



Comparison Invariants for Verifying Control Invariance

Promit Panja and André Platzer

Department of Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
{promit.panja, andre.platzer}@kit.edu

Abstract

Control invariance validates that dynamical systems have a control input that preserves a given property at all times. This paper introduces a set of sound axioms and proof rules in differential dynamic logic (dL) that enable verification of control invariance. First, the *scalar* and *vector comparison principles*, relating a system of differential equations to a comparison system such that invariance properties can be established more easily, are axiomatized in dL. This axiomatization primarily utilizes *differential ghosts*, which are proof-theoretic generalizations of comparison systems. Next, with the comparison principles serving as the basis, *comparison invariants* are introduced, and sound axioms and proof rules are derived. Comparison invariants reduce the question of control invariance to a functional inequality on its Lie derivative for a suitable class of functions, moreover, the right choice of function can result in decidable arithmetic. Furthermore, the perennially popular control barrier functions (CBFs) used in safety-critical control are shown to be a special instance of comparison invariants. This yields an axiomatization of CBFs that leads to a dedicated set of proof rules. The rules allow for the verification of CBFs, which are traditionally used for synthesizing safe controllers without verification. Lastly, comparison invariants are shown to unify several other safety verification techniques, including Darboux invariants and differential invariants, further cementing their versatility.

1 Introduction

Control systems and cyber-physical systems are an integral part of modern society [2, 28], which are deployed in various domains such as aerospace, medical devices, autonomous vehicles, industrial robots, and power grids [24, 34]. Control systems are often termed *safety-critical* when safety is a primary concern, breach of which can cause catastrophic damage including loss of life [2, 24, 28]. Consequently, safety-critical control systems require formal safety guarantees. Making sure that a control system behaves safely under all admissible control inputs has been a central challenge in control theory and formal verification [18, 28, 34]. At the heart of this challenge lies control invariance, the principle that a system’s differential equations can admit control inputs that preserve a desired property.

This paper studies deductive verification of control invariance by developing an axiomatic foundation in *differential dynamic logic* (dL) [23–26, 28–31], a logic for modeling and proving hybrid systems. In order to axiomatically prove control invariance we utilize the *comparison principles* [39] as the underlying proof technique. The comparison principles enable relating a

system of differential equations with a comparison system aiding invariance proofs by establishing properties that were not possible before. Based on the comparison principles we introduce a new general class of invariants called *comparison invariants*, which reduce the question of control invariance to questions about real arithmetic, specifically a functional inequality on the Lie derivative of the invariants, proving sublevel sets as control invariant. Comparison invariants are used to further generalize various safety verification techniques including the ever popular control barrier functions [3].

Control barrier functions are a go-to choice in safety-critical control theory for synthesizing efficient safety enforcing controllers [3–5, 40]. Due to their relative simplicity and their formal safety guarantees control barrier functions have attracted much success, especially in fields like safety-critical robotics. However, these formal guarantees are only enforced by the synthesized controllers if the control barrier function used to synthesize it was valid in the first place. Verifying whether a candidate control barrier function is valid or not is a nontrivial task [11]. Recently, there have been efforts in verifying control barrier functions, primarily utilizing sum-of-squares optimization using semidefinite programming [11], however, in-practice these methods can suffer from potential soundness issues due to round-off errors in floating-point solvers.

Due to a lack of formal logical embeddings for verifying control barrier functions, this work was motivated to bridge this gap by providing an axiomatic foundation for proving control invariance in general, and as a consequence, also control barrier functions. With this work our aim is to provide a unified proof framework in dL, which allows proving various control invariance properties via a general set of axioms and proof rules. Moreover, we propose a logical foundation for verifying control barrier functions for the safety-critical control community for synthesizing safe controllers, in which unverified candidate control barrier functions abound.

Contributions. The main contributions are summarized as follows.

1. We first axiomatize the *scalar* and *vector comparison principles*, and show they are derivable in dL resulting in axioms **SCP** and **VCP**, respectively.
2. Next, we introduce *scalar* and *vector comparison invariants* along with their respective axioms and proof rules, enabling sound verification of control invariance properties in dL.
3. Further, we give an axiomatization of *control barrier functions* (CBFs), scalar and vector, and in the process show how they are a special case of comparison invariants.
4. Additionally, we show the existing dL axioms and proof rules for Darboux invariants and differential invariants, respectively, are derivable from comparison invariants.

2 Background: Differential Dynamic Logic

Differential dynamic logic (dL), a logic for modeling and reasoning about hybrid systems [28], forms the basis of this paper. We will briefly go over the syntax, semantics, and the proof calculus, followed by control systems modeling principles in dL. A detailed exposition of the logic can be found elsewhere [26, 28].

2.1 Syntax

Here we give an overview of the dL syntax, which comprises *terms* and *formulas* that are then combined to form *hybrid programs* modeling hybrid systems.

2.1.1 Terms

The *term language* of **dL** follows the following grammar, where $x \in \mathbb{V}$ is a variable in the set of all variables $\mathbb{V}$, and $c \in \mathbb{Q}$ is a rational constant:

$$p, q ::= x \mid c \mid p + q \mid p \cdot q \mid \eta(p_1, \dots, p_n) \mid (p)'$$

Also, $x = (x_1, \dots, x_n)^\top$ is a vector of variables. It is worth noting that this paper frequently utilizes (fixed) function symbols like $\eta(\cdot)$, which are scalar functions, and $\boldsymbol{\eta}(\cdot) = (\eta_1(\cdot), \dots, \eta_n(\cdot))^\top$ are vector-valued functions, where $\eta_1(\cdot), \dots, \eta_n(\cdot)$ are scalar functions. All the function symbols are assumed to be continuous in their respective arguments. For further distinction, vector of terms are written in bold $\mathbf{p} = (p_1, \dots, p_n)^\top$. The term language also includes *differential terms* $(p)'$ that allow reasoning about derivatives locally, along the evolution of associated *ordinary differential equations* (ODEs) [31], and are defined as follows:

$$(p)' \stackrel{\text{def}}{=} \sum_{i=1}^n \frac{\partial p}{\partial x_i} \cdot x'_i \quad \mathcal{L}_{\mathbf{f}(x)}(p) \stackrel{\text{def}}{=} \sum_{i=1}^n \frac{\partial p}{\partial x_i} \cdot f_i(x).$$

The differential of a term p coincides with its syntactic Lie derivative along the ODE $x' = \mathbf{f}(x)$, with $x' = (x'_1, \dots, x'_n)^\top$ and $\mathbf{f}(x) = (f_1(x), \dots, f_n(x))^\top$, when following the chain rule. Here we use $\mathcal{L}_{\mathbf{f}(x)}(p)$ as an operator and use $\dot{p}$ as the syntactic representation of the Lie derivative for brevity. For two vectors $\mathbf{v}$ and $\mathbf{w}$ the inner product is denoted as $\mathbf{v} \cdot \mathbf{w}$ and the Euclidean norm is $\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$. Since standard norms are not part of **dL** term syntax, squared Euclidean norms are used whenever necessary, as they are definable in first-order logic over real arithmetic ($\text{FOL}_{\mathbb{R}}$) as $\|\mathbf{v}\|^2 = \sum_{i=1}^n v_i^2$.

2.1.2 Formulas

The *formulas* of **dL** are defined by the following grammar, where $\sim \in \{>, \geq, =\}$ is a comparison operator and α is a *hybrid program* [28]:

$$\phi, \psi ::= p \sim q \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall x\phi \mid \exists x\phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi.$$

This grammar extends the standard first-order logic over real arithmetic ($\text{FOL}_{\mathbb{R}}$), with the box $[\alpha]\phi$ and the diamond $\langle\alpha\rangle\phi$ modalities, which express that *all* or *some* runs of the hybrid program α satisfy ϕ , respectively.

2.1.3 Hybrid Programs

The language of *hybrid programs* is defined by the following grammar, where x is a variable, p is a **dL** term, and Q is a formula of first-order real arithmetic [26, 28].

$$\alpha, \beta ::= x' = \mathbf{f}(x) \& Q \mid \alpha; \beta \mid \alpha^*$$

Continuous dynamics are modeled by the n -dimensional system of autonomous¹ ODEs $x' = \mathbf{f}(x) \& Q$ whose evolution is constrained by the domain Q ; the ODE is modeled as $x' = \mathbf{f}(x)$ when $Q \equiv \text{true}$. The continuous and discrete fragments can be combined to model hybrid dynamics using the following program combinators: *sequential composition* $\alpha; \beta$, which executes program α followed by program β , and (bounded) *nondeterministic repetition* α^* , which repeats the program α zero or arbitrarily more number of times. The syntax of the discrete fragment of **dL** and other program combinators can be found elsewhere [28].

¹An ODE of the form $x' = f(x)$ is called *autonomous*, if the right-hand side of the ODE, i.e., $f(x)$ only depends on the state variable x and **not** explicitly on time.

