André Platzer

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

http://symbolaris.com/
Outline

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games

2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics

3. Proofs for CPS

4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Examples
   - Differential Radical Invariants

5. Applications

6. Summary
Outline

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games

2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics

3. Proofs for CPS

4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Examples
   - Differential Radical Invariants

5. Applications

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

Safety guarantees require analytic foundations.

Foundations revolutionized digital computer science & our society.

Need even stronger foundations when software reaches out into our physical world.

How can we provide people with cyber-physical systems they can bet their lives on? — Jeannette Wing

Cyber-physical Systems

CPS combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone.

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

Rationale

1. Safety guarantees require analytic foundations.
2. Foundations revolutionized digital computer science & our society.
3. Need even stronger foundations when software reaches out into our physical world.

How can we provide people with cyber-physical systems they can bet their lives on?

— Jeannette Wing

Cyber-physical Systems

CPS 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 help people

Prerequisite: CPS need to be safe

How do we make sure CPS make the world a better place?
### Benefits of Logical Foundations for CPS V & V

#### Proofs

<table>
<thead>
<tr>
<th>Safety</th>
<th>Formalize system properties: What is “Safe”? “Reach goal”?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>Formalize system models, clarify behavior</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Make assumptions explicit rather than silently</td>
</tr>
<tr>
<td>Constraints</td>
<td>Reveal invariants, switching conditions, operating conditions</td>
</tr>
<tr>
<td>Design</td>
<td>Invariants guide safe controller design</td>
</tr>
<tr>
<td>Constructive</td>
<td>Construct system models along with their proofs</td>
</tr>
</tbody>
</table>

#### Byproducts

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Determine design trade-offs &amp; feasibility early</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesis</td>
<td>Turn high-level models into code &amp; correctness monitors</td>
</tr>
<tr>
<td>Certificate</td>
<td>Proofs as artifacts for certification</td>
</tr>
</tbody>
</table>

#### Tools

| KeYmaera X    | aXiomatic Tactical Theorem Prover for CPS                 |

---

*André Platzer (CMU)*

*Logical Foundations & Cyber-Physical Systems*
CPSs are Multi-Dynamical Systems

CPS Dynamics
CPS are characterized by multiple facets of dynamical systems.

CPS Compositions
CPS combine multiple simple dynamical effects.

Tame Parts
Exploiting compositionality tames CPS complexity.
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

![Graphs showing state evolution over time with discrete and continuous dynamics]
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)
Challenge (Hybrid Systems)

Fixed rule describing state evolution with both
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
CPS Analysis: Other Agents

Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ♦ vs. Demon □)
CPS Analysis: Other Agents

Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ♦ vs. Demon □)

André Platzer (CMU)
Logical Foundations & Cyber-Physical Systems
CPSs are Multi-Dynamical Systems

hybrid systems

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

hybrid games

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

stochastic hybrid sys.

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

distributed hybrid sys.

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

- **Differential Dynamic Logic**
  \[ \mathcal{dL} = \mathcal{DL} + \mathcal{HP} \]

- **Differential Game Logic**
  \[ \mathcal{dGL} = \mathcal{GL} + \mathcal{HG} \]

- **Stochastic Differential Logic**
  \[ \mathcal{SdL} = \mathcal{DL} + \mathcal{SHP} \]

- **Quantified Differential Logic**
  \[ \mathcal{QdL} = \mathcal{FOL} + \mathcal{DL} + \mathcal{QHP} \]

JAR'08, CADE'11, LMCS'12, LICS'12

LICS'12, CADE'15, TOCL'15
Benefits

Logical foundations of cyber-physical systems

1. Multi-dynamical systems
2. Tame complexity by combinations of simple dynamics
3. Compositional programming language for CPS
4. Compositional logics and proof calculi
5. Differentiate logic & Logicalize differentials
6. Proofs for differential equations
7. Solid foundation for theory
8. Many useful applications
9. Education: Foundations of CPS course

Basis for other technology

1. ModelPlex transfers CPS model \( \leadsto \) implementation safety FMSD’16
2. Proof-aware refactoring to co-evolve model + proof FM’14
3. Control envelope design ACC’12
Verified CPS Applications

FM'11, LMCS'12, ICCPS'12, ITSC'11, ITSC'13, IJCAR'12
Verified CPS Applications
Verified CPS Applications

