Numerically-Robust Inductive Proof Rules for Continuous Dynamical Systems

Sicun Gao[1], James Kapinski[2], Jyotirmoy Deshmukh[3], Nima Roohi[1], Armando Solar-Lezama[4], Nikos Arechiga[5], and Soonho Kong[5]

[1] University of California, San Diego
{sicung,nroohi}@ucsd.edu
[2] Toyota R&D
jim.kapinski@toyota.com
[3] University of Southern California
jyotirmoy.deshmukh@usc.edu
[4] Massachusetts Institute of Technology
asolar@csail.mit.edu
[5] Toyota Research Institute
{nikos.arechiga,soonho.kong}@tri.global

Abstract. We formulate numerically-robust inductive proof rules for unbounded stability and safety properties of continuous dynamical systems. These induction rules robustify standard notions of Lyapunov functions and barrier certificates so that they can tolerate small numerical errors. In this way, numerically-driven decision procedures can establish a sound and relative-complete proof system for unbounded properties of very general nonlinear systems. We demonstrate the effectiveness of the proposed rules for rigorously verifying unbounded properties of various nonlinear systems, including a challenging powertrain control model.

1 Introduction

Infinite-time stability and safety properties of continuous dynamical systems are typically established via inductive arguments over continuous time. For instance, proving stability of a dynamical system is similar to proving termination of a program. A system is stable at the origin in the sense of Lyapunov, if one can find a Lyapunov function (essentially a ranking function) that is everywhere positive except for reaching exactly zero at the origin, and never increases over time along the direction of the system dynamics [11]. Likewise, proving unbounded safety of a dynamical system requires one to find a barrier function (or differential invariant [19]) that separates the system's initial state from the unsafe regions, and whenever the system states reach the barrier, the system dynamics always points towards the safe side of the barrier [21]. In both cases, once a candidate certificate (Lyapunov or barrier functions) is proposed, the verification problem is reduced to checking the validity of a universally-quantified first-order formula over real-valued variables. The standard approaches for the validation step use symbolic quantifier elimination [4] or Sum-of-Squares techniques [18,17,24]. However, these algorithms are either extremely expensive or

numerically brittle. Most importantly, they can not handle systems with non-polynomial nonlinearity, and thus fall short of a general framework for verifying practical systems of significant complexity.

The standard approach of checking invariance conditions in program analysis is to use Satisfiability Modulo Theories (SMT) solvers [16]. However, to check the inductive conditions for nonlinear dynamical systems, one has to solve nonlinear SMT problems over real numbers, which are highly intractable or undecidable [23]. Recent work on numerically-driven decision procedures provides a promising direction to bypass this difficulty [5,6]. They have been used for many bounded-time verification and synthesis problems for highly nonlinear systems [12]. However, the fundamental challenge with using numerically-driven methods in inductive proofs is that numerical errors make it impossible to verify the induction steps in the standard sense. Take the Lyapunov analysis of stability properties as an example. A dynamical system is stable if there exists a function that vanishes *exactly* at the origin and its derivatives *strictly* decreases over time. Since *any* numerical error blurs the difference between strict and non-strict inequality, one can conclude that numerically-driven methods are not suitable for verifying these strict constraints. However, proving a system is stable within an arbitrarily tiny neighborhood around the origin is all we really need in practice. Thus, there is a discrepancy between what the standard theory requires and what is needed in practice, or what can be achieved computationally. To bridge this gap, we need to rethink about the fundamental definitions.

In this paper, we formulate new inductive proof rules for continuous dynamical systems for establishing robust notions of stability and safety. These proof rules are practically useful and computationally certifiable in a very general sense. For instance, for stability, we define the notion of ε-stability that requires the system to be stable within an ε-bounded distance from the origin, instead of exactly at the origin. When ε is small enough, ε-stable systems are practically indistinguishable from stable systems. We then define the notion of ε-Lyapunov functions that are sufficient for establishing ε-stability. We then rigorously prove that the ε-Lyapunov conditions are numerically stable and can be correctly determined by δ-complete decisions procedures for nonlinear real arithmetic [7]. In this way, we can rely on various numerically-driven SMT solvers to establish a sound and relative-complete proof systems for unbounded stability and safety properties of highly nonlinear dynamical systems. We believe these new definitions have eliminated the core difficulty for reasoning about infinite-time properties of nonlinear systems, and will pave the way for adapting a wide range of automated methods from program analysis to continuous and hybrid systems. In short, the paper makes the following contributions:

- We define ε-stability and ε-Lyapunov functions in Section 3. We prove that finding ε-Lyapunov functions is sufficient for establishing ε-stability.
- We define two types of robust proof rules for unbounded safety in Section 3, which we call Type 1 and Type 2 ε-barrier functions. The former relies on strict contraction, and the latter relies on reachable-set computation to guarantee bounded escape.

– We prove that δ-complete decision procedures provide a sound and relative-complete proof system for the proposed numerically-robust induction rules, in both Sections 3 and 4.

We demonstrate the effectiveness of the proposed methods on various nonlinear systems in Section 5. Section 2 covers the basic definitions and Section 6 concludes the paper.

Related Work. Several lines of work have proposed relaxed and practical notions to capture the spirit of the stability requirements. Early work from the 1960s introduced practical stability, which defined bounds on system behaviors over finite time horizons [14,26,27,2]. These methods can show whether a system leaves a safe set or enters a goal set over a finite time horizon based on Lyapunov-like functions. Stability defined in this sense is equivalent to estimating the reachable set over a finite time horizon. Thus, the shortcoming is that it may not capture the desired behavior of the system over unbounded time. Similarly, notions of boundedness and ultimate boundedness specify limits on the system behaviors [11]. Boundedness specifies whether the system remains within a given bounded region. Ultimate boundedness specifies that the system eventually returns to the given bounded region. These properties can be established based on Lyapunov-like conditions. Related notions have been generalized to switched systems [29,30]. Also, the related notion of region stability defines systems that eventually enter and remain within a specified set [20]. We present stability concepts that unify and extend the above notions. A related relaxation of the traditional notions of stability includes *almost* Lyapunov functions [15], which allow the strict stability conditions to be neglected in a region near the equilibrium point. The challenge of applying this technique in practice is that the size and shape of the neglected region are not specified a priori, so a constructive technique for specifying a stability region is not straightforward. Our work is related to efforts to construct and check robust barrier certificates using Lyapunov-like functions to ensure that controllers satisfy safety constraints [28]. This work provides a framework in which to specify analytic constraints on controller behaviors. By contrast, our work focuses on providing constraints that can be checked fully automatically. Our notion of ε-barrier functions is closely related to t-barrier certificates from [1], though we choose to focus on distance bounds from the barrier (ε) rather than time bounds that indicate how long it takes for behaviors to re-enter the barrier once it has left (t).

