Verifying Switched System Stability With Logic

Yong Kiam Tan Stefan Mitsch André Platzer

Computer Science Department, Carnegie Mellon University

HSCC, Online, 5 May 2022

Outline

Outline

Hybrid Systems & Stability

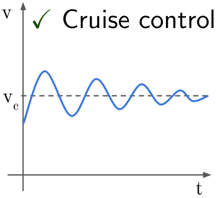
Many real world systems feature **hybrid** (discrete + continuous) dynamics:

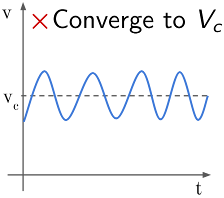
Hybrid Systems & Stability

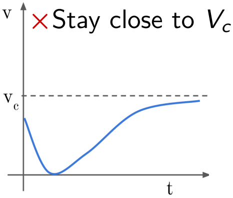
Many real world systems feature **hybrid** (discrete + continuous) dynamics:



Various controllers driving a car near cruising velocity V_c:



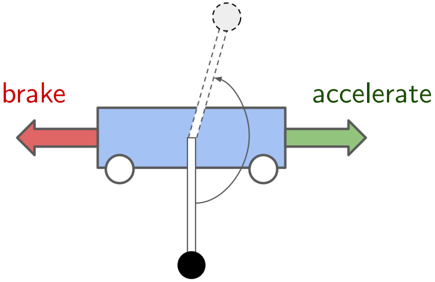




Stability is a key correctness criterion for control systems deserving proofs.
Prior work: Stability verification for ordinary diff. eqs. [TACAS'21].

Switched Systems & Stability

Fact: Hybrid **switching** control can be used to achieve control objectives that cannot otherwise be achieved by purely continuous means.

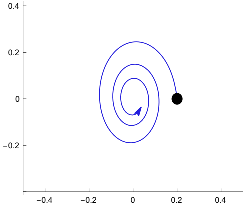


Example: Discontinuity in controller, e.g., with switching, is needed to invert pendulum globally, from all initial states.
Others: Adaptive control, bang-bang control, gain scheduling, . . .

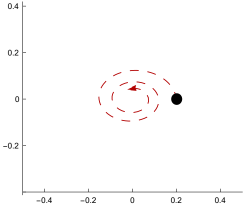
Switched Systems & Stability

Fact: Hybrid **switching** control can be used to achieve control objectives that cannot otherwise be achieved by purely continuous means.

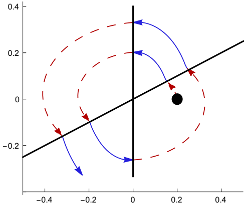
Fact: Discrete switching between stable continuous ODEs can be unstable.



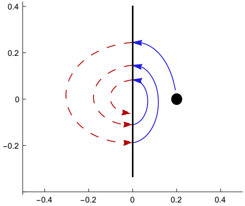
✓Stable ODE



✓Stable ODE



×Unstable switching

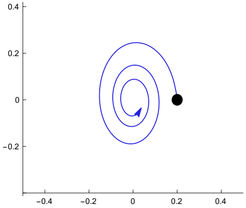


✓Stable switching

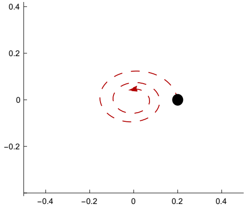
Switched Systems & Stability

Fact: Hybrid **switching** control can be used to achieve control objectives that cannot otherwise be achieved by purely continuous means.

Fact: Discrete switching between stable continuous ODEs can be unstable.



