Last Time

- safety filters

This Time:

- computing safety filters
- HJ reachability

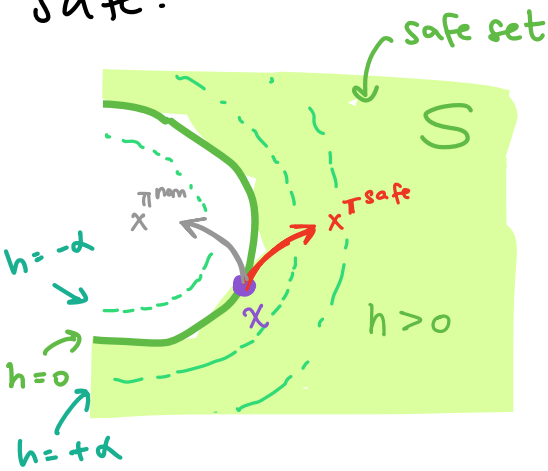
Lecture 3

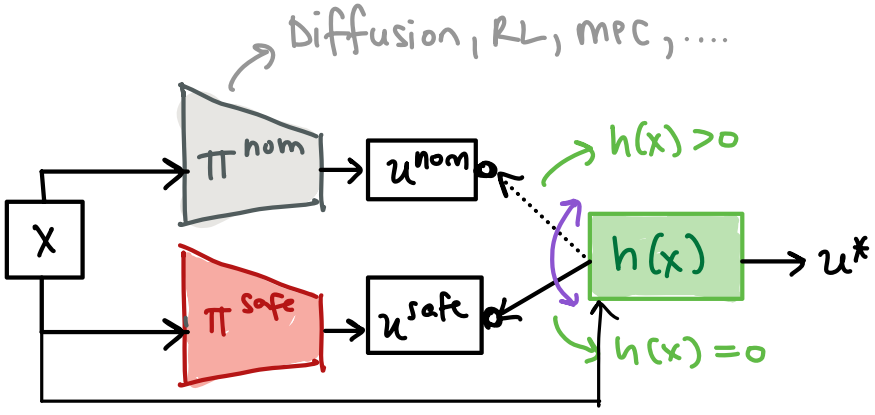
EAIS SP'25

Andrea Bajcsy

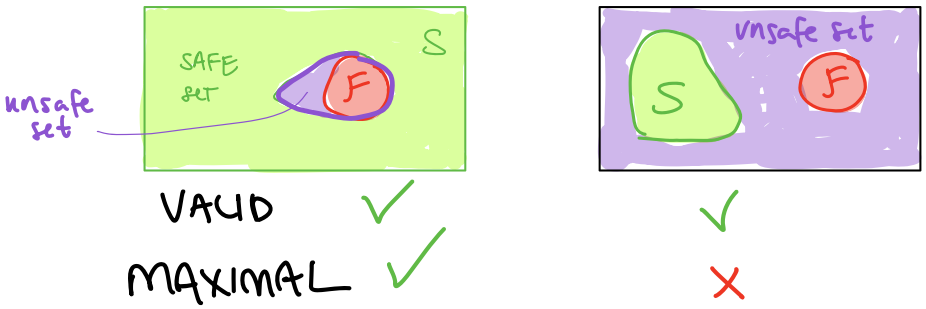
Recall that last lecture we had this idea of a safety filter

It monitors and minimally modifies a base policy to keep it safe:





⚠ Obtaining a VALID and not overly conservative safe set S & $h(x)$ is really challenging for general systems



VALID ✓ ✓

MAXIMAL ✓ ✗

Today, we will tackle this challenge head-on.

Synthesizing (i.e computing) safe sets & safety filters

↳ we will talk about a general, computational framework for getting S and π^{safe} and $h(\cdot)$ that

TODAY → ① is guaranteed to be VALID & MAXIMAL

NEXT → ② naturally handles disturbances robustly

NEXT-NEXT ③ is compatible w/ modern comp. tools { RL, SSL } !

