathcal{U}$ is the feasible control space, and Θ is the feasible parameter space. Furthermore, $x(t, \theta)$ corresponds to the solution to the closed-loop system at time t given the initial condition and parameter (x_0, θ). Notice that we do not allow the parameter θ to vary over the system's trajectory. Once chosen, the parameter θ is fixed.

To quantify our system's safety objective then, we will assume the capacity to measure system safety through evaluations of a candidate barrier function h at specific time instances $t_k = k\Delta t$ where $k \in \mathbb{Z}_+$ and $\Delta t > 0$.

Assumption 2. We have a barrier function $h : \mathcal{X} \times \Theta \to [-m, M]$, $m, M \in \mathbb{R}_+$ with 0-superlevel set $\mathcal{C} = \{(x, \theta) \in \mathcal{X} \times \Theta \mid h(x, \theta) \geq 0\}$. Furthermore, the system's safety objective is satisfied by ensuring continued positivity of h at time intervals t_k, *i.e.* by ensuring $h(x(t_k, \theta), \theta) \geq 0$, $\forall \ k \in \mathbb{Z}_+$ and $(x_0, \theta) \in \mathcal{C}$.

For context, this assumption is not too restrictive. Consider a simple example where the system is to avoid static obstacles that can be placed anywhere in a confined, 2-dimensional space. Then, defining h to be a function of the system state x and static obstacle location θ whose 0-superlevel set does not coincide with the obstacle suffices. Also note that we are not stating that the system objective is to ensure continued positivity of $h(x, \theta)$. Rather, we only assume that the system's objective is satisfied if $h(x, \theta)$ is kept positive, which is a significantly more relaxed assumption. In the latter case, for example, over-approximations of obstacle regions are valid. The former case would require specific knowledge of the obstacle locations which is significantly harder.

With these definitions and assumptions, our formal problem statement will follow.

Problem 1. *For the closed-loop system* (1) *and barrier function h as per Assumption 2, devise a method to verify whether $h(x(t_k, \theta), \theta) \geq 0 \ \forall \ k \leq K \in \mathbb{Z}_+$ and $(x_0, \theta) \in \mathcal{C}$.*

Ideally we would like to guarantee safety for all time indeces $k \in \mathbb{Z}_+$. However, as we intend to use a sampling approach, we require some finite time K by which to stop taking samples of the system trajectory. However, K can be any large positive integer. With our problem formally stated, we will now move describe our approach.

III. MAIN CONTRIBUTIONS

This section will be split into three parts. The first part will outline the overarching idea behind our approach; the second part will state and prove our main result, two lemmas, and a corollary; and the third part will provide numerical examples indicating that we can sample from a distribution integral to our approach. With that, our overarching idea will follow.

Overarching Idea: As stated, we want to verify whether a given safety-critical system ensures continued positivity of a candidate barrier function h at specific time instances t_k as per Assumption 2. To make this verification statement, if we could identify a safety decay rate $\gamma \in [0, 1)$ satisfying the following inequality for this candidate barrier function h:

$$h(x(t_1, \theta), \theta) \geq \gamma h(x_0, \theta), \ (x_0, \theta) \in \mathcal{C}, \quad (2)$$

then we could directly use Theorem 1 to make our desired verification statement. However, checking the veracity of the inequality in (2) would require checking this condition for every possible trajectory emanating from initial conditions $(x_0, \theta) \in \mathcal{C}$. With there being a (potentially) infinite amount of them, this is infeasible.

However, we note that we can rephrase the identification of γ into an optimization problem with infinite constraints:

$$\gamma^* = \underset{\gamma \in \mathbb{R}}{\operatorname{argmax}} \quad \gamma, \quad \text{(BASE-OPT)}$$
$$\text{subject to} \quad h(x(t_1, \theta), \theta) \geq \gamma h(x_0, \theta),$$
$$\forall \ (x_0, \theta) \in \mathcal{C}.$$

If the solution γ^* to (BASE-OPT) were positive, then the inequality in (2) would be true. To relax the number of constraints for (BASE-OPT) and yield a solvable optimization problem, we will instead randomly sample the constraints and generate a scenario program.

This transformation of (BASE-OPT) to an uncertain program for which we can guarantee a scenario solution is the crux of our approach. In doing so, we will randomly sample over feasible robot trajectories and measure system safety throughout. Every measurement will provide a new constraint to the scenario program. Then, a positive scenario solution γ_N^* to the corresponding scenario program constitutes successful maintenance of the inequality (2) with high probability. A negative solution will identify a counterexample. With this overarching idea in mind, we will now be more specific in our statement of our main contributions.

A. Main Results

As mentioned, our approach involves transforming (BASE-OPT) to an uncertain program which we will

solve. In keeping with the notation for scenario optimization utilized earlier, our uncertain program is as follows:

$$\gamma^* = \underset{\gamma \in \mathbb{R}}{\mathrm{argmax}} \quad \gamma, \qquad \text{(BASE-UP)}$$

$$\text{subject to} \qquad \gamma \in \Gamma_\delta, \ \delta = (x_k, x_{k+1}, \theta) \in \Delta.$$

Here, Δ and Γ_δ are as follows:

$$\Delta = \{(x_k, x_{k+1}, \theta) \in \mathcal{X} \times \mathcal{X} \times \Theta \mid h(x_0, \theta) \geq 0\},$$

$$\Gamma_\delta = \left\{ \gamma \in \mathbb{R} \ \middle| \ h(x_{k+1}, \theta) \geq \gamma \, |h(x_k, \theta)| \right\} \qquad (3)$$

Specifically, Δ is the set of transitions x_k to x_{k+1} for trajectories whose initial condition and parameter (x_0, θ) start in the 0-superlevel set of the candidate barrier function h. Γ_δ is the set of all $\gamma \in \mathbb{R}$ that satisfy an inequality similar to (2) for the specific transition x_k to x_{k+1} encoded by δ. The discrepancy with (2) is the absolute value over $h(x_k, \theta)$ - the reason for which will be elucidated in a lemma to follow.

