Theorem Proving for Dynamic Systems

André Platzer

aplatzer@cs.cmu.edu
Logical Systems Lab
Carnegie Mellon University, Pittsburgh, PA
How can we design computers that are guaranteed to interact correctly with the physical world?
Hybrid Systems

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

Challenge

Hybrid Systems

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

1. More than computers: no `NullPointerException` ⇒ safe

More than physics: braking control

\[ v^2 \leq 2b(M - z) \] ⇒ safe

Joint dynamics requires:

\[ SB \geq \frac{v^2}{2b} + a^2 \varepsilon^2 + \frac{a^2 b \varepsilon v}{2} + a^2 \varepsilon^2 + \varepsilon v \]

André Platzer (CMU)

Theorem Proving for Dynamic Systems
Hybrid Systems Analysis: Train Control

Challenge

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

1. More than computers: no NullPointerException ⇒ safe
2. More than physics: braking control $v^2 \leq 2b(M - z) \not\Rightarrow$ safe
Hybrid Systems Analysis: Train Control

Challenge

Hybrid Systems

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

1. More than computers: no NullPointerException \( \not\Rightarrow \) safe
2. More than physics: braking control \( v^2 \leq 2b(M - z) \) \( \not\Rightarrow \) safe
3. Joint dynamics requires:

\[
SB \geq \frac{v^2}{2b} + \frac{a^2\varepsilon^2}{2b} + \frac{a}{b}\varepsilon v + \frac{a}{2}\varepsilon^2 + \varepsilon v \ldots
\]
Hybrid Systems Analysis: Train Control

Challenge

Hybrid Systems

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

Challenge

Hybrid Systems

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

Hybrid Systems

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

Challenge

Hybrid Systems

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

$$\begin{align*}
0 &\quad M \\
\text{V} &\quad \text{z}
\end{align*}$$
Challenge

Hybrid Systems

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

Challenge

Hybrid Systems

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

\[ SB \geq \frac{v^2}{2b} + \frac{a^2\varepsilon^2}{2b} + \frac{a}{b}\varepsilon v + \frac{a}{2}\varepsilon^2 + \varepsilon v \]
dL = DL + HP

André Platzer.
Differential dynamic logic for hybrid systems.
Logic for Hybrid Systems

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

\[ v^2 \leq 2b(M - z) \]

\textbf{André Platzer.}

Differential dynamic logic for hybrid systems.

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

André Platzer.
Differential dynamic logic for hybrid systems.
differential dynamic logic
\( d\mathcal{L} = \text{FOL}_R \)

\[
\begin{align*}
\forall t \geq 0 & \quad \exists z \quad v \leq 1 \land v^2 \leq 2b(M - z)
\end{align*}
\]

André Platzer. 
Differential dynamic logic for hybrid systems. 
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_R \]

\[ v \leq 1 \lor v^2 \leq 2b(M - z) \]

André Platzer.
Differential dynamic logic for hybrid systems.
differential dynamic logic
\[ \mathcal{dL} = \text{FOL}_\mathbb{R} \]

\[ \forall MA \exists SB \ldots \]

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

Andre Platzer.
Differential dynamic logic for hybrid systems.
dL = FOL$_R$ +

$\nu^2 \leq 2b$

André Platzer.  
Differential dynamic logic for hybrid systems.  
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{ML} \]
Logic for Hybrid Systems

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

André Platzer.

Differential dynamic logic for hybrid systems.

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

\[ [z'' = a] \, v^2 \leq 2b \]

\[ v^2 \leq 2b \]

André Platzer.
Differential dynamic logic for hybrid systems.
differential dynamic logic
\[ d\mathcal{L} = \text{FOL}_R + \text{DL} + \text{HP} \]

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

\[ v^2 \leq 2b \]

André Platzer.
Differential dynamic logic for hybrid systems.
dL = FOL\(_R\) + DL + HP

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

Logic for Hybrid Systems

**differential dynamic logic**

\[ d\mathcal{L} = \text{FOL}_R + DL + HP \]

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

Andre Platzer.

Differential dynamic logic for hybrid systems.

