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
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
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)
CPS Analysis: Robot Control

Challenge (Games)

Game rules describing play evolution with both

- Angelic choices (player $\Diamond$ Angel)
- Demonic choices (player $\Box$ 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 $\Box$)
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 ☐)

\[ t \quad a \quad \omega \quad d_x \quad d_y \]
Challenge (Hybrid Games)

Game rules describing play evolution with

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

André Platzer (CMU)
CPSs are Multi-Dynamical Systems

CPS Dynamics

CPS are characterized by multiple facets of dynamical systems.

CPS Compositions

CPS combines multiple simple dynamical effects.

Descriptive simplification

Tame Parts

Exploiting compositionality tames CPS complexity.

Analytic simplification

André Platzer (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**
  \[ QdL = FOL + DL + QHP \]
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 $P$)

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

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 $P$)

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

- All Reals
- Some Reals
- All Runs
- Some Runs

Andre Platzer (CMU)
Differential Dynamic Logic dL: Nondeterminism

Definition (Hybrid program α)

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

Definition (dL Formula P)

\[ e \geq \tilde{e} \mid \neg P \mid P \& Q \mid \forall x \ P \mid \exists x \ P \mid [α]P \mid ⟨α⟩P \]

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 $\alpha$)

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

Definition (dL Formula $P$)

\[
\begin{align*}
e &\geq \bar{e} \mid \neg P \mid P \land Q \mid \forall x \ P \mid \exists x \ P \mid [\alpha]P \mid \langle \alpha \rangle P
\end{align*}
\]
**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
- Some Choice
- All choices resolved in one way
- Differential Equation
- Nondet. Choice
- Nondet. Repeat

André Platzer (CMU)

LFCPS/14: Hybrid Systems & Games
**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 $P$)

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

Modality decides the mode: help/hurt

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

Diamond Angel Ops

- Union \( \bigcup \)
- Choice
- Repeat \( * \)
- \( x' = f(x) \)
- Evolve
- \( ?Q \)
- Challenge

Diamond Demon Ops

- Intersection \( \bigcap \)
- Choice
- Repeat \( \times \)
- \( x' = f(x)^d \)
- Evolve
- \( ?Q^d \)
- Challenge

Duality operator \( d \) passes control between players

Diagram of a chessboard with pieces arranged, illustrating the game operators.
Game Operators

Diamond Angel Ops

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

Duality operator $^d$ passes control between players

Diamond Demon Ops

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

△ 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

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

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

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

Andre Platzer (CMU)
Definable Game Operators

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

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

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

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

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

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

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

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

Axel Ops

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

Demon Ops

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

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

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

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

\( \alpha^\times \equiv \)

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

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

\( ?Q^d \quad ?Q \)
Definable Game Operators

**Diamond Angel Ops**

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

**Diamond Demon Ops**

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

**Logical Expressions**

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

**Diamond Angel Ops**

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

**Diamond Demon Ops**

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

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

- **Angel Ops**
  - $\cup$
  - $\times$
  - $x' = f(x)$
  - $?Q$
  - Choice
  - Repeat
  - Evolve
  - Challenge

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

Mathematical formulas:

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

**Angel Ops**

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

**Demon Ops**

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

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

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

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

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

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

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

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

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

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

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

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

**Definition (dGL Formula P)**

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

**All Reals**

**Some Reals**
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 $P$)

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

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

All Reals Some Reals

André Platzer (CMU)
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 $P$)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\( d \) before \( a \) can compensate for \( v \geq 1 \) before it can compensate for \( v \geq 0 \).
Example: Push-around Cart

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

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

\[ x \geq 0 \]

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

\[ x \geq 0 \]

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

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

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

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

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

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

\[ x \quad v \]

\[ d \quad a \]

\[ v \geq 1 \rightarrow \]

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

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

\[ \not\exists \]

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

counterstrategy \( d := -1 \)

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

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

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

\begin{align*}
\mathbf{w} - e &\leq 1 \land \mathbf{v} = f \\
\langle ((\mathbf{u} := 1 \cap \mathbf{u} := -1); \\
(g := 1 \cup g := -1); \\
\mathbf{t} := 0; \\
(w' = \mathbf{v}, \mathbf{v}' = \mathbf{u}, \mathbf{e}' = f, \mathbf{f}' = g, \mathbf{t}' = 1 & \mathbf{t} \leq 1)^d \\
\times \rangle (\mathbf{w} - e)^2 &\leq 1
\end{align*}

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

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

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

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

\[(w - e)^2 \leq 1 \land v = f \rightarrow
\]
\[
\left[\left((u := 1 \cap u := -1);\right.ight.
\]
\[
\quad (g := 1 \cup g := -1);\right.
\]
\[
\quad t := 0;
\]
\[
\quad (w' = v, v' = u, e' = f, f' = g, t' = 1 \& t \leq 1)
\]
\[
\right)\times (w - e)^2 > 1
\]

