Generalizing SDP-Based Barrier Certificate Synthesis to Unbounded Domains by Dropping Archimedean Condition

Hao Wu
wuhao@ios.ac.cn
State Key Lab. of Computer Science
Institute of Software, Chinese Academy of Sciences&
University of CAS
Beijing, China

Shenghua Feng
fengsh@zgclab.edu.cn
Zhongguancun Laboratory
Beijing, China

Ting Gan
ganting@whu.edu.cn
School of Computer Science, Wuhan University
Wuhan, China

Jie Wang
wangjie212@amss.ac.cn
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing, China

Bican Xia
xbc@math.pku.edu.cn
School of Mathematical Sciences, Peking University
Beijing, China

Naijun Zhan
znj@ios.ac.cn
State Key Lab. of Computer Science
Institute of Software, Chinese Academy of Sciences&
University of CAS
Beijing, China

ABSTRACT

Barrier certificates, which serve as differential invariants that witness system safety, play a crucial role in the verification of cyber-physical systems (CPS). Prevailing computational methods for synthesizing barrier certificates are based on semidefinite programming (SDP) by exploiting *Putinar Positivstellensatz*. Consequently, these approaches are limited by *Archimedean condition*, which requires all variables to be bounded, i.e., systems are defined over bounded domains. For the unbounded case, unfortunately, these methods become conservative and even fail to identify potential barrier certificates.

In this paper, we address this limitation by presenting a new computational method. The main technique we use is the *homogenization approach* [40], which was proposed in optimization community recently, to transform an unbounded optimization problem to a bounded one. Our method can be applied to various definitions of barrier certificates, thus expanding the scope of barrier certificate synthesis in the general sense. Experimental results demonstrate that our approach is more effective while maintaining a comparable level of efficiency.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**; • **Theory of computation → Logic and verification**; • **Mathematics of computing → Semidefinite programming**.

KEYWORDS

Safety, differential invariants, barrier certificates, semidefinite programming, homogenization

1 INTRODUCTION

Background. With recent advancements in optimization theory and computational techniques, Cyber-Physical Systems (CPS), which involve the seamless integration of physical components and software systems, have proliferated across various application domains. A significant subset of CPS, known as safety-critical systems, presents a heightened level of concern. Failures or malfunctions in such systems can lead to severe safety risks for individuals and the environment. Examples of safety-critical CPS include aircraft, automobiles, integrated medical devices, nuclear power plants, and biological systems. As a result, ensuring the safety of these systems has become a primary focus of extensive academic research.

Hybrid systems are mathematical models that involves both continuous dynamics and discrete transitions, and hence are widely used for modelling CPS. One of the key challenges in CPS verification is the safety problem (or dually, the reachability problem) of hybrid systems. This problem aims to demonstrate that a hybrid system, starting from its initial states, never enters an unsafe region. In general, the safety problem of hybrid systems is undecidable [37]. However, for certain sub-classes of hybrid systems, the problem becomes decidable [3, 6, 27, 45]. The most challenging aspect of the safety problem lies in reasoning about the continuous dynamics of hybrid systems, which are typically described by ordinary differential equations (ODEs). Existing approaches can be broadly categorized into two groups, *reachability analysis* and *deductive verification*.

Reachability analysis aims to compute or approximate the set of reachable states. The choice of different set representations leads to various approaches in this field. For example, one can utilize geometric objects (such as hyper-rectangles [50], polytopes [11], ellipsoids [44], zonotopes [31]) or symbolic representations (such as support functions [34], Taylor models [14, 16]) to depict sets of system states and perform set propagation to construct approximations of the reachable set. For a comprehensive survey on this topic, we recommend referring to [2]. Alternatively, simulation-based method represents system states by nearby sampled trajectories and attempt to cover the reachable set by a finite number of neighborhoods of trajectories [10, 24–26, 32]. Another class of methods represents system states by constructing a finite state abstraction of the system, thereby enabling the incorporation of model checking techniques [4, 12, 68].

Deductive verification, derived from Hoare-style program verification [38], offers a method to verify safety without directly computing the reachable set. At the core of deductive verification lies the synthesis of *differential invariants* [47, 55], which extend the concept of inductive invariants to the continuous-time domain. Specifically, a differential invariant is a set of states from which any trajectories starting from it can never escape. With a priori specified template, the invariant generation problem boils down to solving the constraints encoding the invariant condition. When all involved constraints are polynomial, the problem is decidable but has a doubly exponential time complexity [47], according to Tarski's theorem [69] and the complexity for the quantifier elimination procedure [20]. Consequently, considerable efforts have been dedicated to identifying differential invariants that allow for efficient synthesis.

In their seminar work [56], Prajna and Jadbabaie introduced the concept of barrier certificates as witnesses to safety. Namely, a barrier certificate is a real-valued function whose zero sub-level set serves as a differential invariant, separating the set of initial states and the unsafe region. It is important to note that, for the purpose of efficient synthesis, the barrier certificate condition strengthens the general condition of differential invariants. Since then, various definitions of barrier certificates have been proposed, aiming to relax the original barrier certificate conditions while still allowing for efficient synthesis. Examples of such definitions include exponential-type barrier certificates [43], Darboux-type barrier certificates [74], general convex barrier certificates [18] and vector barrier certificates [66], and invariant barrier certificate [70]. Moreover, similar notions of barrier certificates have been developed for verifying systems that involve control inputs [5, 73], disturbances [71], and stochastic dynamics [39, 41, 57]. These extensions broaden the applicability of barrier certificates in various domains. Recently, there are also works aim at generalizing the notion of k-inductiveness for safety verification, leading to the definitions of t-barrier certificates [13] and k-inductive barrier certificates [7, 8].

Sum-of-squares programming [46] is a well-established computational technique for synthesizing barrier certificates and has been employed in most of the works mentioned above. Typically, the barrier certificate conditions are first encoded into constraints involving sum-of-squares polynomials. These constraints are then translated into SDP and solved by numerical solvers. In the encoding step, one can choose to rely on either a sufficient condition or a necessary condition. In scenarios where the domains are bounded, the differences between these two conditions are often overlooked, as their formulations are quite similar. However, when dealing with systems defined over unbounded domains, the sufficient condition tends to be conservative while the necessary condition can not be utilized due to Archimedean condition in Putinar's Positivstellensatz. In such cases, the sufficient condition becomes the sole viable option, potentially leading to conservative results.

Besides sum-of-squares programming, much efforts have been devoted to incorporate other numerical methods for solving the obtained constraints, for instance, interval arithmetic [22, 29, 30], linear programming [64], and data-driven approaches [1, 54, 62, 75, 76].

Contributions. This paper focuses on the computational aspect of barrier certificates. Our main contributions are threefold:

(1) We present the problem of synthesizing barrier certificates as a special class of polynomial feasible problems. We then highlight the differences between utilizing the necessary condition and the sufficient condition. It is worth noting that these distinctions are mostly overlooked in existing works with only a mention in [70]. (See Section 3)
(2) We derive a necessary condition for polynomial feasible problems with unbounded sets by employing a recent advancement in polynomial optimization, known as the *homogenization approach* [40]. This technique enables us to project potentially unbounded regions into bounded regions in the projective space, hence removing the restriction imposed by Archimedean condition. Additionally, we discuss two alternative techniques for obtaining necessary conditions and compare their differences. (See Section 4)
(3) We implement two algorithms based on the sufficient condition and the necessary condition, respectively. These algorithms are tested over a set of benchmarks with unbounded domains adapted from the literature. Experimental results demonstrate that the necessary condition is more expressive than the sufficient condition. (See Sections 5 and 6)

In summary, our contributions include a clear exposition of the problem, an exploration of necessary conditions for unbounded sets, and practical implementations and experimental evaluations of the algorithms.

Organization. The rest of this paper is organized as follows: Section 2 formally defines the safety verification problem and introduces algebraic tools that will be used. Section 3 formulates the barrier certificate synthesis problems as polynomial feasible problems and explains the difference between using the necessary condition and the sufficient condition. Section 4 investigates into three approaches for obtaining necessary conditions for systems over unbounded domains, with an emphasis on the homogenization approach. Section 5 discusses the practical computation details and Section 6 reports the experimental results. Finally, Section 7 concludes the paper.

2 PRELIMINARIES

In this section, we first fix basic notions used throughout this paper. Afterwards, we recap necessary concepts concerning safety verification problems and sum-of-squares relaxations.

Basic Notations. Let $\mathbb{N}$ denote the set of all natural numbers, $\mathbb{N}_{[m,n]}$ represents the index set $\{m, m+1, \ldots, n\}$ for any naturals m, n such that $m \leq n$. Let $\mathbb{R}, \mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ denote the set of reals, non-negative real numbers and the set of positive real numbers respectively. By convention, we use boldface letters to denote vectors and vector-valued functions, e.g., $\boldsymbol{x} = (x_1, \ldots, x_n)$ denotes a state variable and $\boldsymbol{f} = (f_1, \ldots, f_n)$ denotes a vector field. For vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, let $\|\boldsymbol{x}\| \hat{=} \sqrt{\sum_{i=1}^n x_i^2}$ denote the standard Euclidean norm, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \hat{=} \sum_{i=1}^n x_i y_i$ denote the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$.

Let $\mathbb{R}[\boldsymbol{x}]$ denote the set of polynomials in variables $\boldsymbol{x}$ with real coefficients, $\mathbb{R}^d[\boldsymbol{x}]$ denote the set of polynomials with degree up to d. A basic semialgebraic set $\mathcal{K} \subseteq \mathbb{R}^n$ is of the form $\{\boldsymbol{x} \in \mathbb{R}^n : p_1(\boldsymbol{x}) \triangleright 0, \ldots, p_m(\boldsymbol{x}) \triangleright 0\}$, where $p_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and $\triangleright \in \{\geq, >\}$. A basic semialgebraic set is considered *closed* when its defining polynomials contain only non-strict inequalities. Semialgebraic sets are formed as unions of basic semialgebraic sets. i.e., $\bigcup_{i=1}^n \mathcal{K}_i$, where each $\mathcal{K}_i$ is a basic semialgebraic set. For any (semialgebraic) set $S \subseteq \mathbb{R}^n$, $\mathrm{cl}(S)$ denotes the closure of S.

2.1 Safety Verification Problems

We consider a class of dynamical systems featuring differential dynamics governed by ordinary differential equations (ODEs) of autonomous type:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\dot{\boldsymbol{x}}$ denotes its temporal derivative dx/dt, and $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial vector field, i.e., each component f_i of $\boldsymbol{f}$ is a polynomial. Since a polynomial vector field is locally Lipschitz continuous, ODE (1) admits an unique *solution* (or *trajectory*), denoted as $\xi_{\boldsymbol{x}_0} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, from any initial state $\boldsymbol{x}_0 \in \mathbb{R}^n$, such that

$$\text{i) } \xi_{\boldsymbol{x}_0}(0) = \boldsymbol{x}_0, \qquad \text{ii) } \left.\frac{\mathrm{d}\xi_{x_0}}{\mathrm{d}t}\right|_{t=t'} = \boldsymbol{f}(\xi_{\boldsymbol{x}_0}(t')),\ \forall t' \in \mathbb{R}_{\geq 0}.$$

Given a polynomial $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, the Lie derivative of $p(\boldsymbol{x})$ w.r.t. a vector filed f is denoted by $\mathfrak{L}_f p(\boldsymbol{x}) \triangleq \langle \frac{\partial}{\partial \boldsymbol{x}} p(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}) \rangle$. Intuitively, Lie derivative $\mathfrak{L}_f p$ captures the evolution of p along the vector field f.

Safety Verification Problems. Given dynamical system Eq. (1) with domain $\mathcal{X} \subseteq \mathbb{R}^n$, initial set $\mathcal{I} \subset \mathcal{X}$, and unsafe set $\mathcal{U} \subset \mathcal{X}$, the *safety verification problem* asks whether $\mathcal{U}$ is reachable from any state in $\mathcal{I}$ within $\mathcal{X}$. Formally, let $\mathcal{R}$ denote the reachable set

$$\begin{aligned}\mathcal{R} \triangleq \{\boldsymbol{x} \in \mathcal{X} :\ \exists t \in \mathbb{R}_{\geq 0}, \exists \boldsymbol{x}_0 \in \mathcal{I},\ \text{such that}\\ \boldsymbol{x} = \xi_{\boldsymbol{x}_0}(t) \wedge \forall \tau \in [0,t].\ \xi_{\boldsymbol{x}_0}(\tau) \in \mathcal{X}\},\end{aligned}$$

the system is said to be *safe* if $\mathcal{U} \cap \mathcal{R} = \emptyset$, and *unsafe* otherwise.