Remark 1. This paper utilizes systems of ODEs of the form $x' = \mathbf{f}(x, \tau), \tau' = 1 \ \& \ Q$, where the variable τ is specifically meant for tracking time. Due to the ODE $\tau' = 1$, the system of ODEs as a whole is still autonomous in the extended state space [10, Ch. 1, §1.3]. This choice is made so that once can show ODE solutions remain bounded in a compact time interval, which would not have been possible to show syntactically in dL's proof calculus without the addition of the time variable τ .

2.2 Semantics

The semantics of dL are defined such that states $\nu : \mathbb{V} \rightarrow \mathbb{R}$ assign a real value to each variable in $\mathbb{V}$. For some state $\nu \in \mathbb{S}$ and some term p , where $\mathbb{S} \subseteq \mathbb{R}^n$ denotes the set of all states, the semantics $\nu[p]$ is the real value of p when evaluated at ν . For a formula ϕ , the semantics $\llbracket \phi \rrbracket \subseteq \mathbb{S}$ is the set of states in which ϕ holds. First-order connectives have their usual semantics, e.g., $\llbracket \phi \wedge \psi \rrbracket = \llbracket \phi \rrbracket \cap \llbracket \psi \rrbracket$.

Now we present the semantics of formulas most relevant to this paper, i.e., semantics of ODEs under the box modality.² Let $\varphi : [0, T) \rightarrow \mathbb{S}$ (for some $0 < T \leq \infty$), be the unique solution of the ODE $x' = \mathbf{f}(x)$ extended maximally to the right [10, 39] with initial value $\varphi(0) = \nu$, where $\nu \in \mathbb{S}$ is some state:

$$\nu \in \llbracket [x' = \mathbf{f}(x) \ \& \ Q] \psi \rrbracket \text{ iff for all } 0 \leq t < T, \varphi(\xi) \in \llbracket Q \rrbracket \text{ for all } 0 \leq \xi \leq t \text{ such that } \varphi(t) \in \llbracket \psi \rrbracket.$$

Informally it means, the formula $[x' = \mathbf{f}(x) \ \& \ Q] \psi$ is true in the initial state ν if *all* states reached by the solution of the ODE from ν satisfy the postcondition ψ , while staying in the domain constraint Q .

2.3 Proof Calculus

The dL proof calculus enables deductive verification of hybrid dynamical systems by utilizing sound axioms and proof rules of dL via compositional reasoning principles of hybrid programs [26, 28]. All derivations use classical sequent calculus with standard rules for logical connectives and sequents. A sequent is of the form $\Gamma \vdash \Delta$ with semantics $(\bigwedge_{\psi \in \Gamma} \psi) \rightarrow \Delta$, and is valid if and only if the formula is valid. Completed proof branches are marked with *. Since first-order real arithmetic is decidable [8], we assume access to a decision procedure and mark steps justified by real arithmetic with the rule $\mathbb{R}$. An axiom is sound if all its instances are valid. A proof rule is sound if validity of all its premises implies the validity of its conclusion. Axioms and proof rules are derivable if they can be obtained from sound dL axioms and proof rules. The soundness of the base dL axiomatization then guarantees the soundness of all derived axioms and proof rules [26, 28, 31]. An overview of the base dL calculus can be found in various prior works [26, 28, 31].

$$\frac{\Gamma_1 \vdash \phi_1 \quad \dots \quad \Gamma_n \vdash \phi_n}{\vdash \left(\begin{array}{c} \text{hybrid program} \\ \text{reasoning} \end{array} \right)} \quad \Gamma \vdash [\alpha] \phi$$

Figure 1: Proofs proceed upwards by deduction, where each reasoning step is justified by sound dL axioms and proof rules.

2.4 Control Systems in dL

In general, control systems are modeled by parameterized ODEs with an explicit control parameter, which represents the control input. For a (compact) control input set $\mathcal{U} \subseteq \mathbb{R}^m$, following

²The complete dL semantics including semantics of ODEs under the diamond modality can be found elsewhere [26, 28, 31].

is a control system, where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state variable and $u \in \mathcal{U}$ is the control input:

$$x' = \mathbf{f}(x, u)$$

In **dL**, however, control systems are modeled as hybrid programs; discrete programs and continuous dynamics paired together along with nondeterministic repetition.

$$\Gamma \rightarrow \underbrace{[(ctrl; dyn)^*]P}_{\alpha^*} \quad \text{loop} \quad \frac{\Gamma \vdash Inv \quad Inv \vdash [\alpha]Inv \quad Inv \vdash P}{\Gamma \vdash [\alpha^*]P} \quad (1)$$

In the above formula, *ctrl* refers to the (*discrete*) *controller* and *dyn* refers to the *continuous dynamics*, i.e., a system of ODEs. The central reasoning principle for proving hybrid programs is the **loop** rule [28, Ch. 7, §7.3, Lemma 7.3]. This proof rule decomposes the hybrid program α^* in an inductive fashion. The most crucial step is showing that formula *Inv* is an invariant for the hybrid program, more importantly the ODEs in α . Because of this reason the rest of this paper focuses on proving invariants of ODEs, specifically, control invariance of ODEs, which will be defined and looked into in detail in the later sections.

3 Comparison Principle

We begin our endeavor by first taking a look at *comparison systems* and the *comparison principle*, which are fundamental tools in the theory of dynamical systems enabling differential inequality reasoning using differential equations [6, 9, 20, 39].

Comparison System. Informally, a *comparison system* is an auxiliary system of differential equations, which establishes a relation with the original system of differential equations enabling the analysis of properties that were not possible before. For example, for a system of ODEs $x' = \mathbf{f}(x)$, if one wants to show a property $R(x)$ holds true given some property $P(x)$ was true initially, instead of proving this implication directly, which often times can be difficult, one might construct a comparison system $y' = \mathbf{g}(x, y)$ that enables proving from $P(x)$ a new suitable intermediate property $Q(x, y)$, which implies the intended $R(x)$.

$$[x' = \mathbf{f}(x)]P(x) \rightarrow [x' = \mathbf{f}(x), y' = \mathbf{g}(x, y)]Q(x, y) \rightarrow [x' = \mathbf{f}(x)]R(x) \quad (2)$$

At a high-level, the above **dL** formula shows how this relation with a comparison system can be established for a suitable $Q(x, y)$ and $y' = \mathbf{g}(x, y)$.

Remark 2. Of course, formula (2) requires handling side conditions on the solutions of $y' = \mathbf{g}(x, y)$ to maintain soundness. In **dL** this sound reasoning is achieved by *differential ghosts* [26, 31, 38], which are proof-theoretic generalizations of comparison systems. The **dL** axioms **DG** and **BDG** allow introducing auxiliary dynamics that abstract from the existing dynamics to simplify proofs.

Lemma 1 (Differential Ghosts [31, 38]). *The axioms **DG** and **BDG** enable sound additions of linear and nonlinear ghost ODEs, respectively, to the existing systems.*

$$\begin{aligned} \text{DG} \quad & [x' = \mathbf{f}(x) \ \& \ Q]\psi \leftrightarrow \exists y[x' = \mathbf{f}(x), y' = a(x)y + b(x) \ \& \ Q]\psi \\ \text{BDG} \quad & [x' = \mathbf{f}(x), y' = \mathbf{g}(x, y) \ \& \ Q][\|y\|^2 \leq p(x) \rightarrow \\ & ([x' = \mathbf{f}(x) \ \& \ Q]\psi \leftrightarrow [x' = \mathbf{f}(x), y' = \mathbf{g}(x, y) \ \& \ Q]\psi) \end{aligned}$$

A cleverly constructed comparison system can enable reasoning about properties of ODEs with (relative) ease. Once a suitable comparison system has been constructed, there also needs to be a way to relate this system to the original system of ODEs.

The *comparison principle* emerged as a powerful tool in the qualitative theory of dynamical systems [9, 36, 39], and provides a simple mechanism for relating the solutions of two different differential systems. It plays an integral part in the analysis of *partial differential equations* (PDEs) [16]. In deductive verification the comparison principle has been successfully leveraged [27] to prove invariance via Hamilton-Jacobi-Isaacs PDEs for differential hybrid games. For ODEs, the comparison principle has been widely used for stability analysis, generalizing the celebrated Lyapunov's direct method, and certifying invariance [6, 9, 20, 36, 39]. Next we will go over the scalar and vector comparison principles and axiomatize them in dL.

3.1 Scalar Comparison

The scalar comparison principle proves properties about a system of ODEs by relating a scalar differential inequality with another scalar differential equation (comparison system). To demonstrate the scalar comparison principle, consider an n -dimensional system of ODEs $x' = \mathbf{f}(x, t)$ whose solution $x(t)$ exists in some compact time interval $I = [0, T]$ for some $T > 0$. For a scalar function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying the following differential inequality, the corresponding one-dimensional comparison system in z can be constructed

$$h' \geq \eta(h(x), t), \quad z' = \eta(z(t), t) \quad (3)$$

where z is a fresh scalar variable and $\eta : \mathbb{R} \times I \rightarrow \mathbb{R}$ is an appropriately chosen function (the exact criteria are investigated below). The comparison principle enables one to use the solutions $h(x)$, i.e., $h(x(t))$ to relate the solutions $x(t)$ of the original system to the solution $z(t)$ of the comparison system. This sound reasoning is possible due an accompanying comparison theorem.

