Hybrid Systems Verification and Robotics

André Platzer

aplatzer@cs.cmu.edu
Computer Science Department
Carnegie Mellon University, Pittsburgh, PA

http://symbolaris.com/
Outline

1 Hybrid Systems Applications

2 Logic for Hybrid Systems

3 Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary

4 Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness

5 Survey

6 Summary
Can you trust a computer to control physics?
Outline

1 Hybrid Systems Applications

2 Logic for Hybrid Systems

3 Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary

4 Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness

5 Survey

6 Summary
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Hybrid Systems Analysis

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Fixed rule describing state evolution with both

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Fixed rule describing state evolution with both
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both:
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
Outline

1. Hybrid Systems Applications
2. Logic for Hybrid Systems
3. Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary
4. Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness
5. Survey
6. Summary
differential dynamic logic

\[ d\mathcal{L} = DL + HP \]
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_\mathbb{R} \]
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_R \]
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_\mathbb{R} \]
dL = FOL_R

∀ M ∃ SB ...
∀ t ≥ 0 ...

v ≤ 1 ∨ v^2 ≤ 2b(M − z)
dL = FOL<sub>R</sub>

\[ \forall M \exists SB \ldots \]

\[ \forall t \geq 0 \ldots \]

\[ v \leq 1 \lor v^2 \leq 2b(M - z) \]
dL = FOL$_R$ +

$v^2 \leq 2b$
dL = FOL\mathbb{R} + ML

\Box v^2 \leq 2b
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_R + \text{DL} \]
differential dynamic logic
dL = FOLR + DL + HP

\[ v^2 \leq 2b \]

\[ [z'' = a] \ v^2 \leq 2b \]
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_\mathbb{R} + \text{DL} + \text{HP} \]

\[ [\text{if}(z > SB) a := -b; \; z'' = a] \; v^2 \leq 2b \]
differential dynamic logic

\[ \mathcal{dL} = \text{FOL}_\mathbb{R} + \text{DL} + \text{HP} \]

\[ [\text{if}(z > SB) a := -b; \ z'' = a] \ v^2 \leq 2b \]

hybrid program
Logic for Hybrid Systems

\[ d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP} \]

\[ C \rightarrow [\text{if}(z > SB) \ a := -b; \ z'' = a] \ v^2 \leq 2b \]

hybrid program

\[ v^2 \leq 2b \]
Logic for Hybrid Systems

differential dynamic logic
\[ \mathcal{dL} = \text{FOL}_\mathbb{R} + \text{DL} + \text{HP} \]

\[ C \rightarrow [\text{if}(z > SB) a := -b; \ z'' = a] \ v^2 \leq 2b \]

Initial condition

hybrid program

v^2 \leq 2b
Logic for Hybrid Systems

differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP} \]

Initial condition

System dynamics

Hybrid program

\[ C \rightarrow [\text{if}(z > SB) a := -b; \ z'' = a] \forall^2 \leq 2b \]
Logic for Hybrid Systems

differential dynamic logic

dŁ = FOL_R + DL + HP

\[ C \rightarrow [\text{if}(z > SB) \ a := -b; \ z'' = a] \ v^2 \leq 2b \]

Initial condition
System dynamics
Post condition

hybrid program

v^2 \leq 2b

hybrid program

v^2 \leq 2b
Outline

1. Hybrid Systems Applications
2. Logic for Hybrid Systems
3. Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary
4. Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness
5. Survey
6. Summary
Follow all transitions of the system from a set of states
≈ set-valued simulation
Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$. 
Definition (Model Checking Problem)

Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$. 
Definition (Image Computation)

\[ Post_A(Y) := \{ q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A \} \]
Definition (Image Computation)

\[ Post_A(Y) := \{ q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A \} \]
Model Checking in a Nutshell

**Definition (Image Computation)**

\[ Post_A(Y) := \{ q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A \} \]
**Definition (Image Computation)**

\[ Post_A(Y) := \{ q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A \} \]

\[ Post_A^*(Y) := \bigcup_{n \in \mathbb{N}} Post_A^n(Y) = \mu Z. (Y \cup Z \cup Post_A(Z)) \]
Definition (Image Computation)

\[ Post_A(Y) := \{ q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A \} \]
Definition (Image Computation)

\[ Post_A(Y) := \{ q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A \} \]

