Safety Verification of Discrete-Time Systems via Interpolation-Inspired Barrier Certificates

Mohammed Adib Oumer, Vishnu Murali, *Member, IEEE*, Ashutosh Trivedi, and Majid Zamani, *Senior Member, IEEE*

***Abstract*—Barrier certificates provide an effective automated approach to verifying the safety of dynamical systems. A barrier certificate is a real-valued function over states of the system whose zero level set separates the unsafe region from all possible trajectories starting from a given set of initial states. Typically, the system dynamics must be nonincreasing in the value of the barrier certificate with each transition. Thus, the states of the system that are nonpositive with respect to the barrier certificate act as an over-approximation of the reachable states. The search for such certificates is typically automated by first fixing a template of functions and then using optimization and satisfiability modulo theory (SMT) solvers to find them. Unfortunately, it may not be possible to find a single function in this fixed template. To tackle this challenge, we propose the notion of interpolation-inspired barrier certificate. Instead of a single function, an interpolation-inspired barrier certificate consists of a set of functions such that the union of their sublevel sets over-approximate the reachable set of states. We show how one may find interpolation-inspired barrier certificates of a fixed template, even when we fail to find standard barrier certificates of the same template. We present sum-of-squares (SOS) programming as a computational method to find this set of functions and demonstrate effectiveness of this method over a case study.**

***Index Terms*—Hybrid systems, barrier certificates, interpolation, safety.**

I. Introduction

THE NOTION of barrier certificate [1] is a prominent approach used to verify the safety of dynamical systems. A barrier certificate is a real valued function that is nonpositive over the initial states, positive over the unsafe states, and typically nonincreasing as a system evolves. Thus, such a certificate acts as an inductive proof of safety. The search for barrier certificates is effectively automated through the use of optimization [1], [2] or SMT-based [3] approaches. These approaches typically rely on first fixing a template and then make use of the above approaches to search for an appropriate function in this template. Unfortunately one is forced to consider a different template when one is unable to find such a certificate. Inspired by the success of (logical) interpolation [4] in software and hardware system verification, instead of changing the template, we consider a notion of interpolation-inspired barrier certificates. We show how one may find such certificates in a similar fashion to standard barrier certificates, and allow for larger classes of functions to act as proofs of safety.

The problem of safety verification problem aims to provide a rigorous guarantee that all of the system trajectories starting from some initial states avoid visiting unsafe states. A prominent method for safety verification of digital hardware and software systems, is based on inductive invariants [5]. An inductive invariant is any property that is *i*) true over the initial set of states, and *ii*) true over the next state if it is true in the current state. Thus, any inductive invariant is true over all the reachable states of the system via an inductive argument. Ensuring that such a property is not true over the unsafe states thus provides an effective automated approach to safety. Unfortunately, one may not find a single property of interest to be *inductive*. Thus, a common approach in the verification of these systems is to *incrementally* strengthen the property as a conjunction of other properties. One method to adopt such an incremental strengthening approach is that of interpolation-based model checking [4].

In the context of dynamical systems, barrier certificates proposed by [1] act as functional analogs of inductive invariants. Typical approaches to find barrier certificates assume the presence of a single inductive invariant that holds with every single transition step and the barrier certificate value to be nonincreasing with each transition step. Thus, in cases where those conditions are not met, one cannot prove or disprove the system's safety. This letter utilizes ideas inspired by interpolation to find multiple functions that together act as a guarantee of safety via an incremental approach for discrete-time dynamical systems.

Related Work: The use of inductive invariants and incremental inductive proofs for safety for finite state-transition systems has been quite influential [6], [7], [8]. Interpolation [4] and its extension via IC3 [9] are two major algorithms used in incremental proofs. Such proofs first start with a property to prove, and then try to incrementally constrain it, till

Received 16 September 2024; revised 22 November 2024; accepted 16 December 2024. Date of publication 23 December 2024; date of current version 6 January 2025. This work was supported by NSF under Grant CNS-2111688 and Grant CNS-2145184. Recommended by Senior Editor C. Prieur. *(Corresponding author: Mohammed Adib Oumer.)*

The authors are with the Department of Computer Science, University of Colorado, Boulder, CO 80309 USA (e-mail: mohammed.oumer@colorado.edu; vishnu.murali@colorado.edu; ashutosh.trivedi@colorado.edu; majid.zamani@colorado.edu).

Digital Object Identifier 10.1109/LCSYS.2024.3521356

an inductive proof is obtained. In the context of bounded model checking, interpolation unrolls the transition function some k times and finds intermediate interpolants until an inductive invariant formula is found. IC3 uses frames and counterexamples to build incremental formulae one step at a time until an inductive invariant is found.