15-424/624 Foundations of Cyber-Physical Systems students

André Platzer (CMU)
Dynamic Logics for Dynamical Systems

**differential dynamic logic**
\[ d\mathcal{L} = DL + HP \]

**differential game logic**
\[ dG\mathcal{L} = GL + HG \]

**stochastic differential DL**
\[ Sd\mathcal{L} = DL + SHP \]

**quantified differential DL**
\[ Qd\mathcal{L} = FOL + DL + QHP \]
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \phi \quad \alpha \quad \phi \]

\[
\begin{align*}
\text{init} &\rightarrow [\text{if}(SB(x, m)) a := -b] ; x' = v, v' = a)
\end{align*}
\]

\[
\begin{align*}
\text{init} &\rightarrow [\text{if}(SB(x, m)) a := -b] ; x' = v, v' = a)
\end{align*}
\]
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha]\phi \quad \phi \]

\[ x \neq m \]

\[ \begin{align*}
\mathbf{a} & \quad \mathbf{v} & \quad \mathbf{x} \\
0 & \quad -2 & \quad 10 \\
0.5 & \quad -2 & \quad 8 \\
1 & \quad 0 & \quad 6 \\
2 & \quad 2 & \quad 4 \\
3 & \quad 4 & \quad 2 \\
4 & \quad 6 & \quad 0 \\
5 & \quad 8 & \quad -2 \\
6 & \quad 10 & \quad -2 \\
7 & \quad 12 & \quad -2
\end{align*} \]
CPS Analysis

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

\[ [\alpha] \phi \]

\[ \square x \neq m \]

\[ x \neq m \]

\[ x \neq m \]
CPS Analysis

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

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

\[ \phi \] \( x \neq m \)

\[ [\alpha] \phi \] \( x \neq m \)

\[ \alpha \] \( \phi \)

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

\[ t \]

\[ x \]

\[ m \]

ODE

André Platzer (CMU)
CPS Analysis

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

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

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

\[ x \neq m \]

\[ [x \neq m] \]

\[ a := -b \]

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

ODE

assign

\[ a \]

\[ x \]

\[ m \]
Concept (Differential Dynamic Logic) (JAR’08, LICS’12)

\[ [\alpha] \phi \quad \phi \]

\[ x \neq m \]

\[ [\bullet] x \neq m \]

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

\( x' = v, v' = a \)

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

\[
\begin{align*}
\alpha & \quad \phi \\
[\alpha] \phi & \quad \phi
\end{align*}
\]

(seq. compose)

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

André Platzer (CMU)
Logical Foundations & Cyber-Physical Systems
CPS Analysis

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

\[
[\alpha] \phi \\
\downarrow \\
\phi
\]

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

**Test**, **Assign**, **ODE**

seq. compose

nondet. repeat

Andrè Platzer (CMU)
CPS Analysis

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

\[ [\alpha] \phi \]

\[ \alpha \]

\[ \phi \]

\[ [x \neq m] \]

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

\[ x \neq m \]

\[ x \neq m \]

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

\[ \text{all runs} \]

\[ a \]

\[ v \]

\[ m \]
CPS Analysis

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

\[
\begin{align*}
&\text{init: } x \neq m \land b > 0 \\
&\Rightarrow [((\text{if}(SB(x, m)) \ a := -b) ; \ x' = v, v' = a)^*] x \neq m
\end{align*}
\]

André Platzer (CMU)
CPS Analysis

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

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

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

André Platzer (CMU) Logical Foundations & Cyber-Physical Systems
Concept (Differential Dynamic Logic)

$\forall \alpha \phi \rightarrow [\alpha] \phi$

$x \neq m \land b > 0 \rightarrow [((\neg SB(x, m) \cup a := -b) ; x' = v, v' = a)^{*}]x \neq m$

André Platzer (CMU)
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 $P$)**

$$e \geq \bar{e} \mid \neg P \mid P \& Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P$$
Differential Dynamic Logic \( dL \): 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 (dL Formula \( P \))**

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

- **Discrete Assign**
- **Test Condition**
- **Differential Equation**
- **Nondet. Choice**
- **Seq. Compose**
- **Nondet. Repeat**

- **All Reals**
- **Some Reals**
- **All Runs**
- **Some Runs**
Definition (dL semantics) ($[[\cdot]] : \text{Fml} \rightarrow \wp(S)$)

