1. CPS are Multi-Dynamical Systems
   - Hybrid Systems / Games / Stochastic / Distributed Hybrid Systems

2. Differential Dynamic Logic

3. Axioms and Proofs for CPS

4. Differential Invariants for Differential Equations
   - Differential Invariants
   - Example: Elementary Differential Invariants

5. Applications
   - Ground Robot Navigation
   - Airborne Collision Avoidance System
   - KeYmaera X

6. Summary
Which control decisions are safe for aircraft collision avoidance?

Cyber-Physical Systems

CPSs combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone.
CPSs Promise Transformative Impact!

Prospects: Safe & Efficient

Driver assistance
Autonomous cars

Pilot decision support
Autopilots / UAVs

Train protection
Robots near humans

Prerequisite: CPSs need to be safe

How do we make sure CPSs make the world a better place?
Can you trust a computer to control physics?

Depends on how it has been programmed.
And on what will happen if it malfunctions.

Rationale
Safety guarantees require analytic foundations.
A common foundational core helps all application domains.
Foundations revolutionized digital computer science & our society.
Need even stronger foundations when software reaches out into our physical world.

CPSs deserve proofs as safety evidence!

André Platzer (CMU)
Can you trust a computer to control physics?

1. Depends on how it has been programmed
2. And on what will happen if it malfunctions

Rationale

1. Safety guarantees require analytic foundations.
2. A common foundational core helps all application domains.
3. Foundations revolutionized digital computer science & our society.
4. Need even stronger foundations when software reaches out into our physical world.

CPSs deserve proofs as safety evidence!
CPSs are characterized by multiple facets of dynamical systems.

CPS are Multi-Dynamical Systems

CPS Dynamics

CPS Compositions

CPS are characterized by multiple dynamical effects.

Tame Parts

Exploiting compositionality tames CPS complexity.

Descriptive simplification

Analytic simplification
CPSs are Multi-Dynamical Systems

hybrid systems

\[ HS = \text{discrete} + \text{ODE} \]

hybrid games

\[ HG = \text{HS} + \text{adversary} \]

stochastic hybrid sys.

\[ \text{SHS} = \text{HS} + \text{stochastics} \]

distributed hybrid sys.

\[ \text{DHS} = \text{HS} + \text{distributed} \]
Dynamic Logics for Dynamical Systems

- Differential Dynamic Logic
  \[ d\mathcal{L} = DL + HP \]

- Differential Game Logic
  \[ d\mathcal{G} = GL + HG \]

- Stochastic Differential Dynamic Logic
  \[ Sd\mathcal{L} = DL + SHP \]

- Quantified Differential Dynamic Logic
  \[ Qd\mathcal{L} = FOL + DL + QHP \]

JAR'08, CADE'11, LMCS'12, LICS'12, LICS'12, TOCL'15, CADE'15, JAR'17, TOCL'17

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X
Dynamic Logics for Dynamical Systems

Dynamic Logics

- DL has been introduced for programs Pratt’76, Harel, Kozen
- Its real calling are dynamical systems
- DL excels at providing simple + elegant logical foundations for dynamical systems
- CPSs are multi-dynamical systems
- DL for CPS are multi-dynamical
André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM’17 8 / 29

Concept (Differential Dynamic Logic)  

\([\alpha] \varphi \rightarrow \varphi\)
CPS Analysis

Concept (Differential Dynamic Logic) (JAR'08, LICS'12)

\[ [\alpha] \varphi \quad \varphi \]

\[ \square x \neq m \quad x \neq m \]

\[ x \neq m \]

André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM'17  8 / 29
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \rightarrow \varphi \]

\[ x \neq m \]

André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \xrightarrow{\alpha} \varphi \]

\[ x \neq m \]

\[ [\alpha] x \neq m \]

\[ x' = v, v' = a \]

ODE

\[ x' = v, v' = a \]
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \alpha \quad \varphi \]

\[ x \neq m \]

\[ a := -b \quad x' = v, v' = a \]

André Platzer (CMU)
Logic & Proofs for Cyber-Physical Systems with KeYmaera X
iFM’17 8 / 29
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \alpha \quad \varphi \]

\[ x \neq m \]

\[ [\alpha] x \neq m \quad x \neq m \]

\[ (\text{if}(\text{SB}(x, m)) a := -b) \quad x' = v, v' = a \]