\[ Post^*_A(Y) := \bigcup_{n \in \mathbb{N}} Post^n_A(Y) = \mu Z . (Y \cup Z \cup Post_A(Z)) \]
Uncountably state spaces require extra care
Analyse image computation problem in hybrid systems
Approximation refinement techniques and their limits
Representation of regions in state space
Numerical versus symbolic algorithms
1.421 ∈ Q versus x^2 + 2xy term computations
- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  \[ 1.421 \in \mathbb{Q} \text{ versus } x^2 + 2xy \text{ term computations} \]
- Analyse image computation problem in hybrid systems
- Approximation refinement techniques and their limits
- Representation of regions in state space
- Numerical versus symbolic algorithms
  \[1.421 \in \mathbb{Q} \text{ versus } x^2 + 2xy \text{ term computations}\]
Analyse image computation problem in hybrid systems
Approximation refinement techniques and their limits
Representation of regions in state space
Numerical versus symbolic algorithms
\(1.421 \in \mathbb{Q}\) versus \(x^2 + 2xy\) term computations
• Analyse image computation problem in hybrid systems
• Approximation refinement techniques and their limits
• Representation of regions in state space
• Numerical versus symbolic algorithms
  \(1.421 \in \mathbb{Q}\) versus \(x^2 + 2xy\) term computations
AMC($B$ reachable from $I$ in $H$):

1. $A := \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+ \epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow H$ safe
AMC($B$ reachable from $I$ in $H$):

1. $A := \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\implies H$ safe
AMC($B$ reachable from $I$ in $H$):

1. $A := \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe
AMC("$B$ reachable from $I$ in $H$"):

1. $A \leftarrow \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow H$ safe
AMC($B$ reachable from $I$ in $H$):

1. $A := \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe
AMC\((B \text{ reachable from } I \text{ in } H)\):

1. \(A := \text{approx}(H)\) uniformly
2. blur by uniform approximation error +\(\epsilon\)
3. check\((B \text{ reachable from } I \text{ in } A + \epsilon)\)
4. \(B \text{ not reachable } \Rightarrow H \text{ safe}\)
AMC: Exact Image Computation

AMC($B$ reachable from $I$ in $H$):

1. $A := \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow H$ safe

Proposition (Semialgebraic images) (HSCC’07)

check and blur can be implemented for
- $I$ and $B$ semialgebraic (propositional combinations of $p \geq 0$)
- $A$ with polynomial flows over $\mathbb{R}$
- +Piecewise definitions
- +Rational extensions (e.g. multivariate rational splines)
AMC: Image Approximation

AMC($B$ reachable from $I$ in $H$):

1. $A := \text{approx}(H)$ uniformly
2. blur by uniform approximation error $+\epsilon$
3. check($B$ reachable from $I$ in $A + \epsilon$)
4. $B$ not reachable $\Rightarrow$ $H$ safe

Proposition (Existence of approximations) (HSCC’07)

approx exists for all uniform errors $\epsilon > 0$ when

- using polynomials to build $A$
- Flows $\varphi \in C(D, \mathbb{R}^n)$ of $H$
- $D \subset \mathbb{R} \times \mathbb{R}^n$ compact closure of an open set
Approximation can solve problems without effective exact solution
Existence of solutions may be computationally insufficient
Summary: Model Checking

- Image computation in hybrid systems model checking
  - approx uniformly
  - blur by uniform error
  - check for $B$

<table>
<thead>
<tr>
<th>flows</th>
<th>approx / image computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuous</td>
<td>uniform approx exists, but...</td>
</tr>
<tr>
<td>smooth</td>
<td>undecidable by evaluation</td>
</tr>
<tr>
<td>bounded by $b$</td>
<td>decidable</td>
</tr>
<tr>
<td>bound probabilities</td>
<td>probabilistically decidable</td>
</tr>
<tr>
<td>ODE $\ell$-Lipschitz</td>
<td>decidable</td>
</tr>
</tbody>
</table>

- Combine numerical algorithms with symbolic analysis
- Roundabout maneuver unsafe
Outline