$[[e \geq e']] = \{ \omega : [e] \omega \geq [e'] \omega \}$

$[[\neg P]] = (\neg P)$

$[[P \land Q]] = [P] \cap [Q]

$[[P]] = \{ \omega : \nu \models P \text{ for all } \nu : (\omega, \nu) \in [\alpha] \}$

$[[\exists x P]] = \{ \omega : \omega' \models P \text{ for some } x' \in [P] \}$

$[[\alpha ; \beta]] = [\alpha] \circ [\beta]

[[\alpha \cup \beta]] = \bigcup \{ [\alpha], [\beta] \}$

$[[\alpha^*]] = \{ (\omega, \nu) : \nu \models P \text{ for some } \nu : (\omega, \nu) \in [\alpha] \}$

$[[\alpha \cup \beta]] = \bigcup \{ [\alpha], [\beta] \}$

$[[\alpha \cup \beta]] = \bigcup \{ [\alpha], [\beta] \}$

Definition (Hybrid program semantics) ($[[\cdot]] : \text{HP} \rightarrow \wp(S \times S)$)

$[[x := f(x)]] = \{ (\omega, \nu) : \nu = \omega \text{ except } [x] \nu = [f(x)] \omega \}$

$[[x' = f(x)]] = \{ (\omega, \nu) : \nu = \omega \text{ for some duration } r \}$

$[[x' = f(x)]] = \{ (\omega, \nu) : \nu = \omega \text{ for some duration } r \}$

$[[\exists x P]] = \{ \omega : \omega' \models P \text{ for some } \omega' \in [P] \}$

$[[\alpha \cup \beta]] = \bigcup \{ [\alpha], [\beta] \}$
Outline

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games

2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics

3. Proofs for CPS

4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Examples
   - Differential Radical Invariants

5. Applications

6. Summary
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) \] \( P \quad (y'(t) = f(y)) \)

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

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

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

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

\[ I \] \( [\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P) \)

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

LICS'12, CADE'15
Differential Dynamic Logic: Axiomatization

\[ \begin{align*}
\text{[:=]} & \quad [x := e]P(x) \iff P(e) \\
\text{[?] } & \quad [? Q]P \iff (Q \to P) \\
\text{[\text{'}]} & \quad [x' = f(x)]P \iff \forall t \geq 0 [x := y(t)]P \quad (y'(t) = f(y)) \\
\text{[\cup]} & \quad [\alpha \cup \beta]P \iff [\alpha]P \land [\beta]P \\
\text{[;]} & \quad [\alpha; \beta]P \iff [\alpha][\beta]P \\
\text{[*]} & \quad [\alpha^*]P \iff P \land [\alpha][\alpha^*]P \\
\text{K} & \quad [\alpha](P \to Q) \to ([\alpha]P \to [\alpha]Q) \\
\text{I} & \quad [\alpha^*](P \to [\alpha]P) \to (P \to [\alpha^*]P) \\
\text{C} & \quad [\alpha^*]\forall v > 0 (P(v) \to \langle \alpha \rangle P(v-1)) \to \forall v (P(v) \to \langle \alpha^* \rangle \exists v \leq 0 P(v))
\end{align*} \]
Differential Dynamic Logic: Axiomatization

\[ \frac{P}{G[\alpha]P} \]
\[ \frac{P}{\forall x P} \]
\[ \frac{P \rightarrow Q \quad P}{Q} \]

rules of truth

LICS'12, CADE'15
Differential Dynamic Logic: Axiomatization

\[
\begin{align*}
G & \quad \frac{P}{[\alpha]P} \\
& \quad \forall P \\
& \quad \forall x P \\
MP & \quad \frac{P \to Q \quad P}{Q} \\
V & \quad p \to [\alpha]p \quad (FV(p) \cap BV(\alpha) = \emptyset) \\
CT & \quad \frac{f(x) = g(x)}{c(f(x)) = c(g(x))} \\
CQ & \quad \frac{f(x) = g(x)}{p(f(x)) \leftrightarrow p(g(x))} \\
CE & \quad \frac{P \leftrightarrow Q}{C(P) \leftrightarrow C(Q)}
\end{align*}
\]
\[ [x := e] P \leftrightarrow P(e) \]
$$[x := e]P \leftrightarrow P(e)$$

$$[x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y_x(t)]P$$
Proofs for Hybrid Systems

