# COSC-6590/GSCS-6390

# Games: Theory and Applications

## Lecture 14 - Dynamic Games

Luis Rodolfo Garcia Carrillo

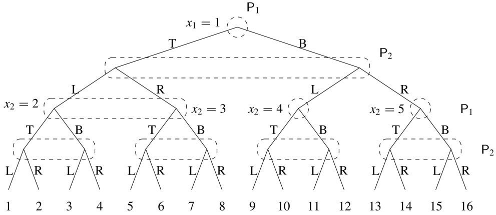
School of Engineering and Computing Sciences
Texas A&M University - Corpus Christi, USA

# Table of contents

# Game Dynamics

# Game Dynamics

Consider a two-player multi-stage game in extensive form




For each stage $k \in \{1, 2, \ldots, K\}$

**1.** $x_k$ : the **node** at which the game enters the $k$th stage

- $x_k$ is called the **state of a game** at the $k$th stage

**2.** $u_k$ : the **action** of player $P_1$ at the $k$th stage

**3.** $d_k$ : the **action** of player $P_2$ at the $k$th stage

## Game Dynamics

**Overall tree structure** can be mathematically described as:

$$\underbrace{x_{k+1}}_{\substack{\text{entry node at}\\ \text{stage } k+1}} = \underbrace{f_k}_{\substack{\text{"dynamics"}\\ \text{at stage } k}} \Bigg( \underbrace{x_k}_{\substack{\text{entry node}\\ \text{at stage } k}}, \underbrace{u_k}_{\substack{P_1\text{'s action}\\ \text{at stage } k}}, \underbrace{d_k}_{\substack{P_2\text{'s action}\\ \text{at stage } k}} \Bigg)$$

$\forall k \in \{1, 2, \ldots, K-1\}$ as shown

| $k$ | $x_k$ | $u_k$ | $d_k$ | $f_k(x_k, u_k, d_k)$ |
|---|---|---|---|---|
| 1 | 1 | T | L | 2 |
| 1 | 1 | T | R | 3 |
| 1 | 1 | B | L | 4 |
| 1 | 1 | B | R | 5 |

| $x_K$ | $u_K$ | $d_K$ | $J_1(x_K, u_K, d_K)$ |
|---|---|---|---|
| 2 | T | L | 1 |
| 2 | T | R | 2 |
| 2 | B | L | 3 |
| 2 | B | R | 4 |
| 3 | T | L | 5 |
| 3 | T | R | 6 |
| 3 | B | L | 7 |
| 3 | B | R | 8 |
| 4 | T | L | 9 |
| 4 | T | R | 10 |
| 4 | B | L | 11 |
| 4 | B | R | 12 |
| 5 | T | L | 13 |
| 5 | T | R | 14 |
| 5 | B | L | 15 |
| 5 | B | R | 16 |

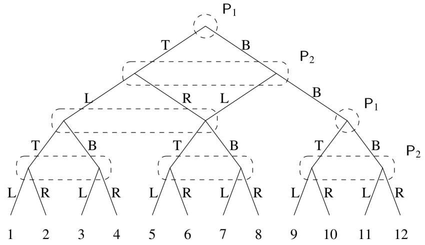
# Game Dynamics

**Tree:** a (connected) graph that has no cycles

- previous description allows for games that are more general

Example:

- games described by graphs that are not trees:




- games with infinitely many stages ($K = \infty$);
- games with action spaces that are not finite sets.

# Game Dynamics

Games whose evolution is represented by an equation such as

$$\underbrace{x_{k+1}}_{\substack{\text{entry node at}\\ \text{stage } k+1}} = \underbrace{f_k}_{\substack{\text{“dynamics”}\\ \text{at stage } k}} \Big( \underbrace{x_k}_{\substack{\text{entry node}\\ \text{at stage } k}} , \underbrace{u_k}_{\substack{P_1\text{'s action}\\ \text{at stage } k}} , \underbrace{d_k}_{\substack{P_2\text{'s action}\\ \text{at stage } k}} \Big)$$

$\forall k \in \{1, 2, \ldots, K-1\}$ are called **dynamic games**

- the equation is called the **dynamics of the game**.

