Verifying Switched System Stability With Logic

Yong Kiam Tan
Carnegie Mellon University
Pittsburgh, PA, USA
yongkiat@cs.cmu.edu

Stefan Mitsch
Carnegie Mellon University
Pittsburgh, PA, USA
smitsch@cs.cmu.edu

André Platzer
Carnegie Mellon University
Pittsburgh, PA, USA
aplatzer@cs.cmu.edu

ABSTRACT

Switched systems are known to exhibit subtle (in)stability behaviors requiring system designers to carefully analyze the stability of closed-loop systems that arise from their proposed switching control laws. This paper presents a formal approach for verifying switched system stability that blends classical ideas from the controls and verification literature using differential dynamic logic (dL), a logic for deductive verification of hybrid systems. From controls, we use standard stability notions for various classes of switching mechanisms and their corresponding Lyapunov function-based analysis techniques. From verification, we use dL's ability to verify quantified properties of hybrid systems and dL models of switched systems as looping hybrid programs whose stability can be formally specified and proven by finding appropriate *loop invariants*, i.e., properties that are preserved across each loop iteration. This blend of ideas enables a trustworthy implementation of switched system stability verification in the KeYmaera X prover based on dL. For standard classes of switching mechanisms, the implementation provides fully automated stability proofs, including searching for suitable Lyapunov functions. Moreover, the generality of the deductive approach also enables verification of switching control laws that require non-standard stability arguments through the design of loop invariants that suitably express specific intuitions behind those control laws. This flexibility is demonstrated on three case studies: a model for longitudinal flight control by Branicky, an automatic cruise controller, and Brockett's nonholonomic integrator.

CCS CONCEPTS

• **Theory of computation** → **Logic and verification**; **Timed and hybrid models**; • **Computing methodologies** → *Computational control theory*; • **Computer systems organization** → *Embedded systems*.

KEYWORDS

switched system stability, loop invariants, differential dynamic logic

ACM Reference Format:
Yong Kiam Tan, Stefan Mitsch, and André Platzer. 2022. Verifying Switched System Stability With Logic. In *25th ACM International Conference on Hybrid Systems: Computation and Control (HSCC '22), May 4–6, 2022, Milan, Italy.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3501710.3519541

HSCC '22, May 4–6, 2022, Milan, Italy

ACM ISBN 978-1-4503-9196-2/22/05.
https://doi.org/10.1145/3501710.3519541

1 INTRODUCTION

Switched systems provide a powerful mathematical paradigm for the design and analysis of discontinuous (or nondifferentiable) control mechanisms [10, 22, 27, 42]. Examples of such mechanisms include: bang-bang controllers that switch between on/off modes; gain schedulers that switch between a family of locally valid linear controllers; and supervisory control, where a supervisor switches between candidate controllers based on logical criteria [22, 27]. However, switched systems are known to exhibit subtle (in)stability behaviors, e.g., switching between stable subsystems can lead to instability [22], so it is important for system designers to adequately justify the stability of their proposed switching designs. Verification and validation are complementary approaches for such justifications: *validation* approaches, such as system simulations or lab experiments, allow designers to check that their models and controllers conform to real world behavior; *verification* approaches yield formal mathematical proofs that the stability properties hold for *all* possible switching decisions everywhere in the model's infinite state space, not just for finitely-many simulated trajectories.

This paper presents a logic-based, deductive approach for verifying switched system stability under various classes of switching mechanisms. The key insight is that control-theoretic stability arguments for switching control can be formally justified by blending techniques from discrete program verification with continuous differential equations analysis using differential dynamic logic (dL), a logic for deductive verification of hybrid systems [32, 33]. Intuitively, switched systems are modeled in dL as looping *hybrid programs* [45], as in the following snippet ($\{\cdot\}^*$ denotes repetition):

$$
\begin{array}{lll}
\{ & u := ctrl(x); & \text{// switching controller (discrete dynamics)} \\
 & x' = f_u(x) & \text{// actuate decision (continuous dynamics)} \\
\}^* & \texttt{@invariant}(\ \ldots\ ) & \text{// switching loop with invariant annotation}
\end{array}
$$

Accordingly, switched system stability is formally specified in dL as first-order quantified safety properties of switching loops (Section 2.2), and the resulting specifications can then be proved rigorously by combining fundamental ideas from verification and control, namely: *i)* identification of *loop invariants* (`@invariant` above), i.e., properties of the (discrete) loop that are preserved across all executions of the loop body, *ii) compositional verification* for separately analyzing the discrete and continuous dynamics of the loop body, and *iii) Lyapunov functions*, i.e., auxiliary energy functions that enable stability analysis for the continuous dynamics.

Section 3 identifies key loop invariants underlying stability arguments for various classes of switching mechanisms and derives sound stability proof rules for those mechanisms. Crucially, these *syntactic derivations* are built from dL's sound foundations for hybrid program reasoning [32, 33], *without* the need to introduce new

mathematical concepts such as non-classical weak solutions or non-differentiable Lyapunov functions [9, 16]. The remaining practical challenge is how to (automatically) find suitable Lyapunov function candidates for a given switching mechanism; the correctness of any generated candidates can be soundly checked in dL. Section 4 adds support for switched systems in the KeYmaera X prover based on dL [12], including a modeling interface for switched systems, sum-of-squares search for Lyapunov function candidates [30, 36], and fully automatic verification of stability specifications for standard switching mechanisms. Notably, the implementation requires *no extensions* to KeYmaera X's soundness-critical core and thereby directly inherits all of KeYmaera X's correctness guarantees [12, 25]. This trustworthiness is necessary for computer-aided verification of complex switching designs because the number of correctness conditions on their Lyapunov functions scales quadratically with the number of switching modes (Section 3.2), making pen-and-paper proofs error-prone or infeasible. Section 5 further applies the deductive approach on three case studies, chosen because each require subtle twists to standard switched system stability arguments:

- *Longitudinal flight control* [4]: This model is parametric (5 parameters, 2 state variables) and its stability justification due to Branicky uses a "noncustomary" Lyapunov function [4, 10] with intricate arithmetic reasoning. The proof uses *ghost switching*, where virtual switching modes are introduced for the sake of stability analysis, analogous to the use of ghost variables in program verification [29, 33, 34].
- *Automatic cruise control* [28]: This hybrid automaton features switching between several modes based on specific guard conditions: standard/emergency braking, accelerating, and PI control. Lyapunov function candidates can be numerically generated [26], but must be corrected for soundness.
- *Brockett's nonholonomic integrator* [7]: A large class of control systems can be transformed to the nonholonomic integrator but this system is not stabilizable by continuous feedback [7, 22]. The stability argument must account for an initial control mode that drives the system into a suitable region before a stabilizing control law can be applied.

These case studies are verified semi-automatically in KeYmaera X, with user guidance to design and prove modified loop invariants that suitably capture the specific intuitions behind their respective control laws. The flexibility and generality of this paper's deductive approach enables such (modified) stability arguments, while ensuring that every step in the argument is rigorously justified using sound dL logical foundations. All proofs are in the supplement [43].

2 BACKGROUND

This section recalls switched systems and their hybrid program models [45]. It then explains how stability for these models is formally specified and verified using differential dynamic logic (dL) [32, 33].

2.1 Switched Systems as Hybrid Programs

2.1.1 Hybrid Programs. The language of *hybrid programs* is generated by the following grammar, where x is a variable, e is a dL term, and Q is a formula of first-order real arithmetic [32, 33].

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

Continuous dynamics are modeled using systems of ordinary differential equations (ODEs) $x' = f(x) \,\&\, Q$ evolving within domain Q; the ODE is written as $x' = f(x)$ when there is no domain constraint, i.e., $Q \equiv \mathit{true}$. Discrete dynamics are modeled using assignments ($x := e$ assigns the value of term e to x) and tests ($?Q$ checks whether condition Q is true in the current state). The program combinators are used to piece together sub-programs to form programs with hybrid dynamics. The combinators are: sequential composition ($\alpha; \beta$ runs α followed by β), nondeterministic choice ($\alpha \cup \beta$ runs α or β nondeterministically), and nondeterministic repetition (α^* repeats α for any number of iterations).

Throughout this paper, $x = (x_1, \ldots, x_n)$ denotes the vector of continuous state variables for the system under consideration. Other variables are used for program auxiliaries, e.g., to describe memory and timing components of switching controllers.

2.1.2 Switched systems. A *switched system* is described by a finite family $\mathcal{P}$ of ODEs $x' = f_p(x), p \in \mathcal{P}$ and a set of *switching signals* $\sigma : [0, \infty) \to \mathcal{P}$ that prescribe the ODE $x' = f_{\sigma(t)}(x)$ to follow at time t along the system's evolution. Tan and Platzer [45] use hybrid programs as formal models for various classes of switching mechanisms; one example is *arbitrary switching* [22] where the system is allowed to follow *any* switching signal in order to model real world systems whose switching behavior is uncontrolled or *a priori* unknown. The hybrid program $\alpha_{\mathsf{arb}} \equiv \left(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \right)^*$ models arbitrary switching analogously to a computer simulation [45, Proposition 1]: on each loop iteration, the program makes a (discrete) nondeterministic choice of switching decision $\bigcup_{p \in \mathcal{P}} (\,\cdot\,)$ to select an ODE $x' = f_p(x)$ which it then follows continuously for an arbitrarily chosen duration before repeating the simulation loop.

The hybrid programs language can be used to model various other classes of switching mechanisms [22, 45], including general *controlled switching*, as illustrated in Section 1, where a (discrete) control law $u := ctrl(x)$ decides the ODE $x' = f_u(x)$ to switch to on each loop iteration. Stability for these models is explained next.

2.2 Stability as Quantified Loop Safety

This paper studies *uniform global pre-asymptotic stability* (UGpAS) for switched systems [16, 17, 22], defined as follows:

Definition 1 (UGpAS [16, 17]). Let $\Phi(x)$ denote the set of all (domain-obeying) solutions[1] $\varphi : [0, T_\varphi] \to \mathbb{R}^n$ for a switched system from state $x \in \mathbb{R}^n$. The origin $0 \in \mathbb{R}^n$ is:

- **uniformly stable** if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that from all initial states $x \in \mathbb{R}^n$ with $\|x\| < \delta$, all solutions $\varphi \in \Phi(x)$ satisfy $\|\varphi(t)\| < \varepsilon$ for all times $0 \le t \le T_\varphi$,
- **uniformly globally pre-attractive** if, for all $\varepsilon > 0, \delta > 0$, there exists $T \ge 0$ such that from all initial states $x \in \mathbb{R}^n$ with $\|x\| < \delta$, all solutions $\varphi \in \Phi(x)$ satisfy $\|\varphi(t)\| < \varepsilon$ for all times $T \le t \le T_\varphi$, and
- **uniformly globally pre-asymptotically stable** if the system is uniformly stable and uniformly globally pre-attractive.

