Last time:

- ☐ zero-sum dynamic games

Lecture 5
IR, Spring '24
Andrea Bajcsy

This time:

- ☐ (finish) zero-sum games
- ☐ HJI Equation
- ☐ safety analysis

Before we start, quick connection / example about open-loop information patterns & order of play....

---

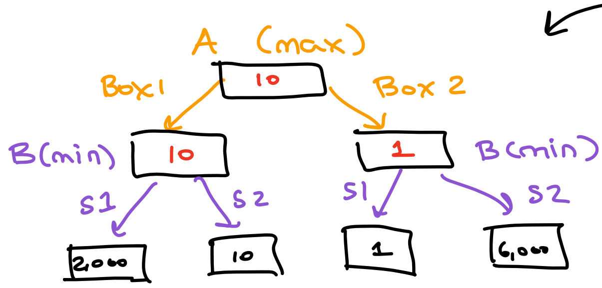
## TOY EXAMPLE [Player A goes first]

Box 1

| s1 | s2 |
|---|---|
| 2,000 | 10 |

Box 2

| s1 | s2 |
|---|---|
| 1 | 6,000 |




$$\max_{u_A \in \mathcal{U}_A}\left(\min_{u_B \in \mathcal{U}_B} J(u_A, u_B)\right)$$

"computed second" but "plays first"

"computed first" in game tree, but "plays second"

⇓

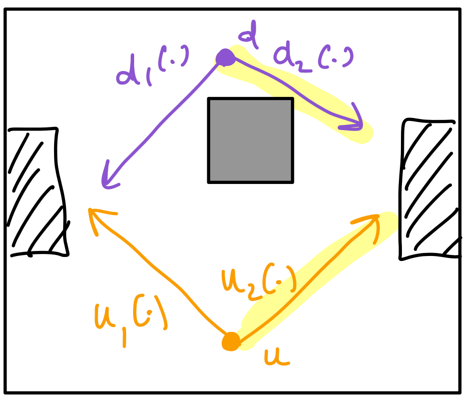
## OPEN-LOOP "SIGNAL" EXAMPLE [Player A goes first]

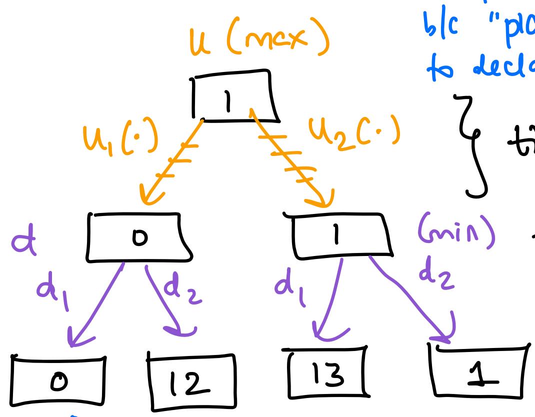
$$\max_{u(\cdot) \in \mathcal{U}_0^T}\left(\min_{d(\cdot) \in \mathbb{D}_0^T} J(u(\cdot), d(\cdot))\right)$$

b/c "plays first", it has to declare entire signal....

Then d responds....

G1




} time interval [0,T]!

} time interval [0,T]!

ex. distance btwn. agents

⇒ OVERLY PESSIMISTIC (from POV of agent u)

when $\mathcal{U}_0^T$, $\mathbb{D}_0^T$ are truly all possible signals, you have this ∞-wide tree!

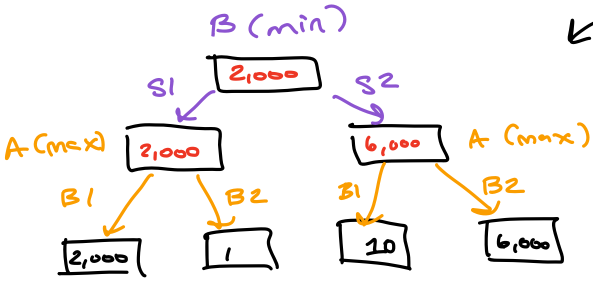
# TOY EXAMPLE [PLAYER B GOES FIRST]