The authors of [10] consider multiple unsafe regions and design multiple barrier certificates corresponding to each unsafe region using the standard formulation. Thus their work presumes that a standard barrier certificate can always be found for a single unsafe region. However, our approach is a less restrictive alternative to the standard barrier certificate formulation. Our approach differs from [11] as follows. They introduce vector barrier certificates, which are a set of functions, where the condition for the initial states is imposed over all the functions and the condition for the unsafe states is imposed over at least one of the functions. Our proposed method imposes the condition for the initial states only over the first function and the condition for the unsafe states over all the functions. Moreover, their formulation enforces the nonincreasing condition by relating a single function to multiple other functions in the vector via a Metzler matrix, whereas our approach enforces the same condition only using one of the functions. We also describe how the functions we compute characterize the reachable set of a dynamic system (See Section III-B for more details). Thus, their formulation is not directly comparable to us except through the fact that they also use multiple functions.

An alternative approach to finding more permissive conditions for barrier certificates has been addressed via the use of k-induction in [12]. They considered two notions of k-inductive barrier certificates that both introduced k and few other hyperparameters as part of their formulations. Setting the values of these hyperparameters is not quite straightforward. In this letter, we have only one hyperparameter k that we increment from 0. More fundamentally, they search for a single function using more relaxed conditions. First, they require that the value of the barrier certificate is nonpositive for k-steps starting from the initial set of states. Second, they require a state to have a nonpositive barrier certificate value after k-steps if the previous k states did so. We calculate multiple functions with the standard nonincrease requirement in the value of the barrier certificate. As such, it could be possible to come across systems where only one of these approaches works. A trivial solution will be to combine the ideas to form a more general notion of barrier certificates.

Organization: Section II discusses preliminary concepts crucial for this letter. The main inspiration and theoretical results are explained in Section III along with an implementation method in Section IV. We discuss a case study showing the efficacy of our results in Section V.

II. Preliminaries

A. Notation

The sets of nonnegative and positive integers are denoted by $\mathbb{N} := \{0, 1, 2, \ldots\}$ and $\mathbb{N}_+ := \{1, 2, 3, \ldots\}$, respectively. The sets of real numbers and positive real numbers are denoted by symbols $\mathbb{R}$ and $\mathbb{R}_+$, respectively. The n-dimensional Euclidean space is denoted by $\mathbb{R}^n$.

Given a collection of sets X_i, $i = \{0, 1, \ldots, N\}$, we use $\bigcup_{i=0}^{N} X_i$ to denote the union of the sets X_i. Given two sets X and Y, we use $X \backslash Y$ for set difference to denote the set of elements in X excluding all the elements of Y (i.e., $X \backslash Y = \{x : x \in X \text{ and } x \notin Y\}$).

For two sets X and Y, a set-valued function map $\mathcal{F}:X \rightrightarrows Y$ is a mapping of elements from X to subsets of Y. For a function map $\mathcal{F} : X \rightrightarrows X$ and $k \in \mathbb{N}$, $\mathcal{F}^k$ denotes the self-composition of $\mathcal{F}$ k-times (i.e., given $x \in X$, $\mathcal{F}^k(x) = \bigcup_{x' \in \mathcal{F}(x)} \{x'' \mid x'' \in \mathcal{F}^{k-1}(x')\}$ with $\mathcal{F}^0(x) = \mathsf{identity}$).

We use $\forall$ and $\exists$ to denote the universal and existential quantifiers, respectively. We use logical operators $\wedge$, $\vee$, $\neg$ and $\implies$ for conjunction (logical AND), disjunction (logical OR), negation (logical NOT) and implication, respectively.

B. Discrete-Time Dynamical Systems

In this letter, we model the systems as discrete-time dynamical systems defined next.

Definition 1: A discrete-time dynamical system is given by the tuple:

$$\mathcal{S} = (X, X_0, \mathcal{F}), \tag{1}$$

over the state set X, set of initial states $X_0 \subseteq X$, and $\mathcal{F}:X \rightrightarrows X$ is a set-valued transition map that describes the evolution of the states of the system. That is, for $x(t)$, the state of the system at time step $t \in \mathbb{N}$, the state of the system in the next time step is given by:

$$x(t+1) \in \mathcal{F}(x(t)),\ \forall x(t) \in X.$$

We use $\mathbf{x}_{x_0} = (x(0), x(1), x(2), \ldots)$ to denote the state sequence of the system $\mathcal{S}$ starting from the initial state $x(0) = x_0 \in X_0$.

Now we define reachable states of discrete-time systems.

Definition 2 (Reachability): We say a state $x(t_1)$ of a system $\mathcal{S}$ is reachable from the state $x(t_0)$ if there exists a state sequence $\mathbf{x}_{x(t_0)}$ which contains $x(t_1)$. That is, $x(t_1) \in \mathcal{F}^i(x(t_0))$, for some $i \in \mathbb{N}$.

Now we define safety for discrete-time systems.

