From Zonotopes to Proof Certificates: A Formal Pipeline for Safe Control Envelopes

Jonathan Hellwig[1] (✉), Lukas Schäfer[2], Long Qian[3], André Platzer[1], and Matthias Althoff[2]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
{jonathan.hellwig, platzer}@kit.edu
[2] Technical University of Munich, Garching, Germany
{lukas.schaefer, althoff}@tum.de
[3] Carnegie Mellon University, Pittsburgh, United States of America
longq@andrew.cmu.edu

Abstract. Synthesizing controllers that enforce both safety and actuator constraints is a central challenge in the design of cyber-physical systems. While zonotope-based reachability methods deliver impressive scalability, only parts of these methods have been formalized. Consequently, no practical tool provides a fully verified end-to-end pipeline, leaving an assurance gap for safety-critical systems. Deductive verification with the hybrid system prover KeYmaera X could, in principle, resolve this assurance gap, but the high-dimensional set representations required for reachability analysis overwhelm its reasoning based on quantifier elimination. To close this gap, we develop a verification pipeline that combines scalability with formal rigor by computing control-invariant sets using high-performance reachability algorithms and verifying them using novel logical proof rules. Computationally intensive zonotope containment tasks are offloaded to efficient numerical backends, which return compact witnesses that KeYmaera X can validate rapidly. We show the practical utility of our approach through representative case studies.

Keywords: deductive verification, reachability analysis, zonotopes, robust control invariant sets, differential dynamic logic

1 Introduction

Autonomous vehicles, aircraft, and other cyber-physical systems increasingly demand advanced control algorithms while never violating stringent safety specifications. Control-envelope synthesis has become a central problem at the intersection of formal verification and control theory [8,30]. Rather than synthesizing a single monolithic controller, in control-envelope synthesis, the goal is to compute a set of control inputs whose execution traces satisfy a formal safety specification. As a result, control envelopes enable decoupling safety from performance: the precomputed control envelope guarantees that *every* admissible input meets the specification, while at runtime an optimization routine can freely

F. Damiani and M. Farrell (Eds.): iFM 2025, LNCS 16194, pp. 248–267, 2026.
DOI: 10.1007/978-3-032-10794-7_13

sample concrete control signals with respect to a secondary performance objective. A promising recent line of work [47] frames control envelope synthesis as the computation of the maximal *robust control invariant* sets (RCI) [10,45,41]—the set of states from which at each sampling instant, there exists a control action that keeps the system in it — thereby guaranteeing safety by construction. Building on this insight, the authors rely on efficient over-approximations of reachable sets as the main computational workhorse. The efficiency of their computations is achieved by symbolic set representations, e.g., zonotopes [4], support functions [2,54], or ellipsoids, that are either closed or can be tightly over-approximated under Minkowski addition and affine transformations, realizing an efficient computation of the reachable set [3]. However, despite their scalability, this reachability-based approach raises two concerns in safety-critical fields:

1. Floating-point uncertainty: Current reachability tools perform every operation with finite precision, so the end result does not have any end-to-end guarantee that rounding errors have not led to a violation of the safety specification.
2. Formally unverified implementations: The highly-tuned numerical kernels are optimized for speed, not transparency. Proving the correctness of their implementations is prohibitively expensive.

Mission-critical control systems — like aircraft, autonomous robots, or self-driving cars — would benefit greatly from numerical pipelines that provide end-to-end correctness guarantees. One way to obtain such rigorous guarantees is to use a theorem prover like KeYmaera X [22] that implements differential dynamics logic (dL) [38,39], a specialized logic designed for specification and deductive verification of cyber-physical systems. Unlike numerical reachability tools [5,18,21], which aim to provide fast but conservative over-approximations of reachable sets, a theorem prover like KeYmaera X works with a symbolic proof calculus whose goal is to construct fully machine-checkable proofs that a control system satisfies its specification.

The key difficulty in combining techniques lies in the fundamental difference of formalisms of dL and classical reachability analysis. In reachability analysis, one works with highly specialized set representations that admit efficient "push-button" computations, but at the price of modelling restrictions. In dL, specifications are expressed as fully general semi-analytic formulas, resulting in a much richer specification language. However, verification in this expressive framework typically demands substantial interactive proof effort for complicated systems. As a result, tasks that are easy on one side, such as scalable numerical over-approximations, can be arduous on the other, such as manual proof construction, and vice versa. Indeed, the theorem-proving and the reachability analysis research fields have been developed largely in parallel, with little cross-pollination.

Thus, our goal with this paper is to build a bridge between the two research fields. Specifically, we formalize the control-envelope synthesis approach using RCIs in dL. By doing so, we enable end-to-end correctness guarantees for an envelope computed by an independent numerical reachability tool. This integration of numerical reachability and deductive verification achieves something neither approach could accomplish alone: efficient numerical computation paired with fully rigorous, machine-checked correctness. During formalization, we encountered the following challenges:

1. *Linking safety specifications to RCIs.* Although RCIs are commonly used as terminal constraints in model predictive control [12,49], no formal dL proof has yet shown that an RCI necessarily satisfies a given safety specification.
2. *Treatment of continuous dynamics.* The algorithms utilized by reachability tools are inherently delicate due to their numerical nature, therefore difficult to implement directly in dL. Whilst the completeness of dL [40] essentially guarantees that all true numerical properties can be deductively proven in principle, more efficient methods are desired for practical problems.
3. *Scalable verification of set containment.* Rigorous proofs of set containment — a central step in reachability analysis methods — involve large arithmetic formulas that are often intractable to verify without specialized methods. KeYmaera X uses a general decision procedure for real-arithmetic formulas, which does not scale to large verification problems.

Our Contribution is to build a formal link between reachability analysis and deductive verification by introducing a dL-based verification pipeline for control-envelopes addressing all three challenges. Specifically, we provide a syntactic derivation showing that if a control envelope is a robust control invariant set, then it automatically satisfies the dL specification. In addition, we leverage Taylor Models in dL, which is general enough to validate the invariant sets synthesized by numerical methods while also having rigorous error bounds that are deductively proven in dL. Finally, to overcome the scalability bottleneck in set containment proofs, we focus on zonotopic control envelopes and extend a known witness theorem for zonotope containment. This enables fast floating-point search for a candidate witness followed by a lightweight certification in KeYmaera X without relying on the slow, general-purpose decision procedure.

2 Preliminaries

This section first recalls the control envelope synthesis problem and then distills the basic principles of differential dynamic logic that support our verification approach. Let $\mathcal{X}_0 \subseteq \mathbb{R}^n$ denote the set of initial states, $\mathcal{X} \subseteq \mathbb{R}^n$ the set of admissible states and $\mathcal{U} \subseteq \mathbb{R}^m$ the set of admissible control inputs. For a *sampling period* $\Delta t > 0$ the *sampled-data system* is given by the ordinary differential equation (ODE)

$$x'(t) = f(x(t), u_{\lfloor t/\Delta t \rfloor}), \quad x_0 \in \mathcal{X}_0, \tag{1}$$

where the control input is zero-order-hold: for each integer $k \geq 0$ we sample at $t_k = k\Delta t$ a feedback law $\mu : \mathbb{R}^n \to \mathbb{R}^m$ generates $u_k := \mu(x(t_k))$, which is held constant on $[t_k, t_{k+1}]$. In the classical control, problem we seek to find one concrete feedback law $\mu : \mathbb{R}^n \to \mathbb{R}^m$ such that every control input is admissible,

$$\mu(x(k\Delta t)) \in \mathcal{U}, \tag{2}$$

and every trajectory of the closed-loop system remains in the admissible set of states,

$$x(t) \in \mathcal{X} \tag{3}$$

for all $t \geq 0$. The control envelope problem lifts this problem from finding a single control law to a family of laws. Concretely, we seek to construct a relation $\mathcal{E} \subseteq \mathbb{R}^n \times \mathbb{R}^m$ between states and control inputs such that *every* feedback law whose sampled control inputs stay within the envelope, automatically satisfy the admissibility and safety specifications.