The UGpAS definition can be understood intuitively for a system with a given switching control mechanism:

[1] A formal construction of the (right-maximal) solution φ for a given switching signal σ is available elsewhere [45, Appendix A].

- *stability* means the mechanism keeps the system close to the origin if the system is initially perturbed close to the origin,
- *global pre-attractivity* means the mechanism drives the system to the origin asymptotically as $t \to \infty$, and
- *uniform* means the stability and pre-attractivity properties are independent of both the nondeterminism in the switching mechanism (e.g., arbitrary switching) and the choice of initial states satisfying $\|x\| < \delta$; for brevity in subsequent sections, "uniform" is elided when describing stability properties.

Remark 1. Switched systems whose solutions are all uniformly bounded in time, i.e., there exists T_m such that for all solutions φ, $T_\varphi \leq T_m$, are trivially pre-attractive. Goebel et al. [16, 17] introduce the notion of *pre-attractivity* as opposed to *attractivity* for hybrid systems because it separates considerations about whether a hybrid system's solutions are *complete*, i.e., solutions exist for all (forward) time, from conditions for stability and attractivity. Pre-attractivity also sidesteps the difficult question of whether a switched system exhibits *Zeno* behavior, i.e., where infinitely many discrete switches occur in finite time [22, 48]. Indeed, it is common in the hybrid and switched systems literature to either *ignore* incomplete solutions or *assume* the models under consideration only have complete solutions [22, 26, 48]. Instead of predicating proofs on these hypotheses, this paper formalizes the (weaker) notion of UGpAS for switched systems, leaving proofs of completeness of solutions out of scope.

The definition of UGpAS nests alternating quantification over real numbers with temporal quantification over the solutions φ of switched systems. This combination of quantifiers can be expressed formally using the formula language of dL [32, 33], whose grammar is shown below, $\sim \in \{=, \neq, \geq, >, \leq, <\}$ is a comparison operator between dL terms $e, \tilde{e}$ and α is a hybrid program:

$$\phi, \psi \ ::= e \sim \tilde{e} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \neg\phi \mid \forall v\, \phi \mid \exists v\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

This grammar extends the first-order language of real arithmetic ($\text{FOL}_\mathbb{R}$) with the box ($[\alpha]\phi$) and diamond ($\langle\alpha\rangle\phi$) modality formulas which express that all or some runs of hybrid program α satisfy postcondition ϕ, respectively. Real arithmetic $\text{FOL}_\mathbb{R}$ is decidable by quantifier elimination [46] and serves as a useful base specification language. Various specifications are equivalently definable in $\text{FOL}_\mathbb{R}$, e.g., Euclidean norm bounds $\|x\| \sim \varepsilon \overset{\text{def}}{\equiv} (\sum_{i=1}^{n} x_i^2) \sim \varepsilon^2$ (for $\varepsilon \geq 0$) and topological operations such as the boundary $\partial\phi$ and closure $\overline{\phi}$ of the set characterized by formula ϕ [3].

The box modality formula $[\alpha]\phi$ expresses *safety* properties ϕ of program α that must hold along all of its executions [33]. When α models a switched system, the box modality quantifies (uniformly) over all times for all solutions arising from the switching mechanism. Accordingly, UGpAS for switched systems is formally specified by nesting the box modality with the first-order quantifiers.

Lemma 2 (UGpAS in differential dynamic logic). *The origin $0 \in \mathbb{R}^n$ for a switched system modeled by program α is UGpAS iff the* dL *formula* $\text{UGpAS}(\alpha)$ *is valid. Variables $\varepsilon, \delta, T, t$ are fresh in α:*

$$\text{UStab}(\alpha) \equiv \forall\varepsilon{>}0\, \exists\delta{>}0\, \forall x\, \big(\|x\| < \delta \to [\alpha]\, \|x\| < \varepsilon\big)$$

$$\text{UGpAttr}(\alpha) \equiv \forall\varepsilon{>}0\, \forall\delta{>}0\, \exists T{\geq}0\, \forall x\, \big(\|x\| < \delta \to [t := 0; \alpha, t' = 1]\, (t \geq T \to \|x\| < \varepsilon)\big)$$

$$\text{UGpAS}(\alpha) \equiv \text{UStab}(\alpha) \wedge \text{UGpAttr}(\alpha)$$

Here, $\text{UStab}(\alpha)$ *and* $\text{UGpAttr}(\alpha)$ *characterize stability and global pre-attractivity of α, respectively. In* $\text{UGpAttr}(\alpha)$*, $\alpha, t' = 1$ denotes the hybrid program obtained from α by augmenting its continuous dynamics so that variable t tracks the progression of time.*

Formulas $\text{UStab}(\alpha)$ and $\text{UGpAttr}(\alpha)$ syntactically formalize in dL the corresponding quantifiers in Def. 1. In $\text{UGpAttr}(\alpha)$, the fresh clock variable t is initialized to 0 and syntactically tracks the progression of time along switched system solutions. The program $\alpha, t' = 1$ can, e.g., be constructed by adding a clock ODE $t' = 1$ to all ODEs in the switched system model α. Accordingly, the postcondition $t \geq T \to \|x\| < \varepsilon$ expresses that the system state norm is bounded by ε after T time units along any switching trajectory, as required in Def. 1. Various other stability notions are of interest in the continuous and hybrid systems literature [13, 17, 22, 28, 35, 42, 44]. These variations can also be formally specified in dL [44] but are left out of scope for this paper.

2.3 Proof Calculus

The dL proof calculus enables formal, deductive verification of UGpAS stability specifications through compositional reasoning principles for hybrid programs [32, 33] and a complete axiomatization for ODE invariants [34]. For example, an important syntactic tool for differential equations reasoning is the *Lie derivative* of term e along ODE $x' = f(x)$, defined as $\mathcal{L}_f(e) \overset{\text{def}}{=} \nabla e \cdot f$. The sound calculation and manipulation of Lie derivatives is enabled in dL through the use of syntactic differentials [32].

All proofs are presented in a classical sequent calculus with the usual rules for manipulating logical connectives and sequents. The semantics of *sequent* $\Gamma \vdash \phi$ is equivalent to the formula $(\bigwedge_{\psi\in\Gamma} \psi) \to \phi$ and a sequent is *valid* iff its corresponding formula is valid. The key (derived) dL proof rule used in this paper is:

$$\text{loop}\ \frac{\Gamma \vdash \mathit{Inv} \quad \mathit{Inv} \vdash [\alpha]\, \mathit{Inv} \quad \mathit{Inv} \vdash \phi}{\Gamma \vdash [\alpha^*]\phi}$$

The loop rule says that, in order to prove validity of the conclusion (below the rule bar), it suffices to prove the three premises (above the rule bar), respectively from left to right: *i)* the initial assumptions Γ imply *Inv*, *ii) Inv* is preserved across the loop body α, i.e., *Inv* is a *loop invariant* for α^*, and *iii) Inv* implies the postcondition ϕ. The identification of loop invariants *Inv* is crucial for formal proofs of UGpAS, as illustrated by the following deductive proof skeleton for stability (a similar skeleton is used for pre-attractivity):

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1 \vdash \phi_1 \quad \cdots \quad \Gamma_k \vdash \phi_k}{\vdots\ (\text{hybrid program reasoning for } \alpha)}}{\Gamma \vdash \mathit{Inv} \qquad \mathit{Inv} \vdash [\alpha]\, \mathit{Inv} \qquad \mathit{Inv} \vdash \|x\| < \varepsilon}\ \text{loop}}{\Gamma \vdash [\alpha^*]\, \|x\| < \varepsilon}}{\vdots\ (\text{logic/arithmetic reasoning for } \Gamma)}}{\vdash \text{UStab}(\alpha^*)}\ \textbf{Deduction}$$

Proofs proceed upwards by deduction, where each reasoning step is justified by sound dL axioms and rules of inference, e.g., the loop rule. The proof skeleton above syntactically *derives* a proof rule that reduces a stability proof for α^* to proofs of its top-most premises, $\Gamma_1 \vdash \phi_1 \ \cdots \ \Gamma_k \vdash \phi_k$. These correspond to required logical

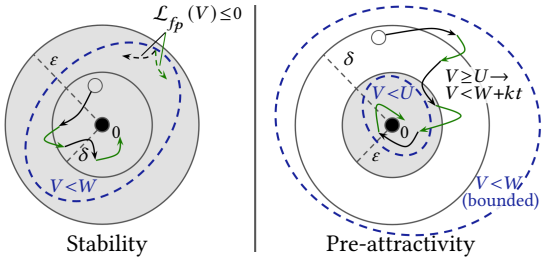


Figure 1: Loop invariants for UGpAS (arbitrary switching), stability (left) and pre-attractivity (right). Switching trajectories are illustrated by alternating black and green arrows.

and arithmetical conditions on Lyapunov functions for various switching mechanisms. The loop invariant step (highlighted in red) crucially ties together these conditions on Lyapunov functions and hybrid program reasoning for switched systems.

3 LOOP INVARIANTS FOR SWITCHED SYSTEM STABILITY

This section identifies loop invariants for proving UGpAS under various classes of switching mechanisms with Lyapunov functions [5, 21, 22]; relevant mathematical arguments are presented briefly (see supplement [43]). Throughout the section, loop invariants are progressively tweaked to account for new design insights behind increasingly complex switching mechanisms.

3.1 Arbitrary and State-Dependent Switching

3.1.1 Arbitrary Switching. Stability for the arbitrary switching model $\alpha_{\mathtt{arb}}$ from Section 2 can be verified by finding a so-called *common Lyapunov function* V for all of the ODEs $x' = f_p(x), p \in \mathcal{P}$ satisfying the following arithmetical conditions [22, 42]:

i) $V(0) = 0$ and $V(x) > 0$ for all $\|x\| > 0$,
ii) V is *radially unbounded*, i.e., for all b, there exists $\gamma > 0$ such that $\|x\| < \gamma$ for all $V(x) \leq b$, and
iii) for each ODE $x' = f_p(x), p \in \mathcal{P}$, the Lie derivative $\mathcal{L}_{f_p}(V)$ satisfies: $\mathcal{L}_{f_p}(V)(0) = 0$ and $\mathcal{L}_{f_p}(V)(x) < 0$ for all $\|x\| > 0$.

