14: Hybrid Systems & Games

Logical Foundations of Cyber-Physical Systems

André Platzer

Carnegie Mellon University
Computer Science Department
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
Learning Objectives
Hybrid Systems & Games

- fundamental principles of computational thinking
- logical extensions
- PL modularity principles
- compositional extensions
- differential game logic
- best/worst-case analysis
- models of alternating computation

- adversarial dynamics
- conflicting actions
- multi-agent systems
- angelic/demonic choice

- multi-agent state change
- CPS semantics
- reflections on choices
1 Learning Objectives

2 Motivation

3 A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls

4 Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer

5 An Informal Operational Game Tree Semantics

6 Summary
CPS Analysis: Robot Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

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

Challenge (Games)

Game rules describing play evolution with both

- Angelic choices (player ♦ Angel)
- Demonic choices (player □ Demon)

<table>
<thead>
<tr>
<th>♦</th>
<th>Tr</th>
<th>Pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trash</td>
<td>1,2</td>
<td>0,0</td>
</tr>
<tr>
<td>Plant</td>
<td>0,0</td>
<td>2,1</td>
</tr>
</tbody>
</table>
CPS Analysis: Robot Control

Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel $\Diamond$ vs. Demon $\Box$)
Challenge (Hybrid Games)

Game rules describing play evolution with

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel $\diamondsuit$ vs. Demon $\Box$)

\[ a \]
\[ \omega \]
\[ d \]
\[ d_x \]
\[ d_y \]
Game rules describing play evolution with
- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
- Adversarial dynamics (Angel ♦ vs. Demon ♣)
CPSs are Multi-Dynamical Systems

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

CPS Compositions
CPS combines multiple simple dynamical effects.

Tame Parts
Exploiting compositionality tames CPS complexity.

Descriptive simplification
Analytic simplification
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**: \( QdL = FOL + DL + QHP \)

\( [\alpha] \psi \)
\( \langle \alpha \rangle \psi \)
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
### Definition (Hybrid program $\alpha$)

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

### Definition (dL Formula $P$)

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

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

\[
\begin{align*}
  & x := e \mid ?Q \mid x' = f(x) \land Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^* \\
\end{align*}
\]

**Definition (dL Formula \( P \))**

\[
\begin{align*}
  & e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x\ P \mid \exists x\ P \mid [\alpha]\ P \mid \langle\alpha\rangle\ P \\
\end{align*}
\]
Differential Dynamic Logic $dL$: Nondeterminism

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

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

**Definition ($dL$ Formula $P$)**

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

Nondeterminism during HP runs
Differential Dynamic Logic dL: Nondeterminism

Definition (Hybrid program $\alpha$)

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

Definition (dL Formula $P$)

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

Nondeterminism during HP runs
Differential Dynamic Logic $dL$: Nondeterminism

Definition (Hybrid program $\alpha$)

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

Definition (dL Formula $P$)

\[
e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha] P \mid \langle \alpha \rangle P
\]
Differential Dynamic Logic dL: Nondeterminism

Definition (Hybrid program $\alpha$)

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

Definition (dL Formula $P$)

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

Modality decides the mode: help/hurt

All choices resolved in one way

Differential Equation

Nondet. Choice

Nondet. Repeat

All Choices

Some Choice
Differential Dynamic Logic dL: Nondeterminism

Definition (Hybrid program $\alpha$)

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

Definition (dL Formula $P$)

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

Modality decides the mode: help/hurt

\[
[\alpha_1]\langle\alpha_2\rangle[\alpha_3]\langle\alpha_4\rangle P \quad \text{only fixed interaction depth}
\]
Angel Ops

- \( \cup \) choice
- \(*\) repeat
- \( x' = f(x) \) evolve
- \(?Q\) challenge

Let Angel be one player
Control & Dual Control Operators

**Angel Ops**
- $\cup$ choice
- $\ast$ repeat
- $x' = f(x)$ evolve
- $?Q$ challenge

**Demon Ops**
- $\cap$ choice
- $\times$ repeat
- $x' = f(x)^d$ evolve
- $?Q^d$ challenge

Let Angel be one player

Let Demon be another player
Duality operator $^d$ passes control between players
Game Operators