The safety verification problem can be readily addressed when the computability of the reachable set $\mathcal{R}$ is established. Nevertheless, for the majority of nonlinear systems, the direct computation, or even approximate estimation, of reachable sets typically proves intractable.

In this paper, we restrict our focus to the case when $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{U}$ are closed basic semialgebraic sets described by

$$\begin{aligned}\mathcal{I} &= \left\{\boldsymbol{x} \in \mathcal{X} : g_i^{\mathcal{I}}(\boldsymbol{x}) \geq 0,\ \text{for } i \in \mathbb{N}_{[1,m_i]}\right\},\\ \mathcal{U} &= \left\{\boldsymbol{x} \in \mathcal{X} : g_i^{\mathcal{U}}(\boldsymbol{x}) \geq 0,\ \text{for } i \in \mathbb{N}_{[1,m_u]}\right\},\\ \mathcal{X} &= \left\{\boldsymbol{x} \in \mathcal{X} : g_i^{\mathcal{X}}(\boldsymbol{x}) \geq 0,\ \text{for } i \in \mathbb{N}_{[1,m_x]}\right\}.\end{aligned}$$

2.2 Sum-of-Squares Relaxations

Sum-of-squares relaxation is a well-established technique for solving polynomial optimization problems. In what follows, we introduce fundamental concepts and theorems pertinent to this technique. For interested readers, we recommend [15, 46] for a detailed treatment of this topic.

Sum-of-Squares Polynomials. Given $S \subseteq \mathbb{R}^n$, we say $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is *nonnegative* (resp. *strictly positive*) over S if $p(\boldsymbol{x}) \geq 0$ (resp. $p(\boldsymbol{x}) > 0$) for any $\boldsymbol{x} \in S$. Sum-of-squares polynomials forms an important subset of polynomials that are nonnegative globally over $\mathbb{R}^n$. A polynomial $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is said to be a *sum-of-squares* polynomial if it can be expressed as $p(\boldsymbol{x}) = \sum_{i=1}^m p_i(\boldsymbol{x})^2$, where $p_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ for each i. Similar to $\mathbb{R}[\boldsymbol{x}]$ and $\mathbb{R}^d[\boldsymbol{x}]$, we use $\Sigma[\boldsymbol{x}]$ and $\Sigma^d[\boldsymbol{x}]$ to denote the set of sum-of-squares polynomials and sum-of-squares polynomials of degree up to d in variables $\boldsymbol{x}$, respectively.

Putinar's Theorem. Let $\mathcal{K}$ be a closed basic semialgebraic set described by

$$\mathcal{K} \triangleq \left\{\boldsymbol{x} \in \mathbb{R}^n : p_1(\boldsymbol{x}) \geq\ 0, \ldots, p_m(\boldsymbol{x}) \geq\ 0\right\}. \tag{2}$$

The set of polynomials

$$\mathbf{QM}(p_1, p_2, \ldots, p_m) \triangleq \{\sigma_0 + \sum_{i=1}^m \sigma_i p_i \mid \sigma_i \in \Sigma[\boldsymbol{x}] \text{ for } i \in \mathbb{N}_{[0,m]}\}$$

is called the *quadratic module* generated by description polynomials of $\mathcal{K}$. A quadratic module **QM** is *Archimedean*, or satisfies *Archimedean condition*, if $N - \|\boldsymbol{x}\|^2 \in \mathbf{QM}$ for some constant $N \in \mathbb{N}$.

Since a sum-of-squares polynomial $\sigma(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]$ is nonnegative over $\mathbb{R}^n$, the following result trivially holds.

Lemma 1. *Given $\mathcal{K}$ as defined in Eq. (2), then*

$$f(\boldsymbol{x}) \in \boldsymbol{QM}(p_1, \ldots, p_m) \implies f(\boldsymbol{x}) \geq 0 \text{ over } \mathcal{K}.$$

An important result in real algebraic geometry is Putinar's Positivstellensatz, which states that, under Archimedean condition, the quadratic module $\mathbf{QM}(p_1, \ldots, p_m)$ contains all polynomials strictly positive over $\mathcal{K}$.

Theorem 2 (Putinar's Positivstellensatz [46, 58]). *Given $\mathcal{K}$ as defined in Eq. (2) and a polynomial $f \in \mathbb{R}[\boldsymbol{x}]$, if $\boldsymbol{QM}(p_1, \ldots, p_m)$ is Archimedean, then*

$$f(\boldsymbol{x}) > 0 \text{ over } \mathcal{K} \implies f(\boldsymbol{x}) \in \boldsymbol{QM}(p_1, \ldots, p_m).$$

We would like to make two remarks regarding the above theorem.

(1) First, it's crucial to note that in Theorem 2, we require $f(\boldsymbol{x}) > 0$ over $\mathcal{K}$, whereas in Lemma 1, we have $f(\boldsymbol{x}) \geq 0$ over $\mathcal{K}$. This distinction will be significant in our theoretical analysis in Section 3 and Section 4. Nonetheless, in practical computations, numerical solvers cannot distinguish between $\geq$ and $>$, so this distinction can be disregarded in practice.
(2) Second, when $\mathcal{K}$ is bounded, we can ensure that Archimedean condition holds by using the "big-ball trick". Given that $\mathcal{K}$ is bounded, there always exists an $N \in \mathbb{N}$ such that
$$\mathcal{K} \subseteq \left\{\boldsymbol{x} \in \mathbb{R}^n : N - \|\boldsymbol{x}\|^2 \geq 0\right\}.$$
This allowing us to assume the description polynomials of $\mathcal{K}$ contain a redundant constraint $N - \|\boldsymbol{x}\|^2 \geq 0$. In this manner, the quadratic module $\mathbf{QM}(p_1, \ldots, p_m, N - \|\boldsymbol{x}\|^2)$ always satisfies Archimedean condition.

Theorem 2 signifies a computationally feasible method for identifying polynomials that exhibit strict positivity over a fundamental semialgebraic set, while ensuring completeness. This method can be applied to polynomial optimization problems. Considering, for example, the problem that minimizing a polynomial $f(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ over a *bounded* semialgebraic set $\mathcal{K}$ as defined in Eq. (2), that is [1]

$$\begin{cases} \max & \gamma \\ s.t. & \forall \boldsymbol{x} \in \mathcal{K} : f(\boldsymbol{x}) - \gamma > 0. \end{cases} \tag{3}$$

Utilizing Theorem 2, we can reformulate the optimization problem (3) into a new program that incorporates sum-of-squares constraints:

$$\begin{cases} \max & \gamma \\ s.t. & f(\boldsymbol{x}) - \gamma = \sigma_0(\boldsymbol{x}) + \sum_{i=1}^{m} \sigma_i(\boldsymbol{x}) p_i(\boldsymbol{x}) + \sigma_{m+1}(\boldsymbol{x})(N - \|\boldsymbol{x}\|^2), \\ & \sigma_i(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \text{ for } i \in \mathbb{N}_{[0,m+1]}. \end{cases} \tag{4}$$

Sum-of-Squares Relaxations. Solving optimization program Prog. (4) directly is intractable, as we lack knowledge regarding the degrees of both $f(\boldsymbol{x})$ and $\sigma_i(\boldsymbol{x})$ for $i \in \mathbb{N}_{[0,m+1]}$. However, it is always possible to approximate Prog. (4) (and its optimal value) by solving a series of new programs, with each new program representing a relaxation of Prog. (4) [53]. The main idea is to impose restrictions on the maximum degree of constraints. For example, given $d \in \mathbb{N}$ such that $d \geq \max\{\deg(f), \deg(p_1), \ldots, \deg(p_m)\}$, the d-th relaxation of Prog. (4) is defined as follows:

$$\begin{cases} \max & \gamma \\ s.t. & f(\boldsymbol{x}) - \gamma = \sigma_0(\boldsymbol{x}) + \sum_{i=1}^{m} \sigma_i(\boldsymbol{x}) p_i(\boldsymbol{x}) + \sigma_{m+1}(\boldsymbol{x})(N - \|\boldsymbol{x}\|^2), \\ & f(\boldsymbol{x}) \in \mathbb{R}^d[\boldsymbol{x}], \quad \sigma_0(\boldsymbol{x}) \in \Sigma^d[\boldsymbol{x}], \quad \sigma_{m+1} \in \Sigma^{d-2}[\boldsymbol{x}], \\ & \sigma_i(\boldsymbol{x}) \in \Sigma^{d-\deg(p_i)}[\boldsymbol{x}], \text{ for } i \in \mathbb{N}_{[1,m]}. \end{cases} \tag{5}$$

where the decision variables consist of the variable γ as well as the unknown coefficients in $\sigma_i(\boldsymbol{x})$. In Section 5, we elucidate the process of converting Prog. (5) into a semidefinite program. This transformation enables efficient polynomial-time solutions, leveraging techniques such as interior-point methods. Furthermore, as the degree bound d increases, the series of sum-of-squares relaxations in Prog. (5) yields progressively more precise approximations for the optimal value of Prog. (3).

3 BARRIER CERTIFICATE CONDITIONS AS POLYNOMIAL FEASIBLE PROBLEMS

In this section, we commence by revisiting various definitions of barrier certificates. Subsequently, from a computational standpoint, we rephrase the problem of synthesizing barrier certificates as a category of polynomial feasibility problems. Following this, we explain why existing methods are conservative when considering unbounded regions and proceed to formalize the primary problem addressed in this paper.

3.1 Barrier Certificates

A *differential invariant* is a subset $\Phi \subseteq \mathcal{X}$ such that any trajectory starting from Φ stays within Φ forever.

[1] In this paper, we do not distinguish between sup and max in optimization programs.

Definition 3 (Differential invariant). *A set $\Phi \subseteq \mathbb{R}^n$ is a differential invariant of the system (1) if and only if*

$$\forall \boldsymbol{x}_0 \in \Phi, \forall t \in \mathbb{R}_{\geq 0}.\ \xi_{\boldsymbol{x}_0}(t) \in \Phi.$$

Utilizing the concept of differential invariants, we can verify the safety of a system without explicitly computing the reachable set. The key idea is to find a differential invariant $Inv \subset \mathcal{X}$ such that $\mathcal{I} \subseteq Inv$ and $\mathcal{U} \subseteq \mathcal{X} \setminus Inv$. According to its definition, Inv serves as an over-approximation of the reachable set $\mathcal{R}$, thereby substantiating safety of the system.

Barrier certificates encapsulate the conditions requisite for a zero sub-level set of the form

$$\left\{\boldsymbol{x} \in \mathbb{R}^n : B(\boldsymbol{x}) \leq 0\right\},$$

to become a differential invariant, where $B(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ is a real-valued differentiable function. To ensure computational tractability, the function $B(\boldsymbol{x})$ is commonly constrained to polynomial forms. Among various certificates, the non-convex barrier certificate stands out as the first simple yet efficacious barrier condition.

Theorem 4 (Non-Convex Barrier Certificates [56]). *Given system (1) with sets $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{U}$, the system is safe if there exists a non-convex barrier certificate, namely a polynomial $B(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ satisfying the following conditions[2]:*

$$\forall \boldsymbol{x} \in \mathcal{I}.\quad B(\boldsymbol{x}) \leq 0, \tag{6}$$

$$\forall \boldsymbol{x} \in \mathcal{U}.\quad B(\boldsymbol{x}) > 0, \tag{7}$$

$$\forall \boldsymbol{x} \in \mathcal{X}.\quad B(\boldsymbol{x}) = 0 \implies \mathfrak{L}_f B(\boldsymbol{x}) < 0. \tag{8}$$

Intuitively, conditions (6) and (7) demand that the barrier $B(\boldsymbol{x})$ separates $\mathcal{I}$ from $\mathcal{U}$, while Eq. (8) imposes constraints on the Lie derivatives of points located on the boundary of $B(\boldsymbol{x})$, thereby encoding the requirement that the zero sub-level set of $B(\boldsymbol{x})$ serves as a differential invariant.

