AA 598B Special Topics

Decision-Making & Control for Safe Interactive Autonomy

Instructor: Prof. Karen Leung

Autumn 2024

https://faculty.washington.edu/kymleung/aa598/

Announcements

- Start your project!
- Homework 3 out this week
  - Will be short
- Long paper discussion on Monday
- Guest speaker on Wednesday

Last time

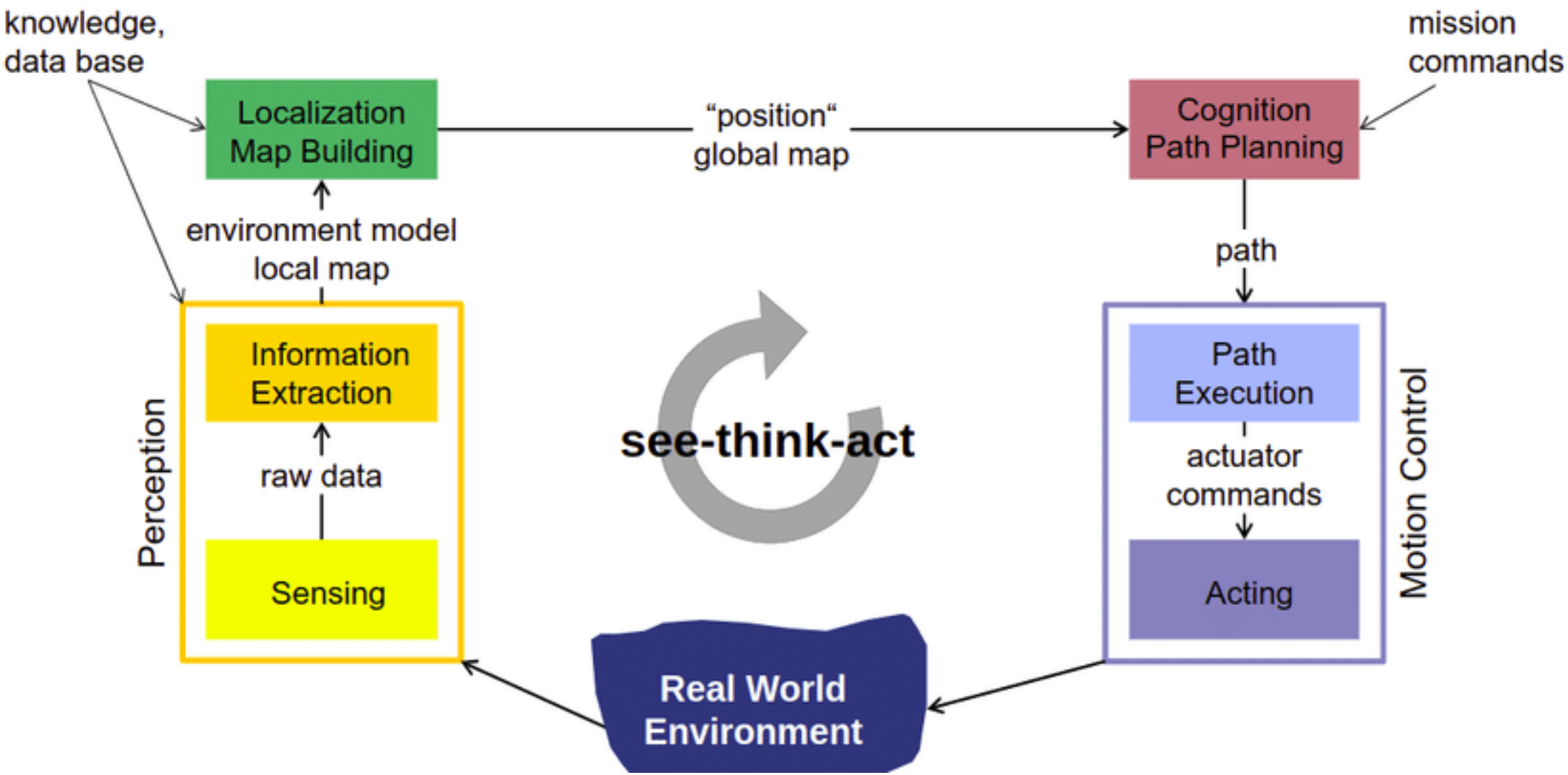
- Wrapped up game theory
- Discussed sampling-based approaches to solving a stochastic trajectory planning problem
  - Monte Carlo estimation of a trajectory tree
  - MPPI as a way to estimate gradients via sampling
  - Hybrid approach: sample then refine
  - Injecting rules into the objective using Signal Temporal Logic

Today

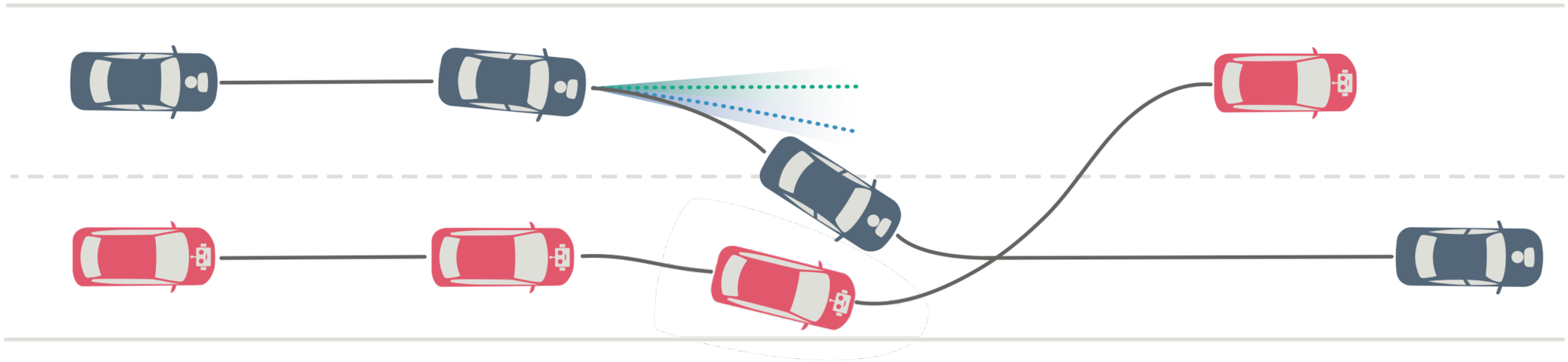
- Start Module #3: Controls
  - Safety
  - Reachability theory
  - Control barrier functions

