Lecture 4

EAIS SP'25

Andrea Bajcsy

Last Time

- ☐ computing safety filters
- ☐ HJ Reachability

This Time:

- ☐ Recap + hands-on exercise
- ☐ code demo!
- ☐ robust safety

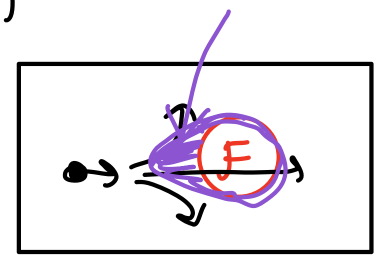
$\boxed{\text{Recap}}$ + HJ Reachability for Computing Safety Filters

Last time, we formalized the problem of safety satisfaction as an optimal control problem:

Continuous-time Safety Problem

signed dist. func. s.t. $\mathcal{F} = \{x : \ell(x) < 0\}$

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



$$\Downarrow$$

"Safety Bellman Backup" (i.e. HJ-VI)

$$\min\Big\{\overbrace{\ell(x) - V(x,t)}^{\text{"remember if failing now"}},\ \underbrace{\frac{\partial V}{\partial t} + \max_{u \in \mathcal{U}} \frac{\partial V}{\partial x} \cdot f(x,u)}_{\text{"try best not to fail in future"}}\Big\} = 0$$

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

Discrete-time safety Problem

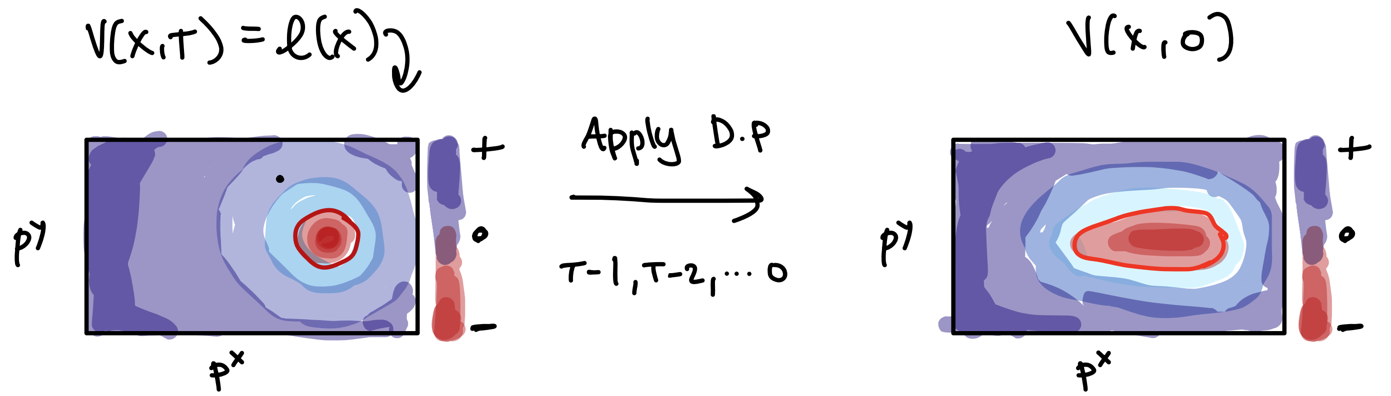
$$V_t(x_t) := \max_{u_t, u_{t+1} \cdots u_T} \min_{\tau \in \{t, t+1, \dots T\}} \ell(x_\tau)$$

$$\Downarrow$$

"Safety Bellman Backup" (discrete-time)

$$V_t(x_t) = \min\Big\{\overbrace{\ell(x_t)}^{\text{remember failing now"}},\ \underbrace{\max_{u_t \in \mathcal{U}} V_{t+1}(f(x_t, u_t))}_{\text{"try best not to fail in future"}}\Big\}$$

$$V_T(x_T) = \ell(x_T)$$



Getting our safety monitor (i.e. the unsafe set boundary)

By design, the zero-sublevel set of $V(\cdot,\cdot)$ encodes our <u>unsafe set</u>!