Box 1

| s1 | s2 |
|---|---|
| 2,000 | 10 |

Box 2

| s1 | s2 |
|---|---|
| 1 | 6,000 |




$$\min_{u_B\in\mathcal{U}_B}\left(\max_{u_A\in\mathcal{U}_A} J(u_A,u_B)\right)$$

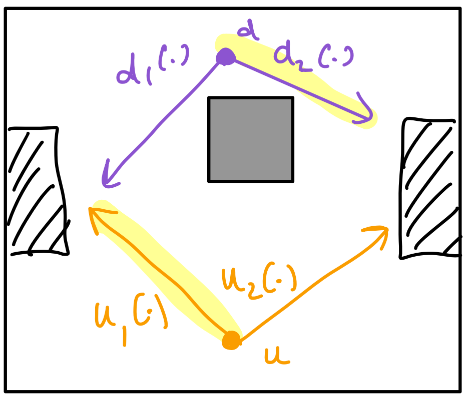
"computed second" but "plays first"

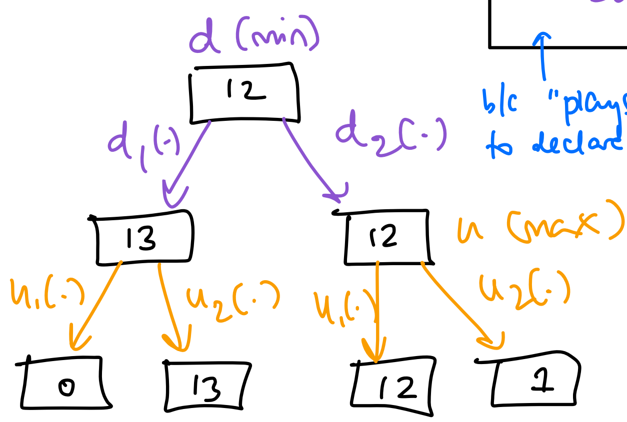
"computed first" in game tree, but "plays second"

⇓

# OPEN-LOOP "SIGNAL" EXAMPLE [PLAYER B goes first]

G1




$$\min_{d(\cdot)\in\mathbb{D}_0^T}\left(\max_{u(\cdot)\in\mathbb{U}_0^T} J(u(\cdot),d(\cdot))\right)$$

b/c "plays first", it has to declare entire signal...

Then u responds

⇒ OVERLY OPTIMISTIC (from POV of agent u)

Now, we are ready to state the ROBUST VALUE FUNCTION. Under non-anticipative strategies, it can be shown that the value function exists!

We can obtain it via the dynamic programming principle:

$$V(x,t) = \max_{\delta[u](\cdot) \in \Gamma_t^T} \min_{u(\cdot) \in \mathbb{U}_t^T} J(x, u(\cdot), \overset{\text{"}d(\cdot)\text{"}}{\delta[u](\cdot)}, t)$$

$$= \max_{\delta[u](\cdot) \in \Gamma_t^{t+\Delta}} \min_{u(\cdot) \in \mathbb{U}_t^{t+\Delta}} \Big\{ \int_t^{t+\Delta} L(x(\tau), u(\tau), \delta[u](\tau))\, d\tau + \underbrace{\max_{\delta[u](\cdot) \in \Gamma_{t+\Delta}^T} \min_{u(\cdot) \in \mathbb{U}_{t+\Delta}^T} \int_{t+\Delta}^T L(x(\tau), u(\tau), \delta[u](\tau))\, d\tau + \ell(x(T))}_{:= V(x(t+\Delta),\, t+\Delta)} \Big\}$$

$:= V(x(t+\Delta), t+\Delta)$ by Tenent of Transition (i.e. principle of optimality in game theory)

$$= \max_{\delta[u](\cdot) \in \Gamma_t^{t+\Delta}} \min_{u(\cdot) \in \mathbb{U}_t^{t+\Delta}} \Big\{ \int_t^{t+\Delta} L(x(\tau), u(\tau), \delta[u](\tau))\, d\tau + V(x(t+\Delta), t+\Delta) \Big\}$$

Similar to before, we simplify $V(x(t+\Delta), t+\Delta)$, take $\Delta \to 0$, we get:

## Hamilton-Jacobi-Isaacs (HJI) Equation

$$\frac{\partial V(x,t)}{\partial t} + \min_{u(t)\in U} \max_{d(t) \in D} \Big\{ L(x(t), u(t), d(t)) + \underbrace{\nabla_x V(x,t)^T f(x,u,d)}_{\text{" } V_{t+1}(f(x_t, u_t, d_t)) \text{ "}} \Big\} = 0$$

in discrete-time

$$V(x,T) = \ell(x)$$

⊛ HJI looks very similar to HJB from last time w/ min operator replaced by min-max.