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

\[ [x' = f(x)]P \leftrightarrow \forall t \geq 0 [x := y_x(t)]P \]
compositional semantics $\Rightarrow$ compositional rules!
$$\left[ \alpha \cup \beta \right] P \iff \left[ \alpha \right] P \land \left[ \beta \right] P$$
Proofs for Hybrid Systems

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

$$([\alpha; \beta]P \leftrightarrow [\alpha][\beta]P)$$
Proofs for Hybrid Systems

\[ [\alpha \cup \beta]P \iff [\alpha]P \land [\beta]P \]

\[ [\alpha; \beta]P \iff [\alpha][\beta]P \]

\[ P \quad P \rightarrow [\alpha]\]

\[ [\alpha^*]P \]

\[ P \rightarrow [\alpha]P \]

\[ P \rightarrow P \]

\[ \alpha \]

\[ \alpha \]

\[ \alpha \]

\[ \omega \]

\[ \nu \]

\[ \nu_1 \]

\[ \nu_2 \]

\[ \omega \]

\[ s \]

\[ \nu \]

\[ \alpha \]

\[ \alpha \]

\[ \alpha \]

\[ \beta \]

\[ \beta \]

\[ \alpha^* \]
Outline

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games

2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics

3. Proofs for CPS

4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Examples
   - Differential Radical Invariants

5. Applications

6. Summary
Theorem (Sound & Complete) (J.Autom.Reas. 2008, LICS’12)

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

Corollary (Complete Proof-theoretical Alignment & Bridging)

proving continuous = proving hybrid = proving discrete
Theorem (Sound & Complete) (J.Autom.Reas. 2008, LICS’12)

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

Corollary (Complete Proof-theoretical Alignment & Bridging)

proving continuous = proving hybrid = proving discrete
**Theorem (Sound & Complete)** (J.Autom.Reas. 2008, LICS’12)

\[ d\mathcal{L} \text{ calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations or discrete dynamics. } \]

**Corollary (Complete Proof-theoretical Alignment & Bridging)**

proving continuous = proving hybrid = proving discrete
Differential Equation Axioms & Differential Axioms

**DW** \( [x' = f(x) & Q]Q \)

**DC** \( ([x' = f(x) & Q]P \leftrightarrow [x' = f(x) & Q \land r(x)]P) \)
\( \leftarrow [x' = f(x) & Q]r(x) \)

**DE** \( [x' = f(x) & Q]P \leftrightarrow [x' = f(x) & Q][x' := f(x)]P \)

**DI** \( [x' = f(x) & Q]P \leftrightarrow (Q \rightarrow P \land [x' = f(x) & Q](P')) \)

**DG** \( [x' = f(x) & Q]P \leftrightarrow \exists y \, [x' = f(x), \, y' = a(x)y + b(x) & Q]P \)

**DS** \( [x' = c() & Q]P \leftrightarrow \forall t \geq 0 \, ((\forall 0 \leq s \leq t \, q(x + c())s)) \rightarrow [x := x + c()t]P \)

\( [':=] [x' := e]p(x') \leftrightarrow p(e) \)

\( +' (e + k)' = (e)' + (k)' \)

\( \cdot' (e \cdot k)' = (e)' \cdot k + e \cdot (k)' \)

\( o' [y := g(x)][y' := 1]((f(g(x)))') = (f(y))' \cdot (g(x))' \)
Differential Invariants for Differential Equations

\[ x' = f(x) \]

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

Logic

Provability theory

Math

Characteristic PDE

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, CADE’15

André Platzer (CMU)
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

Logic
Provability theory

Math
Characteristic PDE

$\frac{dx}{dt} = f(x)$

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

$\text{DI} \geq \text{DI}$,
$\text{DI} > \text{DI}$,
$\text{DI} \leq \text{DI}$,
$\text{DI} = \text{DI}$,
$\text{DI} \land \text{DI}$,
$\text{DI} \lor \text{DI}$

JLogComput’10, CAV’08, FMSD’09, LMCS’12, LICS’12, CADE’15
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, CADE'15
Differential Invariants for Differential Equations