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

Getting safety-preserving policy

We can also compute the <u>optimal safety policy</u> via the optimal value function too!

<u>Cont. time:</u>

$$u^{safe}(x,t) = \arg\max_{u \in \mathcal{U}} \frac{\partial V}{\partial x} \cdot f(x,u)$$

<u>discr. time:</u>

$$u_t^{safe}(x_t) = \arg\max_{u_t \in \mathcal{U}} V_{t+1}(f(x_t, u_t))$$

Safety Filter

$$u(x) = \begin{cases} \pi^{nom} & \text{if } V > 0 \\ u^{safe} & \text{if } V \approx 0 \end{cases}$$

$\boxed{\text{Exercise}}$: Compute a Backwards Reachable Tube (i.e. unsafe set)

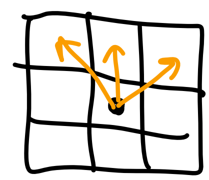
We will operate in discrete state + time to build intuition:

"Safety Bellman" Eqn:

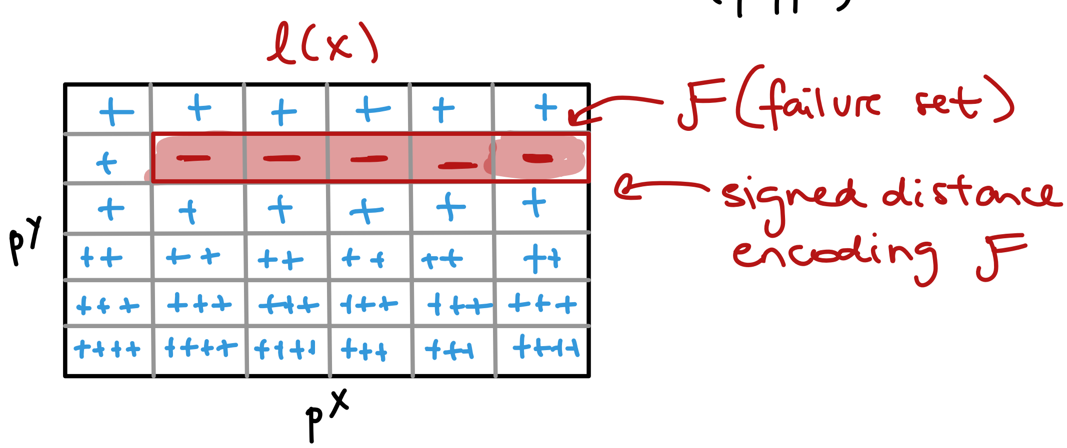
$$V_t(x) = \min \left\{ \ell(x), \max_{u \in \mathcal{U}} V_{t+1}(f(x,u)) \right\}$$

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

- system state: $x = \begin{bmatrix} p^x \\ p^y \end{bmatrix} \in \mathcal{X}$
- system ctrl: $u \in \mathcal{U} = \{ \nwarrow, \uparrow, \nearrow \}$

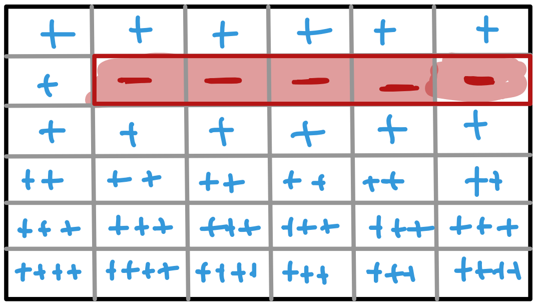


(p^x, p^y)

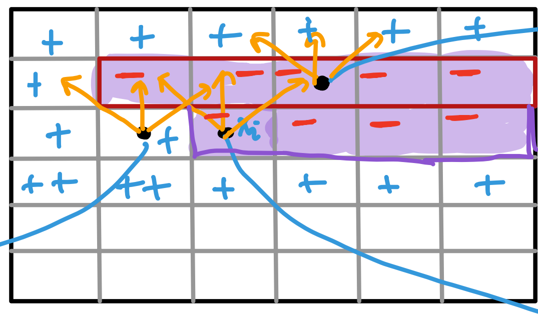


We initialize $V_T(x) = \ell(x)$. Compute $V_{T-1}(x)$ and $V_{T-2}(x)$.