In reachability analysis, one over-approximates ODE trajectories by a set-valued abstraction, a perspective that yields efficient computational properties. This viewpoint naturally leads to the following notion of reachable sets.

Definition 1 (Reachable set). *Let $\Delta t > 0$ be the sampling period and let $\mathcal{E} \subseteq \mathbb{R}^n \times \mathbb{R}^m$ be a control envelope: $\mathcal{E}_x := \{u \in \mathbb{R}^m \mid (x, u) \in \mathcal{E}\}$ denotes the set of control outputs at $x \in \mathbb{R}^n$. Given an initial set $\mathcal{X}_0$, the* reachable set *at time $t \in [0, \Delta t]$ is the set of states*

$$\mathcal{R}(t, \mathcal{X}_0, \mathcal{E}) := \left\{ x \in \mathbb{R}^n \mid \exists x_0 \in \mathcal{X}_0 \exists u \in \mathcal{E}_{x_0} : x = x_0 + \int_0^t f(x(s), u) ds \right\}.$$

The reachable set over the time interval $[0, t]$ is the union of reachable sets, i.e.,

$$\mathcal{R}([0, t], \mathcal{X}_0, \mathcal{E}) = \bigcup_{s \in [0,t]} \mathcal{R}(s, \mathcal{X}_0, \mathcal{E}).$$

The difficulty with reachable-set computations is that we can only evaluate them over finite time horizons. How can we verify that a proposed control envelope $\mathcal{E}$ satisfies the safety property (3) for all $t \geq 0$? The key is to use an inductive argument: leverage finite-horizon reachable-set computations to establish an invariant that guarantees safety over an infinite time horizon. This leads us to the definition of robust control invariants.

Definition 2 (Robust control invariant set [47]). *A set $\mathcal{S} \subset \mathbb{R}^n$ is called a* robust control invariant set *if there exists a control envelope $\mathcal{E} \subset \mathbb{R}^n \times \mathbb{R}^m$ such that*

1. One-step invariance*: $\mathcal{R}(\Delta t, \mathcal{S}, \mathcal{E}) \subseteq \mathcal{S}$,*
2. One-step safety*: $\mathcal{R}([0, \Delta t], \mathcal{S}, \mathcal{E}) \subseteq \mathcal{X}$,*
3. Control-admissibility*: $\forall x_0 \in \mathcal{S} : \mathcal{E}_{x_0} \subseteq \mathcal{U}$.*

The central claim is that the existence of a robust control invariant $\mathcal{S}$ and its associated control envelope $\mathcal{E}$, implies the safety property (3) for all $t \geq 0$. We prove this claim formally in Sec. 3.

2.1 Differential Dynamic Logic

To formally verify a control envelope, we must first introduce differential dynamic logic: its hybrid-program modeling language, its formula specification language, and the proof calculus that enables deductive verification. For a more detailed introduction the reader is referred to the literature [38,39].

Hybrid Programs. The language of hybrid programs is generated by the following grammar, where x is a variable, e is a dL term, Q is a formula of first-order real arithmetic:

$$\alpha, \beta ::= x := e \mid x := * \mid x' = f(x) \ \& \ Q \mid ?P \mid \alpha; \beta \mid \alpha \cup \beta \mid \alpha^*.$$

Continuous dynamics are modeled by $x' = f(x) \ \& \ Q$ evolving in the domain Q. Discrete dynamics are modeled by *assignments* $x := e$, which instantaneously assign the term e to x and *tests* $?Q$, which check whether the formula Q is true in the current state. The *nondeterministic assignment* $x := *$ assigns an arbitrary value to x. Throughout this paper we always use such assignments in conjunction with a test $?Z(x)$, to model non-deterministic assignment restricted to a formula $Z(x)$. To combine the discrete and continuous fragments, there are three program combinators: *sequential composition* $\alpha; \beta$, which first runs α then β; *nondeterministic choice* $\alpha \cup \beta$, which runs either α or β; and finally *nondeterministic repetitions* α^*, which repeats α an arbitrary number of times.

Formulas. The formulas of dL are defined by the following grammar where e, g are terms, P, Q are formulas, x is a variable, and α is a hybrid program:

$$P, Q ::= e \leq g \mid \neg P \mid P \land Q \mid P \rightarrow Q \mid \forall x P \mid [\alpha] P.$$

A dL formula combines first-order arithmetic with model operators that refers to program behavior. Atomic formulas are inequalities $e \leq g$ between real-valued terms. These atomic formulas are composed with Boolean connectives such as $\neg$, $\land$ and $\rightarrow$, together with the first-order quantifier $\forall$. Connectives such as $\lor$ and the quantifier $\exists$ are definable from these primitives. The only distinctive construct is the *box modality* $[\alpha]P$, which asserts that after every execution of the hybrid program α the post-condition P holds.

2.2 Deductive Verification

Formula verification in dL is carried out within a sequent-calculus framework built on sound axioms and inference rules. We write $\Gamma \vdash \Delta$ if the formula Δ is provable from the assumption Γ in the dL proof calculus. The calculus includes propositional rules such as

$$\rightarrow\mathrm{R} \ \frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \rightarrow Q, \Delta} \qquad\qquad \land\mathrm{R} \ \frac{\Gamma \vdash P, \Delta \qquad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \land Q, \Delta}$$

The $\rightarrow$R rule decomposes the implication $P \rightarrow Q$ by adding P in the list of assumptions, while the $\land$R rule splits the conjunctive formula $P \land Q$ into two separate proof goals. Similarly, there are rules that decompose every other logical construct. These inference rules are applied bottom-up, but provability is read top-down: if the premises at the top of each rule are provable, then the conclusion is provable as well.

Box modalities of hybrid program are treated with an axiomatic proof calculus that strips away the program structure step by step, leaving simpler proof goals. A few representative axioms illustrate this:

$$[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P \qquad [\,;]$$

$$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P \qquad [\cup]$$

The $[\,;]$ axiom unfolds the sequential composition into successive modalities, while the $[\cup]$ axiom reduces the nondeterministic choice $\alpha \cup \beta$ to a conjunction over two branches.

By combining the inference rules with the axioms, we can derive new theorems, and every step of the resulting derivation can be mechanically verified by a proof checker. Transformation steps on logical connectives and box modalities are applied until the renaming goal is purely arithmetic. Then, we can invoke a trusted decision procedure, such as quantifier elimination [19]. In the example, we first apply the propositional rule $\to$R, then the $[\,;]$ axiom. After a few further transformation steps we reach purely arithmetic formulas $Q_1(x), \dots, Q_n(x)$, at which point we hand the proof obligations to the trusted decision procedure, finishing the proof.

$$\textbf{Deduction}\uparrow\qquad \to\!\mathrm{R}\,\dfrac{[\,;]\,\dfrac{\dfrac{\dfrac{\vdash Q_1(x)\qquad\dots\qquad\vdash Q_n(x)}{\vdots}}{\Gamma\vdash[\alpha][\beta]\varphi}}{\Gamma\vdash[\alpha;\beta]\varphi}}{\vdash\Gamma\to[\alpha;\beta]\varphi}$$

3 Control Envelope Verification Framework

In this section, we first formalize the control *envelope* synthesis problem from Sec. 2 in dL. To do so, we first model sampled-data systems (1) as a hybrid program. We can then state the safety property (3) formally in dL. Once these components are formalized, we present several key theorems which, taken together, yield a deductive proof of the safety property. This allows us to take a control envelope produced numerically and rigorously verify its adherence to the specification.