The set of all barriers that satisfy Eqs. (6) to (8) comprises a non-convex set, primarily due to the constraint $B(\boldsymbol{x}) = 0$ in Eq. (8). This non-convexity poses challenges when attempting to find a non-convex barrier numerically. To resolve this issue, [56] further strengthens Eq. (8) into a convex condition

$$\forall \boldsymbol{x} \in \mathcal{X}.\ \mathfrak{L}_f B(\boldsymbol{x}) \leq 0, \tag{9}$$

and refers functions that satisfying constraints Eqs. (6), (7) and (9) as *convex barrier certificates.*

Subsequent research in barrier certificates primarily addresses the relaxation of conditions (8) and (9), aiming to enhance expressiveness while preserving the invariant property. In what follows, we recall two important results pertaining to these two research directions.

Theorem 5 (General Convex Barrier Certificates [18]). *Given system (1) with sets $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{U}$, the system is safe if there exists a general convex barrier certificate, namely a polynomial $B(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ satisfying Eq. (6), Eq. (7), and the following condition*

$$\forall \boldsymbol{x} \in \mathcal{X}.\ \mathfrak{L}_f B(\boldsymbol{x}) \leq \omega(B(\boldsymbol{x})), \tag{10}$$

where $\omega : \mathbb{R} \to \mathbb{R}$ is the derivative of some continuously differential function b such that $b' = \omega(b)$, $b(x(0)) \leq 0$, and $b(x(t)) \leq 0$ for all $t \in \mathbb{R}_{\geq 0}$.

[2] The original formulation of condition (8) wrote $\mathfrak{L}_f B(\boldsymbol{x}) \leq 0$, which is incorrect. See Footnote (5) in [66] for more details.

Theorem 5 was initially introduced in [18] as a general approach for relaxing the condition specified in Eq. (9), all the while preserving the convex nature of convex barrier certificates. In practice, to apply Theorem 5, one must predefine the function $\omega(\cdot)$. A straightforward yet effective choice is to define $\omega(x) = \lambda x$, where λ is a real constant number. In this case, the definition simplifies to what is commonly referred to as *exponential-type barrier certificates*, as introduced in [43].

Theorem 6 (Invariant Barrier Certificates [70]). *Give system (1) with sets $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{U}$, the system is safe if there exists an invariant barrier certificate, namely a polynomial $B(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ satisfying Eqs. (6) and (7), and the following condition*

$$\forall \boldsymbol{x} \in \mathcal{X}.\ \bigwedge_{i=1}^{N_{B,f}} \left(\bigwedge_{j=0}^{i-1} \mathfrak{L}_f^j B(\boldsymbol{x}) = 0 \implies \mathfrak{L}_f^i B(\boldsymbol{x}) \le 0 \right), \tag{11}$$

where $N_{B,f} \ge 1$ is an integer (determined by $B(\cdot)$ and f) serves as the completeness threshold, and the operator $\mathfrak{L}_f^i$ denotes the i-th order Lie derivative defined inductively as

$$\mathfrak{L}_f^i B(\boldsymbol{x}) \triangleq \begin{cases} B(\boldsymbol{x}), & i = 0, \\ \langle \frac{\partial}{\partial \boldsymbol{x}} \mathfrak{L}_f^{i-1} B(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}) \rangle, & i \ge 1. \end{cases}$$

Theorem 6 gives the *necessary and sufficient* condition for a polynomial $B(\boldsymbol{x})$ satisfying Eq. (6) and Eq. (7) to be a differential invariant, and therefore is the weakest possible (non-convex) barrier certificate condition.

Remark. *In this paper, our focus lies on the scenario wherein a differential invariant is characterized by the zero sub-level set of a single polynomial. It is worth noting that the condition for a general semialgebraic set to be a differential invariant can be represented in similar forms [47], thereby making the method presented in this paper applicable in such cases as well.*

3.2 Polynomial Feasible Problems

In the following, we abstract away from distinctions among various definitions of barrier certificates and concentrate on the general procedure of resolving constraints. Clearly, the synthesis of a barrier certificate amounts to address the subsequent polynomial feasibility problem, which includes constraints related to nonnegativity and strict positivity of polynomials:

$$\begin{cases} \text{find} & \boldsymbol{a} \\ s.t. & f_i(\boldsymbol{x};\boldsymbol{a}) \ge 0 \quad \text{on } \mathcal{K}_i, i \in \mathbb{I}, \\ & f_j(\boldsymbol{x};\boldsymbol{a}) > 0 \quad \text{on } \mathcal{K}_j, j \in \mathbb{J}, \end{cases} \tag{12}$$

where, for any $r \in \mathbb{I} \cup \mathbb{J}$, $f_r(\boldsymbol{x};\boldsymbol{a})$ is a polynomial in variable $\boldsymbol{x}$ with parameters $\boldsymbol{a}$, and $\mathcal{K}_r$ is a basic closed semialgebraic set defined by

$$\mathcal{K}_r \triangleq \left\{ \boldsymbol{x} \in \mathbb{R}^n : p_{r,1}(\boldsymbol{x}) \ge 0, \ldots, p_{r,m_r}(\boldsymbol{x}) \ge 0 \right\}. \tag{13}$$

Here we permit the polynomials $p_{r,k}$ to incorporate parameters $\boldsymbol{a}$ for $k \in \mathbb{N}_{[1,m_r]}$, but refrain from explicitly specifying these parameters in our notation for simplicity[3].

By utilizing Lemma 1 and Theorem 2, we can derive the sufficient condition and the necessary condition for Prog. (12), respectively.

Theorem 7 (Sufficient Condition). *For any $\epsilon_0 \in \mathbb{R}_{>0}$, if $\boldsymbol{a}_0$ is a solution of Prog. (14), then $\boldsymbol{a}_0$ is also a solution to Prog. (12).*

$$\begin{cases} \textit{find} & \boldsymbol{a} \\ s.t. & f_i(\boldsymbol{x};\boldsymbol{a}) = \sigma_{i,0}(\boldsymbol{x}) + \sum_{k=1}^{m_i} \sigma_{i,k}(\boldsymbol{x}) p_{i,k}(\boldsymbol{x}), \quad \textit{for } i \in \mathbb{I}, \\ & f_j(\boldsymbol{x};\boldsymbol{a}) - \epsilon_0 = \sigma_{j,0}(\boldsymbol{x}) + \sum_{k=1}^{m_j} \sigma_{j,k}(\boldsymbol{x}) p_{j,k}(\boldsymbol{x}), \quad \textit{for } j \in \mathbb{J} \\ & \sigma_{i,k} \in \Sigma[\boldsymbol{x}], \quad \textit{for } i \in \mathbb{I}, k \in \mathbb{N}_{[0,m_i]} \\ & \sigma_{j,k} \in \Sigma[\boldsymbol{x}], \quad \textit{for } j \in \mathbb{J}, k \in \mathbb{N}_{[0,m_j]}. \end{cases} \tag{14}$$

Proof. By directly applying Lemma 1. □

Under Archimedean condition, a necessary condition follows directly form Theorem 2.

Theorem 8 (Necessary Condition, the Bounded Case). *Suppose $\mathcal{K}_r \subseteq \left\{ \boldsymbol{x} \in \mathbb{R}^n : N - \|\boldsymbol{x}\|^2 \ge 0 \right\}$ for all $r \in \mathbb{I} \cup \mathbb{J}$. For any $\epsilon_0 \in \mathbb{R}_{>0}$, if $\boldsymbol{a}_0$ is a solution to Prog. (12), then $\boldsymbol{a}_0$ is also a solution of Prog. (15):*

$$\begin{cases} \textit{find} & \boldsymbol{a} \\ s.t. & f_i(\boldsymbol{x};\boldsymbol{a}) + \epsilon_0 = \sigma_{i,0}(\boldsymbol{x}) + \sum_{k=1}^{m_i} \sigma_{i,k}(\boldsymbol{x}) p_{i,k}(\boldsymbol{x}) \\ & \qquad + \sigma_{i,m_i+1}(\boldsymbol{x})(N - \|\boldsymbol{x}\|^2), \quad \textit{for } i \in \mathbb{I}, \\ & f_j(\boldsymbol{x};\boldsymbol{a}) = \sigma_{j,0}(\boldsymbol{x}) + \sum_{k=1}^{m_j} \sigma_{j,k}(\boldsymbol{x}) p_{j,k}(\boldsymbol{x}) \\ & \qquad + \sigma_{j,m_j+1}(\boldsymbol{x})(N - \|\boldsymbol{x}\|^2), \quad \textit{for } j \in \mathbb{J}, \\ & \sigma_{i,k} \in \Sigma[\boldsymbol{x}], \quad \textit{for } i \in \mathbb{I},\ k \in \mathbb{N}_{[0,m_i+1]} \\ & \sigma_{j,k} \in \Sigma[\boldsymbol{x}], \quad \textit{for } j \in \mathbb{J},\ k \in \mathbb{N}_{[0,m_j+1]}. \end{cases} \tag{15}$$

Proof. By directly applying Theorem 2. □

Most existing works on barrier certificates, such as [18, 43, 56, 66], primarily focus on utilizing the sufficient condition in the form of Prog. (14). Although [70] discusses the necessary condition, it still relies on the sufficient condition in practical computation. The preference for the sufficient condition stems from two reasons. First, when the redundant polynomial $N - \|\boldsymbol{x}\|^2$ is included in the description polynomials of $\mathcal{K}_r$, for $r \in \mathbb{I} \cup \mathbb{J}$, Prog. (14) and Prog. (15) coincide as ϵ_0 approaches 0. Therefore, there is not much loss in expressiveness for utilizing the sufficient condition. Second, employing the necessary condition requires an additional verification step to rule out fake solutions, as $f_i(\boldsymbol{x};\boldsymbol{a}) \ge 0$ is replaced by $f_i(\boldsymbol{x};\boldsymbol{a}) + \epsilon_0 \ge 0$.

However, when $\mathcal{K}_r$ are allowed to be unbounded, the sufficient condition remains available but becomes conservative since the "big-ball trick" can not be employed. Consequently, algorithms relying on Prog. (14) may fail to identify potential solutions. One possible approach is to iteratively solve Prog. (15) while gradually increasing the value of N until a solution is found (the obtained solution can be verified over the unbounded domain). This method is evidently impractical since it necessitates solving a program for each value of N and offers no guarantee of termination.

[3] Polynomials $p_{r,k}$ will contain unknown parameters when we want to synthesize a non-convex barrier, as, for example, constraint $\forall \boldsymbol{x} \in \mathcal{X}.\, B(\boldsymbol{x}) = 0 \implies \mathfrak{L}_f B(\boldsymbol{x}) < 0$ can be equivalently formulated as $\forall \boldsymbol{x} \in \mathcal{X} \cap \{\boldsymbol{x} \in \mathbb{R}^n \mid B(\boldsymbol{x}) = 0\}.\, \mathfrak{L}_f B(\boldsymbol{x}) < 0$.

To summarize, the main problem this paper aims to solve is that: **How can we derive a necessary condition for Prog. (12) with unbounded sets and utilize it in barrier certificate synthesis?**

4 NECESSARY CONDITIONS FOR POLYNOMIAL FEASIBLE PROBLEMS WITH UNBOUNDED SETS

In this section, we examine several approaches that can be employed to address the previously mentioned problem. Among these techniques, the primary focus of this paper is the homogenization approach, which is introduced in Section 4.1. Subsequently, we delve into a discussion of two alternative approaches and offer a comparative analysis with the homogenization approach.