\(-2 \leq a \leq 2\) \\
\(\land b^2 \geq \frac{1}{3}\)
Proof by Symbolic Decomposition

\[ \langle x := f(x) \rangle \phi \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ \langle x := f(x) \rangle \phi \]

\[ \phi_x^{f(x)} \]

\[ x := f(x) \]

\[ V \rightarrow W \]
Proof by Symbolic Decomposition

\[ \langle x := f(x) \rangle \phi \]

\[ x := f(x) \]

\[ f(x) \]

\[ \langle x' = f(x) \rangle \phi \]

\[ x' = f(x) \]

\[ \phi \]

\[ \phi \]

\[ \phi \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ \langle x := f(x) \rangle \phi \]

\[ \phi_x^f(x) \]

\[ x := f(x) \]

\[ \exists t \geq 0 \langle x := y_x(t) \rangle \phi \]

\[ \langle x' = f(x) \rangle \phi \]

\[ x' = f(x) \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ \langle x := f(x) \rangle \phi \]

\[ \exists t \geq 0 \langle x := y_x(t) \rangle \phi \]
Proof by Symbolic Decomposition

\[ [\alpha \cup \beta] \phi \]

\[ \alpha \cup \beta \]

\[ \beta \]

\[ \alpha \]

\[ W_1 \]

\[ W_2 \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ \alpha \cup \beta \]

\[ \phi \]

\[ [\alpha] \phi \land [\beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ \alpha \]

\[ \beta \]

\[ \alpha \cup \beta \]

\[ \phi \]

\[ \phi \]

\[ \phi \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ [\alpha] \phi \land [\beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ \alpha \cup \beta \]

\[ \alpha; \beta \]

\[ [\alpha; \beta] \phi \]

\[ [\alpha] \phi \land [\beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ \alpha \cup \beta \]

\[ \alpha; \beta \]

\[ [\alpha; \beta] \phi \]

\[ [\beta] \phi \]

\[ [\beta] \phi \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ \alpha \cup \beta \]

\[ \alpha \land [\beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ [\alpha; \beta] \phi \]

\[ [\alpha][\beta] \phi \]

\[ [\alpha; \beta] \phi \]

\[ [\alpha] \phi \land [\beta] \phi \]

\[ \alpha \cup \beta \]

\[ \alpha ; \beta \]

\[ [\beta] \phi \]

\[ [\beta] \phi \]

\[ \alpha \]

\[ \beta \]

\[ \phi \]

\[ \phi \]

\[ \phi \]
Proof by Symbolic Decomposition

\[ [\alpha] \phi \land [\beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ [\alpha; \beta] \phi \]

\[ [\alpha][\beta] \phi \]

\[ [\alpha^*] \phi \]

\[ \phi \rightarrow [\alpha] \phi \]
Proof by Symbolic Decomposition

\[\lbrack \alpha \rbrack \phi \land \lbrack \beta \rbrack \phi\]

\[\lbrack \alpha \cup \beta \rbrack \phi\]

\[\lbrack \alpha ; \beta \rbrack \phi\]

\[\lbrack \alpha \rbrack \lbrack \beta \rbrack \phi\]

\[\lbrack \beta \rbrack \phi\]

\[\phi \rightarrow \lbrack \alpha \rbrack \phi\]

\[\lbrack \alpha^* \rbrack \phi\]
Proof by Symbolic Decomposition

\[ [\alpha] \phi \land [\beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ [\alpha \cup \beta] \phi \]

\[ [\alpha; \beta] \phi \]

\[ [\alpha; \beta] \phi \]

\[ [\alpha; \beta] \phi \]

\[ [\beta] \phi \]

\[ \forall_{cl}(\phi \rightarrow [\alpha] \phi) \land \phi \]

André Platzer (CMU)

Theorem Proving for Dynamic Systems
Theorem (Relative Completeness)

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

Corollary (Proof-theoretical Alignment)
verification of hybrid systems = verification of dynamical systems!