❗ tools to solve HJB can also often be used to solve HJI equation!

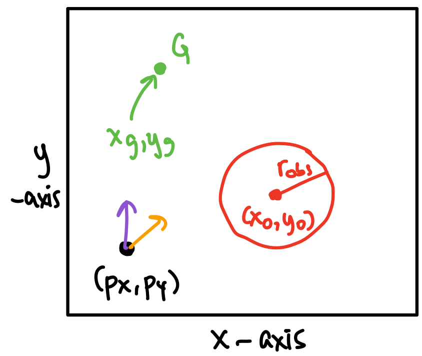
# SUMMARY:

| | OPEN-LOOP | CLOSED-LOOP |
|---|---|---|
| Continuous in $t$ | $V(x,t) = \min\limits_{u(\cdot)\in\mathbb{U}_t^T} \max\limits_{d(\cdot)\in\mathbb{D}_t^T} J(x, u(\cdot), d(\cdot), t)$ | $V(x,t) = \max\limits_{\delta[u](\cdot)\in\Gamma_t^T} \min\limits_{u(\cdot)\in\mathbb{U}_t^T} J(x, u(\cdot), \delta[u](\cdot), t)$ |
| Discrete in $t$ | $V_t(x) = \min\limits_{u^{t:T}} \max\limits_{d^{t:T}} J_t(x, u^{t:T}, d^{t:T})$ <br> $= \min\limits_{u^t} \min\limits_{u^{t+1}} \cdots \min\limits_{u^T} \max\limits_{d^t} \cdots \max\limits_{d^T} J_t(x, u^{t:T}, d^{t:T})$ | feedback policy! <br> $V_t(x) = \min\limits_{u^t} \max\limits_{d^t} \min\limits_{u^{t+1}} \max\limits_{d^{t+1}} \cdots \min\limits_{u^T} \max\limits_{d^T} J_t(x, u^{t:T}, d^{t:T})$ <br> $= \min\limits_{\pi_u} \max\limits_{\pi_d} J_t(x, u^{t:T}_{\pi_u}, d^{t:T}_{\pi_d})$ |

$$\Downarrow$$

$$V_t(x) = \min_{u_t} \max_{d_t} \left\{ L_t(x_t, u_t, d_t) + V_{t+1}(f(x,u,d)) \right\}$$

**EXAMPLE**




State: $x = \begin{bmatrix} p^x \\ p^y \end{bmatrix}$

Dynamics: $\dot{x} = f(x, u, d) \equiv \begin{bmatrix} v_x + d_x \\ v_y + d_y \end{bmatrix}$

Control: $u = [v_x, v_y]$ , $|v_x|, |v_y| \leq 1$ m/s

Disturbance: $d = [d_x, d_y]$ , $|d_x|, |d_y| \leq 0.2$ m/s

Cost Function:

$$J(x, u(\cdot), d(\cdot), t) = \int_{\tau=t}^{T} \overbrace{\left(dist^2(x(\tau), G) + \lambda \cdot obs\_penetration(x(\tau))\right)}^{= L(x)} d\tau$$
$$+ \overbrace{\left(dist^2(x(T), G) + \lambda \cdot obs\_penetration(x(T))\right)}^{\ell(x)}$$

$$dist^2(x(t), G) = (p_x(t) - x_g)^2 + (p_y(t) - y_g)^2$$

$$obs\_penetration(x(t)) = \max\left\{0, \underbrace{r_{obs} - \sqrt{(p_x(t) - x_o)^2 + (p_y(t) - y_o)^2}}_{>0 \text{ if inside obstacle rad (\& deeper in, gets more cost!)}}\right\}$$

Hamiltonian: $H(x, \nabla V)$

$$\frac{\partial V(x,t)}{\partial t} + \boxed{\min_{u(t) \in U} \max_{d(t) \in D} \left\{ L(x(t), u(t), d(t)) + \nabla_x V(x,t)^T f(x, u, d) \right\}} = 0$$

$$H(x, \nabla V) \equiv L(x(t)) + \min_{u(t) \in U} \max_{d(t) \in D} \left\{ \nabla_x V(x,t)^T \begin{bmatrix} v_x + d_x \\ v_y + d_y \end{bmatrix} \right\}$$