4.1 Homogenization Approach

We first fix some notations. Let x_0 be a fresh variable and denote $\tilde{\boldsymbol{x}} = (x_0, \boldsymbol{x})$. Given a polynomial $p(\boldsymbol{x})$ of degree d, the *homogenization* of $p(\boldsymbol{x})$ w.r.t. variable x_0 is a new polynomial $\tilde{p}$ defined by $\tilde{p}(\tilde{\boldsymbol{x}}) \hat{=} x_0^d p(x_1/x_0, \ldots, x_n/x_0)$. Suppose $\mathcal{K}$ is a semialgebraic set as described in Eq. (2), we introduce two related sets as follows:

$$\tilde{\mathcal{K}} \hat{=} \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : \tilde{p_1}(\tilde{\boldsymbol{x}}) \geq 0, \ldots, \tilde{p}_m(\tilde{\boldsymbol{x}}) \geq 0\right\},$$
$$\tilde{\mathcal{K}}^b \hat{=} \tilde{\mathcal{K}} \cap \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 \geq 0, \|\tilde{\boldsymbol{x}}\|^2 = 1\right\}.$$

It is straightforward to see that the projection map

$$\varphi : \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 > 0, \|\tilde{\boldsymbol{x}}\|^2 = 1\right\} \to \mathbb{R}^n, (x_0, \boldsymbol{x}) \mapsto \frac{\boldsymbol{x}}{x_0}$$

defines an one-to-one mapping between $\tilde{\mathcal{K}}^b \setminus \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 = 0\right\}$ and $\mathcal{K}$. By employing the inverse mapping φ^{-1}, we can transform a potentially *unbounded* set into a *bounded* set located on the unit sphere within $\mathbb{R}^{n+1}$. Moreover, note that points with $x_0 = 0$ in $\mathbb{R}^{n+1}$ correspond to points at infinity in $\mathbb{R}^n$. This encourages us to take the points at infinity into consideration. The related concept is captured by the following definition.

Definition 9 (closed at infinity [52]). *A basic semialgebraic set $\mathcal{K}$ is closed at ∞ if*

$$cl(\tilde{\mathcal{K}} \cap \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 > 0\right\}) = \tilde{\mathcal{K}} \cap \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 \geq 0\right\}.$$

We would like to emphasize that closure at infinity is a generic property for semialgebraic set, and its manifestation may be contingent upon the selection of descriptive polynomials [35]. To check whether a semialgebraic set is closed at ∞, one can rely on Thm. 2.11 in [35].

Example 10. *[40] Consider two semialgebraic sets*

$$S_1 = \left\{(x_1, x_2) \in \mathbb{R}^2 : x_1 - x_2^2 \geq 0\right\},$$
$$S_2 = \left\{(x_1, x_2) \in \mathbb{R}^2 : x_1 - x_2^2 \geq 0, x_1 \geq 0\right\}.$$

It is easy to check $S_1 = S_2$. However, the set S_2 is closed at ∞ while S_1 is not, since

$$(0, -1, 0) \in \tilde{S_1} \cap \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 \geq 0\right\},$$
$$(0, -1, 0) \notin cl(\tilde{S_1} \cap \left\{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : x_0 > 0\right\}).$$

The following theorem constitutes the foundational element of the homogenization formulation.

Theorem 11 ([40, Lem 3.2]). *When $\mathcal{K}$ is closed at ∞,*

$$f(\boldsymbol{x}) \geq 0 \text{ over } \mathcal{K} \iff \tilde{f}(\tilde{\boldsymbol{x}}) \geq 0 \text{ over } \tilde{\mathcal{K}}^b$$

Utilizing Theorem 11, we can derive a necessary condition for Prog. (12).

Theorem 12. *Suppose $\mathcal{K}_r$ is closed at ∞ for each $r \in \mathbb{I} \cup \mathbb{J}$. For any $\epsilon_0 \in \mathbb{R}_{>0}$, if $\boldsymbol{a}_0$ is a solution of Prog. (12), then $\boldsymbol{a}_0$ is also a solution of the following program:*

$$\begin{cases} \textit{find} \quad \boldsymbol{a} \\ \textit{s.t.} \quad \tilde{f}_r(\tilde{\boldsymbol{x}}; \boldsymbol{a}) + \epsilon_0 = \sigma_{r,0}(\tilde{\boldsymbol{x}}) + \sum_{k=1}^{m_r} \sigma_{r,k}(\tilde{\boldsymbol{x}}) \tilde{p}_{r,k}(\tilde{\boldsymbol{x}}) + \sigma_{r,m_r+1}(\tilde{\boldsymbol{x}}) \cdot x_0 \\ \qquad\qquad + \tau_r(\tilde{\boldsymbol{x}})(\|\tilde{\boldsymbol{x}}\| - 1), \quad \textit{for } r \in \mathbb{I} \cup \mathbb{J}, \\ \quad \sigma_{r,k}(\tilde{\boldsymbol{x}}) \in \Sigma[\tilde{\boldsymbol{x}}], \quad \textit{for } r \in \mathbb{I} \cup \mathbb{J}, k \in \mathbb{N}_{[0, m_r+1]}, \\ \quad \tau_r(\tilde{\boldsymbol{x}}) \in \mathbb{R}[\tilde{\boldsymbol{x}}] \end{cases} \tag{16}$$

Proof. If $\boldsymbol{a}_0$ is a solution to Prog. (12), then we have

$$f_r(\boldsymbol{x}; \boldsymbol{a}_0) \geq 0 \text{ over } \mathcal{K}_r, \quad \text{for } r \in \mathbb{I} \cup \mathbb{J}.$$

By Theorem 11, we further obtain

$$\tilde{f}_r(\tilde{\boldsymbol{x}}; \boldsymbol{a}_0) \geq 0 \text{ over } \tilde{\mathcal{K}_r}^b, \quad \text{for } r \in \mathbb{I} \cup \mathbb{J}.$$

Thus, according to Theorem 2, for any $\epsilon_0 > 0$ and any $r \in \mathbb{I} \cup \mathbb{J}$, function $\tilde{f}_r(\tilde{\boldsymbol{x}}; \boldsymbol{a}_0) + \epsilon_0$ can be expressed as

$$\tilde{f}_r(\tilde{\boldsymbol{x}}; \boldsymbol{a}) + \epsilon_0 = \sigma_{r,0}(\tilde{\boldsymbol{x}}) + \sum_{k=1}^{m_r} \sigma_{r,k}(\tilde{\boldsymbol{x}}) \tilde{p}_{r,k}(\tilde{\boldsymbol{x}}) + \sigma_{r,m_r+1}(\tilde{\boldsymbol{x}}) \cdot x_0 + (\sigma_{r,+}(\tilde{\boldsymbol{x}}) - \sigma_{r,-}(\tilde{\boldsymbol{x}}))(\|\tilde{\boldsymbol{x}}\| - 1),$$

note that the equality constraint $\|\boldsymbol{x}\|^2 - 1 = 0$ is treated as $\|\boldsymbol{x}\|^2 - 1 \geq 0 \wedge \|\boldsymbol{x}\|^2 - 1 \leq 0$. Since any polynomial can be represented as a difference of two sum-of-squares polynomials, we can replace $\sigma_{r,+}(\tilde{\boldsymbol{x}}) - \sigma_{r,-}(\tilde{\boldsymbol{x}})$ by a new polynomial $\tau(\tilde{\boldsymbol{x}}) \in \mathbb{R}[\tilde{\boldsymbol{x}}]$ and hence Prog. (16) is obtained. □

The following example illustrates the power of homogenization.

Example 13. *Let $\boldsymbol{x} = (x_1, x_2)$. It is well-known that Motzkin's polynomial $M(\boldsymbol{x}) = x_1^2 x_2^4 + x_1^4 x_2^2 - 3x_1^2 x_2^2 + 1$ is positive semidefinite, that is, $M(\boldsymbol{x}) \geq 0$ for any $\boldsymbol{x} \in \mathbb{R}^2$, but is not a sum-of-squares polynomial. This means that $M(\boldsymbol{x}) \notin \boldsymbol{QM}(1) = \Sigma[\boldsymbol{x}]$. Nevertheless, by using homogenization and employing a numerical solver, we can find $\sigma \in \Sigma[\tilde{\boldsymbol{x}}]$ and $\tau \in \mathbb{R}[\tilde{\boldsymbol{x}}]$ such that*

$$\tilde{M}(\tilde{\boldsymbol{x}}) = \sigma \cdot x_0 + \tau \cdot (\|\tilde{\boldsymbol{x}}\| - 1)$$

which implies that $\tilde{M}(\tilde{\boldsymbol{x}}) \in \boldsymbol{QM}(x_0, \|\tilde{\boldsymbol{x}}\| - 1, 1 - \|\tilde{\boldsymbol{x}}\|)$. Moreover, if one is not satisfied with numerical solutions, we can utilize the technique described in [61] to prove the existence of a real solution near the numerical solution.

4.2 Comparison with Other Approaches

In this part, we explore two alternative methodologies that can be employed to establish necessary conditions for polynomial feasible problems involving unbounded sets. Both of these approaches are based on representation theorems that extend Putinar's Positivstellensatz.

Necessary Condition based on S-K Theorem. Given a set of polynomials $p_1, \ldots, p_m$ and an index set $\mathbb{I} \subseteq \mathbb{N}_{[1,m]}$. Let $p_{\mathbb{I}} \doteq \prod_{i \in \mathbb{I}} p_i$, the set

$$\mathbf{P}(p_1, \ldots, p_m) \doteq \left\{ \sum_{\mathbb{I} \subseteq \mathbb{N}_{[1,m]}} \sigma_{\mathbb{I}} p_{\mathbb{I}} : \sigma_{\mathbb{I}} \in \Sigma[\boldsymbol{x}] \right\}$$

is called a *preordering* generated by $p_1, \ldots, p_m$. The following theorem presents a generalization of Theorem 2 and does not rely on Archimedean condition.

Theorem 14 (Stengle-Krivine Theorem[46, 67]). *Given $f \in \mathbb{R}[\boldsymbol{x}]$ and a semialgebraic set $\mathcal{K}$ as defined in Eq. (2), then*

$$f(\boldsymbol{x}) > 0 \text{ over } \mathcal{K} \iff \exists g, h \in \boldsymbol{P}(p_1, \ldots, p_m).\ fg = 1 + h,$$

$$f(\boldsymbol{x}) \geq 0 \text{ over } \mathcal{K} \iff \exists l \in \mathbb{N}, \exists g, h \in \boldsymbol{P}(p_1, \ldots, p_m).\ fg = f^{2l} + h.$$

Since Theorem 14 explicitly distinguishes between $f(\boldsymbol{x}) > 0$ and $f(\boldsymbol{x}) \geq 0$, it can be utilized to derive a *necessary and sufficient* condition for the polynomial feasible problem with unbounded sets. However, a problem arises in the "≥" case, as it requires to fix $l \in \mathbb{N}$ in advance. While there exists a theoretical bound on l that depends on the dimension n and the degree of polynomials $p_1, \ldots, p_m$, it is worth noting that this particular threshold frequently proves impractically large for practical computational applications. Due to this limitation, we choose to utilize exclusively the initial assertion in Theorem 14 to derive a necessary condition, whose proof is quite similar to that of Theorem 12.

Theorem 15. *If $\boldsymbol{a}_0$ is a solution to Prog. (12), then $\boldsymbol{a}_0$ is a solution to the following program.*

$$\begin{cases} \textit{find} \quad \boldsymbol{a} \\ \textit{s.t.} \quad \displaystyle\sum_{\mathbb{I}_i \subseteq \mathbb{N}_{[1,m_i]}} \sigma_{\mathbb{I}_i}(\boldsymbol{x}) p_{\mathbb{I}_i}(\boldsymbol{x}) \cdot (f_i(\boldsymbol{x}; \boldsymbol{a}) + \epsilon_0) = \\ \qquad\qquad 1 + \displaystyle\sum_{\mathbb{I}'_i \subseteq \mathbb{N}_{[1,m_i]}} \sigma_{\mathbb{I}'_i}(\boldsymbol{x}) p_{\mathbb{I}'_i}(\boldsymbol{x}), \quad \textit{for } i \in \mathbb{I}, \\ \quad \displaystyle\sum_{\mathbb{J}_j \subseteq \mathbb{N}_{[1,m_j]}} \sigma_{\mathbb{J}_j}(\boldsymbol{x}) p_{\mathbb{J}_j}(\boldsymbol{x}) \cdot f_j(\boldsymbol{x}; \boldsymbol{a}) = \\ \qquad\qquad 1 + \displaystyle\sum_{\mathbb{J}'_j \subseteq \mathbb{N}_{[1,m_j]}} \sigma_{\mathbb{J}'_j}(\boldsymbol{x}) p_{\mathbb{J}'_j}(\boldsymbol{x}), \quad \textit{for } j \in \mathbb{J}, \\ \quad \sigma_{\mathbb{I}_i}, \sigma_{\mathbb{I}'_i} \in \Sigma[\boldsymbol{x}], \quad \textit{for } i \in \mathbb{I} \\ \quad \sigma_{\mathbb{J}_j}, \sigma_{\mathbb{J}'_j} \in \Sigma[\boldsymbol{x}], \quad \textit{for } j \in \mathbb{J}. \end{cases} \tag{17}$$

Necessary Condition based on P-V Theorem. Another representation theorem extends Theorem 2 by leveraging the denseness of sum-of-squares polynomials. Specifically, when f is nonnegative over a basic semialgebraic set $\mathcal{K}$ as defined in Eq. (2), the quadratic module $\mathbf{QM}(p_1, \ldots, p_m)$ must contain a polynomial that is close to f, although not necessarily equals to f.