◊ Angel Ops

∪ choice
* repeat
\(x' = f(x)\) evolve
?Q challenge

□ Demon Ops

∩ choice
× repeat
\(x' = f(x)^d\) evolve
?Q^d challenge

Duality operator \(d\) passes control between players
Game Operators

- **Angel Ops**
  - $\cup$ choice
  - $\ast$ repeat
  - $x' = f(x)$ evolve
  - $?Q$ challenge

- **Demon Ops**
  - $\cap$ choice
  - $\times$ repeat
  - $x' = f(x)^d$ evolve
  - $?Q^d$ challenge

Duality operator $^d$ passes control between players
Game Operators

Diamond Angel Ops

\[
\begin{align*}
\cup & \quad \text{choice} \\
\ast & \quad \text{repeat} \\
\triangledown & \quad \text{evolve} \\
?Q & \quad \text{challenge}
\end{align*}
\]

Diamond Demon Ops

\[
\begin{align*}
\cap & \quad \text{choice} \\
\times & \quad \text{repeat} \\
\triangledown & \quad \text{evolve} \\
?Q^d & \quad \text{challenge}
\end{align*}
\]

Duality operator \( d \) passes control between players
Definable Game Operators

Angel Ops
- $\cup$ choice
- $\ast$ repeat
- $x' = f(x)$ evolve
- $?Q$ challenge

Demon Ops
- $\cap$ choice
- $\times$ repeat
- $x' = f(x)^d$ evolve
- $?Q^d$ challenge

if $(Q) \alpha$ else $\beta \equiv$ 
while $(Q) \alpha \equiv$

$\alpha \cap \beta \equiv$
$\alpha \times \equiv$
$(x' = f(x) & Q)^d \quad x' = f(x) & Q$
$(x := e)^d \quad x := e$
$?Q^d \quad ?Q$
Definable Game Operators

- **Angel Ops**
  - $\cup$ choice
  - $\ast$ repeat
  - $x' = f(x)$ evolve
  - $?Q$ challenge

- **Demon Ops**
  - $\cap$ choice
  - $\times$ repeat
  - $x' = f(x)^d$ evolve
  - $?Q^d$ challenge

- if $(Q) \alpha$ else $\beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta)$
- while $(Q) \alpha \equiv$
  - $\alpha \cap \beta \equiv$
  - $\alpha \times \equiv$
- $(x' = f(x) \& Q)^d$ $x' = f(x) \& Q$
- $(x := e)^d$ $x := e$
- $?Q^d$ $?Q$
Definable Game Operators

**Angel Ops**

- `∪` choice
- `*` repeat
- `x' = f(x)` evolve
- `?Q` challenge

**Demon Ops**

- `∩` choice
- `×` repeat
- `x' = f(x)^d` evolve
- `?Q^d` challenge

```
if (Q) α else β ≡ (?Q; α) ∪ (?¬Q; β)
while (Q) α ≡ (?Q; α)^* ; ?¬Q
α ∩ β ≡
α ⊓ ≡
(x' = f(x) & Q)^d  x' = f(x) & Q
(x := e)^d  x := e
?Q^d  ?Q
```
Definable Game Operators

**Angel Ops**
- Union: $\cup$
- Choice: choice
- Repeat: $\ast$
- Evolve: $x' = f(x)$
- Challenge: $?Q$

**Demon Ops**
- Intersection: $\cap$
- Choice: choice
- Repeat: $\times$
- Evolve: $x' = f(x)^d$
- Challenge: $?Q^d$

**Formulas**
- If/Else: $\text{if}(Q)\alpha\text{else }\beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta)$
- While: $\text{while}(Q)\alpha \equiv (?Q; \alpha)^*; ?\neg Q$
- Intersection: $\alpha \cap \beta \equiv$
- Cartesian Product: $\alpha \times \equiv$
- Evolve: $x' = f(x) \& Q \equiv (x' = f(x) \& Q)^d$
- Assignment: $x := e \equiv (x := e)^d$
- Challenge: $?Q \equiv ?Q^d$