\( a \)

\( x \)

\( m \)

ODE

assign

test

André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM’17 8 / 29
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \quad \alpha \quad \varphi \]

\[(\text{if} (\text{SB}(x, m)) a := -b) \ ; \ x' = v, v' = a\]

seq. compose

assign

test

ODE

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X iFM’17 8 / 29
André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM’17 8 / 29
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \rightarrow \varphi \]

\[ x \neq m \]

\[ \text{init} \rightarrow \text{post} \]

\[ \alpha \]

\[ \text{all runs} \]

\[ ((\text{if}(\text{SB}(x, m)) a := -b); x' = v, v' = a)^* \]

\[ x \neq m \]

\[ \text{post} \]
CPS Analysis

Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \varphi \quad \alpha \quad \varphi \]

\[ x \neq m \quad x \neq m \quad x \neq m \]

\[ x \neq m \land b > 0 \rightarrow \left[ ((\text{if}(SB(x, m)) \ a := -b) ; \ x' = v, v' = a)^* \right] x \neq m \]

\[ \text{all runs} \]

init

post

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X iFM’17 8 / 29
Differential Dynamic Logic $d\mathcal{L}$: Syntax

**Definition (Hybrid program $\alpha$)**

\[
x := f(x) \mid ?Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^*
\]

**Definition ($d\mathcal{L}$ Formula $P$)**

\[
e \geq \bar{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P
\]
### Differential Dynamic Logic $\mathcal{dL}$: Syntax

#### Definition (Hybrid program $\alpha$)

- **Discrete Assign**: $x := f(x) | ?Q$
- **Test Condition**: $x' = f(x) & Q$
- **Differential Equation**: $\alpha \cup \beta | \alpha; \beta | \alpha^*$

#### Definition ($\mathcal{dL}$ Formula $P$)

- $e \geq \tilde{e}$
- $\neg P$
- $P \land Q$
- $\forall x \; P$
- $\exists x \; P$
- $[\alpha]P$
- $\langle \alpha \rangle P$

**Logic & Proofs for Cyber-Physical Systems with KeYmaera X**

**JAR’08, LICS’12, JAR’17**

André Platzer (CMU)
Differential Dynamic Logic: Axiomatization

\[ x := e \] \( P(x) \leftrightarrow P(e) \)

\[ ? \] \( [? Q] P \leftrightarrow (Q \rightarrow P) \)

\[ ' \] \( [x' = f(x)] P \leftrightarrow \forall t \geq 0 [x := y(t)] P \quad (y'(t) = f(y)) \)

\[ \cup \] \( [\alpha \cup \beta] P \leftrightarrow [\alpha] P \land [\beta] P \)

\[ ; \] \( [\alpha; \beta] P \leftrightarrow [\alpha][\beta] P \)

\[ * \] \( [\alpha^*] P \leftrightarrow P \land [\alpha][\alpha^*] P \)

\( K \) \( [\alpha] (P \rightarrow Q) \rightarrow ([\alpha] P \rightarrow [\alpha] Q) \)

\( L \) \( [\alpha^*] P \leftrightarrow P \land [\alpha^*] (P \rightarrow [\alpha] P) \)

\( C \) \( [\alpha^*] \forall \nu > 0 (P(\nu) \rightarrow \langle \alpha \rangle P(\nu - 1)) \rightarrow \forall \nu (P(\nu) \rightarrow \langle \alpha^* \rangle \exists \nu \leq 0 P(\nu)) \)

 equations of truth

LICS’12, JAR’17
Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

\( \mathcal{dL} \) calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or to discrete dynamics.