Definition 3 (Safety): We say that a system $\mathcal{S}$ given in Definition 1 is safe with respect to a set of initial states $X_0 \subseteq X$ and a set of unsafe states $X_u \subseteq X$ if no state sequence starting from X_0 reaches X_u. That is for any state sequence $\mathbf{x}_{x_0} = (x(0), x(1), \ldots)$ where $x(0) \in X_0$, we have $x(i) \notin X_u$ for all $i \in \mathbb{N}$.

C. Inductive Invariants

We now describe inductive invariants as discussed in [8]. Consider a finite-state system, where the state set is a set of logical values while the initial set of states and transition map are described by propositional logical formula. That is, $X \subseteq \{0, 1\}^n$, $X_0 = \{x \mid I(x) = 1\}$, $\mathcal{F}(x) = \{x' \mid T(x, x') = 1\}$ where the formula $I(x)$ is the initial condition over the system's states x, and $T(x, x')$ is the transition relation from the current state x to the next state x'.

In order to determine whether such a system is safe, we look at the safety property expressed by a logical formula $P(x)$ described over the state variable $x \in X$. We say that such a system satisfies a safety property if, for every reachable

state $x \in X$ from the initial set, we have $P(x)$ to be true (i.e., $P(x) = 1$). A prominent and effective approach to prove safety is through the use of inductive invariants. We say that a formula K is an inductive invariant, if:

- $\forall x \in X$, we have $I(x) \implies K(x)$.
- $\forall x, x' \in X$, we have $K(x) \wedge T(x, x') \implies K(x')$.

Note that any reachable state x satisfies an inductive invariant formula K. Thus, showing that a safety property P is an inductive invariant acts as a proof of safety. When we fail to prove P to be an inductive invariant (that is $I(x) \not\Longrightarrow P(x)$ and/or $P(x) \wedge T(x, x') \not\Longrightarrow P(x')$), we try to *strengthen* P. We say that P^* is an inductive strengthening of a safety property P if there exists a formula F such that $P^* = F \wedge P$ is inductive. In [13], two strengthening strategies are discussed: *i*) using a stronger property, or *ii*) conducting an incremental proof using previously computed formulae. Interpolation [4] is one of these incremental techniques used in the inductive strengthening process and will be discussed broadly in the next section.

D. Barrier Certificates

For safety verification of a system $\mathcal{S}$ as in Definition 1, we now discuss the notion of barrier certificates [1] that provide sufficient conditions for safety.

Definition 4 (Barrier Certificate): A function $\mathcal{B} : X \to \mathbb{R}$ is a barrier certificate for a system $\mathcal{S}$ if:

$$\mathcal{B}(x) \leq 0 \quad \forall x \in X_0, \tag{2}$$

$$\mathcal{B}(x) > 0 \quad \forall x \in X_u, \text{ and} \tag{3}$$

$$\mathcal{B}(x) \leq 0 \implies \mathcal{B}(x') \leq 0 \quad \forall x \in X \backslash X_u, \ \forall x' \in \mathcal{F}(x). \tag{4}$$

The next result borrowed from [1] shows the usefulness of barrier certificates in verifying safety.

Theorem 1: Consider a dynamical system $\mathcal{S}$. If there exists a function $\mathcal{B} : X \to \mathbb{R}$ for $\mathcal{S}$ such that it is a barrier certificate as in Definition 4, then the system is safe.

Observe that the level set $\mathcal{B}(x) = 0$ acts as the barrier that prevents the system from reaching the unsafe set of states. That is, starting from any state $x \in X$, where $\mathcal{B}(x) \leq 0$, the system cannot reach a state $\bar{x} \in X$, where $\mathcal{B}(\bar{x}) > 0$.

Note that the above condition is an inductive invariant for system $\mathcal{S}$ where $P(x) = (B(x) \leq 0)$. For any $x_0 \in X_0$, $P(x_0) = 1$ from condition (2). Now, consider states $x, x' \in X$ such that $x' \in \mathcal{F}(x)$, and suppose that $P(x) = 1$. Following condition (4), $P(x') = 1$. Thus, the two conditions jointly guarantee that the safety property is satisfied for any reachable state x in the state set starting from the initial set.

The search for barrier certificates has been primarily performed by fixing a template. By fixing the template, one considers the barrier certificate to be a linear combination of some fixed basis functions. For example, if it is a polynomial of a fixed degree, then we consider the basis functions to be monomials and aim to find the coefficients corresponding to each basis function. Search techniques, such as Sum-of-Squares (SOS) programming [2] or Satisfiability Modulo Theory (SMT) solvers [3], can be employed to search for these coefficients satisfying conditions (2)-(4). Unfortunately, if no barrier certificate is found, a common approach is to change the template (e.g., increase the degree of the polynomial). Such changes typically make verifying (or searching) for barrier certificates computationally demanding and may lead to inconclusive results as shown below.