Conditions *i)*–*iii)* are generalizations of well-known conditions for stability of ODEs [8, 21] to arbitrary switching. Intuitively, conditions *i)* and *iii)* ensure that V acts as an auxiliary energy function whose value decreases asymptotically to zero (at the origin) along all switching trajectories of the system; the radial unboundedness condition *ii)* ensures that this argument applies to all system states for *global* pre-attractivity [21]. Correctness of these conditions can be proved in dL using loop invariants, see Fig. 1 (explained below).

Stability. The specification $\mathrm{UStab}(\alpha_{\mathtt{arb}})$ requires that all trajectories of $\alpha_{\mathtt{arb}}$ stay in the grey ball $\|x\| < \varepsilon$, starting from a chosen ball $\|x\| < \delta$, see Fig. 1 (left). Condition *i)* guarantees that the ball $\|x\| < \varepsilon$ contains (a connected component of) the sublevel set $V<W$ for some $W>0$ (dashed blue curve) and this sublevel set contains a smaller ball $\|x\| < \delta$ [8, 21]. Condition *iii)* shows that this sublevel set is invariant for each ODE $x' = f_p(x), p \in \mathcal{P}$ because $\mathcal{L}_{f_p}(V)(x) \leq 0$, as illustrated by the dashed black and green arrows for two different switching choices $p \in \mathcal{P}$ both locally pointing inwards on the boundary of the sublevel set. Thus, the formula $Inv_s \equiv \|x\| < \varepsilon \wedge V < W$, which characterizes the blue sublevel set, is an invariant for all possible switching choices in the loop body of $\alpha_{\mathtt{arb}}$, which makes Inv_s a suitable loop invariant for $\mathrm{UStab}(\alpha_{\mathtt{arb}})$.

Pre-attractivity. The specification $\mathrm{UGpAttr}(\alpha_{\mathtt{arb}})$ requires that all trajectories of $\alpha_{\mathtt{arb}}$ stay in the grey ball $\|x\| < \varepsilon$ after a chosen time T, starting from the initial ball $\|x\| < \delta$, see Fig. 1 (right). The ball $\|x\| < \delta$ is bounded, so it is contained in a sublevel set satisfying $V < W$ for some $W > 0$ (outer dashed blue curve); this sublevel set is bounded by condition *ii)*. Like the stability argument, condition *i)* guarantees that there is a sublevel set $V < U$ for some $U > 0$ (inner dashed blue curve) contained in the ball $\|x\| < \varepsilon$, and condition *iii)* shows that the sublevel sets characterized by $V < W$ and $V < U$ are both invariants for every ODE in the loop body of $\alpha_{\mathtt{arb}}$. The set characterized by formula $V \geq U \wedge V \leq W$ is compact and bounded away from the origin, which implies by condition *iii)* that there is a uniform bound $k < 0$ on this set, where for each ODE $x' = f_p(x), p \in \mathcal{P}$, $\mathcal{L}_{f_p}(V)(x) \leq k$. Thus, the value of Lyapunov function V decreases at rate k, regardless of switching choices in the loop body of $\alpha_{\mathtt{arb}}$, *as long as* it has not entered $V < U$. The loop invariant for $\mathrm{UGpAttr}(\alpha_{\mathtt{arb}})$ syntactically expresses this intuition: $Inv_a \equiv V < W \wedge (V \geq U \rightarrow V < W + kt)$. For a sufficiently large choice of T with $W + kT \leq U$, trajectories at time $t \geq T$ satisfy $V < U$ so they are contained in the $\|x\| < \varepsilon$ ball.

The loop invariants identified above enable derivation of a formal dL stability proof rule for $\alpha_{\mathtt{arb}}$ (deferred to a more general version in Corollary 3 below). In fact, since arbitrary switching is the most permissive form of switching [22], UGpAS for any switching mechanism can be soundly justified using the loop invariants above in case a suitable common Lyapunov function can be found.

3.1.2 State-dependent Switching. The state-dependent switching mechanism [22] constrains arbitrary switching by allowing execution of (and switching to) an ODE $x' = f_p(x), p \in \mathcal{P}$ only when the system state is in domain Q_p. This is modeled by the hybrid program $\alpha_{\mathtt{state}} \equiv \left(\bigcup_{p\in\mathcal{P}} x' = f_p(x) \,\&\, Q_p\right)^*$ [45, Proposition 2], where arbitrary switching $\alpha_{\mathtt{arb}}$ corresponds to the special case with $Q_p \equiv \mathit{true}$ for all $p \in \mathcal{P}$.

The same loop invariants for $\alpha_{\mathtt{arb}}$ are used for $\alpha_{\mathtt{state}}$ to derive the following proof rule. For brevity, premises of all derived stability proof rules are implicitly conjunctively quantified over $p \in \mathcal{P}$.

Corollary 3 (UGpAS for state-dependent switching, CLF). *The following proof rule for common Lyapunov function V with three stacked premises is syntactically derivable in* dL.

$$\mathrm{CLF}\ \frac{\begin{array}{l}\vdash V(0) = 0 \wedge \forall x\,(\|x\| > 0 \rightarrow V(x) > 0)\\ \vdash \forall b\,\exists\gamma\,\forall x\,(V(x) \leq b \rightarrow \|x\| \leq \gamma)\\ \vdash \mathcal{L}_{f_p}(V)(0) = 0 \wedge \forall x\,(\|x\| > 0 \wedge \overline{Q_p} \rightarrow \mathcal{L}_{f_p}(V)(x) < 0)\end{array}}{\vdash \mathrm{UGpAS}(\alpha_{\mathtt{state}})}$$

Corollary 3 syntactically derives a slight generalization of conditions *i)*–*iii)* from Section 3.1.1 for $\alpha_{\mathtt{state}}$, where the Lie derivatives $\mathcal{L}_{f_p}(V)(x)$ for each $p \in \mathcal{P}$ are required to be negative on their respective domain closures[2] $\overline{Q_p}$. This generalization is justified by the

[2]The topological closure $\overline{Q}$ of domain Q is needed for soundness of a technical compactness argument used in the pre-attractivity proof (see supplement [43]).

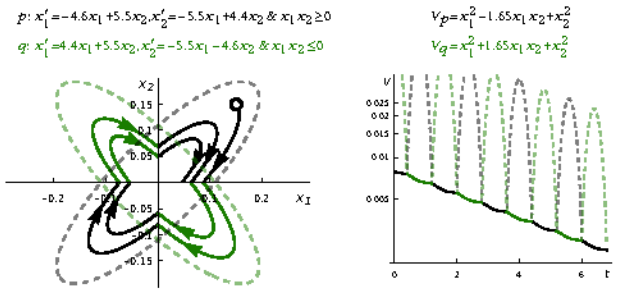


Figure 2: A switching trajectory for Example 7 from Section 4.2 with state-dependent switching (left) and the value of two Lyapunov functions along that trajectory (right, log-scale on vertical axis). Solid lines indicate the active Lyapunov function at time t. Two sublevel sets $V_p, V_q < W = 0.012$ are shown dashed on the left within which the switching trajectory is respectively trapped at any given time.

same loop invariants in Section 3.1.1 because the ODE invariance properties are only required to hold in their respective domains.

The domain asymmetry in α_{state} suggests another way of generalizing the stability arguments, namely, through the use of *multiple Lyapunov functions*, where a (possibly) different Lyapunov function V_p is associated to each $p \in \mathcal{P}$ [5]. Here, the function V_p is responsible for justifying stability within domain Q_p, i.e., its value decreases along system trajectories whenever the system is within Q_p, as illustrated in Fig. 2. Constraints on these functions are obtained by modifying the loop invariants to account for this intuition.

Stability. The stability loop invariant is modified by case splitting disjunctively on the domains $Q_p, p \in \mathcal{P}$, and requiring that the sublevel set characterized by $V_p < W$ is invariant within its respective domain Q_p: $Inv_s \equiv \|x\| < \varepsilon \wedge \bigvee_{p\in\mathcal{P}} (Q_p \wedge V_p < W)$. Similar to Section 3.1.1, the bound W is chosen so that each sublevel set characterized by $V_p < W$ is contained in the ball $\|x\| < \varepsilon$.

Pre-attractivity. The pre-attractivity loop invariant is similarly modified by disjunctively requiring that each V_p decreases along system trajectories when the system is in their respective domains Q_p: $Inv_a \equiv \bigvee_{p\in\mathcal{P}} (Q_p \wedge V_p < W \wedge (V_p \geq U \rightarrow V_p < W + kt))$. The constants U, W, k, T are chosen as appropriate lower or upper bounds for all the Lyapunov functions (see proof of Corollary 4).

Arithmetical conditions for the Lyapunov functions $V_p, p \in \mathcal{P}$ are derived from the modified invariants in the following rule.

Corollary 4 (UGpAS for state-dependent switching, MLF). *The following proof rule for multiple Lyapunov functions $V_p, p \in \mathcal{P}$ with four stacked premises is syntactically derivable in* dL.

$$\text{MLF}\ \frac{\begin{array}{l}\vdash V_p(0) = 0 \wedge \forall x\,(\|x\| > 0 \rightarrow V_p(x) > 0)\\ \vdash \forall b\,\exists\gamma\,\forall x\,(V_p(x) \leq b \rightarrow \|x\| \leq \gamma)\\ \vdash \mathcal{L}_{f_p}(V_p)(0){=}0 \wedge \forall x\,(\|x\|{>}0 \wedge \overline{Q_p} \rightarrow \mathcal{L}_{f_p}(V_p)(x){<}0)\\ \vdash \bigwedge_{q\in\mathcal{P}} \left(Q_p \wedge Q_q \rightarrow V_p = V_q\right)\end{array}}{\vdash \text{UGpAS}(\alpha_{\text{state}})}$$

The top three premises of Corollary 4 are similar to those of Corollary 3, but are now required to hold for each Lyapunov function $V_p, p \in \mathcal{P}$ separately. The (new) bottom premise corresponds to a compatibility condition between the Lyapunov functions arising from the loop invariants. For example, consider the stability loop invariant (similarly for pre-attractivity) and suppose the system currently satisfies disjunct $Q_p \wedge V_p < W$ with V_p justifying stability in domain Q_p. If the system switches to the ODE $x' = f_q(x)$ within domain Q_q, then Lyapunov function V_q becomes the active Lyapunov function which must satisfy $V_q < W$ to preserve the stability loop invariant. The premise $Q_p \wedge Q_q \rightarrow V_p = V_q$ says that the Lyapunov functions V_p, V_q are equal whenever such a switch is possible (in either direction), i.e., when their domains overlap.

