Refinements of Hybrid Dynamical Systems Logic

André Platzer

Karlsruhe Institute of Technology

Carnegie Mellon University
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.
Cyber-Physical Systems Promise Transformative Impact!

Prospects: Safety & Efficiency

(Autonomous) cars  (Auto)Pilot support  Robots near humans

Cyber-Physical Systems

CPSs combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone.
1. Cyber-Physical Systems & Dynamical Systems
2. Differential Dynamic Logic for Multi-Dynamical Systems
3. Proofs for Dynamical Systems
4. Proofs for Differential Equations
5. Proofs for Hybrid System Refinements
6. Proofs for Hybrid Games
7. Applications
8. Summary
Outline

1. Cyber-Physical Systems & Dynamical Systems
2. Differential Dynamic Logic for Multi-Dynamical Systems
3. Proofs for Dynamical Systems
4. Proofs for Differential Equations
5. Proofs for Hybrid System Refinements
6. Proofs for Hybrid Games
7. Applications
8. Summary

André Platzer (KIT || CMU)
Dynamic Logics for Dynamical Systems

- **differential dynamic logic**
  \[
  dL = DL + HP
  \]

- **differential game logic**
  \[
  dGL = GL + HG
  \]

- **stochastic differential DL**
  \[
  SdL = DL + SHP
  \]

- **quantified differential DL**
  \[
  QdL = FOL + DL + QHP
  \]
Dynamical Systems Analysis

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

$[\alpha] \varphi$ $\varphi$

$[\alpha] \varphi$ $\varphi$

$[\alpha] \varphi$ $\varphi$

$x \neq m$

$x \neq m$

$\alpha$

$x \neq m$

$x \neq m$

Refinements of Hybrid Dynamical Systems Logic
Dynamical Systems Analysis

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

$$[\alpha] \varphi$$

$$[\varphi] x \neq m$$

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

$$x \neq m$$

All runs

$$a$$

$$v$$

$$x$$

$$m$$
Dynamical Systems Analysis

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

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

\[ x \neq m \]

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

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]

\[ x \neq m \]
Definition (Hybrid program)

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

Definition (Differential dynamic logic) (JAR’08, LICS’12)

\[ P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \lor Q \mid P \rightarrow Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P \]
Definition (Hybrid program)

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

Definition (Differential dynamic logic) (JAR’08, LICS’12)

\[ P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \lor Q \mid P \rightarrow Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P \]
Definition (Hybrid program)
\[ \alpha, \beta ::= \ x := e \ | \ ?Q \ | \ x' = f(x) \& Q \ | \ \alpha \cup \beta \ | \ \alpha;\beta \ | \ \alpha^* \]

Definition (Differential dynamic logic) (JAR’08,LICS’12)
\[ P, Q ::= e \geq \bar{e} \ | \ \neg P \ | \ P \land Q \ | \ P \lor Q \ | \ P \rightarrow Q \ | \ \forall x \ P \ | \ \exists x \ P \ | \ [\alpha]P \ | \ \langle \alpha \rangle P \]
### Differential Dynamic Logic dL: Semantics

#### Definition (Hybrid program semantics)

\[ [\cdot] : \text{HP} \to \mathcal{P}(\mathcal{S} \times \mathcal{S}) \]

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

#### Compositional semantics

### Definition (dL semantics)

\[ [\cdot] : \text{Fml} \to \mathcal{P}(\mathcal{S}) \]

\[
\begin{align*}
[ e \geq \tilde{e} ] &= \{ \omega : \omega[e] \geq \omega[\tilde{e}] \} \\
[ P \land Q ] &= [P] \cap [Q] \\
[ \langle \alpha \rangle P ] &= [\alpha] \circ [P] = \{ \omega : \nu \in [P] \text{ for some } \nu : (\omega, \nu) \in [\alpha] \} \\
[ [\alpha] P ] &= [\neg \langle \alpha \rangle \neg P ] = \{ \omega : \nu \in [P] \text{ for all } \nu : (\omega, \nu) \in [\alpha] \} \\
[ \exists x P ] &= \{ \omega : \omega'_x \in [P] \text{ for some } r \in \mathbb{R} \} \\
[ \neg P ] &= [P]^c
\end{align*}
\]
Example (Runaround Robot)

\[(x, y) \neq o \rightarrow [( (?Q_{-1}; \omega := -1 \cup ?Q_1; \omega := 1 \cup ?Q_0; \omega := 0); \{ x' = v, y' = w, v' = \omega w, w' = -\omega v \})^*] (x, y) \neq o \]
Example (Dubins Path)

$$\langle((\omega := -1 \cup \omega := 1 \cup \omega := 0)$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\})^*\rangle(x, y) = o$$