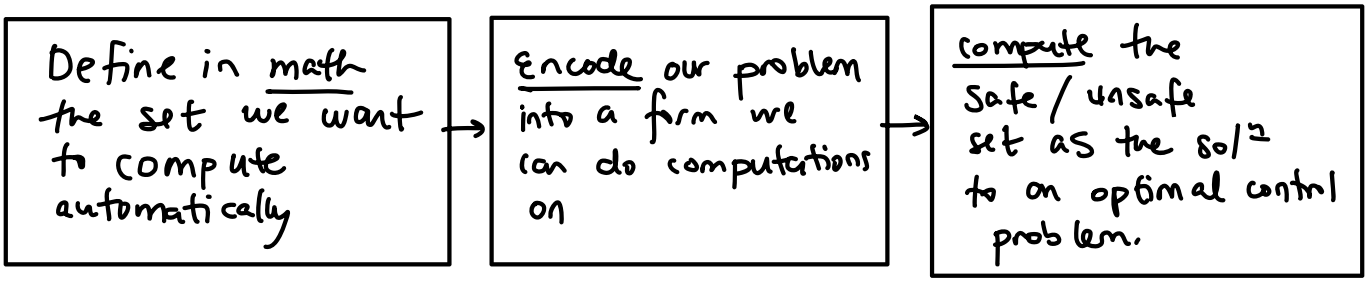
Formalize safety via reachability

We want to compute optimal controllers that ensure our robot never enters failure, AND figure out from which initial conditions is the robot doomed to fail in the future. These questions / objective fall under something called

"reachability analysis".

This is the fundamental problem of identifying "if a certain state of a system is reachable from an initial state of the system."

Safety Analysis Roadmap.

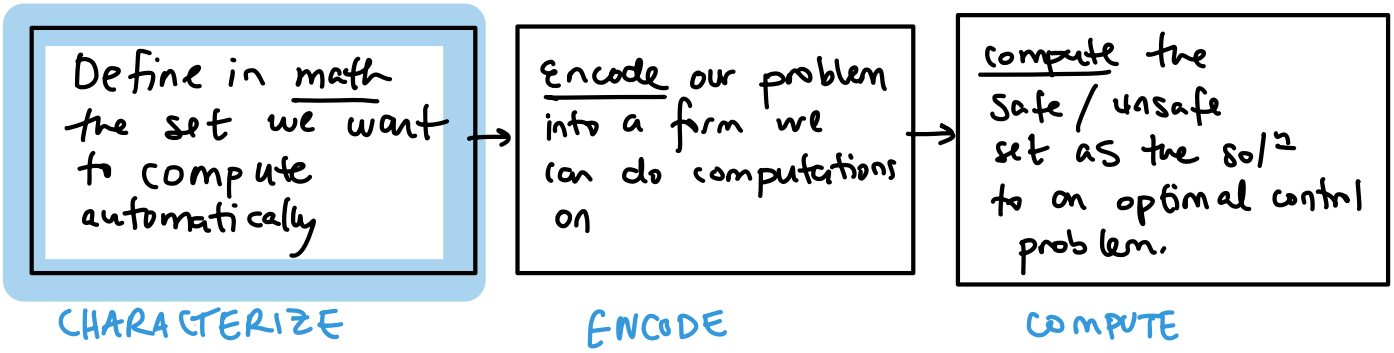


While there are many ways to compute the safe/unsafe set, we will primarily study

Hamilton-Jacobi (HJ) Reachability

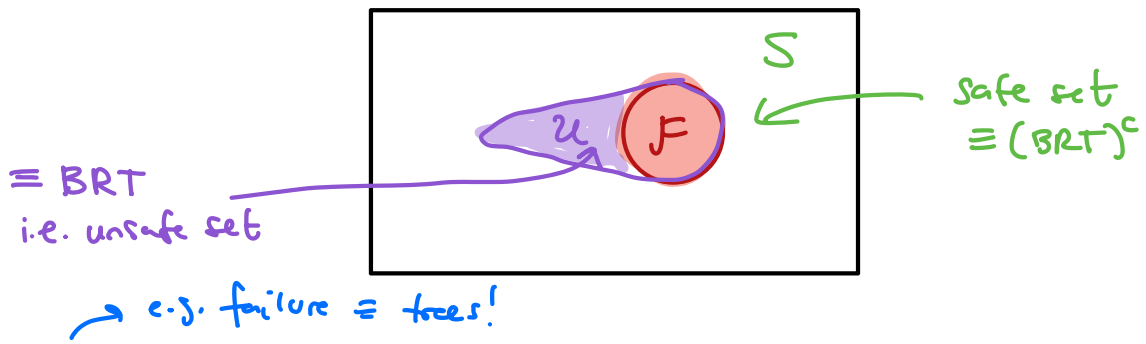
Some nice properties of this paradigm:

1) encode control bounds & state constraints
2) automatically gives both safe sets AND safe policy
3) general nonlinear dyn. systems
4) robustness to uncertainty / other agents + adversaries.



How do we mathematically describe the safe set S and/or the unsafe set $\mathcal{U}$?

The BACKWARDS REACHABLE TUBE (BRT) of a (failure) set $\mathcal{F}$ and a dynamical system $\dot{x} = f(x,u)$ is precisely the set of all states that will eventually reach $\mathcal{F}$ despite the robot's best control efforts.



Let $\mathcal{F} \subset \mathcal{X}$ be the set of states we want to do analysis over.

Let $BRT(t) \subseteq \mathcal{X}$ at time t (typically unsafe set):

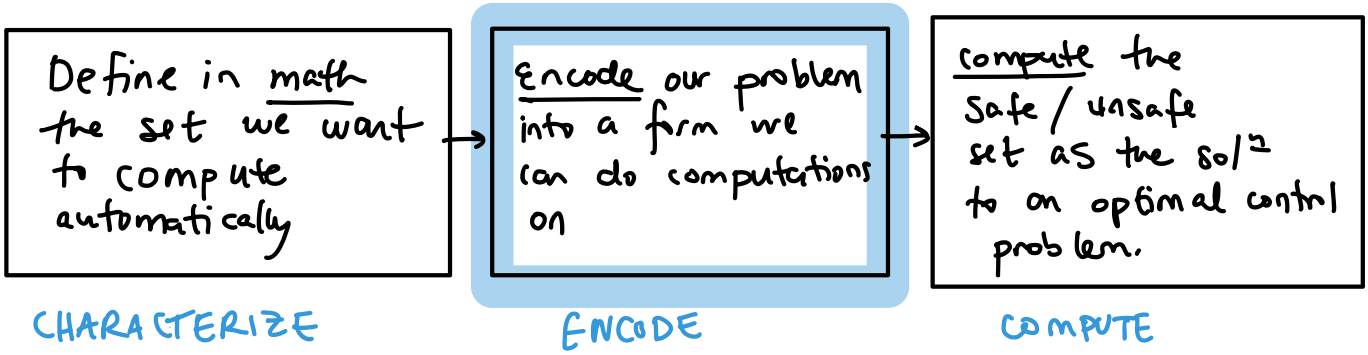
BACKWARDS REACHABLE TUBE (BRT) of set $\mathcal{F} \subset \mathcal{X}$ and system $\dot{x} = f(x,u)$ is:

$$BRT(t) = \{ x \in \mathcal{X} : \forall u(\cdot) \in \mathcal{U}_t^T, \; x_{x,t}^{u(\cdot)}(\tau) \in \mathcal{F} \text{ for some } \tau \in [t,T] \}$$

- $BRT(t)$: set of initial states s.t. ...
- $x \in \mathcal{X}$: initial states
- $\forall u(\cdot) \in \mathcal{U}_t^T$: for all ctrl. signals
- $x_{x,t}^{u(\cdot)}(\tau) \in \mathcal{F}$: the state traj. enters the set $\mathcal{F}$
- for some $\tau \in [t,T]$: at some pt. in time τ

this is the math. def^n of being "doomed"! ∴

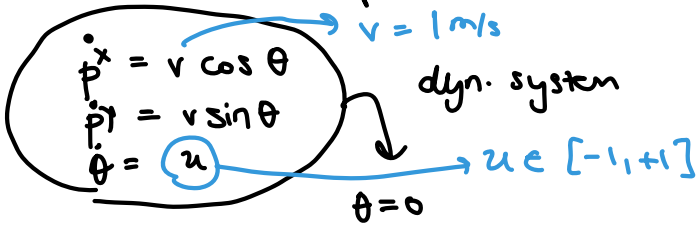
Highlights: Failure set ($\mathcal{F}$) ≡ safety constraints
Unsafe set ($\mathcal{U}$) ≡ BRT



HJ Reachability.