Example 1: Consider a one-dimensional system

$$\mathcal{S} : x(t+1) = 0.5x(t). \tag{5}$$

The state set, initial, and unsafe sets are given by $X = [0, 3]$, $X_0 = [2, 2.3]$, and $X_u = [1.6, 1.9]$, respectively. We consider a linear barrier certificate of the parametric form $\mathcal{B}(x) = mx + b$ and attempt to compute coefficients $m, b \in \mathbb{R}$. We require $\mathcal{B}(2) = 2m + b \leq 0$ from condition (2), and furthermore, we need to ensure that $\mathcal{B}(1.9) = 1.9m + b > 0$ from condition (3). From condition (4), $\mathcal{B}(2) \leq 0 \implies \mathcal{B}(1) = m + b \leq 0$. However, $b \leq -2m$, $b > -1.9m$, $b \leq -m$ cannot all be satisfied simultaneously so there exists no standard linear barrier certificate.

In the next section, we discuss the idea of (logical) interpolation [4], [8], to define a notion of interpolation-inspired barrier certificates. The conditions for these are more permissive as illustrated through Example 1 later.

III. Interpolation and Safety

A. Interpolation

The concept of logical interpolation plays a key role in developing algorithms for bounded model checking (BMC). BMC is a technique that aims to find bugs in hardware and software systems by falsifying safety properties [14]. Consider finite systems as described in Section II-C. In BMC, we unroll the transition relation some $k \in \mathbb{N}$ times until we reach an unsafe state and construct a formula representing all possible execution paths from an initial state to the unsafe state (assuming that all states before the k^{th} step are safe). Let's say x_i is the state after the i^{th} transition. Then the sequence of states for this unrolling is given by:

$$I(x_0) \wedge T(x_0, x_1) \wedge \ldots \wedge T(x_{k-1}, x_k) \wedge \neg P(x_k), \tag{6}$$

where logical formula $P(x)$ describes a safety property.

For $k = 0$, the formula reduces down to $I(x_0) \wedge \neg P(x_0)$. If formula (6) is satisfied, then the system is unsafe. Otherwise, one may use interpolation to try to prove safety by finding an intermediate logical formula or a series of formulae called interpolants via Craig's interpolation theorem [4] as follows.

Theorem 2 (Craig's Interpolation Theorem): Given a pair of clauses (a disjunction of boolean variables or their negation) E and G such that $E \wedge G$ is unsatisfiable, then there exists an intermediate interpolant clause F such that:

- $E \implies F$,
- $F \wedge G$ is unsatisfiable, and
- F refers to the common variables of E and G.

The proof of Theorem 2 can be found in [15].

Based on this theorem, when formula (6) is unsatisfiable, there exists intermediate formulae F_i such that formula (6) can be broken down as follows:

$$\underbrace{I(x_0)}_{E_0(x_0)} \wedge \underbrace{T(x_0, x_1) \wedge \ldots \wedge T(x_{k-1}, x_k) \wedge \neg P(x_k)}_{G_0(x_0, x_1 \ldots, x_k)}$$

is unsatisfiable.

We can then iteratively separate G_0 as follows:

$$\begin{cases} I(x_0) \implies F_0(x_0) \\ F_0(x_0) \wedge T(x_0, x_1) \implies F_1(x_1) \\ \vdots \\ F_{k-1}(x_{k-1}) \wedge T(x_{k-1}, x_k) \implies F_k(x_k), \text{ and} \\ F_k(x_k) \wedge \neg P(x_k) \text{ is unsatisfiable.} \end{cases} \tag{7}$$

Informally, condition (7) can be thought of as a way of representing the over-approximation of reachable sets. That is, the set of $x_i \in X$ where the formula $F_i(x_i)$ is true is an over-approximation of states reachable in i steps and states satisfying $F_i(x_i)$ will not violate the safety property after $(k-i)$ transitions. To get the interpolants, as shown in [8], [9], we start with $k = 0$ and iteratively (incrementally) compute a sequence of interpolants $F_0(x_0) = I(x_0), F_1(x_1), \ldots, F_k(x_k)$ by setting $E(x_i, x_{i+1}) = F(x_i) \wedge T(x_i, x_{i+1})$ and $G(x_{i+1}, \ldots, x_k) = T(x_{i+1}, x_{i+2}) \ldots T(x_{k-1}, x_k) \wedge \neg P(x_k)$ according to Theorem 2. This iterative process is stopped when the union of the initial formula and previous interpolants grows to a fixed point (when all reachable states are contained in that union).

Unfortunately, such a fixed point is not guaranteed for infinite state systems. Next, we consider a notion of barrier certificates inspired by the above BMC paradigm that combines interpolation with that of inductive invariants. Such a combination takes inspiration from [4], [8].

B. Interpolation-Inspired Barrier Certificate (IBC)

Here we introduce a notion of interpolation-inspired barrier certificates (IBC) and demonstrate their efficacy.