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

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

\[ (x, y) \]

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

\[ \left( \left( w := +w \cap w := -w \right); \right. \]

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

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

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

\[
\begin{align*}
\left(\frac{x}{v}\right)^2 (u - w)^2 &\leq 1 \land \\
x &< 0 \land v > 0 \land y = g \Rightarrow \\
\left\langle (w := +w \cap w := -w); \\
((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\})^* \right\rangle x^2 + (y - g)^2 &\leq 1
\end{align*}
\]
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 $\alpha$: operational semantics)

$x := e$

$\omega_x$
Definition (Hybrid game $\alpha$: operational semantics)

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

$\omega := e \circ \omega \in \omega \circ [Q]$

$\omega := e \circ \omega' = f(\omega) \& Q \circ \phi(r) \circ t \circ \phi(t) = 0$

$\omega \in Q \circ \omega \in \omega \in \{ \alpha, \beta \}$
Definition (Hybrid game $\alpha$: operational semantics)

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

$\omega x \omega' = f(x) \land Q \phi (r)$

$\omega ? Q \omega \in \left[ [ Q ] \right]$

$\omega \alpha \cup \beta$

$\alpha \leftarrow \omega \leftarrow \beta$

$\omega \leftarrow \alpha \leftarrow \alpha \leftarrow \alpha$

$s_1 \leftarrow s_i \leftarrow s_\lambda$

$\beta \leftarrow \beta \leftarrow \beta \leftarrow \beta$

$t_1 \leftarrow t_j \leftarrow t_\kappa$

Diagram with states and transitions.
Definition (Hybrid game $\alpha$: operational semantics)

$\omega := [\omega] \times [\omega]$

$\alpha := \omega \in [\alpha] \cup [\beta]$

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

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

$$x := e$$

$$x' = f(x) \land Q(\phi)$$

$$t = \phi(r)$$

$$t = \phi(0)$$

$$\omega? = Q(\omega) \land \in [Q] \cup \beta$$

$$\omega! = Q(\omega)$$

$$\alpha, \beta$$

$$\alpha^*$$

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

$\alpha := e$,
$\omega x := e$,
$x' = f(x) \land Q$,
$\phi(r)$,
$\omega \in \{ [Q] \}$,
$\alpha \cup \beta$,
$\tau \in \{ [\alpha] \}$,
$\lambda \in \{ [\beta] \}$,
$\alpha \tau \leftarrow \alpha \tau$,
$\alpha \tau \rightarrow \alpha \tau$,
$\alpha \tau \rightarrow \alpha \tau$,
$\alpha \tau \rightarrow \alpha \tau$,
$\alpha \tau \rightarrow \alpha \tau$,
$\alpha \tau \rightarrow \alpha \tau$,
$\alpha \tau \rightarrow \alpha \tau$.

André Platzer (CMU)
\(\langle x := 0 \cap x := 1 \rangle^* x = 0\)
Filibusters & The Significance of Finitude

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

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

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

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

wfd \( \approx \) false unless \( x = 0 \)
true

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

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

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

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

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

$\langle x := 0 \cap x := 1 \rangle^* x = 0$

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 $P$)

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

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

All Reals
Some Reals
Angel Wins
Demon Wins
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
7 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 \(\square\) within time \(\varepsilon\)
- Never leave outer \(\square\)

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

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

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

\[
( a_x := *; \ ?(-A \leq a_x \leq A); \\
a_y := *; \ ?(-A \leq a_y \leq A); \quad \text{ // “independent” robot acceleration} \\
t_s := 0 )^d ;
\]
Example (Robot-Demon vs. Angel-Factory Environment)

\[
\left( \begin{array}{c}
(\text{true} \cup (x < e_x \land y < e_y \land ef_1 = 1); \ v_x := v_x + c_x; \ ef_1 := 0) \quad \text{// belt} \\
\cup (e_x \leq x \land y \leq f_y \land ef_2 = 1); \ v_y := v_y + c_y; \ ef_2 := 0) \end{array} \right);
\]

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

\[t_s := 0\]

\[
(x' = v_x, y' = v_y, v'_x = a_x, v'_y = a_y, t' = 1, t'_s = 1 \land t_s \leq \varepsilon)
\]
Example (Robot-Demon vs. Angel-Factory Environment)

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

Proposition (Robot stays in □)

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

Proposition (Stays in □ and leaves ■ on time)

$RF|_x$: RF projected to the x-axis

\[ (x = 0 \land v_x = 0 \land \text{Controllability Assumptions}) \rightarrow \mathcal{RF}|_x(x \in [l_x, r_x] \land (t \geq \varepsilon \rightarrow x \geq x_b)) \]
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.
In Nicola Olivetti and Ashish Tiwari, editors, IJCAR, volume 9706 of
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.