Furthermore, for (BASE-UP) to be an uncertain program, δ must be a random variable with an associated probability distribution π. As such, we will define the probability of sampling any $\delta \triangleq (x_k, x_{k+1}, \theta)$ as the probability of sampling an initial condition and parameter $(x_0, \theta) \in \mathcal{C}$ such that the corresponding closed-loop trajectory to (1) contains the transition encoded by δ. To formalize this distribution π, we can define the indicator function $\mathbb{1}_\delta$ for a given transition $\delta = (x_k, x_{k+1}, \theta_d)$. This function evaluates to 1 for any initial condition and parameter pair (x_0, θ) such that the corresponding closed-loop trajectory to (1) contains the transition specified by δ within K timesteps.

$$\mathbb{1}_\delta(x_0, \theta) = \begin{cases} 1 & \text{if} \begin{cases} \theta = \theta_d, \text{ and,} \\ \exists \, k, k+1 \text{ s.t. } 0 \leq k, k+1 \leq K, \\ x(t_k, \theta) = x_k, \ x(t_{k+1}, \theta) = x_{k+1}, \end{cases} \\ 0 & \text{else.} \end{cases}$$

Then, if we sample initial conditions and parameters (x_0, θ) with a uniform distribution over $\mathcal{C}$, we can implicitly define the probability distribution function π for the random variable δ. In what follows, s is a normalization constant to ensure the total probability integrates to 1 and $\beta = (x_0, \theta)$:

$$\mathbb{P}_{\pi(\delta)}[A \subset \Delta] = \int_A \int_\mathcal{C} \frac{\mathbb{1}_\delta(\beta)}{s} d\beta d\delta. \qquad (4)$$

While it is currently unclear whether we can sample from this distribution π, we will show we can through a few examples in the section to follow. Intuitively though, if we uniformly sample initial condition and parameter pairs (x_0, θ) and record the state trajectory at time-steps $t_k \ \forall \ k \leq K$, the corresponding transitions will be samples of δ from the proposed distribution π.

Under the assumption that we can take N samples δ with corresponding probability distribution π, we can generate a scenario program for (BASE-UP) as follows:

$$\gamma_N^* = \underset{\gamma \in \mathbb{R}}{\mathrm{argmax}} \quad \gamma, \qquad \text{(BASE-RP-N)}$$

$$\text{subject to} \qquad \gamma \in \Gamma_{\delta_i}, \ \forall \ \delta_i \in \{\delta_i\}_{i=1}^N.$$

For a solution γ_N^* to (BASE-RP-N) we can define an associated violation set $F(\gamma)$ and violation probability $V(\gamma)$.

$$F(\gamma) = \{\delta \in \Delta \mid \gamma \notin \Gamma_\delta\}, \quad V(\gamma) = \mathbb{P}_{\pi(\delta)}[F(\gamma)]. \quad (5)$$

Intuitively, $F(\tilde{\gamma})$ is the set of all transitions δ where the system decays to an unsafe behavior quicker than the minimum decay rate identified by $\tilde{\gamma}$ through inequality (2). $V(\tilde{\gamma})$ is the probability of sampling such transitions from π.

Description and Statement of Results: This puts into place all required notation for our main results to follow. Succinctly, we will prove that we can always solve (BASE-RP-N) for any N-sample set of transitions $\{\delta_i\}_{i=1}^N$, and that any solution γ_N^* will be unique. This statement and proof will be formalized through Lemma 1. Then, through Lemma 2 we will prove that the violation probability of our solution $V(\gamma_N^*)$ corresponds to the probability of sampling an unsafe trajectory when uniformly sampling initial conditions and parameters $(x_0, \theta) \in \mathcal{C}$. Then, our main result will use both prior Lemmas and Theorem 2 to lower bound the probability of sampling safe trajectories over all possible initial conditions and parameters $(x_0, \theta) \in \mathcal{C}$ if the solution γ_N^* to (BASE-RP-N) is positive. Finally, Corollary 1 will extend Lemma 2 and state that any negative solution to (BASE-RP-N) corresponds to a counterexample. We will now state these results.

Lemma 1. *Let Assumption 2 hold. The scenario program* (BASE-RP-N) *is always solvable for any N-sample set of transitions $\{\delta_i\}_{i=1}^N$, and the solution γ_N^* is unique.*

Lemma 2. *The violation probability $V(\gamma_N^*)$ for a solution γ_N^* to* (BASE-RP-N) *is equivalent to the probability of uniformly sampling over $\mathcal{C}$ initial conditions and parameters (x_0, θ) whose corresponding trajectory evolves within K time-steps to a transition with a faster safety decay rate than γ_N^*, i.e. with $x_k^\theta = x(t_k, \theta)$,*

$$V(\gamma_N^*) = \mathbb{P}_{\mathrm{U}[\mathcal{C}]} \left[(x_0, \theta) \ \middle| \ \begin{matrix} \exists \, k, k+1 \text{ s.t.} \\ 0 \leq k, k+1 \leq K, \text{ and,} \\ h(x_{k+1}^\theta, \theta), \theta) < \gamma_N^* \left| h(x_k^\theta, \theta) \right| \end{matrix} \right].$$