3.1 Control Envelope Synthesis Problem in dL

We adopt the following conventions: for a function symbol h, let $\partial_t h$ denote syntactic partial derivative with respect to t, and we use ∇_x to denote the syntactic partial gradient with respect to x. We use $\|\cdot\|_\infty$ to denote the infinity norm of $\mathbb{R}^n$. In Sec. 2, we used a calligraphic font to denote semantic sets, e.g. $\mathcal{E}$

for the control envelope, $\mathcal{S}$ for the robust control invariant set, $\mathcal{X}_0$ for the set of initial states, $\mathcal{X}$ for the safety set and $\mathcal{U}$ for the control constraint set. Hereafter, we write the corresponding uppercase predicates, e.g. $E(x, u)$, $S(x, u)$, $X_0(x)$, $X(x)$ and $U(u)$, to denote their syntactic formula counterparts in dL.

We now formulate sampled-data systems. Let x be the state vector, u the control input, and $E(x, u)$ a control envelope formula, and $\Delta t > 0$ the sampling period. We define the *initialization*, *controller* and *plant* components as follows:

$$\begin{aligned}\text{init} &\equiv u := *; ?\exists x E(x, u),\\ \text{ctrl} &\equiv (?t = \Delta t; u := *; ?E(x, u); t := 0\ ) \cup (?t \neq \Delta t),\\ \text{plant} &\equiv \{x' = f(x, u), t' = 1\ \&\ t \leq \Delta t\}.\end{aligned}$$

The controller nondeterministically chooses between two branches. If the current time t has not yet reached the sampling period Δt, it does nothing. Once the sampling instant is reached, a new control action is chosen nondeterministically from within the control envelope E based on the current state x. Note that this controller is an abstraction of any concrete implementation. By isolating only the aspects relevant to safety verification, we simplify reasoning in dL. Any actual controller would, of course, adhere to additional performance criteria. The plant is modeled by the differential equation $x' = f(x, u), t' = 1$ subject to the constraint $t \leq \Delta t$ to ensure a duration of evolution of at most Δt. In dL, differential equations are non-deterministic: the continuous evolution may stop at any state that still satisfies the constraint $t \leq \Delta t$. The dL *sampled-data system* for one sampling period is

$$\text{sys} \equiv \text{ctrl}; \text{plant}\,.$$

The corresponding dL closed-loop system is the nondeterministic repetition of this program sys*. Thus, the dL *control envelope synthesis* problem is to determine an envelope predicate $E(x, u)$ such that the *control-admissibility formula*

$$\exists x E(x, u) \rightarrow U(u), \tag{4}$$

and the *closed-loop safety formula*

$$X_0(x) \wedge t = 0 \rightarrow [\text{init}; \text{sys}^*] X(x) \tag{5}$$

are valid, i.e., true in all states. These two formulas are the formalized counterparts to (2) and (3).

3.2 Deductive Verification of Control Envelopes

Having posed the control envelope synthesis problem in dL, we can take an envelope computed by an unverified numerical tool, formalize its specification in dL, and certify its correctness through an independent verification process. When carrying out deductive verification, several key considerations arise: The control-admissibility formula (4) is purely a first-order statement over the reals, so it can be proved without invoking any of dL's hybrid system axioms. In contrast, the system-safety formula (5) is more challenging: it combines discrete and continuous dynamics, a nondeterministic repetition, and may contain a high-dimensional control envelope. To tackle this challenge for generic sampled-data system, we present a systematic verification strategy. Specifically, we present a sequence of theorems that, when composed, yield a rigorous proof of the desired safety property. The following five theorems outline the high-level structure of our argument:

1. Theorem 3.1: Relates closed-loop safety (5) to the invariance and safety properties of robust control invariant sets.
2. Theorem 3.2: Connects finite-horizon box modalities for continuous systems to Taylor models.
3. Theorem 3.3: A special case of Theorem 3.2 that establishes the RCI invariance condition for zonotopic control envelopes.
4. Theorem 3.4: A special case of Theorem 3.2 used to establish the RCI safety condition for zonotopic control envelopes.
5. Theorem 3.5: Connects the zonotope-containment problem to an efficiently implementable witness-checking problem.

Detailed derivations and proofs of soundness for these theorems are provided in [28].

The first theorem shows that, in dL, the invariance and safety property of robust control invariants imply the closed-loop safety property (5):

Theorem 3.1. *The following is a derived proof rule of* dL*:*

$$\frac{\begin{array}{l} E(x,u), t=0 \vdash [\text{plant}]X(x) \\ E(x,u), t=0 \vdash [\text{plant}](t=\varDelta t \rightarrow \exists u E(x,u)) \end{array}}{X_0(x), t=0 \vdash [\text{init}; \text{sys}^*]X(x)}$$

Informally, the bottom premise states that whenever an initial control-state pair satisfies $E(x,u)$, then, after one continuous evolution lasting $\varDelta t$, we can choose a new control action to remain in $E(x,u)$. The top premise encodes the safety property: if the start in $E(x,u)$, then we always remain in the safety constraint $X(x)$ after running the plant. The safety property is enforced continuously; not just at the sampling points. Taken together, these two properties entail the safety of the closed-loop system for an arbitrary number of execution steps. The premises are much easier to discharge, because they only involve ordinary differential equations. The two main challenges in proving the premises in Theorem 3.1 are

1. constructing rigorous reachable set over-approximations,
2. efficiently discharging the resulting proof obligations, which involve large arithmetic formulas.

We address the first challenge by introducing generalized Taylor models and showing how they can be derived in dL. Before introducing Taylor models, we briefly review the necessary background Picard iteration and interval arithmetic.

Picard iteration Picard iteration is a classical technique that successively constructs polynomial approximations to the solutions of ordinary differential equations. They can be carried out to arbitrarily high order, resulting approximations of whatever precision is required. Let h be a function symbol and let λ be a variable vector such that $\|\lambda\|_\infty \leq 1$. Further, let $x' = f(x)$ be an ODE system with parameter-dependet initial value $x_0 = h(\lambda)$. The sequence of *Picard iterates* $(p_k)_{k\geq 0}$ is recursively defined by

$$p_0(t,\lambda) := h(\lambda), \quad p_{k+1}(t,\lambda) := h(\lambda) + \int_0^t f(p_k(s,\lambda))ds, \quad k \geq 0.$$

Interval arithmetic. A foundational concept we are making use of to discharge arithmetic proof obligations is *interval arithmetic* [37]. We write $\mathbb{IQ}^n$ to denote the set of rational intervals in the n-dimensional vector space $\mathbb{Q}^n$. We write $\mathbf{I} = [\underline{\mathbf{I}}, \overline{\mathbf{I}}] \in \mathbb{IQ}^n$, where $\underline{\mathbf{I}} \in \mathbb{Q}^n$ and $\overline{\mathbf{I}} \in \mathbb{Q}^n$ are the lower and upper bound respectively. In order to simplify, notation we write

$$\begin{aligned} \mathrm{mid}(\mathbf{I}) &:= \frac{\overline{\mathbf{I}} + \underline{\mathbf{I}}}{2}, \\ \mathrm{rad}(\mathbf{I}) &:= \overline{\mathbf{I}} - \underline{\mathbf{I}}, \end{aligned}$$

for an intervals *mid-point* and *radius*. For a variable x and concrete dL term e, we write $[e]_x^{\mathbf{I}}$ to denote its interval evaluation with respect to the interval $\mathbf{I}$.