Example (Runaround Robot)

$$(x, y) \neq o \rightarrow [((?Q_{-1}; \omega := -1 \cup ?Q_1; \omega := 1 \cup ?Q_0; \omega := 0);$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\})^*] (x, y) \neq o$$
Example (Dubins Path)

\[ v^2 + w^2 \neq 0 \rightarrow \langle \langle (\omega := -1 \cup \omega := 1 \cup \omega := 0) \\{ x' = v, y' = w, v' = \omega w, w' = -\omega v \} \rangle \rangle (x, y) = o \]

Example (Runaround Robot)

\[ (x, y) \neq o \rightarrow [((?Q_{-1}; \omega := -1 \cup ?Q_{1}; \omega := 1 \cup ?Q_{0}; \omega := 0); \\{ x' = v, y' = w, v' = \omega w, w' = -\omega v \})^*] (x, y) \neq o \]
Differential Dynamic Logic: Example Properties

Safety: $Q \rightarrow [\alpha]P$

Liveness: $Q \rightarrow \langle \alpha \rangle P$

Stability:

$\forall \varepsilon > 0 \exists \delta > 0 \forall x \left( \mathcal{U}_\delta(x = 0) \rightarrow \left[ x' = f(x) \right] \mathcal{U}_\varepsilon(x = 0) \right)$

Attractivity:

$\exists \delta > 0 \forall x \left( \mathcal{U}_\delta(x = 0) \rightarrow \forall \varepsilon > 0 \langle x' = f(x) \rangle \left[ x' = f(x) \right] \mathcal{U}_\varepsilon(x = 0) \right)$
1. Cyber-Physical Systems & Dynamical Systems
2. Differential Dynamic Logic for Multi-Dynamical Systems
3. Proofs for Dynamical Systems
4. Proofs for Differential Equations
5. Proofs for Hybrid System Refinements
6. Proofs for Hybrid Games
7. Applications
8. Summary
Differential Dynamic Logic $dL$: Axiomatization

$$[\text{:=}] \ [x := e]P(x) \leftrightarrow P(e)$$

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

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

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

$$[;] \ [\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$$

$$[*] \ [\alpha^*]P \leftrightarrow P \land [\alpha][\alpha^*]P$$

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

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

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

LICS’12, JAR’17
Proofs for Dynamical Systems

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

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

\[ [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P) \]
Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

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
Outline

1. Cyber-Physical Systems & Dynamical Systems
2. Differential Dynamic Logic for Multi-Dynamical Systems
3. Proofs for Dynamical Systems
4. Proofs for Differential Equations
5. Proofs for Hybrid System Refinements
6. Proofs for Hybrid Games
7. Applications
8. Summary

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

\[ u^2 \leq v^2 + \frac{9}{2} \rightarrow [u' = -v + \frac{u}{4}(1 - u^2 - v^2), \ v' = u + \frac{v}{4}(1 - u^2 - v^2)] \]

\[ u^2 + v^2 = 1 \rightarrow [u' = -v + \frac{u}{4}(1 - u^2 - v^2), \ v' = u + \frac{v}{4}(1 - u^2 - v^2)] \]

Analyzing ODEs via solutions undoes their descriptive power! Poincaré 1881
Proofs for Differential Equations

**DI** \[x' = f(x)] e \geq 0 \iff e \geq 0 \land [x' = f(x)](e)' \geq 0

**DC** 
\[(x' = f(x) \& Q)P \iff [x' = f(x) \& Q \land C]P \]
\[\iff [x' = f(x) \& Q]C\]

**DG** 
\[x' = f(x) \& Q]P \iff \exists y [x' = f(x), y' = a(x)y + b(x) \& Q]P\]
Proofs for Differential Equations

**DI** \[ \dot{x} = f(x) \] \( e \geq 0 \) ⇐ \( e \geq 0 \land \dot{x} = f(x) \) \((e)'/ \geq 0 \)

**DC** \( \left[ \dot{x} = f(x) \land Q \right] P \iff \left[ \dot{x} = f(x) \land Q \land C \right] P \) 
⇐ \( \left[ \dot{x} = f(x) \land Q \right] C \)

**DG** \( \left[ \dot{x} = f(x) \land Q \right] P \)
⇐ \( \exists y \left[ \dot{x} = f(x), \ y' = a(x)y + b(x) \land Q \right] P \)

\( \omega \left[ (e)' \right] = \sum_x \omega(x') \frac{\partial [e]}{\partial x} (\omega) \)
### Theorem (Algebraic Completeness) (LICS’18, JACM’20)

\[ dL \text{ calculus is a sound & complete axiomatization of algebraic invariants of polynomial differential equations. They are decidable by DI, DC, DG in } dL. \]

### Theorem (Semialgebraic Completeness) (LICS’18, JACM’20)

\[ dL \text{ calculus with RI is a sound & complete axiomatization of semialgebraic invariants of differential equations. They are decidable in } dL. \]
Theorem (Algebraic Completeness) \( \text{(LICS'18,JACM'20)} \)

\[ \text{dL calculus is a sound & complete axiomatization of algebraic invariants of polynomial differential equations. They are decidable} \]

\[ \text{DRI } [x' = f(x) \& Q] e = 0 \iff (Q \rightarrow e'^* = 0) \quad (Q \text{ open}) \]