Theorem 3. *Let Assumption 2 hold, let the scenario program* (BASE-RP-N) *be composed from an N-sample set of transitions $\{\delta_i\}_{i=1}^N$, and let $\epsilon \in [0, 1]$. If $\gamma_N^* \geq 0$, then the following statement is true, with $x_k^\theta = x(t_k, \theta)$:*

$$S(\gamma_N^*) \triangleq \mathbb{P}_{\mathrm{U}[\mathcal{C}]} \left[(x_0, \theta) \mid h(x_k^\theta, \theta) \geq 0, \ \forall \ k = 0, 1, \ldots K \right],$$

$$\mathbb{P}_{\pi(\delta)}^N \left[S(\gamma_N^*) \geq 1 - \epsilon \right] \geq 1 - (1 - \epsilon)^N.$$

Corollary 1. *Let Assumption 2 hold and let the scenario program* (BASE-RP-N) *be composed from an N-sample set of transitions $\{\delta_i\}_{i=1}^N$. If $\gamma_N^* < 0$, then there exists a transition $\delta \in \{\delta_i\}_{i=1}^N$ that corresponds to a safety violation, i.e. $\exists \ \delta \in \{\delta_i\}_{i=1}^N$ s.t. $h(x_{k+1}^\theta, \theta) < 0$ with $x_k^\theta = x(t_k, \theta)$ and $\delta = (x_k^\theta, x_{k+1}^\theta, \theta)$.*

We can summarize Theorem 2 as follows. If Assumption 2 holds, the corresponding scenario program is formed from N samples, and $\gamma_N^* \geq 0$, then the probability of uniformly
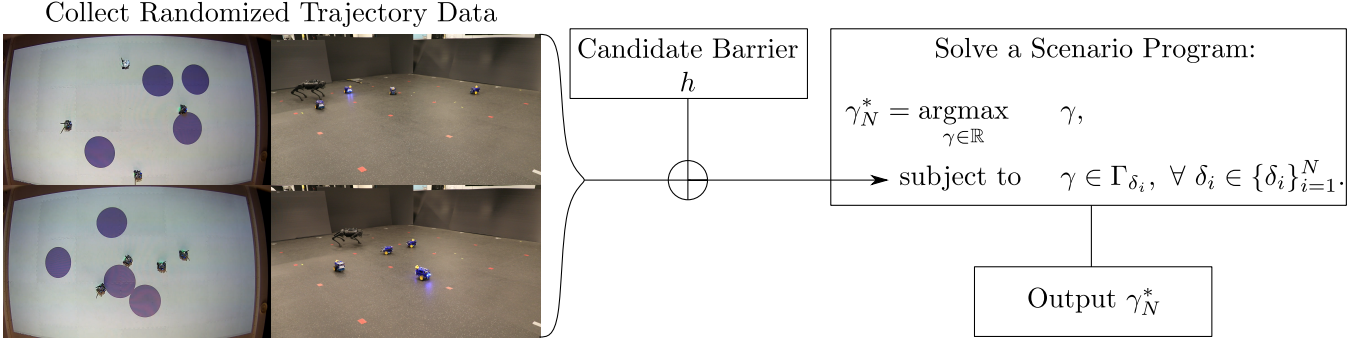
Fig. 2. Shown above is an overview of the process detailed in the paper. By recording state trajectory samples of randomly chosen trajectories and evaluating the transitions under a candidate barrier function h, we inform the constraints for a randomized linear program that identifies the minimum discrete-time decrement condition γ_N^*. As per Theorem 3 if $\gamma_N^* \geq 0$, then we prove that the corresponding system renders the candidate barrier function h a control barrier function with high probability. In other words, if $\gamma_N^* \geq 0$, then the system maintains positivity of h with high probability.

sampling over $\mathcal{C}$ initial conditions and parameters (x_0, θ) such that their corresponding trajectories remain safe for at least K time-steps can be lower bounded with high probability. We will now prove these results.

B. Proofs of Main Results

We will start first with the proof for Lemma 1.

Proof: To start, by definition of the scenario program (BASE-RP-N) and the constraint sets Γ_δ in equation (3), for any N-sized sample set of transitions $\{\delta_i\}_{i=1}^N$, the scenario program (BASE-RP-N) is a linear program maximizing the decision variable γ subject to a set of upper bounds. It is for this reason we required the absolute value over $h(x_k, \theta)$ in (3). Without the absolute value, there could exist a case where the system evolves to a state where $h(x_k, \theta) < 0$, which has the potential of yielding an unsatisfiable set of constraints. With the absolute value, proving solvability of the associated program requires ensuring that all upper bounds are strictly less than infinity.

This arises as evaluations of h are restricted to lie within $[-m, M]$, $M, m \in \mathbb{R}_+$ by Assumption 2. As such, there is guaranteed to be at least one such upper bound $\tilde{\gamma} < \infty$. This guarantees a solution to the corresponding linear program with the solution guaranteed to be unique as it is a solution to a linear program. This neglects to consider those trajectories that eventually end up in a state x_k such that $h(x_k, \theta) = 0$. However, the set of all trajectories that land in the set $h(x_k, \theta) = 0$ for some $k = 0, 1, \ldots, K-1$ is a set of measure 0 with respect to the probability distribution π. Therefore, we can safely neglect such trajectories. ∎

Lemma 1 effectively acts as a disclaimer permitting us to utilize the results of Theorem 2 to bound the violation probabilities of results to our scenario program (BASE-RP-N). This will be useful, for as stated in Lemma 2, this violation probability is equivalent to the probability of sampling marginally "more unsafe" trajectories. The proof for Lemma 2 will follow.