**State-space** of the game: set $\mathcal{X}$ where state $x_k$ takes values.

The **outcome** $J_i$ for a particular $P_i$, $i \in \{1, 2\}$ in a multi-stage game in extensive form is a function of

- state of the game at the last stage $K$, and
- actions taken by the players at this stage

$$J_i(x_K, u_K, d_K)$$

# Game Dynamics

**Game described by a graph that is not a tree**

- different outcomes, depending on how one got to the end

Outcome $J_i$ may depend on all the decisions made by both players from the start of the game:

$$J_i(u_i.d_1, u_1, d_1, \cdots, u_k, d_k)$$

The dynamic game has a **stage-additive cost** when the outcome $J_i$ to be **minimized** is written as

$$\sum_{k=1}^{K} g_k^i(x_k, u_k, d_k)$$

When all $g_k^i = 0$, except for the last $g_K^i$, the game is said to have a **terminal cost**.

# Game Dynamics

When $K = \infty$ we have an infinite horizon game, in which case the previous equation is really a series.

The outcome in

$$J_i(x_K, u_K, d_K)$$

corresponds precisely to a terminal cost.

# Information Structures

# Information Structures

**Open-Loop (OL) dynamic games**

Here, the Players

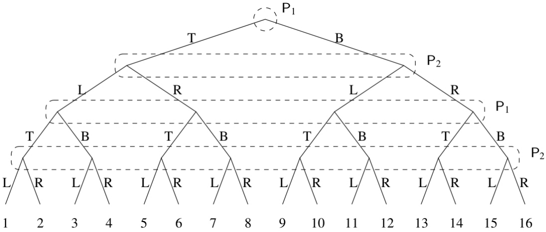
- do not gain any information as the game is played
  - other than the current stage
- must make their decisions solely based on a priori information.

In terms of extensive form representation

- each player has a single information set per stage, which contains all the nodes for that player at that stage

# Information Structures

As in the game




**Policies**: represented as functions of the initial state $x_1$

When $P_1$ uses an OL policy $\gamma^{\mathrm{OL}} := \{\gamma_1^{\mathrm{OL}}, \gamma_2^{\mathrm{OL}}, \ldots, \gamma_K^{\mathrm{OL}}\}$, that player sets

$$u_1 = \gamma_1^{\mathrm{OL}}(x_1), \qquad u_2 = \gamma_2^{\mathrm{OL}}(x_1), \qquad \cdots \qquad u_K = \gamma_K^{\mathrm{OL}}(x_1)$$

# Information Structures

When $P_2$ uses an OL policy $\sigma^{\text{OL}} := \{\sigma_1^{\text{OL}}, \sigma_2^{\text{OL}}, \ldots, \sigma_K^{\text{OL}}\}$, that player sets

$$d_1 = \sigma_1^{\text{OL}}(x_1), \qquad d_2 = \sigma_2^{\text{OL}}(x_1), \qquad \cdots \qquad d_K = \sigma_K^{\text{OL}}(x_1)$$

OL policies are expressed as functions of a (typically fixed) initial state

- this emphasizes that OL policies cannot depend on information collected later in the game

In contrast to state-feedback games.

## Information Structures

**(Perfect) state-feedback (FB) games:**

Here, the Players

- know exactly the state $x_k$ of the game at the entry of the current stage
- can use this information to choose their actions $u_k$ and $d_k$ at that stage

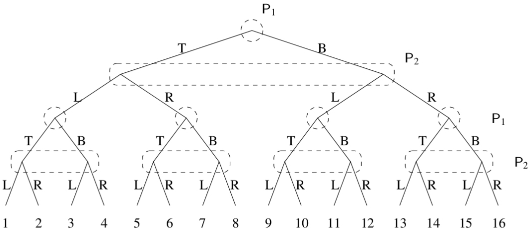
However, they must make these decisions without knowing each others choice (i.e., simultaneous play at each stage).

In terms of extensive form representation

- at each stage of the game there is exactly one information set for each entry-point to that stage.