Theorem (Semialgebraic Completeness) \( \text{(LICS'18,JACM'20)} \)

\[ \text{dL calculus with RI is a sound & complete axiomatization of semialgebraic invariants of differential equations. They are decidable} \]

\[ \text{SAI } \forall x (P \rightarrow [x' = f(x)] P) \iff \forall x (P \rightarrow P'^*) \land \forall x (\neg P \rightarrow (\neg P)'^* \land P) \]

Definable \( e'^* \) is short for \textit{all/significant} Lie derivative w.r.t. ODE

Definable \( e'^*\neg \) is w.r.t. backwards ODE \( x' = -f(x) \). Also for \( P \).

\[ e'^* = 0 \equiv e=0 \land (e')'^* = 0 \]

\[ (P \land Q)'^* \equiv P'^* \land Q'^* \]

\[ e'^* \geq 0 \equiv e \geq 0 \land (e=0 \rightarrow (e')'^* \geq 0) \]

\[ (P \lor Q)'^* \equiv P'^* \lor Q'^* \]
Takeaway: Hybrid Systems Logic

Differential dynamic logic

- Logical lingua franca for control systems
- Safety, liveness, controllability, stability are definable by $[\cdot], \langle \cdot \rangle, \forall, \exists$
- Specification and verification interlinked
- Compositional verification helps scale for well-engineered systems
- Small-core complete axiomatization (2000 LOC)
- Differential equation invariants decidable by $dL$ proof
- Significant applications in KeYmaera X theorem prover
Outline

1. Cyber-Physical Systems & Dynamical Systems
2. Differential Dynamic Logic for Multi-Dynamical Systems
3. Proofs for Dynamical Systems
4. Proofs for Differential Equations
5. Proofs for Hybrid System Refinements
6. Proofs for Hybrid Games
7. Applications
8. Summary
Dynamical Systems Relations Analysis

Concept (Differential Refinement Logic) (LICS’16)

\[ \alpha \leq \beta \]

event-triggered

\[
(u \in G(x); x' = f(x) \& Q(x))^*
\]

\[ a \]

\[ v \]

\[ x \]

\[ m \]
Concept (Differential Refinement Logic) (LICS’16)

\[ \alpha \leq \beta \]

event-triggered

\[ [(u \in G(x); x' = f(x) \& Q(x))^*] \text{ safe} \]
Dynamical Systems Relations Analysis

Concept (Differential Refinement Logic) (LICS’16)

\[ \alpha \leq \beta \]

- time-triggered
- event-triggered

\[
\begin{align*}
(u := g(x); x' = f(x) & t \leq T)^*] & \text{ safe} & (u \in G(x); x' = f(x) & Q(x))^* & \text{ safe}
\end{align*}
\]
Dynamical Systems Relations Analysis

Concept (Differential Refinement Logic) (LICS’16)

\[ \alpha \leq \beta \]

time-triggered implementable

event-triggered verifiable

\[
[(u := g(x); x' = f(x) & t \leq T)^*] \text{safe} \quad [(u \in G(x); x' = f(x) & Q(x))^*] \text{safe}
\]
Concept (Differential Refinement Logic) (LICS’16)

\[ (u := g(x); x' = f(x) & t \leq T)^* \] safe ← \[ (u \in G(x); x' = f(x) & Q(x))^* \] safe

\( \alpha \leq \beta \)

time-triggered
implementable

event-triggered
verifiable

\[ a \]

\[ v \]

\[ x \]

\[ m \]
Concept (Differential Refinement Logic) (LICS’16)

\[ \alpha \leq \beta \]

time-triggered implementable

event-triggered verifiable

\[ (u := g(x); x' = f(x) & t \leq T)^* \leq (u \in G(x); x' = f(x) & Q(x))^* \]
Definition (Hybrid program)

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

Definition (Differential refinement logic) (LICS’16)

\[ P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \rightarrow Q \mid \forall x \, P \mid \exists x \, P \mid [\alpha]P \mid \langle \alpha \rangle P \mid \alpha \leq \beta \]

refines
Definition (Hybrid program)

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

Definition (Differential refinement logic) (LICS’16)

\[ P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \rightarrow Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P \mid \alpha \leq \beta \]

Refinements of Hybrid Dynamical Systems Logic

\[ \omega \]

\[ \beta \text{-span} \]

\[ [\alpha]P \]

\[ \langle \beta \rangle P \]

\[ \alpha \text{-span} \]

\[ \alpha \leq \beta \]

\[ \text{refines} \]

\[ \alpha \text{ less behavior} \]
Differential Refinement Logic dRL: Semantics

Definition (Hybrid program semantics) \((\llbracket \cdot \rrbracket : \text{HP} \to \mathcal{P}(\mathcal{S} \times \mathcal{S}))\)

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

compositional semantics

Definition (dRL semantics) \((\llbracket \cdot \rrbracket : \text{Fml} \to \mathcal{P}(\mathcal{S}))\)