Proof: We will start with the definition of the violation probability for our optimal solution $V(\gamma_N^*)$ as per (5).

$$V(\gamma_N^*) = \mathbb{P}_{\pi(\delta)}[\delta \in \Delta \mid \gamma_N^* \notin \Gamma_\delta].$$

Here, Γ_δ is defined in equation (3). For this proof, it will be useful to define the following indicator function:

$$\neg \mathbb{1}_{\Gamma_\delta}(\gamma) = \begin{cases} 1 & \text{if } \gamma \notin \Gamma_\delta, \\ 0 & \text{else.} \end{cases}$$

Then we can rewrite the violation probability as follows:

$$V(\gamma_N^*) = \int_\Delta \pi(\delta) \neg \mathbb{1}_{\Gamma_\delta}(\gamma_N^*) d\delta.$$

By definition of our probability distribution π in equation (4), we can rewrite the above equation with $\beta = (x_0, \theta)$:

$$V(\gamma_N^*) = \int_\Delta \int_\mathcal{C} \frac{\mathbb{1}_\delta(\beta) \neg \mathbb{1}_{\Gamma_\delta}(\gamma_N^*)}{s} d\beta d\delta.$$

However, the interior integrand only evaluates to 1 for those initial condition and parameter pairs (x_0, θ) such that there exist time-steps $k, k+1$ where $0 \leq k, k+1 \leq K$ and $h(x_{k+1}^\theta, \theta) < \gamma_N^* |h(x_k^\theta, \theta)|$. Here, $x_k^\theta = x(t_k, \theta)$. As such, the above probability corresponds to the probability of sampling such initial condition and parameter pairs from the uniform distribution over $\mathcal{C}$. This completes the proof. ∎

Lemma 2 effectively states that the violation probability for our scenario program (BASE-RP-N) corresponds to picking those trajectories that are marginally "more unsafe" than the worst-case sampled trajectory. In other words, one minus the violation probability then corresponds to the probability of picking those trajectories that are at least as safe as the worst-case sampled trajectory. This notion is formalized in Theorem 3 the proof for which will follow.

Proof: With the assumptions behind Theorem 3, we know that Lemma 2 holds. This let's us define the success probability $S(\gamma_N^*) = 1 - V(\gamma_N^*)$. Mathematically this success probability is defined as follows with $x_k^\theta = x(k, \theta)$:

$$S(\gamma_N^*) = \mathbb{P}_{\mathrm{U}[\mathcal{C}]} \left[(x_0, \theta) \, \middle| \, \begin{array}{l} \forall \, k = 0, 1, \ldots, K-1 \\ h(x_{k+1}^\theta, \theta) \geq \gamma_N^* |h(x_k^\theta, \theta)| \end{array} \right].$$

For all such sampled trajectories, $h(x_0^\theta, \theta) \geq 0$ as the initial condition and parameter (x_0, θ) are sampled uniformly over the 0-superlevel set of h $\mathcal{C}$. As a result, in light of Theorem 1 we can rewrite the condition for the success probability:

$$S(\gamma_N^*) = \mathbb{P}_{\mathrm{U}[\mathcal{C}]} \left[(x_0, \theta) \, | h(x_k^\theta, \theta) \geq 0, \, \forall \, k = 0, 1, \ldots, K \right].$$

This satisfies the first equality for Theorem 3.

For the inequality in Theorem 3, we first note that Lemma 1 permits us to use the results of Theorem 2. This lets us upper bound the violation probability to high confidence. Note that for our problem $d = 1$ which lets us simplify the right hand side of the inequality in Theorem 2. Specifically, for some $\epsilon \in [0, 1]$,

$$\mathbb{P}_{\pi(\delta)}^N \left[V(\gamma_N^*) \leq \epsilon \right] \geq 1 - (1-\epsilon)^N.$$

Then the final result holds due to definition of the success probability $S(\gamma_N^*)$.

$$\mathbb{P}_{\pi(\delta)}^N \left[S(\gamma_N^*) \geq 1 - \epsilon \right] \geq 1 - (1-\epsilon)^N. \qquad \blacksquare$$

This ends the proof for our main result - that if the solution to our randomized linear program (BASE-RP-N) is positive, *i.e.* $\gamma_N^* \geq 0$, then with high probability the system maintains positivity of the candidate barrier function h for at least K time-steps. What if $\gamma_N^* < 0$, however? Corollary 1 indicates that such a scenario corresponds to a counterexample and its proof will follow.

Proof: The proof for this corollary stems primarily from the definition of the constraint spaces Γ_δ in (3). Specifically, by Lemma 1, we know a solution to (BASE-RP-N) must exist for any sample set of transitions $\{\delta_i\}_{i=1}^N$. As mentioned in the proof for Lemma 1 this is primarily due to the fact that for any set of samples, (BASE-OPT) is a linear program maximizing a scalar decision variable γ subject to a set of upper bounds b_i. If the solution $\gamma_N^* < 0$ this implies that at least one upper bound $b_i < 0$. Based on definition of the constraint space Γ_δ then, this implies that

$$\exists\, \delta = (x_k^\theta, x_{k+1}^\theta, \theta) \in \{\delta_i\}_{i=1}^N \;\; \text{s.t.} \;\; \frac{h\left(x_{k+1}^\theta, \theta\right)}{\left| h\left(x_k^\theta, \theta\right) \right|} < 0.$$

As the denominator for the associated fraction is always positive, this implies that $\exists\, \delta \in \{\delta_i\}_{i=1}^N$ such that $h(x_{k+1}^\theta, \theta) < 0$, concluding the proof. $\blacksquare$

This concludes all proofs for our results. As mentioned, however, these results hinge on the capacity to take samples of the random variable δ with distribution π. The following section will show a few examples indicating that we can sample from our proposed distribution.

C. Sampling from our Proposed Distribution

As mentioned earlier, it is unclear whether we can sample from our proposed distribution π as defined in equation (4). However, we offered a method to take samples from this distribution. Our method first uniformly randomly samples the initial condition and parameter pair (x_0, θ) from $\mathcal{C}$ and records the resulting state trajectory at time-steps t_k, $\forall\, k = 0, 1, \ldots, K$. This section will show a few numerical examples indicating that this method does produce samples of δ distributed by π.