1. Hybrid Systems Applications
2. Logic for Hybrid Systems
3. Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary
4. Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness
5. Survey
6. Summary
Verify using many simple symbolic proof steps
\[ \begin{align*}
\text{[:=]} & \quad [x := \theta][(x)] \phi x \leftrightarrow [(x)] \phi \theta \\
\text{[?] } & \quad [?H] \phi \leftrightarrow (H \rightarrow \phi) \\
\text{[’] } & \quad [x’ = f(x)] \phi \leftrightarrow \forall t \geq 0 [x := y(t)] \phi \quad (y’(t) = f(y)) \\
\text{[∪] } & \quad [\alpha \cup \beta] \phi \leftrightarrow [\alpha] \phi \land [\beta] \phi \\
\text{[;] } & \quad [\alpha; \beta] \phi \leftrightarrow [\alpha][\beta] \phi \\
\text{[*] } & \quad [\alpha^*] \phi \leftrightarrow \phi \land [\alpha][\alpha^*] \phi \\
\text{K } & \quad [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha] \phi \rightarrow [\alpha] \psi) \\
\text{I } & \quad [\alpha^*](\phi \rightarrow [\alpha] \phi) \rightarrow (\phi \rightarrow [\alpha^*] \phi) \\
\text{C } & \quad [\alpha^*] \forall v > 0 (\varphi(v) \rightarrow \langle \alpha \rangle \varphi(v - 1)) \rightarrow \forall v (\varphi(v) \rightarrow \langle \alpha^* \rangle \exists v \leq 0 \varphi(v))
\end{align*} \]
Proofs for Hybrid Systems

\[
\frac{\phi_x^\theta}{[x := \theta]\phi}
\]
\[
\begin{align*}
\phi_x^\theta & \quad [x := \theta] \phi \\
\forall t \geq 0 [x := y_x(t)] & \quad [x' = f(x)] \phi
\end{align*}
\]
Proofs for Hybrid Systems

$\phi^\theta_x$  
$[x := \theta]\phi$

$\forall t \geq 0 [x := y_x(t)] \phi$  
$[x' = f(x)] \phi$

$\phi^\theta_x$  
$x := \theta$  
$\phi$

$\phi^\theta_x$  
$x := \theta$  
$\phi$

$\phi^\theta_x$  
$x := \theta$  
$\phi$

$x' = f(x)$  
$\phi$

$x := y_x(t)$
Proofs for Hybrid Systems

compositional semantics ⇒ compositional rules!
Proofs for Hybrid Systems

\[
\frac{[\alpha] \phi \land [\beta] \phi}{[\alpha \cup \beta] \phi}
\]
Proofs for Hybrid Systems

\[ [\alpha] \phi \land [\beta] \phi \quad \frac{\alpha \cup \beta}{[\alpha \cup \beta] \phi} \]

\[ [\alpha][\beta] \phi \quad \frac{\alpha; \beta}{[\alpha; \beta] \phi} \]

\[ [\alpha][\beta] \phi \quad [\beta] \phi \quad \beta \quad \phi \]
Proofs for Hybrid Systems

$\left[\alpha\right] \phi \land \left[\beta\right] \phi$

$\frac{\left[\alpha\right] \phi \land \left[\beta\right] \phi}{\left[\alpha \cup \beta\right] \phi}$

$\left[\alpha\right] \left[\beta\right] \phi$

$\frac{\left[\alpha\right] \left[\beta\right] \phi}{\left[\alpha ; \beta\right] \phi}$

$\phi \quad (\phi \to \left[\alpha\right] \phi)$

$\frac{\phi \to \left[\alpha\right] \phi}{\left[\alpha^*\right] \phi}$
\begin{align*}
v \geq 0, z < m \rightarrow & \exists t \geq 0 \langle z := -\frac{b}{2} t^2 + vt + z \rangle z > m \\
v \geq 0, z < m \rightarrow & \langle z' = v, v' = -b \rangle z > m \\
v \geq 0 \land z < m \rightarrow & \langle z' = v, v' = -b \rangle z > m
\end{align*}
\begin{align*}
\text{Deduction Modulo (Free Variables for Automation)}
\text{For requantification, not for unification} \\
v \geq 0, z < m & \rightarrow T \geq 0 \\
v \geq 0, z < m & \rightarrow \langle z := -\frac{b}{2} T^2 + v T + z \rangle z > m \\
v \geq 0, z < m & \rightarrow T \geq 0 \land \langle z := -\frac{b}{2} T^2 + v T + z \rangle z > m \\
v \geq 0, z < m & \rightarrow \exists t \geq 0 \langle z := -\frac{b}{2} t^2 + vt + z \rangle z > m \\
v \geq 0, z < m & \rightarrow \langle z' = v, v' = -b \rangle z > m \\
v \geq 0 \land z < m & \rightarrow \langle z' = v, v' = -b \rangle z > m
\end{align*}
\[
v \geq 0, z < m \rightarrow \exists T \left( \ldots T \geq 0 \land -\frac{b}{2} T^2 + v T + z > m \right)
\]