Corollary (Complete Proof-theoretical Bridge)

proving continuous \( = \) proving hybrid \( = \) proving discrete
Differential Invariants for Differential Equations

\[ \dot{x} = f(x) \]

\[ \dot{y} = g(x, y) \]

\[ \text{inv} \]

\[ \text{DI} \geq, \text{DI} 

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, CAV’08, FMSP’09, LMCS’12, LICS’12, ITP’12, JAR’17
Differential Invariants for Differential Equations

\[
\begin{align*}
\dot{x} &= f(x) \\
\dot{y} &= g(x, y)
\end{align*}
\]

\[\text{inv} \quad \text{DI} = \text{DI} = \wedge, \vee \]

\[\text{Differential Invariant} \quad \text{Differential Cut} \quad \text{Differential Ghost}\]

\[\text{Logic} \quad \text{Provability theory} \quad \text{Math} \quad \text{Characteristic PDE}\]

\[\text{JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17}\]
Differential Invariants for Differential Equations

Differential Invariant

\[ \frac{dx}{dt} = f(x) \]

Differential Cut

\[ \frac{dy}{dt} = g(x, y) \]

Differential Ghost

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X
Differential Invariants for Differential Equations

**Differential Invariant**

\[ x' = f(x) \]

**Differential Cut**

\[ y' = g(x, y) \]

**Differential Ghost**

\( x = f(t) \)

---

**Logic**

Provability theory

**Math**

Characteristic PDE

---

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17

---

André Platzer (CMU)
Logic & Proofs for Cyber-Physical Systems with KeYmaera X
iFM'17 12 / 29
Differential Invariants for Differential Equations

Differential Invariant

\[ DI \geq, \land, \lor \]

Differential Cut

\[ DI \geq, =, \land, \lor \]

Differential Ghost

\[ x' = f(x) \]

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM'17 12 / 29
Differential Invariants for Differential Equations

Differential Invariant

\[ \frac{dx}{dt} = f(x) \]

Differential Cut

\[ \frac{dy}{dt} = g(x, y) \]

Differential Ghost

\[ \text{inv} \]

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17
Differential Invariants for Differential Equations

\[ x' = f(x) \]

**Differential Invariant**

**Differential Cut**

**Differential Ghost**

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17
Differential Invariants for Differential Equations

Differential Invariant

\[ \frac{dy}{dx} = g(x, y) \]

Differential Cut

\[ \frac{dx}{dt} = f(x) \]

Differential Ghost

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17

André Platzer (CMU)
Differential Invariants for Differential Equations

Differential Invariant

\[ \frac{dx}{dt} = f(x) \]

Differential Cut

\[ \frac{dy}{dt} = g(x, y) \]

Differential Ghost

\[ x' = f(x) \]

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X iFM’17 12 / 29
Differential Invariants for Differential Equations

**Differential Invariant**

\[ x' = f(x) \]

**Differential Cut**

\[ y' = g(x, y) \]

**Differential Ghost**

\[ y' = g(x, y) \]

Logic
- Provability theory

Math
- Characteristic PDE

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, ITP’12, JAR’17
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

\begin{align*}
  &x' = f(x) \\
y' = g(x, y)
\end{align*}

\begin{align*}
  \text{Logic} &\quad \text{Math} \\
  \text{Provability theory} &\quad \text{Characteristic PDE}
\end{align*}

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, ITP'12, JAR'17

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X
Differential Invariant

\[ Q \vdash [x' := f(x)](P)' \]
\[ P \vdash [x' = f(x) & Q]P \]

Differential Cut

\[ P \vdash [x' = f(x) & Q]C \quad P \vdash [x' = f(x) & Q \land C]P \]
\[ P \vdash [x' = f(x) & Q]P \]

Differential Ghost

\[ P \leftrightarrow \exists y G \quad G \vdash [x' = f(x), y' = g(x, y) & Q]G \]
\[ P \vdash [x' = f(x) & Q]P \]
**Differential Invariant**

\[
Q \vdash [x' := f(x)](P)'
\]

\[
P \vdash [x' = f(x) \& Q]P
\]

**Differential Cut**

\[
P \vdash [x' = f(x) \& Q]C
\]

\[
P \vdash [x' = f(x) \& Q \land C]P
\]

\[
P \vdash [x' = f(x) \& Q]P
\]

**Differential Ghost**

\[
P \leftrightarrow \exists y \ G
\]

\[
G \vdash [x' = f(x), y' = g(x, y) \& Q]G
\]

\[
P \vdash [x' = f(x) \& Q]P
\]

if new \( y' = g(x, y) \) has a global solution
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y & \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]
\[ \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \& \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

damped oscillator
\( \omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0 \)