We will first provide three simple systems to act as a replacement for the continuous system we are trying to verify. The first will be a system that continuously oscillates around the perimeter of a circle with radius $r = 1$, the other will be a system that exponentially decays to 0, and the third

will be a system that exponentially decays to a parameterized point $\theta \in [-1.5, 0, 1.5]$. We will not mention their ODEs for motion and only mention their solutions.

$$x = [1, \phi]^T, \qquad\qquad x(t) = [1, \phi_0 + 0.1t], \quad \text{(a)}$$
$$x = [x], \qquad\qquad\qquad x(t) = [e^{-0.5t}]. \qquad\qquad \text{(b)}$$
$$x = [x],\; \theta \in [-1.5, 0, 1.5], \;\; x(t) = [e^{-3t} + \theta]. \qquad \text{(c)}$$

For sampling purposes then, the respective sample spaces per system are as follows:

$$\text{for (a) } \mathcal{C} = \mathcal{X} = [0, 2\pi],$$
$$\text{for (b) } \mathcal{C} = \mathcal{X} = [-1, 1], \qquad\qquad\qquad (6)$$
$$\text{for (c) } \mathcal{C} = \mathcal{X} \times \Theta = [-3, 3] \times [-1.5, 0, 1.5].$$

Per our method then, we will take 500 trials of each system, forward simulating each system $K = 200$ steps per trial with $\Delta t = 0.05$. For (c) we will increase the trial number to 1000 trials as we are parameterizing the system via θ as well. To generate these trials, we will uniformly randomly sample an initial condition (and parameter θ for (c)) with the spaces shown in equation (6). For data portrayal purposes, we will show in Figure 3 the initial states x_k for each sampled transition $\delta = (x_k, x_{k+1}, \theta)$, as they suffice to showcase our method's ability to capture the intended distribution π.

For system (a) if we uniformly sample initial conditions and forward simulate the same number of steps for each initial condition, we expect the initial states x_k for each transition to follow a uniform distribution. This is indeed the case as seen in the left figure in Figure 3. For system (b), we expect the initial states x_k to be localized to and symmetric about 0, with an exponentially higher rate of samples closer to 0 than farther out. This harmonizes with the numerical results shown in the center figure in Figure 3. Finally, for system (c), we expect a response similar to that for system (b) but for three different "peaks" centered on the choice of $\theta \in [-1.5, 1, 1.5]$. As seen in the right figure in Figure 3, this is indeed the case. Therefore, these examples show that we can sample from our proposed distribution π with the method we describe, and will now use this method to verify a system simulator and its hardware counterpart.

IV. EXAMPLES

In this section, we will verify or find counterexamples for systems with pre-existing controllers. Furthermore, we will provide numerical evidence that the stated inequality in Theorem 3 is true. We will start with verifying the Robotarium simulator [30] which will provide numerical results supporting the results of Lemma 2 and Theorem 3.

A. Verifying the Robotarium Simulator

The robots in the robotarium are modeled via unicycle dynamics which are as follows:

$$x = \begin{bmatrix} x, \\ y, \\ \theta \end{bmatrix}, \;\; \dot{x} = \begin{bmatrix} v\cos(\theta), \\ v\sin(\theta), \\ \omega, \end{bmatrix}, \;\; u = [v, \omega]^T,$$
$$\mathcal{X} = [-1.2, 1.2] \times [-0.6, 0.6] \times [0, 2\pi], \;\; P = [I_2, \; \mathbf{0}_{2x1}]$$

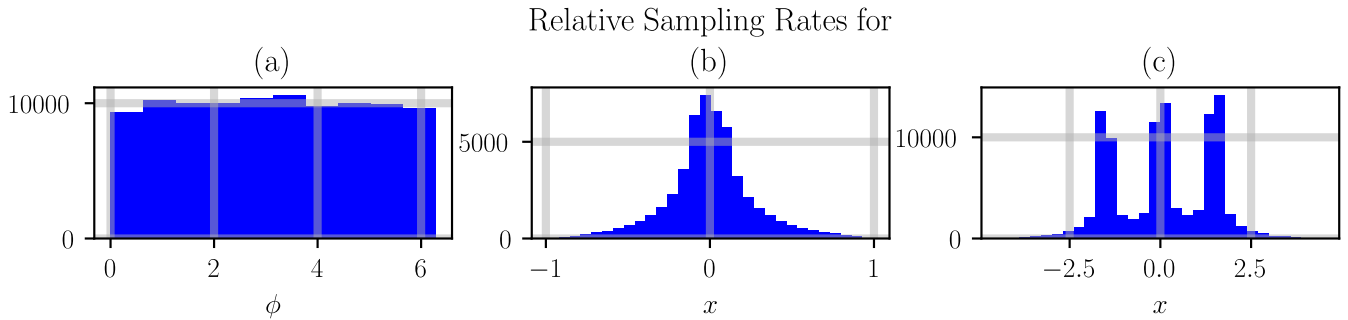Relative Sampling Rates for

(a) (b) (c)

Fig. 3. Relative sampling rates of the initial state x_k for transitions recorded by our uncertain parameter $\delta = (x_k, x_{k+1}, \theta)$. The systems for which these transitions are sampled are given in equations (a)-(c). In each case, however, our proposed sampling method does produce numerical estimates for the distribution of the initial states x_k that align with our expectations based on definition of our proposed distribution π in equation (4).