3.2 Controlled Switching

This section turns to *controlled switching* models [45], where an explicit controller program is responsible for making logical switching decisions between the ODEs $x' = f_p(x), p \in \mathcal{P}$. This is in contrast to earlier models $\alpha_{\text{arb}}, \alpha_{\text{state}}$ which exhibit *autonomous switching*, i.e., without an explicit control logic [6, 22]. General controlled switching is modeled by the hybrid program α_{ctrl}:

$$\alpha_{\text{ctrl}} \equiv \underset{\text{initialization}}{\alpha_i}; \Big(\overbrace{\alpha_u}^{\text{switching controller}}; \overbrace{\bigcup_{p\in\mathcal{P}} \left(?u = p; x' = f_p(x,y), y' = g_p(x,y)\ \&\ Q_p\right)}^{\alpha_p\ \text{(plant, actuate decision)}}\Big)^*$$

The model α_{ctrl} uses three subprograms: α_i initializes the system, then α_u (modeling the switching controller) and α_p (modeling the continuous plant dynamics) are run in a switching loop. The discrete programs α_i, α_u decide on values for the control output $u = p, p \in \mathcal{P}$ and the program α_p responds to this output by evolving the corresponding ODE $x' = f_p(x,y), y' = g_p(x,y)\ \&\ Q_p$. The programs α_i, α_u must not modify the system state variables x, but they may modify other auxiliaries, including *auxiliary continuous state* variables y used to model timers or integral terms used in controllers, see Section 5.2. This control-plant loop is a typical structure for hybrid systems modeled in dL [31, 33], e.g., the controller α_u below models the discrete switching logic present in hybrid automata [6, 18, 31] (without jumps in the system state):

$$\alpha_u \equiv \bigcup_{p\in\mathcal{P}} \left(?u = p; \bigcup_{q\in\mathcal{P}} \left(?G_{p,q}; R_{p,q}; u := q\right)\right) \qquad (1)$$
$$R_{p,q} \equiv y_1 := e_1; y_2 := e_2; \ldots; y_k := e_k$$

For each mode $p \in \mathcal{P}$, the switching controller may nondeterministically switch to mode $q \in \mathcal{P}$ if the *guard* formula $G_{p,q}$ is true in the current state ($G_{p,p} \equiv true$ for self-transitions); if the transition is taken, the *reset map* $R_{p,q}$ sets the values of auxiliary state variables $y_1, \ldots, y_k$ respectively to the value of terms $e_1, \ldots, e_k$.

Stability analysis for controlled switching proceeds by identifying suitable loop invariants Inv for α_{ctrl}. A powerful proof technique applied here is *compositional reasoning* [31, 33] which separately analyses the discrete (α_i, α_u) and continuous (α_p) dynamics, and then lifts those results to the full hybrid dynamics. This idea is exemplified by the following derived variation of the loop rule:

$$\text{loopT}\ \frac{\Gamma \vdash [\alpha_i]\,Inv \quad Inv \vdash [\alpha_u]\,Inv \quad Inv \vdash [\alpha_p]\,Inv \quad Inv \vdash \phi}{\Gamma \vdash [\alpha_i;(\alpha_u;\alpha_p)^*]\phi}$$

The premises of rule loopT say that system initialization α_i puts the system in a state satisfying the invariant Inv, and that Inv is

compositionally preserved by *both* the discrete switching logic α_u and the continuous dynamics α_p. This rule is applied to analyze stability for two important special instances of $\alpha_{\texttt{ctrl}}$ next.

3.2.1 Guarded State-dependent Switching. The instance $\alpha_{\texttt{guard}}$ corresponds to the automata controller from (1) with $\alpha_i \equiv \bigcup_{p\in\mathcal{P}} u := p$ and guard formulas $G_{p,q}$. It does not use auxiliaries y nor the reset map $R_{p,q}$. This model adds *hysteresis* [19] to the state-dependent switching model from Section 3.1.2, so that switching decisions at each $G_{p,q}$ depend explicitly on the current discrete mode u in addition to the continuous state. This design change is reflected in the loop invariants and in the corresponding proof rule below.

Stability. The stability loop invariant is modified (cf. Section 3.1.2) to case split on the possible discrete modes $u = p$ rather than the ODE domains: $Inv_s \equiv \|x\| < \varepsilon \wedge \bigvee_{p\in\mathcal{P}} \left(u = p \wedge V_p < W\right)$.

Pre-attractivity. The pre-attractivity loop invariant is modified similarly: $Inv_a \equiv \bigvee_{p\in\mathcal{P}} \left(u{=}p \wedge V_p{<}W \wedge (V_p \geq U \rightarrow V_p < W{+}kt)\right)$.

Corollary 5 (UGpAS for guarded state-dependent switching, MLF). *The following proof rule for multiple Lyapunov functions $V_p, p \in \mathcal{P}$ with four stacked premises is syntactically derivable in* dL.

$$\mathrm{MLF}_G\ \frac{\begin{array}{l}\vdash V_p(0) = 0 \wedge \forall x\,(\|x\| > 0 \rightarrow V_p(x) > 0)\\ \vdash \forall b\,\exists\gamma\,\forall x\,(V_p(x) \leq b \rightarrow \|x\| \leq \gamma)\\ \vdash \mathcal{L}_{f_p}(V_p)(0){=}0 \wedge \forall x\,(\|x\|{>}0 \wedge \overline{Q_p} \rightarrow \mathcal{L}_{f_p}(V_p)(x){<}0)\\ \vdash \bigwedge_{q\in\mathcal{P}} \left(G_{p,q} \rightarrow V_q \leq V_p\right)\end{array}}{\vdash \mathrm{UGpAS}(\alpha_{\texttt{guard}})}$$

The premises of rule MLF_G are identical to those from MLF except the bottom premise, which derives from loopT and unfolding the controller α_u with dL's hybrid program axioms, e.g., the following proof skeleton shows the unfolding for the stability loop invariant Inv_s corresponding to a switch from mode p to mode q:

$$\textbf{Unfold}\ \uparrow\ \frac{\dfrac{\dfrac{\vdash G_{p,q} \rightarrow V_q \leq V_p}{V_p < W \vdash G_{p,q} \rightarrow V_q < W}\ \textbf{Arithmetic}\ \uparrow}{u = p \wedge V_p < W \vdash [?G_{p,q}; u := q](u = q \wedge V_q < W)}}{Inv_s \vdash [\alpha_u] Inv_s}$$

Unlike rule MLF, the bottom premise of rule MLF_G only uses an inequality, because the guards $G_{p,q}$ determine permissible switching.

3.2.2 Time-dependent Switching. The instance $\alpha_{\texttt{time}}$ shown below models *time-dependent switching*, where the controller α_u makes switching decisions based on the time τ elapsed in each mode.

$$\alpha_{\texttt{time}} \equiv \begin{cases} \alpha_i \equiv \tau := 0; \bigcup\limits_{p\in\mathcal{P}} u := p \\ \alpha_u \equiv \bigcup\limits_{p\in\mathcal{P}} \left(?u = p; \bigcup\limits_{q\in\mathcal{P}} (?\theta_{p,q} \leq \tau; \tau := 0; u := q)\right) \\ \alpha_p \equiv \bigcup\limits_{p\in\mathcal{P}} (?u = p; x' = f_p(x), \tau' = 1 \,\&\, \tau \leq \Theta_p) \end{cases}$$

The controller α_u enables switching from mode p to q when a *minimum* dwell time $0 \leq \theta_{p,q} \leq \tau$ has elapsed and resets the timer whenever such a switch occurs. Conversely, the plant α_p restricts modes with a *maximum* dwell time $\tau \leq \Theta_p, \Theta_p > 0$; an unbounded dwell time $\Theta_p = \infty$ is represented by the domain constraint *true*. Dwell time restrictions can be used to stabilize systems that switch between stable *and unstable* modes [47]. Intuitively, the system should stay in stable modes for sufficient duration ($\theta_{p,q} \leq \tau$) while it should avoid staying in unstable modes for too long ($\tau \leq \Theta_p$).

To reason about stability for $\alpha_{\texttt{time}}$, consider Lyapunov function conditions $\mathcal{L}_{f_p}(V_p)(x) \leq -\lambda_p V_p$, where λ_p is a constant associated with each mode $p \in \mathcal{P}$. This condition bounds the value of V_p along the solution of $x' = f_p(x)$ by either a decaying exponential for stable modes ($\lambda_p > 0$) or a growing exponential for unstable modes ($\lambda_p \leq 0$). Let $\mathcal{S} = \{p \in \mathcal{P}, \lambda_p > 0\}$ and $\mathcal{U} = \{p \in \mathcal{P}, \lambda_p \leq 0\}$ be the indexes of the stable and unstable modes in the loop invariants below, and let $e^{(\cdot)}$ denote the real exponential function, which is definable in dL by differential axiomatization [31, 34].

Stability. The stability loop invariant expresses the required exponential bounds with a case split depending if $p \in \mathcal{S}$ or $p \in \mathcal{U}$:

$$Inv_s \equiv \tau \geq 0 \wedge \|x\| < \varepsilon \wedge \left(\begin{array}{l}\bigvee\limits_{p\in\mathcal{S}} \left(u = p \wedge V_p < We^{-\lambda_p\tau}\right) \vee \\ \bigvee\limits_{p\in\mathcal{U}} \left(u = p \wedge V_p < We^{-\lambda_p(\tau-\Theta_p)} \wedge \tau \leq \Theta_p\right)\end{array}\right)$$

For $p \in \mathcal{S}$, $e^{-\lambda_p\tau}$ is the accumulated decay factor for V_p after staying in the stable mode for time τ. For $p \in \mathcal{U}$, $e^{-\lambda_p(\tau-\Theta_p)}$ is a buffer factor for the growth of V_p in the unstable mode so that $V_p < W$ still holds at the maximum dwell time $\tau = \Theta_p$. In both cases, the internal timer variable is non-negative ($\tau \geq 0$).