We know connection btwn. the BRT & the failure set; lets talk about computing the BRT!

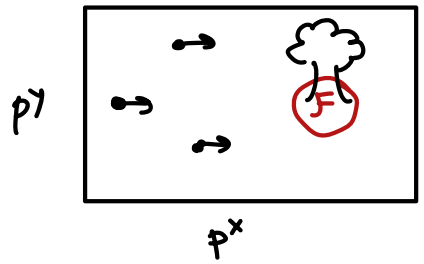
HJ Reachability uses LEVEL SET METHODS to convert the BRT / constraint satisfaction problem into an optimal ctrl. problem.



Here's the process:

(1) We have the failure set $\mathcal{F} \subset \mathcal{X}$

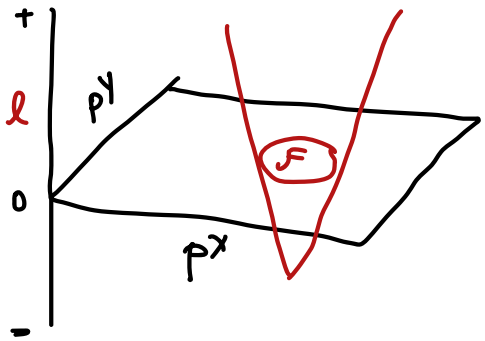
(2) Define a function $\ell(x): \mathcal{X} \to \mathbb{R}$ to implicitly represent this failure set:



$$\ell(x) < 0 \iff x \in \mathcal{F}$$

e.g. signed dist. func. is what we use in practice!

signed-dist. > 0 when x outside $\mathcal{F}$
signed-dist < 0 when x inside $\mathcal{F}$
signed-dist $= 0$ when x on $\partial\mathcal{F}$

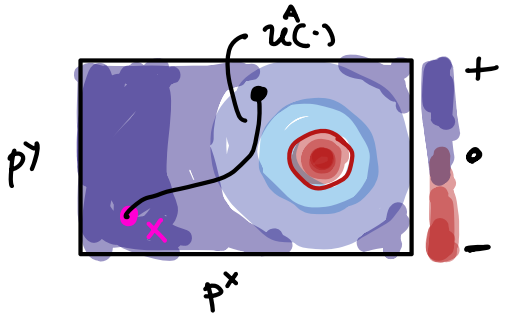


(3) Now, we want to optimize $u(\cdot)$ with respect to $\ell(x)$ since this is our optimal control cost function!

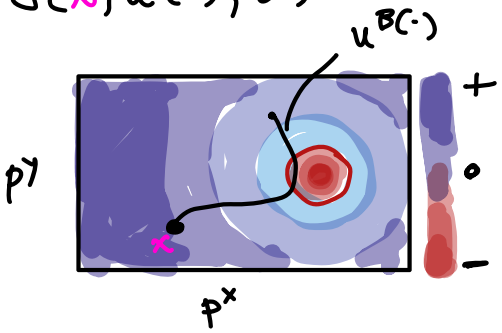
$$J(x, u(\cdot), t) := \min_{\tau \in [t,T]} \ell(x^u_{x,t}(\tau))$$

("closest our system got to failure when applying $u(\cdot)$ and starting from x")

$J(x, u^A(\cdot), t) > 0$



$J(x, u^B(\cdot), t) < 0$



⊛ By looking @ the sign of the cost $J(\cdot,\cdot,\cdot)$ we can tell if the traj. ever entered $\mathcal{F}$ given $u(\cdot)$!