Each robot in the robotarium has a Lyapunov-based controller that drives it from its current position to a desired orientation x_d in its state space $\mathcal{X}$. When multiple robotarium robots are asked to ambulate in the same, confined space, their control inputs are filtered in a barrier-based quadratic program to ensure that the robots never collide [24]. As such, given a nominal radius r_s that the robots are to maintain, a candidate barrier function h would be

$$h(x^1, x^2, \ldots, x^{N_R}) = \min_{i \neq j, \ i,j \in [1,2,\ldots,N_R]} \|P(x^i - x^j)\| - r_s.$$

Concatenating all the state vectors of the N_R robots in the robotarium in $\mathbf{x}^T = [x^{1T}, x^{2T}, \ldots x^{NT}]$, we get the following candidate barrier function required of Assumption 2:

$$h(\mathbf{x}) = \min_{i \neq j, \ i,j \in [1,2,\ldots,N_R]} \|P(x^i - x^j)\| - r_s. \quad (7)$$

This results in the following verification problem. For the closed-loop robotarium simulator, devise a method to determine whether $h(\mathbf{x}(t_k, \mathbf{x}_d)) \geq 0 \ \forall \ k \leq K = 100 \in \mathbb{Z}_+$ and $(\mathbf{x}_0, \mathbf{x}_d) \in \mathcal{C}$. Here, h is as per (7), our parameter $\theta^T = [x_d^{1T}, x_d^{2T}, \ldots x_d^{N_R}] \in \Theta = \mathcal{X}^{N_R}$, x_d^j is the desired pose for robot j, and $N_R = 3$.

Numerical Results: To determine the safety of the robotarium simulator, we sampled $N_0 = 100$ initial conditions and desired poses $(\mathbf{x}_0, \mathbf{x}_d)$ from the uniform distribution over the 0-superlevel set of the candidate barrier function h U[$\mathcal{C}$]. Then, we simulated each closed-loop trajectory for $K = 100$ time-steps with $\Delta t = 0.03$ and recorded all transitions $\delta = (\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{x}_d)$. Then, we calculated the minimum decay constant γ_N^* as per (BASE-RP-N) for the $N = 10000$ transition samples taken, resulting in $\gamma_N^* \approx 0.953 \geq 0$.

As per Lemma 2 and Theorem 3 then, the probability of sampling a violating initial condition and goal $(\mathbf{x}_0, \mathbf{x}_d)$ from U[$\mathcal{C}$] can be upper bounded by some $\epsilon \in [0, 1]$. More accurately, Theorem 3 states that the probability that the system maintains positivity of h for at least $K = 100$ time-steps can be lower bounded as follows:

$$\mathbb{P}^N_{\pi(\delta)}[S(\gamma_N^*) \geq 1 - \epsilon] \geq 1 - (1 - \epsilon)^N.$$

To determine the high probability lower bound $1 - \epsilon$, we set the right-hand side of the outer probability to be $= 1 - 10^{-6}$ and calculate the corresponding violation probability upper bound ϵ that satisfies this inequality with $N = 10000$, the

TABLE I

ROBOTARIUM SIMULATOR VERIFICATION DATA

N_0	K	N	γ_N^*	ϵ	$V(\gamma_N^*)$	$S(\gamma_N^*)$
100	100	10000	0.953	0.0014	≈ 0	≈ 1

number of samples taken. This results in an upper bound $\epsilon = 0.0014$. In other words, according to the results of Theorem 3, the probability of sampling an initial condition and goal pair $(\mathbf{x}_0, \mathbf{x}_d)$ from U[$\mathcal{C}$] such that the corresponding trajectory does not maintain positivity of h for at least $K = 100$ time-steps should be lower than $\epsilon = 0.0014$ with minimum probability $1 - 10^{-6}$.

To verify this last statement, we sampled $N_0 = 50000$ initial condition and goal pairs $(\mathbf{x}_0, \mathbf{x}_d)$ from U[$\mathcal{C}$], simulated each trajectory for $K = 100$ timesteps, and recorded the minimum barrier value ℓ for the sampled trajectory. We used the fraction of trajectories with minimum barrier value $\ell < 0$ to approximate the probability of sampling a trajectory that does not maintain positivity of h for at least 100 time-steps. The complete information for this verification process and validation of our verification method can be found in Table I. As shown in this information, the true violation probability estimate is indeed lower than our calculated upper bound ϵ. This serves as numerical validation of our verification method, at least with respect to the Robotarium simulator.

Verifying our Scenario Approach: The aforementioned results provide numerical evidence supporting the results of Theorem 3 and Lemma 2. However, this does not show repeatability of our results. Specifically, we state via use of the scenario approach that the violation probability of our calculated solution γ_N^* can be upper bounded with high probability with respect to the distribution π by which transition samples δ are drawn, i.e.

$$\mathbb{P}^N_{\pi(\delta)}[V(\gamma_N^*) \leq \epsilon] \geq 1 - (1 - \epsilon)^N.$$

To show the above statement holds, we performed the same verification procedure as prior 50 separate times and recorded the calculated minimum safety decay rate γ_N^* each time. To show that our results are also system independent, we performed the same procedure with a four-agent system as well and extended the maximum number of time-steps to $K = 200$ in the four-agent case. Then, for each verification
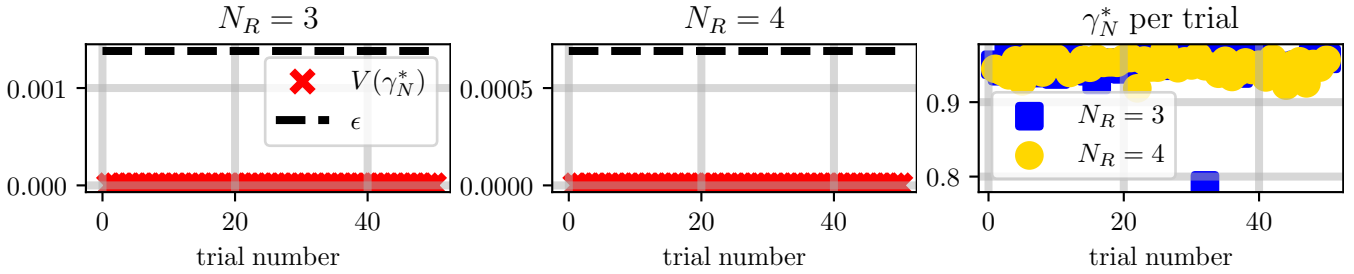
Fig. 4. Compilation of results for each of the 50 trial runs of verifying the robotarium simulator with $N_R = 3, 4$ robots. A complete explanation of the results is in Section IV-A. Notably, however, over all 100 separate trials performed, the true violation probability of our calculated decay constant γ_N^* is always less than our theorized upper bound ϵ. Just to note, while it may seem like the true violation probability is equal to 0 in all cases, it is not. The variance in the true violation probability is on the order of 10^{-5} and is just not visible at the scale shown.