Definition 1 (Local Lipschitz [39, §10.IV]). A function $\eta(\cdot)$ is said to be *locally Lipschitz* if the following formula is valid, $B_r(a) \equiv t \geq 0 \wedge \|x - a\|^2 < r^2 \wedge \|y - a\|^2 < r^2$ is an open ball of radius r .

$$\text{LocLip}(\eta) \stackrel{\text{def}}{=} \forall \delta \forall a \exists r \exists k \forall t \forall x \forall y (t \leq \delta^2 \wedge B_r(a) \rightarrow \|\eta(x, t) - \eta(y, t)\|^2 \leq k^2 \|x - y\|^2).$$

Theorem 1 (Scalar Comparison Theorem [39, §9.IX]). *Let g and h be scalar functions differentiable in $I = [\xi, \xi + \delta] \subseteq \mathbb{R}$ and $\eta : D \rightarrow \mathbb{R}$ locally Lipschitz for some real domain D . If $h(\xi) \geq g(\xi)$ and $h' - \eta(h, t) \geq g' - \eta(g, t)$ in I , then $h \geq g$ in I .*

This theorem is a general version, which relates two differentiable functions on some (compact) interval. Following is a corollary based on Theorem 1 that directly relates to the system in (3) and is more useful in our case.

Corollary 1. *Let h and z be differentiable scalar functions defined on the interval $I = [0, T] \subset \mathbb{R}$ where $T > 0$. For some locally Lipschitz function $\eta : \mathbb{R} \times I \rightarrow \mathbb{R}$, if $h(0) \geq z(0)$, and if $h' \geq \eta(h, t)$ and $z' = \eta(z, t)$ holds in I , then $h(t) \geq z(t)$ for all $t \in I$.*

Proof. Since $z' - \eta(z, t) = 0$ and $h' - \eta(h, t) \geq 0$, also $h' - \eta(h, t) \geq z' - \eta(z, t)$ in I . The rest follows directly from Theorem 1. $\square$

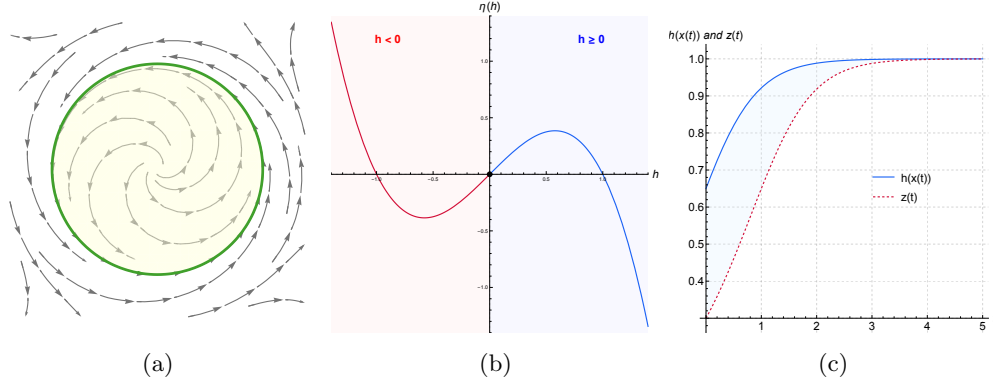


Figure 2: The phase portrait of the system α_e with the invariant $h \geq 0$ from Example 1 is shown in (a). Figure (b) shows how $\eta(h)$ changes as h changes. The right figure (c) shows that $h(x(t))$ is lower bounded by the solution $z(t)$ of the comparison system $z' = \eta(z)$.

An important ingredient of Theorem 1 and Corollary 1 is that the function $\eta(\cdot)$ must be locally Lipschitz, making sure that there are no pathological behaviors in the solution of the differential inequality and the comparison system [39]. The validity of the formula $\text{LocLip}(\eta)$ from Definition 1 ensures that the function $\eta(\cdot)$ is locally Lipschitz, and can be soundly utilized in **dL** to carry out proofs. Moreover, if $\eta(\cdot)$ is a polynomial, then $\text{LocLip}(\eta)$ is a first-order logic formula over the reals ($\text{FOL}_{\mathbb{R}}$), and hence decidable [8].

Remark 3. The function $\eta(\cdot)$, which acts as the bridge relating the differential inequality and the comparison system is often called the *comparison function* [17]. Different comparison functions with different properties can be chosen depending on the proof requirement [17].

Example 1 (Invariance via Comparison). Consider the following 2D dynamical system

$$\alpha_e \equiv x'_1 = -\frac{1}{2}(1 - x_1^2 - x_2^2)(x_1^2 + x_2^2 - 2)x_1 - x_2, \quad x'_2 = -\frac{1}{2}(1 - x_1^2 - x_2^2)(x_1^2 + x_2^2 - 2)x_2 + x_1$$

here $x = (x_1, x_2)^\top \in \mathbb{R}^2$. For the system α_e and the function $h(x) = 1 - x_1^2 - x_2^2$, $h(x) \geq 0$ is an invariant, as illustrated³ in Figure 2a. The invariance of $h \geq 0$ can be proved using the comparison principle from Corollary 1 by showing $\dot{h} \geq \eta(h)$, where $\eta(h) := h - h^3$, holds true. As we can see $\eta(\cdot)$ is a nonlinear locally Lipschitz function, Figure 2b shows how $\eta(h)$ changes when $h \geq 0$ or $h < 0$. Now we can construct the comparison system $z' = \eta(z)$ whose solutions $z(t)$ lower bounds $h(x(t))$ whenever $h(x(0)) \geq z(0) \geq 0$, making $h \geq 0$ an invariant, as can be seen in Figure 2c.

Lemma 2 (Scalar Comparison Principle). *The axiom **SCP** internalizes the scalar comparison principle and is derived syntactically in **dL**, where the term h represents $h(x)$, and $h, z, \eta(\cdot)$ are scalars.*

$$\begin{aligned} \text{SCP} \quad h \geq z \wedge \text{LocLip}(\eta) \rightarrow \\ ([x' = \mathbf{f}(x), \tau' = 1 \ \& \ Q] \dot{h} \geq \eta(h, \tau) \rightarrow [x' = \mathbf{f}(x), \tau' = 1, z' = \eta(z, \tau) \ \& \ Q] h \geq z) \end{aligned}$$

Proof (Sketch). The axiom is proven syntactically in **dL**. The full proof can be found in the technical report [22, Appendix B.1]. At a high level, the proof utilizes axiom **BDG** to introduce

³Figure 2 is only meant for illustration purposes and is not a formal proof. The invariance of $h \geq 0$ can be formally proven in **dL** by utilizing axiom **SCP**.

the comparison system $z' = \eta(z, \tau)$ in the antecedent to match the succedent. This enables proving the new invariant $\exp(k\tau) \cdot (h - z) \geq 0$ using axioms **DC** and **DI_⋈**. Where $\exp(\cdot)$ is the real exponential function, k is the Lipschitz constant, and τ is a variable for tracking time with the associated ODE $\tau' = 1$. This new invariant paired with the fact $\eta(\cdot)$ being locally Lipschitz, i.e., **LocLip**(η), basically says $\exp(k\tau) \cdot (h - z)$ is monotonically increasing, hence proving the original invariant $h \geq z$. $\square$

Axiom **SCP** in the above lemma internalizes Corollary 1 in **dL**, which enables sound utilization of the comparison principle in various invariance proofs. At a high level, axiom **SCP** says for all runs of $x' = \mathbf{f}(x), \tau' = 1, z' = \eta(z, \tau) \ \& \ Q$, h is lower-bounded by z , if $h \geq z$ was true initially, and for all runs of $x' = \mathbf{f}(x), \tau' = 1 \ \& \ Q$, the inequality $\dot{h} \geq \eta(h, \tau)$ holds true, for locally Lipschitz $\eta(\cdot)$. Next we will generalize to vectorial case.

3.2 Vector Comparison

Similar to the scalar comparison principle, the *vector comparison principle* relates the vectorial differential inequality $\mathbf{h}' \geq \boldsymbol{\eta}(\mathbf{h})$ to the comparison system $z' = \boldsymbol{\eta}(z)$. Here $z = (z_1, \dots, z_n)^\top$ is fresh vector of variables and $\boldsymbol{\eta}(\cdot)$ is a locally Lipschitz vector-valued function. Consequently, one might think Theorem 1 and Corollary 1 can be naïvely carried over to the vectorial case to prove $\mathbf{h} \geq z$ with component-wise ordering, i.e., $\bigwedge_{i=1}^n h_i \geq z_i$. Unfortunately, this is not true in general and more regularity assumptions are required on $\boldsymbol{\eta}(\cdot)$ [39, §10, Supplement I]. In order to better understand why that is the case consider the following trivial counterexample.