Pre-attractivity. The pre-attractivity loop invariant has similar exponential decay and growth bounds for each $p \in \mathcal{P}$ in the current mode. In addition, it has an overall exponential decay term $e^{-\sigma(t-\tau)}$ for some $\sigma > 0$, which ensures that the value of V_p tends to 0 as $t \rightarrow \infty$ for all switching trajectories; recall t is the global clock introduced in the specification of pre-attractivity in Lemma 2.

$$Inv_a \equiv \tau \geq 0 \wedge t \geq \tau \wedge \left(\begin{array}{l}\bigvee\limits_{p\in\mathcal{S}} \left(u = p \wedge V_p < We^{-\sigma(t-\tau)}e^{-\lambda_p\tau}\right) \vee \\ \bigvee\limits_{p\in\mathcal{U}} \left(u = p \wedge V_p < We^{-\sigma(t-\tau)}e^{-\lambda_p(\tau-\Theta_p)} \wedge \tau \leq \Theta_p\right)\end{array}\right)$$

Intuitively, $e^{-\sigma(t-\tau)}$ is the accumulated *overall* decay factor for V_p *until* the switch to mode p which occurred at time $t - \tau$, while $e^{-\lambda_p\tau}$ (resp. $e^{-\lambda_p(\tau-\Theta_p)}$) is the *current* decay (resp. growth) factor *since* the switch to mode p.

Corollary 6 (UGpAS for time-dependent switching, MLF). *The following proof rule for multiple Lyapunov functions $V_p, p \in \mathcal{P}$ with five stacked premises is syntactically derivable in* dL.

$$\mathrm{MLF}_\tau\ \frac{\begin{array}{l}\vdash V_p(0) = 0 \wedge \forall x\,(\|x\| > 0 \rightarrow V_p(x) > 0)\\ \vdash \forall b\,\exists\gamma\,\forall x\,(V_p(x) \leq b \rightarrow \|x\| \leq \gamma)\\ \vdash \mathcal{L}_{f_p}(V_p) \leq -\lambda_p V_p\\ Inv_s \vdash [\alpha_u] Inv_s \qquad Inv_a \vdash [\alpha_u] Inv_a\end{array}}{\vdash \mathrm{UGpAS}(\alpha_{\texttt{time}})}$$

The two red premises on the bottom row are expanded to arithmetical conditions on V_p by unfolding the program structure of α_u with dL *axioms in the supplement [43].*

The bottom premises of MLF_τ and MLF_G exemplify a key benefit of dL stability reasoning: conditions on V_p that arise from Inv_s, Inv_a are derived by systematically unfolding the discrete dynamics of α_u with sound dL axioms. This enables automatic, *correct-by-construction* derivation of those conditions, which is especially important for controlled switching because the number of possible transitions scales quadratically $|\mathcal{P}|^2$ with the number of modes $|\mathcal{P}|$.

4 KEYMAERA X IMPLEMENTATION

This section presents a prototype implementation of switched systems support in the KeYmaera X prover based on dL [12]. The implementation consists of ≈2700 lines and, crucially, does not require any extension to KeYmaera X's existing soundness-critical core. Accordingly, verification results for switched systems obtained through this implementation directly inherit the strong correctness properties guaranteed by the design of KeYmaera X [12, 25].

4.1 Modeling and Proof Interface

The implementation builds on KeYmaera X's proof IDE [24] to provide a convenient interface for modeling switching mechanisms, as shown in Fig. 3. The interface allows users to express switching mechanisms intuitively by rendering automaton plots while abstracting away the underlying hybrid programs. It provides templates for switched systems following the switching mechanisms of Section 3: state-dependent, guarded, timed, and general controlled switching (tabs "Autonomous", "Guarded", "Timed", "Generic" in Fig. 3). From these templates, KeYmaera X automatically generates programs and stability specifications, ensuring that they have the correct dL hybrid program and formula structure.

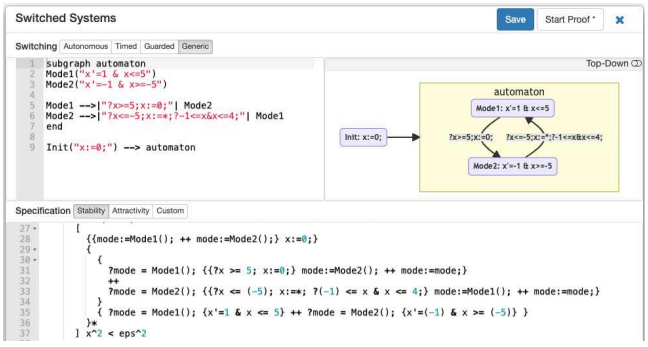


Figure 3: Screenshot of the KeYmaera X switched systems modeling editor: automata input on top-left, rendered automaton top-right, generated hybrid program and specification(s) in dL at the bottom

Switched systems are represented internally with a common interface `SwitchedSystem` which is currently implemented by four classes: `StateDependent` $\alpha_{\texttt{state}}$, `Guarded` $\alpha_{\texttt{guard}}$, `Timed` $\alpha_{\texttt{time}}$, and `Controlled` $\alpha_{\texttt{ctrl}}$. The `SwitchedSystem` interface provides default stability and pre-attractivity specifications, which can be adapted by users on the UI if needed. Corollaries 3–6 are implemented as UGpAS *proof tactics* in KeYmaera X's Bellerophon tactic language [11]. These tactics automate all of the reasoning steps underlying stability proofs for their respective switching mechanisms, so that users only need to input candidate Lyapunov functions for KeYmaera X to (attempt to) complete their proofs. Additionally, when candidates are not provided by the user, the implementation uses sum-of-squares programming [30, 36] to automatically generate candidate Lyapunov functions for a subset of switching designs. The generated candidates are checked for correctness by KeYmaera X so the generator does not need to be trusted for correctness of the resulting proofs. Table 1 summarizes the available proof tactics and Lyapunov function generation for classes of switching mechanisms.

Table 1: Available tactics in KeYmaera X for switched systems stability proofs and Lyapunov function generation.

`SwitchedSystem`	Common Lyap. Proof	Common Lyap. Gen.	Multiple Lyap. Proof	Multiple Lyap. Gen.
`StateDependent` $\alpha_{\texttt{state}}$	✓	✓	✓	✓
`Guarded` $\alpha_{\texttt{guard}}$	✓	✓	✓	✓
`Timed` $\alpha_{\texttt{time}}$	✓	✓	✓	—
`Controlled` $\alpha_{\texttt{ctrl}}$	✓	✓	—	—

Table 2: Stability proofs for examples drawn from the literature. The "Time" columns indicate time (in seconds) to run the KeYmaera X proofs, × indicates incomplete proof. A ✓ in the "Gen." column indicates successful Lyapunov function(s) generation, ? indicates that a candidate was generated but with numerical issues, and — indicates inapplicability. In the latter two cases (?, —) known Lyapunov functions from the literature were used for the proofs (if available).

Example	Model	Time (Stab.)	Time (Attr.)	Gen.
1 [5, Ex. 2.1]	$\alpha_{\texttt{state}}$	2.6	3.0	✓
2 [19, Motiv. ex.]	$\alpha_{\texttt{state}}$	2.2	2.3	✓
3 [19, Ex. 1]	$\alpha_{\texttt{state}}$	3.3	4.1	✓
4 [19, Ex. 2 & 3]	$\alpha_{\texttt{guard}}$	2.8	3.8	?
5 [36, Ex. 6]	$\alpha_{\texttt{guard}}$	×	×	?
6 [42, Ex. 2.45]	$\alpha_{\texttt{arb}}$	19.4	11.1	✓
7 [42, Ex. 3.25]	$\alpha_{\texttt{state}}$	2.4	2.9	✓
8 [42, Ex. 3.49]	$\alpha_{\texttt{time}}$	4.4	5.6	—
9 [47, Ex. 1]	$\alpha_{\texttt{time}}$	4.7	5.3	—
10 [47, Ex. 2]	$\alpha_{\texttt{time}}$	256.9	×	—

4.2 Examples

The implementation is tested on a suite of examples drawn from the literature [5, 19, 36, 42] featuring various switching mechanisms, with results summarized in Table 2. These examples have a 2 dimensional state space and switch between 2 modes except Example 6 (3 dimensions, 2 modes) and Example 4 (2 dimensions, 4 modes).

The proof tactics successfully prove most of the examples across various switching mechanisms. For Example 5, a suitable Lyapunov function (without numerical errors) could not be found. For the time-dependent switching models (Examples 8–10), KeYmaera X internally uses verified polynomial Taylor approximations to the exponential function for decidability of arithmetic [3, 46]; Example 10 needs a high degree approximation (15 terms in the polynomial) for

sufficient accuracy and its attractivity proof could not be completed in reasonable time.