attempt, we calculated the true violation probability of the our solution γ_N^* by uniformly sampling another $N_v = 1000$ initial condition and goal pairs and simulating the corresponding trajectories for the appropriate number of time-steps as well - $K = 100$ for the three-agent case $N_R = 3$ and $K = 200$ for the four-agent case $N_R = 4$. Then, we recorded the fraction of transitions δ that required decay constants $\gamma < \gamma_N^*$ as an estimate of the true violation probability of our solution γ_N^*. Finally, we calculated the high probability upper bound ϵ to our violation probability $V(\gamma_N^*)$ by setting $1 - (1 - \epsilon)^N = 1 - 10^{-6}$ and calculated the ϵ satisfying this condition. For $N_R = 3$ $\epsilon = 0.0014$ and for $N_R = 4$ $\epsilon = 0.0007$. This discrepancy in ϵ arises as we changed the maximum simulation time-step from $K = 100$ to $K = 200$ from the three to four-agent case respectively.

All information for this procedure can be found in Figure 4. Notably, over all 100 trials performed, the calculated minimum decay constant $\gamma_N^* \geq 0$. This acts as further numerical support for the results of Lemma 2 and Theorem 3. Specifically, our prior results indicated that with high probability, uniformly sampled trajectories should consistently render positive the candidate barrier function h for at least 100 time-steps. Indeed, over all 100 trials performed wherein each trial 100 trajectories were sampled, the candidate barrier function h stayed positive for at least 100 time-steps. Furthermore, for all 100 trials, the true violation probability $V(\gamma_N^*) < \epsilon$ - our calculated upper bound. This shows the repeatability and accuracy of our verification attempts at least with respect to the Robotarium simulator. The next section aims to show similar results on hardware systems as well.

B. Hardware Verification with Limited Data

In this section, we will verify two hardware systems, the Robotarium, and the Unitree A1 Quadruped steered by a variant of the controller presented in [31]. The robotarium experiments will provide further numerical validation of our scenario approach to verification. Namely, we will calculate a minimum safety decay constant γ_N^* based on 100 randomly sampled hardware trajectories and show the true violation probability of this solution $V(\gamma_N^*)$ is indeed upper bounded as stated via our approach. The quadruped examples show how we can also make this verification statement with limited data independent of the system-to-be-verified.

Robotarium: The setting for the hardware version of the

TABLE II
ROBOTARIUM HARDWARE VERIFICATION DATA

N_0	K	N	γ_N^*	ϵ	$V(\gamma_N^*)$	$S(\gamma_N^*)$
100	100	10000	-3.057	0.0014	≈ 0	≈ 1

robotarium is the same as mentioned in the simulation section prior for the three-agent case. Our goal will be to verify the same probabilistic statement as prior: $\mathbb{P}_{\pi(\delta)}^N [V(\gamma_N^*) \leq \epsilon] \geq 1 - (1 - \epsilon)^N$. Here, π is now the unknown distribution from which transitions δ of the hardware system are drawn. To generate our minimum safety decay rate γ_N^*, we sampled $N_0 = 100$ initial condition and goal pairs, recorded the resulting trajectory for $K = 100$ time-steps, and calculated γ_N^* as per (BASE-RP-N) - this is the same procedure as we followed in the simulation case. To verify our corresponding probabilistic result, we also sampled $N_v = 400$ initial condition and goal pairs, recorded the resulting trajectory for $K = 100$ time-steps, and recorded the fraction ℓ of transitions δ requiring a safety decay rate $\gamma < \gamma_N^*$. We used this fraction ℓ as an approximation of the true violation probability. We also calculated the violation probability upper bound ϵ by setting the right-hand side of the earlier probability statement to $1 - 10^{-6}$ as prior. This yields an upper bound $\epsilon = 0.0014$. All this information can be found in Table II.

Notice that in this case, the procedure identified a counterexample as $\gamma_N^* < 0$, and as per Corollary 1, there should exist a transition δ in the sampled set whereby the system evolves to make negative the candidate barrier h. This is indeed the case as shown in Figure 5, as for one of the trials, the minimum value of the barrier function h was negative. That being said, over the other 400 trajectories sampled, none exhibited a transition corresponding to a safety decay rate $\gamma < \gamma_N^*$. As a result, we estimate that the true violation probability $V(\gamma_N^*) < \epsilon$ our calculated upper bound - a statement which we expected to hold with high probability, and it indeed does in this case.