Taylor models. Equipped with interval arithmetic and Picard polynomial approximations, we are now ready to define Taylor models in dL [11].

Definition 3 (Taylor models). *Let $p(t,\lambda)$ be a concrete dL term and $\mathbf{I}(t)$ a dL interval. Then a dL* Taylor model *is a tuple $(p, \mathbf{I})$ whose associated formula is*

$$\mathrm{TM}_{p,\mathbf{I}}\big(x,\lambda,t\big) \;\equiv\; \underline{\mathbf{I}}(t) \leq x - p(t,\lambda) \leq \overline{\mathbf{I}}(t).$$

For its derivative formula, we write

$$\partial_t \mathrm{TM}_{p,\mathbf{I}}\big(x,\lambda,t\big) \;\equiv\; \partial_t \underline{\mathbf{I}}(t) < f(x) - \partial_t p(t,\lambda) < \partial_t \overline{\mathbf{I}}(t).$$

Theorem 3.2. *Let $(p, \mathbf{I})$ be a Taylor model. Let h be a function symbol and $X_0(x) \equiv \exists\lambda\,(x = h(\lambda) \wedge \|\lambda\|_\infty \leq 1)$. The following is a sound derived proof rule of dL:*

$$\frac{\begin{array}{l} \exists\lambda\exists t(\mathrm{TM}_{p,\mathbf{I}}\big(x,\lambda,t\big) \wedge 0 \leq t \leq \Delta t \wedge \|\lambda\|_\infty \leq 1) \;\vdash P(x) \\ X_0(x) \vdash \exists\lambda\left(\mathrm{TM}_{p,\mathbf{I}}\big(x,\lambda,0\big) \wedge \|\lambda\|_\infty \leq 1\right) \\ \mathrm{TM}_{p,\mathbf{I}}\big(x,\lambda,t\big),\; 0 \leq t \leq \Delta t,\; \|\lambda\|_\infty \leq 1 \vdash \partial_t \mathrm{TM}_{p,\mathbf{I}}\big(x,\lambda,t\big) \end{array}}{X_0(x),\; t = 0 \vdash\; [x' = f(x), t' = 1 \;\&\; t \leq \Delta t]P(x)}$$

Zonotope reachable set We now focus our attention on the zonotope case and set out to prove premise 1 of Theorem 3.1. Our strategy is to successively rewrite the proof goal until it is an instance of zonotope containment. Once in that form, a lightweight numerical solver can supply a witness that one zonotope is contained in the other.

Definition 4 (Zonotope). *Let G be an $n \times p$ generator* dL *matrix, c an n center* dL *vector, and let λ be a p* dL *vector. The* zonotope formula *associated with $\langle c,\ G\rangle$ is given by*

$$\langle c,\ G\rangle(x) \ \equiv\ \exists\lambda\big(x = c + G\,\lambda \wedge \|\lambda\|_\infty \le 1\big).$$

In order to simplify arithmetic reasoning, our goal is to linearize the Picard iterates. This helps us to over-approximate the reachable set at the time instance $\varDelta t$ by a zonotope. We have the following theorem:

Lemma 3.1 (Linear interval abstraction). *Let $p(x)$ be a concrete* dL *polynomial in x. Then, for the interval $\mathbf{I} \in \mathbb{IQ}^n$ and the interval remainder*

$$\mathbf{R} = \big[p(x) - p(0) - \nabla_x p(0)^\top x\big]_x^{\mathbf{I}}$$

the following formula is a sound axiom of dL*:*

$$\underline{\mathbf{I}} \le x \le \bar{\mathbf{I}} \to \exists\xi(p(x) = p(0) + \nabla_x p(0)^\top x + \mathrm{mid}(\mathbf{R}) + \frac{1}{2}\,\mathrm{rad}(\mathbf{R})\xi \wedge \|\xi\|_\infty \le 1).$$

Theorem 3.3 (Zonotope reachable set for discrete time instance). *Let $(p, \mathbf{I})$ be a Taylor model of the ODE system $x' = f(x)\ \&\ t \le \varDelta t$. Let*

$$\mathbf{R} = \big[p(t,\lambda) - p(t,0) - \nabla_\lambda p(t,0)^\top \lambda\big]_{(t,\lambda)}^{[0,\varDelta t]\times[-1,1]^n}$$

be the remainder of its linear interval abstraction. Finally, we define the dL *center vector and generator matrix*

$$b := p(\varDelta t, 0) + \mathrm{mid}(\mathbf{I}(\varDelta t)) + \mathrm{mid}(\mathbf{R}),$$
$$H := [\nabla_\lambda p(\varDelta t, 0),\ \frac{1}{2}\,\mathrm{rad}(\mathbf{I}(\varDelta t)),\ \frac{1}{2}\,\mathrm{rad}(\mathbf{R})].$$

Then, the following is a sound derived proof rule of dL:

$$\frac{\begin{array}{l}\langle b,\ H\rangle(x,u) \vdash \langle c_x,\ G_x\rangle(x,u)\\ \langle c,\ G\rangle(x,u) \vdash \exists\lambda\,\big(\mathrm{TM}_{p,\mathbf{I}}(x,u,\lambda,0) \wedge \|\lambda\|_\infty \le 1\big)\\ \mathrm{TM}_{p,\mathbf{I}}(x,u,\lambda,t),\ 0 \le t \le \varDelta t,\ \|\lambda\|_\infty \le 1 \vdash \partial_t \mathrm{TM}_{p,\mathbf{I}}(x,u,\lambda,t)\end{array}}{\langle c,\ G\rangle(x,u), t = 0 \vdash [\mathrm{plant}]\,(t = \varDelta t \to \exists u\,\langle c,\ G\rangle(x,u))}$$

We now turn our attention to second premise in Theorem 3.1. Unlike the first premise, which involves a single sampling instant, this condition must hold for every $t \in [0, \varDelta t]$. In other words, we must enclose the entire time-tube of states captured by $\mathcal{R}\big([0,\varDelta t]\,, \mathcal{X}_0, \mathcal{E}\big)$. We tackle this problem by constructing a single zonotope that encloses the reachable set at every time instant $t \in [0, \varDelta t]$.