Counterexample 1. For the system $x' = \mathbf{f}(x)$, where $x \in \mathbb{R}^2$ and $\mathbf{f}(x) = (0, 0)^\top$, let $\mathbf{h}(x) = (0, 0)^\top$ and $\boldsymbol{\eta}(y_1, y_2) = (-y_2, 0)^\top$, where $\mathbf{h} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ and $\boldsymbol{\eta} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Clearly $\boldsymbol{\eta}(\cdot)$ is a vector of polynomials, hence, locally Lipschitz, also the following inequality holds

$$\dot{\mathbf{h}}(x) = (0, 0)^\top, \quad \boldsymbol{\eta}(\mathbf{h}) = (0, 0)^\top, \quad \text{hence,} \quad \dot{\mathbf{h}} \geq \boldsymbol{\eta}(\mathbf{h}).$$

Now consider the comparison system $z' = \boldsymbol{\eta}(z)$, where $z \in \mathbb{R}^2$ is fresh, with the initial condition $z(0) = (0, -1)^\top$, this implies $\mathbf{h}(x(0)) \geq z(0)$ holds. Next compute the solutions of the comparison system

$$z' = \boldsymbol{\eta}(z), \text{ i.e., } z'_1 = -z_2, \ z'_2 = 0, \quad \text{whose solutions are, } z(t) = (t, -1)^\top.$$

From this we can clearly see for all $t > 0$ and any initial value $x(0)$ that $\mathbf{h}(x(t)) \not\geq z(t)$, which shows $\boldsymbol{\eta}(\cdot)$ just being locally Lipschitz is not enough.

The missing property is, *if and only if* the locally Lipschitz vector-valued function $\boldsymbol{\eta}(\cdot)$ is *quasimonotone increasing* then Theorem 1 and Corollary 1 can be lifted to the vectorial case [39, §10.XII]. This property can be encoded as a logical formula and is defined as follows.

Definition 2 (Quasimonotone Increasing [39, §10.XII]). A vector-valued function $\boldsymbol{\eta}(\cdot)$ is called *quasimonotone increasing* if the following formula is true.

$$\text{QuasiMon}(\boldsymbol{\eta}) \stackrel{\text{def}}{=} \forall t \forall x \forall y \left(x \geq y \rightarrow \bigwedge_{i=1}^n (x_i = y_i \rightarrow \eta_i(x, t) \geq \eta_i(y, t)) \right)$$

Where $x \geq y$ is component-wise ordering i.e., $x_i \geq y_i$ for $i = 1, \dots, n$.

This quasimonotonicity fact combined with the local Lipschitz condition can now be utilized to lift the scalar comparison theorem to the vectorial case as stated in the following theorem.

Theorem 2 (Vector Comparison Theorem [39, §10.XII]). *Let $\eta : D \rightarrow \mathbb{R}^n$ be quasimonotone increasing and locally Lipschitz for some real domain D , and let $\mathbf{v}$ and $\mathbf{w}$ be differentiable on $I = [\xi, \xi + \delta] \subseteq \mathbb{R}$. If $\mathbf{v}(\xi) \geq \mathbf{w}(\xi)$ and $\mathbf{v}' - \eta(\mathbf{v}, t) \geq \mathbf{w}' - \eta(\mathbf{w}, t)$ in I , then $\mathbf{v} \geq \mathbf{w}$ in I .*

Similar to the scalar case, the above theorem is a general version for differentiable functions on some compact interval. Following is a direct corollary to Theorem 2, which is more suited to relate properties of ODEs.

Corollary 2. *For some differentiable vector-valued functions $\mathbf{h}$ and z , let $\eta(\cdot)$ be quasimonotone increasing and locally Lipschitz. If $\mathbf{h}' \geq \eta(\mathbf{h}, t)$ and $z' = \eta(z, t)$ hold, along with $\mathbf{h}(0) \geq z(0)$ on $I = [0, T] \subset \mathbb{R}$ for some $T > 0$. Then for all $t \in I$ we have $\mathbf{h} \geq z$.*

Proof. $\mathbf{h}' \geq \eta(\mathbf{h}, t)$ and $z' = \eta(z, t)$ imply $\mathbf{h}' - \eta(\mathbf{h}, t) \geq z' - \eta(z, t)$. Rest by Theorem 2. $\square$

This observation termed as the *vector comparison principle* is axiomatized in $\mathbf{dL}$, which helps in invariance proofs for vectorial postconditions. The following lemma shows the axiom **VCP** that internalizes Corollary 2 and is derived syntactically in $\mathbf{dL}$'s proof calculus.

Lemma 3 (Vector Comparison Principle). *The following axiom derives syntactically in $\mathbf{dL}$ and enables reasoning about ODE properties using the vector comparison principle. Where $\mathbf{h}$ represents $\mathbf{h}(x)$, and $\mathbf{h}$, z , and $\eta(\cdot)$ are vectors.*

$$\begin{aligned} \text{VCP} \quad & \mathbf{h} \geq z \wedge \text{QuasiMon}(\eta) \wedge \text{LocLip}(\eta) \rightarrow \\ & ([x' = \mathbf{f}(x), \tau' = 1 \ \& \ Q] \dot{\mathbf{h}} \geq \eta(\mathbf{h}, \tau) \rightarrow [x' = \mathbf{f}(x), \tau' = 1, z' = \eta(z, \tau) \ \& \ Q] \mathbf{h} \geq z) \end{aligned}$$

Proof (Sketch). The proof follows similar to the scalar case by introducing the comparison system $z' = \eta(z, \tau)$ using axiom **BDG**, which enables proving the new invariant $\exp(k\tau) \cdot (\mathbf{h} - z) \geq 0$ using **DC** and **DI_≥**. But unlike the scalar case, this time it has to be shown that each component of z is upper-bounded by the respective component of $\mathbf{h}$. In order to show this first $\exp(k\tau) \cdot (\mathbf{h} - z) \geq 0$ is equivalently rewritten as $\bigwedge_{i=1}^n (\exp(k\tau) \cdot (h_i - z_i) \geq 0)$, then we construct an intermediate vector $\tilde{z} = (h_1, \dots, h_{i-1}, z_i, h_{i+1}, \dots, h_n)^\top$ that agrees with every component of $\mathbf{h}$ except for the i -th, where it agrees with z , this enables showing $\|\mathbf{h} - \tilde{z}\|^2 = (h_i - z_i)^2$. This paired with the fact that $\eta(\cdot)$ is locally Lipschitz and more importantly quasimonotone increasing enables proving $\exp(k\tau) \cdot (\mathbf{h} - z) \geq 0$. The full proof can be found in the technical report [22, Appendix B.1]. $\square$

4 Control Invariance

With the two comparison principle axioms **SCP** and **VCP** in place we can move forward with our axiomatic analysis of control invariance.

Example 2 (Adaptive Cruise Control (ACC)). The following ODE models the dynamics of a vehicle following another vehicle in straight line with the explicit time variable $\tau \in \mathbb{R}$

$$\alpha_{\text{acc}} \equiv x' = u, x_l' = v_l, \tau' = 1 \ \& \ Q.$$

Here $x, x_l \in \mathbb{R}$ are the positions of the follower and lead vehicle, respectively. Variable $u \in \mathcal{U}$ is the control input, where $\mathcal{U} \subseteq \mathbb{R}$ is some compact set, v_l is the constant velocity of the lead

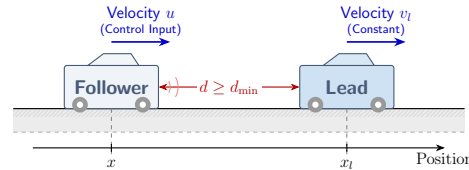


Figure 3: Adaptive cruise control system as modeled in Example 2.

vehicle, and $Q \equiv v_l \geq 0 \wedge T \geq \tau$ is the domain constraint. For dynamics α_{acc} we can model an adaptive cruise control system, as shown in Figure 3, in **dL**, which maintains a safe distance from the lead vehicle as follows

$$\text{Safe} \rightarrow [(\text{ctrl}; \alpha_{\text{acc}})^*] \text{Safe} \quad (4)$$

where $\text{Safe} \equiv x_l - x \geq d_{\min}$. Taking a closer look, the formula Safe encodes the safety condition that says the distance between the lead vehicle and the follower vehicle must be greater than or equal to the minimum distance ($d_{\min}$). Formula (4) can be proven safe in **dL** for the right implementation of the controller (ctrl). But, before one can construct such a controller one would like to prove whether a control input $u \in \mathcal{U}$ even exists that ensures safety, essentially asking the more general question of control invariance, which is formally defined next.

Definition 3 (Control Invariance). Given a control system $x' = \mathbf{f}(x, u)$, where $x \in \mathbb{R}^n$ and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ with $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ locally Lipschitz, and some set $S \subseteq \mathbb{R}^n$, *control invariance* is the property that ensures there exists some control input $u \in \mathcal{U}$ such that when starting from the set S , the solution of the control system remains in the set S for all future times in the domain of existence i.e., $\varphi(t) \in S, \forall t \in [0, T]$, for some $T > 0$. The set S is also known as a *control invariant set*.