2 Background

2.1 Dynamical Systems

Throughout the paper, we use the following definition of an n-dimensional autonomous dynamical system:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t)),\ x(0) \in \mathsf{init} \text{ and } \forall t \in \mathbb{R}_{\geq 0},\ x(t) \in D, \tag{1}$$

where an open set $D \subseteq \mathbb{R}^n$ is the state space, $\mathsf{init} \subseteq D$ is a set of initial states, and $f : D \to \mathbb{R}^n$ is a vector field specified by Lipschitz-continuous functions on each dimension. For notational simplicity, *all variable and function symbols can represent vectors*. When vectors are used in logic formulas, they represent conjunctions of the formulas for each dimension. For instance, when $x = (x_1, \ldots, x_n)$, we write $x = 0$ to denote the formula $x_1 = 0 \wedge \cdots \wedge x_n = 0$. For any system defined by (1), we write its solution function as

$$F : D \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n,\ F(x(0), t) = x(0) + \int_0^t f(x(s))\mathrm{d}s. \tag{2}$$

Note that F usually does not have an analytic form. However, since f is Lipschitz-continuous, F exists and is unique. We will often use Lie derivatives to measure the change of a scalar function along the flow defined by another vector field:

Definition 1 (Lie Derivative). *Let $f : D \to \mathbb{R}^n$ define a vector field. Write the i^{th} component of f as f_i. Let $V : D \to \mathbb{R}$ be a differentiable scalar function. The Lie derivative of V over f is defined as $\nabla_f V(x) = \sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i$.*

2.2 First-order Language over the Reals $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$

We will make extensive use of first-order formulas over real numbers with Type 2 computable functions [25] to express and infer properties of nonlinear dynamical systems. Definition 2 introduces the syntax of these formulas.

Definition 2 (Syntax of $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$). *Let $\mathcal{F}$ be the class of all Type 2 computable functions over real numbers. We define:*

$$t ::= x_i \mid f(t(x)),\ \textit{where } f \in \mathcal{F}, \textit{ possibly constant};$$
$$\varphi ::= \top \mid \bot \mid t(x) > 0 \mid t(x) \geq 0 \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x_i \varphi \mid \forall x_i \varphi.$$

We regard $\neg\varphi$ as an operation that is defined inductively as usual. For instance, $\neg(t > 0)$ is defined as $-t \geq 0$, and $\neg(\exists x_i \varphi)$ is defined as $\forall x_i \neg\varphi$. For any $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$ terms u and v, variable x, and $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$ predicate φ, we write $\exists^{[u,v]} x\varphi$ and $\forall^{[u,v]} x\varphi$ to denote $\exists x(u \leq x \wedge x \leq v \wedge \varphi)$ and $\forall x((u \leq x \wedge x \leq v) \to \varphi)$, respectively, which applies to open intervals too. Next, Definition 3 introduces syntactic perturbation of formulas in $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$.

Definition 3 (δ-Strengthening and Robust Formulas [7]). *Let $\delta \in \mathbb{Q}^+$ be arbitrary. Let φ be an arbitrary $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$ formula. The δ-strengthening of φ, denoted by $\varphi^{+\delta}$, is obtained from φ by replacing every atomic predicate of the form $t(x) > 0$ and $t(x) \geq 0$ with $t(x) - \delta > 0$ and $t(x) - \delta \geq 0$, respectively. We say φ is* δ-robust *iff $\varphi^{+\delta} \leftrightarrow \varphi$.*

Definition 4 (δ-Complete Decision Procedures [7]). *Let S be a class of $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-sentences. We say a decision procedure is δ-complete over S iff for any $\varphi \in S$, the procedure correctly returns one of the following answers:*

- true : *φ is true.*

– δ-false : $\varphi^{+\delta}$ *is false.*
When the two cases overlap, either decision can be returned.

It follows that if φ is δ-robust, then a δ-complete decision procedure can correctly determine the truth value of φ.

3 Robust Proofs for Stability

We first focus on stability. We will define the notion of ε-stability, as a relaxation of the standard Lyapunov stability, and then define ε-Lyapunov functions, which are sufficient for proving ε-stability in a robust way.

3.1 Stability and Lyapunov Functions

Conventionally, ε and δ are used to best highlight the connection with ε-δ conditions for continuity. We will mostly reserve the use of ε for defining conditions that are robust under ε-bounded numerical errors. Thus, we replace ε by τ in the standard definitions to avoid confusion.

Definition 5 (Stability). *We say the system in (1) is stable at the origin in the sense of Lyapunov, iff for any τ-ball neighborhood of the origin, there exists a δ-ball around the origin, such that, if the system starts within the δ-ball then it never escapes the τ-ball. We capture the definition by the following $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-formula:*

$$\mathsf{Stable}(f) \equiv_{df} \forall^{(0,\infty)}\tau \exists^{(0,\infty)}\delta \forall^{D} x_0 \forall^{[0,\infty)} t \Big(\|x_0\| < \delta \rightarrow \|F(x_0,t)\| < \tau \Big)$$

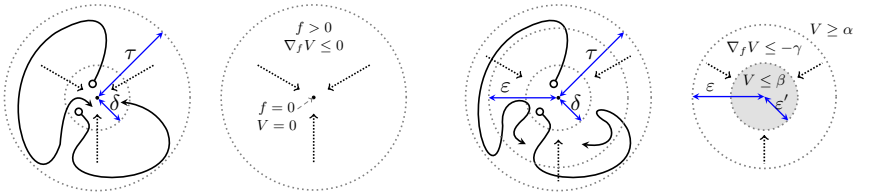
Definition 6 (Lyapunov Function). *Consider a dynamical system given in the form of (1), and let $V : D \to \mathbb{R}$ be a differentiable function. We say V is a non-strict Lyapunov function for the system, iff the following predicate is true:*

$$\mathsf{LF}(f,V) \equiv_{df} \ (V(0) = 0) \wedge (f(0) = 0) \wedge \forall^{D\setminus\{0\}} x \ \Big(V(x) > 0 \wedge \nabla_f V(x) \leq 0 \Big)$$

Proposition 1. *For any dynamical system defined by f, if there exists a Lyapunov function V, then the system is stable. Namely, $\mathsf{LF}(f,V) \to \mathsf{Stable}(f)$.*