# Information Structures

As in the game




**Policies:** represented as functions of the current state

When $P_1$ uses a FB policy $\gamma^{\text{FB}} := \{\gamma_1^{\text{FB}}, \gamma_2^{\text{FB}}, \dots, \gamma_K^{\text{FB}}\}$, that player sets

$$u_1 = \gamma_1^{\text{FB}}(x_1), \qquad u_2 = \gamma_2^{\text{FB}}(x_2), \qquad \cdots \qquad u_K = \gamma_K^{\text{FB}}(x_K)$$

# Information Structures

When $P_2$ uses a FB policy $\sigma^{\text{FB}} := \{\sigma_1^{\text{FB}}, \sigma_2^{\text{FB}}, \ldots, \sigma_K^{\text{FB}}\}$, that player sets

$$d_1 = \sigma_1^{\text{FB}}(x_1), \qquad d_2 = \sigma_2^{\text{FB}}(x_2), \qquad \cdots \qquad d_K = \sigma_K^{\text{FB}}(x_K)$$

Now that we defined admissible sets of policies (i.e., action spaces) and how these translate to outcomes through the dynamics of the game, the general definitions introduced in **Lecture 9** specify unambiguously what is meant by a security policy or a NE for these games.

# Continuous-Time Differential Games

# Continuous-Time Differential Games

Dynamic Games formulated in continuous time

1. state $x(t)$ varies continuously with time on a given interval $t \in [0, t]$
2. players continuously select actions $u(t)$ and $d(t)$ on $[0, t]$, which determine the evolution of the states.

If state $x(t)$ is an $n$-vector of real numbers whose evolution is determined by a differential equation, the game is called a **differential game**.

We consider differential games with dynamics of the form

$$\underbrace{\dot{x}(t)}_{\substack{\text{state}\\\text{derivative}}} = \underbrace{f}_{\substack{\text{game}\\\text{dynamics}}}\Big(\underbrace{t}_{\text{time}}, \underbrace{x(t)}_{\substack{\text{current}\\\text{state}}}, \underbrace{u(t)}_{\substack{P_1\text{'s action}\\\text{at time } t}}, \underbrace{d(t)}_{\substack{P_2\text{'s action}\\\text{at time } t}}\Big), \quad \forall t \in [0, T]$$

# Continuous-Time Differential Games

Each $P_i, \in \{1,2\}$ wants to **minimize** a cost of the form

$$J_i := \underbrace{\int_0^T g_i\big(t, x(t), u(t), d(t)\big)dt}_{\text{cost along trajectory}} \;+\; \underbrace{q_i(x(T))}_{\text{final cost}}$$

**Notation:** when $T = \infty$ we have an infinite horizon game. The final cost term is absent.

We also consider OL policies of the form

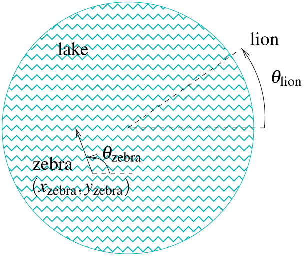
$$u(t) = \gamma^{\text{OL}}(t, x(0)), \qquad d(t) = \sigma^{\text{OL}}(t, x(0)), \qquad \forall t \in [0, T]$$

and (perfect) FB policies of the form

$$u(t) = \gamma^{\text{FB}}(t, x(t)), \qquad d(t) = \sigma^{\text{FB}}(t, x(t)), \qquad \forall t \in [0, T]$$

# Continuous-Time Differential Games

**Example 14.1** (Zebra in the lake). Game is depicted as




$P_1$ is a **zebra** that swims with a speed of $v_{\text{zebra}}$ in a circular lake with radius $R$

$P_2$ is a **lion** that runs along the perimeter of the lake with maximum speed of $v_{\text{lion}} > v_{\text{zebra}}$

# Continuous-Time Differential Games

Notation:

- $(x_{\text{zebra}}, y_{\text{zebra}})$: the position of the zebra
- $\theta_{\text{zebra}}$: the orientation of the zebra

We have that

$$\dot{x}_{\text{zebra}} = v_{\text{zebra}} \cos\theta_{\text{zebra}}, \qquad \dot{y}_{\text{zebra}} = v_{\text{zebra}} \sin\theta_{\text{zebra}}, \qquad \theta_{\text{zebra}} \in [0, 2\pi)$$

Notation:

- $\theta_{\text{lion}}$ the angular position of the lion
- $\omega_{\text{lion}}$ the angular velocity of the lion

