Last Time

- uncertainty

This Time:

- safety

Lecture 16

Fall '26

Andrea Bajcsy

What is safety?

It's very nuanced and means diff. things to diff. communities. In HRI, there are many "layers" where safety can be enforced:

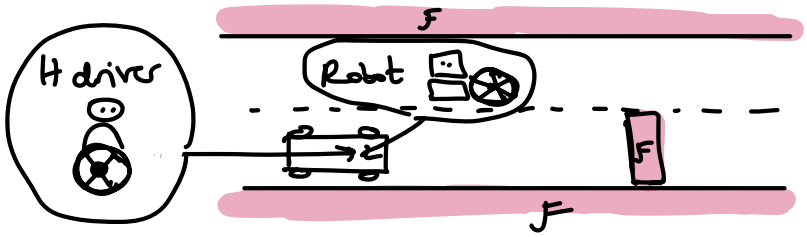
ex.
- Safe Control → compliant ctrl. so robot can't exert too much force on person during physical HRI
- Safe Motion Plan → constraints on motions that robots plan (ex. collisions)
- Psychological Safety → humans perceive robots to be safe

→ from (older) survey: Lasota et. al. "Survey of Safe HRI."

Today, we will focus on **safety as constraint satisfaction**

$s_t \notin F, \forall t \in \{0, \dots T\}$ ← set of hazardous states you never want to enter!

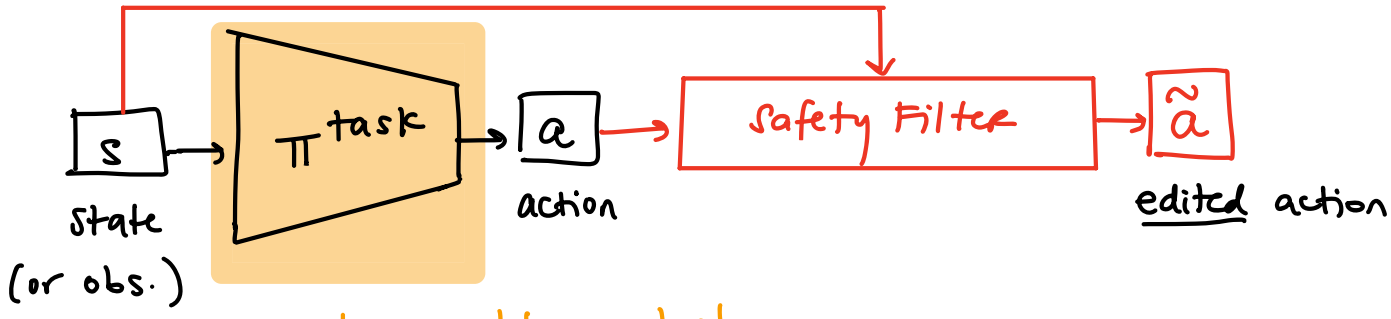
ex. in racing, the car wants to help it never collide with lane boundaries or obstacles



ex. a robot manipulator should never spill hot coffee on the person!



Specifically, we are going to focus on one way that a robot can enforce safety constraints, while allowing ANY task driven policy — like a human! — to do its thing ☺ It's called a safety filter:



ex. human teleoperator!
VLA!

Safety Filtering

Goal A "wrapper" we can put around ANY "base" policy or decision-making system to:

(1) monitor if the system is @ "risk" of violating $s \notin F$

(2) adjust the robot's base policy @ runtime to prevent future failure. ($s_t \notin F \ \forall t \in \{0, 1, \dots\}$)

The idea is that we continue applying our nominal strategy for decision-making until safety is at risk; otherwise, apply a safety controller which steers away!

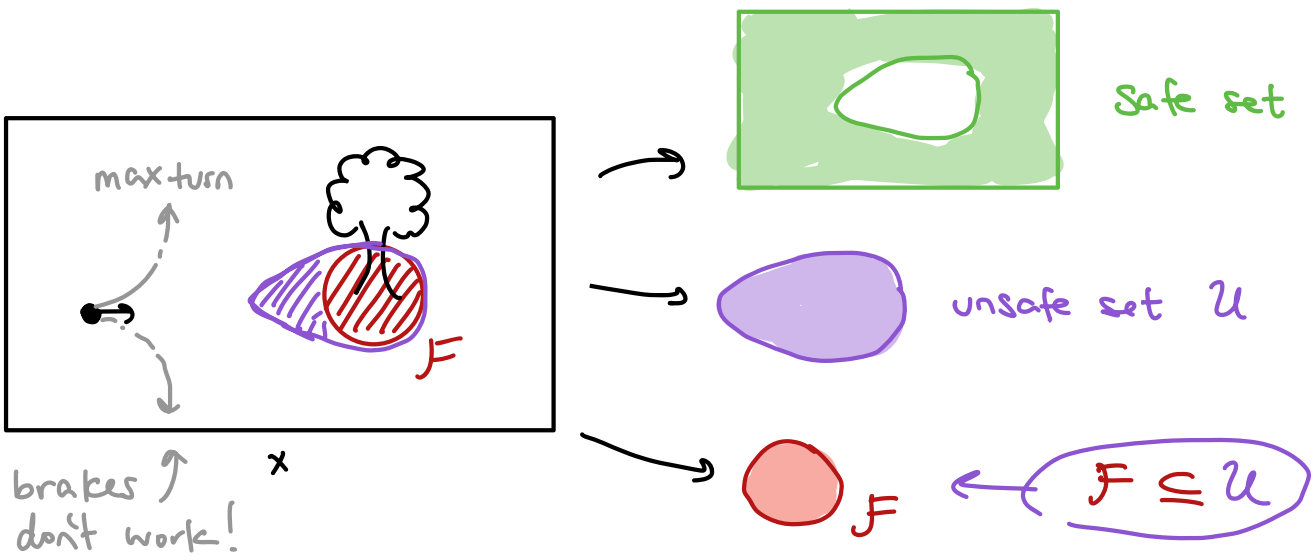
ex. SIMPLEST STRATEGY — switching!