\( \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] \ 2\omega^2 xx' + 2yy' \leq 0 \)

\( \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \ & \ \omega \geq 0 \land d \geq 0] \ \omega^2 x^2 + y^2 \leq c^2 \)

Damped oscillator
Differential Invariants for Differential Equations

\[
\omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0
\]

\[
\omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

damped oscillator
Differential Invariants for Differential Equations

\[ \omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0 \]

\[ \omega \geq 0 \land d \geq 0 \vdash \begin{aligned} x' &= y \quad \text{[x':=y]} \\ y' &= -\omega^2 x - 2d\omega y \quad \text{[y':=-\omega^2 x - 2d\omega y]} \end{aligned} \]

\[ 2\omega^2 xx' + 2yy' \leq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash \begin{aligned} x' &= y \quad \text{[x'=y]} \\ y' &= -\omega^2 x - 2d\omega y \quad \text{[y'=-\omega^2 x - 2d\omega y]} \end{aligned} \]

\[ \omega \geq 0 \land d \geq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \]

Diagram of a damped oscillator in 2D space, indicating the need in the domain.
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \& \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \& \omega \geq 0 \& d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

increasingly damped oscillator
\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 & \omega \geq 0] \ d \geq 0
\]

increasingly damped oscillator
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \land \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega \geq 0 \vdash [d' := 7] d' \geq 0 \]

\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \land \omega \geq 0] d \geq 0 \]

increasingly damped oscillator
\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 \& \omega \geq 0 \& d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 \& \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\omega \geq 0 \vdash 7 \geq 0
\]

\[
\omega \geq 0 \vdash [d' := 7] \ d' \geq 0
\]

\[
d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 \& \omega \geq 0] \ d \geq 0
\]

increasingly damped oscillator
\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 & \omega \geq 0 \wedge d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 & \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\star
\]

\[
\omega \geq 0 \vdash 7 \geq 0
\]

\[
\omega \geq 0 \vdash [d' := 7] d' \geq 0
\]

\[
d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 & \omega \geq 0] d \geq 0
\]

increasingly damped oscillator
\[ \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d \omega y] 2\omega^2 xx' + 2yy' \leq 0 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \& \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]
\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \& \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega \geq 0 \vdash 7 \geq 0 \]
\[ \omega \geq 0 \vdash [d' := 7] d' \geq 0 \]
\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d \omega y, d' = 7 \& \omega \geq 0] d \geq 0 \]

increasingly damped oscillator
Differential Cuts for Differential Equations

\( \omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0 \)

\( \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0 \)

\( \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 \& \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \)

\( \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 \& \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \)

\[ \begin{align*}
\omega \geq 0 & \vdash 7 \geq 0 \\
\omega \geq 0 & \vdash [d' := 7] d' \geq 0 \\
d' \geq 0 & \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, \ d' = 7 \& \omega \geq 0] d \geq 0 
\end{align*} \]

increasingly damped oscillator
\[
\omega \geq 0 \land d \geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0
\]

\[
\omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2d\omega y] 2\omega^2 xx' + 2yy' \leq 0
\]

\[
\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2
\]

\[
\omega \geq 0 \vdash 7 \geq 0
\]

\[
\omega \geq 0 \vdash [d' := 7] d' \geq 0
\]

\[
d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d' = 7 \land \omega \geq 0] d \geq 0
\]

increasingly damped oscillator
Differential Cuts for Differential Equations

\[ \omega \geq 0 \land d \geq 0 \vdash 2 \omega^2 x y + 2 y (-\omega^2 x - 2 d \omega y) \leq 0 \]

\[ \omega \geq 0 \land d \geq 0 \vdash [x' := y][y' := -\omega^2 x - 2 d \omega y] 2 \omega^2 x x' + 2 y y' \leq 0 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2 d \omega y, d' = 7 \& \omega \geq 0 \land d \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2 d \omega y, d' = 7 \& \omega \geq 0] \omega^2 x^2 + y^2 \leq c^2 \]

\[ \omega \geq 0 \vdash 7 \geq 0 \]

\[ \omega \geq 0 \vdash [d' := 7] d' \geq 0 \]

\[ d \geq 0 \vdash [x' = y, y' = -\omega^2 x - 2 d \omega y, d' = 7 \& \omega \geq 0] d \geq 0 \]