- \([\alpha \leq \beta] = \{\omega : \{\nu : (\omega, \nu) \in \llbracket \alpha \rrbracket\} \subseteq \{\nu : (\omega, \nu) \in \llbracket \beta \rrbracket\}\}\)
- \([e \geq \tilde{e}] = \{\omega : \omega[e] \geq \omega[\tilde{e}]\}\)
- \([\neg P] = \llbracket P \rrbracket^c\)
- \([P \land Q] = \llbracket P \rrbracket \cap \llbracket Q \rrbracket\)
- \([\langle \alpha \rangle P] = \llbracket \alpha \rrbracket \circ \llbracket P \rrbracket = \{\omega : \nu \in \llbracket P \rrbracket \text{ for some } \nu : (\omega, \nu) \in \llbracket \alpha \rrbracket\}\)
- \([\llbracket \alpha \rrbracket P]\) = \([\neg \langle \alpha \rangle \neg P]\) = \{\omega : \nu \in \llbracket P \rrbracket \text{ for all } \nu : (\omega, \nu) \in \llbracket \alpha \rrbracket\}\)
Differential Refinement Logic: Axiomatization

\[ \frac{P \rightarrow [\alpha]Q \quad P \rightarrow \gamma \leq \alpha}{P \rightarrow [\gamma]Q} \]

\[ \frac{P \rightarrow \langle \alpha \rangle Q \quad P \rightarrow \alpha \leq \gamma}{P \rightarrow \langle \gamma \rangle Q} \]

\[ P \rightarrow \alpha_1 \leq \alpha_2 \quad P \rightarrow [\alpha_1](\beta_1 \leq \beta_2) \]

\[ P \rightarrow (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2) \]

\[ P \rightarrow \alpha_1 \leq \beta \land \alpha_2 \leq \beta \]

\[ P \rightarrow \alpha_1 \cup \alpha_2 \leq \beta \]

\[ P \rightarrow [\alpha^*](\alpha; \gamma \leq \gamma) \quad P \rightarrow [\alpha^*](\beta \leq \gamma) \]

\[ P \rightarrow \alpha^*; \beta \leq \gamma \]

\[ P \rightarrow \beta \leq \gamma \quad P \rightarrow \gamma; \alpha \leq \gamma \]

\[ P \rightarrow \beta; \alpha^* \leq \gamma \]

\[ P \rightarrow [\alpha^*](\alpha \leq \beta) \]

\[ P \rightarrow \alpha^* \leq \beta^* \]

\[ \forall y (\langle \alpha \rangle x = y \rightarrow \langle \beta \rangle x = y) \]

\[ P \rightarrow [\alpha]P \leftrightarrow \alpha \leq (x := *; ?P) \]

\[ P \rightarrow \alpha_1 \leq \alpha_2 \quad \beta_1 \leq \beta_2 \]

\[ P \rightarrow (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2) \]
\[
\begin{align*}
\text{[≤]} & \\
\frac{P \rightarrow [\alpha]Q}{P \rightarrow [\gamma]Q} \\
\frac{P \rightarrow \langle \alpha \rangle Q}{P \rightarrow \langle \gamma \rangle Q} \\
\frac{P \rightarrow \alpha_1 \leq \alpha_2}{P \rightarrow [\alpha_1](\beta_1 \leq \beta_2)} \\
\frac{P \rightarrow \alpha_1 \leq \beta \land \alpha_2 \leq \beta}{P \rightarrow \alpha_1 \cup \alpha_2 \leq \beta} \\
\frac{P \rightarrow [\alpha^*](\alpha; \gamma \leq \gamma)}{P \rightarrow \alpha^*; \beta \leq \gamma} \\
\frac{P \rightarrow \beta \leq \gamma}{P \rightarrow \gamma; \alpha \leq \gamma} \\
\frac{P \rightarrow [\alpha^*](\alpha \leq \beta)}{P \rightarrow \alpha^* \leq \beta^*}
\end{align*}
\]
Takeaway: Hybrid System Refinements

Differential refinement logic
- Event-triggered control: Easy to verify but hard to implement
- Time-triggered control: Easy to implement but hard to verify
- Best of both worlds: verify event-triggered, implement time-triggered
- dRL proofs identify required conditions (e.g., event invariance)
- Implementation model \( \neq \) verification model
- Iterative design reduces risk, increases repeated effort
- Hierarchical proof structuring by refinement