\[
v \geq 0, z < m \rightarrow T \geq 0
\]

\[
v \geq 0, z < m \rightarrow \langle z := \frac{-b}{2} T^2 + v T + z \rangle z > m
\]

\[
v \geq 0, z < m \rightarrow \exists t \geq 0 \langle z := -\frac{b}{2} t^2 + v t + z \rangle z > m
\]

\[
v \geq 0, z < m \rightarrow \langle z' = v, v' = -b \rangle z > m
\]

\[
v \geq 0 \land z < m \rightarrow \langle z' = v, v' = -b \rangle z > m
\]
\[
\begin{align*}
v \geq 0, z < m &\rightarrow \text{QE}\left( \exists T \left( \ldots T \geq 0 \land -\frac{b}{2} T^2 + v T + z > m \right) \right) \\
v \geq 0, z < m &\rightarrow T \geq 0 \\
v \geq 0, z < m &\rightarrow \left\langle z := -\frac{b}{2} T^2 + v T + z \right\rangle z > m \\
v \geq 0, z < m &\rightarrow \exists t \geq 0 \left\langle z := -\frac{b}{2} t^2 + vt + z \right\rangle z > m \\
v \geq 0, z < m &\rightarrow \left\langle z' = v, v' = -b \right\rangle z > m \\
v \geq 0 \land z < m &\rightarrow \left\langle z' = v, v' = -b \right\rangle z > m
\end{align*}
\]
\[ v \geq 0, z < m \rightarrow v^2 > 2b(m - z) \]

\[
\begin{align*}
& v \geq 0, z < m \rightarrow T \geq 0 \\
& v \geq 0, z < m \rightarrow \langle z := -\frac{b}{2}T^2 + vT + z \rangle z > m \\
& v \geq 0, z < m \rightarrow T \geq 0 \land \langle z := -\frac{b}{2}T^2 + vT + z \rangle z > m \\
& v \geq 0, z < m \rightarrow \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > m \\
& v \geq 0, z < m \rightarrow \langle z' = v, v' = -b \rangle z > m \\
& v \geq 0 \land z < m \rightarrow \langle z' = v, v' = -b \rangle z > m
\end{align*}
\]
For requantification, not for unification

\[ v \geq 0, \ z < m \rightarrow \forall \ T \left( \exists T \left( \ldots T \geq 0 \land -\frac{b}{2} T^2 + v T + z > m \right) \right) \]

\[ v \geq 0, \ z < m \rightarrow -\frac{b}{2} T^2 + v T + z > m \]

\[ v \geq 0, \ z < m \rightarrow T \geq 0 \land \langle z := -\frac{b}{2} T^2 + v T + z \rangle z > m \]

\[ v \geq 0, \ z < m \rightarrow \exists t \geq 0 \langle z := -\frac{b}{2} t^2 + vt + z \rangle z > m \]

\[ v \geq 0, \ z < m \rightarrow \langle z' = v, \ v' = -b \rangle z > m \]

\[ v \geq 0 \land z < m \rightarrow \langle z' = v, \ v' = -b \rangle z > m \]
| Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008) | $\mathcal{dL}$ calculus is a sound & complete axiomatization of hybrid systems relative to differential equations. | ![Proof 15pp] |
Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)
\[ d\mathcal{L} \text{ calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.} \]

Proof 15pp

Theorem (Discrete Relative Completeness) (LICS’12)
\[ d\mathcal{L} \text{ calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.} \]

Proof +10pp
Complete Proof Theory of Hybrid Systems

Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)

\( d\mathcal{L} \) calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.

Proof 15pp

Theorem (Discrete Relative Completeness) (LICS’12)

\( d\mathcal{L} \) calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.

Proof +10pp

System

Hybrid

Continuous

Discrete

André Platzer (CMU)
Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)

$dL$ calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.

Proof 15pp

Theorem (Discrete Relative Completeness) (LICS’12)

$dL$ calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.

Proof +10pp
Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)
\( dL \) calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.

Corollary (Relative Decidability)
Verification & synthesis decidable relative to differential equations.

Corollary (Relative Extension)
All research on differential equations extends to hybrid systems.

Theorem (Discrete Relative Completeness) (LICS’12)
\( dL \) calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.