Safe control

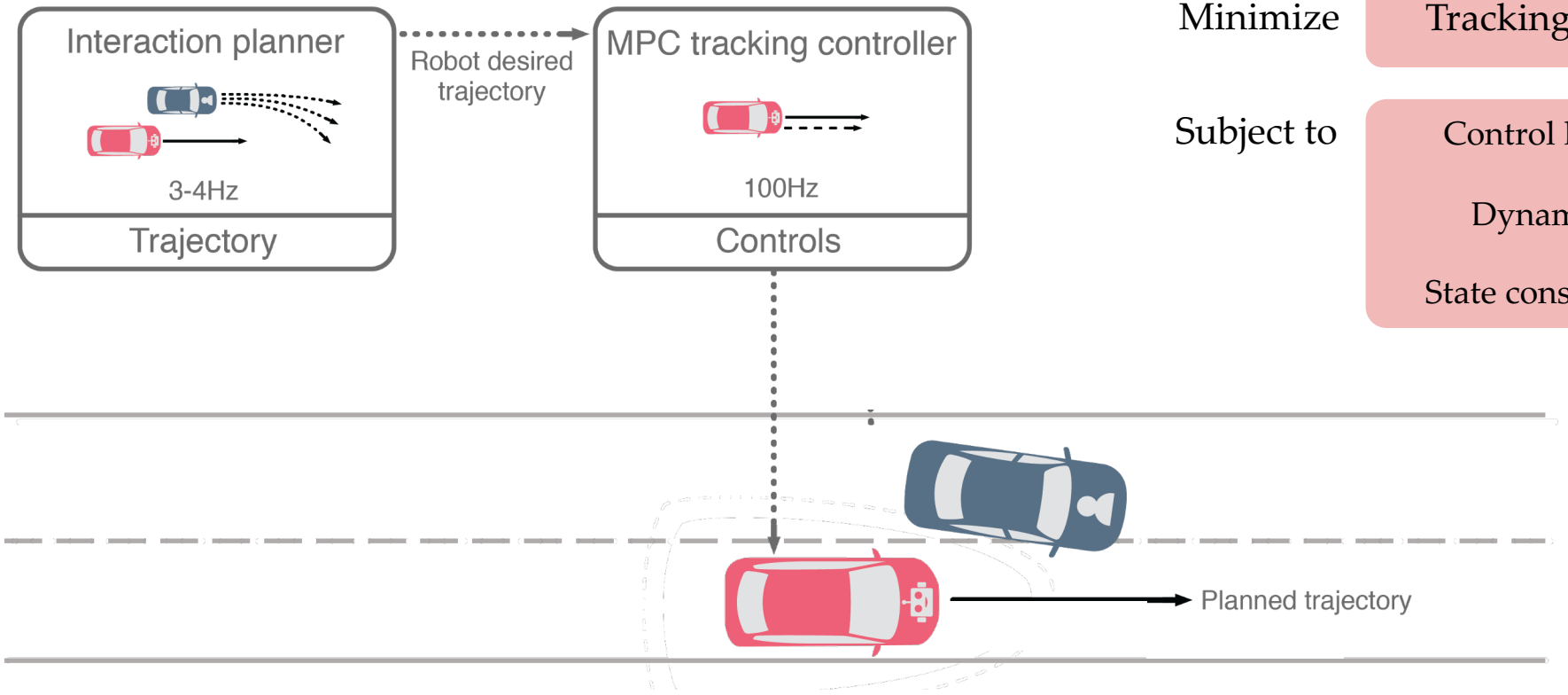
Module #3


knowledge,
data base
Localization
Map Building
"position"
global map
Cognition
Path Planning
mission
commands
environment model
local map
path
Information
Extraction
raw data
Sensing
Perception
see-think-act
Path
Execution
actuator
commands
Acting
Motion Control
Real World
Environment

Motivating example: Traffic-weaving



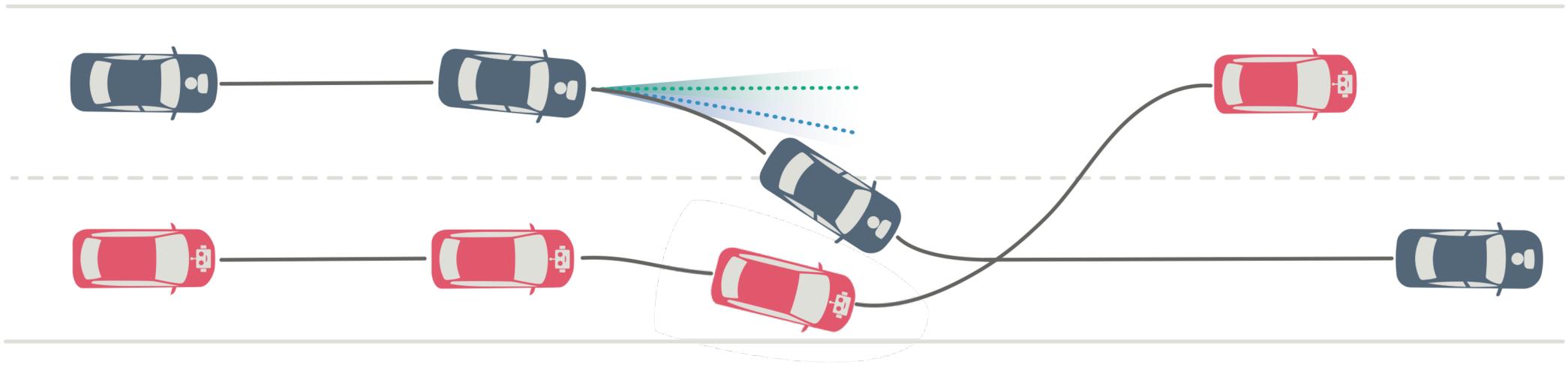
(Standard) Planning and control stack



Minimize: Tracking error

Subject to:
- Control limits
- Dynamics
- State constraints

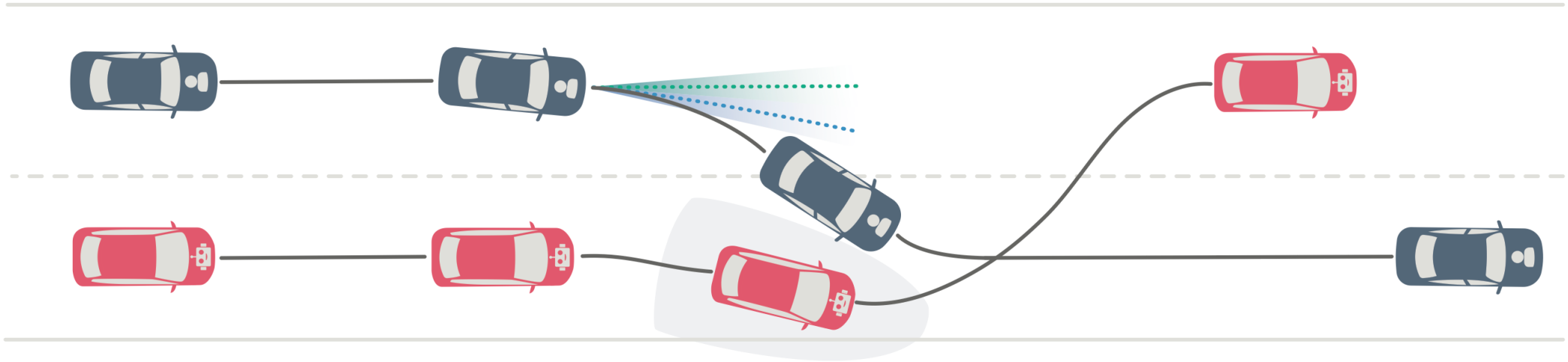
What if the human does something unexpected? What if the prediction model makes a "mistake"? Do we trust our prediction model + planner?



We need a *safety filter*

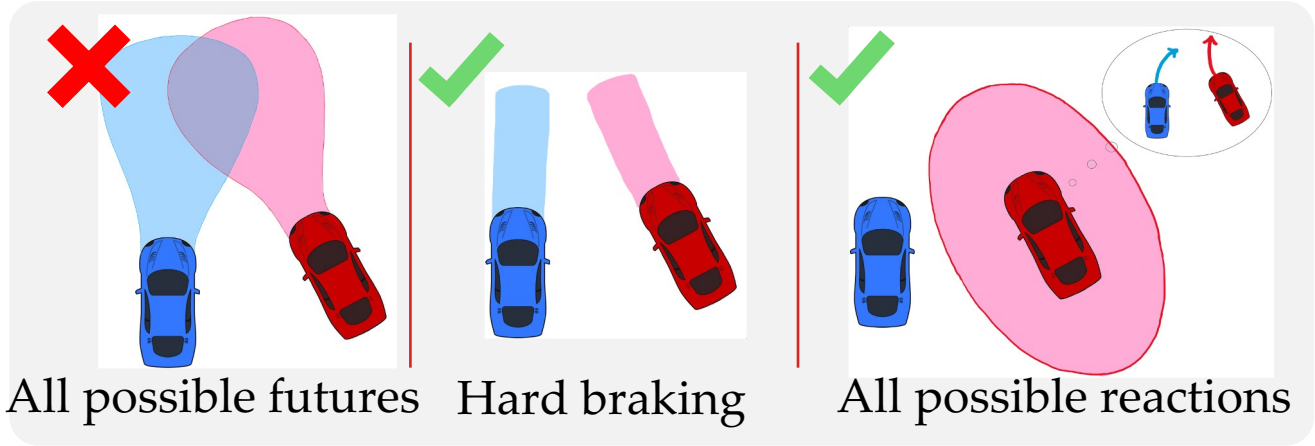
- **Safety filter**: A module that checks the computed controls and adjusts the control as needed.

How to define safety?
What to adjust it to?



To brake or not to brake?