Relations \( \alpha \leq \beta \) between hybrid systems models are just as useful as properties \([\alpha] \varphi\) of hybrid systems models.
Simultaneous logical language integration is best.
1 Cyber-Physical Systems & Dynamical Systems
2 Differential Dynamic Logic for Multi-Dynamical Systems
3 Proofs for Dynamical Systems
4 Proofs for Differential Equations
5 Proofs for Hybrid System Refinements
6 Proofs for Hybrid Games
7 Applications
8 Summary
Example: Goalie in Robot Soccer

\[ x < 0 \land v > 0 \land y = g \rightarrow \]
\[ \langle (w := +w \land w := -w); \]
\[ ((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \rangle x^2 + (y - g)^2 \leq 1 \]
Example: Goalie in Robot Soccer

\[ x < 0 \land v > 0 \land y = g \rightarrow \]
\[ \langle (w := +w \cap w := -w); \]
\[ (u := +u \cup u := -u); \{x' = v, y' = w, g' = u\} \rangle^* x^2 + (y - g)^2 \leq 1 \]
Example: Goalie in Robot Soccer

\[ x < 0 \land v > 0 \land y = g \rightarrow \]
\[ \langle (w := +w \land w := -w) ; \\
(\{u := +u \lor u := -u\}; \{x' = v, y' = w, g' = u\}^*) \rangle x^2 + (y - g)^2 \leq 1 \]
Example: Goalie in Robot Soccer

\[ x < 0 \land v > 0 \land y = g \rightarrow \left( (w := +w \cap w := -w); ((u := +u \cup u := -u); \{ x' = v, y' = w, g' = u \})^* \right) x^2 + (y - g)^2 \leq 1 \]
Example: Goalie in Robot Soccer

\[
x < 0 \land v > 0 \land y = g \rightarrow
\left< \left( w := +w \land w := -w \right); \right.
\left( u := +u \lor u := -u \right); \left\{ x' = v, y' = w, g' = u \right\} \right)^* x^2 + (y - g)^2 \leq 1
\]
Goalie's Secret

\[ \left( \frac{x}{v} \right)^2 (u - w)^2 \leq 1 \land \]
\[ x < 0 \land v > 0 \land y = g \rightarrow \]
\[ \left( w := +w \land w := -w \right); \]
\[ \left( (u := +u \lor u := -u); \{ x' = v, y' = w, g' = u \} \right)^* \]}
\[ x^2 + (y - g)^2 \leq 1 \]
**Definition (Hybrid game)**

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

**Definition (Differential game logic)**

\( P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid P \lor Q \mid P \rightarrow Q \mid \forall x P \mid \exists x P \mid [\alpha]P \mid \langle \alpha \rangle P \)

\[ \begin{align*}
\llbracket \alpha \rrbracket(x) & \downarrow \llbracket \alpha \cup \beta \rrbracket(x) \\
\llbracket \beta \rrbracket(x) & \downarrow \llbracket \alpha ; \beta \rrbracket(x) \\
\llbracket \alpha^* \rrbracket(x) & \downarrow \llbracket \alpha^d \rrbracket(x) \\
\end{align*} \]
**Definition (Hybrid game)**

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

**Definition (Differential game logic)**

\[
P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \& Q \mid P \lor Q \mid P \rightarrow Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha] P \mid \langle \alpha \rangle P
\]
### Differential Game Logic: Denotational Semantics

**Definition (Hybrid game \( \alpha \))**

\[
\begin{align*}
\varsigma_x := e(X) &= \{ \omega \in \mathcal{I} : \omega^\omega_x[e] \in X \} \\
\varsigma_x' := f(x)(X) &= \{ \varphi(0) \in \mathcal{I} : \varphi(r) \in X, \frac{d\varphi(t)(x)}{dt}(\zeta) = \varphi(\zeta)[f(x)] \text{ for all } \zeta \} \\
\varsigma ? Q(X) &= [Q] \cap X \\
\varsigma_{\alpha \cup \beta}(X) &= \varsigma_{\alpha}(X) \cup \varsigma_{\beta}(X) \\
\varsigma_{\alpha ; \beta}(X) &= \varsigma_{\alpha}(\varsigma_{\beta}(X)) \\
\varsigma_{\alpha^*}(X) &= \bigcap \{ Z \subseteq \mathcal{I} : X \cup \varsigma_{\alpha}(Z) \subseteq Z \} \\
\varsigma_{\alpha^d}(X) &= (\varsigma_{\alpha}(X^c))^c
\end{align*}
\]

### Definition (dGL Formula \( P \))

\[
\begin{align*}
[e \geq \tilde{e}] &= \{ \omega \in \mathcal{I} : \omega[e] \geq \omega[\tilde{e}] \} \\
[\lnot P] &= ([P])^c \\
[P \land Q] &= [P] \cap [Q] \\
[\langle \alpha \rangle P] &= \varsigma_{\alpha}([P]) \\
[[\alpha] P] &= \delta_{\alpha}([P])
\end{align*}
\]