Theorem 3.4 (Zonotope reachable set for time intervals). *Let* $(p, \mathbf{I})$ *be a Taylor model for the ODE system* $x' = f(x) \;\&\; t \leq \Delta t$. *We define the remainder polynomial*

$$\begin{aligned} r(t,\lambda,\xi) := {} & p(t,\lambda) + \operatorname{mid}(\mathbf{I}(t)) + \frac{1}{2}\operatorname{rad}(\mathbf{I}(t))^\top \xi - p(0,0) - \operatorname{mid}(\mathbf{I}(0)) \\ & - \partial_t p(0,0)\, t - \partial_t \operatorname{mid}(\mathbf{I}(t))\big|_{t=0}\, t - \nabla_\lambda p(0,0)^\top \lambda - \tfrac{1}{2}\operatorname{rad}(\mathbf{I}(0))^\top \xi. \end{aligned}$$

To bound its range over the domain $\mathbf{J} := [0, \Delta t] \times [-1,1]^n \times [-1,1]^n$, *we introduce the error interval*

$$\mathbf{R} := [\, r(t,\lambda,\xi)\,]^{\mathbf{J}}_{(t,\lambda,\xi)}\,.$$

Finally, we define the dL *center vector and generator matrix*

$$\begin{aligned} b &:= p(0,0) + \operatorname{mid}(\mathbf{I}(0)) + \operatorname{mid}(\mathbf{R}), \\ H &:= [\partial_t p(0,0) + \partial_t \operatorname{mid}(\mathbf{I}(t))|_{t=0},\ \nabla_\lambda p(0,0),\ \frac{1}{2}\operatorname{rad}(\mathbf{I}(0)), \frac{1}{2}\operatorname{rad}(\mathbf{R})]. \end{aligned}$$

Then, the following is a sound derived proof rule of dL*:*

$$\frac{\begin{array}{l} \langle b,\ H\rangle(x) \vdash X(x) \\ \langle c,\ G\rangle(x,u) \vdash \exists\lambda\,\big(\mathrm{TM}_{p,\mathbf{I}}(x,u,\lambda,0) \wedge \|\lambda\|_\infty \leq 1\big) \\ \mathrm{TM}_{p,\mathbf{I}}(x,u,\lambda,t),\ 0 \leq t \leq \Delta t,\ \|\lambda\|_\infty \leq 1 \vdash \partial_t \mathrm{TM}_{p,\mathbf{I}}(x,u,\lambda,t) \end{array}}{\langle c,\ G\rangle(x,u), t = 0 \vdash [\mathrm{plant}]X(x)}$$

Witness checks for arithmetic goals. Recall that our earlier theorems translate the reachability problem into a single question of zonotope containment. In practice, the zonotopes that bound control envelopes usually involve on the order of ten to twenty generators. Consequently, the resulting formulas can become quite large. A straightforward quantifier-elimination procedure is not a good fit for this problem, because its computational costs are prohibitive. A common approach in reachability analysis is to invoke a witness theorem [46, Cor. 4]: for zonotopes $\langle c,\ G\rangle(x)$, $\langle b,\ H\rangle(x)$ the containment

$$\forall x\big(\langle c,\ G\rangle(x) \to \langle b,\ H\rangle(x)\big)$$

holds whenever one can find a matrix Γ and vector β that satisfy

$$H\Gamma = G, \quad b - c = H\beta, \quad \|(\Gamma,\beta)\|_\infty \leq 1. \tag{6}$$

These linear formulas define a witness (Γ, β), which can be computed with an efficient linear program. To obtain a rigorous proof, we must produce an exact witness. That means solving the linear program with exact rational arithmetic — an operation that is orders of magnitude slower than solving with floating-point numbers. Instead of insisting on exact equality, we weaken the witness condition slightly, allowing for small perturbations. These relaxed conditions can be solved efficiently in floating-point arithmetic with a numerical linear programming solver and then rationalized with the residual margin ensuring the conditions still hold.

Theorem 3.5 (Zonotope containment). *The following is a derived proof rule of* dL*:*

$$\cfrac{\begin{array}{l}\vdash HH^{+} = I \\ \vdash b - c = H\beta \\ \vdash \|H\Gamma - G\|_{\infty} \leq \varepsilon \\ \vdash \|(\Gamma, \beta)\|_{\infty} \leq 1 - \varepsilon \, \|H^{+}\|_{\infty}\end{array}}{\langle c,\ G\rangle(x) \vdash \langle b,\ H\rangle(x)}$$

Equipped with Theorem 3.1 to Theorem 3.5, we have all the necessary ingredients to carry out an end-to-end verification of a concrete system.

4 Evaluation

In this section, we compute the control envelopes of two examples and demonstrate how to verify them using the theorem from Sec. 3. The key challenge here is to handle the different representations that are used in reachability tools and the theorem prover KeYmaera X.

To compute RCI sets for sampled-data systems, we use the approach by Schäfer et al. [47]. The authors represent both the RCI and over-approximations of the reachable sets as zontopes. Using the witness conditions (6), the one-step invariance, the one-step safety and control-admissibility in Def. 2 are formulated as constraints in an optimization problem returning an RCI set with maximum volume. In order to verify the envelope in dL's formalism, we need a post-processing step: we rationalize center vector and generator matrix of the output zonotopes[4]. Then, we compute a provably correct dL Taylor model using the approach implemented in KeYmaera X[5]. Finally, we verify the zonotope containment.

4.1 Double Integrator

The dynamics of the double integrator are governed by the following differential equations [25, Sec. V.A]:

$$\begin{aligned} x_1' &= x_2 + w_1, \\ x_2' &= \frac{1}{m}u + w_2, \end{aligned}$$

where the system states are the position x_1 and the velocity x_2 of the point-mass, the system input is the force $u = F$, and the weight of the point-mass is $m = 1\text{kg}$. The state constraints are $|x_1| \leq 1\text{m}$ and $|x_2| \leq 1{}^{\text{m}}/_{\text{s}}$, the input

[4] A suitable operator is implemented in the MATLAB function `rat`, see https://de.mathworks.com/help/matlab/ref/rat.html.

[5] https://github.com/LS-Lab/KeYmaeraX-release/blob/master/keymaerax-core/src/main/scala/org/keymaerax/btactics/TaylorModel.scala

constraint is $|u| \leq 1\text{N}$, and the set of disturbances is $w_1 \in [-0.1, 0.1]^{\text{m}}/_{\text{s}}$ and $w_2 \in [-0.1, 0.1]^{\text{m}}/_{\text{s}^2}$. The sampling time is $\Delta t = 0.1\,\text{s}$. In dL, we use a slightly enlarged initial condition given by the formula

$$X_0(x, u) \equiv \exists\lambda \left(x_1 = \frac{11}{10}\lambda_1 \wedge x_2 = \frac{11}{10}\lambda_2 \wedge u = \frac{11}{10}\lambda_3 \wedge \|\lambda\|_\infty \leq 1 \right).$$

By Picard iteration we obtain the following polynomial approximation

$$\begin{aligned} p_{x_1}(t, \lambda) &= \lambda_1 + t\lambda_2 + \frac{t^2}{2}\lambda_3, \\ p_{x_2}(t, \lambda) &= \lambda_2 + t\lambda_3, \\ p_u(t, \lambda) &= \lambda_3. \end{aligned}$$

With interval arithmetic we then obtain the following provable error bounds:

$$\begin{aligned} \underline{\mathbf{I}}_{x_1}(t) &= -101020 \cdot 10^{-11}t \quad & \overline{\mathbf{I}}_{x_1}(t) &= 101020 \cdot 10^{-11}t, \\ \underline{\mathbf{I}}_{x_2}(t) &= -10^{-6}t \quad & \overline{\mathbf{I}}_{x_2}(t) &= 10^{-6}t, \\ \underline{\mathbf{I}}_{u}(t) &= -10^{-6}t \quad & \overline{\mathbf{I}}_{u}(t) &= 10^{-6}t. \end{aligned}$$

Together, the polynomial p and the error interval $\mathbf{I}(t)$ yield the Taylor model $(p, \mathbf{I})$ on $[0, \Delta t]$. We numerically compute an RCI set and, using the Taylor model, check the premises of Theorem 3.3 and Theorem 3.4 (See Fig. 1a). Note that, although we visually verify containment here, a formal proof can be derived directly by applying Theorem 3.5.