✓Stable ODE



✓Stable ODE

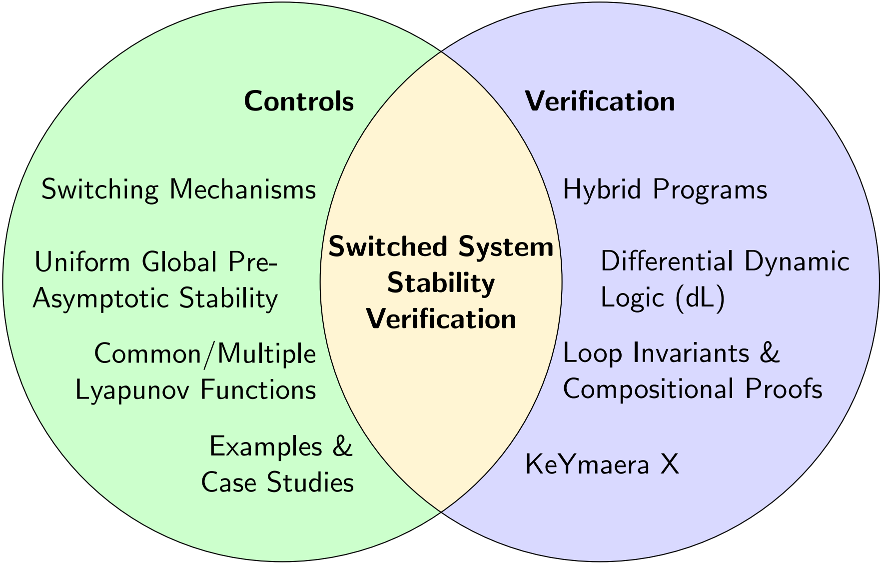
Challenge: Need adequate stability justification for switching designs, e.g., state-dependent [1, 5], time-dependent [8], automata-based [3, 4], . . .

This work: Trustworthy, uniform stability verification framework for switching designs by combining ideas from controls & verification.

×Unstable switching

✓Stable switching

Verifying Switched System Stability With Logic



Outline

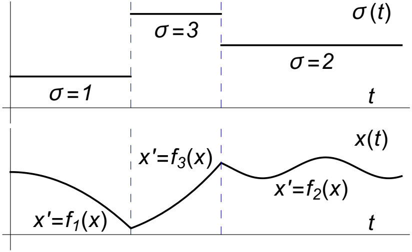
Switched Systems

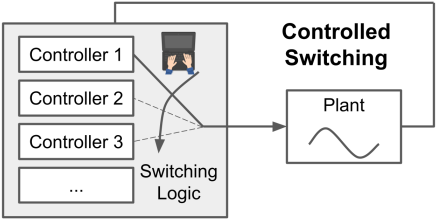
Switched systems consist of a family of continuous ODEs and a discrete switching signal choosing between those ODEs.

Switched system:

$$x' = f_{\sigma(t)}(x)$$

- $x' = f_p(x), p \in \mathcal{P}$, finite family of autonomous ODEs
- $\sigma(t)$, switching signal chooses ODE to follow at time t





Hybrid Programs

Differential dynamic logic (dL) uses the **hybrid programs** language to model hybrid systems.

Hybrid programs:

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

ODE	Discrete Assign.	Test	Seq. Compose	Nondet. Choice	Nondet. Loop

Hybrid Programs

Differential dynamic logic (dL) uses the **hybrid programs** language to model hybrid systems.

Hybrid programs:

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

ODE | Discrete Assign. | Test | Seq. Compose | Nondet. Choice | Nondet. Loop

Properties of hybrid program α are specified in dL's formula language.

Specifications:

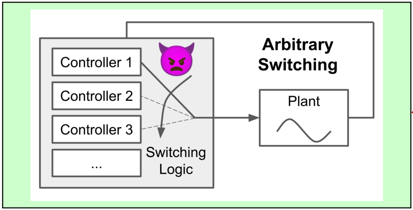
$$\phi, \psi ::= e \sim \tilde{e} \mid \phi \wedge \psi \mid \cdots \mid \forall x\, \phi \mid \exists x\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

Compare $\geq, >, =$ | And, Or, etc. | For all, Exists | ϕ true after all α runs | ϕ true after some α run

Red boxes are key for switched system stability specifications.