**compositional semantics**
Differential Game Logic: Axiomatization

\[ \Box [\alpha]P \leftrightarrow \neg \langle \alpha \rangle \neg P \]

\[ \langle := \rangle \langle x := e \rangle p(x) \leftrightarrow p(e) \]

\[ \langle ' \rangle \langle x' = f(x) \rangle P \leftrightarrow \exists t \geq 0 \langle x := y(t) \rangle P \]

\[ \langle ? \rangle \langle ? Q \rangle P \leftrightarrow (Q \land P) \]

\[ \langle U \rangle \langle \alpha \cup \beta \rangle P \leftrightarrow \langle \alpha \rangle P \lor \langle \beta \rangle P \]

\[ \langle ; \rangle \langle \alpha ; \beta \rangle P \leftrightarrow \langle \alpha \rangle \langle \beta \rangle P \]

\[ \langle * \rangle \langle \alpha^* \rangle P \leftrightarrow P \lor \langle \alpha \rangle \langle \alpha^* \rangle P \]

\[ \langle d \rangle \langle \alpha^d \rangle P \leftrightarrow \neg \langle \alpha \rangle \neg P \]

\[ \begin{array}{c}
P \rightarrow Q \\
\frac{\langle \alpha \rangle P \rightarrow \langle \alpha \rangle Q}{MP} \\
P \lor \langle \alpha \rangle Q \rightarrow Q \quad \text{(FP)} \\
\frac{\langle \alpha^* \rangle P \rightarrow Q}{MP} \\
P \rightarrow Q \quad \text{(MP)} \\
\frac{p \rightarrow Q}{\forall x Q} \quad \text{(x \not\in \text{FV}(p))} \\
\end{array} \]

\[ \begin{array}{c}
\varphi \\
\frac{\psi(\cdot)}{\varphi_{p(\cdot)}} \quad \text{(US)}
\end{array} \]
Takeaway: Hybrid Games

Differential game logic
- True adversarial competition
- Analytic competition: different agents reach decisions independently
- Cause: misunderstandings, interference, disturbance, different goals
- More general semantics, tame axiomatics
- Compositional verification
- Small-core complete axiomatization in KeYmaera X theorem prover
- Differential game invariants for differential hybrid games
- Almost everything is characterizable via hybrid games
- Arbitrarily nested inductive / coinductive concepts over augmented $\mathbb{R}$
Outline

1. Cyber-Physical Systems & Dynamical Systems
2. Differential Dynamic Logic for Multi-Dynamical Systems
3. Proofs for Dynamical Systems
4. Proofs for Differential Equations
5. Proofs for Hybrid System Refinements
6. Proofs for Hybrid Games
7. Applications
8. Summary
Applications with Impact

Prospects: Safety & Efficiency

| (Autonomous) cars | (Auto)Pilot support | Robots near humans |

Cyber-Physical Systems

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

STTT’17, TECS’22
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared ($\approx 899 \times 10^6$ counterexamples).

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).

**Safe Version: Action Issued = CL1500**
Followed by Most Extreme Up/Down-sense Available

ACAS X issues Maintain advisory instead of CL1500
Airborne Collision Avoidance Games in ACAS X

- Ownship and intruder aircraft both maneuver
- Intruder aircraft chooses actions independently
- ACAS X is a hybrid game

Identified safe region for each advisory symbolically

Proved safety for hybrid games flight model in KeYmaera X
Airborne Collision Avoidance Games in ACAS X

- Ownship and intruder aircraft both maneuver
- Intruder aircraft chooses actions independently
- ACAS X is a hybrid game

Identified safe region for each advisory symbolically

Proved safety for hybrid games flight model in KeYmaera X
Airborne Collision Avoidance Games in ACAS X

- Ownship and intruder aircraft both maneuver
- Intruder aircraft chooses actions independently
- ACAS X is a hybrid game

1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid games flight 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

André Platzer (KIT || CMU) Refinements of Hybrid Dynamical Systems Logic ABZ’23
<table>
<thead>
<tr>
<th>Safety</th>
<th>Invariant + Safe Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>$|p - o|_\infty &gt; \frac{s^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\epsilon^2 + \epsilon s\right)$</td>
</tr>
<tr>
<td>passive</td>
<td>$s \neq 0 \rightarrow |p - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\epsilon^2 + \epsilon(s + V)\right)$</td>
</tr>
<tr>
<td>+ sensor</td>
<td>$|\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\epsilon^2 + \epsilon(s + V)\right) + \Delta p$</td>
</tr>
<tr>
<td>+ disturb.</td>
<td>$|p - o|_\infty &gt; \frac{s^2}{2b\Delta a} + V \frac{s}{b\Delta a} + \left(\frac{A}{b\Delta a} + 1\right)\left(\frac{A}{2}\epsilon^2 + \epsilon(s + V)\right)$</td>
</tr>
<tr>
<td>+ failure</td>
<td>$|\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V \frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\epsilon^2 + \epsilon(v + V)\right) + \Delta p + g\Delta$</td>
</tr>
<tr>
<td>friendly</td>
<td>$|p - o|_\infty &gt; \frac{s^2}{2b} + \frac{V^2}{2b_0} + V\left(\frac{s}{b} + \tau\right) + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\epsilon^2 + \epsilon(s + V)\right)$</td>
</tr>
</tbody>
</table>
Approach: Safety Proofs for Autonomous CPS

KeYmaera X generates proofs

actions: \{ \text{acc, brake} \}

motion: \dddot{x} = a

Monitor transfers safety

ModelPlex proof synthesizes

Compliance Monitor

Model Safety

Proof and invariant search

Autonomous CPS

Model
Further Dynamical Systems Challenges

CPSs deserve proofs as safety evidence!