$$\tilde{a} = \begin{cases} \pi^{task} & \text{if system is safe} \\ \pi^{safe} & \text{if safety @ risk} \end{cases}$$

Q How do we know safety is @ risk, and the safe ctrl.?

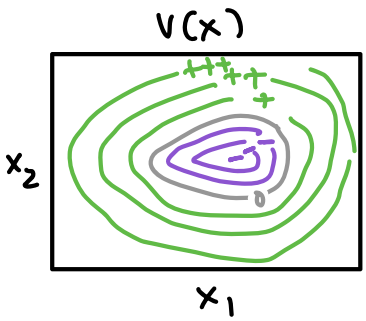
Thought exercise: biking through Frick Park ☺

A Safety @ risk when you are doomed despite best effort!



* Take Embodied AI Safety to learn more! ☺

For the purposes of this class*, assume I give you a priori the safe set & unsafe set as encoded in a function $V(x)$:

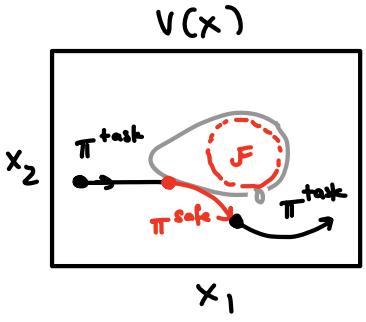


$$\begin{cases} V(x) > 0 & \text{if in safe set} \\ V(x) = 0 & \text{if on boundary} \\ V(x) < 0 & \text{if in unsafe set} \end{cases}$$

Now, our switching safety can be:

$$\tilde{a} = \begin{cases} \pi^{task} & \text{if } V(x) > 0 \quad \checkmark \text{safe for all time!} \\ \pi^{safe} & \text{if } V(x) = 0 \quad \text{⚠ point of no return!} \end{cases}$$

aside: getting π^{safe} outside class scope but intuitively it returns actions that maximize the value



Here is a visual of how this switching safety filter may work.

BUT, if π^{task} is person, this discrete switching is super jarring!

Another idea: **Oh et. al. "Human-centered Safety Filter", RSS '25**

Let's smoothly adjust π^{task} when π^{task} is person!

ALGO At each timestep, safety filter solves:

$$\tilde{a} = \min_{a \in A} \| a - a_H \|_2^2 \quad \text{s.t. } V(f(s, a)) \geq 0$$

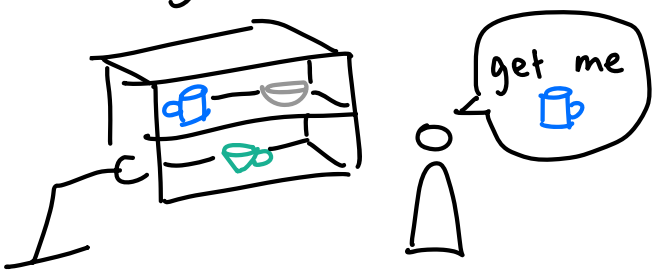
- $a_H = \pi^{task}$: human's action
- $\| a - a_H \|_2^2$: minimally deviate
- $\tilde{a}$: "edited" action
- s: current state
- $f(s, a)$: 1-step dynamics
- $V(f(s, a)) \geq 0$: need to always be in safe set!

Intuition: minimally edit human's actions s.t. the actions are still safe!

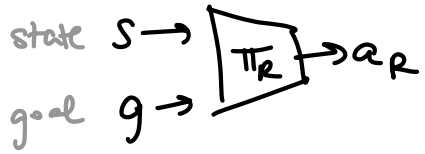
! This is a type of shared controller ☺

This type of safety is good, but its enforced @ the low-level action-space ⇒ can safety filtering be useful @ higher levels of abstraction beyond actions?

Running example:



Robot is running goal-conditioned policy:



(ex. trained via BC)

⚠ This $\pi_R(a|s,g)$ is not guaranteed to safely succeed!

ex. cup could be out of reach, π_R could be imprecise and drop / break the cup ...

In **Jeong et. al, "Robots that Suggest Safe Alternatives", IROS '25** we thought robots should know their capabilities – what they can / can't safely do – and suggest alternative goals when they can't safely succeed @ what person asked!

IDEA: alternative suggestion can be modeled as safety filtering in goal space, rather than action space:

BEFORE:

$$\tilde{a} = \min_{a} \| a - a_H \|_2^2$$

$$\text{s.t. } V(f(s,a)) \geq 0$$

"Find an action that is similar to the person's as long as next state is safe"

NOW:

$$\tilde{g} = \min_{g} d(g, g_H)$$

$$\text{s.t. } V^{\pi}(s, g) \geq 0$$

(distance function)

"Find a goal that is similar to the human's original goal as long as the robot's policy stays safe"