Switched Systems as Hybrid Programs [ADHS'21]

Repeat switching in a loop



$\Longleftrightarrow$

Hybrid program:

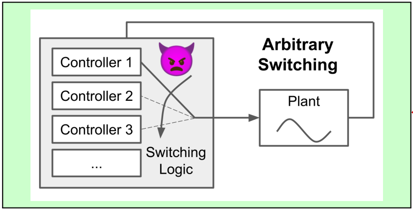
$$\alpha_{\mathtt{arb}} \equiv \Big(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \Big)^*$$

$$\bigcup_{p \in \mathcal{P}} \alpha_p \equiv \alpha_1 \cup \alpha_2 \cup \ldots \cup \alpha_m$$

Each iteration nondet. picks an ODE to run

Switched Systems as Hybrid Programs [ADHS'21]

Repeat switching in a loop

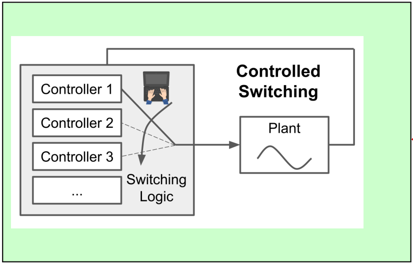


Hybrid program:

$$\alpha_{\mathtt{arb}} \equiv \left(\bigcup_{p\in\mathcal{P}} x' = f_p(x)\right)^*$$

$$\bigcup_{p\in\mathcal{P}} \alpha_p \equiv \alpha_1 \cup \alpha_2 \cup \ldots \cup \alpha_m$$

Each iteration nondet. picks an ODE to run



Hybrid program (simplified):

$$\alpha_{\mathtt{ctrl}} \equiv \left(u := ctrl(x); x' = f_u(x)\right)^*$$

Switching uses specification and reasoning for loops in dL.

Uniform Global Pre-Asymptotic Stability [Goebel et al.]

Switched system model α & specification

Switched system is **UGpAS** iff:

- **Unif. Stable**: for all $\varepsilon > 0$, there exists $\delta > 0$, all switching solutions φ from $\|\varphi(0)\| < \delta$ satisfy $\|\varphi(t)\| < \varepsilon$ for all times.
- **Unif. Pre-Attractive**: for all $\varepsilon > 0, \delta > 0$, there exists $T \geq 0$, all switching solutions φ from $\|\varphi(0)\| < \delta$ satisfy $\|\varphi(t)\| < \varepsilon$ for all times $T \leq t$.

dL UGpAS specification:

- **Unif. Stable**:

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

- **Unif. Pre-Attractive**:

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

Uniform Global Pre-Asymptotic Stability [Goebel et al.]

Switched system model α & specification

Switched system is **UGpAS** iff:

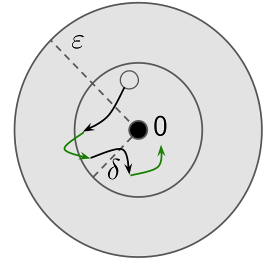
- **Unif. Stable**: for all $\varepsilon > 0$, there exists $\delta > 0$, all switching solutions $\boldsymbol{\varphi}$ from $\|\boldsymbol{\varphi}(0)\| < \delta$ satisfy $\|\boldsymbol{\varphi}(t)\| < \varepsilon$ for all times.

dL UGpAS specification:

- **Unif. Stable**:

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x \left(\|x\| < \delta \rightarrow [\alpha]\, \|x\| < \varepsilon\right)$$

This talk: Focuses on deductive dL proofs of (uniform) stability for switched systems, i.e., "if system starts close to origin, it stays close".