Corollary (Compositionality)
hybrid systems can be verified by recursive decomposition

André Platzer.
Differential dynamic logic for hybrid systems.
## Soundness and Completeness

**Theorem (Relative Completeness)**

\[ \mathcal{DL} \text{ calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.} \]

**Proof Outline 15p**

**Corollary (Proof-theoretical Alignment)**

verification of hybrid systems $=$ verification of dynamical systems!

**Corollary (Compositionality)**

hybrid systems can be verified by recursive decomposition

---

**André Platzer.**

Differential dynamic logic for hybrid systems.

Hybrid Systems Analysis: Air Traffic Control
Hybrid Systems Analysis: Air Traffic Control

\[
\begin{bmatrix}
    x_1' \\
    x_2' \\
    \vartheta'
\end{bmatrix} =
\begin{bmatrix}
    -v_1 + v_2 \cos \vartheta + \omega x_2 \\
    v_2 \sin \vartheta - \omega x_1 \\
    \varrho - \omega
\end{bmatrix}
\]
Example ("Solving" differential equations)

\[ x_1(t) = \frac{1}{\omega \varphi} \left( x_1 \omega \varphi \cos t \varphi - v_2 \omega \cos t \varphi \sin \vartheta + v_2 \omega \cos t \varphi \cos t \varphi \sin \vartheta - v_1 \varphi \sin t \omega + x_2 \omega \varphi \sin t \omega - v_2 \omega \cos \vartheta \cos t \varphi \sin t \omega - v_2 \omega \sqrt{1 - \sin^2 \vartheta} \sin t \omega + v_2 \omega \cos \vartheta \cos t \omega \sin t \varphi + v_2 \omega \sin \vartheta \sin t \omega \sin t \varphi \right) \ldots \]
Example ("Solving" differential equations)

\[
\forall t \geq 0 \quad \frac{1}{\omega_2} \left( x_1 \omega_2 \cos t \omega - v_2 \omega \cos t \omega \sin \vartheta + v_2 \omega \cos t \omega \cos \varphi \sin \vartheta - v_1 \varphi \sin t \omega \\
+ x_2 \omega_2 \sin t \omega - v_2 \omega \cos \vartheta \cos \varphi \sin t \omega - v_2 \omega \sqrt{1 - \sin \vartheta^2} \sin t \omega \\
+ v_2 \omega \cos \vartheta \cos t \omega \sin \varphi + v_2 \omega \sin \vartheta \sin t \omega \sin t \varphi \right) \ldots
\]
Idea: Exploit Vector Field of Differential Equations

“Definition” (Differential Invariant)

“Logical formula that remains true in the direction of the dynamics”
Idea: Exploit Vector Field of Differential Equations

“Definition” (Differential Invariant)

“Logical formula that remains true in the direction of the dynamics”
Idea: Exploit Vector Field of Differential Equations

“Definition” (Differential Invariant)

“Logical formula that remains true in the direction of the dynamics”
“Definition” (Differential Invariant)

“Logical formula that remains true in the direction of the dynamics”
Differential Induction for Aircraft Roundabouts

$\vdash [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2$
Differential Induction for Aircraft Roundabouts

\[ \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \ldots \]

\[ \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \]

\[ \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \]

\[ [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, \ldots] (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]
Differential Induction for Aircraft Roundabouts

⊢ \frac{\partial \|x-y\|^2}{\partial x_1} x_1' + \frac{\partial \|x-y\|^2}{\partial y_1} y_1' + \frac{\partial \|x-y\|^2}{\partial x_2} x_2' + \frac{\partial \|x-y\|^2}{\partial y_2} y_2' \geq \frac{\partial p^2}{\partial x_1} x_1' \ldots