Well, it depends…

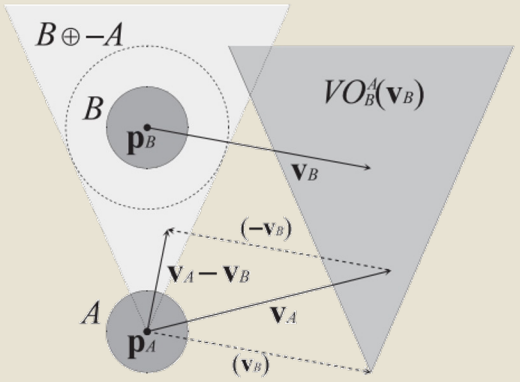


What constitutes as *reasonable* behavior?

Different stakeholders make different assumptions

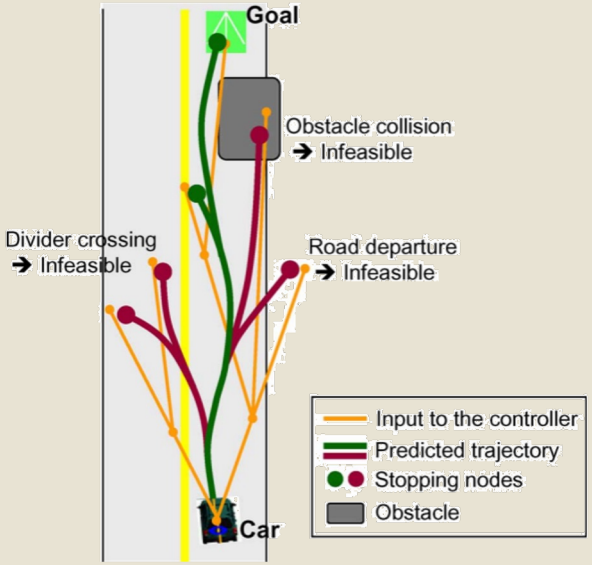
Examples of safe control strategies

Velocity obstacles



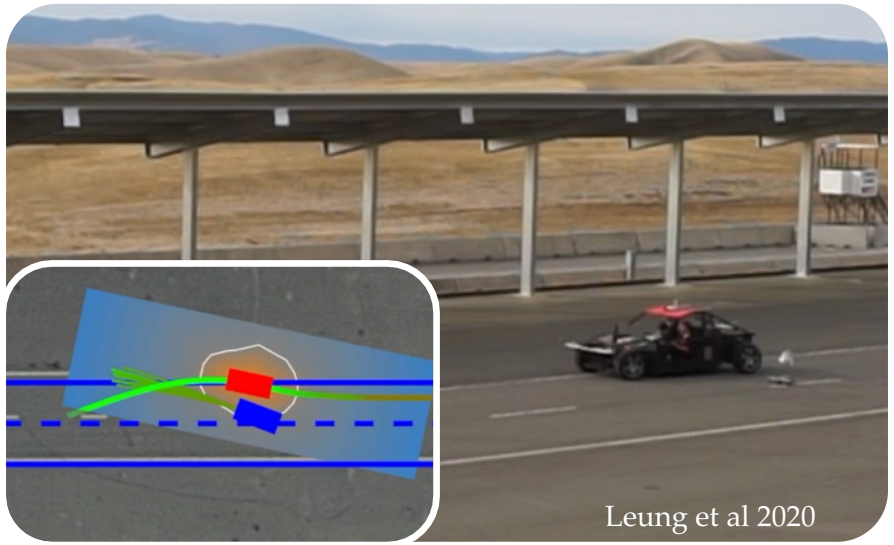
Van der Berg et al 2008

Contingency planning



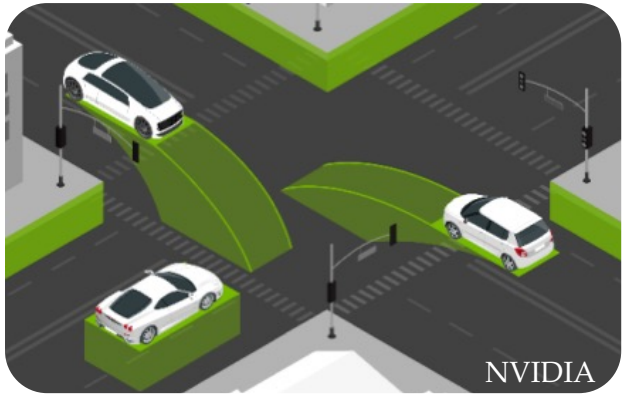
Kuwata et al 2009

Hamilton-Jacobi reachability



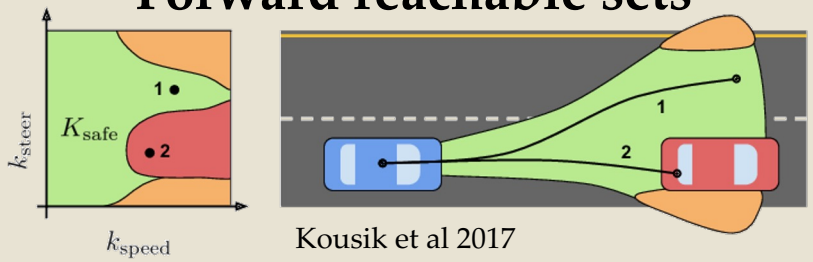
Leung et al 2020

Safety force field (SFF)



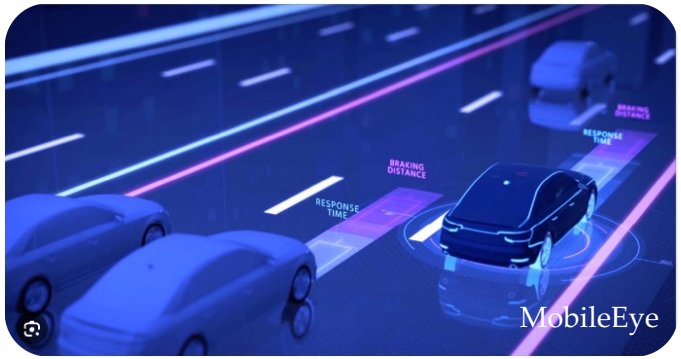
NVIDIA

Forward reachable sets



Kousik et al 2017

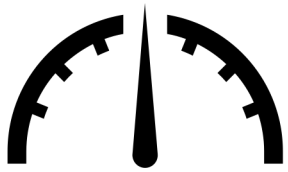
Responsibility sensitive safety



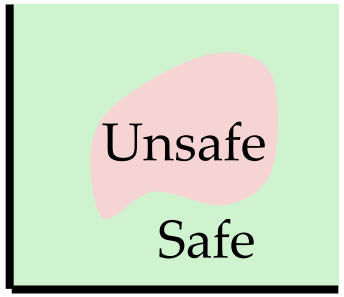
MobileEye

A **safety concept** helps determine (1) how safe and (2) what are allowable safe controls to take

