Last Time:

- ☐ Trajectory Forecasting

This Time:

- ☐ Collab. / Coord. / Assist.

Lecture 10

HRI, FALL '25

Andrea Bajcsy

Collaboration / Assistance / Coordination

- COLLABORATION: ex. R and H are moving a couch together
  - → R & H share the same objective
  - → —"— share the workspace (w/ two diff. embodiments)
  - → —"— both often know the objective
  - → —"— act as a team ⇒ R must plan jointly w/ H

- ASSISTANCE: ex. robot is being teleoperated by human to skewer food
  - → R & H share the same objective
  - → —"— share the workspace (via the shared embodiment)
  - → only H perfectly knows the objective
  - → R's job is to help, despite uncertainty ⇒ R must infer & then assist
    - ↳ preserve some autonomy of H!

- COORDINATION: ex. autonomous car & human car passing @ intersection.
  - → R and H have different but not necessarily opposing objectives
    (ex. R → left lane w/o crash
    H → right lane w/o crash)
    differ   share
  - → —"— share the workspace (w/ diff. embodiments)
  - → H's objective is not necessarily known to R & vice versa
  - → R must coordinate w/ H to accomplish its goal w/o hindering H.

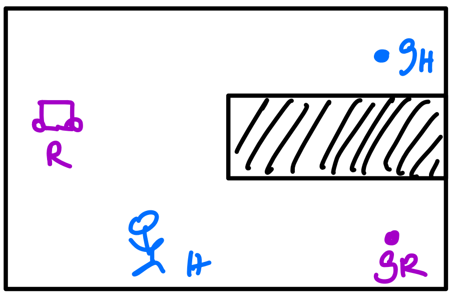
Q How do we model these diff. types of interactions mathematically?

⇒ TODAY: MDPs / POMDPs

⇒ NEXT WEEK: Game Theory

INTERACTION AS AN MDP / POMDP

• COORDINATION — Navigation example



→ model the robot's behavior in a shared workspace w/ human, whose behavior we (the robot) do NOT control

→ R must avoid H while also being efficient in getting to g_R

• state: $s = (s_H, s_R)$ // joint state space of H & R

• action: $a := a_R \in A_R$ // robot only controls its own actions! (this MDP is from the "robot's POV")

• reward:

$$r(s,a) = \begin{cases} -100 \cdot \mathbb{1}\{s_H = s_R\} & \text{// safety wrt. } H \\ -10 \cdot \mathbb{1}\{s_R \in obs\} & \text{// safety wrt. env.} \\ -dist(s_R, g_R) & \text{// efficiency} \\ \vdots & \end{cases}$$

• transition:

$$P(s'|s,a) \overset{\text{by defn}}{=} P(s_R', s_H' \mid s_R, s_H, a_R)$$

$$= \underbrace{P(s_R' \mid s_H', s_R, s_H, a_R)}_{} \cdot \boxed{P(s_H' \mid s_R, s_H, a_R)}$$

product rule

(*)

Robot dynamics ✓ $= P(s_R' \mid s_R, a_R)$ b/c given R's current state & action, R's next state is independent of H.

$$\circledast \quad P(s_H' \mid s_R, s_H, a_R) = \sum_{a_H} P(s_H', a_H \mid s_H, s_R, a_R) \quad \text{// marginalize out H's control action}$$

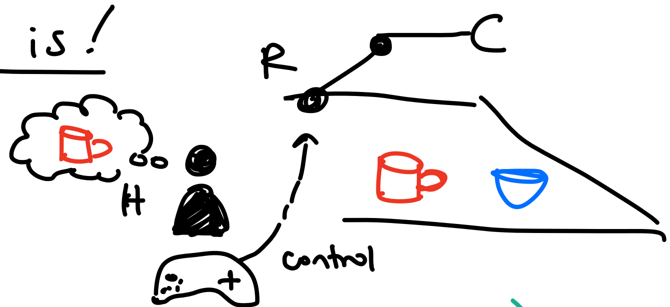
$$= \sum_{a_H} \underbrace{P(s_H' \mid s_H, s_R, a_R, a_H)} \cdot P(a_H \mid s_H, s_R, a_R) \quad \text{// prod. rule}$$