3.2 Epsilon-Stability

The standard definitions of stability requires a system to stabilize within arbitrarily small neighborhoods around the origin. However, very small neighborhoods are practically indistinguishable from the origin. Thus, it is practically sufficient to prove that a system is stable within some sufficiently small neighborhood. We capture this intuition by making a minor change to the standard definition, by simply putting a lower bound ε on the τ parameter in Definition 5. As a result, the system is required to exhibit the same behavior as standard stable systems outside the ε-ball, but can behave arbitrarily within the ε-ball (for instance, oscillate around the origin). The formal definition is as follows:



(a) Stability (b) Lyapunov Function (c) ε-Stability (d) ε-Lyapunov Function

Fig. 1: Standard and ε-relaxed notions of stability and Lyapunov functions

Definition 7 (Epsilon-Stability). *Let $\varepsilon \in \mathbb{R}_+$ be arbitrary. We say a dynamical system in (1) is ε-stable at the origin in the sense of Lyapunov, iff it satisfies the following condition:*

$$\mathsf{Stable}_\varepsilon(f) \equiv_{df} \forall^{[\varepsilon,\infty)}\tau \exists^{(0,\infty)}\delta \forall^{D} x_0 \forall^{[0,\infty)} t \Big(\|x_0\| < \delta \rightarrow \|F(x_0,t)\| < \tau \Big)$$

In words, for any $\tau \geq \varepsilon$, there exists δ such that all trajectories that start within the δ-ball will stay within a τ-ball around the origin.

Note that the only difference with the standard definition is that τ is *bounded from below* by a positive ε instead of 0. The definition is depicted in Figure 1c, which shows the difference with the standard notion in Figure 1a. Since the only difference with the standard definition is the lower bound on the universally quantified τ, it is clear that ε-stability is strictly weaker than standard stability.

Proposition 2. *For any $\varepsilon \in \mathbb{R}_+$, $\mathsf{Stable}(f) \rightarrow \mathsf{Stable}_\varepsilon(f)$.*

Thus, any system that is stable in the standard definition is also ε-stable for any $\varepsilon \in \mathbb{R}_+$. On the other hand, one can always choose small enough ε such that an ε-stable system is practically indistinguishable from stable systems in the standard definition.

3.3 Epsilon-Lyapunov Function

We now define the corresponding notion of Lyapunov function that can be used for proving ε-stability. The robustness problem in the standard definition comes from the singularity of the origin. With the relaxed notion of stability, the system may oscillate within some ε-neighborhood of the origin. With the relaxation, we now have room for constructing a few nested neighborhoods that can trap the trajectories in a way that is robust under sufficiently small perturbations. To achieve this, we make use of balls of different sizes, as shown in the following definition. We write $\mathcal{B}_\varepsilon$ to denote open ε-balls around the origin.

Definition 8 (Epsilon-Lyapunov Functions). *Let $V : D \rightarrow \mathbb{R}$ be a differentiable scalar function defined for the system in (1), and let $\varepsilon \in \mathbb{R}_+$ be an arbitrary value. We say V is an ε-Lyapunov function for the system, iff it satisfies the following conditions:*

1. *Outside the ε-ball, there is some positive lower bound on the value of V. Namely, there exists $\alpha \in \mathbb{R}_+$ such that for any $x \in D \setminus \mathcal{B}_\varepsilon$, $V(x) \geq \alpha$.*
2. *Inside the ε-ball, there is a strictly smaller ε'-ball in which the value of V is bounded from above, to create a gap with its values outside the ε-ball. Formally, there exists $\varepsilon' \in (0, \varepsilon)$ and $\beta \in (0, \alpha)$ such that for all $x \in \mathcal{B}_{\varepsilon'}$, $V(x) \leq \beta$.*
3. *The Lie derivative of V is strictly negative outside of $\mathcal{B}_{\varepsilon'}$. Formally, there exists $\gamma \in \mathbb{R}_+$ such that for all $x \in D \setminus \mathcal{B}_{\varepsilon'}$, the Lie derivative of V along f satisfies $\nabla_f V(x) \leq -\gamma$.*

In sum, the three conditions can be expressed with the following $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-formula:

$$\begin{aligned}\mathsf{LF}_\varepsilon(f, V) \equiv_{df}\ & \exists^{(0,\varepsilon)}\varepsilon' \exists^{(0,\infty)}\alpha \exists^{(0,\alpha)}\beta \exists^{(0,\infty)}\gamma \\ & \forall^{D\setminus\mathcal{B}_\varepsilon} x\Big(V(x) \geq \alpha\Big) \wedge \forall^{\mathcal{B}_{\varepsilon'}} x\Big(V(x) \leq \beta\Big) \\ & \wedge \forall^{D\setminus\mathcal{B}_{\varepsilon'}} x\Big(\nabla_f V(x) \leq -\gamma\Big)\end{aligned}$$

It is important to note that ε', α, β, and γ, are not fixed constants, but existentially quantified variables. Thus the condition can hold true for infinitely many values of these parameters, which is critical to robustness. The only free variable in the formula is ε, used in $\mathcal{B}_\varepsilon$ and the bound for ε'. Note also that neither of $\mathsf{LF}_\varepsilon(f, V)$ and the standard definition $\mathsf{LF}(f, V)$ implies the other.

Remark 1. The logical structure of $\mathsf{LF}_\varepsilon(f, V)$ is seemingly more complex than the standard Lyapunov conditions in Definition 6 because of the extra existential quantification. In Theorem 3, we show that it does not add computational complexity in checking the conditions.

The key result is that the conditions for an ε-Lyapunov function are sufficient for establishing ε-stability.

Theorem 1. *If there exists an ε-Lyapunov function V for a dynamical system defined by f, then the system is ε-stable. Namely, $\mathsf{LF}_\varepsilon(f, V) \to \mathsf{Stable}_\varepsilon(f)$.*

Proof. Let $\tau \geq \varepsilon$ be arbitrary, and let $\alpha, \gamma \in \mathbb{R}_+$, $\beta \in (0, \alpha)$, and $\varepsilon' \in (0, \varepsilon)$ be as specified by the definition of $\mathsf{LF}_\varepsilon(f, V)$. Let $x_0 \in \mathcal{B}_{\varepsilon'}$ be an arbitrary point. For any $t \in \mathbb{R}_{\geq 0}$, let $x(t) \coloneqq F(x_0, t)$ be the system state as defined in (2). We use contradiction to prove for any $t \in \mathbb{R}_+$, inequality $\|x(t)\| < \varepsilon \leq \tau$ holds. Since $\varepsilon' < \varepsilon$ and $F(x_0, .)$ is continuous, we know t_1 and t_2 with the following conditions exists ($\partial\mathcal{B}_{\varepsilon'}$ and $\partial\mathcal{B}_\varepsilon$ are boundaries of the corresponding balls):

$$0 \leq t_1 < t_2 \leq t, \quad x(t_1) \in \partial\mathcal{B}_{\varepsilon'}, \quad x(t_2) \in \partial\mathcal{B}_\varepsilon, \quad \forall^{(t_1,t_2)} t'\Big(x(t') \in \mathcal{B}_\varepsilon \setminus \mathcal{B}_{\varepsilon'}\Big)$$