4.2 Controlled Jet Engine

Next, we consider the Moore-Greitzer model of a jet engine [33,40] whose dynamics are governed by

$$\begin{aligned} x_1' &= -x_2 - \frac{3}{2}x_1^2 - \frac{1}{2}x_1^3 + w, \\ x_2' &= u, \end{aligned} \tag{7}$$

where the system states are the mass flow x_1 and the pressure rise x_2. The state constraints are $|x_1| \leq 0.2$ and $|x_2| \leq 0.2$, the input constraint is $|u| \leq 0.3$, and the set of disturbances is $w \in [-0.025, 0.0.025]$. Measurements are taken with a sampling time of $\Delta t = 0.1$ time units. Computing the associated dL Taylor model with conservative initial conditions

$$X_0(x, u) \equiv \exists\lambda \left(x_1 = \frac{3}{10}\lambda_1 \wedge x_2 = \frac{3}{10}\lambda_2 \wedge u = \frac{3}{10}\lambda_3 \wedge \|\lambda\|_\infty \leq 1 \right)$$

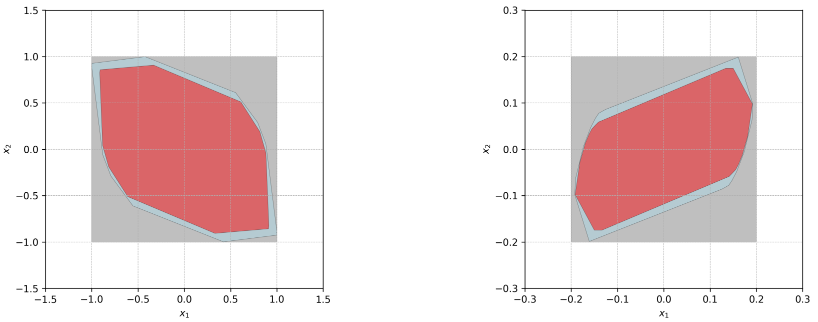
yields the following provable polynomial approximation

$$\begin{aligned} p_{x_1}(t, \lambda) &= \lambda_1 - t\lambda_2 + \frac{3t}{2}\lambda_1^2 - \frac{t^2}{2}\lambda_3 - \frac{3t^2}{2}\lambda_1\lambda_2 - \frac{t}{2}\lambda_1^3, \\ p_{x_2}(t, \lambda) &= \lambda_2 + t\lambda_3, \\ p_u(t, \lambda) &= \lambda_3, \end{aligned}$$

with error bounds

$$\begin{aligned}\underline{\mathbf{I}}_{x_1}(t) = -28605705206 \cdot 10^{-11}t \quad & \overline{\mathbf{I}}_{x_1}(t) = 27585076206 \cdot 10^{-11}t,\\ \underline{\mathbf{I}}_{x_2}(t) = -10^{-6}t \quad & \overline{\mathbf{I}}_{x_2}(t) = 10^{-6}t,\\ \underline{\mathbf{I}}_{u}(t) = -10^{-6}t \quad & \overline{\mathbf{I}}_{u}(t) = 10^{-6}t.\end{aligned}$$

Again, using the Taylor model $(p, \mathbf{I})$, we can compute the zonotopes form Theorem 3.3 and Theorem 3.4 (See Fig. 1b).



(a) Double integrator

(b) Jet engine

Fig. 1: Comparison of the numerically computed robust control-invariant set (blue), its Taylor-model reachability over-approximation after one sampling period (red), and the safety region (gray).

5 Related Work

Reachability analysis. Reachability analysis itself can be categorized into several techniques: simulation-based techniques [24], Hamilton-Jacobi techniques [16], and set propagation techniques [3]. The disadvantage of simulation-based techniques and Hamilton-Jacobi techniques is that they scale exponentially in the number of continuous state variables, while many set propagation techniques scale polynomially [3]. Because the main purpose of including reachability analysis into theorem proving is to deduce properties of complex dynamics with potentially many continuous state variables, we will focus on set propagation techniques. This technique is also currently predominantly used in the International Competition on Verifying Continuous and Hybrid Systems [6,23]. Further advantages of set-based reachability analysis are that it can be fully automated [55,54] and easily interpreted due to its resemblance with the numerical simulation of systems.

Computing Invariant Sets Using Reachability Analysis Computing invariant sets has a rich history in control theory due to their many applications: they serve as terminal regions in model predictive control [44] or are employed as part of supervisory safety-filters in, e.g., learning-based control [36,53]. Since we aim at reducing conservatism, we focus on the computation of the largest invariant set, also known as the maximal invariant set. The maximal invariant set can be obtained using the standard set recursion introduced in [10,45]. However, this procedure usually fails to terminate in finite time and the computational complexity of the required set operations restricts the applicability to low-dimensional systems.

The latter also holds for algorithms for approximating the maximal invariant set by gridding the sate space [14,15]. Thus, most approaches in the literature formulate an optimization problem to compute a possibly large invariant set.

The most popular set representations of the invariant set are ellipsoids [17,35] and polytopes [41,27,20,9]. Due to their low representation complexity, algorithms using ellipsoids as the set representation scale better to higher-dimensional systems at the cost of more conservative results. On the other hand, polytopic invariant sets enable more flexibility and, thus, larger invariant sets while sacrificing computational efficiency.

Level sets are an even more flexible representation of invariant sets that can be computed using Hamilton-Jacobi reachability analysis [58,57] and control-barrier functions [7,43,56]. To circumvent the exponential complexity of solving the associated partial differential equation numerically (Hamilton-Jacobi reachability analysis) or synthesizing the barrier certificate from simulations [7], the computation of an invariant set can be relaxed into a (sequence of) semidefinite program(s) using sum-of-squares programming. However, an invariance-enforcing controller must be designed prior to computing the invariant set [58,57] or the approach suffers from poor scalability due to the large number of variables of the semi-definite program [32,1].

The approaches reviewed above typically either consider discrete-time systems, e.g., [20,35] and, thus, do not check constraint satisfaction in between sampling times, or enforce invariance for the continous-time dynamical system, e.g., [7,9], which is unnecessarily conservative. As an alternative, invariant sets for sampled-data systems have been characterized in [42]: sampled-data systems are continuous-time systems that are controlled by a digital controller; similary, measurements are only taken at discrete points in time [36]. Crucially, the system can leave the invariant set in between sampling times, which reduces conservatism. This notion of invariance has been employed in [26] to introduce so-called safe sets of linear systems. Since this approach represents the invariant set as a zonotope, invariant sets of high-dimensional systems can be computed efficiently. This concept has been extended to ellipsoidal sets in [34] and nonlinear systems in [48,47].

Deductive verification of control problems Differential dynamic logic (dL) has been successfully applied in several control domains, including air traffic control, train control, and ground robots to formally prove safety properties [13,29,31]. It has also been employed for the deductive verification of control system stability

[52,51]. The control-system meta-model has been extended to incorporate the environment, with a focus on identifying conditions that prevent proofs of safety from being invalidated by modeling errors [50]. In contrast, our work focuses on a simplified controller-plant model and formalizes in dL the verification of synthesized controllers by reducing closed-loop analysis to continuous-time safety and discrete-time invariance.

The problem of control-envelope synthesis has been studied in the context of dL before [8,30]. Both of these works approach the problem from a logical perspective and do not leverage existing techniques and tools developed in the field of reachability analysis field. By comparison, our work integrates these two viewpoints by combining established numerical methods.