Differential Invariant

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

Differential Cut

Differential Ghost

Logic

Provability

Math

Characteristic PDE

JLogComput ’10, CAV ’08, FMSD ’09, LMCS ’12, LICS ’12, CADE ’15

André Platzer (CMU)
Logical Foundations & Cyber-Physical Systems

23 / 42
Differential Invariants for Differential Equations

\[ x' = f(x) \]

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

Logic
- Provability theory

Math
- Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15
Differential Invariants for Differential Equations

Differential Invariant

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

Differential Cut

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

Differential Ghost

\[ 0 \]

Logic

Provability theory

Math

Characteristic PDE

\[ JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15 \]
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

DI ≤,∧,∨ → DI ≥,∧,∨ → DI ≥,=,∧,∨ → DI

Logic
Provability theory

Math
Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS’12, LICS’12, CADE’15

André Platzer (CMU)
Logical Foundations & Cyber-Physical Systems
Differential Invariants for Differential Equations

**Differential Invariant**

- \( \frac{dx}{dt} = f(x) \)
- \( \frac{dy}{dt} = g(x, y) \)

**Differential Cut**

- \( \exists t \forall x \) \( x(t) \)

**Differential Ghost**

- \( x' = f(x) \)

---

**Logic**

- Provability theory

**Math**

- Characteristic PDE

---

Andrè Platzer (CMU)
Logical Foundations & Cyber-Physical Systems
23 / 42
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, CADE’15

André Platzer (CMU)
Differential Invariants for Differential Equations

Differential Invariant

Differential Cut

Differential Ghost

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

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

\[ \text{inv} \]

Logic

Provability theory

Math

Characteristic PDE

JLogComput'10, CAV'08, FMSD'09, LMCS'12, LICS'12, CADE'15
Differential Equation Axioms

Axiom (Differential Weakening) (CADE’15)

\[
\text{DW} \quad [x' = f(x) \& Q]Q
\]

Differential equations cannot leave their evolution domains. Implies:

\[
[x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q](Q \rightarrow P)
\]
Axiom (Differential Cut) (CADE’15)

DC

\[
\begin{align*}
([x' = f(x) \& Q]P & \leftrightarrow [x' = f(x) \& Q \land r(x)]P) \\
& \leftrightarrow [x' = f(x) \& Q]r(x)
\end{align*}
\]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Axiom (Differential Cut) (CADE’15)

\[
DC \quad ([x' = f(x) & Q]P \leftrightarrow [x' = f(x) & Q \land r(x)]P) \\
\quad \leftrightarrow [x' = f(x) & Q]r(x)
\]

DC is a cut for differential equations.

DC is a differential modal modus ponens K.

Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Axiom (Differential Cut) (CADE’15)

\[ DC \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \land r(x)]P) \leftarrow [x' = f(x) \& Q]r(x) \]

DC is a cut for differential equations.
DC is a differential modal modus ponens K. Can't leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Differential Equation Axioms

Axiom (Differential Cut) (CADE’15)

\[
\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \land r(x)]P) \\
\leftarrow [x' = f(x) \& Q]r(x)
\]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \(r(x)\), then might as well restrict state space to \(r(x)\).
Axiom (Differential Cut) (CADE'15)

\[ \text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \& r(x)]P) \]
\[ \leftrightarrow [x' = f(x) \& Q]r(x) \]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Axiom (Differential Cut) (CADE’15)

\[ DC \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \land r(x)]P) \]
\[ \leftrightarrow [x' = f(x) \& Q]r(x) \]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Axiom (Differential Cut) (CADE’15)

\[
\text{DC} \quad ([x' = f(x) \& Q]P \leftrightarrow [x' = f(x) \& Q \land r(x)]P) \\
\quad \leftrightarrow [x' = f(x) \& Q]r(x)
\]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Axiom (Differential Cut) (CADE’15)

DC

\[ ([x' = f(x) \land Q] P \iff [x' = f(x) \land Q \land r(x)] P) \iff [x' = f(x) \land Q] r(x) \]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Differential Equation Axioms

Axiom (Differential Cut) (CADE’15)

\[ \text{DC} \quad ([x' = f(x) \& Q] P \iff [x' = f(x) \& Q \land r(x)] P) \leftarrow [x' = f(x) \& Q] r(x) \]

DC is a cut for differential equations.
DC is a differential modal modus ponens K.
Can’t leave \( r(x) \), then might as well restrict state space to \( r(x) \).
Differential Equation Axioms

Axiom (Differential Invariant) (CADE’15)

\[ \text{DI } [x' = f(x) \& Q]P \leftrightarrow (Q \rightarrow P \& [x' = f(x) \& Q](P)') \]

Differential invariant: \( p(x) \) true now and its differential \((p(x))' \) true always

What’s the differential of a formula???