If we want to stay safe, control should maximize J!

$$V(x,t) := \max_{u(\cdot) \in \mathcal{U}_t^T} J(x, u(\cdot), t)$$

$$= \max_{u(\cdot) \in \mathcal{U}_t^T} \left[\min_{\tau \in [t,T]} \ell(x^u_{x,t}(\tau)) \right]$$

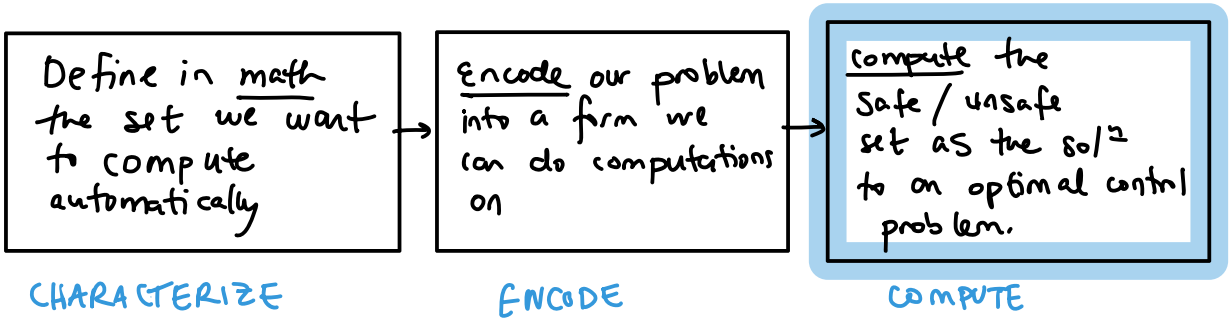
- If $V(x,t) < 0$ for some state x_0, then this means that the controller $u(\cdot)$ tried, but failed, to prevent failure despite its best efforts. $\Rightarrow$ $\boxed{x_0 \in \text{BRT}!}$
- If $V(x,t) \geq 0$ for some x_0, then this means there exists a ctrl signal $u(\cdot)$ that can prevent failure $\Rightarrow$ $\boxed{x_0 \notin \text{BRT}!}$

Once we obtain $V(x,t)$, we also obtain the unsafe set (BRT)!

this is the unsafe set!

$$\boxed{\text{BRT}(t) \equiv \{ x : V(x,t) < 0 \}}$$

Now, we have an optimal ctrl. problem whose solution will automatically give us unsafe set (BRT) — how do we solve?



Hmm, but the min-over-time is not the usually running cost...

Good news — principle of dyn. programming still works!

$$V(x,t) := \max_{u(\cdot)} \min_{\tau \in [t,T]} \ell(x(\tau))$$

(state traj indexed @ τ)

$$= \max_{u(\cdot)} \min \left\{ \min_{\tau \in [t, t+\delta]} \ell(x(\tau)),\ \min_{s \in [t+\delta, T]} \ell(x(s)) \right\}$$

(split in time)

either the min happens "now" — or it will happen in the future

$\vdots$

SKIPPED for now (hint: apply principle of D.P ↑)

Eventually, you recover:

HJ Variational Inequality (HJ-VI)

$$\min\left\{ \ell(x) - V(x,t),\ \frac{\partial V}{\partial t} + \max_{u \in \mathcal{U}} \frac{\partial V}{\partial x} \cdot f(x,u) \right\} = 0$$

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

($\ell(x) - V(x,t)$: keeps track of entering $\mathcal{F}$; $\frac{\partial V}{\partial t}$: change in value over time; $\max_{u \in \mathcal{U}} \frac{\partial V}{\partial x} \cdot f(x,u)$: best value change via action u^*; $f(x,u) \equiv \dot{x}$)

↑ continuous time ($t \in \mathbb{R}$) ↓ discrete time ($t \in \mathbb{Z}$)

$$V_t(x) = \min\left\{ \ell(x),\ \max_{u \in \mathcal{U}} V_{t+1}(\underbrace{f(x,u)}_{\equiv x_{t+1}}) \right\}$$

$V_T(x) = \ell(x)$

("remember if failing" — try best not to fail in future)