5 CASE STUDIES

This section presents three case studies applying the deductive verification approach to justify various non-standard stability arguments in KeYmaera X.[3]

5.1 Canonical Max System

Branicky [4] investigates the longitudinal dynamics of an aircraft with an elevator controller that mediates between two control objectives: *i)* tracking potentially unsafe pilot input and *ii)* respecting safety constraints on the aircraft's angle of attack. Assuming a state feedback control law, the model is transformed to the following *canonical max system* [4, Remark 5], with state variables x, y and parameters a, b, f, g, γ satisfying $a, b, a-f, b-g > 0$ and $\gamma \leq 0$.

$$x' = y, y' = -ax - by + \max(fx + gy + \gamma, 0) \tag{2}$$

The right-hand side of system (2) is non-differentiable but the equations can be equivalently rewritten as a family of two ODEs corresponding to either possibility for the $\max(fx+gy+\gamma, 0)$ term in the equation for y' as follows, where the system follows ODE Ⓐ in domain $fx+gy+\gamma \leq 0$ and ODE Ⓑ in domain $fx+gy+\gamma \geq 0$.

$$Ⓐ \equiv x' = y, y' = -ax - by$$
$$Ⓑ \equiv x' = y, y' = -(a-f)x - (b-g)y + \gamma$$

Stability of this parametric system is *not* directly provable using standard techniques for state-dependent switching presented in Section 3.1.2. For example, the ODE Ⓐ stabilizes the system to the origin but the ODE Ⓑ stabilizes to the point $(-\frac{\gamma}{a-f}, 0)$, away from the origin for $\gamma < 0$. Branicky proves global asymptotic stability of (2) with the following "noncustomary" [10] Lyapunov function involving a nondifferentiable integrand:

$$V = \frac{1}{2}y^2 + \int_0^x a\xi - \max(f\xi + \gamma, 0)d\xi \tag{3}$$

The key idea used to deductively prove stability here instead is *ghost switching*: analogous to ghost variables in program verification which are added for the sake of program proofs [29, 33, 34], ghost switching modes do not change the physical dynamics of the system but are introduced for the purposes of the stability analysis. Here, ghost switching between $fx+\gamma \leq 0$ and $fx+\gamma \geq 0$ is used to obtain closed form representations for the integral in (3). This yields an instance of state-dependent switching $\alpha_{\texttt{state}}$ with 4 switching modes and the corresponding stability specification P_m:

$$\alpha_m \equiv \left(Ⓐ_1 \cup Ⓐ_2 \cup Ⓑ_1 \cup Ⓑ_2\right)^* \quad p \equiv fx+gy+\gamma \quad q \equiv fx+\gamma$$
$$Ⓐ_1 \equiv Ⓐ \,\&\, p \leq 0 \wedge q \leq 0 \qquad Ⓐ_2 \equiv Ⓐ \,\&\, p \leq 0 \wedge q \geq 0$$
$$Ⓑ_1 \equiv Ⓑ \,\&\, p \geq 0 \wedge q \leq 0 \qquad Ⓑ_2 \equiv Ⓑ \,\&\, p \geq 0 \wedge q \geq 0$$
$$P_m \equiv a{>}0 \wedge b{>}0 \wedge a{-}f{>}0 \wedge b{-}g{>}0 \wedge f{\neq}0 \wedge \gamma{\leq}0 \rightarrow \text{UGpAS}(\alpha_m)$$

The ghost switching modes enable a multiple Lyapunov function argument for stability using the following modified closed-form representations of Branicky's Lyapunov function (3), with $V_1 = \frac{1}{2}(bcx^2 + 2cxy + y^2) + \frac{a}{2}x^2$ for $Ⓐ_1, Ⓑ_1$ and $V_2 = \frac{1}{2}(bcx^2 + 2cxy + y^2) + \frac{a}{2}x^2 - \frac{(fx+\gamma)^2}{2f}$ for $Ⓐ_2, Ⓑ_2$.[4] The sub-terms highlighted in red for V_1, V_2 are closed form expressions for $\int_0^x a\xi - \max(f\xi+\gamma, 0)d\xi$ where $f\xi+\gamma \leq 0$ and $f\xi+\gamma \geq 0$ respectively. The Lyapunov functions V_1, V_2 are modified from (3) to use a quadratic form with an additional constant c satisfying constraints $0 < c < b, c < b-g, c < \frac{(a-f)(b-g)}{a-f+g^2}, c < \frac{a(b-g)}{a+g^2}$ (such a constant always exists under the assumptions on a, b, f, g). This technical modification is required to prove UGpAS for α_m directly with the Lyapunov functions. Branicky's earlier proof requires LaSalle's principle [4].

Another challenging aspect of this case study is verification of the *parametric* arithmetical conditions for V_1, V_2, i.e., stability is verified for *all* possible parameter values a, b, f, g, γ that satisfy the assumptions in P_m. Such questions are decidable in theory [3, 46], but are difficult for automated solvers in practice (even out of reach of solvers that require numerically bounded parameters [14]). KeYmaera X enables a user-aided proof of the required arithmetic conditions. For example, the Lie derivative of the Lyapunov function V_1 for $Ⓑ_1$ is given by $V_1' = -(b-c)y^2 - acx^2 + (cx+y)(fx+gy+\gamma)$, where V_1' is required to be strictly negative away from the origin for stability. The arithmetical argument is as follows: if $cx+y \leq 0$, then by constraint $fx+gy+\gamma \geq 0$, V_1' satisfies $V_1' \leq -(b-c)y^2 - acx^2$. Otherwise, $cx+y > 0$, then by constraint $fx+\gamma \leq 0$, V_1' satisfies $V_1' \leq -(b-g-c)y^2 - acx^2 + gcxy$. In either case, the RHS bound is a negative definite quadratic form by the earlier choice of parameter c and therefore, V_1' is negative away from the origin.

5.2 Automated Cruise Control

Oehlerking [28, Sect. 4.6] verifies the stability of an automatic cruise controller modeled as a hybrid automaton with 6 operating modes and 11 transitions between them: normal proportional-integral (PI) control, acceleration, service braking (2 modes), and emergency braking (2 modes). Figure 4 shows an abridged version of the corresponding KeYmaera X model (using $\alpha_{\texttt{ctrl}}$) with the PI control mode, where v is the relative velocity to be controlled to $v = 0$ and x, t are auxiliary integral and timer variables used in the controller. Briefly, this controller is designed to use the PI controller near $v = 0$ for stability, while its other control modes drive the system toward $v = 0$ by accelerating or braking.

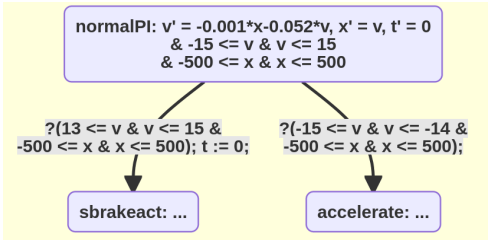
Lyapunov function candidates for this model can be successfully generated using the Stabhyli [26] stability tool for hybrid automata. However, Stabhyli (with default configurations) outputs a Lyapunov function candidate for the PI control mode that is numerically unsound, see the supplement [43]; this is a known issue with Stabhyli for control modes at the origin [26]. For this case study, the issue is manually resolved by truncating terms with very small magnitude coefficients in the generated output and then checking in KeYmaera X that the arithmetical conditions for the PI mode are satisfied *exactly* for the truncated candidate.

Further insights from the controller design are used in the UGpAS proof in KeYmaera X. Briefly, stability only concerns states and modes that are active near the origin so the stability argument and loop invariant only need to mention a single Lyapunov function for the PI control mode, while choosing δ (in Def. 1) sufficiently small

[3] See https://github.com/LS-Lab/KeYmaeraX-projects/blob/master/stability/UGpAS

[4] An important technical requirement for V_2 to be well-defined is $f \neq 0$. The case with $f = 0$ is also verified in KeYmaera X but the details are omitted here for brevity. It does not require ghost switching and uses only V_1 as its common Lyapunov function.

```
normalPI("v' = -0.001*x-0.052*v, x' = v, t' = 0
        & -15 <= v & v <= 15 & -500 <= x & x <= 500")
normalPI -->|"?(13 <= v & v <= 15 &
            -500 <= x & x <= 500); t := 0;"| sbrakeact
normalPI -->|"?(-15 <= v & v <= -14 &
            -500 <= x & x <= 500);"| accelerate
... // Other modes
```




```
\forall eps ( eps > 0 -> // Abridged stability specification
  ...
  [ ... // Initialize
  {  { ... ++ // Transitions for other modes
       ?mode = normalPI();
       { {?13 <= v & v <= 15 & -500 <= x & x <= 500; t := 0;}
         mode := sbrakeact(); ++
         ?-15 <= v & v <= -14 & -500 <= x & x <= 500;
         mode := accelerate(); ++
         mode := mode; } }
     { ...  ++ // Plant ODEs for other modes
       ?mode = normalPI();
       { v' = -0.001*x-0.052*v, x' = v, t' = 0 &
         -15 <= v & v <= 15 & -500 <= x & x <= 500 } }
  }*  // Switching loop
  ] v^2 < eps^2
```

Figure 4: Snippets of an automated cruise controller [28] modeled as a (switching) hybrid automaton. Users express the automaton within the description language (top left) and KeYmaera X visualizes the automaton on-the-fly (bottom left). The implementation automatically generates the appropriate hybrid program representation and UGpAS specification (right); `++`, `&`, `()` denote choice, conjunction, and constants in KeYmaera X's ASCII syntax respectively.

so that none of the other modes can be entered.[5] Similarly, pre-attractivity only requires reasoning about *asymptotic* convergence to the origin for the PI control mode so it suffices to show that the system leaves all other modes in finite time.

5.3 Brockett's Nonholonomic Integrator

Verification of stabilizing control laws for Brockett's nonholonomic integrator [7] is of significant interest because stability for a large class of models can be reduced to that of the integrator via coordinate transformations, e.g., Liberzon [22] transforms a unicycle model to the integrator and provides a stabilizing switching control law corresponding to parking of the unicycle. The nonholonomic integrator is described by the system of differential equations $x' = u, y' = v, z' = xv - yu$, with state variables x, y, z and state feedback control inputs $u = u(x, y, z), v = v(x, y, z)$ (to be determined below). Notably, this is a classical example of a system that is not stabilizable by purely continuous feedback control. Intuitively, no choice of controls u, v can produce motion along the z-axis ($x = y = 0$). Thus, to stabilize the system to the origin, the controller must first drive the system away from the z-axis before switching to a control law that stabilizes the system from states away from the z-axis. This intuition can be realized using two different switching strategies that are analogous to the event-triggered and time-triggered CPS design paradigms respectively [33].

5.3.1 Event-triggered Controller. Bloch and Drakunov [2] use the switching controller $u = -x + ay\,\mathrm{sign}(z), v = -y - ax\,\mathrm{sign}(z)$ to asymptotically stabilize the integrator in the region $\frac{a}{2}(x^2+y^2) \geq |z|$ for any given constant $a > 0$. This controller first drives the system towards the plane $z = 0$ and, once it reaches the plane, *slides* along the plane towards the origin. The closed-loop system is modeled as an instance of state-dependent switching $\alpha_{\texttt{state}}$ with 3 modes depending on the sign of z and specification P_e:

$$\text{Ⓐ} \equiv x' = -x + ay, y' = -y - ax, z' = -a(x^2 + y^2)\,\&\, z \geq 0$$

$$\text{Ⓑ} \equiv x' = -x - ay, y' = -y + ax, z' = a(x^2 + y^2)\,\&\, z \leq 0$$

$$\text{Ⓒ} \equiv x' = -x, y' = -y, z' = 0\,\&\, z = 0 \quad \alpha_e \equiv \Big(\text{Ⓐ} \cup \text{Ⓑ} \cup \text{Ⓒ}\Big)^*$$

$$P_e \equiv a > 0 \rightarrow \mathrm{UStab}(\alpha) \wedge$$
$$\forall \delta{>}0\, \forall \varepsilon{>}0\, \exists T{\geq}0\, \forall x, y, z \left(\|x, y, z\| < \delta \wedge \frac{a}{2}(x^2 + y^2) \geq |z| \rightarrow [t := 0; \alpha_e, t' = 1](t \geq T \rightarrow \|x, y, z\| < \varepsilon)\right)$$

The specification P_e is identical to UGpAS except it restricts pre-attractivity to the applicable region $\frac{a}{2}(x^2 + y^2) \geq |z|$ for the controller.[6] Its verification uses the squared norm $V = x^2 + y^2 + z^2$ as a common Lyapunov function. The key modification to the pre-attractivity proof, cf. Section 3.1, is to use (and verify) the fact that $\frac{a}{2}(x^2 + y^2) \geq |z|$ is a loop invariant of α_e. This additional invariant corresponds to the fact that the controller keeps the system within its applicable region (if the system is initially within that region).