World state



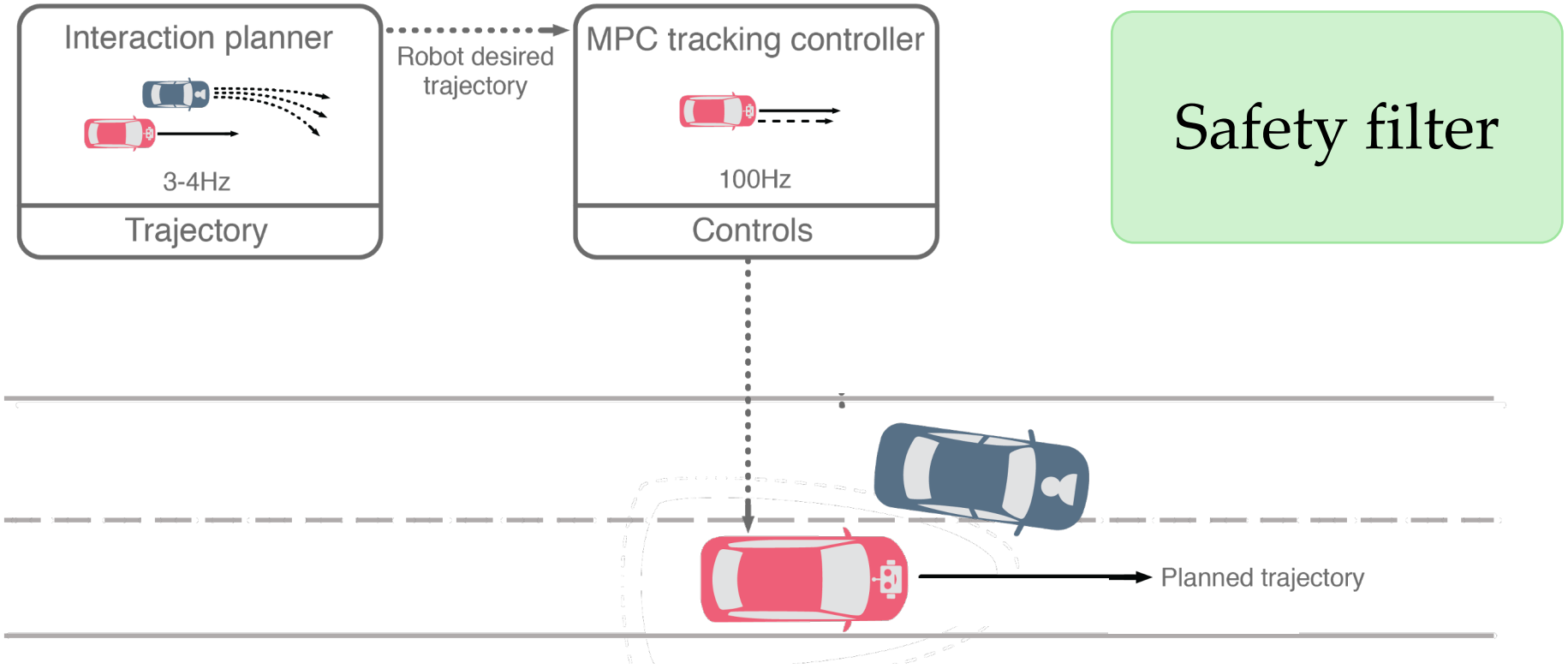
Measure of safety



Set of allowable safe controls

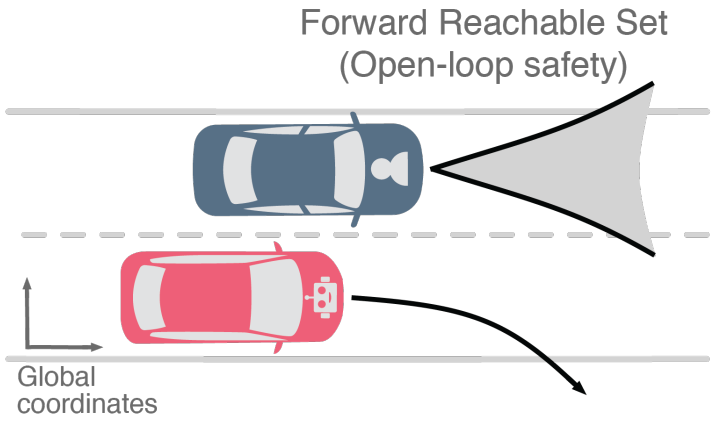
Also, how to *apply* a safety concept to the autonomy stack?

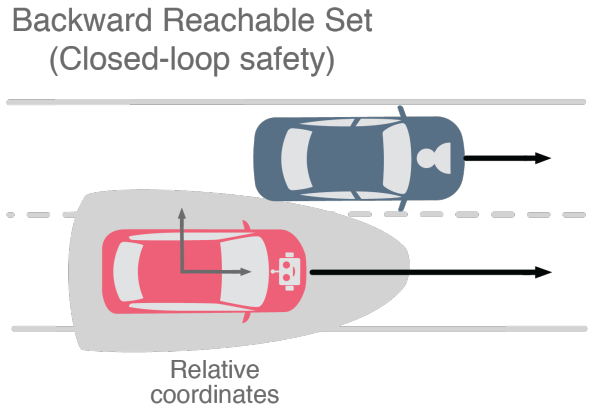
(Safe) Planning and control stack

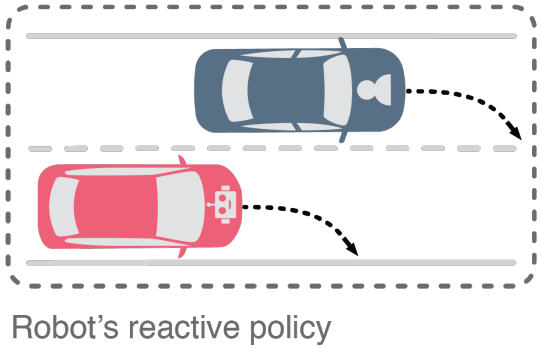


How do we design a safety filter?

Hamilton-Jacobi reachability







Backward reachability

Sequential decision-making
Dynamic programming

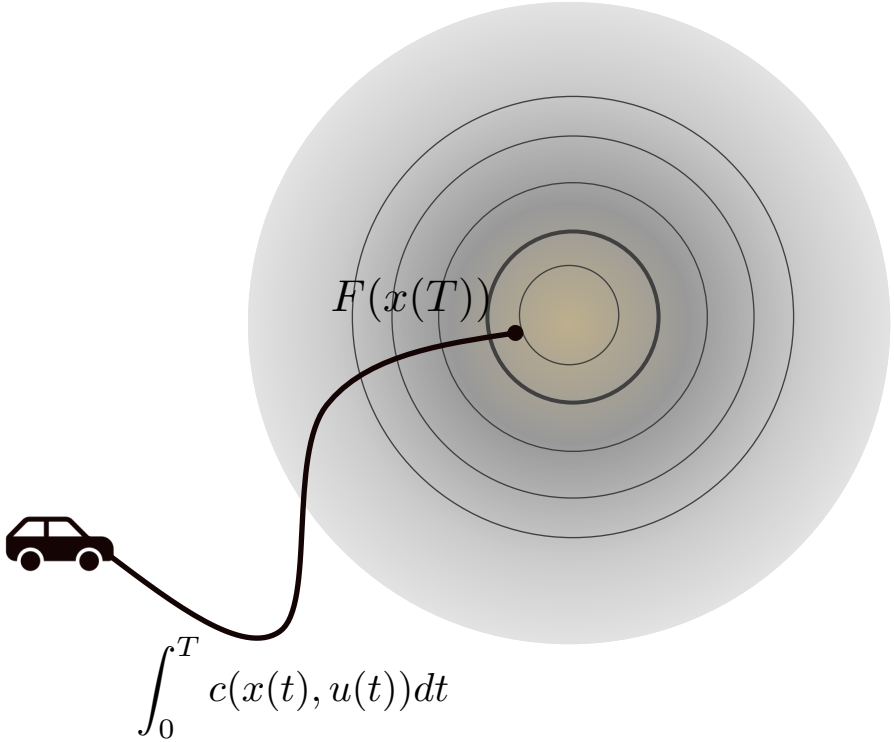
$$\min_u \int_0^T c(x(t), u(t))dt + F(x(T))$$
$$\text{s.t. } \dot{x} = f(x, u)$$
$$u \in \mathcal{U}$$
$$x(0) = x_0$$

Hamilton-Jacobi-Bellman Equation

$$\frac{\partial V(x,t)}{\partial t} + \min_u \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u) + c(x,u) \right\} = 0$$
$$V(x,0) = F(x)$$

But we only care if we reach the target set

$$\min_u \int_0^T c(x(t), u(t))dt + F(x(T))$$
$$\text{s.t. } \dot{x} = f(x, u)$$
$$u \in \mathcal{U}$$
$$x(0) = x_0$$