⊢ [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, \ldots] (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2
Differential Induction for Aircraft Roundabouts

\[ \frac{\partial}{\partial x_1} \| x - y \|^2 d_1 + \frac{\partial}{\partial y_1} \| x - y \|^2 e_1 + \frac{\partial}{\partial x_2} \| x - y \|^2 d_2 + \frac{\partial}{\partial y_2} \| x - y \|^2 e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]

\[ \frac{\partial}{\partial x_1} \| x - y \|^2 d_1 - \frac{\partial}{\partial y_1} \| x - y \|^2 e_1 - \frac{\partial}{\partial x_2} \| x - y \|^2 d_2 - \frac{\partial}{\partial y_2} \| x - y \|^2 e_2 = -\omega (d_2 - e_2) \]

\[ x_1' = d_1, \quad d_1' = -\omega d_2, \quad x_2' = d_2, \quad d_2' = \omega d_1, \ldots \]

\[ (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]
\[ \vdash \frac{\partial \|x - y\|^2}{\partial x_1} d_1 + \frac{\partial \|x - y\|^2}{\partial y_1} e_1 + \frac{\partial \|x - y\|^2}{\partial x_2} d_2 + \frac{\partial \|x - y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]

\[ \vdash [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots] (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]
Differential Induction for Aircraft Roundabouts

\[ 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]

\[ \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]

\[ [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots] (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]
Differential Induction for Aircraft Roundabouts

\[ 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]

\[ \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]

\[ x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots \]

\[ (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]

\[ d'_1 = -\omega d_2, e'_1 = -\omega e_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots \]

\[ d_1 - e_1 = -\omega(x_2 - y_2) \]
\[ \vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0 \]
\[ \vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]
\[ \vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial \mathbf{p}_2}{\partial x_1} d_1 \ldots \]
\[ \vdash [x_1' = d_1, d_1' = -\omega d_2, x_2' = d_2, d_2' = \omega d_1, \ldots](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p_2^2 \]

.. \quad \vdash [d_1' = -\omega d_2, e_1' = -\omega e_2, x_2' = d_2, d_2' = \omega d_1, \ldots]d_1 - e_1 = -\omega(x_2 - y_2)
\[ \vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0 \]
\[ \vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]
\[ \vdash \frac{\partial \|x - y\|^2}{\partial x_1} d_1 + \frac{\partial \|x - y\|^2}{\partial y_1} e_1 + \frac{\partial \|x - y\|^2}{\partial x_2} d_2 + \frac{\partial \|x - y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]
\[ \vdash [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, ..](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]
Differential Induction for Aircraft Roundabouts

\[ \vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0 \]

\[ \vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]

\[ \vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]

\[ \vdash [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]
\[ \vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0 \]
\[ \vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]
\[ \vdash \frac{\partial \parallel x-y \parallel^2}{\partial x_1} d_1 + \frac{\partial \parallel x-y \parallel^2}{\partial y_1} e_1 + \frac{\partial \parallel x-y \parallel^2}{\partial x_2} d_2 + \frac{\partial \parallel x-y \parallel^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]
\[ \vdash [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]

\[ \vdash \frac{\partial (d_1-e_1)}{\partial d_1} (-\omega d_2) + \frac{\partial (d_1-e_1)}{\partial e_1} (-\omega e_2) = -\frac{\partial \omega(x_2-y_2)}{\partial x_2} d_2 - \frac{\partial \omega(x_2-y_2)}{\partial y_2} e_2 \]
\[ \ldots \vdash [d'_1 = -\omega d_2, e'_1 = -\omega e_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots]d_1 - e_1 = -\omega(x_2 - y_2) \]
Differential Induction for Aircraft Roundabouts

\[\vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0\]

\[\vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0\]

\[\vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots\]

\[\vdash [x'_1 = d_1, \quad d'_1 = -\omega d_2, \quad x'_2 = d_2, \quad d'_2 = \omega d_1, \ldots](x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2\]

\[\vdash -\omega d_2 + \omega e_2 = -\omega(d_2 - e_2)\]

\[\vdash \frac{\partial(d_1 - e_1)}{\partial d_1} (-\omega d_2) + \frac{\partial(d_1 - e_1)}{\partial e_1} (-\omega e_2) = -\frac{\partial \omega(x_2 - y_2)}{\partial x_2} d_2 - \frac{\partial \omega(x_2 - y_2)}{\partial y_2} e_2\]