We have that

$$\dot{\theta}_{\text{lion}} = \omega_{\text{lion}}, \qquad \omega_{\text{lion}} \in \left[-\frac{v_{\text{lion}}}{R}, +\frac{v_{\text{lion}}}{R}\right]$$

# Continuous-Time Differential Games

Defining a state vector

$$x(t) := [x_{\text{zebra}}(t) \quad y_{\text{zebra}}(t) \quad \theta_{\text{lion}}(t)]'$$

the equations

$$\dot{x}_{\text{zebra}} = v_{\text{zebra}} \cos\theta_{\text{zebra}}, \qquad \dot{y}_{\text{zebra}} = v_{\text{zebra}} \sin\theta_{\text{zebra}}, \qquad \theta_{\text{zebra}} \in [0, 2\pi)$$

$$\dot{\theta}_{\text{lion}} = \omega_{\text{lion}}, \qquad \omega_{\text{lion}} \in \left[-\frac{v_{\text{lion}}}{R}, +\frac{v_{\text{lion}}}{R}\right]$$

can be written as in

$$\underbrace{\dot{x}(t)}_{\substack{\text{state}\\ \text{derivative}}} = \underbrace{f}_{\substack{\text{game}\\ \text{dynamics}}} \Big(\underbrace{t}_{\text{time}}, \underbrace{x(t)}_{\substack{\text{current}\\ \text{state}}}, \underbrace{u(t)}_{\substack{P_1\text{'s action}\\ \text{at time } t}}, \underbrace{d(t)}_{\substack{P_2\text{'s action}\\ \text{at time } t}}\Big), \quad \forall t \in [0, T]$$

where the actions of the players are:

$$u(t) = \theta_{\text{zebra}}(t) \in [0, \pi) \qquad d(t) = \omega_{\text{lion}}(t) \in \left[-\frac{v_{\text{lion}}}{R}, +\frac{v_{\text{lion}}}{R}\right]$$

# Continuous-Time Differential Games

Assume that the zebra wants to get out of the lake as soon as possible without being captured.

The zebra's cost is of the form

$$J_1 = \begin{cases} T_{\text{exit}} & \text{zebra exits the lake safely at time } T_{\text{exit}} \\ +\infty & \text{zebra gets caught when it exits.} \end{cases}$$

A **zero-sum game**: the lion wants to maximize $J_1$

- or equivalently minimize $J_2 := -J_1$.

# Continuous-Time Differential Games

Trick to write such a cost in an integral form such as

$$J_i := \underbrace{\int_0^T g_i\big(t, x(t), u(t), d(t)\big)dt}_{\text{cost along trajectory}} \quad + \quad \underbrace{q_i(x(T))}_{\text{final cost}}$$

Freeze the state when the zebra reaches the shore, which amounts to replacing

$$\dot{x}_{\text{zebra}} = v_{\text{zebra}} \cos\theta_{\text{zebra}}, \qquad \dot{y}_{\text{zebra}} = v_{\text{zebra}} \sin\theta_{\text{zebra}}, \qquad \theta_{\text{zebra}} \in [0, 2\pi)$$

and

$$\dot{\theta}_{\text{lion}} = \omega_{\text{lion}}, \qquad\qquad \omega_{\text{lion}} \left[-\frac{v_{\text{lion}}}{R}, +\frac{v_{\text{lion}}}{R}\right]$$

# Continuous-Time Differential Games

By

$$\begin{bmatrix} \dot{x}_{\text{zebra}} \\ \dot{y}_{\text{zebra}} \\ \dot{\theta}_{\text{lion}} \end{bmatrix} = \begin{cases} \begin{bmatrix} v_{\text{zebra}} \cos\theta_{\text{zebra}} \\ v_{\text{zebra}} \sin\theta_{\text{zebra}} \\ \omega_{\text{lion}} \end{bmatrix} & x_{\text{zebra}^2} + y_{\text{zebra}^2} < R^2 \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & x_{\text{zebra}^2} + y_{\text{zebra}^2} = R^2 \end{cases}$$