$P(s_H' \mid s_H, a_H)$ b/c given H's state + action the H's next state is indep. of R.

$$= \sum_{a_H} \underbrace{P(s_H' \mid s_H, a_H)}_{\text{H's dynamics / transition func.}} \cdot \underbrace{P(a_H \mid \overset{(s_H, s_R)}{s}, a_R)}_{\text{H's policy (predictive model!)}}$$

- ASSISTANCE as a POMDP

So far, robot had its own known goal / reward; now the R is helping H achieve their goal, <u>BUT it doesn't know what the goal is!</u>



- state: $s = (s_H, s_R, \theta)$

  s_H, s_R: state space is physical AND internal state

  θ: reward parameter (e.g. goal)

- $a := a_R \in A_R$ // robot only controls itself
- $r(s,a) =$ "achieve θ" // reward means executing the task the way the H wants
- $o := a_H \in A_H$ // observations of human's actions! use this as <u>evidence</u> for inferring θ.
- transition: $P(s' \mid s, a) =$

$$= P(s_H', s_R', \theta' \mid s_R, s_H, \theta, a_R)$$

common assumption that $\theta' = \theta$ but not true in general

$\vdots$ (exercise ☺)

$$= \underbrace{P(s_R' \mid s_R, a_R)}_{R \text{ dynamics}} \cdot \sum_{a_H} \underbrace{P(s_H' \mid s_H, a_H)}_{H \text{ dynamics}} \cdot \underbrace{P(a_H \mid s_H, \theta)}_{H\text{'s policy } \boxed{\text{given intent}}}$$

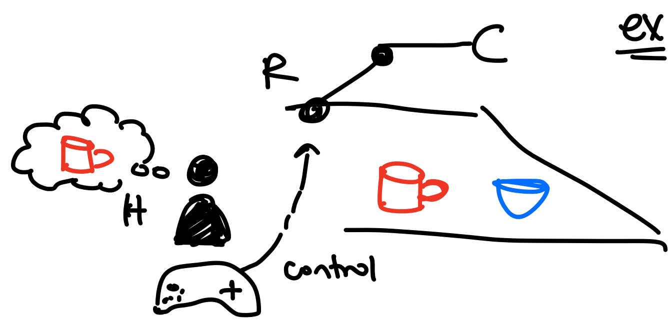
- observation model:

$$P(o \mid s) = P(a_H \mid s_H, s_R, \theta) \overset{\text{assump.}}{=} P(a_H \mid s_H, \theta)$$ } given by H and their policy

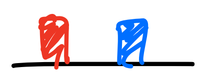
IN THEORY, solving this for the robot's optimal assistance strategy would involve the robot maintaining a belief $b(\theta) := P(\theta \mid a_H^{0:t}, s_H^{0:t})$ and solving POMDP to yield <u>information-gathering actions!</u>

IN PRACTICE, very hard to do but, a common approx. is to solve something called a $\boxed{\text{Q-MDP}}$

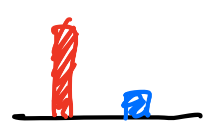
<u>ex</u>. SHARED CONTROL



(1) <u>infer</u> which goal H wants

$b^0(\theta) =$ 

$\vdots$

$b^t(\theta)$ 

(2) <u>robot policy</u> via QMDP:

$$a_R^* = \arg\max_{a_R} \sum_{\theta} b(\theta) \cdot Q_\theta(s, a_R)$$

③ <u>blend</u> human's command with robot's assistive action

$$a_{exec.} = (1-\alpha)\, a_H + \alpha\, a_R^* \quad , \quad \alpha \in [0,1]$$

if you want to see more blending schemes, see

[Dragan & Srinivasa, 2013]