Theorem 16 (Putinar-Vasilescu Positivstellensatz [59]). *Given a semialgebraic set $\mathcal{K}$ as defined in Eq. (2) and define $\theta \doteq 1 + \|\boldsymbol{x}\|^2$. Let $f \in \mathbb{R}[\boldsymbol{x}]$ and $d \in \mathbb{N}$ such that $2d > \deg(f)$, then for all $\epsilon > 0$, there exists $k \in \mathbb{N}$ such that*

$$f(\boldsymbol{x}) \geq 0 \text{ over } \mathcal{K} \implies \theta^{2k}(f(\boldsymbol{x}) + \epsilon\theta^d) \in \boldsymbol{QM}(p_1, \ldots, p_m).$$

The ideal case is that $\epsilon = k = 0$, where Theorem 16 degenerates to Theorem 2. Following a similar argument of Theorem 12, we can obtain the following necessary condition.

Theorem 17. *Suppose $\boldsymbol{a}_0$ is a solution of Prog. (12), and $d \in \mathbb{N}$ a constant integer such that $2d > \deg(f_r)$ for any r, then there exists $k_r \in \mathbb{N}$ for all $r \in \mathbb{I} \cup \mathbb{J}$ such that $\boldsymbol{a}_0$ is a solution of the following program:*

$$\begin{cases} \textit{find} \quad \boldsymbol{a} \\ \textit{s.t.} \quad \theta^{2k_r}(f_r(\boldsymbol{x}; \boldsymbol{a}) + \epsilon\theta^d) = \sigma_{r,0}(\boldsymbol{x}) + \displaystyle\sum_{k=1}^{m_r} \sigma_{r,k}(\boldsymbol{x}) p_{r,k}(\boldsymbol{x}), r \in \mathbb{I} \cup \mathbb{J} \\ \quad \sigma_{r,k}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \quad \textit{for } r \in \mathbb{I} \cup \mathbb{J}, k \in \mathbb{N}_{[0,m_r+1]} \end{cases} \tag{18}$$

Comparison. Both Theorem 15 and Theorem 17 encode a polynomial feasible problem into constraints involving sum-of-squares polynomials. Nevertheless, when we contrast these two methods with the homogenization approach, it becomes apparent that they are not practically viable for computational purposes.

The primary disadvantage of Theorem 15 is the exponential increase in the number of introduced sum-of-squares polynomials. For instance, if we consider $\mathcal{K} \subset \mathbb{R}^n$ as a polyhedron described by n linear inequalities, we would need to introduce 2^n unknown sum-of-squares polynomials to characterize $f(\boldsymbol{x}; \boldsymbol{a}) > 0$ over $\mathcal{K}$. Clearly, this approach becomes impractical due to the exponential increase in complexity.

Regarding Theorem 17, note that k_r is unknown and depends on polynomials $f_r, p_{r,1}, \ldots, p_{r,k_m}$ as well as ϵ. Theoretical findings have established the existence of a threshold value c, such that $k_r \geq c$ for each $r \in \mathbb{I} \cup \mathbb{J}$ implies Prog. (18) is solvable [49, Prop. 1]. However, it is worth noting that this threshold value is often impractically large for practical computations.

5 PRACTICAL COMPUTATIONS

In this section, we demonstrate how to use the sufficient condition (Theorem 7) and the necessary condition (Theorem 12) to synthesize general convex barrier certificates (Theorem 5) in practice. We will briefly discuss the cases for synthesizing non-convex barrier certificates and invariant barrier certificates, as they lead to more complex optimization problems beyond SDP.

To synthesize a barrier certificate, we begin by selecting a template, which is a parametric polynomial $B(\boldsymbol{x}; \boldsymbol{a})$ with unknown parameters $\boldsymbol{a}$. This template serves as a representation for the barrier certificate we aim to synthesize. For example, if we intend to synthesize a barrier certificate $B(\boldsymbol{x}) \in \mathbb{R}[x_1, x_2]$ of degree 2, we can set $B(\boldsymbol{x}; \boldsymbol{a})$ as

$$B(\boldsymbol{x}; \boldsymbol{a}) = a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4 x_1 + a_5 x_2 + a_6,$$

which can represent any polynomial of degree 2. Without loss of generality, we always assume that $B(\boldsymbol{x}; \boldsymbol{a})$ is linear in $\boldsymbol{a}$.

Note that a prerequisite for employing the homogenization approach is that the sets $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{U}$ are closed at ∞. In the context of safety verification, the description polynomials of these sets are typically not too complex and the property can be checked manually by definition. When a semialgebraic set is not closed at ∞, we need to find an alternative representation for this set, for example, by adding redundant polynomials as in Example 10. In what follows, we assume that $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{U}$ are closed at ∞.

In order to leverage well-developed optimization techniques, we transform the polynomial feasible problem Prog. (12) into an optimization problem by introducing a real variable γ as the optimization objective. The transformed problem can be formulated as:

$$\begin{cases} \max & \gamma \\ s.t. & f_i(\boldsymbol{x};\boldsymbol{a}) - \gamma \geq 0 \quad \text{ on } \mathcal{K}_i, i \in \mathbb{I}, \\ & f_j(\boldsymbol{x};\boldsymbol{a}) - \gamma > 0 \quad \text{ on } \mathcal{K}_j, j \in \mathbb{J}. \end{cases} \tag{19}$$

It is evident that the original problem Prog. (12) is feasible if and only if $\gamma^* \geq 0$ where γ^* is the optimal value of Prog. (19).

Remark. *A natural idea might be to treat ϵ_0 as a variable and attempt to minimize it. However, minimizing ϵ_0 often leads to significant numerical errors, since ϵ_0 can be made arbitrarily small by scaling the coefficients of sum-of-squares polynomials and parameters $\boldsymbol{a}$. Therefore, we fix ϵ_0 to be a small positive constant (in our experiments, 10^{-5}), and instead maximize the newly introduced variable γ.*

Sufficient Condition. Similar to Theorem 7. we can obtain a strengthened version of Eq. (19) by applying Lemma 1. Then, we derive a series of sum-of-squares relaxations for the resulted optimization problem given by

$$\begin{cases} \max & \gamma \\ s.t. & -B(\boldsymbol{x};\boldsymbol{a}) - \gamma = \sigma_0^{\mathcal{I}}(\boldsymbol{x}) + \sum_{i=1}^{m_i} \sigma_i^{\mathcal{I}}(\boldsymbol{x}) g_i^{\mathcal{I}}(\boldsymbol{x}), \\ & B(\boldsymbol{x};\boldsymbol{a}) - \epsilon_0 - \gamma = \sigma_0^{\mathcal{U}}(\boldsymbol{x}) + \sum_{i=1}^{m_u} \sigma_i^{\mathcal{U}}(\boldsymbol{x}) g_i^{\mathcal{U}}(\boldsymbol{x}), \\ & \omega(B(\boldsymbol{x};\boldsymbol{a})) - \mathfrak{L}_f B(\boldsymbol{x};\boldsymbol{a}) - \gamma = \sigma_0^{\mathcal{X}}(\boldsymbol{x}) + \sum_{i=1}^{m_x} \sigma_i^{\mathcal{X}}(\boldsymbol{x}) g_i^{\mathcal{X}}(\boldsymbol{x}), \\ & \sigma_0^{\mathcal{I}} \in \Sigma^{d_1}[\boldsymbol{x}], \sigma_0^{\mathcal{U}} \in \Sigma^{d_2}[\boldsymbol{x}], \sigma_0^{\mathcal{X}} \in \Sigma^{d_3}[\boldsymbol{x}] \\ & \sigma_i^{\mathcal{I}} \in \Sigma^{d_1 - \deg(g_i^{\mathcal{I}})}[\boldsymbol{x}], \text{ for } i \in \mathbb{N}_{[1,m_i]}, \\ & \sigma_i^{\mathcal{U}} \in \Sigma^{d_2 - \deg(g_i^{\mathcal{U}})}[\boldsymbol{x}], \text{ for } i \in \mathbb{N}_{[1,m_u]}, \\ & \sigma_i^{\mathcal{X}} \in \Sigma^{d_3 - \deg(g_i^{\mathcal{X}})}[\boldsymbol{x}], \text{ for } i \in \mathbb{N}_{[1,m_x]}. \end{cases} \tag{20}$$

where d_1, d_2, and d_3 are degree bounds for the three polynomial equality constraints respectively. Naturally, we require d_1, d_2, and d_3 are large enough such that sum-of-squares constraints above are well-defined.

Necessary Condition. As for the necessary condition, we first derive the homogenization formulation of Eq. (19) based on Theorem 11. Then, by applying Theorem 2, we obtain a relaxed version of Eq. (19), whose sum-of-squares relaxations are given by

$$\begin{cases} \max & \gamma \\ s.t. & -\tilde{B}(\tilde{\boldsymbol{x}};\boldsymbol{a}) - \gamma x_0^{\deg B} + \epsilon_0 = \sigma_0^{\mathcal{I}}(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m_i} \sigma_i^{\mathcal{I}}(\tilde{\boldsymbol{x}}) \tilde{g}_i^{\mathcal{I}}(\tilde{\boldsymbol{x}}) \\ & \qquad + \sigma_{m_i+1}^{\mathcal{I}}(\tilde{\boldsymbol{x}}) \cdot x_0 + \tau^{\mathcal{I}}(\tilde{\boldsymbol{x}})(\|\tilde{\boldsymbol{x}}\|^2 - 1), \\ & \tilde{B}(\tilde{\boldsymbol{x}};\boldsymbol{a}) - \gamma x_0^{\deg B} + \epsilon_0 = \sigma_0^{\mathcal{U}}(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m_u} \sigma_i^{\mathcal{U}}(\tilde{\boldsymbol{x}}) \tilde{g}_i^{\mathcal{U}}(\tilde{\boldsymbol{x}}) \\ & \qquad + \sigma_{m_u+1}^{\mathcal{U}}(\tilde{\boldsymbol{x}}) \cdot x_0 + \tau^{\mathcal{U}}(\tilde{\boldsymbol{x}})(\|\tilde{\boldsymbol{x}}\|^2 - 1), \\ & \tilde{H}(\tilde{\boldsymbol{x}};\boldsymbol{a}) - \gamma x_0^{\deg H} + \epsilon_0 = \sigma_0^{\mathcal{X}}(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m_x} \sigma_i^{\mathcal{X}}(\tilde{\boldsymbol{x}}) \tilde{g}_i^{\mathcal{X}}(\tilde{\boldsymbol{x}}) \\ & \qquad + \sigma_{m_i+1}^{\mathcal{X}}(\tilde{\boldsymbol{x}}) \cdot x_0 + \tau^{\mathcal{X}}(\tilde{\boldsymbol{x}})(\|\tilde{\boldsymbol{x}}\|^2 - 1), \\ & \sigma_0^{\mathcal{I}} \in \Sigma^{d_1}[\tilde{\boldsymbol{x}}], \sigma_0^{\mathcal{U}} \in \Sigma^{d_2}[\tilde{\boldsymbol{x}}], \sigma_0^{\mathcal{X}} \in \Sigma^{d_3}[\tilde{\boldsymbol{x}}], \\ & \tau^{\mathcal{I}} \in \mathbb{R}^{d_1-2}[\tilde{\boldsymbol{x}}], \tau^{\mathcal{U}} \in \mathbb{R}^{d_2-2}[\tilde{\boldsymbol{x}}], \tau^{\mathcal{X}} \in \mathbb{R}^{d_3-2}[\tilde{\boldsymbol{x}}], \\ & \sigma_{m_i+1}^{\mathcal{I}} \in \Sigma^{d_1-1}[\tilde{\boldsymbol{x}}], \sigma_{m_u+1}^{\mathcal{U}} \in \Sigma^{d_2-1}[\tilde{\boldsymbol{x}}], \sigma_{m_x+1}^{\mathcal{X}} \in \Sigma^{d_3-1}[\tilde{\boldsymbol{x}}], \\ & \sigma_i^{\mathcal{I}} \in \Sigma^{d_1 - \deg(g_i^{\mathcal{I}})}[\tilde{\boldsymbol{x}}], \text{ for } i \in \mathbb{N}_{[1,m_i]} \\ & \sigma_i^{\mathcal{U}} \in \Sigma^{d_2 - \deg(g_i^{\mathcal{U}})}[\tilde{\boldsymbol{x}}], \text{ for } i \in \mathbb{N}_{[1,m_u]} \\ & \sigma_i^{\mathcal{X}} \in \Sigma^{d_3 - \deg(g_i^{\mathcal{X}})}[\tilde{\boldsymbol{x}}], \text{ for } i \in \mathbb{N}_{[1,m_x]}. \end{cases} \tag{21}$$

where $H(\boldsymbol{x};\boldsymbol{a}) \hat{=} \omega(B(\boldsymbol{x};\boldsymbol{a})) - \mathfrak{L}_f B(\boldsymbol{x};\boldsymbol{a})$ and d_1, d_2, and d_3 are defined similar to Prog. (20).

