14: Hybrid Systems & Games
Logical Foundations of Cyber-Physical Systems

André Platzer
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

CT

M&C  CPS

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

multi-agent state change
CPS semantics
reflections on choices
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
CPS Analysis: Robot Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- Discrete dynamics (control decisions)
- Continuous dynamics (differential equations)
CPS Analysis: Robot Control

Challenge (Hybrid Systems)

Fixed rule describing state evolution with both

- **Discrete dynamics** (control decisions)
- **Continuous dynamics** (differential equations)
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 $\|$ vs. Demon $\|$)

\[ \begin{align*}
    a & \quad t \\
    v & \quad t \\
p & \quad t
\end{align*} \]
Challenge (Hybrid Games)

Game rules describing play evolution with

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

Game rules describing play evolution with

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

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

\[
\begin{align*}
    d_x & \quad d_y \\
    d & \quad t
\end{align*}
\]
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**
\[ QdL = FOL + DL + QHP \]
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 $\alpha$)

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

### 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: Syntax

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

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

#### 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 (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
### 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
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 $P$)

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

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

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

$[\alpha_1]\langle\alpha_2[\alpha_3]\langle\alpha_4\rangle P$ only fixed interaction depth
Let Angel be one player
Let Angel be one player

Let Demon be another player

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

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

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\cup$</td>
<td>choice</td>
</tr>
<tr>
<td>$\ast$</td>
<td>repeat</td>
</tr>
<tr>
<td>$x' = f(x)$</td>
<td>evolve</td>
</tr>
<tr>
<td>$?Q$</td>
<td>challenge</td>
</tr>
</tbody>
</table>

### Demon Ops

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\cap$</td>
<td>choice</td>
</tr>
<tr>
<td>$\times$</td>
<td>repeat</td>
</tr>
<tr>
<td>$x' = f(x)^d$</td>
<td>evolve</td>
</tr>
<tr>
<td>$?Q^d$</td>
<td>challenge</td>
</tr>
</tbody>
</table>

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

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

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

Duality operator $^d$ passes control between players

André Platzer (CMU)
Game Operators

Diamond Angel Ops

\[ \begin{align*}
&\cup \quad \text{choice} \\
&\ast \quad \text{repeat} \\
&x' = f(x) \quad \text{evolve} \\
&?Q \quad \text{challenge}
\end{align*} \]

Diamond Demon Ops

\[ \begin{align*}
&\cap \quad \text{choice} \\
&\times \quad \text{repeat} \\
&x' = f(x)^d \quad \text{evolve} \\
&?Q^d \quad \text{challenge}
\end{align*} \]

Duality operator $^d$ passes control between players

André Platzer (CMU)
Game Operators

\[ \begin{align*}
\text{Angel Ops} & : \quad \bigcup & \text{choice} \\
& & \times & \text{repeat} \\
& x' = f(x) & \text{evolve} \\
Q & ? & \text{challenge} \\
\text{Demon Ops} & : \quad \bigcap & \text{choice} \\
& & \times & \text{repeat} \\
& x' = f(x)^d & \text{evolve} \\
Q^d & ? & \text{challenge} \\
\end{align*} \]

Duality operator \( d \) passes control between players

André Platzer (CMU)
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

\[
\text{if}(Q) \alpha \text{ else } \beta \equiv \\
\text{while}(Q) \alpha \equiv \\
\alpha \cap \beta \equiv \\
\alpha \times \equiv \\
(x' = f(x) \& Q)^d \\
(x := e)^d \\
?Q^d \\
?Q
\]
Definable Game Operators

\[ \begin{align*}
\text{Angel Ops} & : \quad \bigcup \quad \text{choice} \\
& \quad \ast \quad \text{repeat} \\
& \quad x' = f(x) \quad \text{evolve} \\
& \quad ?Q \quad \text{challenge} \\
\text{Demon Ops} & : \quad \bigcap \quad \text{choice} \\
& \quad \times \quad \text{repeat} \\
& \quad x' = f(x)^d \quad \text{evolve} \\
& \quad ?Q^d \quad \text{challenge}
\end{align*} \]

\[ \begin{align*}
\text{if}(Q) \alpha \text{ else } \beta & \equiv (?Q; \alpha) \cup (?\neg Q; \beta) \\
\text{while}(Q) \alpha & \equiv \\
\alpha \cap \beta & \equiv \\
\alpha \times & \equiv \\
(x' = f(x) \& Q)^d & x' = f(x) \& Q \\
(x := e)^d & x := e \\
?Q^d & ?Q
\end{align*} \]
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