André Platzer (CMU)
Definable Game Operators

Angel Ops

- $\cup$ choice
- $\ast$ repeat
- $x' = f(x)$ evolve
- $?Q$ challenge

Demon Ops

- $\cap$ choice
- $\times$ repeat
- $x' = f(x)^d$ evolve
- $?Q^d$ challenge

\[
\text{if}(Q) \alpha \text{ else } \beta \equiv (\ ?Q; \alpha) \cup (\ ?\neg Q; \beta)
\]

\[
\text{while}(Q) \alpha \equiv (\ ?Q; \alpha)^\ast ; \ ?\neg Q
\]

\[
\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d
\]

\[
\alpha \times \equiv
\]

\[
(x' = f(x) & Q)^d \quad x' = f(x) & Q
\]

\[
(x := e)^d \quad x := e
\]

\[
?Q^d \quad ?Q
\]
Definable Game Operators

**Angel Ops**
- `∪` choice
- `∗` repeat
- `x' = f(x)` evolve
- `?Q` challenge

**Demon Ops**
- `∩` choice
- `×` repeat
- `x' = f(x)^d` evolve
- `?Q^d` challenge

**Formulas**
- `if(Q) α else β ≡ (?Q; α) ∪ (?¬Q; β)`
- `while(Q) α ≡ (?Q; α)^∗; ?¬Q`
- `α ∩ β ≡ (α^d ∪ β^d)^d`
- `α × ≡ ((α^d)^∗)^d`
- `(x' = f(x) & Q)^d`  
- `(x := e)^d`  
- `?Q^d`  
- `?Q`
Definable Game Operators

$$\begin{align*}
\text{Angel Ops} & : & \cup & \text{choice} \\
& & * & \text{repeat} \\
& & x' = f(x) & \text{evolve} \\
& & ?Q & \text{challenge} \\
\text{Demon Ops} & : & \cap & \text{choice} \\
& & \times & \text{repeat} \\
& & x' = f(x)^d & \text{evolve} \\
& & ?Q^d & \text{challenge}
\end{align*}$$

if\,(Q)\,\alpha\,\text{else}\,\beta \equiv (?Q;\,\alpha) \cup (?\neg Q;\,\beta)$$

while\,(Q)\,\alpha \equiv (?Q;\,\alpha)^*;\,?\neg Q$$

$$\begin{align*}
\alpha \cap \beta & \equiv (\alpha^d \cup \beta^d)^d \\
\alpha \times & \equiv ((\alpha^d)^*)^d \\
(x' = f(x) \& Q)^d & \not\equiv x' = f(x) \& Q \\
(x := e)^d & \equiv x := e \\
?Q^d & \equiv ?Q
\end{align*}$$
Definable Game Operators

**Angel Ops**
- $\cup$
- $\ast$
- $x' = f(x)$
- $?Q$
- choice
- repeat
- evolve
- challenge

**Demon Ops**
- $\cap$
- $\times$
- $x' = f(x)^d$
- $?Q^d$
- choice
- repeat
- evolve
- challenge

\[\text{if}(Q) \alpha \text{ else } \beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta)\]
\[\text{while}(Q) \alpha \equiv (?Q; \alpha)^*; ?\neg Q\]
\[\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d\]
\[\alpha \times \equiv ((\alpha^d)^*)^d\]
\[(x' = f(x) \& Q)^d \not\equiv x' = f(x) \& Q\]
\[(x := e)^d \equiv x := e\]
\[?Q^d \not\equiv ?Q\]
Definable Game Operators

**Angel Ops**

- `∪` choice
- `*` repeat
- `x' = f(x)` evolve
- `?Q` challenge

**Demon Ops**

- `∩` choice
- `×` repeat
- `x' = f(x)^d` evolve
- `?Q^d` challenge

```
if(Q) α else β ≡ (?Q; α) ∪ (?¬Q; β)
while(Q) α ≡ (?Q; α)*; ?¬Q
α ∩ β ≡ (α^d ∪ β^d)^d
α × ≡ ((α^d)^*)^d
(x' = f(x) & Q)^d ≡ x' = f(x) & Q
(x := e)^d ≡ x := e
?Q^d ≠ ?Q
```
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
Definition (Hybrid game $\alpha$)

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

Definition (Hybrid game $\alpha$)

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

Definition (Hybrid game $\alpha$)