Informally, control invariance of S is a property that the system admits, from every state in S , at least one admissible control input that keeps the system inside S as long as the solution exists. One can observe that control invariance asks a more general question of “whether a control input even exists” such that a controller can be constructed for enforcing control invariance. Control invariance properties can be modeled in **dL** by existentially quantifying over the control input u for the control system $x' = \mathbf{f}(x, u)$ i.e.,

$$S \rightarrow \exists u[x' = \mathbf{f}(x, u) \ \& \ Q]S. \quad (5)$$

The validity of formulas of the form (5) proves that there always exist some control input that keeps S invariant, implying control invariance. This shows that proving control invariance is more general than proving a control system stays invariant for a concrete controller implementation, like the adaptive cruise control hybrid program (4). Now we move on to developing sound axioms and proof rules, which enable verifying control invariance in **dL**, this is made possible by comparison invariants, which will be delved into next.

4.1 Comparison Invariants

We now introduce *comparison invariants*, which enable proving control invariance by proving a functional inequality on their Lie derivative. Comparison invariants are based on the ideas of the comparison principles with an additional regularity enforcement on the comparison functions.

Definition 4 (Class $\mathcal{G}$ Functions [17]). A scalar function $\eta(\cdot)$ and a vector-valued function $\boldsymbol{\eta}(\cdot)$ are called (extended) *class $\mathcal{G}$* if the functions are zero at zero and nondecreasing. They are characterized by the following logical formulas.

$$\mathcal{G}(\eta) \stackrel{\text{def}}{=} \forall t \forall x \forall y (\eta(0, t) = 0 \wedge (x \geq y \rightarrow \eta(x, t) \geq \eta(y, t))) \quad \mathcal{G}_{\text{vec}}(\boldsymbol{\eta}) \stackrel{\text{def}}{=} \bigwedge_{i=1}^n \mathcal{G}(\eta_i).$$

The following simple counterexample demonstrates why extra care needs to be taken when handling control systems and the reason comparison invariants require the added regularity over just locally Lipschitz comparison functions even for the scalar case.

Counterexample 2. Consider the control system $x' = u$, where $x \in \mathbb{R}$ and $u \in \mathcal{U} = [-1, 0]$, with the requirement that $h(x) \geq 0$, with $h(x) := x$, stays invariant. In order to show invariance of $h(x) \geq 0$, by the comparison principle from Corollary 1, the system needs to satisfy the inequality

$$\dot{h}(x) - \eta(h, t) \geq 0.$$

Now choosing $\eta(h, t) := -1$, which is locally Lipschitz, but does not satisfy $\eta(0, t) = 0$ to be a class $\mathcal{G}$ function as per Definition 4. We get the inequality $u + 1 \geq 0$, since $\dot{h}(x) = x' = u$. This shows that every admissible control in $\mathcal{U}$ satisfies this inequality, but when the system starts at $x(0) = 0$, and the controller chooses $u = -1$, we get $x(t) = -t < 0$, meaning $h(x) \geq 0$ is no longer invariant, and the system is unsafe. This demonstrates the need for $\eta(0, t) = 0$, to make sure trajectories cannot escape at the boundary. Now choosing $\eta(h, t) := -h$, which is locally Lipschitz and also satisfies $\eta(0, t) = 0$, but is decreasing and hence is not class $\mathcal{G}$. Now the inequality becomes $u + h \geq 0$, although this condition still prevents an immediate boundary crossing, its dependence on h is reversed: increasing the value of $h(x)$ decreases $\eta(h, t)$ and changes the comparison bound in the opposite direction. Consequently, the ordering of the comparison condition is not consistent with the ordering of the $h(x)$ values. By requiring $\eta(\cdot)$ to be nondecreasing, class $\mathcal{G}$ functions rule out this reversed dependence, which is useful for construction of practical controllers.

Definition 5 (Comparison Invariant). For a control system $x' = \mathbf{f}(x, u)$ with a control admissible set $\mathcal{U} \subseteq \mathbb{R}^m$, and a set $S = \{x \in \mathbb{R}^n : h \succcurlyeq 0\}$, where $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is a scalar function and $\succcurlyeq$ is either $\geq$ or $>$, h is said to be a *comparison invariant* if there exists a comparison function $\eta(\cdot)$ that is locally Lipschitz and class $\mathcal{G}$ such that it satisfies the inequality

$$\forall x \in S. \exists u \in \mathcal{U}. (\mathcal{L}_{\mathbf{f}(x, u)}(h) + \eta(h, t) \geq 0)$$

implying control invariance, essentially $h \succcurlyeq 0$ stays invariant, where $t \in [0, T]$ for some $T > 0$.

A key requirement for comparison invariants is the existence of a comparison function that is not just locally Lipschitz but also of class $\mathcal{G}$. As shown by Counterexample 2, this added regularity ensures that the Lie derivative of the comparison invariant h does not get too negative, while at the same time allowing the solution of the control system to, in a sense, move “forward.” The comparison function $\eta(\cdot)$ being class $\mathcal{G}$ means whenever $h = 0$, the Lie derivative becomes $\dot{h} \geq 0$. Semantically, this invariance argument is corroborated by the comparison principle, which ensures h is lower bounded by the solution of the comparison system $z' = \eta(z)$, whenever $h(0) \geq z(0) \geq 0$. Next we axiomatize scalar comparison invariants.

Theorem 3 (Scalar Comparison Invariants). *The scalar comparison invariants axiom **SCI** and proof rule **sCI** are derivable in dL. Here $\mathcal{U}$ characterizes a compact control input set. ($\succcurlyeq$ is either $\geq$ or $>$.)*

$$\text{SCI} \quad \text{LocLip}(\eta) \wedge \mathcal{G}(\eta) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0) \rightarrow (h \succcurlyeq 0 \rightarrow \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] h \succcurlyeq 0)$$

$$\text{sCI} \quad \frac{\vdash \text{LocLip}(\eta) \wedge \mathcal{G}(\eta) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{h \succcurlyeq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] h \succcurlyeq 0}$$

Proof (Sketch). The axiom **SCI** derives syntactically in dL’s proof calculus, the full proof can be found in the technical report [22, Appendix B.2.1]. The proof utilizes axiom **BDG** to introduce the comparison system $z' = \eta(z, \tau)$, where z is a fresh scalar variable and $\eta(\cdot)$ is a locally Lipschitz class $\mathcal{G}$ function. Then the scalar comparison principle axiom **SCP** finishes the proof. Proof rule **sCI** is also derived syntactically as follows.

$$\frac{\textcolor{brown}{\wedge R}, \textcolor{brown}{dW} \frac{\vdash \text{LocLip}(\eta) \wedge \mathcal{G}(\eta) \quad Q \vdash \exists u(\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{\vdash \text{LocLip}(\eta) \wedge \mathcal{G}(\eta) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u(\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{\textcolor{brown}{SCI} \frac{h \succcurlyeq 0 \vdash \exists u[x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] h \succcurlyeq 0}$$

The proof utilizes axiom **SCI**, followed by **AR** and **dW** resulting in the required premises. $\square$

From Theorem 3, proof rule **sCI** enables proving control invariance by reducing the box modality to two open premises with purely real arithmetic formulas. Importantly, when the comparison function $\eta(\cdot)$ is a polynomial the two resulting premises are formulas of first-order logic over the reals ($\text{FOL}_{\mathbb{R}}$) and are, hence, decidable by *quantifier elimination* [8].

Example 3 (ACC Revisited). Now we take another look at the adaptive cruise control system from Example 2. Recall $\alpha_{\text{acc}} \equiv x' = u, x'_l = v_l, \tau' = 1 \ \& \ Q$ with $Q \equiv v_l \geq 0 \wedge T \geq \tau$. Also recall, $\text{Safe} \equiv x_l - x \geq d_{\min}$, we can encode this safety condition as $h := x_l - x - d_{\min}$. Now we would like to prove control invariance for this system by proving the following **dL** formula

$$h \geq 0 \rightarrow \exists u[\alpha_{\text{acc}}] h \geq 0.$$

From our previous developments we know we can prove such formulas using the scalar comparison invariants proof rule **sCI**:

$$\textcolor{brown}{sCI} \frac{\vdash \text{LocLip}(\eta) \wedge \mathcal{G}(\eta) \quad Q \vdash \exists u(\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{h \geq 0 \vdash \exists u[x' = u, x'_l = v_l, \tau' = 1 \ \& \ Q] h \geq 0}$$

The Lie derivative of h is computed as $\dot{h} = (x_l)' - (x)' - (d_{\min})' = v_l - u$, observe $d_{\min}$ is a constant. Now in order to complete the proof we need to choose a class $\mathcal{G}$ function $\eta(\cdot)$, a sufficient choice is the linear function $\eta(h, \tau) := \lambda h$, for some positive λ . This function is a polynomial, hence locally Lipschitz, and satisfies the conditions of class $\mathcal{G}$ from Definition 4.