**Quantifiers**
- $\text{if}(Q) \alpha \text{ else } \beta \equiv (Q; \alpha) \cup (\neg Q; \beta)$
- $\text{while}(Q) \alpha \equiv (Q; \alpha)\ast; ?\neg Q$
- $\alpha \cap \beta \equiv$
- $\alpha \times \equiv$
- $(x' = f(x) \& Q)^d$
- $x' = f(x) \& Q$
- $(x := e)^d$
- $x := e$
- $?Q^d$
- $?Q$
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

if\((Q)\alpha\) else \(\beta\) \(\equiv\) \((?Q;\alpha) \cup (?\neg Q;\beta)\)

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

\(\alpha \cap \beta \equiv\)

\(\alpha \times \equiv\)

\((x' = f(x) \& Q)^d\)

\((x := e)^d\)

\(?Q^d\)

\(?Q\)
Definable Game Operators

**Angel Ops**
- $\bigcup$
- $\ast$
- $x' = f(x)$
- $?Q$

**Demon Ops**
- $\bigcap$
- $\times$
- $x' = f(x)^d$
- $?Q^d$

**Game Operators**

- $\alpha \equiv (\alpha^{d} \cup \beta^{d})^{d}$
- $\alpha^{\times} \equiv (x' = f(x) \& Q)^{d}$
- $(x := e)^{d} \equiv x := e$
- $?Q^{d} \equiv ?Q$
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

\[
\begin{align*}
\text{if}(Q) \alpha \text{ else } \beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta) \\
\text{while}(Q) \alpha \equiv (?Q; \alpha)^*; ?\neg Q \\
\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d \\
\alpha^\times \equiv ((\alpha^d)^*)^d \\
(x' = f(x) \& Q)^d & \quad x' = f(x) \& Q \\
(x := e)^d & \quad x := e \\
?q^d & \quad ?Q
\end{align*}
\]
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

if $(Q) \alpha$ else $\beta \equiv (?Q; \alpha) \cup (?\neg Q; \beta)$

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

$\alpha \cap \beta \equiv (\alpha^d \cup \beta^d)^d$

$\alpha^\times \equiv ((\alpha^d)^*)^d$

$(x' = f(x) \& Q)^d \not\equiv x' = f(x) \& Q$

$(x := e)^d \quad x := e$

$?Q^d \quad ?Q$
### 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

\[
\begin{align*}
\text{if}(Q) \alpha & \text{ else } \beta \equiv (Q; \alpha) \cup (\neg Q; \beta) \\
\text{while}(Q) \alpha & \equiv (Q; \alpha)^*; \neg Q \\
\alpha \cap \beta & \equiv (\alpha^d \cup \beta^d)^d \\
\alpha \times & \equiv ((\alpha^d)^*)^d \\
(x' = f(x) & Q)^d & \neq x' = f(x) & Q \\
(x := e)^d & \equiv x := e \\
?Q^d & \equiv ?Q
\end{align*}
\]
Definable Game Operators

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

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

\[
\begin{align*}
\text{if}(Q)\; \alpha \text{ else } \beta & \equiv (\ ?Q ; \alpha ) \cup ( \neg Q ; \beta ) \\
\text{while}(Q)\; \alpha & \equiv (\ ?Q ; \alpha )^* ; \neg Q \\
\alpha \cap \beta & \equiv (\alpha^d \cup \beta^d)^d \\
\alpha^\times & \equiv ((\alpha^d)^*)^d \\
(x' = f(x) \& Q)^d & \neq x' = f(x) \& Q \\
(x := e)^d & \equiv x := e \\
?Q^d & \neq ?Q
\end{align*}
\]
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 $\alpha$)

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

Discrete Assign  Test Game  Differential Equation  Choice Game  Seq. Game  Repeat Game
Hybrid Games: Syntax

Definition (Hybrid game $\alpha$)

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

- Discrete Assign
- Test Game
- Differential Equation
- Choice Game
- Seq. Game
- Repeat Game
- Dual Game
Example: Push-around Cart

\[
\begin{align*}
&d \in \{1, -1\};
&v \in \{1, -1\};
&x' = v, v' = a + d
\end{align*}
\]

Hybrid systems can't say that \(a\) is Angel's choice and \(d\) is Demon's choice.
Example: Push-around Cart