And then defining

$$J_1 := \int_0^\infty g(x_{\text{zebra}}, y_{\text{zebra}}, \theta_{\text{lion}})dt$$

where

$$g(x_{\text{zebra}}, y_{\text{zebra}}, \theta_{\text{lion}}) = \begin{cases} 1 & x_{\text{zebra}^2} + y_{\text{zebra}^2} < R^2 \\ 1 & x_{\text{zebra}} = R\cos\theta_{\text{lion}}, \quad y_{\text{zebra}} = R\sin\theta_{\text{lion}} \text{ (zebra is caught)} \\ 0 & \text{otherwise (zebra reaches shore away from lion)} \end{cases}$$

# Continuous-Time Differential Games

This game is only meaningful in the context of **state-feedback policies**

**The lion has no chance of capturing the zebra unless the lion can see the zebra.**

# Differential Games with Variable Termination Time

# Differential Games with Variable Termination Time

A less convoluted way to formalize pursuit-evasion games

Consider the usual continuous-time dynamics

$$\underbrace{\dot{x}(t)}_{\substack{\text{state}\\\text{derivative}}} = \underbrace{f}_{\substack{\text{game}\\\text{dynamics}}}\big(\underbrace{t}_{\text{time}}, \underbrace{x(t)}_{\substack{\text{current}\\\text{state}}}, \underbrace{u(t)}_{\substack{P_1\text{'s action}\\\text{at time } t}}, \underbrace{d(t)}_{\substack{P_2\text{'s action}\\\text{at time } t}}\big), \quad \forall t \in [0, T]$$

but costs to be minimized by each player $P_i$ of the form

$$J_i := \underbrace{\int_0^{T_{\text{end}}} g_i\big(t, x(t), u(t), d(t)\big)\,dt}_{\text{cost along trajectory}} + \underbrace{q_i\big(T_{\text{end}}, x(T_{\text{end}})\big)}_{\text{final cost}}$$

where $T_{\text{end}}$ is

- the first time at which the state $x(t)$ enters a closed set $\mathcal{X}_{\text{end}} \subset \mathbb{R}^n$, or
- $T_{\text{end}} = +\infty$ in case $x(t)$ never enters $\mathcal{X}_{\text{end}}$

# Differential Games with Variable Termination Time

Think of $\mathcal{X}_{\text{end}}$ as the set of states at which the game terminates

- the evolution of $x(t)$ is irrelevant after this time.

The states in $\mathcal{X}_{\text{end}}$ are often called the **game-over** states.

# Continuous-Time Differential Games

**Example** (Zebra in the lake, continuation)
Game can be formalized as a differential game with dynamics

$$\begin{bmatrix} \dot{x}_{\text{zebra}} \\ \dot{y}_{\text{zebra}} \\ \dot{\theta}_{\text{lion}} \end{bmatrix} = \begin{bmatrix} v_{\text{zebra}} \cos\theta_{\text{zebra}} \\ v_{\text{zebra}} \sin\theta_{\text{zebra}} \\ \omega_{\text{lion}} \end{bmatrix}, \quad \theta_{\text{zebra}} \in [0, \pi), \quad \omega_{\text{lion}} \in \left[-\frac{v_{\text{lion}}}{R}, +\frac{v_{\text{lion}}}{R}\right]$$

and a cost

$$J_1 := \int_0^{T_{\text{end}}} dt \quad + \quad q(x(T_{\text{end}}))$$

where $T_{\text{end}}$ is the first time at which the state $x(t)$ enters the set

$$\mathcal{X}_{\text{end}} := \left\{(x_{\text{zebra}}, y_{\text{zebra}}, \theta_{\text{lion}},) \subset \mathbb{R}^3 : x^2_{\text{zebra}} + y^2_{\text{zebra}} \geq R^2\right\}$$

of **safe** configurations for the zebra to reach the shore.

# Continuous-Time Differential Games

The final cost

$$q(x) := \begin{cases} 0 & \text{if } (x_{\text{zebra}}, y_{\text{zebra}}) \neq (R\cos\theta_{\text{lion}}, R\sin\theta_{\text{lion}}) \\ \infty & \text{otherwise} \end{cases}$$

greatly penalizes the zebra (minimizer) for being caught.

End of Lecture

**14 - Dynamic Games**

Questions?