Definition 5: Consider a discrete-time dynamical system $\mathcal{S}$ as in Definition 1. A set of functions $\mathcal{B}_i : X \to \mathbb{R}$, for all $0 \leq i \leq k$, is an IBC for $\mathcal{S}$ if the following hold:

$$\mathcal{B}_0(x) \leq 0 \qquad \forall x \in X_0, \tag{8}$$

$$\mathcal{B}_i(x) > 0 \qquad \forall x \in X_u,\ 0 \leq i \leq k, \tag{9}$$

$$\mathcal{B}_i(x) \leq 0 \implies \mathcal{B}_{i+1}(x') \leq 0 \quad \forall x \in X \backslash X_u, \forall x' \in \mathcal{F}(x),\ 0 \leq i < k, \tag{10}$$

$$\mathcal{B}_k(x) \leq 0 \implies \mathcal{B}_k(x') \leq 0 \quad \forall x \in X \backslash X_u, \forall x' \in \mathcal{F}(x). \tag{11}$$

The next theorem shows the usefulness of IBC.

Theorem 3: Consider a discrete-time dynamical system $\mathcal{S}$ as in Definition 1. If there exists an IBC $\mathcal{B}_i : X \to \mathbb{R}$, $0 \leq i \leq k$, for $\mathcal{S}$ as in Definition 5, then the system is safe.

Proof: We prove the above theorem by contradiction. Assume system $\mathcal{S}$ satisfies conditions (8)-(11) but is not safe. Then there exists a state sequence $\mathbf{x}_{x_0} = (x(0), x(1), x(2), \ldots)$ such that $x(0) \in X_0$, and $x(j) \in X_u$ for some $j \in \mathbb{N}$. From condition (9), we have $\mathcal{B}_i(x(j)) > 0$ for all $0 \leq i \leq k$. We consider two cases for the value of j.

- If $j < k$. If $j = 0$, then we must have $\mathcal{B}_0(x(j)) \leq 0$ following condition (8). This is a contradiction. If $0 < j < k$, then following condition (8) and via induction through condition (10), we must have the value of $\mathcal{B}_j(x(j)) \leq 0$. This is again a contradiction.
- If $j \geq k$, then similar to the earlier case, we have $\mathcal{B}_k(x(k)) \leq 0$ following conditions (8) and (10). Via condition (11), and induction, we must have $\mathcal{B}_k(x(j)) \leq 0$, which is again a contradiction. ■

Corollary 1: Consider a system $\mathcal{S}$ with a corresponding IBC as in Definition 5. Any reachable state $x_j \in X$, $j \in \mathbb{N}$, from initial state $x_0 \in X_0$ satisfies:

$$x_j \in \{x \mid \mathcal{B}_i(x) \leq 0 \quad \forall x \in X\},$$

where $i = j$ when $0 \leq j \leq k$ and $i = k$ when $j > k$.

Following conditions (8) and (10), by inductive reasoning, a state that evaluates to a nonpositive value of the barrier certificate at the i^{th} step evaluates to a nonpositive value of the barrier certificate at the $(i+1)^{th}$ step. It follows that $\mathcal{B}_i(x)$ represents a barrier certificate that over-approximates all states x that are reachable in i-steps from the initial states in X_0 for all $0 \leq i \leq k$. From condition (11), $\mathcal{B}_k(x)$ over-approximates all states that are reachable in more than k-steps from the set X_0. This interpretation of $\mathcal{B}_i(x)$, is formally stated in Corollary 1.

Corollary 2: Consider a system $\mathcal{S}$ with a corresponding IBC as in Definition 5. All reachable states $x_j \in X$ of $\mathcal{S}$, $\forall j \in \mathbb{N}$, satisfy:

$$x_j \in \bigcup_{i=0}^{k} \{x \mid \mathcal{B}_i(x) \leq 0 \quad \forall x \in X\}.$$

Corollary 2 formally states that the reachable states of the system are contained in the union of all the zero-sublevel sets of the IBC.

Note that by setting $k = 0$ in Definition 5, condition (10) is inapplicable and conditions (8), (9) and (11) reduce to the standard barrier certificate conditions as in Definition 4. This is relevant for the implementation as we first start with $k = 0$ to find a standard barrier certificate. We then increment k by one only if we fail, and check for satisfiability of conditions (8)-(11). We repeat the above until we find an IBC or we reach a maximum number k_{max}. Any IBC found for $k > 0$ indicates that a standard barrier certificate with the given template could not be found. Also observe that once an IBC is found for a given $k \in \mathbb{N}$, we guarantee that an IBC can be found for all $j > k$. In this sense, k is the minimum integer that forms an IBC for a given fixed template.

We now show that one can find an IBC even if a standard barrier certificate cannot be found, based on Example 1.

Example 1(Continued): We consider a set of linear barrier certificate functions of the parametric form $\mathcal{B}_i(x) = m_i x + b_i$ and attempt to compute coefficients $m_i, b_i \in \mathbb{R}$, with an upper bound $i \leq k_{max} = 3$ such that the collection of $\mathcal{B}_i(x)$ is an IBC as defined in Definition 5. We utilize SOS optimization [2] to solve conditions (8)-(11) and find the coefficients. See Section IV for the formulation.