Could repeatedly diffcut in formulas to help the proof
Application Highlights

Obstacle Avoidance + Ground Navigation

Train Control Brakes

Airborne Collision Avoidance (ACAS X)

Ship Cooling

\[ x' = f(x) \]

\[ y' = g(x, y) \]

André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM’17 16 / 29
Ground Robot Obstacle Avoidance: Verify

- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle

1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X
Ground Robot Obstacle Avoidance: Verify

- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle

1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

André Platzer (CMU)  Logic & Proofs for Cyber-Physical Systems with KeYmaera X  iFM’17 17 / 29
Ground Robot Obstacle Avoidance: Verify

- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle

1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X
Fundamental safety question for ground robot navigation
When will which control decision avoid obstacles?
Depends on safety objective, physical capabilities of robot + obstacle

1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X
Ground Robot Obstacle Avoidance: Verify

- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle

1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X iFM’17 17 / 29
<table>
<thead>
<tr>
<th>Safety</th>
<th>Invariant</th>
<th>+ Safe Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon s\right) )</td>
<td></td>
</tr>
<tr>
<td>passive</td>
<td>( s \neq 0 \rightarrow |p - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon(s + V)\right) )</td>
<td></td>
</tr>
<tr>
<td>+ sensor</td>
<td>( |\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon(s + V)\right) + \Delta_p )</td>
<td></td>
</tr>
<tr>
<td>+ disturb</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} \Delta_a + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon(s + V)\right) )</td>
<td></td>
</tr>
<tr>
<td>+ failure</td>
<td>( |\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon(v + V)\right) + \Delta_p + g\Delta )</td>
<td></td>
</tr>
<tr>
<td>friendly</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b} + \frac{V^2}{2b_o} + V \left(\frac{s}{b} + \tau\right) + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon(s + V)\right) )</td>
<td></td>
</tr>
<tr>
<td>Safety</td>
<td>Invariant</td>
<td>+ Safe Control</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>static</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b} ) + ( \left( \frac{A}{b} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon s \right) )</td>
<td></td>
</tr>
<tr>
<td>passive</td>
<td>( s \neq 0 \rightarrow |p - o|_\infty &gt; \frac{s^2}{2b} ) + ( V \frac{s}{b} + \left( \frac{A}{b} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon(s + V) \right) )</td>
<td></td>
</tr>
<tr>
<td>+ sensor</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b} ) + ( V \frac{s}{b} + \left( \frac{A}{b} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon(s + V) \right) ) + ( \Delta p )</td>
<td></td>
</tr>
<tr>
<td>+ disturb.</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b \Delta_a} + V \frac{s}{b \Delta_a} ) + ( \left( \frac{A}{b \Delta_a} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon(s + V) \right) )</td>
<td></td>
</tr>
<tr>
<td>+ failure</td>
<td>( |\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} ) + ( \left( \frac{A}{b} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon(v + V) \right) + \Delta_p + g \Delta )</td>
<td></td>
</tr>
<tr>
<td>friendly</td>
<td>( |p - o|_\infty &gt; \frac{s^2}{2b} + \frac{V^2}{2b_o} + V \left( \frac{s}{b} + \tau \right) ) + ( \left( \frac{A}{b} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon(s + V) \right) )</td>
<td></td>
</tr>
</tbody>
</table>
Airborne Collision Avoidance System ACAS X: Verify

- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions

1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

TACAS'15, EMSOFT'15, STTT'17
ACAS X table comparison shows safe advisory in 97.7% of the 648,591,384,375 states compared (15,160,434,734 counterexamples).

ACAS X issues DNC advisory, which induces collision unless corrected.
Airborne Collision Avoidance System ACAS X: Refine

- Conservative, so too many counterexamples
- Settle for: safe for a little while, with safe future advisory possibility
- Safeable advisory: a subsequent advisory can safely avoid collision

1. Identified safeable region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared ($\approx 899 \times 10^6$ counterexamples).