- Verified CPS implementations by ModelPlex
- Correct CPS execution
- CPS proof and tactic languages+libraries
- Big CPS built from safe components
- ODE invariance
- ODE liveness
- ODE stability
- Invariant generation
- Safe AI autonomy in CPS
- Refinement + system property proofs
- CPS information flow
- Hybrid games
- Constructive hybrid games

FMSD’16
PLDI’18
ITP’17
STTT’18
JACM’20
FAC’21
TACAS’21
FMSD’21
AAAI’18
LICS’16
LICS’18
TOCL’15
IJCAR’20

discrete
continuous
adversarial
nondet
stochastic
Refinements of Hybrid Dynamical Systems Logic

**differential dynamic logic**

\[ dL = DL + HP \]

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

**KeYmaera X**

- Logic & Proofs for CPS
- Programming languages
- Theorem proving
- Multi-dynamical systems

André Platzer (KIT || CMU)
I Part: Elementary Cyber-Physical Systems
2. Differential Equations & Domains
3. Choice & Control
4. Safety & Contracts
5. Dynamical Systems & Dynamic Axioms
6. Truth & Proof
7. Control Loops & Invariants
8. Events & Responses
9. Reactions & Delays

II Part: Differential Equations Analysis
10. Differential Equations & Differential Invariants
11. Differential Equations & Proofs
12. Ghosts & Differential Ghosts
13. Differential Invariants & Proof Theory

III Part: Adversarial Cyber-Physical Systems
14-17. Hybrid Systems & Hybrid Games

IV Part: Comprehensive CPS Correctness
Appendix

- Soundness and Completeness
- Uniform Substitution
- ModelPlex Runtime Model Validation
- Robot Applications
- Safe AI in CPS
Appendix

- Soundness and Completeness
- Uniform Substitution
- ModelPlex Runtime Model Validation
- Robot Applications
- Safe AI in CPS
Theorem (Sound & Complete) (JAR’08, LICS’12, JAR’17)

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

\[ \models P \iff \text{FOD} \vdash_{dL} P \]
**Theorem (Sound & Complete)**

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

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
Uniform Substitution

Theorem (Soundness) replace all occurrences of \( p(\cdot) \)

\[
\text{US} \quad \frac{\phi}{\sigma(\phi)}
\]

provided \( \text{FV}(\sigma|_{\Sigma(\theta)}) \cap \text{BV}(\otimes(\cdot)) = \emptyset \) for each operation \( \otimes(\theta) \) in \( \phi \)

i.e. bound variables \( U = \text{BV}(\otimes(\cdot)) \) of no operator \( \otimes \)
are free in the substitution on its argument \( \theta \)

\( (U\text{-admissible}) \)

\[
\begin{align*}
\text{US} \quad [a \cup b]p(\bar{x}) & \iff [a]p(\bar{x}) \land [b]p(\bar{x}) \\
[x := x + 1 \cup x' = 1]x \geq 0 & \iff [x := x + 1]x \geq 0 \land [x' = 1]x \geq 0
\end{align*}
\]
Uniform Substitution

Theorem (Soundness) replace all occurrences of $p(\cdot)$

$$US \quad \frac{\phi}{\sigma(\phi)}$$

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ for each operation $\otimes(\theta)$ in $\phi$

i.e. bound variables $U = BV(\otimes(\cdot))$ of no operator $\otimes$ are free in the substitution on its argument $\theta$ (U-admissible)

$$[v := f]p(v) \leftrightarrow p(f)$$

$$[v := -x][x' = v] x \geq 0 \leftrightarrow [x' = -x] x \geq 0$$
Uniform Substitution

Theorem (Soundness) replace all occurrences of $p(\cdot)$

\[
US \quad \frac{\phi}{\sigma(\phi)}
\]

provided $\text{FV}(\sigma | \Sigma(\theta)) \cap \text{BV}(\otimes(\cdot)) = \emptyset$ for each operation $\otimes(\theta)$ in $\phi$

i.e. bound variables $U = \text{BV}(\otimes(\cdot))$ of no operator $\otimes$

are free in the substitution on its argument $\theta$ (U-admissible)

If you bind a free variable, you go to logic jail!

\[
[v := f]p(v) \leftrightarrow p(f)
\]

Clash

\[
[v := -x][x' = v] x \geq 0 \leftrightarrow [x' = -x] x \geq 0
\]
ModelPlex ensures that verification results about models apply to CPS implementations.

Insights

Verification results about models transfer to the CPS when validating model compliance.

Compliance with model is characterizable in logic $dL$.

Compliance formula transformed by $dL$ proof to monitor.

Correct-by-construction provably correct model validation at runtime.
ModelPlex ensures that verification results about models apply to CPS implementations.