Backward reachable set

Hamilton-Jacobi-Bellman Equation

$$\frac{\partial V(x,t)}{\partial t} + \min_u \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u) + c(x,u) \right\} = 0$$

$$V(x,0) = F(x)$$

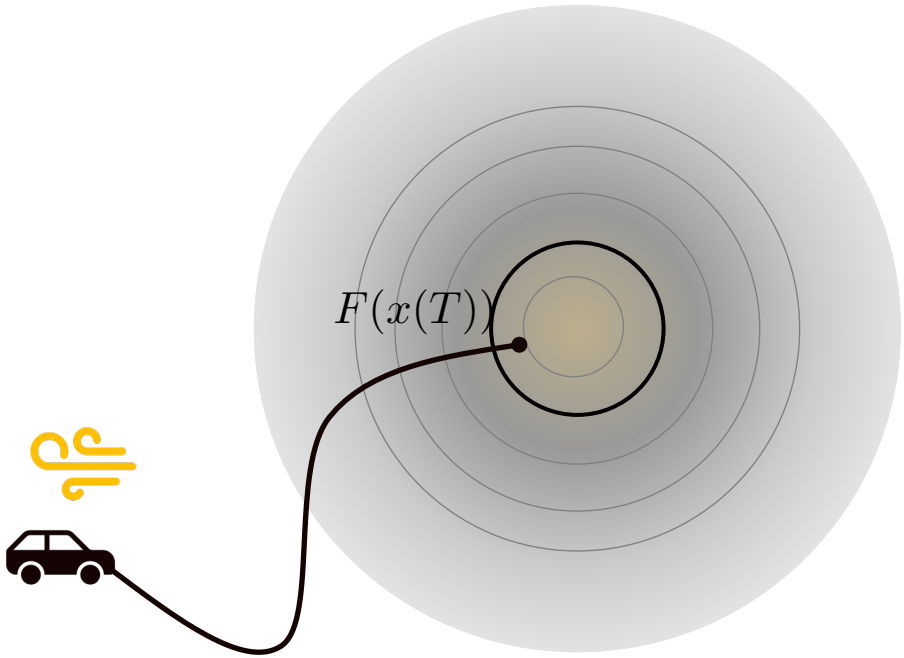
The value function represents the terminal cost of the system at the final time.

$$F(x) \leq 0 \Leftrightarrow x \in \mathcal{T}$$

$$F(x) > 0 \Leftrightarrow x \notin \mathcal{T}$$

$$V(x,-T) < 0 \Rightarrow \text{System will end up in } \mathcal{T}$$

Adding disturbances

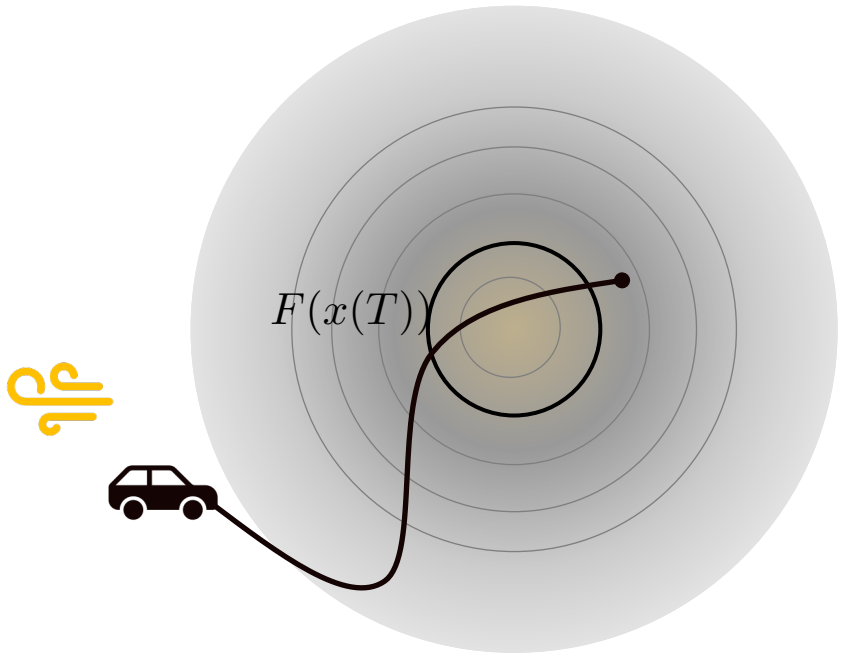


Hamilton-Jacobi-Isaacs Equation

$$\frac{\partial V(x,t)}{\partial t} + \min_{u} \max_{d} \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,d) \right\} = 0$$

$$V(x,0) = F(x)$$

Backward reachable tube



Hamilton-Jacobi-Isaacs Equation

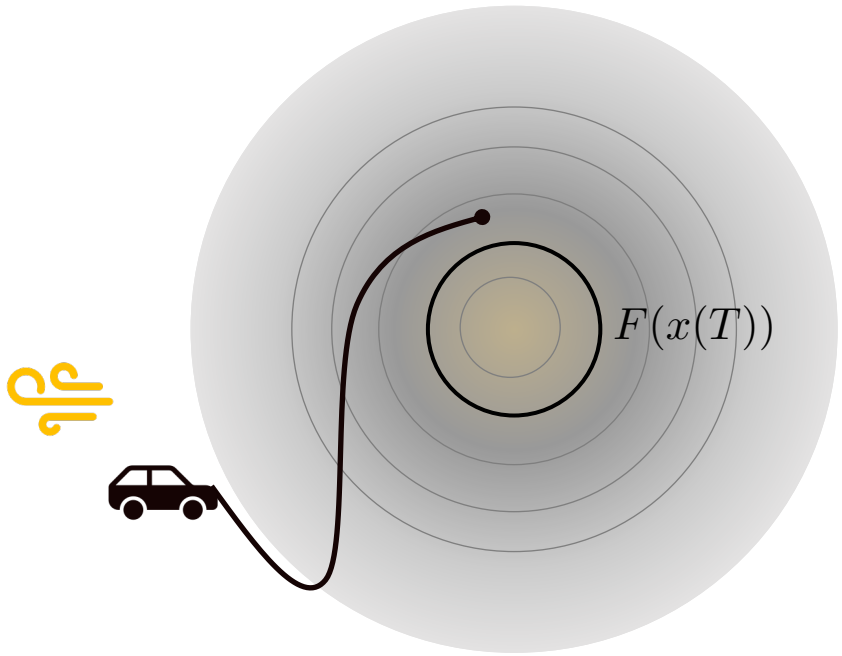
$$\frac{\partial V(x,t)}{\partial t} + \min(0, \mathcal{H}(x,t)) = 0$$

$$\mathcal{H}(x,t) = \min_{u} \max_{d} \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,d) \right\}$$

$$V(x,0) = F(x)$$

"Take the minimum value along the trajectory"

Also consider *avoiding* the target set



Hamilton-Jacobi-Isaacs Equation

$$\frac{\partial V(x,t)}{\partial t} + \max_u \min_d \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u) \right\} = 0$$

$$V(x,0) = F(x)$$

Is this an easy problem to solve?

- Unfortunately, solving this *exactly* suffers from the curse of dimensionality
  - Can use function approximations
- For small systems & control/disturbance dynamics, this is very tractable

HJ reachability JAX toolbox

https://github.com/StanfordASL/hj_reachability

Example system: `Air3d`

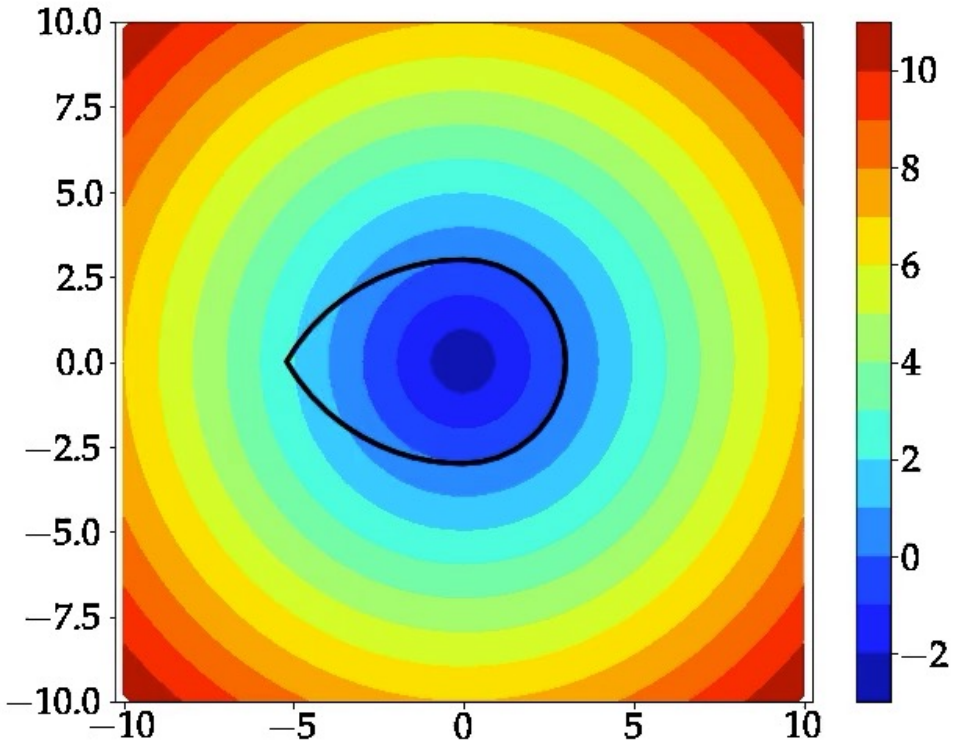
```
dynamics = hj.systems.Air3d()
grid = hj.Grid.from_lattice_parameters_and_boundary_conditions(hj.sets.Box(np.array([-6., -10., 0.]),
                                                                            np.array([20., 10., 2 * np.pi])),
                                                               (51, 40, 50),
                                                               periodic_dims=2)
values = jnp.linalg.norm(grid.states[..., :2], axis=-1) - 5

solver_settings = hj.SolverSettings.with_accuracy("very_high",
                                                  hamiltonian_postprocessor=hj.solver.backwards_reachable_tube)
```

`hj.step`: propagate the HJ PDE from `(time, values)` to `target_time`.