**Counterexample: Action Issued = Maintain**
Followed by Most Extreme Up/Down-sense Advisory Available

ACAS X issues Maintain advisory instead of CL1500
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (≈899 $10^6$ counterexamples).

ACAS X issues Maintain advisory instead of CL1500
Verified CPS Applications: Trains & Airplanes

ICFEM’09, JAIS’14, TACAS’15, EMSOFT’15, FM’09, HSCC’11, HSCC’13, TACAS’14, RSSRail’17
Verified CPS Applications: Cars

FM’11, LMCS’12, ICCPS’12, ITSC’11, ITSC’13, IJCAR’12
Verified CPS Applications: Robots

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X

HSCC’13, RSS’13, CADE’12, IJRR’17
KeYmaera X aXiomatic Tactical Theorem Prover for CPS

Proof search

KeYmaera X

generates proofs

ModelPlex proof

Model
Safety
Compliance
Monitor

Trustworthy
Uniform substitution
Sound & complete
Small core: 1700 LOC

Flexible
Proof automation
Interactive UI
Programmable

Customizable
Scala+Java API
Command line
REST API

ctrl: \( a := -b; \)
plant: \( x'' = a \)

André Platzer (CMU)
Logic & Proofs for Cyber-Physical Systems with KeYmaera X
iFM’17 24 / 29
Disclaimer: Self-reported estimates of the soundness-critical lines of code + rules
Acknowledgments

Students and postdocs of the Logical Systems Lab at Carnegie Mellon Brandon Bohrer, Nathan Fulton, Sarah Loos, João Martins, Yong Kiam Tan Khalil Ghorbal, Jean-Baptiste Jeannin, Stefan Mitsch
Logical foundations make a big difference for CPS, and vice versa

\[ \mathcal{dL} = DL + HP \]

- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

1. Multi-dynamical systems
2. Combine simple dynamics
3. Tame complexity
4. www.keymaeraX.org

Numerous wonders remain to be discovered
Logical foundations make a big difference for CPS, and vice versa

**differential dynamic logic**

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

- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

**KeYmaera X**

Numerous wonders remain to be discovered
Future CPS Challenges

Numerous wonders remain to be discovered

- Scalable continuous stochastics
- Concurrent CPS
- Real arithmetic: Scalable and verified
- Verified CPS implementations, ModelPlex
- Correct CPS execution
- CPS-conducive tactic languages+libraries
- Tactics exploiting CPS structure/linearity/…
- Invariant generation
- Tactics & proofs for reachable set computations
- Parallel proof search & disprovers
- Correct model transformation
- Inspiring applications

CPSs deserve proofs as safety evidence!

André Platzer (CMU) Logic & Proofs for Cyber-Physical Systems with KeYmaera X iFM’17 28 / 29
Logical Foundations of Cyber-Physical Systems

- Logic
- Theorem Proving
- Proof Theory
- Model Checking
- Computer Algebra
- Algebraic Geometry
- Differential Algebra
- Lie Algebra
- Analysis
- Differential Equations
- Viscosity PDE Solutions
- Carathéodory Solutions
- Dynamical Systems
- Stochastic Solutions
- Doob’s Supermartingales
- Dynkin’s Infinitesimal Generators
- Numerical Integration
- Error Analysis
- Hermite Interpolation
- Stochastic Differential Equations
- Differential Generators
- Stochastic Processes
- Numerical Algorithms
- Decision Procedures
- Proof Search Procedures
- Fixpoints & Lattices
- Closure Ordinals
- Algorithms
- Model Checking
- Logical Foundations of Cyber-Physical Systems

André Platzer (CMU)