\[\vdash [d'_1 = -\omega d_2, \quad e'_1 = -\omega e_2, \quad x'_2 = d_2, \quad d'_2 = \omega d_1, \ldots]d_1 - e_1 = -\omega(x_2 - y_2)\]
Differential Induction & Differential Cuts

\[ \vdash 2(x_1 - y_1)(-\omega(x_2 - y_2)) + 2(x_2 - y_2)\omega(x_1 - y_1) \geq 0 \]
\[ \vdash 2(x_1 - y_1)(d_1 - e_1) + 2(x_2 - y_2)(d_2 - e_2) \geq 0 \]
\[ \vdash \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]
\[ \vdash [x'_1 = d_1, d'_1 = -\omega d_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots] (x_1 - y_1)^2 + (x_2 - y_2)^2 \geq p^2 \]

Proposition (Differential cut saturation)

If \( F \) is a differential invariant of \([x' = \theta \land H]\phi\), then
\[ [x' = \theta \land H]\phi \quad \text{iff} \quad [x' = \theta \land H \land F]\phi \]

\[ \vdash -\omega d_2 + \omega e_2 = -\omega(d_2 - e_2) \]
\[ \vdash \frac{\partial(d_1-e_1)}{\partial d_1}(-\omega d_2) + \frac{\partial(d_1-e_1)}{\partial e_1}(-\omega e_2) = -\frac{\partial \omega(x_2-y_2)}{\partial x_2} d_2 - \frac{\partial \omega(x_2-y_2)}{\partial y_2} e_2 \]
\[ \vdash [d'_1 = -\omega d_2, e'_1 = -\omega e_2, x'_2 = d_2, d'_2 = \omega d_1, \ldots] d_1 - e_1 = -\omega(x_2 - y_2) \]
\[ \frac{\partial \|x-y\|^2}{\partial x_1} d_1 + \frac{\partial \|x-y\|^2}{\partial y_1} e_1 + \frac{\partial \|x-y\|^2}{\partial x_2} d_2 + \frac{\partial \|x-y\|^2}{\partial y_2} e_2 \geq \frac{\partial p^2}{\partial x_1} d_1 \ldots \]

\[ x_1' = d_1, \quad d_1' = -\omega d_2, \quad x_2' = d_2, \quad d_2' = \omega d_1, \ldots \]

\[ d_1 - e_1 = -\omega (x_2 - y_2) \]
Successful Hybrid Systems Analysis

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

\[ ?d >= 0 & do^2 - d^2 <= 2b(m - mo) & vde s >= 0 \]

\[ vdes := * \]

\[ d := * \]

\[ m := * \]

\[ mo := m \]

\[ do := d \]

\[ state := brake \]

\[ ?v <= vdes \]

\[ ?v >= vdes \]

\[ t := 0 \]

\[ ?m - z <= SB | state = brake \]

\[ ?m - z >= SB & state != brake \]

\[ a := -b \]

\[ ?a >= 0 & a <= amax \]

\[ a := * \]

\[ ?a <= 0 & a >= -b \]

\[ a := * \]

\[ t := 0 \]
\[ [\alpha] \Box \phi \]

**Strategy**

**Rule Engine**

**Proof**

**Input File**

**Rule**

**Mathematica**

**QEPCAD**

**Orbital**

**KeYmaera Prover**

**Solvers**

- Mathematica
- QEPCAD
- Orbital

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\[ \psi \rightarrow [\alpha] \phi \]
Theorem Proving for Dynamic Systems

Differential dynamic logic

\[ d\mathcal{L} = DL + HP \]

Verifying hybrid systems:
- Logic for hybrid systems++
- Compositional calculi
- Algorithms
- Challenging applications

KeYmaera
Verifying hybrid systems:
- Logic for hybrid systems++
- Compositional calculi
- Algorithms
- Challenging applications
André Platzer.

Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.

André Platzer.

Differential dynamic logic for hybrid systems.

André Platzer.

Differential-algebraic dynamic logic for differential-algebraic programs.

André Platzer and Edmund M. Clarke.

Computing differential invariants of hybrid systems as fixedpoints.

André Platzer and Jan-David Quesel.

KeYmaera: A hybrid theorem prover for hybrid systems.