$\alpha, \beta ::= x := e \mid ?Q \mid x' = f(x) \& Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \alpha^d$
Example: Push-around Cart

Hybrid systems can't say that $a$ is Angel's choice and $d$ is Demon's. Only that there are choices.
Example: Push-around Cart

\[(a := 1 \cup a := -1); (d := 1 \cup d := -1)^d; \{x' = v, v' = a + d\}\]
Example: Push-around Cart

\[
\left( (a := 1 \cup a := -1) ; (d := 1 \cup d := -1)^d ; \{ x' = v, v' = a + d \} \right)^* \\
\left( (d := 1 \cup d := -1)^d ; (a := 1 \cup a := -1) ; \{ x' = v, v' = a + d \} \right)^*
\]
Example: Push-around Cart

\[
((a := 1 \cup a := -1); (d := 1 \cap d := -1); \{x' = v, v' = a + d\})^* 
\]

\[
((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* 
\]
Example: Push-around Cart

\[\left(\left(a := 1 \cup a := -1\right); \left(d := 1 \cap d := -1\right); \{x' = v, v' = a + d\}\right)^*\]

\[\left(\left(d := 1 \cap d := -1\right); \left(a := 1 \cup a := -1\right); \{x' = v, v' = a + d\}\right)^*\]

\[\text{HP} \left(\left(d := 1 \cup d := -1\right); \left(a := 1 \cup a := -1\right); \{x' = v, v' = a + d\}\right)^*\]
Example: Push-around Cart

\[ ((a := 1 \cup a := -1); (d := 1 \cap d := -1); \{x' = v, v' = a + d\})^* \]

\[ ((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \]

HP \[ ((d := 1 \cup d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \]

Hybrid systems can’t say that \( a \) is Angel’s choice and \( d \) is Demon’s. Only that there are choices.
Definition (Hybrid game $\alpha$)

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

Definition (Hybrid game $\alpha$)

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

Definition (dGL Formula $P$)

$$
P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \& Q \mid \forall x P \mid \exists x P \mid \langle \alpha \rangle P \mid [\alpha] P
$$
Differential Game Logic: Syntax

Definition (Hybrid game $\alpha$)

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

Definition (dGL Formula $P$)

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

Discrete Assign Test Game Differential Equation Choice Game Seq. Game Repeat Game

All Reals Some Reals
Differential Game Logic: Syntax

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

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

### Definition (dGL Formula $P$)

$$
P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x P \mid \exists x P \mid \langle \alpha \rangle P \mid [\alpha]P
$$
Differential Game Logic: Syntax

Definition (Hybrid game $\alpha$)

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

Definition (dGL Formula $P$)

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

Discrete Assign

Test Game

Differential Equation

Choice Game

Seq. Game

Repeat Game

Dual Game

All Reals

Some Reals

Angel Wins
Differential Game Logic: Syntax

Definition (Hybrid game $\alpha$)

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

Definition (dGL Formula $P$)

$$P, Q ::= e \geq \bar{e} \mid \neg P \mid P \& Q \mid \forall x \, P \mid \exists x \, P \mid \langle \alpha \rangle P \mid [\alpha] P$$
Simple Examples

$$\langle (x := x + 1; (x' = 1)^d \cup x := x - 1)^* \rangle (0 \leq x < 1)$$

$$\langle (x := x + 1; (x' = 1)^d \cup (x := x - 1 \cap x := x - 2))^* \rangle (0 \leq x < 1)$$
Simple Examples

\[ \vdash \langle (x := x + 1; (x' = 1)^d \cup x := x - 1)^* \rangle (0 \leq x < 1) \]

\[ \langle (x := x + 1; (x' = 1)^d \cup (x := x - 1 \cap x := x - 2))^* \rangle (0 \leq x < 1) \]
Simple Examples

\[
\models \langle (x := x + 1; (x' = 1)^d \cup x := x - 1)^* \rangle \ (0 \leq x < 1)
\]

\[\not\models \langle (x := x + 1; (x' = 1)^d \cup (x := x - 1 \land x := x - 2))^* \rangle \ (0 \leq x < 1)\]
Example: Push-around Cart

\[ v \geq 1 \rightarrow \]