What’s the meaning of a differential term . . . in a state???
Axiom (Differential Effect) (CADE’15)

\[
\text{DE } [x' = f(x) \& Q] P \leftrightarrow [x' = f(x) \& Q][x' := f(x)] P
\]

Effect of differential equation on differential symbol \( x' \)

\([x' := f(x)]\) instantly mimics continuous effect \([x' = f(x)]\) on \( x' \)

\([x' := f(x)]\) selects vector field \( x' = f(x) \) for subsequent differentials
Differential Equation Axioms

Axiom (Differential Ghost) (CADE’15)

\[ DG \ [x' = f(x) & Q]P \leftrightarrow \exists y \ [x' = f(x), \ y' = a(x)y + b(x) & Q]P \]

Differential ghost/auxiliaries: extra differential equations that exist
Can cause new invariants
“Dark matter” counterweight to balance conserved quantities
Differential Equation Axioms

Axiom (Differential Solution) (CADE’15)

\[
\text{DS } [x' = c() & Q]P \iff \forall t \geq 0 \left( (\forall 0 \leq s \leq t q(x+c(s))) \rightarrow [x := x+c()t]P \right)
\]

Differential solutions: solve differential equations with DG, DC and inverse companions
DI proves a property of an ODE inductively by its differentials

DE exports vector field, possibly after DW exports evolution domain

CE+CQ reason efficiently in Equivalence or eQuational context

G isolates postcondition

[′:=] differential substitution uses vector field

[′·] differential computations are axiomatic (US)

\[
(f(x) \cdot g(x))' = (f(x))' \cdot g(x) + f(x) \cdot (g(x))' \\
(x \cdot x)' = (x)' \cdot x + x \cdot (x)' \\
(x \cdot x)' = x' \cdot x + x \cdot x'
\]

\[
QE \quad \therefore x^3 \cdot x + x \cdot x^3 \geq 0 \\
[':=] \quad \therefore [x' := x^3] x' \cdot x + x \cdot x' \geq 0 \\
G \quad \therefore [x' = x^3][x' := x^3] x' \cdot x + x \cdot x' \geq 0 \\
CE \quad \therefore \quad \therefore [x' = x^3][x' := x^3] (x \cdot x \geq 1)' \\
DE \quad \therefore \quad \therefore [x' = x^3] (x \cdot x \geq 1)' \\
DI \quad \quad \therefore \quad x \cdot x \geq 1 \therefore [x' = x^3] x \cdot x \geq 1
\]
Lemma (Differential lemma)

If $\varphi \models x' = f(x) \land Q$ for duration $r > 0$, then for all $0 \leq \zeta \leq r$:

\[
\frac{d\llbracket e \rrbracket \varphi(t)}{dt}(\zeta) = \llbracket (e)' \rrbracket \varphi(\zeta)
\]

Lemma (Differential assignment)

If $\varphi \models x' = f(x) \land Q$ then $\varphi \models P \leftrightarrow [x' := f(x)]P$

Lemma (Derivations)

\[
\begin{align*}
(e + k)' &= (e)' + (k)' \\
(e \cdot k)' &= (e)' \cdot k + e \cdot (k)'
\end{align*}
\]

\begin{align*}
[y := e][y' := 1][(f(e))' &= (f(y))' \cdot (e)'] \quad \text{for } y, y' \not\in e \\
(c())' &= 0 \quad \text{for arity 0 functions/numbers } c()
\end{align*}
Theorem (Differential radical invariant characterization)

\[
h = 0 \Rightarrow \bigwedge_{i=0}^{N-1} h_p^{(i)} = 0
\]

\[
h = 0 \Rightarrow [x' = p] h = 0
\]

characterizes all algebraic invariants, where \( N = \text{ord} \sqrt[\prime]{h} \), i.e.