Insights

- Verification results about models transfer to the CPS when validating model compliance.
- Compliance with model is characterizable in logic $dL$.
- Compliance formula transformed by $dL$ proof to monitor.
- Correct-by-construction provably correct model validation at runtime.
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

Identified safe region for each safety notion symbolically
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

- Pass parking
- Avoid/Follow
- Head-on
- Turn

Identified safe region for each safety notion symbolically

Proved safety for hybrid systems ground robot model in KeYmaera X

André Platzer (KIT || CMU)
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 (KIT || CMU)
Refinements of Hybrid Dynamical Systems Logic
ABZ’23 42 / 35
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 (KIT || CMU)
Refinements of Hybrid Dynamical Systems Logic
ABZ’23 42 / 35
### Safety Invariant + Safe Control

<table>
<thead>
<tr>
<th>Static</th>
<th>$|p - o|_\infty &gt; \frac{s^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon s\right)$</th>
</tr>
</thead>
<tbody>
<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>
</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>
</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>
</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>
</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>
</tr>
<tr>
<td>Safety</td>
<td>Invariant</td>
</tr>
<tr>
<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>
</tr>
<tr>
<td>passive</td>
<td>$s \neq 0 \Rightarrow |p - o|_\infty &gt; \frac{s^2}{2b} + Vs + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon (s + V)\right)$</td>
</tr>
<tr>
<td>+ sensor</td>
<td>$|\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon (s + V)\right)$</td>
</tr>
<tr>
<td>+ disturb.</td>
<td>$|p - o|_\infty &gt; \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon (s + V)\right)$</td>
</tr>
<tr>
<td>+ failure</td>
<td>$|\hat{p} - o|_\infty &gt; \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon (v + V)\right)$</td>
</tr>
<tr>
<td>friendly</td>
<td>$|p - o|_\infty &gt; \frac{s^2}{2b} + \frac{V^2}{2b_0} + V\left(\frac{s}{b} + \tau\right) + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon (s + V)\right)$</td>
</tr>
</tbody>
</table>

**Question**

How to find and justify constraints? Proof!
Reinforcement Learning learns from experience of trying actions
RL chooses an action, observes outcome, reinforces in policy if successful
ModelPlex monitor inspects each decision, vetoes if unsafe
ModelPlex monitor gives early feedback about possible future problems. No need to wait till disaster strikes and propagate back.
dL benefits from RL optimization.

RL benefits from dL safety signal.
Learning to Act Safely in a CPS

\[ \text{observe} \ \text{accel} \cup \text{brake} \]

- **Theorem**: Safe policy if ODE accurate
- **Experiment**: Graceful recovery outside ODE \( \leadsto \) quantitative ModelPlex
  - Detect modeled versus unmodeled state space \( \leadsto \) ModelPlex

AAAI’18, ITC’18, TACAS’19, QEST’19

André Platzer (KIT || CMU)
Learning to Act Safely in a CPS

What’s safe when off model?

accel ∪ brake

observe
Learning to Act Safely in a CPS with Multiple Models

What’s safe with multiple possible models?

accel ∪ brake

observe

AAAI’18, ITC’18, TACAS’19, QEST’19
ModelPlex monitors conjunction of all plausible models
Learning to Act Safely in a CPS with Multiple Models

Remove incompatible models after contradictory observation
Learning to Act Safely in a CPS with Multiple Models

Plan differentiating experiment \( \leadsto \) predictive monitor distinctions

André Platzer (KIT ∥ CMU)

Refinements of Hybrid Dynamical Systems Logic

AAAI'18, ITC'18, TACAS'19, QEST'19
Convergence

Plausible models converge to true model a.s., if possible
Modify model to fit observations by verification-preserving model update. Safety proofs reified: modify model + proof tactic to preserve fit + safety
 André Platzer.
Logics of dynamical systems.
In LICS [23], pages 13–24.

 André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.
doi:10.1007/978-3-319-63588-0.

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

 André Platzer.
The complete proof theory of hybrid systems.
In LICS [23], pages 541–550.
doi:10.1109/LICS.2012.64.

 André Platzer and Yong Kiam Tan.
Differential equation invariance axiomatization.


**André Platzer.**

Logic & proofs for cyber-physical systems.


doi:10.1007/978-3-319-40229-1_3.

**André Platzer.**

Differential dynamic logic for hybrid systems.


**André Platzer.**

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


Special issue for selected papers from CSL’10.

André Platzer.

André Platzer.

André Platzer.
Differential game logic. 

André Platzer.
Differential hybrid games. 
Yong Kiam Tan and André Platzer.
An axiomatic approach to existence and liveness for differential equations.
doi:10.1007/s00165-020-00525-0.

Yong Kiam Tan and André Platzer.
Deductive stability proofs for ordinary differential equations.
doi:10.1007/978-3-030-72013-1_10.

Sarah M. Loos and André Platzer.
Differential refinement logic.


Nathan Fulton and André Platzer. Verifiably safe off-model reinforcement learning.