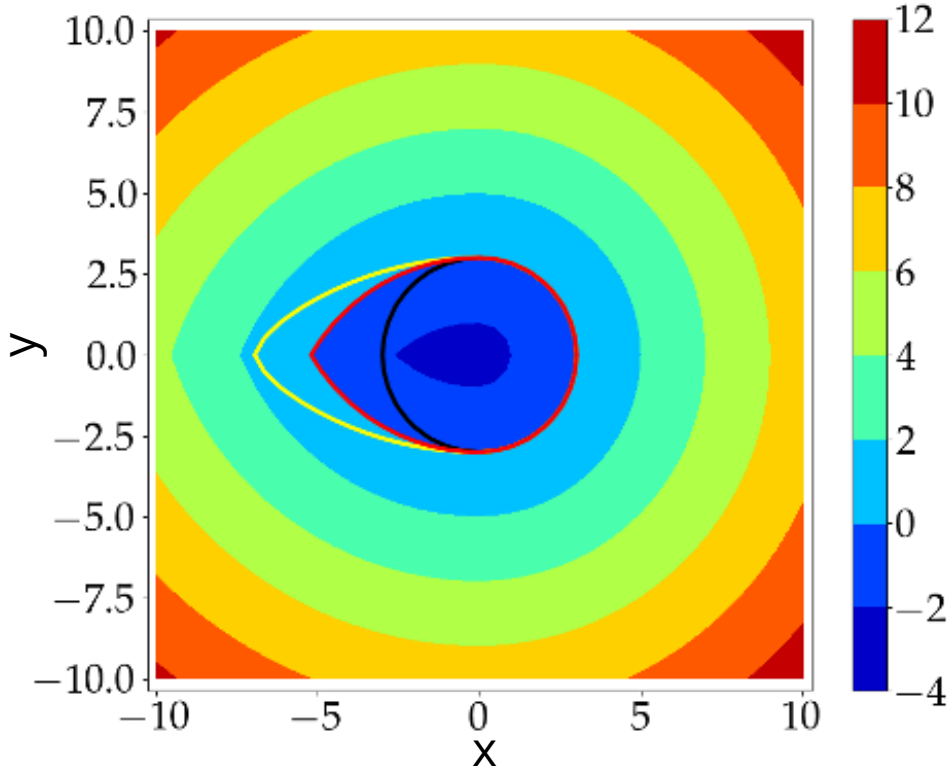
```
time = 0.
target_time = -2.8
target_values = hj.step(solver_settings, dynamics, grid, time, values, target_time)
```

Dubins car example (avoid)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ u+d \end{bmatrix}$$