$$\frac{\textcolor{brown}{R} \frac{\textcolor{brown}{cut}, \textcolor{brown}{\exists L} \frac{\textcolor{brown}{SCI} \frac{\textcolor{brown}{R} \frac{\lambda > 0 \vdash \text{LocLip}(\lambda h) \wedge \mathcal{G}(\lambda h)}{\vdash \text{LocLip}(\eta) \wedge \mathcal{G}(\eta)} \quad \textcolor{brown}{cut}, \textcolor{brown}{\exists L} \frac{d_{\min} > 0, \lambda > 0, Q \vdash \exists u(\mathcal{U} \wedge \dot{h} + \lambda h \geq 0)}{Q \vdash \exists u(\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}}{h \geq 0 \vdash \exists u[x' = u, x'_l = v_l, \tau' = 1 \ \& \ Q] h \geq 0}}{\textcolor{brown}{SCI} \frac{h \geq 0 \vdash \exists u[x' = u, x'_l = v_l, \tau' = 1 \ \& \ Q] h \geq 0}}$$

Now continuing with the right branch with the unfolded definition of Q , this resulting branch only involves first-order logic formulas over the reals, and hence is decidable [8]. Depending on the specification of the control constraint set $\mathcal{U}$ the branch can be closed by real arithmetic **R**, proving control invariance.

$$\textcolor{brown}{R} \frac{\textcolor{brown}{SCI} \frac{d_{\min} > 0, \lambda > 0, v_l \geq 0, T \geq \tau \geq 0 \vdash \exists u(\mathcal{U} \wedge v_l - u + \lambda h \geq 0)}{d_{\min} > 0, \lambda > 0, v_l \geq 0, T \geq \tau \geq 0 \vdash \exists u(\mathcal{U} \wedge \dot{h} + \lambda h \geq 0)}}{\textcolor{brown}{SCI} \frac{h \geq 0 \vdash \exists u[x' = u, x'_l = v_l, \tau' = 1 \ \& \ Q] h \geq 0}}$$

For the control constraint set $\mathcal{U} \equiv u_{\max} \geq u \wedge u \geq 0$, control invariance of the system can be proven, as $u = 0$ is a valid control input that maintains the safe distance, moreover, the upper bound $u_{\max}$ can be computed from the Lie derivative as $v_l + \lambda h \geq u$.

Now, similar to how the scalar comparison principle was lifted to the vectorial case, scalar comparison invariants can also be lifted to the so-called *vector comparison invariants*.

Definition 6 (Vector Comparison Invariants). For a control system $x' = \mathbf{f}(x, u)$ with a control admissible set $\mathcal{U} \subseteq \mathbb{R}^m$, and a set $S = \{x \in \mathbb{R}^n : \mathbf{h} \succcurlyeq 0\}$, where $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a vector-valued function and $\succcurlyeq$ is either $\geq$ or $>$ with component-wise ordering i.e., $h_i \succcurlyeq 0$ for $i = 1, \dots, n$, $\mathbf{h}$ is said to be a comparison invariant if there exists a vector-valued comparison function $\boldsymbol{\eta}(\cdot)$ that is locally Lipschitz and class $\mathcal{G}$ such that it satisfies the following inequality

$$\forall x \in S. \exists u \in \mathcal{U}. (\mathcal{L}_{\mathbf{f}(x, u)}(\mathbf{h}) + \boldsymbol{\eta}(\mathbf{h}, t) \geq 0)$$

that implies control invariance, hence $\mathbf{h} \succcurlyeq 0$ stays invariant, where $t \in [0, T]$ for some $T > 0$.

As we can see from the above definition, vector comparison invariants lift scalar comparison invariants to n -dimensional vectors with component-wise ordering of the inequalities. Consequently, scalar comparison invariants are a special case of vector comparison invariants when $n = 1$. Next we axiomatize vector comparison invariants for vectorial control invariance proofs.

Theorem 4 (Vector Comparison Invariants). *The vector comparison invariants axiom **VCI** and proof rule **vCI** derive syntactically in **dL**. The compact control input set is represented by the formula $\mathcal{U}$. ($\succcurlyeq$ is either $\geq$ or $>$.)*

$$\begin{array}{l} \text{VCI} \quad \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{G}_{\text{vec}}(\boldsymbol{\eta}) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau) \geq 0) \rightarrow \\ \quad (\mathbf{h} \succcurlyeq 0 \rightarrow \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \mathbf{h} \succcurlyeq 0) \\ \\ \text{vCI} \quad \frac{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{G}_{\text{vec}}(\boldsymbol{\eta}) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau) \geq 0)}{\mathbf{h} \succcurlyeq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \mathbf{h} \succcurlyeq 0} \end{array}$$

Proof (Sketch). Similar to the proofs of the previous axioms, the proof of axiom **VCI** uses axiom **BDG** to introduce the comparison system $z' = \boldsymbol{\eta}(z, \tau)$, where z is a fresh vector of variables, then the proof is finished by axiom **VCP**. Similarly, the proof rule with two open premises is derived as follows. The full proof can be found in the technical report [22, Appendix B.2.1].

$$\begin{array}{l} \text{AR, dW} \quad \frac{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{G}_{\text{vec}}(\boldsymbol{\eta}) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau) \geq 0)}{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{G}_{\text{vec}}(\boldsymbol{\eta}) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau) \geq 0)} \\ \text{vCI} \quad \frac{}{\mathbf{h} \succcurlyeq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \mathbf{h} \succcurlyeq 0} \end{array}$$

The required premises are derived using **VCI**, **AR**, and **dW**. □

With this we successfully axiomatized *scalar* and *vector comparison invariants* in **dL**.

Comparison invariants are an important and general class of invariants, which will be further showcased in their use for axiomatizing control barrier functions next.

4.2 Control Barrier Functions

First introduced by Ames *et al.* [3–5], control barrier functions (CBFs) have risen to popularity in the past decade as a go-to method for synthesizing safe controllers. CBF-based quadratic programming as an optimal control method provides an efficient way for enforcing safety with formal guarantees [3]. However, these formal safety guarantees are only enforced by the synthesized controller if the CBF used to synthesize it was a valid CBF in the first place, and verifying which is a nontrivial task [11].

Definition 7 (Class $\mathcal{K}$ Functions [17]). A function is called an (extended) class $\mathcal{K}$ function if it is zero at zero and strictly increasing. Class $\mathcal{K}$ functions are encoded as the following.

$$\mathcal{K}(\eta) \stackrel{\text{def}}{=} \forall t \forall x \forall y (\eta(0, t) = 0 \wedge (x > y \rightarrow \eta(x, t) > \eta(y, t))) \quad \mathcal{K}_{\text{vec}}(\boldsymbol{\eta}) \stackrel{\text{def}}{=} \bigwedge_{i=1}^n \mathcal{K}(\eta_i)$$

Where $\eta(\cdot)$ and $\boldsymbol{\eta}(\cdot)$ are scalar and vector-valued functions, respectively.

Observe that class $\mathcal{K}$ functions have a stronger requirement of strictly increasing as compared to class $\mathcal{G}$, which are nondecreasing, meaning every class $\mathcal{K}$ function is also class $\mathcal{G}$, but not the other way around.

Definition 8 (Control Barrier Function [5]). For a sublevel set $S = \{x \in \mathbb{R}^n : h(x) \geq 0\}$, where $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable scalar function, $h(x)$ is called a *control barrier function* for the control system $x' = \mathbf{f}(x, u)$ with the control admissible set $\mathcal{U}$ if there exists a class $\mathcal{K}$ function $\gamma(\cdot)$ such that the following inequality holds:

$$\forall x \in S. \exists u \in \mathcal{U}. (\mathcal{L}_{\mathbf{f}(x, u)}(h(x)) + \gamma(h(x)) \geq 0)$$

implying $h(x) \geq 0$ stays invariant along $x' = \mathbf{f}(x, u)$.

Remark 4. Definition 8 gives the standard definition of CBFs as found in the literature [3–5, 11], where the class $\mathcal{K}$ function $\gamma(\cdot)$ is the special time-independent case of Definition 7. In order to reason about CBFs in **dL**, however, we use Definition 7 with the special time-tracking variable τ in order to facilitate proofs, as previously mentioned in Remark 1.

We can already see that CBFs are essentially (scalar) comparison invariants with the added regularity that the comparison function $\eta(\cdot)$ is class $\mathcal{K}$, rather than the more general class $\mathcal{G}$. The key distinction is that class $\mathcal{K}$ functions are strictly increasing, so the allowable system behavior changes predictably with the CBF value. In practice, this gives CBFs additional structure for balancing safety with other control objectives, such as stability, tracking, or performance.