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

\[
\left( (a := 1 \cup a := -1) ; \left( d := 1 \cup d := -1 \right)^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

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

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

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

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

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

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

Hybrid systems can't say that \(a\) is Angel's choice and \(d\) is Demon's.
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 $P$)

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

Definition (Hybrid game $\alpha$)

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

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

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

Definition (dGL Formula \( P \))

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

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

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

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

All
Reals
Some
Reals
Angel
Wins
Demon
Wins

André Platzer (CMU)  
LFCPS/14: Hybrid Systems & Games
Simple Examples

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\[ \vdash v \geq 1 \rightarrow \]

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

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

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

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

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

boring by skip
Example: Push-around Cart

$\vdash v \geq 1 \rightarrow \quad \text{d before a can compensate}$

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

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

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

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

\[ \not\exists \]

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

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

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

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

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

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

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

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

\[ x \geq 0 \]

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

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

\[ x \geq 0 \]

\[ ((d := 2 \cap d := -2); (a := 2 \cup a := -2); \{x' = v, v' = a + d\})^* \]

\[ x \geq 0 \]
Example: Push-around Cart

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

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

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

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

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

\[
\begin{align*}
\models v \geq 1 \rightarrow & \quad d \text{ before } a \text{ can compensate} \\
& \quad [((d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\})^*] v \geq 0 \\
\not\models & \quad \langle((d := 1 \land d := -1); (a := 1 \lor a := -1); \{x' = v, v' = a + d\})^*\rangle x \geq 0 \\
\models & \quad \langle((d := 1 \land d := -1); (a := 2 \lor a := -2); \{x' = v, v' = a + d\})^*\rangle x \geq 0 \\
\models & \quad \langle((d := 2 \land d := -2); (a := 2 \lor a := -2); a := d \text{ then } a := 2 \text{ sign } v \\
& \quad t := 0; \{x' = v, v' = a + d, t' = 1 \& t \leq 1\})^*\rangle x^2 \geq 100
\end{align*}
\]
Example: WALL•E and EVE Robot Dance

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

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

WALL•E at \( w \) plays Demon’s part controlling \( u \)
Example: WALL•E and EVE Robot Dance and the World

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

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

WALL•E at \(w\) plays Demon’s part controlling \(u\) and world time
(\(w - e\))^2 \leq 1 \land v = f \rightarrow \\
\left[\left(\left(u := 1 \cap u := -1\right); \right.
\left.\left(g := 1 \cup g := -1\right);\right.
\right.

\left.t := 0;\right.
\)

\left\{ w' = v, v' = u, e' = f, f' = g, t' = 1 \& t \leq 1 \right\} \\
) \times \left(\left(\left(w - e\right)^2 > 1 \right.
\right.

WALL•E at \(w\) plays Demon’s part controlling \(u\) and world time \(t\) 

EVE at \(e\) plays Angel’s part controlling \(g\)
Example: Goalie in Robot Soccer

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

\[(x, y) \rightarrow (v, +w)\]

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

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

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

\[
\left( \frac{x}{v} \right)^2 (u - w)^2 \leq 1 \land \\
x < 0 \land v > 0 \land y = g \rightarrow \\
\left\langle (w := +w \cap w := -w); \right. \\
\left. ((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}^*) \right\rangle x^2 + (y - g)^2 \leq 1
\]
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)

\[ x := e \]
Definition (Hybrid game $\alpha$: operational semantics)

$$x' = f(x) \& Q$$
Definition (Hybrid game $\alpha$: operational semantics)
Definition (Hybrid game $\alpha$: operational semantics)

$$\omega := \omega$$

$$\omega := \omega \omega \alpha$$

$$\omega := \omega \omega \alpha \phi (\tau)$$

$$\omega := \omega \phi (\tau)$$

$$\omega := \omega \omega \alpha \ast$$

$$\omega := \omega \ast$$

$$\omega := \omega \ast$$

$$\omega := \omega \ast$$

$$\omega := \omega \ast$$

$$\omega := \omega \ast$$

André Platzer (CMU)
Definition (Hybrid game $\alpha$: operational semantics)
Definition (Hybrid game $\alpha$: operational semantics)
Definition (Hybrid game $\alpha$: operational semantics)