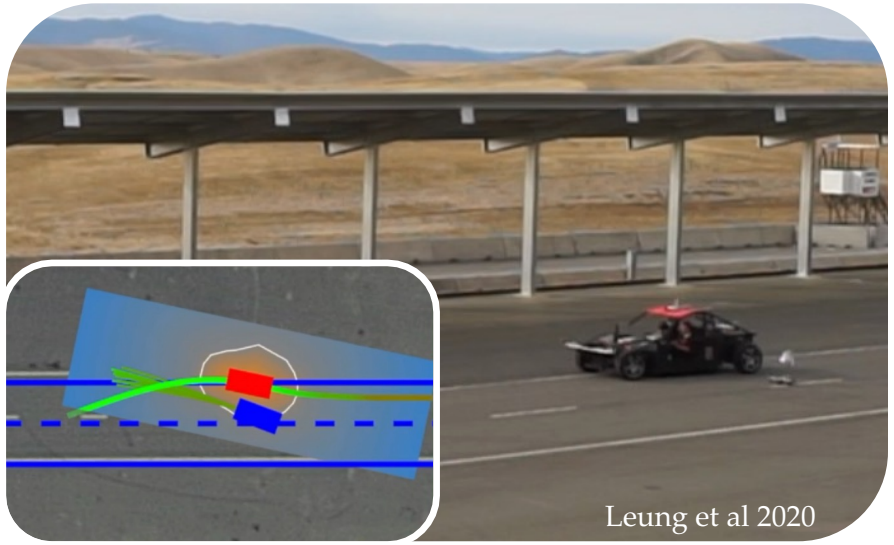


With (yellow) and without (red) disturbance



Can apply to *relative dynamics*

Hamilton-Jacobi reachability



Hamilton-Jacobi-Isaacs partial differential equation

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

$$V(x,0) = v(x)$$

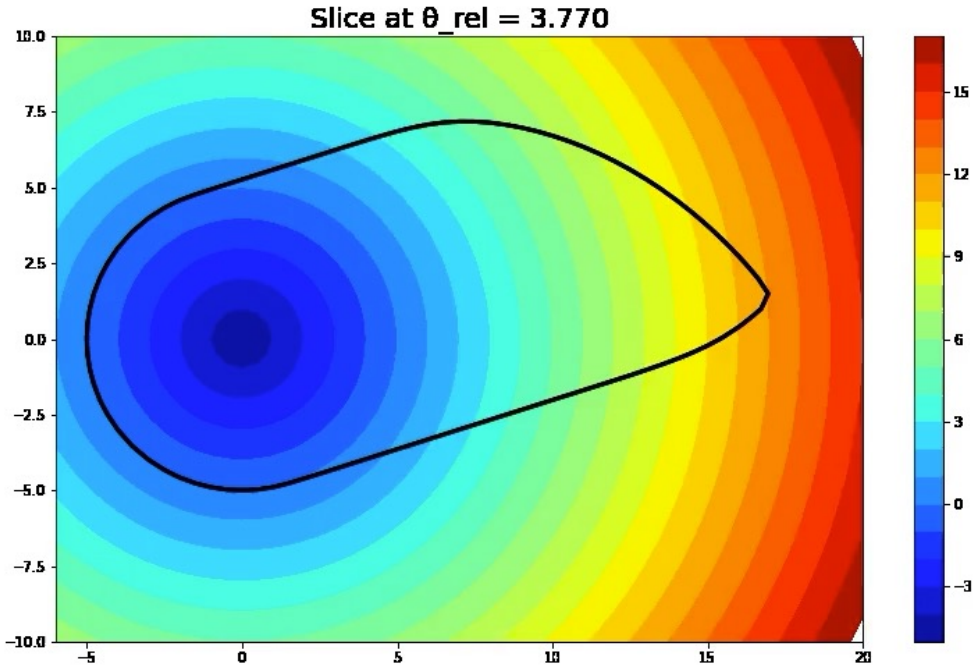
Measure of safety

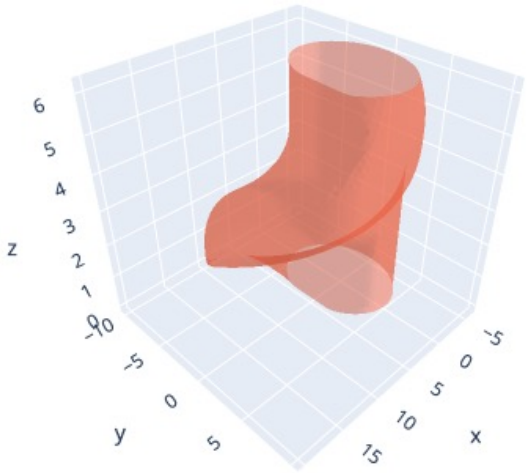
Robot avoids collision

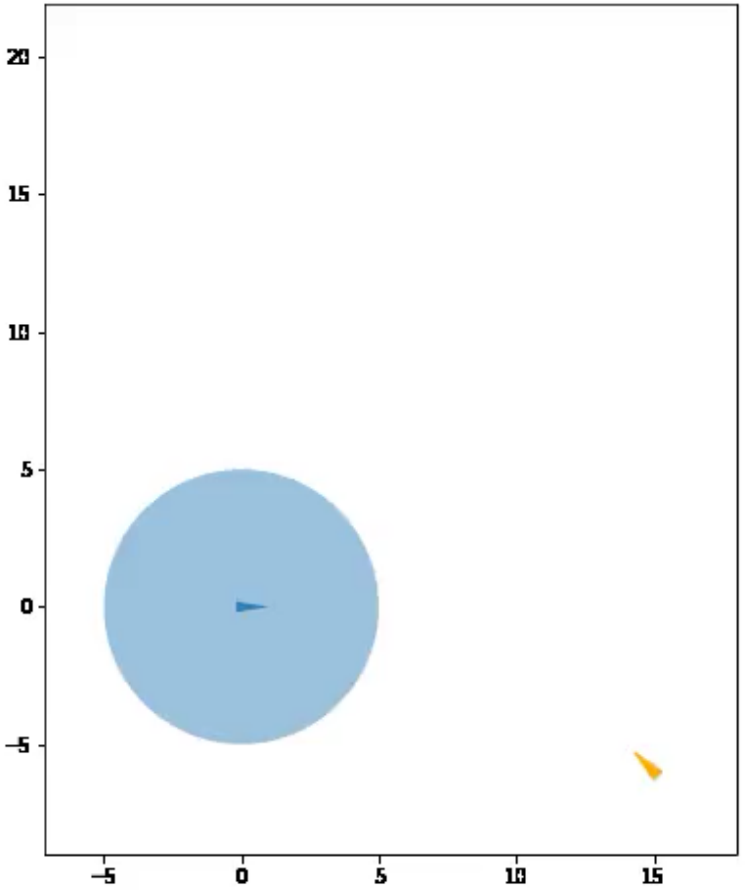
Human seeks collision

Aircraft collision avoidance

$$\begin{bmatrix} \dot{x}_{\text{rel}} \\ \dot{y}_{\text{rel}} \\ \dot{\theta}_{\text{rel}} \end{bmatrix} = \begin{bmatrix} -v_{\text{a}} + v_{\text{b}} \cos\theta_{\text{rel}} + y_{\text{rel}} u_{\text{a}} \\ v_{\text{b}} \sin\theta_{\text{rel}} - x_{\text{rel}} u_{\text{a}} \\ u_{\text{b}} - u_{\text{a}} \end{bmatrix}$$