An IBC was found for $k = 1$ with $\mathcal{B}_0(x) = -0.143x + 0.281$ and $\mathcal{B}_1(x) = 2.539x - 4.002$. Figure 1 shows the IBC computed along with the relevant initial and unsafe sets. Function $\mathcal{B}_0(x)$ is always nonpositive for the initial states and positive for the unsafe states as expected. Observe that the set of states that are reachable in one step from the initial state are within the zero level set of $\mathcal{B}_1(x)$ but not $\mathcal{B}_0(x)$.

We also note that IBCs do not automatically replace a higher degree polynomial standard barrier certificate with a set of linear ones in all cases. In fact, one can trivially see that

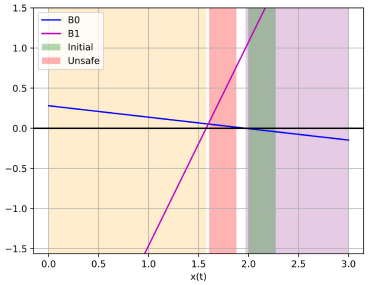

Fig. 1. IBC with $k = 1$. The purple and orange shaded regions represent the sublevel sets of $\mathcal{B}_0(x)$ and $\mathcal{B}_1(x)$ respectively. They together over-approximate the reachable states.

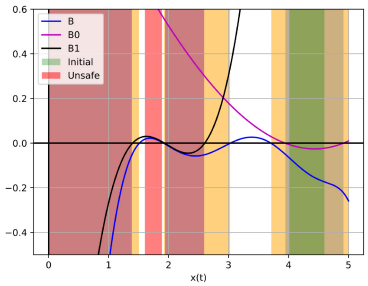

Fig. 2. Standard barrier certificate and IBC for system in Example 1. The (overlapped) orange shaded regions represent the sublevel set of $\mathcal{B}(x)$. The gray (overlapping green and orange) and purple (overlapping orange) shaded regions represent the sublevel sets of $\mathcal{B}_0(x)$ and $\mathcal{B}_1(x)$, respectively. The green shaded region appears darker due to overlap with other colors.

for the system in Example 1, if we modify the unsafe set to be $X_u = [0.8, 0.95] \cup [1.6, 1.9]$, there is no combination of linear functions to satisfy the IBC conditions. This is because no linear function can be positive over the given unsafe states and be nonpositive in the region $[1, 1.15]$, the set of states reachable in one step from the initial states $X_0 = [2, 2.3]$. However, IBCs can allow us to find a set of lower degree polynomials as opposed to larger degree polynomials for barrier certificates. To better illustrate this point, we use the following example to highlight the benefit of IBCs over merely incrementing the degree of a standard barrier certificate. For the same simple system given in Example 1, we keep the same unsafe set $X_u = [1.6, 1.9]$ and modify the state set and initial set to be $X = [0, 5]$ and $X_0 = [4, 4.6]$, respectively. Using SOS, we see that the safety of the system can be verified using a degree seven polynomial as a standard barrier certificate and not any degree less than seven. On the other hand, we only need two cubic polynomials in the form of IBC to verify safety. Figure 2 shows the plot of these barrier certificates. $\mathcal{B}(x)$ shows the standard barrier certificate while $\mathcal{B}_0(x)$ and $\mathcal{B}_1(x)$ represent the IBC functions. The sublevel set of $\mathcal{B}(x)$ as well as the union of the sublevel sets of $\mathcal{B}_0(x)$ and $\mathcal{B}_1(x)$ over-approximate the reachable states of the system. In this example, we also notice that the over-approximation obtained from the IBC functions is tighter than that of the standard barrier certificate. As such, IBCs could potentially remedy the concern of increase in degree by allowing us to use a combination of much lower degree polynomials as proof of safety. A potential concern with IBCs is that it is not obvious as to when to choose to increment k versus the degree of the polynomial template for successful search results.

IV. Computation of IBC Using SOS

In this section, we provide a suitable computational method using sum-of-squares (SOS) optimization [2] for synthesizing IBC based on Definition 5. Here, we consider a single valued transition map $\mathcal{F}$, which is a deterministic transition function denoted by $f : X \to X$.

When function f is polynomial and the initial set X_0 and unsafe set X_u are semi-algebraic [16], conditions (8)-(11) can be cast as a collection of SOS constraints in order to compute a polynomial IBC of a predefined degree.

Assumption 1: The state set X is a subset of $\mathbb{R}^n$, and the transition function $f : X \to X$ is a polynomial function of the state x. Furthermore, sets X, X_0 and X_u are semi-algebraic and can be described as vectors of polynomial inequalities: $X = \{x \in \mathbb{R}^n \mid g(x) \geq 0\}$, $X_0 = \{x \in \mathbb{R}^n \mid g_0(x) \geq 0\}$, and $X_u = \{x \in \mathbb{R}^n \mid g_u(x) \geq 0\}$ where the inequalities are element-wise.