Quadruped: In a similar fashion as prior, we hope to verify the quadruped's ability to maintain positivity of a simple, 2-norm barrier function:

$$h(x, \theta_1, \theta_2, \theta_3, \theta_4) = \min_{i=1,2,3,4} \|x - \theta_i\| - 0.35.$$

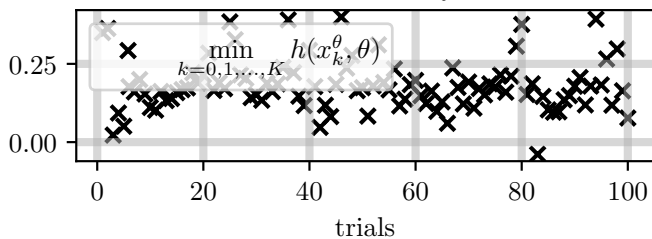Here, we assume that our "state" x is a projection of the true

Fig. 5. Minimum barrier value for each of the 100 trials taken for the verification process of the robotarium hardware as stated in subsection IV-B. Notice that in one trial the system fails to keep positive the candidate barrier h indicating a counterexample.

quadruped state onto the $x - y$ plane and constrained to lie within the state space $\mathcal{X} = [-1, 2]^2$. Also, θ_i is the location in the $x - y$ plane of one of the 4 stationary obstacles. As such, our total parameter vector $\theta \in \Theta = [-1, 2]^8$. Then, our verification problem is very similar to that which we had for the robotarium - see if the quadruped coupled with the controller in [31] can keep positive this candidate barrier function h for at least $K = 150$ time-steps with $\Delta t = 0.1$.

To make a probabilistic verification statement in this vein, we sampled $N_0 = 50$ initial conditions and parameter pairs (x_0, θ) from the uniform distribution over the 0-superlevel set for the candidate barrier function h U$[\mathcal{C}]$. We recorded the resulting trajectories for $K = 150$ time-steps and recorded all transitions. To be more specific about this process, we recorded state data at 1000 Hz and recorded as the state x_k the sampled state whose timestamp was nearest to the desired time $k\Delta t$. This yielded $N = 7500$ transition samples and a calculated safety decay constant $\gamma_N^* = 0.3931$. Furthermore, we expect the violation probability for our solution $V(\gamma_N^*)$ to be upper bounded by $\epsilon = 0.0019$ with minimum probability $1 - 10^{-6}$. This entire procedure required 12.5 minutes of system data. This is why we claim that we can verify hardware systems to high minimum probability with limited data and are confident in the validity of our approach based on the repeatability and validity analyses carried out earlier.

V. Conclusion

In this paper, we detail a randomized verification method for safety-critical systems with limited data via a scenario approach based on barrier functions. We showed that by uniformly sampling initial conditions and parameters and recording the resulting state trajectory, we can determine the constraints for a randomized linear program designed to identify the minimum safety decay constant γ required by the system in its attempt to maintain the positivity of a candidate barrier function h. Given a sufficient number of trajectory samples, the positivity or lack thereof of this constant γ provides us either a probabilistic verification statement or counterexample, respectively. Finally, we showed that this procedure works across multiple systems, both simulated and real ones, and verified our probabilistic verification statements by taking copious samples of the same systems and showing our results holds. As future work, we hope to extend our analysis to the case where the system dynamics are

corrupted by additive noise and identify a similar approach for continuous barrier functions as well.

References

[1] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 708–717.

[2] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3717–3724.

[3] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhwani, "Learning stability certificates from data," *arXiv preprint arXiv:2008.05952*, 2020.

[4] S. M. Khansari-Zadeh and A. Billard, "Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.

[5] H. Ravanbakhsh and S. Sankaranarayanan, "Learning control lyapunov functions from counterexamples and demonstrations," *Autonomous Robots*, vol. 43, no. 2, pp. 275–307, 2019.

[6] ——, "Robust controller synthesis of switched systems using counterexample guided framework," in *2016 international conference on embedded software (EMSOFT)*. IEEE, 2016, pp. 1–10.

[7] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, 2010, pp. 101–110.

[8] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 81–87.

[9] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.

[10] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 1, pp. 96–101, 2018.

[11] ——, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 3, pp. 757–762, 2019.

[12] P. Giesl and S. Hafstein, "Review on computational methods for lyapunov functions," *Discrete & Continuous Dynamical Systems-B*, vol. 20, no. 8, p. 2291, 2015.

[13] T. A. Johansen, "Computation of lyapunov functions for smooth nonlinear systems using convex optimization," *Automatica*, vol. 36, no. 11, pp. 1617–1626, 2000.

[14] J. Anderson and A. Papachristodoulou, "Advances in computational lyapunov analysis using sum-of-squares programming," *Discrete & Continuous Dynamical Systems-B*, vol. 20, no. 8, p. 2361, 2015.

[15] R. Bobiti and M. Lazar, "A sampling approach to finding lyapunov functions for nonlinear discrete-time systems," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 561–566.

[16] 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.

[17] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

[18] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2011, pp. 254–257.

[19] A. Donzé, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in *International Conference on Computer Aided Verification*. Springer, 2010, pp. 167–170.

[20] T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, H. Ravanbakhsh, M. Vazquez-Chanlatte, and S. A. Seshia, "Verifai: A toolkit for the design and analysis of artificial intelligence-based systems," *arXiv preprint arXiv:1902.04245*, 2019.

[21] S. Ghosh, F. Berkenkamp, G. Ranade, S. Qadeer, and A. Kapoor, "Verifying controllers against adversarial examples with bayesian optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7306–7313.

[22] A. Corso, R. J. Moss, M. Koren, R. Lee, and M. J. Kochenderfer, "A survey of algorithms for black-box safety validation," *arXiv preprint arXiv:2005.02979*, 2020.

[23] P. Akella, W. Ubellacker, and A. D. Ames, "Test and evaluation of quadrupedal walking gaits through sim2real gap quantification," *arXiv preprint arXiv:2201.01323*, 2022.

[24] 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, 2016.

[25] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." in *Robotics: Science and Systems*, vol. 13. Cambridge, MA, USA, 2017.

[26] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[27] ——, "Wait-and-judge scenario optimization," *Mathematical Programming*, vol. 167, no. 1, pp. 155–189, 2018.

[28] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, 2012.

[29] ——, "Stochastic model predictive control of lpv systems via scenario optimization," *Automatica*, vol. 49, no. 6, pp. 1861–1866, 2013.

[30] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[31] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2021.