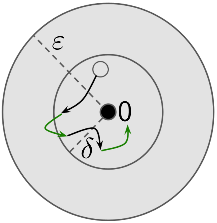
Outline

Stability Under Arbitrary Switching

Hybrid program & Stability:

$$\alpha_{\texttt{arb}} \equiv \Big(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \Big)^*$$

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\texttt{arb}}]\, \|x\| < \varepsilon \big)$$

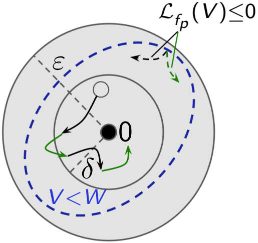


Stability Under Arbitrary Switching

Hybrid program & Stability:

$$\alpha_{\texttt{arb}} \equiv \Big(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \Big)^*$$

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\texttt{arb}}]\, \|x\| < \varepsilon \big)$$



Arithmetic conditions on common **Lyapunov function** V for all modes:

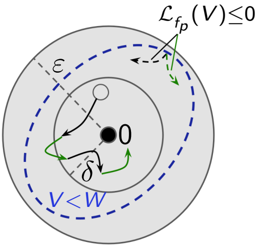
- $V(0) = 0$ and $V(x) > 0$ for all $\|x\| > 0$;
- for each ODE $x' = f_p(x), p \in \mathcal{P}$, the Lie derivative $\mathcal{L}_{f_p}(V)$ satisfies $\mathcal{L}_{f_p}(V) \le 0$.

Stability Under Arbitrary Switching

Hybrid program & Stability:

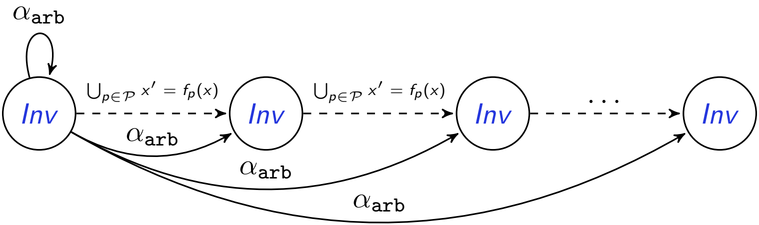
$$\alpha_{\texttt{arb}} \equiv \Big(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \Big)^*$$

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\texttt{arb}}]\, \|x\| < \varepsilon \big)$$



$Inv \equiv \|x\| < \varepsilon \wedge V < W$

Loop invariant *Inv* is preserved across all loop iterations for $\alpha_{\texttt{arb}}$:

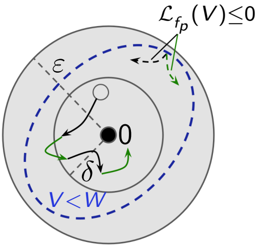


Stability Under Arbitrary Switching

Hybrid program & Stability:

$$\alpha_{\mathtt{arb}} \equiv \Big(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \Big)^*$$

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\mathtt{arb}}]\, \|x\| < \varepsilon \big)$$



$$Inv \equiv \|x\| < \varepsilon \wedge V < W$$

Formal proof syntactically deduces sound arithmetic conditions on V:

Deduction

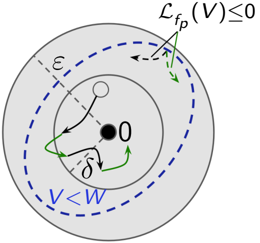
$$\dfrac{\dfrac{\vdots}{\Gamma \vdash Inv} \quad \dfrac{\dfrac{\vdash V(0)=0 \quad \|x\|{>}0 \vdash V(x) > 0 \quad \vdash \mathcal{L}_{f_p}(V)(x) \le 0}{\cdots \quad \text{(hybrid program reasoning)}}}{Inv \vdash [\bigcup_{p \in \mathcal{P}} x' = f_p(x)]\, Inv} \quad \dfrac{\vdots}{Inv \vdash \|x\| < \varepsilon}}{\dfrac{\Gamma \vdash [\alpha_{\mathtt{arb}}]\, \|x\| < \varepsilon}{\dfrac{\cdots \quad \text{(logic/arithmetic reasoning for } \Gamma)}{\vdash \forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\mathtt{arb}}]\, \|x\| < \varepsilon \big)}}}\,\text{loop}$$