\[ [((d := 1 \cup d := -1)^d; (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^*] v \geq 0 \]
Example: Push-around Cart

\[ x \]  
\[ v \]

\[ \vdash v \geq 1 \rightarrow \]
\[ \left[ \left( d := 1 \cap d := -1 \right) ; \left( a := 1 \cup a := -1 \right) ; \{ x' = v , v' = a + d \} \right]^* \] \[ v \geq 0 \]
Example: Push-around Cart

\[ \models v \geq 1 \rightarrow \]

\[
\left[ ((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \right] v \geq 0
\]

\[ x \geq 0 \land v \geq 0 \rightarrow \]

\[
\left[ ((d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\})^* \right] x \geq 0
\]

\(d\) before \(a\) can compensate
Example: Push-around Cart

\[d \text{ before } a \text{ can compensate}\]

\[
\models v \geq 1 \rightarrow \\
\left[\left(d := 1 \land d := -1\right); \left(a := 1 \lor a := -1\right); \{x' = v, v' = a + d\}\right]^* v \geq 0
\]

\[
\models x \geq 0 \land v \geq 0 \rightarrow \\
\left[\left(d := 1 \land d := -1\right); \left(a := 1 \lor a := -1\right); \{x' = v, v' = a + d\}\right]^* x \geq 0
\]
Example: Push-around Cart

\[ \models v \geq 1 \rightarrow d \text{ before } a \text{ can compensate} \]

\[ \left[ (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\} \right]^* v \geq 0 \]

\[ x \geq 0 \quad \rightarrow \]

\[ \left\langle (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{x' = v, v' = a + d\} \right]^* \right\rangle x \geq 0 \]
Example: Push-around Cart

\[\begin{align*}
\models v \geq 1 \rightarrow & \quad d \text{ before } a \text{ can compensate} \\
\left[\left(\left(d := 1 \land d := -1\right); \left(a := 1 \cup a := -1\right); \{x' = v, v' = a + d\}\right)^*\right] v \geq 0 \\
\models x \geq 0 \rightarrow & \quad \text{boring by skip} \\
\left\langle\left(\left(d := 1 \land d := -1\right); \left(a := 1 \cup a := -1\right); \{x' = v, v' = a + d\}\right)^*\right\rangle x \geq 0
\end{align*}\]
Example: Push-around Cart

$d$ before $a$ can compensate

\[ v \geq 1 \rightarrow \]

\[ \left[ \left( \left( d := 1 \cap d := -1 \right); \left( a := 1 \cup a := -1 \right); \{ x' = v, v' = a + d \} \right) \right]^* \] $v \geq 0$

\[ \left\langle \left( \left( d := 1 \cap d := -1 \right); \left( a := 1 \cup a := -1 \right); \{ x' = v, v' = a + d \} \right) \right\rangle \] $x \geq 0$
Example: Push-around Cart

\[ v \geq 1 \rightarrow d \text{ before } a \text{ can compensate} \]

\[ [(d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\}]^* v \geq 0 \]

\[ \not\exists \]

\[ \langle [(d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\}]^* \rangle x \geq 0 \]
Example: Push-around Cart

\[ \vdash v \geq 1 \rightarrow \]
\[ \left[ \left( (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{ x' = v, v' = a + d \} \right)^* \right] v \geq 0 \]

\[ \not\vdash \]
\[ \left\langle \left( (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{ x' = v, v' = a + d \} \right)^* \right\rangle x \geq 0 \]
\[ \left\langle \left( (d := 1 \cap d := -1); (a := 2 \cup a := -2); \{ x' = v, v' = a + d \} \right)^* \right\rangle x \geq 0 \]