$(v_x, v_y)$ $(d_x, d_y)$

Let's write $\nabla_x V(x,t)$ as $[p_1(x) \quad p_2(x)]$

$$= L(x(t)) + \min_{u(t) \in U} \max_{d(t) \in D} \left\{ [p_1(x) \quad p_2(x)]^T \begin{bmatrix} v_x + d_x \\ v_y + d_y \end{bmatrix} \right\}$$

$$= L(x(t)) + \min_{u(t)\in U} \max_{d(t)\in D} \left\{ p_1(x)(v_x + d_x) + p_2(x)(v_y + d_y) \right\}$$

$$= L(x(t)) + \min_{u(t)\in U} \max_{d(t)\in D} \left\{ p_1(x) v_x + p_1(x) d_x + p_2(x) v_y + p_2(x) d_y \right\}$$

$v_x, v_y, d_x, d_y$ terms are separable so:

$$= L(x(t)) + \min_{v_x} p_1(x) v_x + \min_{v_y} p_2(x) v_y + \max_{d_x} p_1(x) d_x + \max_{d_y} p_2(x) d_y$$

Since $|v_x|, |v_y| \leq 1$ and $|d_x|, |d_y| \leq 0.2$ then we want to choose upper / lower bound to maximize or minimize the quantity $\Rightarrow$ this is determined by the sign of $p_1(x)$, $p_2(x)$.

<u>OPTIMAL</u> $u = [v_x, v_y] \Rightarrow$

$$\begin{cases} v_x^* = \mathbb{1}\{\text{sign}(p_1(x)) > 0\} * (-v^{max}) + \mathbb{1}\{\text{sign}(p_1(x)) \leq 0\} * (v^{max}) \\ v_y^* = \mathbb{1}\{\text{sign}(p_2(x)) > 0\} * (-v^{max}) + \mathbb{1}\{\text{sign}(p_2(x)) \leq 0\} * (v^{max}) \end{cases}$$

<u>OPTIMAL</u> $d = [d_x, d_y] \Rightarrow$

$$\begin{cases} d_x^* = \mathbb{1}\{\text{sign}(p_1(x)) > 0\} * (v^{max}) + \mathbb{1}\{\text{sign}(p_1(x)) \leq 0\} * (-v^{max}) \\ d_y^* = \mathbb{1}\{\text{sign}(p_2(x)) > 0\} * (v^{max}) + \mathbb{1}\{\text{sign}(p_2(x)) \leq 0\} * (-v^{max}) \end{cases}$$

<u>OPTIMAL Hamiltonian:</u>

$$H(x, \nabla V) = L(x(t)) + p_1(x) \cdot (v_x^*) + p_2(x) \cdot (v_y^*) + p_1(x) \cdot (d_x^*) + p_2(x) \cdot (d_y^*)$$

$$= L(x(t)) + |p_1(x)| \cdot (-v^{max}) + |p_2(x)| \cdot (-v^{max}) + |p_1(x)| \cdot (d^{max}) + |p_2(x)| \cdot (d^{max})$$

in this case, Ham. value can be re-written as this b/c:

e.g. $p_1(x) \leq 0 \Rightarrow v_x^* = v^{max} \Rightarrow p_1(x) \cdot v^{max} \leq 0 \equiv -|p_1(x)| \cdot v^{max}$

$p_1(x) > 0 \Rightarrow v_x^* = -v^{max} \Rightarrow p_1(x) \cdot (-v^{max}) < 0 \equiv -|p_1(x)| \cdot v^{max}$

## Formalizing safety via reachability

We now have a handle on how to solve general robust optimal control problems w/ potentially multi-agents. But what if we wanted to ensure that our system abides by some state constraints? For example, what if we want to synthesize an optimal control that guarantees that our robot never hits an obstacle? What are the initial conditions from which robot is doomed to collide?

These questions fall under reachability analysis which is a fundamental problem of identifying "if a certain state of a system is reachable from an initial state of the system":

⇒ fundamental to program analysis, to dynamical systems, to biology!

### Safety Analysis Roadmap

| Define in MATH the set we want to compute automatically | → | Encode the problem computationally / numerically | → | Compute the unsafe set + safety-preserving controller ⇒ as an O.C. problem! |
|---|---|---|---|---|
| ↑ characterize | | ↑ ENCODE | | ↑ COMPUTE |