Stability Under Arbitrary Switching

Hybrid program & Stability:

$$\alpha_{\texttt{arb}} \equiv \Big(\bigcup_{p \in \mathcal{P}} x' = f_p(x) \Big)^*$$

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\texttt{arb}}]\, \|x\| < \varepsilon\big)$$



$$Inv \equiv \|x\| < \varepsilon \wedge V < W$$

Formal proof syntactically deduces sound arithmetic conditions on V:

Summarized as a derived dL proof rule and implemented in KeYmaera X:

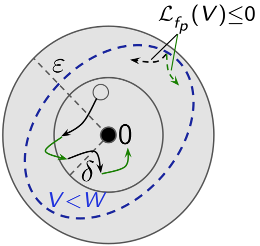
$$\text{CLF}\ \frac{\vdash V(0) = 0 \quad \|x\|{>}0 \vdash V(x) > 0 \quad \vdash \mathcal{L}_{f_p}(V)(x) \le 0}{\vdash \forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\|x\| < \delta \rightarrow [\alpha_{\texttt{arb}}]\, \|x\| < \varepsilon\big)}$$

Stability Under Controlled Switching

Hybrid program & Stability:

$$\alpha_{\mathtt{ctrl}} \equiv \left(u := ctrl(x); x' = f_u(x)\right)^*$$

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x \left(\|x\| < \delta \rightarrow [\alpha_{\mathtt{ctrl}}]\, \|x\| < \varepsilon\right)$$



$$Inv \equiv \|x\| < \varepsilon \wedge \bigvee_{p \in \mathcal{P}} \left(u = p \wedge V_p < W\right)$$

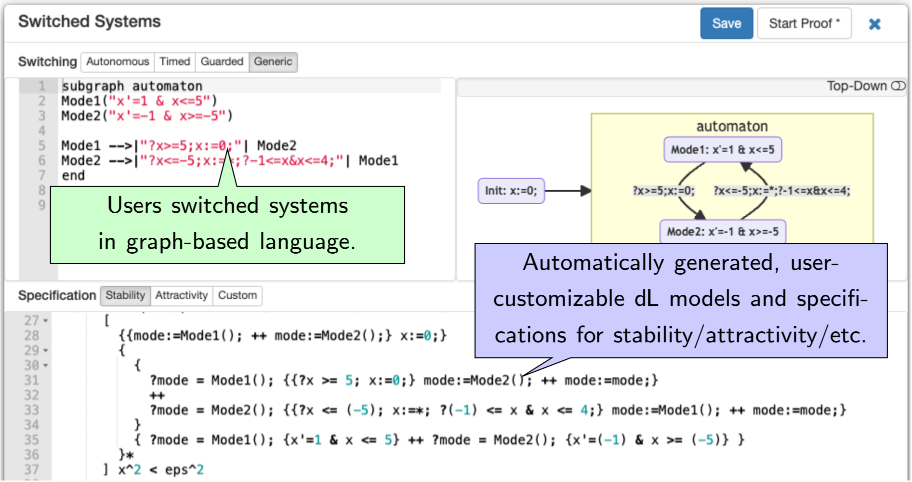
Compositional proof yields correct-by-construction conditions on V_p:

Deduction

$$\dfrac{\dfrac{\vdots}{\Gamma \vdash Inv} \quad \dfrac{\dfrac{\dfrac{\cdots}{Inv \vdash [u := ctrl(x)]\, Inv} \quad \dfrac{\cdots}{Inv \vdash [x' = f_u(x)]\, Inv}}{\cdots \quad \text{(hybrid program reasoning)}}}{Inv \vdash [u := ctrl(x); x' = f_u(x)]\, Inv} \quad \dfrac{\vdots}{Inv \vdash \|x\| < \varepsilon}}{\dfrac{\Gamma \vdash [\alpha_{\mathtt{ctrl}}]\, \|x\| < \varepsilon}{\dfrac{\cdots \quad \text{(logic/arithmetic reasoning for } \Gamma)}{\vdash \forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x \left(\|x\| < \delta \rightarrow [\alpha_{\mathtt{ctrl}}]\, \|x\| < \varepsilon\right)}}}\ \text{loop}$$

Outline

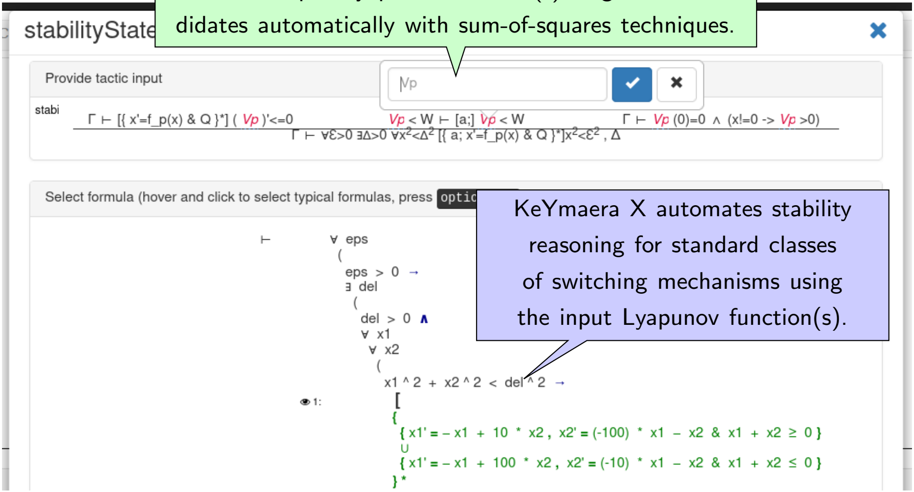
KeYmaera X Modeling & Proof Interface



The implementation adds switched system support to KeYmaera X's IDE and fully automates arguments for **standard** switching designs.

KeYmaera X Modeling & Proof Interface

Users can input Lyapunov function(s) or generate candidates automatically with sum-of-squares techniques.



KeYmaera X automates stability reasoning for standard classes of switching mechanisms using the input Lyapunov function(s).

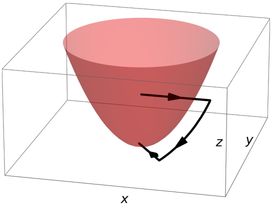
The implementation adds switched system support to KeYmaera X's IDE and fully automates arguments for **standard** switching designs.

Case Studies (see paper)

Semi-automated verification of **non-standard** switching design/arguments:

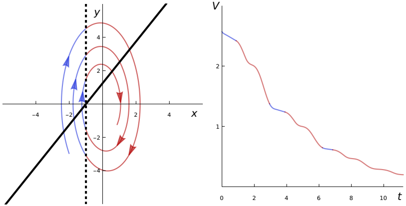
Nonholonomic Integrator:

$$x' = u, y' = v, z' = xv - yu$$

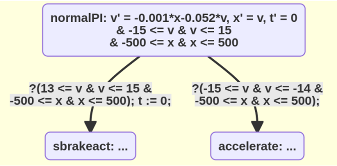


Canonical Max System:

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



Automatic cruise controller:



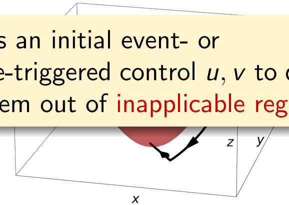

```
\forall eps ( eps > 0 -> ... // Abridged stability specification
  [ ... // Initialize
  {  { ... ++ // Transitions for other modes
       ?mode = normalPI();
       { {?13 <= v & v <= 15 & -500 <= x & x <= 500; t := 0;}
         mode := sbrakeact(); ++ ... } }
     { ...  ++ // Plant ODEs for other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

Case Studies (see paper)

Semi-automated verification of **non-standard** switching design/arguments:

Nonholonomic Integrator:

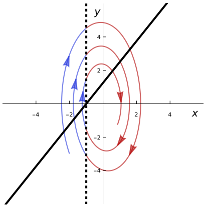
$$x' = u, y' = v, z' = xv - yu$$

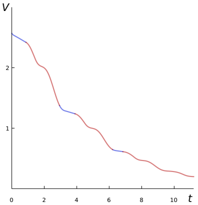


Uses an initial event- or time-triggered control u, v to drive system out of inapplicable region.

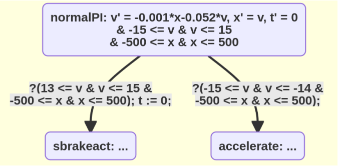
Canonical Max System:

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





Automatic cruise controller:

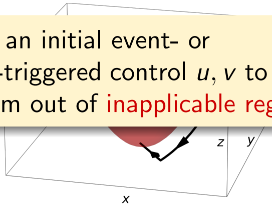



```
\forall eps ( eps > 0 -> ... // Abridged stability specification
  [ ... // Initialize
  {  { ... ++ // Transitions for other modes
       ?mode = normalPI();
       { {?13 <= v & v <= 15 & -500 <= x & x <= 500; t := 0;}
         mode := sbrakeact(); ++ ... } }
     { ...  ++ // Plant ODEs for other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

Semi-automated verification of **non-standard** switching design/arguments:

Nonholonomic Integrator:

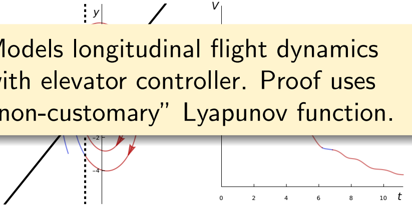
$$x' = u, y' = v, z' = xv - yu$$



Uses an initial event- or time-triggered control u, v to drive system out of inapplicable region.

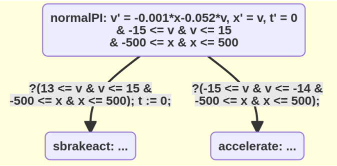
Canonical Max System:

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



Models longitudinal flight dynamics with elevator controller. Proof uses "non-customary" Lyapunov function.

Automatic cruise controller:



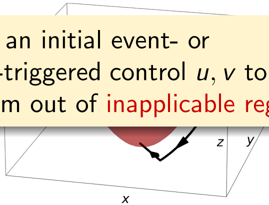

```
\forall eps ( eps > 0 -> ... // Abridged stability specification
  [ ... // Initialize
  {  { ... ++ // Transitions for other modes
       ?mode = normalPI();
       { {?13 <= v & v <= 15 & -500 <= x & x <= 500; t := 0;}
         mode := sbrakeact(); ++ ... } }
     { ...  ++ // Plant ODEs for other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

Case Studies (see paper)

Semi-automated verification of **non-standard** switching design/arguments:

Nonholonomic Integrator:

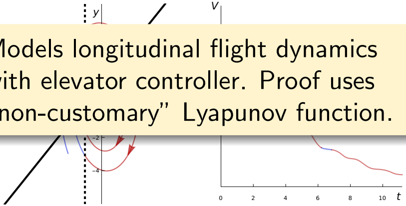
$$x' = u, y' = v, z' = xv - yu$$



Uses an initial event- or time-triggered control u, v to drive system out of inapplicable region.

Canonical Max System:

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

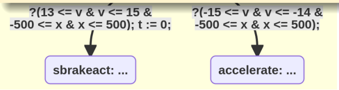


Models longitudinal flight dynamics with elevator controller. Proof uses "non-customary" Lyapunov function.

Automatic cruise controller:

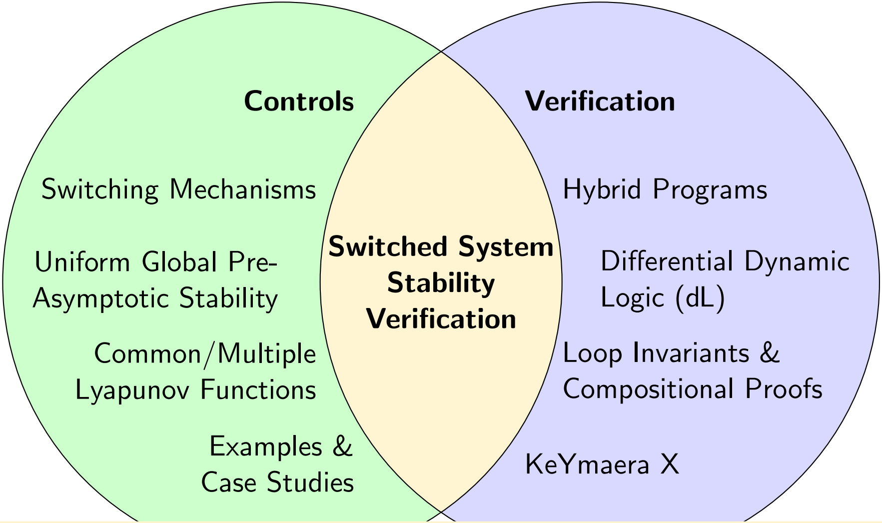
```
\forall eps ( eps > 0 -> ... // Abridged stability specification
  [ ... // Initialize
      ?mode = normalPI();
      { v' = -0.001*x-0.052*v, x' = v, t' = 0 & ... } }
  }*  // Switching loop
  ] v^2 < eps^2
```

Hybrid automaton with 6 modes and 11 transitions: PI control, acceleration, service braking (2 modes), and emergency braking (2 modes).



Outline

Verifying Switched System Stability With Logic



This work: Automated support for modeling and trustworthy stability verification of various switching designs using dL and KeYmaera X.

References I

[1] Branicky, M. S. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Trans. Autom. Control., 43(4):475–482.

[2] Goebel, R., Sanfelice, R. G., and Teel, A. R. (2012). Hybrid Dynamical Systems: Modeling, Stability, and Robustness. Princeton University Press.

[3] Möhlmann, E. and Theel, O. E. (2013). Stabhyli: a tool for automatic stability verification of non-linear hybrid systems. In Belta, C. and Ivancic, F., editors, HSCC, pages 107–112. ACM.

[4] Podelski, A. and Wagner, S. (2006). Model checking of hybrid systems: From reachability towards stability. In Hespanha, J. P. and Tiwari, A., editors, HSCC, volume 3927 of LNCS, pages 507–521. Springer.

[5] Prajna, S. and Papachristodoulou, A. (2003). Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In ACC, volume 4, pages 2779–2784 vol.4.

References II

[6] Tan, Y. K. and Platzer, A. (2021a). Deductive stability proofs for ordinary differential equations. In Groote, J. F. and Larsen, K. G., editors, TACAS, volume 12652 of LNCS, pages 181–199. Springer.

[7] Tan, Y. K. and Platzer, A. (2021b). Switched systems as hybrid programs. In Jungers, R. M., Ozay, N., and Abate, A., editors, ADHS, volume 54 of IFAC-PapersOnLine, pages 247–252. Elsevier.

[8] Zhai, G., Hu, B., Yasuda, K., and Michel, A. N. (2001). Stability analysis of switched systems with stable and unstable subsystems: An average dwell time approach. Int. J. Syst. Sci., 32(8):1055–1061.