SOS handles optimization problems when the constraints are written as conjunctions of one another. However, an IBC as defined in Definition 5 requires the satisfaction of logical implications (conditions (10) and (11)), which cannot be checked using the SOS approach. Therefore, we reformulate conditions (10) and (11) as more conservative ones that are compatible with SOS optimization and are given by:

$$\tau_i \mathcal{B}_{i+1}(f(x)) - \mathcal{B}_i(x) \leq 0 \quad \forall x \in X \backslash X_u, \ 0 \leq i < k, \tag{12}$$

$$\tau_k \mathcal{B}_k(f(x)) - \mathcal{B}_k(x) \leq 0 \quad \forall x \in X \backslash X_u, \tag{13}$$

where $\tau_i \in \mathbb{R}_+$, $0 \leq i \leq k$.

These conservative conditions require the barrier certificate to be nonincreasing throughout the evolution of the system by some factor τ_i while the implication-based statements require the barrier certificate values to be below the zero level set for all reachable states of the system.

Under Assumption 1, conditions (8), (9), (12) and (13) can be formulated as a set of SOS constraints, as follows.

Lemma 1: Consider a discrete-time dynamical system $\mathcal{S}$. Suppose Assumption 1 holds for $\mathcal{S}$ and there exist constants $k \in \mathbb{N}$, $\epsilon \in \mathbb{R}_+$, polynomials of same degree $\mathcal{B}_i(x)$ and SOS polynomials $\lambda_0(x)$, $\lambda_{u,i}(x)$, $\lambda_i(x)$, $\hat{\lambda}(x)$ of appropriate dimensions such that:

$$-\mathcal{B}_0(x) - \lambda_0^T(x) g_0(x), \tag{14}$$

$$\mathcal{B}_i(x) - \epsilon - \lambda_{u,i}^T(x) g_u(x) \quad \text{for all } 0 \leq i \leq k, \tag{15}$$

$$\mathcal{B}_i(x) - \tau_i \mathcal{B}_{i+1}(f(x)) - \lambda_i^T(x) g(x) \quad \text{for all } 0 \leq i < k, \tag{16}$$

$$\mathcal{B}_k(x) - \tau_k \mathcal{B}_k(f(x)) - \hat{\lambda}^T(x) g(x), \tag{17}$$

are SOS polynomials where x is the state variable over X.

Then the set of functions $\mathcal{B}_i(x)$, $0 \leq i \leq k$, is an IBC following Definition 5. Note that ϵ is introduced in condition (15) to convert the strict inequality in condition (9) to an inclusive inequality.

Observe that our formulation can help address the issue of computational complexity with regards to SOS as follows: the search for traditional barrier certificates via SOS has a complexity that is polynomial in $O(\binom{n+d}{d} \times \binom{n+d}{d})$ [2, Th. 3.3],

where n is the dimension of the system, and $2d$ is the degree of the SOS polynomial. The term $O(\binom{n+d}{d} \times \binom{n+d}{d})$ represents the number of decision variables introduced in converting the SOS problem to an SDP characterization. This suffers in a polynomial fashion with respect to the degree $2d$. By allowing for lower degrees of functions to act as certificates in our formulation, we suffer a constant additional complexity cost ($O(1)$) when the number of interpolating functions is constant. This grows linearly as we increase the value of k (if k is not fixed, then we suffer with a complexity that is $O(k)$, and not polynomially in k). Thus, one may use existing techniques and reduce the computational burden on the search for the certificate.

V. Case Study

For our case study, we consider the discrete-time Lotka-Volterra type prey–predator model with state variables v, p denoting the victim/prey and the predator, respectively. The dynamics is given by the following difference equations:

$$\begin{cases} v(t+1) = v(t) + T(\alpha v(t)(1 - v(t)) - \beta v(t)p(t)), \\ p(t+1) = p(t) - T(\gamma p(t) - \delta v(t)p(t)), \end{cases} \tag{18}$$

where $T = 0.1s$ is the sampling time, $\alpha = 1.1$ is the growth rate of the prey, $\beta = 0.4$ is the death rate of the prey, $\gamma = 0.4$ is the death rate of the predator, and $\delta = 0.1$ is the growth rate of the predator. The state set, initial set, and unsafe set are given by $X = [0, 10] \times [0, 5]$, $X_0 = [6, 7] \times [2, 3]$, and $X_u = [3, 5] \times [0, 3]$, respectively. We first consider a degree three polynomial function in two variables as our parametric template of the barrier certificate $\mathcal{B}(v, p)$ and attempt to compute suitable coefficients such that $\mathcal{B}(v, p)$ is a standard barrier certificate as in Definition 4. We used TSSOS [17] in Julia to reformulate conditions (2)-(4) as SOS optimization problem as described in the previous section with $\epsilon = 10^{-5}$. However, we found that no suitable coefficients exist. Therefore, one cannot verify the safety of this system using a standard barrier certificate.