In fact, α_e can be extended to a globally stabilizing controller, as modeled by $\alpha_{\hat{e}}$ below (`if`, `else` branching is supported as an abbreviation in KeYmaera X [33]):

$$\text{Ⓓ} \equiv x' = u, y' = v, z' = xv - yu\,\&\, \frac{a}{2}(x^2 + y^2) \leq |z|$$

$$\text{Ⓔ} \equiv x' = u, y' = v, z' = xv - yu\,\&\, \frac{a}{2}(x^2 + y^2) \geq |z|$$

$$\alpha_{\hat{e}} \equiv \Big(\texttt{if}\big(\frac{a}{2}(x^2 + y^2) \geq |z|\big)\ \{\text{Ⓐ} \cup \text{Ⓑ} \cup \text{Ⓒ}\}$$
$$\texttt{else}\ \Big\{\ \texttt{if}((x - y)z \leq 0)\{u := c; v := c\}$$
$$\texttt{else}\{u := -c; v := -c\};$$
$$\{\text{Ⓓ} \cup \text{Ⓔ}\}\ \Big\}\Big)^*$$

[5] In fact, the PI controller equations are exactly those of a linearized pendulum, which has known Lyapunov functions [21, 44]. It could be interesting to modify Stabhyli to accept user-provided Lyapunov function hints for certain modes.

[6] The applicable region is equivalently characterized by the real arithmetic formula $(z{\geq}0 \rightarrow \frac{a}{2}(x^2 + y^2){\geq}z) \wedge (z{\leq}0 \rightarrow \frac{a}{2}(x^2 + y^2){\geq}{-z})$, omitted for brevity.

If the system is in the applicable region (outer **if** branch), then the previous controller from α_e is used. Otherwise, outside the applicable region (outer **else** branch), the system applies a constant control $c > 0$ chosen to drive the system into the applicable region. The pair of ODEs Ⓓ and Ⓔ model an event-trigger in dL [33], where the switching controller is triggered to make its next decision when the system reaches the switching surface $\frac{a}{2}(x^2+y^2) = |z|$.

The specification $P_{\hat{e}} \equiv a{>}0 \wedge c{>}0 \rightarrow \mathrm{UGpAS}(\alpha_{\hat{e}})$ is proved by modifying the loop invariants to account for an initial period where the system is outside the applicable region. For example, the stability loop invariant $Inv_s \equiv (\neg \frac{a}{2}(x^2+y^2) \geq |z| \rightarrow |z| < \delta) \wedge (\frac{a}{2}(x^2+y^2) \geq |z| \rightarrow \|x,y,z\| < \varepsilon)$ expresses that the controller keeps $|z|$ sufficiently small with $|z| < \delta$ to preserve stability outside the applicable region. The pre-attractivity loop invariant is similarly split between the two cases, with an explicit time estimate on the time it takes for the system to enter the applicable region.

5.3.2 Time-triggered Controller. The time-triggered switching strategy [33], modeled by α_τ below, is similar to that proposed by Liberzon [22, Section 4.2]. If the system is on the z-axis and away from the origin Ⓐ, the controller sets an internal stopwatch τ and drives the system away from the axis for maximum duration $T_0 > 0$ with $u = z, v = z$. Otherwise Ⓑ, the controller drives the system towards the origin along a parabolic curve of the form $\frac{a}{2}(x^2+y^2) = z$.

$$\alpha_\tau \equiv \Big(\textbf{if}(x = 0 \wedge y = 0 \wedge z \neq 0)\ \{$$
$$\text{Ⓐ} \qquad \tau := 0; x' = z, y' = z, z' = xz - yz \,\&\, \tau \leq T_0 \quad \}$$
$$\textbf{else}\ \{\ a := \frac{2z}{x^2+y^2};$$
$$\text{Ⓑ} \qquad x' = -x + ay, y' = -y - ax, z' = -a(x^2+y^2) \quad \}\Big)^*$$

The specification $P_\tau \equiv T_0 > 0 \rightarrow \mathrm{UGpAS}(\alpha_\tau)$ is again proved by analyzing both cases of the controller in the loop invariants, e.g., with the pre-attractivity invariant Inv_a:

$$\big(x = 0 \wedge y = 0 \wedge z \neq 0 \rightarrow |z| < \delta \wedge t = 0\big) \wedge$$
$$\big(\neg(x = 0 \wedge y = 0 \wedge z \neq 0) \rightarrow$$
$$\|x,y,z\| > \varepsilon \rightarrow \|x,y,z\|^2 < \delta^2(2T_0^2+1) - \varepsilon^2(t - T_0)\big)$$

The top conjunct says the system may start transiently on the z-axis (away from $z = 0$) at time $t = 0$. The bottom conjunct gives explicit bounds on $\|x,y,z\|$, which, for sufficiently large $t \geq T$, implies that the system enters $\|x,y,z\| < \varepsilon$ as required for pre-attractivity. The transient term $\delta^2(2T_0^2+1)$ upper bounds the (squared) norm of the system state after starting on the z-axis in ball $\|x,y,z\| < \delta$ and following mode Ⓐ for the maximum stopwatch duration $\tau = T_0$.

6 RELATED WORK

Switched Systems. Comprehensive introductions to the analysis and design of switching control can be found in the literature [10, 22, 42]. An important design consideration (which this paper sidesteps, cf. Remark 1) is whether a given switched or hybrid system has complete solutions [16, 17, 23, 48]. Justification of such design considerations, and other stability notions of interest for switching designs, e.g., quadratic, region, or set-based stability [16, 17, 22, 35, 42], can be done in dL with appropriate formal specifications of the desired properties from the literature [31, 33, 44, 45]. Another complementary question is how to design a switching control law that *stabilizes* a given system. Switching design approaches are often guided by underlying stability arguments [22, 37, 42]; the loop invariants from Section 3 are expected to help guide correct-by-construction synthesis of such controllers.

Stability Analysis and Verification. Corollaries 3–6 formalize various Lyapunov function-based stability arguments from the literature [5, 47] using loop invariants, yielding trustworthy, computer-checked stability proofs in KeYmaera X [11, 12]. Other computer-aided approaches for switched system stability analysis are based on finding Lyapunov functions that satisfy the requisite arithmetical conditions [20, 26, 28, 36, 39, 40]. Although the search for such functions can often be done efficiently with numerical techniques [26, 30, 36], various authors have emphasized the need to check that their outputs satisfy the arithmetical conditions *exactly*, i.e., without numerical errors compromising the resulting stability claims [1, 20, 38] (see, e.g., Section 5.2). This paper's deductive approach goes further as it comprehensively verifies *all* steps of the stability argument down to its underlying discrete and continuous reasoning steps [32, 33]. The generality of this approach is precisely what enables verification of various classes of switching mechanisms all within a common logical framework (Section 3) and verification of non-standard stability arguments (Section 5). Alternative approaches to stability verification are based on abstraction [15, 41] and model checking [35].

7 CONCLUSION

This paper shows how to deductively verify switched system stability, using dL's nested quantification over hybrid programs to specify stability, and dL's axiomatics to prove those specifications. Loop invariants—a classical technique from verification—are used to succinctly capture the desired properties of a given switching design; through deductive proofs, these invariants yield systematic, correct-by-construction derivation of the requisite arithmetical conditions on Lyapunov functions for stability arguments in implementations. An interesting direction for future work is to use other Lyapunov function generation techniques [20, 26, 28, 40], which—thanks to the presented approach—do not have to be trusted since their results can be checked independently by KeYmaera X. This would enable fully automated, yet sound and trustworthy verification of switched system stability based on dL's parsimonious hybrid program reasoning principles.

ACKNOWLEDGMENTS

We thank Thomas Baar and the anonymous reviewers for their helpful feedback on this paper. This material is based upon work supported by the National Science Foundation under Grant No. CNS-1739629. This research was sponsored by the AFOSR under grant number FA9550-16-1-0288.

REFERENCES

[1] Daniele Ahmed, Andrea Peruffo, and Alessandro Abate. 2020. Automated and Sound Synthesis of Lyapunov Functions with SMT Solvers. In *TACAS (LNCS, Vol. 12078)*, Armin Biere and David Parker (Eds.). Springer, 97–114. https://doi.org/10.1007/978-3-030-45190-5_6

[2] Anthony Bloch and Sergey Drakunov. 1996. Stabilization and tracking in the nonholonomic integrator via sliding modes. *Systems & Control Letters* 29, 2 (1996), 91–99. https://doi.org/10.1016/S0167-6911(96)00049-7

[3] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. 1998. *Real Algebraic Geometry*. Springer, Heidelberg. https://doi.org/10.1007/978-3-662-03718-8

[4] Michael S. Branicky. 1994. Analyzing continuous switching systems: theory and examples. In *ACC*, Vol. 3. 3110–3114. https://doi.org/10.1109/ACC.1994.735143

[5] Michael S. Branicky. 1998. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Autom. Control.* 43, 4 (1998), 475–482. https://doi.org/10.1109/9.664150

[6] Michael S. Branicky. 2005. Introduction to Hybrid Systems. In *Handbook of Networked and Embedded Control Systems*, Dimitrios Hristu-Varsakelis and William S. Levine (Eds.). Birkhäuser, 91–116. https://doi.org/10.1007/0-8176-4404-0_5

[7] R. W. Brockett. 1983. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*. Birkhauser, 181–191.