Logic & Proofs for Cyber-Physical Systems with KeYmaera X

iFM'17 29 / 29
Differential Dynamic Logic $\mathcal{DL}$: Semantics

**Definition (Hybrid program semantics)**

$\llbracket \cdot \rrbracket : \text{HP} \to \wp(\mathcal{S} \times \mathcal{S})$

\begin{align*}
\llbracket x := e \rrbracket & = \{ (\omega, \nu) : \nu = \omega \text{ except } \llbracket x \rrbracket \nu = \llbracket e \rrbracket \omega \} \\
\llbracket ?Q \rrbracket & = \{ (\omega, \omega) : \omega \in \llbracket Q \rrbracket \} \\
\llbracket x' = f(x) \rrbracket & = \{ (\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for some duration } r \} \\
\llbracket \alpha \cup \beta \rrbracket & = \llbracket \alpha \rrbracket \cup \llbracket \beta \rrbracket \\
\llbracket \alpha ; \beta \rrbracket & = \llbracket \alpha \rrbracket \circ \llbracket \beta \rrbracket \\
\llbracket \alpha^* \rrbracket & = \bigcup_{n \in \mathbb{N}} \llbracket \alpha^n \rrbracket 
\end{align*}

**Definition (dL semantics)**

$\llbracket \cdot \rrbracket : \text{Fml} \to \wp(\mathcal{S})$

\begin{align*}
\llbracket e \geq \tilde{e} \rrbracket & = \{ \omega : \llbracket e \rrbracket \omega \geq \llbracket \tilde{e} \rrbracket \omega \} \\
\llbracket \neg P \rrbracket & = \llbracket P \rrbracket^C \\
\llbracket P \land Q \rrbracket & = \llbracket P \rrbracket \cap \llbracket Q \rrbracket \\
\llbracket \langle \alpha \rangle P \rrbracket & = \llbracket \alpha \rrbracket \circ \llbracket P \rrbracket = \{ \omega : \nu \in \llbracket P \rrbracket \text{ for some } \nu : (\omega, \nu) \in \llbracket \alpha \rrbracket \} \\
\llbracket [\alpha] P \rrbracket & = \llbracket \neg \langle \alpha \rangle \neg P \rrbracket = \{ \omega : \nu \in \llbracket P \rrbracket \text{ for all } \nu : (\omega, \nu) \in \llbracket \alpha \rrbracket \} \\
\llbracket \exists x P \rrbracket & = \{ \omega : \omega^r_x \in \llbracket P \rrbracket \text{ for some } r \in \mathbb{R} \} 
\end{align*}
André Platzer.
Logic & proofs for cyber-physical systems.
doi:10.1007/978-3-319-40229-1_3.

André Platzer.
Logics of dynamical systems.
In LICS [34], pages 13–24.

André Platzer.
Differential dynamic logic for hybrid systems.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.

André Platzer.
Differential game logic.

André Platzer.
Differential hybrid games.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [34], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
André Platzer.


André Platzer.

A uniform substitution calculus for differential dynamic logic. In Felty and Middeldorp [35], pages 467–481. doi:10.1007/978-3-319-21401-6_32.

André Platzer.


André Platzer and Edmund M. Clarke.

Computing differential invariants of hybrid systems as fixedpoints.
André Platzer.

André Platzer.


André Platzer and Jan-David Quesel.


Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer.
A formally verified hybrid system for the next-generation airborne collision avoidance system.
doi:10.1007/978-3-662-46681-0_2.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer.
Formal verification of ACAS X, an industrial airborne collision avoidance system.
doi:10.1109/EMSOFT.2015.7318268.

Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völß, and André Platzer.
KeYmaera X: An axiomatic tactical theorem prover for hybrid systems.
In Felty and Middeldorp [35], pages 527–538.
doi:10.1007/978-3-319-21401-6_36.
Stefan Mitsch and André Platzer.  
ModelPlex: Verified runtime validation of verified cyber-physical system models.  
Special issue of selected papers from RV’14.  

André Platzer, Jan-David Quesel, and Philipp Rümmer.  
Real world verification.  
doi:10.1007/978-3-642-02959-2_35.

Nathan Fulton, Stefan Mitsch, Brandon Bohrer, and André Platzer.  
Bellerophon: Tactical theorem proving for hybrid systems.  
doi:10.1007/978-3-319-66107-0_14.

André Platzer.
Logical Foundations of Cyber-Physical Systems.
URL: http://www.springer.com/978-3-319-63587-3.

André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
doi:10.1007/978-3-642-14509-4.

Thomas A. Henzinger.
The theory of hybrid automata.
doi:10.1109/LICS.1996.561342.

Jennifer M. Davoren and Anil Nerode.
Logics for hybrid systems.

Ashish Tiwari.
Abstractions for hybrid systems.