Solution:



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

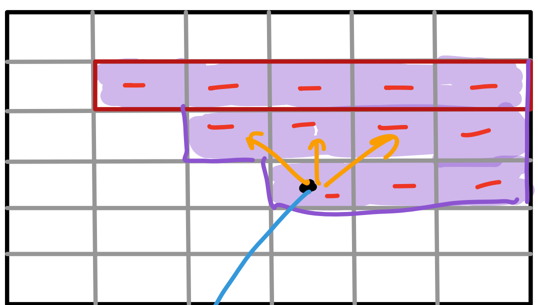


$V_{T-1}(x_k) = \min\{-, +\} = -$

$V_{T-1}(x_{T-1})$

$V_{T-1}(x_j) = \min\{+, +\} = +$

$V_{T-1}(x_i) = \min\{\overset{\ell(x_i)}{+}, \overset{\max V_T(f(x_i, u))}{-}\} = -$



$V_{T-2}(x_{T-2})$

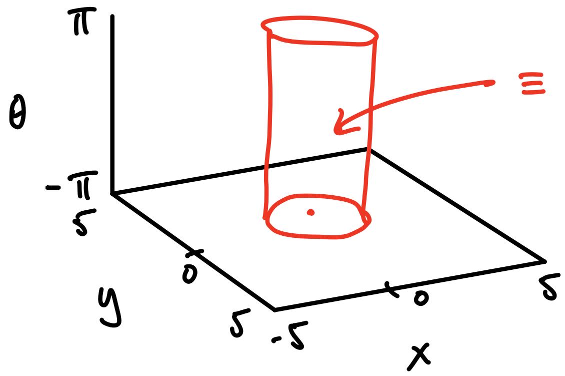
$V_{T-2}(x_m) = \min\{++, -\} = -$

Code Demo (in MATLAB solver helperOC + levelSetToolbox)

• system:

$$\left.\begin{aligned} \dot{x} &= v\cos\theta, \quad v = 1.5\ m/s \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= u, \quad u \in [-0.5, 0.5] \end{aligned}\right\} \text{Dubins' car}$$

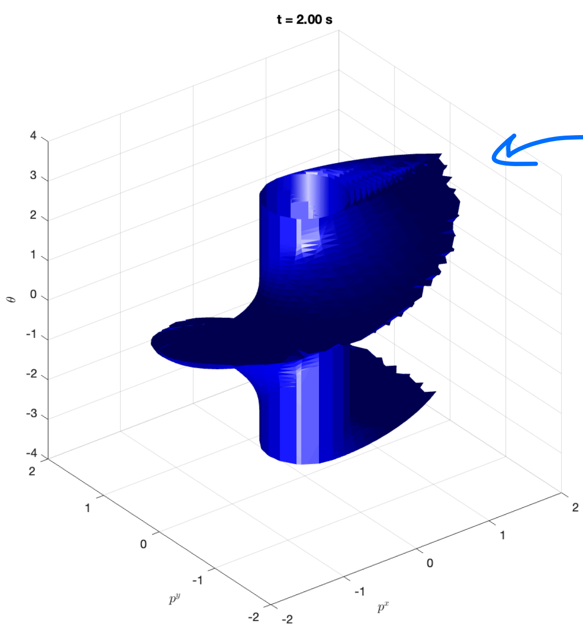
• failure set:



$$\equiv \mathcal{F} = \{(x,y,\theta) : \forall \theta \in [-\pi,\pi],\ \|(x,y)\|_2^2 < 0.5^2\}$$

"you have failed if you enter the circle @ origin with radius 0.5 at ANY orientation"

• computed unsafe set (i.e. backwards reachable tube):