It is important to note that relying solely on the necessary condition can sometimes result in unsound solutions. However, in practical applications, we can address these issues by ensuring that ϵ_0 is set to a sufficiently small value and by conducting posterior verification.

Translating into SDP. Let $\mathbf{m}_d(\boldsymbol{x})$ be a column vector with all monomials in $\boldsymbol{x}$ of degree up to d. For example, when $\boldsymbol{x} = (x_1, x_2)$, $\mathbf{m}_2(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$. Any polynomial $p(\boldsymbol{x}) \in \mathbb{R}^{2d}[\boldsymbol{x}]$ can be represented by

$$p(\boldsymbol{x}) = \mathbf{m}_d(\boldsymbol{x})^\top C_p \mathbf{m}_d(\boldsymbol{x}), \tag{22}$$

where $C_p \in \mathbb{R}^{\binom{r+d}{d} \times \binom{r+d}{d}}$ is a real symmetric matrix called the *Gram matrix*. It is well-known that $p(\boldsymbol{x})$ belongs to $\Sigma^{2d}[\boldsymbol{x}]$ if and only if its Gram matrix C_p is positive semidefinite, i.e., $\boldsymbol{x}^\top C_p \boldsymbol{x} \geq 0$ for any $\boldsymbol{x} \in \mathbb{R}^{\binom{r+d}{d}}$, denoted $C_p \succeq 0$. Therefore, both Prog. (20) and Prog. (21) can be translated in to a standard semidefinite program, where the decision variables are the unknown coefficients of sum-of-squares polynomials and parameters $\boldsymbol{a}$.

Complexity. Roughly speaking, the complexity for solving sum-of-squares relaxations depends on the number of monomials in $\mathbf{m}_d(\boldsymbol{x})$, i.e., $\binom{n+d}{n}$, which is an upper bound for unknown coefficients in the template and unknown sum-of-squares polynomials. As for the homogenization formulation Prog. (21), since an extra variable x_0 is introduced, the number of monomials in unknown sum-of-squares polynomials becomes $\binom{n+d+1}{n} = (1 + \frac{d}{n+1})\binom{n+d}{n}$. In practice, the sum-of-squares relaxation is tractable when d and n are relatively small (with $\binom{n+d}{n}$ of up to a few hundreds [61]) and such increase can be ignored. Furthermore, the efficiency can be (significantly) improved by exploiting algebraic structures of the dynamical systems [48].

Table 1: Experimental results for synthesizing general convex barrier certificates.

					Sufficient		Necessary	
system name	dim	deg(f)	unbounded	deg(B)	time(s)	verified	time(s)	verified
vector[66]	2	1	$\mathcal{X}$	2	0.01	✓	0.03	✓
			$\mathcal{I},\mathcal{U},\mathcal{X}$	2	0.01	✗	0.06	✓
barrier[56]	2	3	$\mathcal{X}$	2	0.01	✓	0.14	✗
			$\mathcal{I},\mathcal{U},\mathcal{X}$	2	0.01	✗	0.19	✓
lie-der[47]	2	1	$\mathcal{X}$	1	0.01	✗	0.04	✓
			$\mathcal{I},\mathcal{U},\mathcal{X}$	3	0.02	✓	0.29	✓
arch1[65]	2	5	$\mathcal{X}$	4	0.06	✗	0.51	✓
			$\mathcal{I},\mathcal{U},\mathcal{X}$	1	0.01	✓	0.10	✓
arch2[65]	2	2	$\mathcal{X}$	3	0.02	✓	0.11	✓
			$\mathcal{I},\mathcal{U},\mathcal{X}$	1	0.01	✗	0.03	✓
arch3[65]	2	3	$\mathcal{X}$	2	0.01	✓	0.03	✓
			$\mathcal{I},\mathcal{U},\mathcal{X}$	1	0.01	✗	0.03	✓
arch4[65]	2	2	$\mathcal{X}$	3	0.01	✗	0.07	✓
			$\mathcal{U},\mathcal{X}$	2	0.01	✗	0.12	✓
nagumo[63]	2	3	$\mathcal{X}$	2	0.01	✓	0.10	✓
			$\mathcal{U},\mathcal{X}$	4	0.03	✗	0.33	✓
lotka[33]	3	2	$\mathcal{X}$	5	0.22	✗	2.39	?
			$\mathcal{U},\mathcal{X}$	1	0.01	✗	0.06	✓
lorenz[22]	3	2	$\mathcal{X}$	5	0.15	?	1.57	?
			$\mathcal{U},\mathcal{X}$	1	0.02	✗	0.07	✓
lyapunov[60]	3	3	$\mathcal{X}$	5	0.15	✗	2.32	?
			$\mathcal{U},\mathcal{X}$	5	0.32	✗	2.59	?

dim: system dimension; deg(f): maximal flow-field degree; unbounded: the unbounded region(s) for each benchmark instance; deg(B): degree of barrier certificate template, searched from 1; time: SDP solving time; verified: indicates whether the synthesized barrier certificates can be verified by MATHEMATICA. ✓: valid solution. ✗: no solution or invalid solution. ?: beyond the capacity of symbolic methods in MATHEMATICA.

Taming Numerical Issues in SDP. Though SDP-based techniques are widely used in verification and synthesis problems, the result given by SDP solvers can be unreliable due to their inherent numerical issues. In the following, we discuss several techniques to mitigate such numerical problems with respect to our synthesis problem.

Anterior Validation: One robust approach to validate SDP solving is to strengthen the constraints before solving them. As proposed in [61] and extended in [28], this method requires the user to first compute an upper bound ϵ for the numerical errors in the results. Then, the original constraints $C_p \succeq 0$ are replaced by their strengthened versions $C_p \succeq \epsilon I$. If the strengthened constraints are solvable, then a sound solution is obtained. A disadvantage is that the strengthening of constraints may rule out potential solutions.

Intermediate Enhancement: Different representation of polynomials may impact the solution given by SDP solvers. In our formulation, we use the standard monomial basis to represent a polynomial and extract its Gram matrix. Alternatively, we can use other non-trivial polynomial bases such as scaled monomial basis, Bernstein basis, and Chebyshev basis. While using different monomial basis does not alter the number of decision variables, there may be significant differences in the numerical stability during floating-point computation [15, Section 3.1.5].

Posterior Verification: When a numerical solution, say $\hat{B}(\boldsymbol{x})$, is returned by the solver, we can either check its soundness by using exact symbolic methods [19] or prove there exists an actual solution $B(\boldsymbol{x})$ nearby [61]. To check the soundness of $\hat{B}(\boldsymbol{x})$, we can use symbolic solvers, such as Redlog [23] or Z3 [21], to verify that $\hat{B}(\boldsymbol{x})$ satisfies the corresponding barrier certificate conditions. This method is relatively easier to employ. However, when the dimension or degree of $\hat{B}(\boldsymbol{x})$ is too large, even checking the soundness symbolically can be difficult. To prove the existence of a nearby solution, one needs to consider the floating-point arithmetic and verify the sum-of-squares polynomials satisfy the condition given in [61, Prop. 2].

Certainly, we can also resort to SDP solvers with higher precision to reduce numerical errors, such as multiple-precision or arbitrary-precision solvers [42, 51]. However, the unsoundness caused by numerical issues cannot be completely eliminated. Furthermore, while exact SDP solvers [36] relies on symbolic methods and fundamentally avoids numerical problems, currently they can only handle very small problem instances and are not suitable for our synthesis problem.

Beyond SDP. For synthesizing of non-convex barrier certificates or invariant barrier certificates, the homogenization approach and sum-of-squares relaxations are still applicable, but the resulted constraints are no longer SDP due to the arise of cross products of parameters $\boldsymbol{a}$ and unknown coefficients of sum-of-squares polynomials. In general, synthesizing such barrier certificates amounts to solving bilinear matrix inequalities, which is known to be NP-hard. Different schemes have been proposed to solve these constraints [17, 70].

6 EXPERIMENTS

Implementation. We implemented the barrier certificate synthesis procedure in JULIA programming language, interfaced with SUMOFSQUARES package [72] for formulating SOS relaxations and MOSEK solver [9] for solving the underlying SDP. All experiments were performed on a 1.4GHz Intel Core i5 laptop with 8GB of RAM running MacOS. The code and benchmarks are publicly available online[4].

Experiment Settings. The goal of our experiments was to compare the differences between employing the sufficient condition Theorem 7 and the necessary condition Theorem 12 to synthesize barrier certificates over unbounded domains. To this end, we focused on general convex barrier certificates (see Theorem 5) and collected a set of dynamical systems of dimension 2 and 3 from the literature. For each benchmark system, we designed two problem instances. In the first instance, we only let the domain $\mathcal{X} = \mathbb{R}^n$ be unbounded, while in the second instance, we further let the initial set $\mathcal{I}$ and/or the unsafe region $\mathcal{U}$ be unbounded (not necessarily contain the original bounded counterparts). For each problem instance, we searched for barrier certificates from degree 1 and reported the minimum degree such that either Prog. (20) or Prog. (21) is solvable.

In practical computation, we set $\omega(x) = \lambda x$ with $\lambda = -1$. The selection of different value for λ was discussed in [43] and was not a focus of this paper. When the degree $\deg(B)$ was fixed, we solved the sum-of-squares relaxations of Prog. (20) and Prog. (21) respectively with $d_1 = d_2 = \deg(B) + 4$ and $d_3 = \deg(\mathfrak{L}_f B) + 4$, where the parameter 4 was an empirical parameter for obtaining a close solution.

Furthermore, to mitigate numerical errors, we employed three techniques in our algorithms: (1) We used the scaled monomial basis as defined in [15, Sec. 3.1.5] instead of the standard monomial basis. (2) We ignored those coefficients in the numerical solution $\hat{B}(\boldsymbol{x})$ with very small absolute values (less than 10^{-5}). (3) We utilized MATHEMATICA to symbolically verify that the numerical solution $\hat{B}(\boldsymbol{x})$ satisfies the barrier certificate conditions (Note that this step also rules out unsound solutions caused by using the necessary condition). The timeout for the verification procedure was set to be 20 minutes.

Empirical Observations. Table 1 reports the experimental results, we mainly compare the results from two perspectives.

Expressiveness: The necessary condition is much more expressive than the sufficient condition. Specifically, using necessary condition succeeds in synthesizing barrier certificates for all but one 2-dim problem instances, while using the sufficient condition fails in more than half of them. We suspect that the exceptional case (i.e., the first benchmark of barrier) is due to numerical errors, as MATHEMATICA can not find a counter-example violating the barrier certificate conditions when the absolute values of x_1 and x_2 are less than 10^8. For the second instance in lorenz and the second instance in lotka, where barrier certificates of degree 1 exist, the necessary condition can find a valid solution while the sufficient condition can not. As for the rest 3-dim problem instances, both methods fail to synthesis a barrier certificate when $\deg(B) \leq 4$. When $\deg(B) = 5$, the results returned by the sufficient condition can usually be proven wrong easily, which is not the case for the necessary condition. In some sense, this also suggests that the results given by the necessary condition are more likely to be valid.