\(d\) before \(a\) can compensate

counterstrategy \(d := -1\)
Example: Push-around Cart

\[ v \geq 1 \Rightarrow \quad \text{d before a can compensate} \]

\[ \left[ \left( d := 1 \cap d := -1 \right); \left( a := 1 \cup a := -1 \right); \left\{ x' = v, v' = a + d \right\} \right]^* \] \quad v \geq 0

\[ \nv

\[ \n\]

\[ \left( d := 1 \cap d := -1 \right); \left( a := 1 \cup a := -1 \right); \left\{ x' = v, v' = a + d \right\} \right]^* \] \quad x \geq 0

\[ \n\]

\[ \left\{ x' = v, v' = a + d \right\} \quad t := 0; \quad \left\{ t' = 1 \right\} \quad t \leq 1 \]

\[ \left( d := 1 \cap d := -1 \right); \left( a := 2 \cup a := -2 \right); \left\{ x' = v, v' = a + d \right\} \right]^* \] \quad x \geq 0
Example: Push-around Cart

\[ v \geq 1 \rightarrow \]

\[ \lbrack (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{ x' = v, v' = a + d \} \rbrack^* \] \( v \geq 0 \)

\[ \not\models \]

counterstrategy \( d := -1 \)

\[ \langle (d := 1 \cap d := -1); (a := 1 \cup a := -1); \{ x' = v, v' = a + d \} \rangle^* \] \( x \geq 0 \)

\[ \models \langle (d := 1 \cap d := -1); (a := 2 \cup a := -2); \{ x' = v, v' = a + d \} \rangle^* \] \( x \geq 0 \)

\[ \langle (d := 2 \cap d := -2); (a := 2 \cup a := -2); t := 0; \{ x' = v, v' = a + d, t' = 1 \& t \leq 1 \} \rangle^* \] \( x^2 \geq 100 \)
Example: Push-around Cart

\[ x \rightarrow \] \[ v \rightarrow \]

\[ \vdash v \geq 1 \rightarrow \]
\[ \left[ \left( \left( d := 1 \cap d := -1 \right) ; \left( a := 1 \cup a := -1 \right) ; \{ x' = v, v' = a + d \} \right)^* \right] v \geq 0 \]

\[ \not\vdash \left( \left( d := 1 \cap d := -1 \right) ; \left( a := 1 \cup a := -1 \right) ; \{ x' = v, v' = a + d \} \right)^* \right) x \geq 0 \]

\[ \vdash \left( \left( d := 1 \cap d := -1 \right) ; \left( a := 2 \cup a := -2 \right) ; \{ x' = v, v' = a + d \} \right)^* \right) x \geq 0 \]

\[ \vdash \left( \left( d := 2 \cap d := -2 \right) ; \left( a := 2 \cup a := -2 \right) ; a := d \text{ then } a := 2 \text{ sign } v \right) \]
\[ t := 0 ; \{ x' = v, v' = a + d, t' = 1 \text{ & } t \leq 1 \} \right)^* \right) x^2 \geq 100 \]
Example: WALL·E and EVE Robot Dance

(EVE at $e$ plays Angel’s part controlling $g$
WALL·E at $w$ plays Demon’s part controlling $u$

\[
(w - e)^2 \leq 1 \land v = f \rightarrow \\
\left\langle ((u := 1 \cap u := -1); \right.
\left(g := 1 \cup g := -1); \right.
\left.t := 0; \right.
\left\{ w' = v, v' = u, e' = f, f' = g, t' = 1 \& t \leq 1 \right\}^d
\right) \times (w - e)^2 \leq 1
\]
Example: WALL·E and EVE Robot Dance and the World

\[(w - e)^2 \leq 1 \land v = f \rightarrow \]
\[\langle ((u := 1 \land u := -1); \\
(g := 1 \lor g := -1); \\
t := 0; \\
\{w' = v, v' = u, e' = f, f' = g, t' = 1 \land t \leq 1 \}^d \rangle \times (w - e)^2 \leq 1 \]

EVE at \(e\) plays Angel’s part controlling \(g\)

WALL·E at \(w\) plays Demon’s part controlling \(u\) and world time
Example: EVE and WALL·E Robot Dance

\[(w - e)^2 \leq 1 \land v = f \rightarrow \]

\[
\left[ \left( (u := 1 \cap u := -1); \\
(g := 1 \cup g := -1); \right) \\
t := 0; \right]
\]

\[
\left\{ w' = v, v' = u, e' = f, f' = g, t' = 1 \land t \leq 1 \right\}
\]

\[\times \] \[ (w - e)^2 > 1 \]

WALL·E at \( w \) plays Demon’s part controlling \( u \) and world time

EVE at \( e \) plays Angel’s part controlling \( g \)
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