We know $V(x(t_1)) \leq \beta < \alpha \leq V(x(t_2))$ and hence $V(x(t_1)) < V(x(t_2))$ are both true; however, this is in contradiction with the mean value theorem and the fact that $\mathcal{B}_\varepsilon \subset D$ and $\forall^{D\setminus\mathcal{B}_{\varepsilon'}} x\big(\nabla_f V(x) < -\gamma\big)$. □

Remark 2. Proof of Theorem 1 shows that once state of the system enters $\mathcal{B}_{\varepsilon'}$, it never leaves $\mathcal{B}_{\varepsilon}$. However, it would be still possible for the state to leave $\mathcal{B}_{\varepsilon'}$. One the other hand, since closure of $\mathcal{B}_{\varepsilon} \setminus \mathcal{B}_{\varepsilon'}$ is bounded, and for every x in this area, V is continuous at x and $\nabla_f V(x) \leq -\gamma$, no trajectory can be trapped in the closure of $\mathcal{B}_{\varepsilon} \setminus \mathcal{B}_{\varepsilon'}$. Therefore, even though state of the system might leave $\mathcal{B}_{\varepsilon'}$, it will visit inside of this ball infinitely often.

Example 1. Consider the time-reversed Van der Pol system given by the following dynamics. Figure 3 shows the vector field of this system around the origin.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 \\ (x_1^2 - 1)x_2 + x_1 \end{bmatrix}$$

A Lyapunov function $z^T P z$, where z^T is $[x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3]$, and P is the 9×9 constant matrix given in [8], is a 6-degree polynomial that can be obtained using simulation-guided techniques from [10]. Using dReal [9] with $\delta := 10^{-25}$ and the Euclidean norm, we are able to prove that $z^T P z$ is a 10^{-12}-Lyapunov function. Table 1 lists the parameters used for this proof.

3.4 Automated Proofs with Delta-Decisions

We now prove that unlike the conventional conditions, the new inductive proof rules are numerically robust. It follows that δ-decision procedures provide a sound and relative-complete proof system for establishing the conditions in the following sense:

- (Soundness) A δ-complete decision procedure is always correct when it confirms the existence of an ε-Lyapunov function.
- (Relative Completeness) For a given ε-inductive certificate, there exists $\delta > 0$ such that a δ'-complete procedure is able to verify it, for any $0 < \delta' \leq \delta$.

To prove these properties, the key fact is that the continuity of the functions in the induction conditions ensures that there is room for numerical errors in the conditions. Consequently, the formulas allow δ-perturbations in their parameters. This is captured by Lemma 1, and the proof is given in [8].

Lemma 1. *For any $\varepsilon \in \mathbb{R}_+$, there exists $\delta \in \mathbb{Q}_+$ such that $\mathsf{LF}_\varepsilon(f, V)$ is δ-robust.*

Note that if a formula ϕ is δ-robust then for every $\delta' \in (0, \delta)$, ϕ is δ'-robust as well. The soundness and relative-completeness then follow naturally.

Theorem 2 (Soundness). *If a δ-complete decision procedure confirms that $\mathsf{LF}_\varepsilon(f, V)$ is true then V is indeed an ε-Lyapunov function, and f is ε-stable.*

Proof. Using Definition 4, we know $\mathsf{LF}_\varepsilon(f, V)$, exactly as specified in Definition 8, is true. Therefore, V is ε-Lyapunov. Using Theorem 1, f is ε-stable. □

Theorem 3 (Relative Completeness). *For any $\varepsilon \in \mathbb{R}_+$, if $\mathsf{LF}_\varepsilon(f, V)$ is true then there exists $\delta \in \mathbb{Q}_+$ such that any δ-complete decision procedure must return that $\mathsf{LF}_\varepsilon(f, V)$ is true.*

Proof. Fix an arbitrary $\varepsilon \in \mathbb{R}_+$ for which $\mathsf{LF}_\varepsilon(f, V)$ is true. Let $\phi \coloneqq \mathsf{LF}_\varepsilon(f, V)$, and using Lemma 1, let $\delta \in \mathbb{Q}_+$ be such that ϕ is δ-robust. Since ϕ is true, we conclude $\phi^{+\delta}$ is true as well. Using Definition 4, no δ-complete decision procedure can return δ-false for ϕ. □

We remark that the quantifier alternation used in Definition 8 can be eliminated without extra search steps. It confirms that we only need to run SMT solving to handle the universally quantified subformula. The reason is that the α, β, and γ parameters can be found by estimating the range of $V(x)$ and $\nabla_f V(x)$ in the different neighborhoods. In fact, we can rewrite $\mathsf{LF}_\varepsilon(f, V)$ in the following way to eliminate the use of α, β, and γ:

$$\mathsf{LF}_\varepsilon(f, V) \leftrightarrow \exists^{(0,\varepsilon)}\varepsilon' \left(\sup_{x \in \mathcal{B}_{\varepsilon'}} V(x) < \inf_{x \in D \setminus \mathcal{B}_\varepsilon} V(x) \wedge \sup_{x \in D \setminus \mathcal{B}_{\varepsilon'}} \nabla_f V(x) < 0 \right)$$

Note that in this form the universal quantification is implicit in the sup and inf operators. In this way, the formula is existentially quantified on only ε', which can then be handled by binary search. This is an efficient way of checking the conditions in practice. We also remark that without this method, the original formulation with multiple parameters can be directly solved as $\exists\forall$-formulas as well using more expensive algorithms [13].

4 Robust Proofs for Safety

In this section, we define two types of ε-barrier functions that are robust to numerical perturbations.