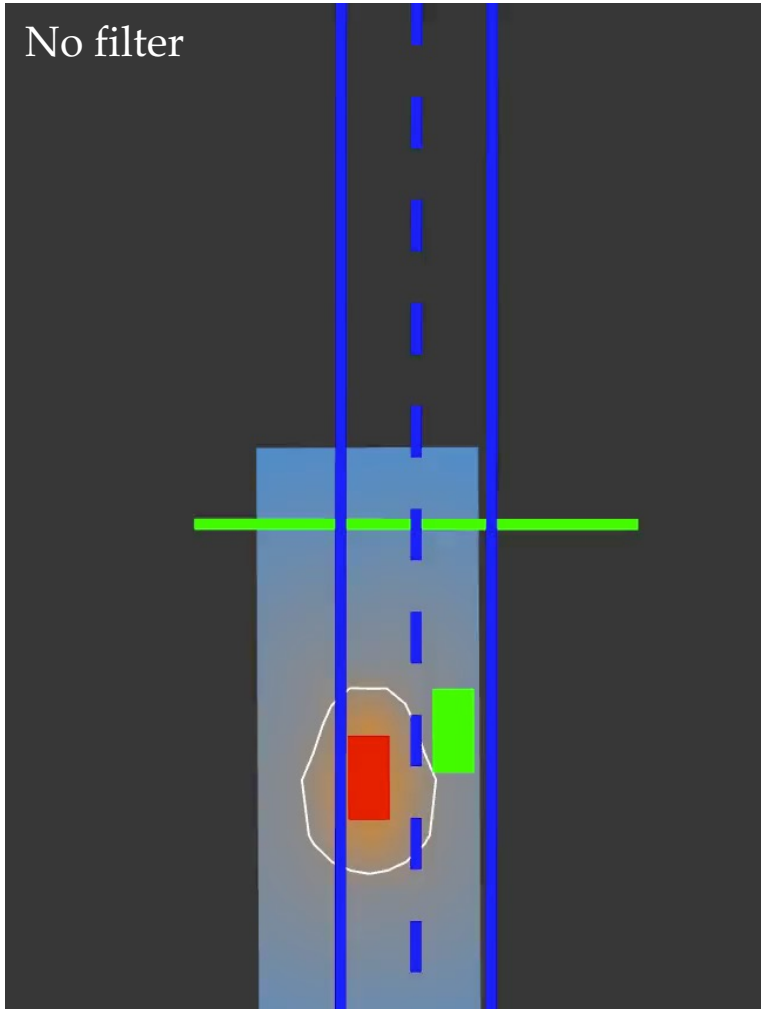




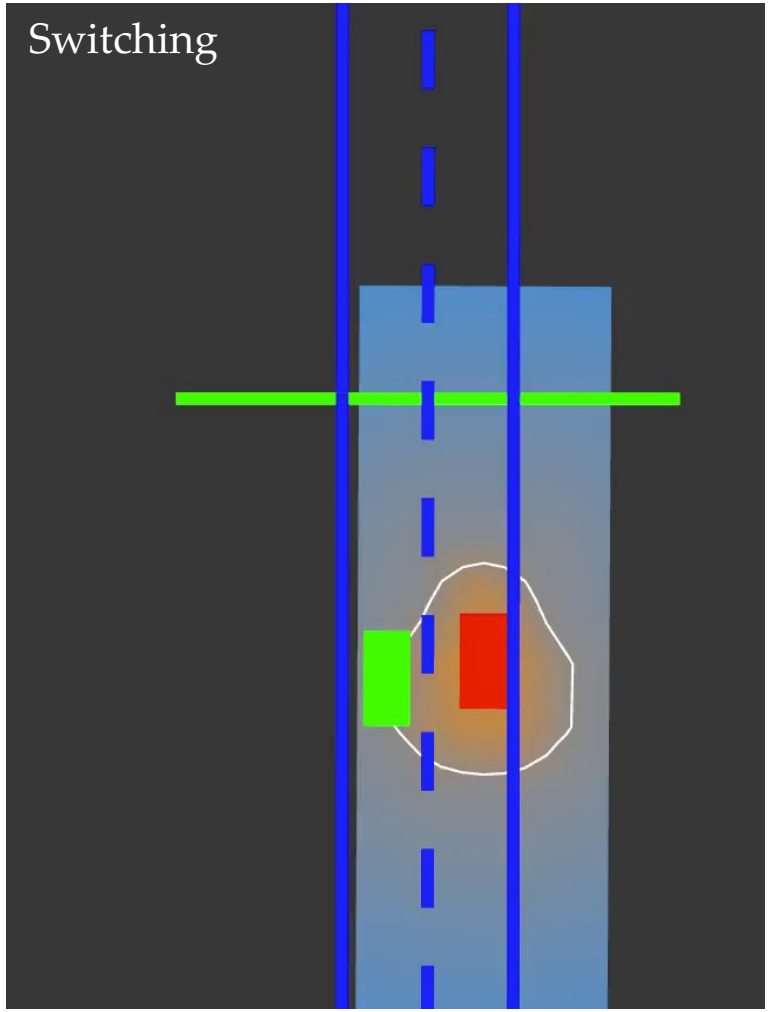


Types of safety filter logic

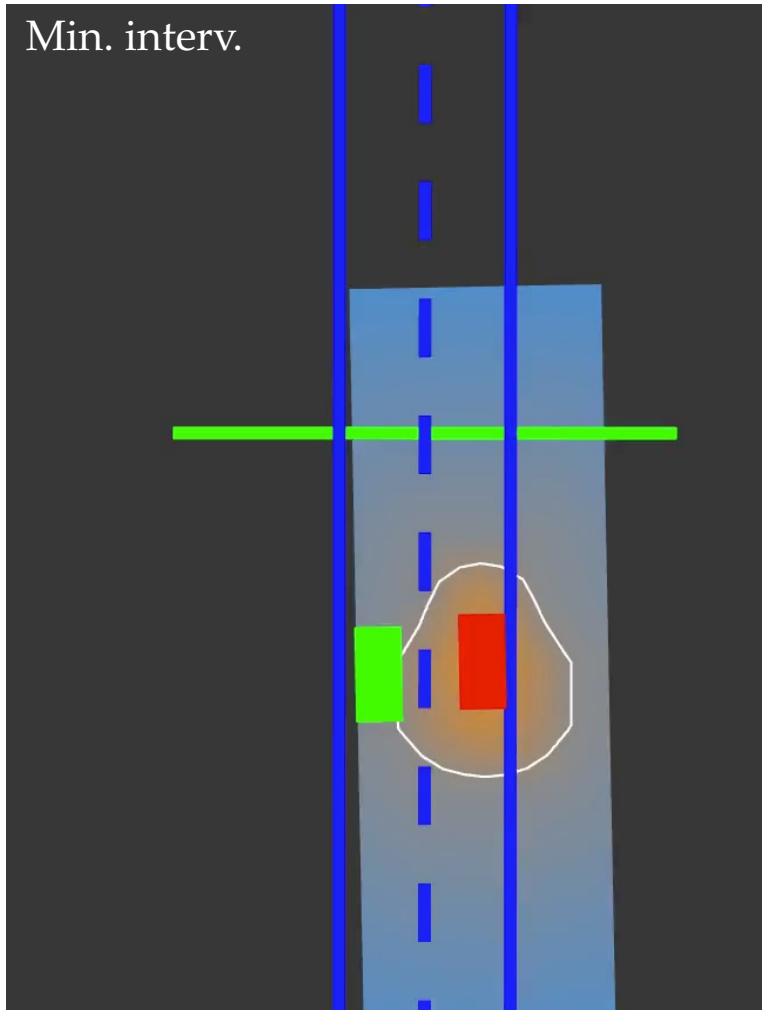
Using HJ BRTs as safety filters



https://youtube.com/shorts/ydhvZ29HHBo?feature=share



https://youtube.com/shorts/5uyLyYURqwM?feature=share



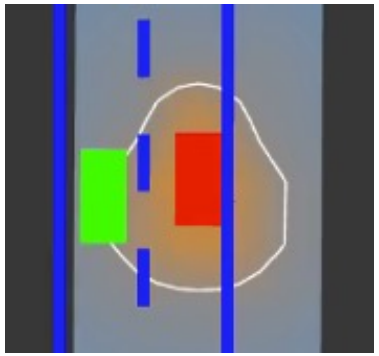
https://youtube.com/shorts/3Mq6-QPljbg?feature=share

What are "reasonable" assumptions about how other agents behave?

Hamilton-Jacobi-Isaacs partial differential equation

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

$$V(x,0) = v(x)$$



Overly-conservative assumptions lead to impractical safety concepts!

How to pick "reasonable" choices for U and D?

Leung, K., Bajcsy, A., Schmerling, E., and Pavone, M., **Towards data-driven synthesis of autonomous vehicle safety concepts,** https://arxiv.org/abs/2107.14412, 2022

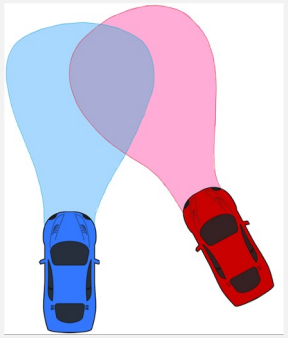
Unification of Safety Concepts via Optimal Control Theory

Depends on the assumptions you make about other agents when evaluating safety

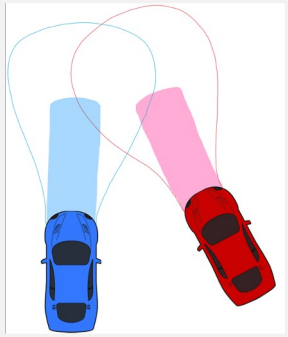
Hamilton-Jacobi Reachability

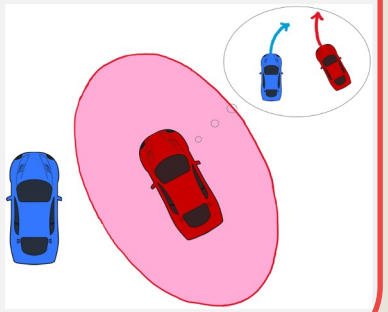
Open-loop
"non-reactive" policies

Closed-loop
"reactive" policies

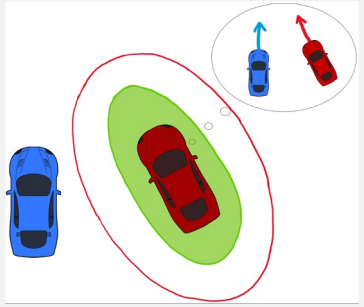


. . .





. . .



Consider *all* possible behaviors
Full forward reachable set

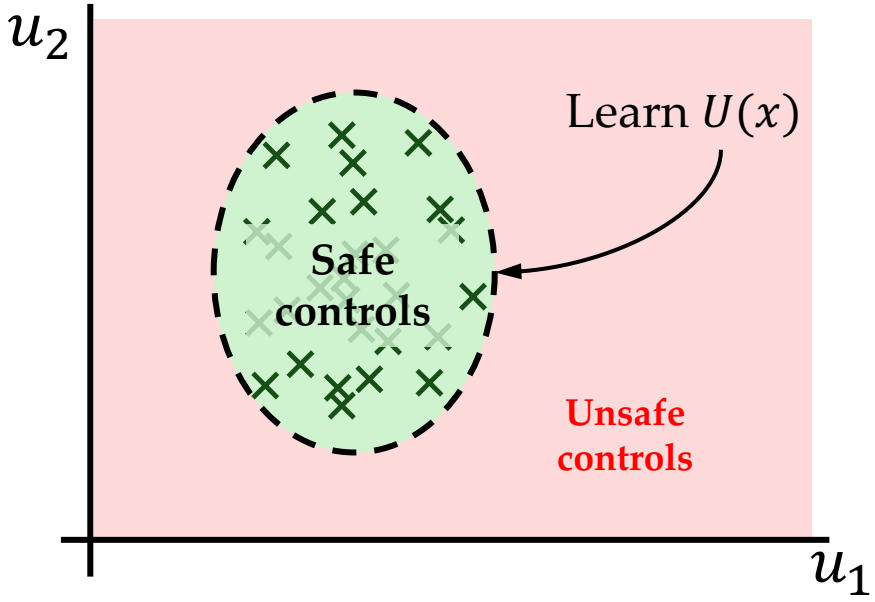
Consider only a *subset* of possible behaviors
e.g., hard-braking ***(SFF)***

Guard against *all* possible policies
Including worst-case outcomes

Guard against a *subset* of possible policies
Assumptions on other agent's behaviors

We can learn "reasonable controls" from human-human interaction data

Given a dataset of states and controls: $(x^{(1)}, u^{(1)}), (x^{(2)}, u^{(2)}), \ldots, (x^{(N)}, u^{(N)})$ we want to learn $U(x)$



Key insight: Humans take controls that keep them safe. Taking controls outside the boundary will lead to an undesirable outcome.

Data lives inside a control invariant set.

Need to learn a control invariant set!

Leung, K., Veer, S., Schmerling, E., and Pavone, M.,
Learning Autonomous Vehicle Safety Concepts from Demonstrations
American Control Conference, 2023, https://arxiv.org/abs/2210.02761

Control barrier functions describe control invariant sets

Control invariant set

Control barrier functions describe control invariant sets

Control barrier function

Choi et al 2021 https://arxiv.org/abs/2104.02808

Control barrier functions describe control invariant sets

Control barrier function

HJ value functions are valid CBFs!

Choi et al 2021 https://arxiv.org/abs/2104.02808

Safety filter as a QP

Idea: Project control into feasible safe control set

Potential issues?