\[
h_p^{(N)} = \sum_{i=0}^{N-1} g_i h_p^{(i)} \quad (g_i \in \mathbb{R}[x]) \quad h_p^{(i+1)} = [x' := p](h_p^{(i)})'
\]

Corollary (Algebraic Invariants Decidable)

Algebraic invariants of algebraic differential equations are decidable.
Case Study: Longitudinal Dynamics of an Airplane

Study (6th Order Longitudinal Flight Equations)

\[
\begin{align*}
   u' &= \frac{X}{m} - g \sin(\theta) - qw & \text{axial velocity} \\
   w' &= \frac{Z}{m} + g \cos(\theta) + qu & \text{vertical velocity} \\
   x' &= \cos(\theta)u + \sin(\theta)w & \text{range} \\
   z' &= -\sin(\theta)u + \cos(\theta)w & \text{altitude} \\
   \theta' &= q & \text{pitch angle} \\
   q' &= \frac{M}{l_{yy}} & \text{pitch rate}
\end{align*}
\]

\(X\) : thrust along \(u\) \quad \(Z\) : thrust along \(w\) \quad \(M\) : thrust moment for \(w\)

\(g\) : gravity \quad \(m\) : mass \quad \(l_{yy}\) : inertia second diagonal

with Khalil Ghorbal TACAS’14
Case Study: Longitudinal Dynamics of an Airplane

Result (DRI Automatically Generates Invariant Functions)

\[
\begin{align*}
\frac{M_z}{I_{yy}} + g\theta + \left(\frac{X}{m} - qw\right) \cos(\theta) + \left(\frac{Z}{m} + qu\right) \sin(\theta) \\
\frac{M_x}{I_{yy}} - \left(\frac{Z}{m} + qu\right) \cos(\theta) + \left(\frac{X}{m} - qw\right) \sin(\theta) \\
- q^2 + \frac{2M\theta}{I_{yy}}
\end{align*}
\]

with Khalil Ghorbal TACAS'14
Case Study: Dubins Dynamics of 2 Airplanes

Result (DRI Automatically Generates Invariants)

\[ \omega_1 = 0 \land \omega_2 = 0 \rightarrow v_2 \sin \vartheta x = (v_2 \cos \vartheta - v_1)y > p(v_1 + v_2) \]

\[ \omega_1 \neq 0 \lor \omega_2 \neq 0 \rightarrow -\omega_1 \omega_2(x^2 + y^2) + 2v_2\omega_1 \sin \vartheta x + 2(v_1\omega_2 - v_2\omega_1 \cos \vartheta)y \]

\[ + 2v_1 v_2 \cos \vartheta > 2v_1 v_2 + 2p(v_2|\omega_1| + v_1|\omega_2|) + p^2|\omega_1 \omega_2| \]
Outline

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games

2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics

3. Proofs for CPS

4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Examples
   - Differential Radical Invariants

5. Applications

6. Summary
Arnold Platzer (CMU)

Verified CPS Applications

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

HSCC'13, RSS'13, CADE'12
Verified CPS Applications

15-424/624 Foundations of Cyber-Physical Systems students
Surgical Robot Verification: Skull-base Surgery

Virtual fixture boundary

Redesign to predictive control

HSCC’13
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

André Platzer (CMU) Logical Foundations & Cyber-Physical Systems
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.

TACAS’15, EMSOFT’15
Airborne Collision Avoidance System ACAS X: Refine

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

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 ($\approx 31.6$ to $898.7 \times 10^6$ counterexamples).

ACAS X issues Maintain advisory instead of CLI1500
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared ($\approx 31.6$ to $898.7 \times 10^6$ counterexamples).

ACAS X issues Maintain advisory instead of CLI1500
Outline

1. CPS are Multi-Dynamical Systems
   - Hybrid Systems
   - Hybrid Games

2. Dynamic Logic for Multi-Dynamical Systems
   - Syntax
   - Semantics

3. Proofs for CPS

4. Theory of CPS
   - Soundness and Completeness
   - Differential Invariants
   - Examples
   - Differential Radical Invariants

5. Applications

6. Summary
CPSs are Multi-Dynamical Systems

Hybrid Systems:
$$\text{HS} = \text{discrete} + \text{ODE}$$

Hybrid Games:
$$\text{HG} = \text{HS} + \text{adversary}$$

Stochastic Hybrid Systems:
$$\text{SHS} = \text{HS} + \text{stochastics}$$

Distributed Hybrid Systems:
$$\text{DHS} = \text{HS} + \text{distributed}$$
Dynamic Logics for Dynamical Systems