6 Conclusion

In this paper, we established a link between two traditionally separate research fields: reachability analysis and theorem proving. We showed how zonotope-based reachable-set computations can be encoded in the dL formalism and how a control envelope can be formally verified. Although the differing levels of representation between these tools posed nontrivial technical challenges, our case studies demonstrate that these obstacles can be overcome. By combining the computational efficiency of reachability analysis with the deductive rigor of theorem proving, we achieve a verification workflow that is both scalable and formally sound. This work represents only the first step toward a more unified formal-methods ecosystem. In future work, we plan to explore how reachability methods can be even more tightly integrated, reducing the boundaries between research fields.

Acknowledgements

This work was supported by the German Research Foundation - SFB 1608 - under grant number 501798263 and by an Alexander von Humbolt Professorship. Additionally, we would like to thank Fabian Immler for his valuable Taylor model implementation in KeYmaera X.

References

1. Ahmadi, A.A., Hall, G., Papachristodoulou, A., Saunderson, J., Zheng, Y.: Improving efficiency and scalability of sum of squares optimization: Recent advances and limitations. In: IEEE Conference on Decision and Control. pp. 453–462 (2017). doi: 10.1109/CDC.2017.8263706
2. Althoff, M., Frehse, G.: Combining zonotopes and support functions for efficient reachability analysis of linear systems. In: Proc. of the 55th IEEE Conference on Decision and Control. pp. 7439–7446 (2016). doi: 10.1109/CDC.2016.7799418
3. Althoff, M., Frehse, G., Girard, A.: Set propagation techniques for reachability analysis. Annual Review of Control, Robotics, and Autonomous Systems **4**(1), 369–395 (2021). doi: 10.1146/annurev-control-071420-081941

4. Althoff, M., Krogh, B.H.: Zonotope bundles for the efficient computation of reachable sets. In: Proc. of the 50th IEEE Conference on Decision and Control. pp. 6814–6821 (2011). doi: 10.1109/CDC.2011.6160872
5. Althoff, M.: An Introduction to CORA 2015. In: ARCH14-15. 1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems. pp. 120–87. doi: 10.29007/zbkv
6. Althoff, M., Forets, M., Schilling, C., Wetzlinger, M.: ARCH-COMP24 category report: Continuous and hybrid systems with linear continuous dynamics. In: Frehse, G., Althoff, M. (eds.) Proc. of the 11th Int. Workshop on Applied Verification for Continuous and Hybrid Systems. EPiC Series in Computing, vol. 103, pp. 15–38. EasyChair (2024). doi: 10.29007/7xf3
7. Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., Tabuada, P.: Control barrier functions: Theory and applications. In: European Control Conference. pp. 3420–3431 (2019). doi: 10.23919/ECC.2019.8796030
8. Aréchiga, N., Krogh, B.: Using verified control envelopes for safe controller design. In: 2014 American Control Conference. pp. 2918–2923 (Jun 2014). doi: 10.1109/ACC.2014.6859307
9. Ben Sassi, M.A., Girard, A.: Controller synthesis for robust invariance of polynomial dynamical systems using linear programming. Systems & Control Letters **61**(4), 506–512 (2012). doi: 10.1016/j.sysconle.2012.01.004
10. Bertsekas, D.: Infinite time reachability of state-space regions by using feedback control. IEEE Transactions on Automatic Control **17**(5), 604–613 (1972). doi: 10.1109/TAC.1972.1100085
11. Berz, M., Makino, K.: Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models. Reliable Computing **4**(4), 361–369 (Nov 1998). doi: 10.1023/A:1024467732637
12. Blanchini, F.: Set invariance in control. Automatica **35**(11), 1747–1767 (1999)
13. Bohrer, R., Tan, Y.K., Mitsch, S., Sogokon, A., Platzer, A.: A Formal Safety Net for Waypoint-Following in Ground Robots. IEEE Robotics and Automation Letters **4**(3), 2910–2917 (Jul 2019). doi: 10.1109/LRA.2019.2923099
14. Bravo, J., Limon, D., Alamo, T., Camacho, E.: On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach. Automatica **41**(9), 1583–1589 (2005). doi: 10.1016/j.automatica.2005.04.015
15. Brown, S., Khajenejad, M., Yong, S.Z., Martínez, S.: Computing controlled invariant sets of nonlinear control-affine systems. In: IEEE Conference on Decision and Control. pp. 7830–7836 (2023). doi: 10.1109/CDC49753.2023.10383613
16. Chen, M., Tomlin, C.J.: Hamilton–Jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management. Annual Review of Control, Robotics, and Autonomous Systems **1**, 333–358 (2018). doi: 10.1146/annurev-control-060117-104941
17. Chen, W.H., O'Reilly, J., Ballance, D.: On the terminal region of model predictive control for non-linear systems with input/state constraints. International Journal of Adaptive Control and Signal Processing **17**, 195–207 (2003). doi: 10.1002/acs.731
18. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Flow*: An Analyzer for Nonlinear Hybrid Systems. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification. pp. 258–263. Springer, Berlin, Heidelberg (2013). doi: 10.1007/978-3-642-39799-8_18
19. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In: Goos, G., Hartmanis, J., Brinch Hansen, P., Gries, D., Moler, C., Seegmüller, G., Wirth, N., Brakhage, H. (eds.) Automata Theory

and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975, vol. 33, pp. 134–183. Springer Berlin Heidelberg, Berlin, Heidelberg (1975). doi: 10.1007/3-540-07407-4_17

20. Fiacchini, M., Alamo, T., Camacho, E.: On the computation of convex robust control invariant sets for nonlinear systems. Automatica **46**(8), 1334–1338 (2010). doi: 10.1016/j.automatica.2010.05.007
21. Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., Maler, O.: SpaceEx: Scalable Verification of Hybrid Systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Computer Aided Verification. pp. 379–395. Springer, Berlin, Heidelberg (2011). doi: 10.1007/978-3-642-22110-1_30
22. Fulton, N., Mitsch, S., Quesel, J.D., Völp, M., Platzer, A.: KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In: Proc. of Automated Deduction. pp. 527–538. Springer (2015). doi: 10.1007/978-3-319-21401-6_36
23. Geretti, L., Sandretto, J.A.D., Althoff, M., Benet, L., Collins, P., Forets, M., Mitsch, S., Schilling, C., Tillet, J., Wetzlinger, M.: ARCH-COMP24 category report: Continuous and hybrid systems with nonlinear dynamics. In: Frehse, G., Althoff, M. (eds.) Proc. of the 11th Int. Workshop on Applied Verification for Continuous and Hybrid Systems. EPiC Series in Computing, vol. 103, pp. 39–63. EasyChair (2024). doi: 10.29007/21ch
24. Girard, A., Pappas, G.J.: Verification using simulation. In: Hybrid Systems: Computation and Control. pp. 272–286. LNCS 3927, Springer (2006). doi: 10.1007/11730637_22
25. Gruber, F., Althoff, M.: Computing safe sets of linear sampled-data systems. IEEE Control Systems Letters **5**(2), 385–390 (2021). doi: 10.1109/LCSYS.2020.3002476
26. Gruber, F., Althoff, M.: Scalable robust output feedback mpc of linear sampled-data systems. In: IEEE Conference on Decision and Control. pp. 2563–2570 (2021). doi: 10.1109/CDC45484.2021.9683384
27. Gupta, A., Falcone, P.: Full-complexity characterization of control-invariant domains for systems with uncertain parameter dependence. IEEE Control Systems Letters **3**(1), 19–24 (2019). doi: 10.1109/LCSYS.2018.2849714
28. Hellwig, J., Schäfer, L., Qian, L., Platzer, A., Althoff, M.: From Zonotopes to Proof Certificates: A Formal Pipeline for Safe Control Envelopes (Sep 2025). doi: 10.48550/arXiv.2509.20301
29. Jeannin, J.B., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., Platzer, A.: Formal verification of ACAS X, an industrial airborne collision avoidance system. In: 2015 International Conference on Embedded Software (EMSOFT). pp. 127–136 (Oct 2015). doi: 10.1109/EMSOFT.2015.7318268
30. Kabra, A., Laurent, J., Mitsch, S., Platzer, A.: CESAR: Control Envelope Synthesis via Angelic Refinements. In: Finkbeiner, B., Kovács, L. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 144–164. Springer Nature Switzerland, Cham (2024). doi: 10.1007/978-3-031-57246-3_9
31. Kabra, A., Mitsch, S., Platzer, A.: Verified Train Controllers for the Federal Railroad Administration Train Kinematics Model: Balancing Competing Brake and Track Forces. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **41**(11), 4409–4420 (Nov 2022). doi: 10.1109/TCAD.2022.3197690
32. Korda, M., Henrion, D., Jones, C.N.: Convex computation of the maximum controlled invariant set for polynomial control systems. SIAM Journal on Control and Optimization **52**(5), 2944–2969 (2014). doi: 10.1137/130914565
33. Krstic, M., Kanellakopoulos, I., Kokotovic, P.V.: Nonlinear and Adaptive Control Design. Wiley-Interscience, 1 edn. (1995)