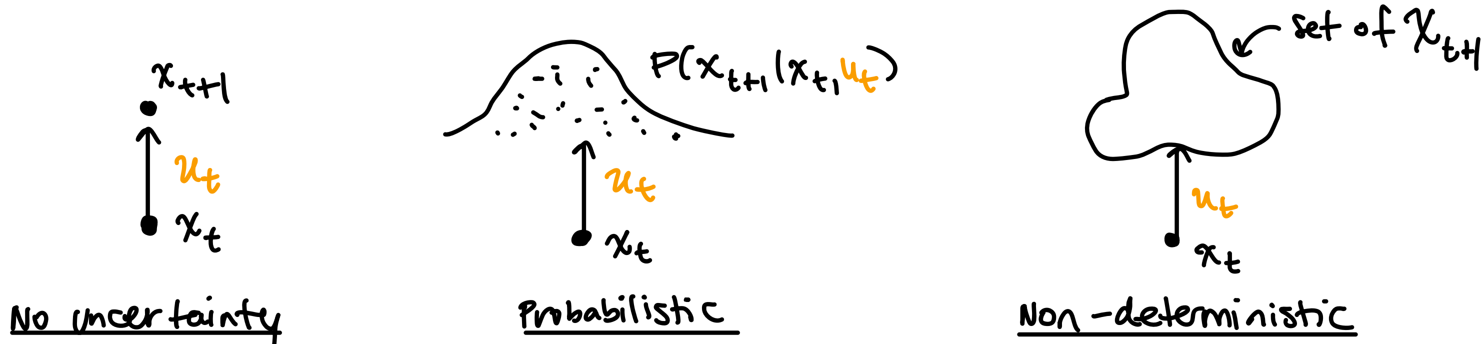
BRT of dubins' car!

(!) Way easier as an engineer to specify failure set $\mathcal{F}$ than to specify BRT. That's why we want to compute BRT given $\mathcal{F}$!

Robustifying Safety

So far, we have assumed that our dynamical system perfectly evolves via $\dot{x} = f(x,u)$ with no uncertainty.

This isn't realistic for many real-world scenarios (e.g. friction!)



There are two ways to model uncertainty:

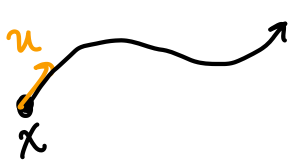
1) probabilistic uncertainty (i.e. "I have observed data")

2) non-derministic uncertainty (i.e "I have minimal additional info.")

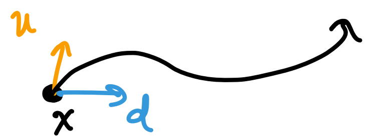
How should we handle the design of our safety filter (+ analysis) to handle uncertainty?

Typically, we do this via modelling another "input" that influences the state evolution:

BEFORE: → **NOW:**



$$\dot{x} = f(x, u)$$



$$\dot{x} = f(x, u, d)$$

u: "robot" "ego"

d: "disturbance"

In robust safety, we take a non-deterministic view of uncertainty and assume that $d \in \mathcal{D}$ is chosen from some bounded set and we want our robot to be <u>ROBUST</u> to the <u>WORST POSSIBLE</u> sequence of d's!

<u>ROBUST BACKWARDS REACHABLE TUBE</u> (BRT) of a set $\mathcal{F} \subset \mathcal{X}$ and dynamical system $\dot{x} = f(x, u, d)$ is:

$$BRT(t) := \Big\{ x \in \mathcal{X} : \overbrace{\forall u(\cdot) \in \mathbb{U}_t^T}^{\text{for all things the robot could do}},\ \overbrace{\exists\, d(\cdot) \in \mathbb{D}_t^T}^{\text{there is something the disturbance could do ....}},\ \underbrace{\mathbf{x}_{x,t}^{u,d}(\tau) \in \mathcal{F}}_{\text{that leads the robot to failure}} \text{ for some time } \tau \in [t, T] \Big\}$$

This is the set of all starting states from which no matter the controller's effort, the disturbance can push system into $\mathcal{F}$.

The way we will formulate an "optimal control" problem whose solution represents this unsafe set will be via:

$$\underbrace{\text{zero-sum}}_{\text{there is a winner + loser}} \quad \underbrace{\text{dynamic}}_{\text{game evolves over time}} \quad \underbrace{\text{games}}_{\text{result / outcome depends on 2+ players / inputs}}$$