Proof 15pp

Proof +10pp
Outline

1. Hybrid Systems Applications
2. Logic for Hybrid Systems
3. Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary
4. Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness
5. Survey
6. Summary
\[ [\alpha] \Box \phi \quad \alpha \quad \phi \]

\[ \langle \alpha \rangle^P \phi \quad P(\phi) \]

\[ \psi \rightarrow [\alpha] \phi \]

\[ \psi \rightarrow [\alpha] \phi \]

\[ \psi \rightarrow [\alpha] \phi \]

\[ \psi \rightarrow [\alpha] \phi \]

\[ \psi \rightarrow [\alpha] \phi \]

---

KeYmaera Prover

Strategy

Rule Engine

Proof

Input File

Rule base

Solvers

Mathematica

QEPCAD

Orbital

For

∧,

∪,

:=

do decompose

Repeat until fixedpoint

Details

x′ = ...

do diffsat

for α ∗ do loopsat

---

André Platzer (CMU)

Hybrid Systems Verification and Robotics

RSS-FMRA 22 / 25
Successful Hybrid Systems Proofs

\[ SB := \left( \frac{\text{amax}}{b} + 1 \right) \cdot \text{ep} \cdot v + \frac{v^2 - d^2}{2 \cdot b} + \frac{\left( \frac{\text{amax}}{b} + 1 \right) \cdot \text{amax} \cdot \text{ep}^2}{2} \]

\[ ?d \geq 0 \land d^2 - d^2 \leq 2 \cdot b \cdot (m - m_0) \land v_{de s} \geq 0 \]

\[ v_{des} := * \]

\[ d := * \]

\[ m := * \]

\[ m_0 := m \]

\[ d_0 := d \]

\[ \text{state := brake} \]

\[ ?v \leq v_{des} \]

\[ ?v \geq v_{des} \]

\[ a := -b \]

\[ ?a \geq 0 \land a \leq \text{amax} \]

\[ a := * \]

\[ ?a \leq 0 \land a \geq -b \]

\[ a := * \]

\[ t := 0 \]

\[ ?m - z \leq SB \mid \text{state} = \text{brake} \]

\[ ?m - z \geq SB \mid \text{state} \neq \text{brake} \]
Successful Hybrid Systems Proofs

André Platzer (CMU)

Hybrid Systems Verification and Robotics

RSS-FMRA 23 / 25
Successful Hybrid Systems Proofs

Andre Platzer (CMU)
Hybrid Systems Verification and Robotics

RSS-FMRA 23 / 25
Outline

1. Hybrid Systems Applications
2. Logic for Hybrid Systems
3. Model Checking
   - Successive Image Computation
   - Image Computation in Hybrid Systems
   - Approximation Refinement Model Checking
   - Summary
4. Proofs for Hybrid Systems
   - Proof Rules
   - Soundness and Completeness
5. Survey
6. Summary
Hybrid system models
Discrete dynamics
Continuous dynamics
Correctness properties
Safety, liveness . . .

Model checking
Logic & proofs
Cyber-physical systems
Differential invariants

KeYmaera
Thomas A. Henzinger.
The theory of hybrid automata.

Rajeev Alur.
Formal verification of hybrid systems.

André Platzer.
Logics of dynamical systems.

André Platzer.
Differential dynamic logic for hybrid systems.

André Platzer.
*Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.*
André Platzer and Jan-David Quesel.
KeYmaera: A hybrid theorem prover for hybrid systems.

Ian M. Mitchell and Jeremy A. Templeton.
A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems.

Stefan Ratschan and Zhikun She.
Safety verification of hybrid systems by constraint propagation-based abstraction refinement.

Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler.
SpaceEx: Scalable verification of hybrid systems.

Goran Frehse.
PHAVer: algorithmic verification of hybrid systems past HyTech.

André Platzer and Edmund M. Clarke.
The image computation problem in hybrid systems model checking.

Pieter Collins.
Optimal semicomputable approximations to reachable and invariant sets.

Edmund M. Clarke, Ansgar Fehnker, Zhi Han, Bruce H. Krogh, Joël Ouaknine, Olaf Stursberg, and Michael Theobald.
Abstraction and counterexample-guided refinement in model checking of hybrid systems.


Alongkrit Chutinan and Bruce H. Krogh.
Computational techniques for hybrid system verification.


Carla Piazza, Marco Antoniotti, Venkatesh Mysore, Alberto Policriti, Franz Winkler, and Bud Mishra.
Algorithmic algebraic model checking I: Challenges from systems biology.