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

\[ x < 0 \land v > 0 \land y = g \rightarrow \]

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

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

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

Goalie’s Secret

\[
\left(\frac{x}{v}\right)^2 (u - w)^2 \leq 1 \land
\]

\[
x < 0 \land v > 0 \land y = g \rightarrow
\]

\[
\langle (w := +w \cap w := -w); (u := +u \cup u := -u); \{x' = v, y' = w, g' = u\} \rangle^* \mid x^2 + (y - g)^2 \leq 1
\]
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
Definition (Hybrid game $\alpha$: operational semantics)

\[
x := e
\]

\[
\omega_x^\omega[e]
\]
Definition (Hybrid game $\alpha$: operational semantics)

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

\[ \omega \]

\[ \varphi(0) \quad \varphi(t) \quad \varphi(r) \]
Definition (Hybrid game $\alpha$: operational semantics)

\[ \omega \xrightarrow{?Q} \omega \in [Q] \]

Diagram:
- **Node**: $\omega$
- **Label**: $?Q \omega \in [Q]$
Definition (Hybrid game $\alpha$: operational semantics)

\[ \omega \vdash e \]

\[ \omega \vdash x \]

\[ \omega \vdash x' = f(x) \& \varphi(r) \]

\[ \varphi(t) \]

\[ \varphi(0) \]

\[ \alpha \cup \beta \]

\[ \text{left} \]

\[ \text{right} \]

\[ \text{repeat} \]

\[ \text{stop} \]
Definition (Hybrid game $\alpha$: operational semantics)

$\omega := \omega \[ \[ e \] \] x$

$t_1 \xrightarrow{\alpha} t_i \xrightarrow{\alpha} t_\lambda$

$r_1 \xrightarrow{\beta} r_i \xrightarrow{\beta} r_\lambda$

$\alpha; \beta$

André Platzer (CMU)  
LFCPS/14: Hybrid Systems & Games
Definition (Hybrid game $\alpha$: operational semantics)

$\alpha^*$

André Platzer (CMU)

LFCPS/14: Hybrid Systems & Games

LFCPS/14 21 / 24
Definition (Hybrid game $\alpha$: operational semantics)
\((x := 0 \cap x := 1)^* x = 0\)
\[(x := 0 \land x := 1)^* x = 0\]

\[\text{false unless } x = 0\]
\( \langle x' = 1^d; x := 0 \rangle^* x = 0 \)

\( \langle x := 0; x' = 1^d \rangle^* x = 0 \)

\( \langle x := 0 \cap x := 1 \rangle^* x = 0 \)

\( \overset{wfd}{\sim} \text{false unless } x = 0 \)
true

\[ \langle (x' = 1^d; x := 0)^* \rangle x = 0 \]

\[ \langle (x := 0; x' = 1^d)^* \rangle x = 0 \]

\[ \langle (x := 0 \land x := 1)^* \rangle x = 0 \]

\[ \rightsquigarrow_{wfd} \text{false unless } x = 0 \]
Well-defined games can’t be postponed forever!
Outline

1. Learning Objectives
2. Motivation
3. A Gradual Introduction to Hybrid Games
   - Choices & Nondeterminism
   - Control & Dual Control
   - Demon’s Derived Controls
4. Differential Game Logic
   - Syntax of Hybrid Games
   - Syntax of Differential Game Logic Formulas
   - Examples
   - Push-around Cart
   - Robot Dance
   - Example: Robot Soccer
5. An Informal Operational Game Tree Semantics
6. Summary
Differential Game Logic: Syntax

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

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

**Definition (dGL Formula $P$)**

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \& Q \mid \forall x P \mid \exists x P \mid \langle \alpha \rangle P \mid [\alpha] P$$
Summary

Differential game logic

\[ dGL = GL + HG = dL + d \]

- Differential game logic
- Logic for hybrid games
- Compositional PL + logic
- Discrete + continuous + adversarial
- Operational semantics (informally)

Next chapter

1. Formal semantics
Example: Robot Factory
Example: Robot Factory Decentralized Automation

Model
- \((x, y)\) robot coordinates
- \((v_x, v_y)\) velocities
- conveyor belts may instantaneously increase robot’s velocity by \((c_x, c_y)\)