Corollary 3 (Control Barrier Functions). *The following axiom and proof rule derive syntactically in **dL** and can be utilized to prove the validity of scalar control barrier functions. Here $\mathcal{U}$ characterizes a compact set.*

$$\begin{array}{l} \text{CBF} \quad \text{LocLip}(\eta) \wedge \mathcal{K}(\eta) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0) \rightarrow \\ \quad (h \geq 0 \rightarrow \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] h \geq 0) \\ \\ \text{cbf} \quad \frac{\vdash \text{LocLip}(\eta) \wedge \mathcal{K}(\eta) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{h \geq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] h \geq 0} \end{array}$$

Proof (Sketch). Axiom **CBF** is proven syntactically in **dL** using axiom **SCI**. The full proof can be found in the technical report [22, Appendix B.2.2]. The proof rule **cbf** is derived as follows.

$$\begin{array}{l} \text{AR, dW} \quad \frac{\vdash \text{LocLip}(\eta) \wedge \mathcal{K}(\eta) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{\vdash \text{LocLip}(\eta) \wedge \mathcal{K}(\eta) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)} \\ \text{CBF} \quad \frac{\vdash \text{LocLip}(\eta) \wedge \mathcal{K}(\eta) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{h} + \eta(h, \tau) \geq 0)}{h \geq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] h \geq 0} \end{array}$$

The required two open premises derive with axiom **CBF**, followed by **AR** and **dW**. □

The above corollary shows how CBFs can be axiomatized in **dL**, and happen to be a special instance of comparison invariants. The resulting axiom **CBF** and proof rule **cbf** enable proving control invariance using a valid CBF and conversely verify the validity of candidate CBFs.

As per Definition 8, standard CBFs are scalar functions, in many instances, however, these are not enough to enforce safety, especially when multiple safety constraints need to be satisfied at the same time. To this end, the so-called *vector control barrier functions* extend the standard scalar CBFs to vector-valued functions [1, 33]. Similar to scalar CBFs, these vector CBFs enforce safety by ensuring $\mathbf{h}(x) \geq 0$, where the inequality is component-wise i.e., $h_i(x) \geq 0$ for $i = 1, \dots, n$, stays invariant. The following corollary axiomatizes vector CBFs in **dL** as a special instance of vector comparison invariants.

Corollary 4 (Vector Control Barrier Functions). *The following axiom and proof rule can be used to verify vector control barrier functions, and are derivable in **dL**. Here the formula $\mathcal{U}$ characterizes a compact set.*

$$\begin{array}{l} \text{VCBF} \quad \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{K}_{\text{vec}}(\boldsymbol{\eta}) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau) \geq 0) \rightarrow \\ \quad (\mathbf{h} \geq 0 \rightarrow \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \mathbf{h} \geq 0) \\ \\ \text{vcbf} \quad \frac{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{K}_{\text{vec}}(\boldsymbol{\eta}) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau) \geq 0)}{\mathbf{h} \geq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \mathbf{h} \geq 0} \end{array}$$

Proof (Sketch). Both the axiom **VCBF** and proof rule **vcbf** are derived syntactically in **dL**. In particular, the proof of axiom **VCBF** utilizes the vector comparison invariants axiom **VCI**. The complete proof can be found in the technical report [22, Appendix B.2.2]. The proof rule derives from axiom **VCBF** as follows.

$$\begin{array}{l} \text{AR, dW} \quad \frac{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{K}_{\text{vec}}(\boldsymbol{\eta}) \quad Q \vdash \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau))}{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{K}_{\text{vec}}(\boldsymbol{\eta}) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau))} \\ \text{VCBF} \quad \frac{\vdash \text{LocLip}(\boldsymbol{\eta}) \wedge \mathcal{K}_{\text{vec}}(\boldsymbol{\eta}) \wedge [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \exists u (\mathcal{U} \wedge \dot{\mathbf{h}} + \boldsymbol{\eta}(\mathbf{h}, \tau))}{\mathbf{h} \geq 0 \vdash \exists u [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q] \mathbf{h} \geq 0} \end{array}$$

□

This axiomatization shows one can prove *control invariance* properties in **dL** using a valid control barrier function with the right choice of comparison function. Consequently, it is also possible to verify a candidate control barrier function in **dL** for a particular control system. In the next section we will see more special instances of comparison invariants.

5 Special Instances

We have already seen in the previous section how the axiomatization of control barrier functions, both scalar and vector, was made possible using comparison invariants. This section gives some more special cases of comparison invariants including *Darboux invariants* and *differential invariants*, further bolstering their versatility.

5.1 Darboux Invariants

Here we look at an important class of invariants known as *Darboux invariants* as they are inspired by Darboux polynomials [31, Section 3]. At a high-level any polynomial p that satisfies the polynomial inequality $\dot{p} \geq gp$ for some cofactor polynomial g ensures $p \geq 0$ stays invariant.

Semantically this invariance property is ensured by Grönwall's lemma [31, 39, §29.VI]. It will be shown that the same invariance observation can be drawn using the comparison principle, specifically, comparison invariants. Moreover, the two existing dL axioms **DBX**_≥ and **VDBX** internalizing Darboux invariants are derived syntactically using comparison invariants.

Corollary 5 (Scalar Darboux [31, Lemma 3.1]). *The existing dL axiom **DBX**_≥ for scalar Darboux invariants is derivable using axiom **SCI**. (≥ is either ≥ or >.)*

$$\text{DBX}_{\geq} \quad [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q](p)' \geq gp \rightarrow (p \succ 0 \rightarrow \exists u[x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q]p \succ 0)$$

Proof (Sketch). The axiom is derived when the comparison function is defined as $\eta(p, \tau) = g \cdot p$ for some cofactor polynomial $g = g(x, \tau)$. The full derivation can be found in the technical report [22, Appendix B.3]. $\square$

The corollary above showed the derivation of scalar Darboux invariants using scalar comparison invariants. Similarly, vectorial Darboux invariants can also be derived using vector comparison invariants as shown in the following corollary.

Corollary 6 (Vectorial Darboux [31, Lemma 4.1]). *The existing dL axiom **VDBX** for vectorial Darboux invariants is derivable using axiom **VCI**.*

$$\text{VDBX} \quad [x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q](\mathbf{p})' = G\mathbf{p} \rightarrow (\mathbf{p} = 0 \rightarrow \exists u[x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q]\mathbf{p} = 0)$$

Proof (Sketch). The axiom is derived by first rewriting $\mathbf{p} = 0 \leftrightarrow -\mathbf{p} \geq 0 \wedge \mathbf{p} \geq 0$, after which axiom **VCI** is applied with comparison functions, $\boldsymbol{\eta}(\mathbf{p}, \tau) = G \cdot \mathbf{p}$. Where G is an essentially positive matrix or *Metzler matrix* [36, 39]. The full proof can be found in the technical report [22, Appendix B.3]. $\square$

As we saw from Corollary 5 and Corollary 6, comparison invariants generalize Darboux invariants. Next we will see another special instance of comparison invariants, namely, differential invariants.

5.2 Differential Invariants

Another simple consequence of (scalar) comparison invariants are differential invariants. Differential invariants relate invariance of scalar inequalities of the form $p \succ 0$ to their Lie derivative inequality $\dot{p} \geq 0$ over the entire domain Q . The following shows the derivation of the differential invariants proof rule **dI**_≥ tailored to control systems.

Corollary 7 (Differential Invariants [28]). *The existing differential invariants proof rule **dI**_≥ is a special instance of comparison invariants and is derivable using rule **sCI**. (≥ is either ≥ or >.)*

$$\text{dI}_{\geq} \quad \frac{Q \vdash \dot{p} \geq 0}{p \succ 0 \vdash \exists u[x' = \mathbf{f}(x, u), \tau' = 1 \ \& \ Q]p \succ 0}$$

Proof (Sketch). The proof is straightforward when the comparison function is the zero function, i.e., $\eta(p, \tau) = 0 \cdot p$. The full derivation can be found in the technical report [22, Appendix B.3]. $\square$

These special instances showcase the versatility of comparison invariants as the central unifying idea behind various proof techniques for invariance verification. It also shows that the chosen comparison systems in the comparison principle axioms **SCP** and **VCP** are very general, and can relate various invariance properties with the right choice of the scalar and vector comparison functions $\eta(\cdot)$ and $\boldsymbol{\eta}(\cdot)$, respectively.

6 Related Work

Invariance in Dynamical Systems. Checking whether a given set is invariant under the flow of an ODE has been a central interest in the field of dynamical systems [7, 19, 34, 39], going all the way back to the theorem of Nagumo, which gives conditions for proving (positive) invariance of closed sets [7, 19]. The comparison principle, a prominent technique for relating solutions of two differential equations using differential inequalities, provides an alternative method for proving invariance of both open and closed sets [39]. More (relatively) recently barrier certificates, first introduced by Prajna *et al.* [32], gained popularity as a Lyapunov-like technique for verifying invariance [21]. Vector barrier certificates introduced by Sogokon *et al.* [36], generalizing scalar barrier certificates, primarily utilized the comparison principle as the basis.

Control Invariance. A similar question of whether a set is invariant under the flow of a control system has also been a central concern in control theory [7, 10, 34]. In the field of reachability analysis, geometric set representations like zonotopes and ellipsoids have been utilized to characterize control invariant sets [7, 35]. First introduced by Ames *et al.* [4], control barrier functions have gained immense popularity in the past decade for being an efficient means for synthesizing safety enforcing controllers [3, 5, 40]. Recently, Clark [11] gave a semialgebraic framework using semidefinite programming for verifying control barrier functions.