[8] Carmen Chicone. 2006. *Ordinary Differential Equations with Applications, Second Edition*. Springer-Verlag New York. https://doi.org/10.1007/0-387-35794-7

[9] Jorge Cortes. 2008. Discontinuous dynamical systems. *IEEE Control Systems Magazine* 28, 3 (2008), 36–73. https://doi.org/10.1109/MCS.2008.919306

[10] Raymond A. Decarlo, Michael S. Branicky, Stefan Pettersson, and Bengt Lennartson. 2000. Perspectives and results on the stability and stabilizability of hybrid systems. *Proc. IEEE* 88, 7 (2000), 1069–1082. https://doi.org/10.1109/5.871309

[11] Nathan Fulton, Stefan Mitsch, Brandon Bohrer, and André Platzer. 2017. Bellerophon: Tactical Theorem Proving for Hybrid Systems. In *ITP (LNCS, Vol. 10499)*, Mauricio Ayala-Rincón and César A. Muñoz (Eds.). Springer, 207–224. https://doi.org/10.1007/978-3-319-66107-0_14

[12] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. 2015. KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems. In *CADE (LNCS, Vol. 9195)*, Amy P. Felty and Aart Middeldorp (Eds.). Springer, Cham, 527–538. https://doi.org/10.1007/978-3-319-21401-6_36

[13] Sicun Gao, James Kapinski, Jyotirmoy V. Deshmukh, Nima Roohi, Armando Solar-Lezama, Nikos Aréchiga, and Soonho Kong. 2019. Numerically-Robust Inductive Proof Rules for Continuous Dynamical Systems. In *CAV (LNCS, Vol. 11562)*, Isil Dillig and Serdar Tasiran (Eds.). Springer, 137–154. https://doi.org/10.1007/978-3-030-25543-5_9

[14] Sicun Gao, Soonho Kong, and Edmund M. Clarke. 2013. dReal: An SMT Solver for Nonlinear Theories over the Reals. In *CADE (LNCS, Vol. 7898)*, Maria Paola Bonacina (Ed.). Springer, 208–214. https://doi.org/10.1007/978-3-642-38574-2_14

[15] Miriam García Soto and Pavithra Prabhakar. 2020. Abstraction based verification of stability of polyhedral switched systems. *Nonlinear Analysis: Hybrid Systems* 36 (2020), 100856. https://doi.org/10.1016/j.nahs.2020.100856

[16] Rafal Goebel, Ricardo G. Sanfelice, and Andrew R. Teel. 2009. Hybrid dynamical systems. *IEEE Control Systems Magazine* 29, 2 (2009), 28–93. https://doi.org/10.1109/MCS.2008.931718

[17] Rafal Goebel, Ricardo G. Sanfelice, and Andrew R. Teel. 2012. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

[18] Thomas A. Henzinger. 1996. The Theory of Hybrid Automata. In *LICS*. IEEE Computer Society, 278–292.

[19] Martin Johansson and Anders Rantzer. 1998. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Autom. Control.* 43, 4 (1998), 555–559. https://doi.org/10.1109/9.664157

[20] James Kapinski, Jyotirmoy V. Deshmukh, Sriram Sankaranarayanan, and Nikos Aréchiga. 2014. Simulation-guided Lyapunov analysis for hybrid dynamical systems. In *HSCC*, Martin Fränzle and John Lygeros (Eds.). ACM, 133–142. https://doi.org/10.1145/2562059.2562139

[21] Hassan K. Khalil. 1992. *Nonlinear systems*. Macmillan Publishing Company, New York.

[22] Daniel Liberzon. 2003. *Switching in Systems and Control*. Birkhäuser. https://doi.org/10.1007/978-1-4612-0017-8

[23] John Lygeros, Karl Henrik Johansson, Slobodan N. Simic, Jun Zhang, and Shankar S. Sastry. 2003. Dynamical properties of hybrid automata. *IEEE Trans. Autom. Control.* 48, 1 (2003), 2–17. https://doi.org/10.1109/TAC.2002.806650

[24] Stefan Mitsch and André Platzer. 2016. The KeYmaera X proof IDE: Concepts on usability in hybrid systems theorem proving. In *3rd Workshop on Formal Integrated Development Environment (EPTCS, Vol. 240)*, Catherine Dubois, Paolo Masci, and Dominique Méry (Eds.). 67–81. https://doi.org/10.4204/EPTCS.240.5

[25] Stefan Mitsch and André Platzer. 2020. A Retrospective on Developing Hybrid Systems Provers in the KeYmaera Family - A Tale of Three Provers. In *Deductive Software Verification: Future Perspectives - Reflections on the Occasion of 20 Years of KeY*, Wolfgang Ahrendt, Bernhard Beckert, Richard Bubel, Reiner Hähnle, and Matthias Ulbrich (Eds.). LNCS, Vol. 12345. Springer, 21–64. https://doi.org/10.1007/978-3-030-64354-6_2

[26] Eike Möhlmann and Oliver E. Theel. 2013. Stabhyli: a tool for automatic stability verification of non-linear hybrid systems. In *HSCC*, Calin Belta and Franjo Ivancic (Eds.). ACM, 107–112. https://doi.org/10.1145/2461328.2461347

[27] A. S. Morse. 1995. Control Using Logic-Based Switching. In *Trends in Control*, Alberto Isidori (Ed.). Springer London, London, 69–113. https://doi.org/10.1007/978-1-4471-3061-1_4

[28] Jens Oehlerking. 2011. *Decomposition of stability proofs for hybrid systems*. Ph. D. Dissertation. Carl von Ossietzky University of Oldenburg. https://oops.uni-oldenburg.de/id/eprint/1375

[29] Susan S. Owicki and David Gries. 1976. Verifying Properties of Parallel Programs: An Axiomatic Approach. *Commun. ACM* 19, 5 (1976), 279–285. https://doi.org/10.1145/360051.360224

[30] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P. A. Parrilo, M. M. Peet, and D. Jagt. 2021. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. `http://arxiv.org/abs/1310.4716`. Available from `https://github.com/oxfordcontrol/SOSTOOLS`.

[31] André Platzer. 2010. *Logical Analysis of Hybrid Systems - Proving Theorems for Complex Dynamics*. Springer. https://doi.org/10.1007/978-3-642-14509-4

[32] André Platzer. 2017. A Complete Uniform Substitution Calculus for Differential Dynamic Logic. *J. Autom. Reasoning* 59, 2 (2017), 219–265. https://doi.org/10.1007/s10817-016-9385-1

[33] André Platzer. 2018. *Logical Foundations of Cyber-Physical Systems*. Springer, Cham. https://doi.org/10.1007/978-3-319-63588-0

[34] André Platzer and Yong Kiam Tan. 2020. Differential Equation Invariance Axiomatization. *J. ACM* 67, 1, Article 6 (2020), 66 pages. https://doi.org/10.1145/3380825

[35] Andreas Podelski and Silke Wagner. 2006. Model Checking of Hybrid Systems: From Reachability Towards Stability. In *HSCC (LNCS, Vol. 3927)*, João P. Hespanha and Ashish Tiwari (Eds.). Springer, 507–521. https://doi.org/10.1007/11730637_38

[36] S. Prajna and A. Papachristodoulou. 2003. Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In *ACC*, Vol. 4. 2779–2784 vol.4. https://doi.org/10.1109/ACC.2003.1243743

[37] Hadi Ravanbakhsh and Sriram Sankaranarayanan. 2015. Counter-Example Guided Synthesis of control Lyapunov functions for switched systems. In *CDC*. IEEE, 4232–4239. https://doi.org/10.1109/CDC.2015.7402879

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

[39] Sriram Sankaranarayanan, Xin Chen, and Erika Ábrahám. 2013. Lyapunov Function Synthesis Using Handelman Representations. In *NOLCOS*, Sophie Tarbouriech and Miroslav Krstic (Eds.). International Federation of Automatic Control, 576–581. https://doi.org/10.3182/20130904-3-FR-2041.00198

[40] Zhikun She and Bai Xue. 2014. Discovering Multiple Lyapunov Functions for Switched Hybrid Systems. *SIAM J. Control. Optim.* 52, 5 (2014), 3312–3340. https://doi.org/10.1137/130934313

[41] Miriam García Soto and Pavithra Prabhakar. 2018. Averist: Algorithmic Verifier for Stability of Linear Hybrid Systems. In *HSCC*, Maria Prandini and Jyotirmoy V. Deshmukh (Eds.). ACM, 259–264. https://doi.org/10.1145/3178126.3178154

[42] Zhendong Sun and Shuzhi Sam Ge. 2011. *Stability Theory of Switched Dynamical Systems*. Springer. https://doi.org/10.1007/978-0-85729-256-8

[43] Yong Kiam Tan, Stefan Mitsch, and André Platzer. 2021. Verifying Switched System Stability With Logic. *CoRR* abs/2111.01928 (2021). arXiv:2111.01928 https://arxiv.org/abs/2111.01928

[44] Yong Kiam Tan and André Platzer. 2021. Deductive Stability Proofs for Ordinary Differential Equations. In *TACAS (LNCS, Vol. 12652)*, Jan Friso Groote and Kim Guldstrand Larsen (Eds.). Springer, 181–199. https://doi.org/10.1007/978-3-030-72013-1_10

[45] Yong Kiam Tan and André Platzer. 2021. Switched Systems as Hybrid Programs. In *ADHS (IFAC-PapersOnLine, Vol. 54)*, Raphaël M. Jungers, Necmiye Ozay, and Alessandro Abate (Eds.). Elsevier, 247–252. https://doi.org/10.1016/j.ifacol.2021.08.506

[46] Alfred Tarski. 1951. *A Decision Method for Elementary Algebra and Geometry*. RAND Corporation, Santa Monica, CA.

[47] Guisheng Zhai, Bo Hu, Kazunori Yasuda, and Anthony N. Michel. 2001. Stability analysis of switched systems with stable and unstable subsystems: An average dwell time approach. *Int. J. Syst. Sci.* 32, 8 (2001), 1055–1061. https://doi.org/10.1080/00207720116692

[48] Jun Zhang, Karl Henrik Johansson, John Lygeros, and Shankar Sastry. 2001. Zeno hybrid systems. *Int. J. Robust Nonlinear Control.* 11, 5 (2001), 435–451. https://doi.org/10.1002/rnc.592