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

Stochastic Differential Dynamic Logic
\[ \mathcal{SdL} = \mathcal{DL} + \mathcal{SHP} \]

Quantified Differential Dynamic Logic
\[ \mathcal{QdL} = \mathcal{FOL} + \mathcal{DL} + \mathcal{QHP} \]

Differential Game Logic
\[ \mathcal{dGL} = \mathcal{GL} + \mathcal{HG} \]


JAR'08, CADE'11, LMCS'12, LICS'12

LICS'12, CADE'15, TOCL'15

André Platzer (CMU)
Differential Dynamic Logic: Axiomatization

\[ x := e \] \iff P(e) \\
\[ ? Q \] \iff (Q \to P) \\
\[ x' = f(x) \] \iff \forall t \geq 0 \ x := y(t) \ P \quad (y'(t) = f(y)) \\
\[ \cup \] \iff [\alpha] P \land [\beta] P \\
\[ ; \] \iff [\alpha][\beta] P \\
\[ * \] \iff P \land [\alpha][\alpha^*] P \\
K \quad [\alpha] (P \to Q) \to ([\alpha] P \to [\alpha] Q) \\
I \quad [\alpha^*] (P \to [\alpha] P) \to (P \to [\alpha^*] P) \\
C \quad [\alpha^*] \forall v > 0 (P(v) \to \langle \alpha \rangle P(v-1)) \to \forall v (P(v) \to \langle \alpha^* \rangle \exists v \leq 0 P(v))
DW \[ x' = f(x) \& Q \] Q

DC \[ (x' = f(x) \& Q)P \leftrightarrow [x' = f(x) \& Q \land r(x)]P \]
\[ \leftarrow [x' = f(x) \& Q]r(x) \]

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

DI \[ x' = f(x) \& Q \] P \leftrightarrow (Q \rightarrow P \land [x' = f(x) \& Q]P')

DG \[ x' = f(x) \& Q \] P \leftrightarrow \exists y [x' = f(x), y' = a(x)y + b(x) \& Q]P

DS \[ x' = c() \& Q \] P \leftrightarrow \forall t \geq 0 \left( (\forall 0 \leq s \leq t q(x + c()s)) \rightarrow [x := x + c()t]P \right)

\[':=\] \[ x' := e \] p(x') \leftrightarrow p(e)

\[ +' (e + k)' = (e)' + (k)' \]

\[ \cdot' (e \cdot k)' = (e)' \cdot k + e \cdot (k)' \]

\[ o' [y := g(x)][y' := 1]((f(g(x)))') = (f(y))' \cdot (g(x))' \]
**differential dynamic logic**

\[ dL = DL + HP \]

- Multi-dynamical systems
- Combine simple dynamics
- Tame complexity
- Logic & proofs for CPS
- Theory for CPS
- Applications

**Course: Foundations of CPS**
André Platzer.
Logics of dynamical systems.
In LICS [23], pages 13–24.

André Platzer.
Foundations of cyber-physical systems.

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

André Platzer.
Differential dynamic logic for hybrid systems.
André Platzer.
A uniform substitution calculus for differential dynamic logic.
doi:10.1007/978-3-319-21401-6_32.

André Platzer.
Differential game logic.

André Platzer.
The complete proof theory of hybrid systems.
In *LICS* [23], 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.
Stochastic differential dynamic logic for stochastic hybrid programs.
doi:10.1007/978-3-642-22438-6_34.

Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Special issue of selected papers from RV’14.

Stefan Mitsch, Jan-David Quesel, and André Platzer.
Refactoring, refinement, and reasoning: A logical characterization for hybrid systems.


André Platzer.
The structure of differential invariants and differential cut elimination.

André Platzer.
A differential operator approach to equational differential invariants.
doi:10.1007/978-3-642-32347-8_3.

Khalil Ghorbal and André Platzer.
Characterizing algebraic invariants by differential radical invariants.
Yanni Kouskoulas, David W. Renshaw, André Platzer, and Peter Kazanzides.  
Certifying the safe design of a virtual fixture control algorithm for a surgical robot.  
ACM, 2013.  

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.