We then reformulated conditions (8)-(11) as an SOS optimization problem via Lemma 1. By taking $k_{max} = 3$, $\epsilon = 10^{-5}$ and the same parametric form as above for $\mathcal{B}_i(v, p)$, we obtain the following IBC with $k = 1$:

$$\begin{aligned} \mathcal{B}_0(v,p) = {} & 0.13305v^3 + 0.16826v^2p - 0.11002vp^2 \\ & + 0.30322p^3 - 1.52807v^2 - 3.55885vp \\ & + 2.1532p^2 + 5.77075v + 3.22429p + 5.34744 \\ \mathcal{B}_1(v,p) = {} & 0.24199v^3 + 0.22602v^2p + 0.14704vp^2 \\ & + 0.1285p^3 + 2.99671v^2 - 0.13358vp \\ & - 0.5105p^2 - 6.71922v + 1.58361p - 12.69347 \end{aligned}$$

Figure 3 shows the union of the zero-sublevel sets of the IBC. These do not include the set of unsafe states.

VI. Conclusion

We proposed a notion of interpolation-inspired barrier certificate (IBC) for discrete-time systems, which relax the conditions of a standard barrier certificate by incrementally finding functions that together guarantee safety. We presented SOS optimization as a technique of computing IBC under some assumptions. Using an example and a case study, we demonstrated that given a barrier certificate template, one may find IBC even when standard barrier certificates do not exist for a system. Given that SOS-based approaches are not computationally tractable for systems with larger dimensions, we hope that the potential to find multiple low degree polynomials via IBC will alleviate these concerns. As future work, we plan to extend this letter for stochastic systems and explore how to extend IC3 for barrier certificates. We also plan to explore the use of IBCs in controller synthesis.

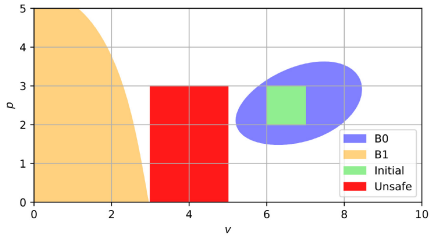

Fig. 3. IBC sublevel sets for Lotka-Volterra type model. The axes show the state variables v and p. The blue and purple shaded regions show the sublevel sets of $\mathcal{B}_0(v,p)$ and $\mathcal{B}_1(v,p)$ respectively.

References

[1] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Proc. Int. Workshop Hybrid Syst. Comput. Control*, 2004, pp. 477–492.
[2] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Math. Program.*, vol. 96, pp. 293–320, May 2003.
[3] L. De Moura and N. Bjørner, "Satisfiability modulo theories: Introduction and applications," *Commun. ACM*, vol. 54, no. 9, pp. 69–77, 2011.
[4] K. L. McMillan, "Interpolation and SAT-based model checking," in *Proc. 15th Int. Conf. Comput. Aided Verification*, 2003, pp. 1–13.
[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2022.
[6] G. Cabodi, S. Nocco, and S. Quer, "Strengthening model checking techniques with inductive invariants," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 1, pp. 154–158, Jan. 2009.
[7] L. Zhang, M. R. Prasad, and M. S. Hsiao, "Incremental deductive & inductive reasoning for SAT-based bounded model checking," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, 2004, pp. 502–509.
[8] A. R. Bradley, "SAT-based model checking without unrolling," in *Proc. Int. Workshop Verification, Model Checking, Abstract Interpretation*, 2011, pp. 70–87.
[9] A. R. Bradley, "Understanding IC3," in *Proc. Int. Conf. Theory Appl. Satisfiability Testing*, 2012, pp. 1–14.
[10] A. J. Barry, A. Majumdar, and R. Tedrake, "Safety verification of reactive controllers for UAV flight in cluttered environments using barrier certificates," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 484–490.
[11] A. Sogokon, K. Ghorbal, Y. K. Tan, and A. Platzer, "Vector barrier certificates and comparison systems," in *Proc. Int. Symp. Formal Methods*, 2018, pp. 418–437.
[12] M. Anand, V. Murali, A. Trivedi, and M. Zamani, "Safety verification of dynamical systems via k-inductive barrier certificates," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 1314–1320.
[13] Z. Manna and A. Pnueli, *Temporal Verification of Reactive Systems: Safety*. New York, NY, USA: Springer, 2012.
[14] A. Biere, "Bounded model checking," in *Handbook of Satisfiability*. Amsterdam, The Netherlands: IOS Press, 2021, pp. 739–764.
[15] A. S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[16] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*, vol. 36. Heidelberg, Germany: Springer, 2013.
[17] J. Wang, V. Magron, and J.-B. Lasserre, "TSSOS: A moment-SOS hierarchy that exploits term sparsity," *SIAM J. Optim.*, vol. 31, no. 1, pp. 30–58, 2021.