$\omega := e$  $\omega$  $\omega$  $\omega$

$x := e$  $\omega$  $\omega$

$x' = f(x)$  $\&$  $Q$  $\phi(r)$

$\phi(t)$  $\phi(0)$

$?Q\omega ?Q\omega \in \llbracket Q \rrbracket$  $\omega$  $\alpha \cup \beta$

$\lambda\alpha$  $\lambda\beta$  $\lambda\alpha$  $\lambda\beta$

$\alpha^*$  $\alpha^*$  $\alpha^*$  $\alpha^*$

$\text{repeat}$  $\text{stop}$  $\text{repeat}$  $\text{stop}$

André Platzer (CMU)
\[(x := 0 \cap x := 1)^* x = 0\]
\[(x := 0 \cap x := 1)^* x = 0\]

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

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

\[ \langle x := 0 \cap x := 1 \rangle^* x = 0 \]

wfd \[\mapsto\] false unless \( x = 0 \)
\[ \langle (x' = 1^d; x := 0)^* \rangle x = 0 \]
\[ \langle (x := 0; x' = 1^d)^* \rangle x = 0 \]
\[ \langle (x := 0 \land x := 1)^* \rangle x = 0 \]

\[ \text{wfd} \] \[ \sim \] false unless \( x = 0 \)
Filibusters & The Significance of Finitude

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

wfd 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 $\alpha$)

$$\alpha, \beta ::= \begin{array}{l}
    x := e \\
    ?Q \\
    x' = f(x) \& Q \\
    \alpha \cup \beta \\
    \alpha; \beta \\
    \alpha^* \\
    \alpha^d
\end{array}$$

Definition (dGL Formula $P$)

$$P, Q ::= \begin{array}{l}
    e \geq \bar{e} \\
    \neg P \\
    P \& Q \\
    \forall x P \\
    \exists x P \\
    \langle \alpha \rangle P \\
    [\alpha] P
\end{array}$$
Differential game logic

\[ \text{dGL} = \text{GL} + \text{HG} = \text{dL} + d \]

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

Next chapter

1. Formal semantics
Outline

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)

\[
\left( ?true \cup ( ?(x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0 ) \quad \text{// belt}
\right.
\]
\[
\cup ( ?(e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0 ) ) ;
\]
Example (Robot-Demon vs. Angel-Factory Environment)

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

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

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

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

\[
t_s := 0 )^d ;
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( \begin{array}{l}
\text{(true)} \cup (x < e_x \land y < e_y \land \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0) \quad \text{// belt} \\
\cup (e_x \leq x \land y \leq f_y \land \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0) \\
\end{array} \right); \\
\begin{array}{l}
\text{a}_x := *; \ ?(-A \leq a_x \leq A); \\
\text{a}_y := *; \ ?(-A \leq a_y \leq A); \quad \text{// “independent” robot acceleration} \\
t_s := 0 \big)^d; \\
\end{array}
\]

\[
\begin{array}{l}
(x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \& t_s \leq \varepsilon ); \\
\end{array}
\]

\[
\left( \begin{array}{l}
\end{array} \right)^* 
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( (\text{true} \cup (x < e_x \wedge y < e_y \wedge \text{eff}_1 = 1); \ v_x := v_x + c_x; \ \text{eff}_1 := 0) \quad \text{// belt} \\
\cup (e_x \leq x \wedge y \leq f_y \wedge \text{eff}_2 = 1); \ v_y := v_y + c_y; \ \text{eff}_2 := 0) \right);
\]
\[
(a_x := *; \ (-A \leq a_x \leq A); \\
\quad a_y := *; \ (-A \leq a_y \leq A); \quad \text{// “independent” robot acceleration} \\
\quad t_s := 0 )^d;
\]
\[
\left( (x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \wedge t_s \leq \varepsilon ) \right);
\]
\[
\cap (a_x v_x \leq 0 \wedge a_y v_y \leq 0)^d; \quad \text{// brake} \\
\quad \text{if } v_x = 0 \text{ then } a_x := 0 \text{ fi}; \quad \text{// per direction: no time lock} \\
\quad \text{if } v_y = 0 \text{ then } a_y := 0 \text{ fi}; \\
\left( (x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \wedge t_s \leq \varepsilon ) \right)^*.
\]
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)

\[\models (x = 0 \land v_x = 0 \land \text{Controllability Assumptions}) \rightarrow [RF|_x](x \in [l_x, r_x] \land (t \geq \varepsilon \rightarrow x \geq x_b))\]
André Platzer.

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