Efficiency: It is evident that employing the necessary condition leads to an increase in the time overhead for SDP solving across all benchmarks. This observation aligns with our initial expectations, as the homogenization formulation introduces a constant increase in the number of sum-of-squares polynomials and a polynomial growth in the size of Gram matrix. Nevertheless, when the dimensions of systems and the degrees of barrier certificate templates are not large, the constraints can still be solved efficiently within comparable time. The efficiency loss resulting from these factors is currently not a significant bottleneck. For systems of larger dimensions and templates of higher degrees, the time required for posterior verification becomes considerably longer and dominates the SDP solving procedure.

7 CONCLUSION

This paper addresses the problem of synthesizing barrier certificates over unbounded regions. Previous approaches to this problem have primarily relied on a sufficient condition, as the necessary condition based on Putinar's Positivstellensatz is only applicable in bounded cases. Our main contribution lies in the generalization of the necessary condition to unbounded cases, achieved through the utilization of the homogenization approach derived from optimization theory. The resulting constraints are less conservative when compared to those obtained from the sufficient condition. Experimental results substantiate the efficacy of our approach, demonstrating its enhanced expressiveness and ability to synthesize more barrier certificates in comparison to existing methods.

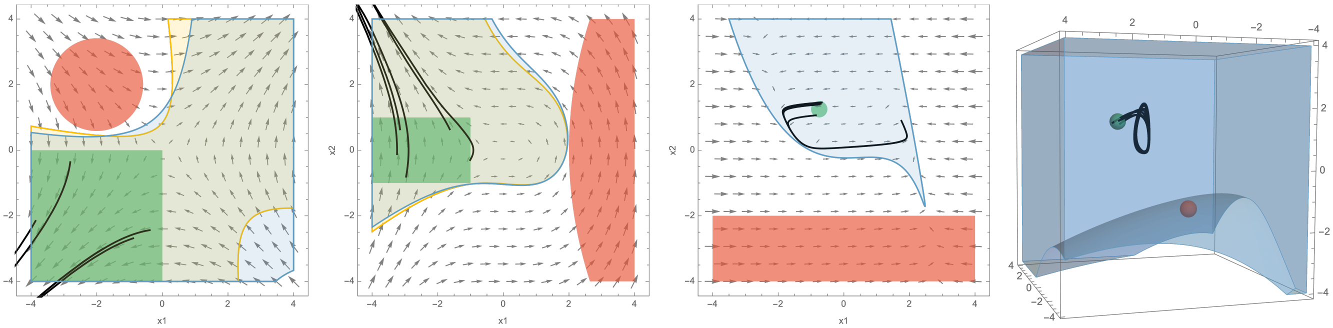
While our paper primarily focuses on synthesizing barrier certificates for differential dynamical systems, it is crucial to note that our method can be readily extended to other types of systems, including hybrid systems and systems with control, disturbance, or stochastic dynamics. Furthermore, our method can also be utilized in related verification problems such as Lyapunov function synthesis or program invariant generation.

REFERENCES

[1] Alessandro Abate, Daniele Ahmed, Alec Edwards, Mirco Giacobbe, and Andrea Peruffo. 2021. FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. In *24th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2021*, Sergiy Bogomolov and Raphaël M. Jungers (Eds.). ACM, 24:1–24:11. https://doi.org/10.1145/3447928.3456646

[2] Matthias Althoff, Goran Frehse, and Antoine Girard. 2021. Set Propagation Techniques for Reachability Analysis. *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), 369–395. https://doi.org/10.1146/ANNUREV-CONTROL-071420-081941

[4]The link is temporarily removed for the review process.



(a) vector-1 (b) lie-der-2 (c) nagumo-2 (d) lotka-1 (unverified)

green region: initial set $\mathcal{I}$; red region: unsafe region $\mathcal{U}$; black solid curves: sampled trajectories $\xi_{\boldsymbol{x}_0}$;
light blue (resp. yellow) region: sub-level set of the synthesized barrier certificate by using the necessary (resp. sufficient) condition.

Figure 1: Portraits of four selected examples.

[3] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. 1995. The Algorithmic Analysis of Hybrid Systems. *Theoretical Computer Science* 138, 1 (1995), 3–34. https://doi.org/10.1016/0304-3975(94)00202-T

[4] Rajeev Alur, Thomas A. Henzinger, Gerardo Lafferriere, and George J. Pappas. 2000. Discrete abstractions of hybrid systems. *Proc. IEEE* 88, 7 (2000), 971–984. https://doi.org/10.1109/5.871304

[5] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. 2017. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Automat. Control* 62, 8 (2017), 3861–3876. https://doi.org/10.1109/TAC.2016.2638961

[6] Hirokazu Anai and Volker Weispfenning. 2001. Reach Set Computations Using Real Quantifier Elimination. In *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001 (Lecture Notes in Computer Science, Vol. 2034)*, Maria Domenica Di Benedetto and Alberto L. Sangiovanni-Vincentelli (Eds.). Springer, 63–76. https://doi.org/10.1007/3-540-45351-2_9

[7] Mahathi Anand, Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. 2021. Safety Verification of Dynamical Systems via k-Inductive Barrier Certificates. In *2021 60th IEEE Conference on Decision and Control CDC 2021*. IEEE, 1314–1320. https://doi.org/10.1109/CDC45484.2021.9682889

[8] Mahathi Anand, Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. 2022. k-Inductive Barrier Certificates for Stochastic Systems. In *Hybrid Systems: Computation and Control, 25th ACM International Conference*, Ezio Bartocci and Sylvie Putot (Eds.). ACM, 12:1–12:11. https://doi.org/10.1145/3501710.3519532

[9] MOSEK ApS. 2019. *MOSEK Optimizer API for Julia. Version 10.1.13.* https://docs.mosek.com/latest/juliaapi/index.html

[10] Eugene Asarin, Thao Dang, and Antoine Girard. 2007. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica* 43, 7 (2007), 451–476. https://doi.org/10.1007/S00236-006-0035-7

[11] Eugene Asarin, Thao Dang, Oded Maler, and Olivier Bournez. 2000. Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems. In *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000 (Lecture Notes in Computer Science, Vol. 1790)*, Nancy A. Lynch and Bruce H. Krogh (Eds.). Springer, 20–31. https://doi.org/10.1007/3-540-46430-1_6

[12] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking.* MIT Press.

[13] Stanley Bak. 2018. *t*-Barrier Certificates: A Continuous Analogy to *k*-Induction. In *6th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2018 (IFAC-PapersOnLine, Vol. 51)*, Alessandro Abate, Antoine Girard, and Maurice Heemels (Eds.). Elsevier, 145–150. https://doi.org/10.1016/J.IFACOL.2018.08.025

[14] Martin Berz and Georg Hoffstätter. 1998. Computation and Application of Taylor Polynomials with Interval Remainder Bounds. *Reliable Computing* 4, 1 (1998), 83–97. https://doi.org/10.1023/A:1009958918582

[15] Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. 2012. *Semidefinite optimization and convex algebraic geometry.* SIAM.

[16] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. 2012. Taylor Model Flowpipe Construction for Non-linear Hybrid Systems. In *Proceedings of the 33rd IEEE Real-Time Systems Symposium, RTSS 2012*. IEEE Computer Society, 183–192. https://doi.org/10.1109/RTSS.2012.70

[17] Xin Chen, Chao Peng, Wang Lin, Zhengfeng Yang, Yifang Zhang, and Xuandong Li. 2020. A Novel Approach for Solving the BMI Problem in Barrier Certificates Generation. In *Computer Aided Verification - 32nd International Conference, CAV 2020 (Lecture Notes in Computer Science, Vol. 12224)*, Shuvendu K. Lahiri and Chao Wang (Eds.). Springer, 582–603. https://doi.org/10.1007/978-3-030-53288-8_29

[18] Liyun Dai, Ting Gan, Bican Xia, and Naijun Zhan. 2017. Barrier certificates revisited. *Journal of Symbolic Computation* (2017), 62–86. https://doi.org/10.1016/j.jsc.2016.07.010

[19] Liyun Dai, Bican Xia, and Naijun Zhan. 2013. Generating Non-linear Interpolants by Semidefinite Programming. In *Computer Aided Verification - 25th International Conference (Lecture Notes in Computer Science, Vol. 8044)*. Springer, 364–380. https://doi.org/10.1007/978-3-642-39799-8_25

[20] James H Davenport and Joos Heintz. 1988. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation* 5, 1-2 (1988), 29–35.

[21] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. 2008. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference (Lecture Notes in Computer Science, Vol. 4963)*. Springer, 337–340. https://doi.org/10.1007/978-3-540-78800-3_24

[22] Adel Djaballah, Alexandre Chapoutot, Michel Kieffer, and Olivier Bouissou. 2017. Construction of parametric barrier functions for dynamical systems using interval analysis. *Automatica* 78 (2017), 287–296. https://doi.org/10.1016/j.automatica.2016.12.013

[23] Andreas Dolzmann and Thomas Sturm. 1997. Redlog: Computer algebra meets computer logic. *Acm Sigsam Bulletin* 31, 2 (1997), 2–9.

[24] Alexandre Donzé and Oded Maler. 2007. Systematic Simulation Using Sensitivity Analysis. In *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007 (Lecture Notes in Computer Science, Vol. 4416)*, Alberto Bemporad, Antonio Bicchi, and Giorgio C. Buttazzo (Eds.). Springer, 174–189. https://doi.org/10.1007/978-3-540-71493-4_16

[25] Parasara Sridhar Duggirala, Sayan Mitra, and Mahesh Viswanathan. 2013. Verification of annotated models from executions. In *Proceedings of the International Conference on Embedded Software, EMSOFT 2013*, Rolf Ernst and Oleg Sokolsky (Eds.). IEEE, 26:1–26:10. https://doi.org/10.1109/EMSOFT.2013.6658604

[26] Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. 2016. Locally optimal reach set over-approximation for nonlinear systems. In *2016 International Conference on Embedded Software, EMSOFT 2016*, Petru Eles and Rahul Mangharam (Eds.). ACM, 6:1–6:10. https://doi.org/10.1145/2968478.2968482

[27] Ting Gan, Mingshuai Chen, Yangjia Li, Bican Xia, and Naijun Zhan. 2018. Reachability Analysis for Solvable Dynamical Systems. *IEEE Trans. Autom. Control.* 63, 7 (2018), 2003–2018. https://doi.org/10.1109/TAC.2017.2763785

[28] Ting Gan, Bican Xia, Bai Xue, Naijun Zhan, and Liyun Dai. 2020. Nonlinear Craig Interpolant Generation. In *Computer Aided Verification - 32nd International Conference (Lecture Notes in Computer Science, Vol. 12224)*. Springer, 415–438. https://doi.org/10.1007/978-3-030-53288-8_20

[29] Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. 2012. δ-Complete Decision Procedures for Satisfiability over the Reals. In *Automated Reasoning - 6th International Joint Conference, IJCAR 2012 (Lecture Notes in Computer Science, Vol. 7364)*, Bernhard Gramlich, Dale Miller, and Uli Sattler (Eds.). Springer, 286–300. https://doi.org/10.1007/978-3-642-31365-3_23

[30] Sicun Gao, Soonho Kong, and Edmund M. Clarke. 2013. dReal: An SMT Solver for Nonlinear Theories over the Reals. In *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction (Lecture Notes in Computer Science, Vol. 7898)*, Maria Paola Bonacina (Ed.). Springer, 208–214. https://doi.org/10.1007/978-3-642-38574-2_14

[31] Antoine Girard. 2005. Reachability of Uncertain Linear Systems Using Zonotopes. In *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005 (Lecture Notes in Computer Science, Vol. 3414)*, Manfred Morari and Lothar Thiele (Eds.). Springer, 291–305. https://doi.org/10.1007/978-3-540-31954-2_19

[32] Antoine Girard and George J. Pappas. 2006. Verification Using Simulation. In *Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006 (Lecture Notes in Computer Science, Vol. 3927)*, João P. Hespanha and Ashish Tiwari (Eds.). Springer, 272–286. https://doi.org/10.1007/11730637_22

[33] Eric Goubault, Jacques-Henri Jourdan, Sylvie Putot, and Sriram Sankaranarayanan. 2014. Finding non-polynomial positive invariants and lyapunov functions for polynomial systems through Darboux polynomials. In *American Control Conference, ACC 2014*. IEEE, 3571–3578. https://doi.org/10.1109/ACC.2014.6859330

[34] Colas Le Guernic and Antoine Girard. 2009. Reachability Analysis of Hybrid Systems Using Support Functions. In *Computer Aided Verification, 21st International Conference, CAV 2009 (Lecture Notes in Computer Science, Vol. 5643)*, Ahmed Bouajjani and Oded Maler (Eds.). Springer, 540–554. https://doi.org/10.1007/978-3-642-02658-4_40

[35] Feng Guo, Li Wang, and Guangming Zhou. 2014. Minimizing rational functions by exact Jacobian SDP relaxation applicable to finite singularities. *Journal of Global Optimization* 58, 2 (2014), 261–284. https://doi.org/10.1007/s10898-013-0047-0

[36] Didier Henrion, Simone Naldi, and Mohab Safey El Din. 2021. Exact algorithms for semidefinite programs with degenerate feasible set. *Journal of Symbolic Computation* 104 (2021), 942–959. https://doi.org/10.1016/j.jsc.2020.11.001

[37] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. 1995. What's decidable about hybrid automata?. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, STOC 1995*, Frank Thomson Leighton and Allan Borodin (Eds.). ACM, 373–382. https://doi.org/10.1145/225058.225162

[38] Charles Antony Richard Hoare. 1969. An axiomatic basis for computer programming. *Commun. ACM* 12, 10 (1969), 576–580.