Deductive Verification. Unlike the numerical methods [7, 11, 35], deductive verification methods utilize a symbolic and logical approach to proving invariance of dynamical systems. To this end Taly *et al.* [37], and Ghorbal *et al.* [13] have shown ODE invariance can be proven using a hierarchy of sound but incomplete proof rules. In the logic dL , completeness of algebraic and semialgebraic invariance has been shown by Platzer and Tan [30], with later extensions to Noetherian functions [31]. Synthesis of control invariants, specifically control envelopes, have been successfully undertaken by Kabra *et al.* [15], using differential game logic (dGL). Recently Hellwig *et al.* [14], showed the verification of zonotope-based control envelopes in dL . Control invariance has also been studied by Platzer [27] from the lens of differential hybrid games.

7 Conclusion

In this paper we presented an axiomatization for verifying control invariance in differential dynamic logic (dL). This was made possible by first axiomatizing the *scalar* and *vector comparison principles*, which served as the central proof principles underlying our proofs. We introduced *scalar* and *vector comparison invariants* leading to a dedicated set of axioms and proof rules that reduce proving control invariance to proving a functional inequality on its Lie derivative. Finally, we axiomatized control barrier functions, which along with Darboux and differential invariants, were shown to be special instances of comparison invariants.

For future work we will be implementing the presented axioms and proof rules in the theorem prover KeYmaera X [12]. Since all the presented axioms and proof rules were derived syntactically in dL , they can be implemented as tactics on top of a soundness-critical prover kernel [12]. We will further be looking at deriving completeness results for proving control invariance in dL .

Acknowledgements. We thank the anonymous reviewers for their helpful feedback. We also thank Adrian Kulmburg for his feedback on an earlier version of this work including his suggestion of utilizing the fact that locally Lipschitz functions are bounded on a compact domain in the proofs. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263.

References

- [1] Ahmed Allibhoy and Jorge Cortés. Control-barrier-function-based design of gradient flows for constrained nonlinear programming. *IEEE Transactions on Automatic Control*, 69(6):3499–3514, 2023. doi:10.1109/TAC.2023.3306492.
- [2] Rajeev Alur. *Principles of cyber-physical systems*. MIT press, 2015. doi:10.1017/9781107588981.
- [3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. Ieee, 2019. doi:10.23919/ECC.2019.8796030.
- [4] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE conference on decision and control*, pages 6271–6278. IEEE, 2014. doi:10.1109/CDC.2014.7040372.
- [5] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016. doi:10.1109/TAC.2016.2638961.
- [6] Richard Bellman. Vector lyapunov functions. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(1):32–34, 1962. doi:10.1137/0301003.
- [7] Franco Blanchini, Stefano Miani, et al. *Set-theoretic methods in control*, volume 78. Springer, 2008. doi:10.1007/978-3-319-17933-9.
- [8] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real algebraic geometry*, volume 36. Springer Science & Business Media, 2013. doi:10.1007/978-3-662-03718-8.
- [9] Debasish Chatterjee and Daniel Liberzon. Stability analysis of deterministic and stochastic switched systems via a comparison principle and multiple lyapunov functions. *SIAM Journal on Control and Optimization*, 45(1):174–206, 2006. doi:10.1137/040619429.
- [10] Carmen Chicone. *Ordinary differential equations with applications*. Springer, 2006. doi:10.1007/978-3-031-51652-8.
- [11] Andrew Clark. A semi-algebraic framework for verification and synthesis of control barrier functions. *IEEE Transactions on Automatic Control*, 2024. doi:10.1109/TAC.2024.3497001.
- [12] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völz, and André Platzer. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In Amy P. Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 527–538. Springer, 2015. doi:10.1007/978-3-319-21401-6_36.
- [13] Khalil Ghorbal, Andrew Sogokon, and André Platzer. A hierarchy of proof rules for checking positive invariance of algebraic and semi-algebraic sets. *Computer Languages, Systems and Structures*, 47(1):19–43, 2017. doi:10.1016/j.cl.2015.11.003.
- [14] Jonathan Hellwig, Lukas Schäfer, Long Qian, André Platzer, and Matthias Althoff. From zonotopes to proof certificates: A formal pipeline for safe control envelopes. In Ferruccio Damiani and Marie Farrell, editors, *iFM*, volume 16194 of *LNCS*, pages 248–267. Springer, 2026. doi:10.1007/978-3-032-10794-7_13.
- [15] Aditi Kabra, Jonathan Laurent, Stefan Mitsch, and André Platzer. CESAR: Control envelope synthesis via angelic refinements. In Bernd Finkbeiner and Laura Kovács, editors, *TACAS*, volume 14570 of *LNCS*, pages 144–164. Springer, 2024. doi:10.1007/978-3-031-57246-3_9.
- [16] Bernhard Kawohl and Nikolay Kutev. Comparison principle and Lipschitz regularity for viscosity solutions of some classes of nonlinear partial differential equations. *FUNKCIALAJ EKVACIOJ SERIO INTERNACIA*, 43(2):241–254, 2000.
- [17] Christopher M Kellett. A compendium of comparison function results. *Mathematics of Control, Signals, and Systems*, 26(3):339–374, 2014. doi:10.1007/s00498-014-0128-8.
- [18] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall, Upper Saddle River, NJ, 2002.
- [19] Thomas Mejstrik. Some remarks on Nagumo’s theorem. *Czechoslovak mathematical journal*,

- 62(1):235–242, 2012. doi:10.1007/s10587-012-0008-7.
- [20] Sergey G Nersesov and Wassim M Haddad. On the stability and control of nonlinear dynamical systems via vector Lyapunov functions. *IEEE Transactions on automatic control*, 51(2):203–215, 2006. doi:10.1109/TAC.2005.863496.
 - [21] Promit Panja and André Platzer. Correct-by-construction barrier certificate synthesis for safety verification of continuous dynamical systems. In *64th IEEE Conference on Decision and Control, CDC 2023, Rio de Janeiro, Brazil, December 9-12, 2023*, pages 7701–7707. IEEE, 2023. doi:10.1109/CDC57313.2023.11312663.
 - [22] Promit Panja and André Platzer. Comparison invariants for verifying control invariance. *arXiv preprint arXiv:2608.24598*, 2026.
 - [23] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008. doi:10.1007/s10817-008-9103-8.
 - [24] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, Heidelberg, 2010. doi:10.1007/978-3-642-14509-4.
 - [25] André Platzer. The complete proof theory of hybrid systems. In *LICS*, pages 541–550. IEEE, 2012. doi:10.1109/LICS.2012.64.
 - [26] André Platzer. A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reas.*, 59(2):219–265, 2017. doi:10.1007/s10817-016-9385-1.
 - [27] André Platzer. Differential hybrid games. *ACM Trans. Comput. Log.*, 18(3):19:1–19:44, 2017. doi:10.1145/3091123.
 - [28] André Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer, Cham, 2018. doi:10.1007/978-3-319-63588-0.
 - [29] André Platzer and Long Qian. Axiomatization of compact initial value problems: Open properties. *J. ACM*, 72(6):1–51, 2025. doi:10.1145/3763228.
 - [30] André Platzer and Yong Kiam Tan. Differential equation axiomatization: The impressive power of differential ghosts. In Anuj Dawar and Erich Grädel, editors, *LICS*, pages 819–828, New York, 2018. ACM. doi:10.1145/3209108.3209147.
 - [31] André Platzer and Yong Kiam Tan. Differential equation invariance axiomatization. *J. ACM*, 67(1):6:1–6:66, 2020. doi:10.1145/3380825.
 - [32] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *International workshop on hybrid systems: Computation and control*, pages 477–492. Springer, 2004. doi:10.1007/978-3-540-24743-2_32.
 - [33] Wei Ren, Jingjie Li, Junlin Xiong, and Xi-Ming Sun. Vector control Lyapunov and barrier functions for safe stabilization of interconnected systems. *SIAM Journal on Control and Optimization*, 61(5):3209–3233, 2023. doi:10.1137/22m1530422.
 - [34] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013. doi:10.1007/978-1-4757-3108-8.
 - [35] Lukas Schäfer, Felix Gruber, and Matthias Althoff. Scalable computation of robust control invariant sets of nonlinear systems. *IEEE Transactions on Automatic Control*, 69(2):755–770, 2023. doi:10.1109/TAC.2023.3275305.
 - [36] Andrew Sogokon, Khalil Ghorbal, Yong Kiam Tan, and André Platzer. Vector barrier certificates and comparison systems. In Klaus Havelund, Bill Roscoe, and Jan Peleska, editors, *FM*, volume 10951 of *LNCS*, pages 418–437. Springer, 2018. doi:10.1007/978-3-319-95582-7_25.
 - [37] Ankur Taly and Ashish Tiwari. Deductive verification of continuous dynamical systems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (2009)*, pages 383–394. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPIcs.FSTTCS.2009.2334.
 - [38] Yong Kiam Tan and André Platzer. An axiomatic approach to existence and liveness for differential equations. *Formal Aspects Comput.*, 33(4):461–518, 2021. Special issue for selected papers from

- FM'19. doi:10.1007/s00165-020-00525-0.
- [39] Wolfgang Walter. *Ordinary differential equations*, volume 182. Springer Science & Business Media, 2013.
- [40] Wei Xiao and Calin Belta. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2021. doi:10.1109/TAC.2021.3105491.