34. Kulmburg, A., Schafer, L., Althoff, M.: Approximability of the Containment Problem for Zonotopes and Ellipsotopes. IEEE Transactions on Automatic Control pp. 1–16 (2025). doi: 10.1109/TAC.2025.3583624
35. Lazar, M., Tetteroo, M.: Computation of terminal costs and sets for discrete–time nonlinear MPC. IFAC-PapersOnLine **51**(20), 141–146 (2018). doi: 10.1016/j.ifacol.2018.11.006
36. Mitchell, I.M., Yeh, J., Laine, F.J., Tomlin, C.J.: Ensuring safety for sampled data systems: An efficient algorithm for filtering potentially unsafe input signals. In: IEEE Conference on Decision and Control. pp. 7431–7438 (2016). doi: 10.1109/CDC.2016.7799417
37. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics (Jan 2009). doi: 10.1137/1.9780898717716
38. Platzer, A.: A Complete Uniform Substitution Calculus for Differential Dynamic Logic. Journal of Automated Reasoning **59**(2), 219–265 (Aug 2017). doi: 10.1007/s10817-016-9385-1
39. Platzer, A.: Logical Foundations of Cyber-Physical Systems. Springer International Publishing, Cham (2018). doi: 10.1007/978-3-319-63588-0
40. Platzer, A., Qian, L.: Axiomatization of Compact Initial Value Problems: Open Properties. Journal of the ACM . doi: 10.1145/3763228
41. Rakovic, S.V., Baric, M.: Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks. IEEE Transactions on Automatic Control **55**(7), 1599–1614 (2010). doi: 10.1109/TAC.2010.2042341
42. Raković, S., Fontes, F., Kolmanovsky, I.: Reachability and invariance for linear sampled–data systems. IFAC-PapersOnLine **50**(1), 3057–3062 (2017). doi: 10.1016/j.ifacol.2017.08.675
43. Rauscher, M., Kimmel, M., Hirche, S.: Constrained robot control using control barrier functions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 279–285 (2016). doi: 10.1109/IROS.2016.7759067
44. Rawlings, J.B., Mayne, D.Q., Diehl, M.M.: Model Predictive Control: Theory, Computation, and Design. Nob Hill Publishing, LLC (2022)
45. Rungger, M., Tabuada, P.: Computing robust controlled invariant sets of linear systems. IEEE Transactions on Automatic Control **62**(7), 3665–3670 (2017). doi: 10.1109/TAC.2017.2672859
46. Sadraddini, S., Tedrake, R.: Linear Encodings for Polytope Containment Problems. In: 2019 IEEE 58th Conference on Decision and Control (CDC). pp. 4367–4372 (Dec 2019). doi: 10.1109/CDC40024.2019.9029363
47. Schäfer, L., Gruber, F., Althoff, M.: Scalable computation of robust control invariant sets of nonlinear systems. IEEE Transactions on Automatic Control **69**(2), 755–770 (2024). doi: 10.1109/TAC.2023.3275305
48. Schäfer, L., Althoff, M.: Computing robust control invariant sets of nonlinear systems using polynomial controller synthesis. In: Proc. of the American Control Conference (2024). doi: 10.23919/ACC60939.2024.10644939
49. Schürmann, B., Kochdumper, N., Althoff, M.: Reachset model predictive control for disturbed nonlinear systems. In: 2018 IEEE Conference on Decision and Control (CDC). pp. 3463–3470 (2018). doi: 10.1109/CDC.2018.8619781
50. Selvaraj, Y., Krook, J., Ahrendt, W., Fabian, M.: On proving that an unsafe controller is not proven safe. Journal of Logical and Algebraic Methods in Programming **137**, 100939 (Feb 2024). doi: 10.1016/j.jlamp.2023.100939

51. Tan, Y.K., Mitsch, S., Platzer, A.: Verifying Switched System Stability With Logic. In: Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control. pp. 1–11. HSCC '22, Association for Computing Machinery, New York, NY, USA (May 2022). doi: 10.1145/3501710.3519541
52. Tan, Y.K., Platzer, A.: Deductive Stability Proofs for Ordinary Differential Equations. In: Groote, J.F., Larsen, K.G. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 181–199. Springer International Publishing, Cham (2021). doi: 10.1007/978-3-030-72013-1_10
53. Wabersich, K.P., Zeilinger, M.N.: Linear model predictive safety certification for learning-based control. In: 2018 IEEE Conference on Decision and Control (CDC). pp. 7130–7135 (2018). doi: 10.1109/CDC.2018.8619829
54. Wetzlinger, M., Kochdumper, N., Bak, S., Althoff, M.: Fully-automated verification of linear systems using reachability analysis with support functions. In: Proc. of the 26th ACM International Conference on Hybrid Systems: Computation and Control (2023). doi: 10.1145/3575870.3587121
55. Wetzlinger, M., Kulmburg, A., Althoff, M.: Adaptive parameter tuning for reachability analysis of nonlinear systems. In: Proc. of the 24th International Conference on Hybrid Systems: Computation and Control. HSCC '21, Association for Computing Machinery (2021). doi: 10.1145/3447928.3456643
56. Xu, X., Tabuada, P., Grizzle, J.W., Ames, A.D.: Robustness of control barrier functions for safety critical control. IFAC-PapersOnLine **48**(27), 54–61 (2015). doi: 10.1016/j.ifacol.2015.11.152
57. Xue, B., Wang, Q., Zhan, N., Wang, S., She, Z.: Synthesizing robust domains of attraction for state-constrained perturbed polynomial systems. SIAM Journal on Control and Optimization **59**(2), 1083–1108 (2021). doi: 10.1137/19M125220X
58. Xue, B., Wang, Q., Zhan, N., Fränzle, M.: Robust invariant sets generation for state-constrained perturbed polynomial systems. In: International Conference on Hybrid Systems: Computation and Control. pp. 128–137 (2019). doi: 10.1145/3302504.3311810