Primary objectives of the robot
- Leave within time \(\varepsilon\)
- Never leave outer

Challenges
- Distributed, physical environment
- Possibly conflicting secondary objectives
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( \texttt{true} \cup \left( (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0 \right) \quad \text{// belt} \\
\cup \left( (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0 \right) \right); 
\]

\*
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\big( true \cup \left( (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0 \right) \big) \quad \text{// belt}
\]

\[
\cup \left( (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0 \right) \big)
\]

\[
( a_x := \ast; \ ?(-A \leq a_x \leq A) ;
\]

\[
a_y := \ast; \ ?(-A \leq a_y \leq A) ; \quad \text{// “independent” robot acceleration}
\]

\[
t_s := 0 \big) \quad \text{d} ;
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( \text{true} \cup \left( (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0 \right) \right) \quad \text{// belt}
\]
\[
\cup \left( (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0 \right) \right); \\
(a_x := *; \ -A \leq a_x \leq A; \\
\ a_y := *; \ -A \leq a_y \leq A; \quad \text{// “independent” robot acceleration}
\]
\[
\left( x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \land t_s \leq \epsilon \right);
\]

\[
\ast
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( \text{true} \cup \left( (x < e_x \land y < e_y \land \text{eff}_1 = 1) ; \ v_x := v_x + c_x ; \ \text{eff}_1 := 0 \right) \right) \qquad \text{// belt}
\]

\[
\cup \left( (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1) ; \ v_y := v_y + c_y ; \ \text{eff}_2 := 0 \right) ) ;
\]

\[
(a_x := * ; \ ?(-A \leq a_x \leq A) ; \ a_y := * ; \ ?(-A \leq a_y \leq A) ; \quad \text{// “independent” robot acceleration}
\]

\[
t_s := 0 )^d ;
\]

\[
\left( x' = v_x , y' = v_y , v'_x = a_x , v'_y = a_y , t' = 1 , t'_s = 1 \ & \ t_s \leq \varepsilon \right) ;
\]

\[
\cap \left( (a_x v_x \leq 0 \land a_y v_y \leq 0) \right)^d ; \quad \text{// brake}
\]

\[
\text{if } v_x = 0 \text{ then } a_x := 0 \text{ fi} ; \quad \text{// per direction: no time lock}
\]

\[
\text{if } v_y = 0 \text{ then } a_y := 0 \text{ fi} ;
\]

\[
\left( x' = v_x , y' = v_y , v'_x = a_x , v'_y = a_y , t' = 1 , t'_s = 1 \ & \ t_s \leq \varepsilon \land a_x v_x \leq 0 \land a_y v_y \leq 0 \right) )^* \]
Proposition (Robot stays in □)

\[ \models (x = y = 0 \land v_x = v_y = 0 \land \text{Controllability Assumptions}) \rightarrow [RF](x \in [l_x, r_x] \land y \in [l_y, r_y]) \]

Proposition (Stays in □ and leaves on time)

\[ RF|_x: RF \text{ projected to the } x\text{-axis} \]

\[ \models (x = 0 \land v_x = 0 \land \text{Controllability Assumptions}) \rightarrow [RF|_x](x \in [l_x, r_x] \land (t \geq \varepsilon \rightarrow x \geq x_b)) \]
André Platzer.
*Logical Foundations of Cyber-Physical Systems.*
Springer, Cham, 2018.
doi:10.1007/978-3-319-63588-0.

André Platzer.
Differential game logic.

André Platzer.
Logics of dynamical systems.
In LICS [12], pages 13–24.

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 uniform substitution calculus for differential dynamic logic.

André Platzer.
Differential hybrid games.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [12], pages 541–550.
doi:10.1109/LICS.2012.64.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.

Special issue for selected papers from CSL’10.

**André Platzer.**
Stochastic differential dynamic logic for stochastic hybrid programs.
In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*,
doi:10.1007/978-3-642-22438-6_34.

**Jan-David Quesel and André Platzer.**
Playing hybrid games with KeYmaera.
In Bernhard Gramlich, Dale Miller, and Ulrike Sattler, editors, *IJCAR*,
doi:10.1007/978-3-642-31365-3_34.