Proving unbounded safety requires the use of barrier functions. The idea is that if one can find a barrier function that separates initial conditions from the set of unsafe states, such that no trajectories can cross the barrier from the safe to the unsafe side, then the system is safe. Here we use a formulation similar to the that of Prajna [21]. The standard conditions on barrier functions include constraints on the vector field of the system at the exact boundary of the barrier set, which introduces robustness problems. We show that it is possible to avoid these problems using two different formulations, which we call Type 1 and Type 2 ε-barrier functions. Type 1 ε-barrier functions strengthen the original definition and requires strict contraction of the barrier. Instead of only asking the system to be contractive exactly on the barrier's border, we force it to be contractive when reaching any state within a small distance from the border. Type 2 ε-barrier functions allow the system to escape the barrier for a controllable distance and a limited period of time. It should then return to the interior of the safe region. Type 1 ε-barriers can be seen as a subclass of Type 2 ε-barriers. The benefit for allowing bounded escape is that the shape of the barrier no longer needs to be an invariant set, which can be particularly helpful when the shape of the system invariants cannot be determined or expressed symbolically. The downside to Type 2 ε-barriers is that checking the corresponding conditions requires integration of the dynamics, which can be expensive but can still be handled

by δ-complete decision procedures. The intuition behind the two definitions is shown in Figure 2 and will be explained in detail in this section.

4.1 Safety and Barrier Functions

Before formally introducing robust safety and ε-barrier functions, we define the safety and barrier functions first. It is easy to see that the robustness problem with the barrier functions is similar to that of Lyapunov functions: if the boundary is exactly separating the safe and unsafe regions then the inductive conditions are not robust, since deviations in the variables by even a small amount from the barrier will make it impossible to complete the proof.

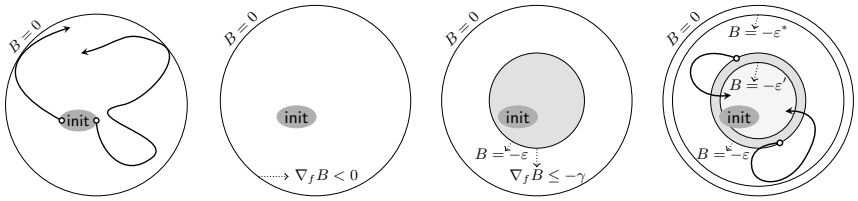
Definition 9 (Safety). *Let $B : D \to \mathbb{R}$ be a scalar function defined for the system in (1). We say $B \leq 0$ defines a safe (or forward invariant) set for the system, iff the following formula is true:*

$$\mathsf{Safe}(f, \mathsf{init}, B) \equiv_{df} \forall^D x_0 \forall^{[0,\infty)} t \Big(\mathsf{init}(x_0) \to B(F(x_0, t)) \leq 0 \Big).$$

Definition 10 (Barrier Function). *Let $B : X \to \mathbb{R}$ be a differentiable scalar function defined for the system in (1). We say B is a barrier function for the system, iff the following formula is true:*

$$\mathsf{Barrier}(f, \mathsf{init}, B) \equiv_{df} \forall^D x \left(\Big(\mathsf{init}(x) \to B(x) \leq 0 \Big) \wedge \Big(B(x) = 0 \to \nabla_f B(x) < 0 \Big) \right)$$

Proposition 3. $\mathsf{Barrier}(f, \mathsf{init}, B) \to \mathsf{Safe}(f, \mathsf{init}, B)$.



(a) Safety (b) Barrier (c) Type 1 ε-Barrier (d) Type 2 ε-Barrier

Fig. 2: Type 1 and Type 2 ε-Barriers

4.2 Type 1: Strict Contraction

In the standard definition, the boundary of the barrier set is typically a manifold defined by equality, which is not numerically robust. To avoid this problem, we need the barrier boundary to be *belt-shaped* in the sense that there is a clear gap between the safe and unsafe regions. The idea is as shown in Figure 2c: we need a second and stronger barrier defined by $B = -\varepsilon$ for some reasonable ε, so that the system is clearly separated from $B = 0$. The formal definition is as follows.

Definition 11 (ε-Barrier Certificates). *Let $\varepsilon \in \mathbb{R}_+$ be arbitrary. A differentiable scalar function $B : D \to \mathbb{R}$ is an ε-barrier function iff the following conditions are true:*
- *For all x, $\mathsf{init}(x)$ implies $B(x) \leq -\varepsilon$.*
- *There exists $\gamma \in \mathbb{R}_+$ such that for all x, $B(x) = -\varepsilon$ implies $\nabla_f B(x) \leq -\gamma$.*

Formally, the condition is defined as

$$\mathsf{Barrier}_\varepsilon(f, \mathsf{init}, B) \equiv_{df} \forall^D x \Big(\mathsf{init}(x) \to B(x) \leq -\varepsilon\Big) \\ \wedge \exists^{(0,\infty)} \gamma \forall^D x \Big(B(x) = -\varepsilon \to \nabla_f B(x) \leq -\gamma\Big)$$

It should be intuitively clear from the definition that the existence of ε-barrier functions is sufficient for establishing invariants and safety properties. The new requirement is that the system stays robustly within the barrier, by the area defined by $-\varepsilon \leq B(x) \leq 0$.

Theorem 4. *For any $\varepsilon \in \mathbb{R}_+$, $\mathsf{Barrier}_\varepsilon(f, \mathsf{init}, B) \to \mathsf{Safe}(f, \mathsf{init}, B)$.*

Proof. Assume $\mathsf{Barrier}_\varepsilon(f, \mathsf{init}, B)$ is true. It is easy to see $\mathsf{Barrier}(f, \mathsf{init}, B+\varepsilon)$, as specified in Definition 10, is also true. Therefore, using Proposition 3, we know $\mathsf{Safe}(f, \mathsf{init}, B + \epsilon)$ and hence $\mathsf{Safe}(f, \mathsf{init}, B)$ are both true. □

It is clear that there is room for numerically perturbing the size of the area and still obtaining a robust proof. The proof is similar to the one for Lemma 1 as shown in [8].

Theorem 5. *For any $\varepsilon \in \mathbb{R}_+$, there exists $\delta \in \mathbb{Q}_+$ such that $\mathsf{Barrier}_\varepsilon(f, \mathsf{init}, B)$ is a δ-robust formula.*

Example 2 (Type 1 ε-Barrier for timed-reversed Van der Pol). Consider the time-reversed Van der Pol system introduced in Example 1. We use the same example to demonstrate the effect of numerical errors in proving barrier certificates. The level sets of the Lyapunov functions in the stable region are barrier certificates; however, for the barriers that are very close to the limiting cycle, numerical sensitivity becomes a problem. In experiments, when $\varepsilon = 10^{-5}$ and $\delta = 10^{-4}$, we can verify that the level set $z^T P z = 90$, is a Type 1 ε-barrier. Table 2 lists parameters used in this proof. Figure 3 (Left) shows the direction field for the timed-reversed Van der Pol dynamics, the border of the set $z^T P z \leq 90$, which we prove is a type 1 ε-barrier, and the boundary of set $z^T P z \leq 110$, which is clearly not a barrier, since it is outside of the limit cycle.