[39] Chao Huang, Xin Chen, Wang Lin, Zhengfeng Yang, and Xuandong Li. 2017. Probabilistic Safety Verification of Stochastic Hybrid Systems Using Barrier Certificates. *ACM Transactions on Embedded Computing Systems* 16, 5s (2017), 186:1–186:19. https://doi.org/10.1145/3126508

[40] Lei Huang, Jiawang Nie, and Ya-Xiang Yuan. 2023. Homogenization for polynomial optimization with unbounded sets. *Mathematical Programming* 200, 1 (2023), 105–145. https://doi.org/10.1007/s10107-022-01878-5

[41] Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. 2021. Formal Synthesis of Stochastic Systems via Control Barrier Certificates. *IEEE Trans. Automat. Control* 66, 7 (2021), 3097–3110. https://doi.org/10.1109/TAC.2020.3013916

[42] Mioara Joldes, Jean-Michel Muller, and Valentina Popescu. 2017. Implementation and Performance Evaluation of an Extended Precision Floating-Point Arithmetic Library for High-Accuracy Semidefinite Programming. In *24th IEEE Symposium on Computer Arithmetic*. IEEE Computer Society, 27–34. https://doi.org/10.1109/ARITH.2017.18

[43] Hui Kong, Fei He, Xiaoyu Song, William N. N. Hung, and Ming Gu. 2013. Exponential-Condition-Based Barrier Certificate Generation for Safety Verification of Hybrid Systems. In *Computer Aided Verification, 25th International Conference (Lecture Notes in Computer Science, Vol. 8044)*, Natasha Sharygina and Helmut Veith (Eds.). Springer, 242–257. https://doi.org/10.1007/978-3-642-39799-8_17

[44] Alexander B. Kurzhanski and Pravin Varaiya. 2000. Ellipsoidal Techniques for Reachability Analysis. In *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000 (Lecture Notes in Computer Science, Vol. 1790)*, Nancy A. Lynch and Bruce H. Krogh (Eds.). Springer, 202–214. https://doi.org/10.1007/3-540-46430-1_19

[45] Gerardo Lafferriere, George J. Pappas, and Sergio Yovine. 2001. Symbolic Reachability Computation for Families of Linear Vector Fields. *Journal of Symbolic Computation* 32, 3 (2001), 231–253. https://doi.org/10.1006/JSCO.2001.0472

[46] Jean Bernard Lasserre. 2009. *Moments, positive polynomials and their applications*. Vol. 1. World Scientific.

[47] Jiang Liu, Naijun Zhan, and Hengjun Zhao. 2011. Computing semi-algebraic invariants for polynomial dynamical systems. In *Proceedings of the 11th International Conference on Embedded Software, EMSOFT 2011*, Samarjit Chakraborty, Ahmed Jerraya, Sanjoy K. Baruah, and Sebastian Fischmeister (Eds.). ACM, 97–106. https://doi.org/10.1145/2038642.2038659

[48] Victor Magron and Jie Wang. 2023. *Sparse polynomial optimization: theory and practice*. World Scientific.

[49] Ngoc Hoang Anh Mai, Jean-Bernard Lasserre, and Victor Magron. 2022. Positivity certificates and polynomial optimization on non-compact semialgebraic sets. *Mathematical Programming* 194, 1 (2022), 443–485. https://doi.org/10.1007/s10107-021-01634-1

[50] Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud. 2009. *Introduction to Interval Analysis*. SIAM. https://doi.org/10.1137/1.9780898717716

[51] Maho Nakata. 2010. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP,-QD and-DD.. In *2010 IEEE International Symposium on Computer-Aided Control System Design*. IEEE, 29–34.

[52] Jiawang Nie. 2012. Discriminants and nonnegative polynomials. *Journal of Symbolic Computation* 47, 2 (2012), 167–191. https://doi.org/10.1016/j.jsc.2011.08.023

[53] Pablo A Parrilo. 2000. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology.

[54] Andrea Peruffo, Daniele Ahmed, and Alessandro Abate. 2021. Automated and Formal Synthesis of Neural Barrier Certificates for Dynamical Models. In *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021 (Lecture Notes in Computer Science, Vol. 12651)*, Jan Friso Groote and Kim Guldstrand Larsen (Eds.). Springer, 370–388. https://doi.org/10.1007/978-3-030-72016-2_20

[55] André Platzer and Edmund M. Clarke. 2008. Computing Differential Invariants of Hybrid Systems as Fixedpoints. In *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, NJ, USA, July 7-14, 2008, Proceedings (Lecture Notes in Computer Science, Vol. 5123)*, Aarti Gupta and Sharad Malik (Eds.). Springer, 176–189. https://doi.org/10.1007/978-3-540-70545-1_17

[56] Stephen Prajna and Ali Jadbabaie. 2004. Safety Verification of Hybrid Systems Using Barrier Certificates. In *Hybrid Systems: Computation and Control, 7th International Workshop (Lecture Notes in Computer Science, Vol. 2993)*, Rajeev Alur and George J. Pappas (Eds.). Springer, 477–492. https://doi.org/10.1007/978-3-540-24743-2_32

[57] Stephen Prajna, Ali Jadbabaie, and George J. Pappas. 2007. A Framework for Worst-Case and Stochastic Safety Verification Using Barrier Certificates. *IEEE Trans. Automat. Control* 52, 8 (2007), 1415–1428. https://doi.org/10.1109/TAC.2007.902736

[58] Mihai Putinar. 1993. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal* 42, 3 (1993), 969–984.

[59] Mihai Putinar and Florian-Horia Vasilescu. 1999. Solving moment problems by dimensional extension. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* 328, 6 (1999), 495–499.

[60] Stefan Ratschan and Zhikun She. 2010. Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-Like Functions. *SIAM Journal on Control and Optimization* 48, 7 (2010), 4377–4394. https://doi.org/10.1137/090749955

[61] Pierre Roux, Yuen-Lam Voronin, and Sriram Sankaranarayanan. 2018. Validating numerical semidefinite programming solvers for polynomial invariants. *Formal Methods in System Design* 53, 2 (2018), 286–312. https://doi.org/10.1007/s10703-017-0302-y

[62] Ali Salamati and Majid Zamani. 2022. Data-Driven Safety Verification of Stochastic Systems via Barrier Certificates: A Wait-and-Judge Approach. In *Learning for Dynamics and Control Conference, L4DC 2022 (Proceedings of Machine Learning Research, Vol. 168)*, Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel J. Kochenderfer (Eds.). PMLR, 441–452. https://proceedings.mlr.press/v168/salamati22a.html

[63] Mohamed Amin Ben Sassi, Antoine Girard, and Sriram Sankaranarayanan. 2014. Iterative computation of polyhedral invariants sets for polynomial dynamical systems. In *53rd IEEE Conference on Decision and Control, CDC 2014*. IEEE, 6348–6353. https://doi.org/10.1109/CDC.2014.7040384

[64] Mohamed Amin Ben Sassi, Sriram Sankaranarayanan, Xin Chen, and Erika Ábrahám. 2016. Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis. *IMA Journal of Mathematical Control and Information* 33, 3 (2016), 723–756. https://doi.org/10.1093/imamci/dnv003

[65] Andrew Sogokon, Khalil Ghorbal, and Taylor T. Johnson. 2016. Non-linear Continuous Systems for Safety Verification. In *ARCH@CPSWeek 2016, 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems (EPiC Series in Computing, Vol. 43)*, Goran Frehse and Matthias Althoff (Eds.). EasyChair, 42–51. https://doi.org/10.29007/W94N

[66] Andrew Sogokon, Khalil Ghorbal, Yong Kiam Tan, and André Platzer. 2018. Vector Barrier Certificates and Comparison Systems. In *Formal Methods, 22nd International Symposium (Lecture Notes in Computer Science, Vol. 10951)*, Klaus Havelund, Jan Peleska, Bill Roscoe, and Erik P. de Vink (Eds.). Springer, 418–437. https://doi.org/10.1007/978-3-319-95582-7_25

[67] Gilbert Stengle. 1974. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Math. Ann.* 207 (1974), 87–97.

[68] Paulo Tabuada. 2009. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer. http://www.springer.com/mathematics/applications/book/978-1-4419-0223-8

[69] Alfred Tarski. 1951. *A decision method for elementary algebra and geometry*. University of California Press, Berkeley.

[70] Qiuye Wang, Mingshuai Chen, Bai Xue, Naijun Zhan, and Joost-Pieter Katoen. 2022. Encoding inductive invariants as barrier certificates: Synthesis via difference-of-convex programming. *Information and Computation* 289, Part (2022), 104965. https://doi.org/10.1016/j.ic.2022.104965

[71] Qiuye Wang, Yangjia Li, Bican Xia, and Naijun Zhan. 2017. Generating semi-algebraic invariants for non-autonomous polynomial hybrid systems. *Journal of Systems Science and Complexity* 30, 1 (2017), 234–252. https://doi.org/10.1007/S11424-017-6226-1

[72] Tillmann Weisser, Benoıt Legat, Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. 2019. Polynomial and Moment Optimization in Julia and JuMP. In *JuliaCon*. https://pretalx.com/juliacon2019/talk/QZBKAU/

[73] Xiangru Xu, Paulo Tabuada, Jessy W. Grizzle, and Aaron D. Ames. 2015. Robustness of Control Barrier Functions for Safety Critical Control. In *5th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2015 (IFAC-PapersOnLine, Vol. 48)*, Magnus Egerstedt and Yorai Wardi (Eds.). Elsevier, 54–61. https://doi.org/10.1016/J.IFACOL.2015.11.152

[74] Xia Zeng, Wang Lin, Zhengfeng Yang, Xin Chen, and Lilei Wang. 2016. Darboux-type barrier certificates for safety verification of nonlinear hybrid systems. In *International Conference on Embedded Software, 2016*, Petru Eles and Rahul Mangharam (Eds.). ACM, 11:1–11:10. https://doi.org/10.1145/2968478.2968484

[75] Hanrui Zhao, Niuniu Qi, Lydia Dehbi, Xia Zeng, and Zhengfeng Yang. 2023. Formal Synthesis of Neural Barrier Certificates for Continuous Systems via Counterexample Guided Learning. *ACM Transactions on Embedded Computing Systems* 22, 5s (2023), 146:1–146:21. https://doi.org/10.1145/3609125

[76] Hengjun Zhao, Xia Zeng, Taolue Chen, and Zhiming Liu. 2020. Synthesizing barrier certificates using neural networks. In *HSCC '20: 23rd ACM International Conference on Hybrid Systems: Computation and Control, Sydney, New South Wales, Australia, April 21-24, 2020*, Aaron D. Ames, Sanjit A. Seshia, and Jyotirmoy Deshmukh (Eds.). ACM, 25:1–25:11. https://doi.org/10.1145/3365365.3382222