The conditions for ε-Lyapunov and ε-barrier functions look very similar, but there is an important difference. In the case of Lyapunov functions, we do not evaluate the Lie derivative of the balls. Thus, the balls do not define barrier sets. On the other hand, the level sets of Lyapunov functions always define barriers.

Remark 3. The ε-barrier functions can also be used as a sufficient condition for ε-stability, if a barrier can be found within the ε-ball required in ε-stability.

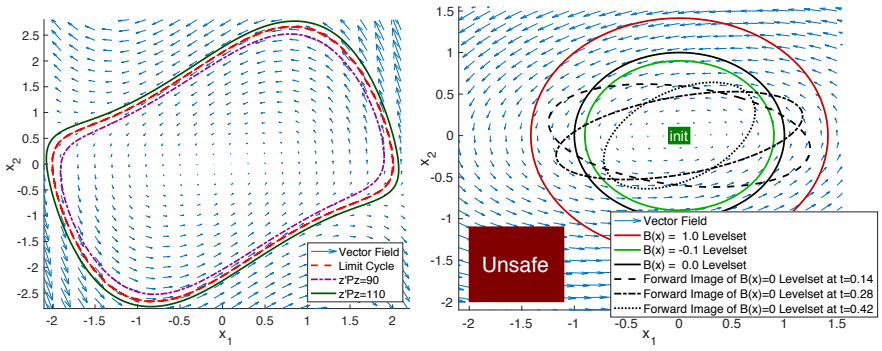


Fig. 3: (Left) Van der Pol Example (Right) Type 2 Barrier Example

Remark 4. A technical requirement for proving robustness of the ε-barrier conditions is that $\neg\mathsf{init}$ defines a simple set that can be over-approximated, such that for every $\varepsilon \in \mathbb{R}_+$, there is $\delta \in \mathbb{R}_+$ such that for any point that satisfies $\neg\mathsf{init}^{+\delta}$ there is an ε-close point that satisfies $\neg\mathsf{init}$. A sufficient condition for this restriction is that init be of the form $(\bigwedge_i a_i \leq x_i \leq b_i) \rightarrow \varphi(x)$, where $a_i, b_i \in \mathbb{Q}$ are arbitrary constants, and φ is a quantifier-free formula with only strict inequalities [22].

4.3 Type 2: Bounded Escape

We now introduce the second set of conditions for establishing ε-invariant sets. This set of conditions can be used only when the ε-variations are considered. This notion is inspired by the notion of k-step invariants [3] for discrete-time systems. The ε-margin that we allow at the boundary of the invariants allows us to exploit more techniques. Using reachable set computation, we can directly check if all states stay within the barrier set at each step. To ensure that the conditions are inductive and useful, we need to impose the following two requirements:

- (Contraction) Similar to the strengthening in barrier certificates, we require that the system does not *sit at the boundary*: the dynamics at the boundary should be contracting. The difference with Type 1 ε-barriers is that, this condition is not imposed through the vector field on the boundary. Instead, it is a reachability condition: after some amount of time, all states should return to the interior of an appropriate set.
- (Bounded Escape) Before reaching back to the invariant set, we allow the system to step outside the invariant, but only up to a bounded distance from the boundary.

The intuition is depicted in Figure 2d. In the formal definition, we parameterize the conditions with the time for contraction and the maximum deviation from the invariant set, as follows.

Definition 12 (Type 2 Barrier Functions). *Let $T, \varepsilon \in \mathbb{R}_+$ be arbitrary. We say a continuous scalar function B defines a (T, ε)-elastic barrier function, iff the following conditions hold:*

1. *For any x, $\mathsf{init}(x)$ implies $B(x) \leq -\varepsilon$.*
2. *There exists $\varepsilon' > \varepsilon$ such that any state in $B(x) \leq -\varepsilon$ will enter $B(x) \leq -\varepsilon'$ after time T.*
3. *During time $[0, T]$, the system may step outside of $B(x) \leq -\varepsilon$ but there exists some $\varepsilon^* \in (0, \varepsilon]$ such that all states stay within $B(x) \leq -\varepsilon^*$.*

In all, we define the conditions with the following formula

$$\begin{aligned}\mathsf{Barrier}_{T,\varepsilon}(f, \mathsf{init}, B) \equiv_{df} \forall^D x\Big(\mathsf{init}(x) \to B(x) \leq -\varepsilon\Big) \\ \wedge \exists^{(0,\varepsilon]}\varepsilon^* \forall^D x \forall^{[0,T]} t\Big((B(x) = -\varepsilon) \to B(F(x,t)) \leq -\varepsilon^*\Big) \\ \wedge \exists^{(\varepsilon,\infty)}\varepsilon' \forall^D x\Big((B(x) = -\varepsilon) \to B(F(x,T)) \leq -\varepsilon'\Big)\end{aligned}$$

Theorem 6, shows that conditions in Definition 12 ensure that the system never leaves the invariant $B \leq 0$. The key is the second condition: induction works because all states come back to the interior of the set defined by $B \leq -\varepsilon$. With the third condition only, we cannot perform induction because the set may keep growing.

Theorem 6. *For any $T, \varepsilon \in \mathbb{R}_+$, $\mathsf{Barrier}_{T,\varepsilon}(f, \mathsf{init}, B) \to \mathsf{Safe}(f, \mathsf{init}, B)$.*

Proof. For the purpose of contradiction, suppose starting from $x_0 \in \mathsf{init}$, the system is unsafe. Using continuity of the barrier B and the solution function F, let $t \in \mathbb{R}_{\geq 0}$ be a time at which $B(x(t)) = 0$, where $x(t)$ is by definition $F(x_0, t)$. By the 1^{st} property in Definition 12, we know $B(x_0) \leq -\varepsilon < 0$. Using continuity of B and F, let $t' \in [0, t)$ be the supremum of all times at which $B(x(t')) = -\varepsilon$. By the 3^{rd} property in Definition 12, we know $t - t' > T$, and by the 2^{nd} property in Definition 12, we know $B(x(t' + T)) \leq -\varepsilon' < -\varepsilon$. Using continuity of B and F, we know there is a time $t'' \in (t' + T, t)$ at which $B(x(t'')) = -\varepsilon$. However, this is in contradiction with t' being the supremum. □

Theorem 7. *For any $\varepsilon \in \mathbb{R}_+$, there exists $\delta \in \mathbb{Q}_+$ such that $\mathsf{Barrier}_{T,\varepsilon}(f, \mathsf{init}, B)$ is a δ-robust formula.*

Example 3. We use this example to show how Type 2 ε-barriers can be used to establish safety. Consider the following system.

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-0.1 & -10 \\ 4 & -2\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix}$$

Let init be the set $\{x \mid -0.1 \leq x_1 \leq 0.1, -0.1 \leq x_2 \leq 0.1\}$, and let U, the unsafe set, be the set $\{x \mid -2.0 \leq x_1 \leq -1.1, -2.0 \leq x_2 \leq -1.1\}$. The system is stable and safe with respect to the designated unsafe set. However, the safety cannot be shown using any invariant of the form $B(x) \coloneqq x_1^2 + x_2^2 - c \leq 0$, where $c \in \mathbb{Q}_+$ is a constant, in the standard definition. This is because the vector field on the

boundary of such sets do not satisfy the inductive conditions. Nevertheless, we can show that for $c = 1$, $B(x)$ is a Type 2 ε-barrier. The dReal query verifies the conditions with $\varepsilon = 0.1$. Since $U(x) \to B(x) > \epsilon$ and $\mathsf{init}(x) \to B(x) < -\varepsilon'$, we know that the system cannot reach any unsafe states. Figure 3 (Right), illustrates the example. The green set at the center represents init, and the red set represents unsafe set U. The $B(x) = 0$ level set is not invariant, as evidenced in the figure by the forward images at $t = 0.14$ and $t = 0.28$ leaving the set; however, as the dReal query proves, the reachable set over $0 \leq t \leq 10$ does not leave the $B(x) = 1.0$ level set and is completely contained in the $B(x) = -0.1$ level set by $t = 0.4$. Since $U(x) \to B(x) > 1.0$ and $\mathsf{init}(x) \to B(x) < -0.1$, then the system cannot reach any state in U.

5 Experiments

In this section, we show examples of nonlinear systems that can be verified to be ε-stable or safe with ε-barriers.

Example	α	β	γ	ε	ε'	Time (s)
T.R. Van der Pol	2.10×10^{-23}	1.70×10^{-23}	10^{-25}	10^{-12}	5×10^{-13}	0.05
Norm. Pend.	7.07×10^{-23}	3.97×10^{-23}	10^{-50}	10^{-12}	5×10^{-13}	0.01
Moore-Greitzer	2.95×10^{-19}	2.55×10^{-19}	10^{-20}	10^{-10}	5×10^{-11}	0.04

Table 1: Results for the ε-Lyapunov functions. Each Lyapunov function is of the form $z^T P z$, where z is a vector of monomials over the state variables. We report the constant values satisfying the ε-Lyapunov conditions, and the time that verification of each example takes (in seconds).

Example	ℓ	ε	γ	degree(z)	size of P	Time (s)
T.R. Van der Pol	90	10^{-5}	10^{-5}	3	9×9	6.47
Norm. Pend.	$[0.1, 10]$	10^{-2}	10^{-2}	1	2×2	0.08
Moore-Greitzer	$[1.0, 10]$	10^{-1}	10^{-1}	4	5×5	13.80
PTC	0.01	10^{-5}	10^{-5}	2	14×14	428.75

Table 2: Results for the ε-barrier functions. Each barrier function $B(x)$ is of the form $z^T P z - \ell$, where z is a vector of monomials over x. We indicate the highest degree of the monomials used in z, the size of the P, the level ℓ used for each barrier function, and the value of ε and γ used to the check $\nabla_f B(x) < -\gamma$.

Table 1 contains parameters we use to verify requirements of Definition 8 for ε-Lyapunov functions in our examples. Table 2 contains parameters we use

to verify requirements of Definition 11 for Type 1 ε-barrier functions in our examples. The ε-Lyapunov functions in these examples are of the form $V(x) \coloneqq z^T P z$, where z is a vector of products of the state variables and P is a constant matrix obtained using simulation-guided techniques from [10]. All the P matrices are given in [8].

Time-Reversed Van der Pol. The time-reversed Van der Pol system has been used as an example in the previous sections. Figure 3 (Left) shows the direction field of this system around the origin. Using dReal with $\delta \coloneqq 10^{-25}$, we are able to establish a 10^{-12}-Lyapunov function and a 10^{-5}-barrier function.

Normalized Pendulum. A standard pendulum system has continuous dynamics containing a transcendental function, which causes difficulty for many techniques. Here, we consider a normalized pendulum system with the following dynamics, in which x_1 and x_2 represent angular position and velocity, respectively. In our experiment, using $\delta = 10^{-50}$, we can prove that function $V \coloneqq x^T P x$ is ε-Lyapunov, where $\varepsilon \coloneqq 10^{-12}$.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\sin(x_1) - x_2 \end{bmatrix} \tag{3}$$

Using $\delta \coloneqq 0.01$, we are able to prove that for *any* value $\ell \in [0.1, 10]$, the function $B(x) \coloneqq x^T P x - \ell$, with x being the system state, and P a constant matrix given in [8], is a Type 1 0.01-barrier function.

Moore-Greitzer Jet Engine. Next, we consider a simplified version of the Moore-Greitzer model for a jet engine. The system has the following dynamics, in which x_1 and x_2 are states related to mass flow and pressure rise.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 - \frac{3}{2}x_1^2 - \frac{1}{2}x_1^3 \\ 3x_1 - x_2 \end{bmatrix} \tag{4}$$

In our experiment, using $\delta = 10^{-20}$ and $z \coloneqq [x_1^2, x_1 x_2, x_2^2, x_1, x_2]^T$, we can prove that function $V \coloneqq z^T P z$ is ε-Lyapunov, where $\varepsilon \coloneqq 10^{-10}$.

Using dReal with $\delta \coloneqq 0.1$, we are able to prove that for *any* value $\ell \in [1, 10]$, the function $B(x) \coloneqq z^T P z - \ell$, with x being the system state, z being the vector of monomials defined in the previous section, and P a constant matrix given in [8], is a Type 1 0.1-barrier function.

Powertrain Control System. Next, we consider a closed-loop model of a powertain control (PTC) system for an automotive application. The system dynamics consist of four state variables, two associated with a plant and two for a controller. The plant models fuel and air dynamics of an internal combustion engine and the controller is designed to regulate the air-fuel (A/F) ratio within a given range of an optimal value, referred as stoichiometric value. Two states related to the plant represent the manifold pressure, p, and the ratio between actual A/F ratio and stoichiometric value, r. The two associated with the controller are the estimated manifold pressure, p_{est}, and the internal state of the PI

controller, i. The system is highly nonlinear, with the following dynamics

$$\begin{aligned}
\dot{p} &= c_1 \left(2\hat{u_1} \sqrt{\frac{p}{c_{11}} - \left(\frac{p}{c_{11}}\right)^2} - \left(c_3 + c_4 c_2 p + c_5 c_2 p^2 + c_6 c_2^2 p\right) \right) \\
\dot{r} &= 4 \left(\frac{c_3 + c_4 c_2 p + c_5 c_2 p^2 + c_6 c_2^2 p}{c_{13}(c_3 + c_4 c_2 p_{est} + c_5 c_2 p_{est}^2 + c_6 c_2^2 p_{est})(1 + i + c_{14}(r - c_{16}))} - r \right) \\
\dot{p}_{est} &= c_1 \left(2\hat{u_1} \sqrt{\frac{p}{c_{11}} - \left(\frac{p}{c_{11}}\right)^2} - c_{13} \left(c_3 + c_4 c_2 p_{est} + c_5 c_2 p_{est}^2 + c_6 c_2^2 p_{est}\right) \right) \\
\dot{i} &= c_{15}(r - c_{16})
\end{aligned}$$

which followed the detailed description of the model and the constant parameter values in [10]. We verified that there exists a function of the form $B(x) = z^T P z - 0.01$ (z consist of 14 monomials with a maximum degree of 2), where $\nabla_f B(x) < -\gamma$, when $B(x) = -\varepsilon$.

6 Conclusion

We formulated new inductive proof rules for stability and safety for dynamical systems. The rules are numerically robust, making them amenable to verification using automated reasoning tools such as those based on δ-decision procedures. We presented several examples demonstrating the value of the new approach, including safety verification tasks for highly nonlinear systems. The examples show that the framework can be used to prove stability and safety for examples that were out of reach for existing tools. The new framework relies on the ability to generate reasonable candidate Lyapunov functions, which are analogous to ranking functions from program analysis. Future work will include improved techniques for efficiently generating the ε-Lyapunov and ε-barrier functions and related theoretical questions.

7 Acknowledgement

Our work is supported by the United States Air Force and DARPA under Contract No. FA8750-18-C-0092, AFOSR No. FA9550-19-1-0041, and the National Science Foundation under NSF CNS No. 1830399. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

1. Stanley Bak. t-barrier certificates: A continuous analogy to k-induction. *IFAC Conference on Analysis and Design of Hybrid Systems*, 2018.
2. Stephen R. Bernfeld and V. Lakshmikantham. Practical stability and lyapunov functions. *Tohoku Math. J. (2)*, 32(4):607–613, 1980.

3. Ruxandra Bobiti and Mircea Lazar. A delta-sampling verification theorem for discrete-time, possibly discontinuous systems. In *HSCC*, 2015.
4. George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In H. Brakhage, editor, *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, pages 134–183, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.
5. Martin Fränzle, Christian Herde, Tino Teige, Stefan Ratschan, and Tobias Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *JSAT*, 1(3-4):209–236, 2007.
6. Sicun Gao, Jeremy Avigad, and Edmund Clarke. Delta-complete decision procedures for satisfiability over the reals. In *Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, pages 286–300, 2012.
7. Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. Delta-decidability over the reals. In *LICS*, pages 305–314. IEEE Computer Society, 2012.
8. Sicun Gao, James Kapinski, Jyotirmoy Deshmukh, Nima Roohi, Armando Solar-Lezama, Nikos Arechiga, and Soonho Kong. Numerically-Robust Inductive Proof Rules for Continuous Dynamical Systems (extended version). https://dreal.github.io/CAV19/, 2019.
9. Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In Maria Paola Bonacina, editor, *CADE*, pages 208–214, 2013.
10. James Kapinski, Jyotirmoy V. Deshmukh, Sriram Sankaranarayanan, and Nikos Aréchiga. Simulation-guided lyapunov analysis for hybrid dynamical systems. In *Hybrid Systems: Computation and Control*, 2014.
11. Hassan K. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, (N.J.), 1996.
12. Soonho Kong, Sicun Gao, Wei Chen, and Edmund Clarke. dReach: δ-reachability analysis for hybrid systems. In Christel Baier and Cesare Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 200–205, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
13. Soonho Kong, Armando Solar-Lezama, and Sicun Gao. Delta-decision procedures for exists-forall problems over the reals. In *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part II*, pages 219–235, 2018.
14. J.P. LaSalle and S. Lefschetz. *Stability by Liapunov's direct method: with applications*. Mathematics in science and engineering. Academic Press, 1961.
15. D. Liberzon, C. Ying, and V. Zharnitsky. On almost lyapunov functions. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 3083–3088, Dec 2014.
16. David Monniaux. A survey of satisfiability modulo theory. In *CASC*, volume 9890 of *Lecture Notes in Computer Science*, pages 401–425. Springer, 2016.
17. Antonis Papachristodoulou and Stephen Prajna. *Analysis of Non-polynomial Systems Using the Sum of Squares Decomposition*, pages 23–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
18. Pablo Parrilo. Structured semidenite programs and semialgebraic geometry methods in robustness and optimization. *PhD thesis*, 08 2000.
19. André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In Aarti Gupta and Sharad Malik, editors, *CAV*, volume 5123 of *LNCS*, pages 176–189. Springer, 2008.

20. Andreas Podelski and Silke Wagner. Model checking of hybrid systems: From reachability towards stability. In *Hybrid Systems: Computation and Control*, pages 507–521. Springer, 2006.
21. Stephen Prajna. *Optimization-based methods for nonlinear and hybrid systems verification*. PhD thesis, California Institute of Technology, Pasadena, CA, USA, 2005. AAI3185641.
22. Nima Roohi, Pavithra Prabhakar, and Mahesh Viswanathan. Relating syntactic and semantic perturbations of hybrid automata. In *CONCUR*, pages 26:1–26:16, 2018.
23. Alfred Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley, 2nd edition, 1951.
24. Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44:2669–2675, 2008.
25. Klaus Weihrauch. *Computable Analysis: An Introduction*. Springer Publishing Company, Incorporated, 1st edition, 2013.
26. Leonard Weiss and EF Infante. On the stability of systems defined over a finite time interval. *Proceedings of the National Academy of Sciences of the United States of America*, 54(1):44, 1965.
27. Leonard Weiss and EF Infante. Finite time stability under perturbing forces and on product spaces. *Automatic Control, IEEE Transactions on*, 12(1):54–59, 1967.
28. Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.
29. Guisheng Zhai and A.N. Michel. On practical stability of switched systems. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 3, pages 3488–3493 vol.3, Dec 2002.
30. Guisheng Zhai and Anthony N Michel. Generalized practical stability analysis of discontinuous dynamical systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 1663–1668. IEEE, 2003.