AA/EE/ME 548: Linear Multivariable Control

Lecture 09

4/28/2025

Announcements

- Homework 2 out

Instructions

Download this jupyer notebook (button at the top of the page or download from the Github repository). Provide your answers as Markdown text, Python code, and/or produce plots as appropriate. The notebook should run all the cells in order without errors.
Submit both the `.ipynb` and a `.pdf` to Canvas.

Make sure the `.pdf` has all the relevant outputs showing. To save as `.pdf` you can first export the notebook as `.html`, open it in a browers and then "Print to PDF".

NOTE: As we will be sharing the files for peer grading, please keep your submission anonymous.

Announcements

- Homework 2 out
  - Start early! Or at least look at it early. Some parts may be helpful for your project!
  - Small update to dynamaxsys – please reinstall
- Project proposal due this week
  - Everyone must submit a proposal, even for group projects
  - Instructions: https://uw-ctrl.github.io/lmc-book/project/project.html
- Please sign release form
- Week 4 folks, worked examples due Wednesday
- HW1 folks, solutions due Friday
- Use mailing list to contact teaching team

Last week

- Bellman equation (discrete time)

$$V^*(x,t) = \min_{u \in \mathcal{U}(x)} \Big(J(x,u,t) + V^*(f(x,u,t), t+1) \Big)$$

- Infinite horizon case—add a discount term

$$V^{(k+1)}(x) \leftarrow \min_{u \in \mathcal{U}(x)} \Big(J(x,u,t) + \gamma V^{(k)}(f(x,u,t)) \Big)$$

- Stochastic case—consider *expected value*

$$V^*(x_t,t) = \min_{u_t \in \mathcal{U}(x_t)} \mathbb{E}_{x_{t+1} \sim p(x_{t+1} \mid x_t, u_t, t)} \Big[J(x_t,u_t,t) + V^*(x_{t+1}, t+1) \Big]$$

This week

- Continuous time: Hamilton-Jacobi-Bellman equation
- Robust case: Hamilton-Jacobi-Isaacs equation
- HJ reachability: Special case of HJI equation
- Linear quadratic regulator

Continuous time setting

- Optimal control problem for continuous time
  - Integral instead of summation
  - ODE dynamics instead of difference equation
  - Control signal instead of control sequence

$$
\begin{aligned}
\min_{u(\cdot)\in\mathbb{U}[0,T]} \quad & \int_0^T J(x,u,t)dt + J_T(x(T)) \\
\text{subject to} \quad & \dot{x} = f(x,u,t) \\
& x_0 = x(0) \\
& u \in \mathcal{U}(x), x \in \mathcal{X}
\end{aligned}
$$

Continuous time setting

- What happens when $\Delta t \rightarrow 0$?

$$V^*(x,t) = \min_{u \in \mathcal{U}(x)} \Big(J(x,u,t) + V^*(f(x,u,t), t+1) \Big)$$

Continuous time setting

$$V^*(x,t) = \min_{u \in \mathcal{U}(x)} \left(\epsilon J(x,u,t) + V^*(x(t+\epsilon), t+\epsilon) \right)$$

$$0 = \min_{u \in \mathcal{U}(x)} \left(\epsilon J(x,u,t) + V^*(x(t+\epsilon), t+\epsilon) - V^*(x,t) \right)$$

$$0 = \lim_{\epsilon \to 0} \min_{u \in \mathcal{U}(x)} \left(J(x,u,t) + \frac{V^*(x(t+\epsilon), t+\epsilon) - V^*(x,t)}{\epsilon} \right)$$

$$0 = \min_{u \in \mathcal{U}(x)} \left(J(x,u,t) + \frac{\partial V}{\partial t}(x,t) + \nabla V(x,t)^T f(x,u,t) \right)$$

Hamilton-Jacobi-Bellman equation

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \left(J(x,u,t) + \nabla V(x,t)^T f(x,u,t) \right) = 0$$

$$V(x,T) = J_T(x)$$

- Solve PDE *backward* in time
- Final value condition defined by terminal cost function

What about disturbances?

$\dot{x} = f(x, u, \mathbf{d}, t)$ Suppose there is a disturbance term

- View problem as a two-player zero-sum game
  - Control "u" minimizes cost
  - Disturbance "d" maximizes cost

Just add a "max over d" term!

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \left(J(x,u,t) + \nabla V(x,t)^T f(x,u,t) \right) = 0$$

HJB equation with disturbance

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \max_{\boldsymbol{d} \in \boldsymbol{\mathcal{D}}(\boldsymbol{x})} \left(J(x,u,\boldsymbol{d},t) + \nabla V(x,t)^T f(x,u,\boldsymbol{d},t) \right) = 0$$

OR

$$\frac{\partial V}{\partial t}(x,t) + \max_{\boldsymbol{d} \in \boldsymbol{\mathcal{D}}(\boldsymbol{x})} \min_{u \in \mathcal{U}(x)} \left(J(x,u,\boldsymbol{d},t) + \nabla V(x,t)^T f(x,u,\boldsymbol{d},t) \right) = 0$$

Hamilton-Jacobi-Isaacs equation

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \max_{d \in \mathcal{D}(x)} \left(J(x,u,d,t) + \nabla V(x,t)^T f(x,u,d,t) \right) = 0$$

- Robust setting: Compute the value function with optimal "worst-case" disturbance policy
- At every step, assume the disturbance is optimally working against the controller

The key equations so far…

- Bellman equation

$$V^*(x,t) = \min_{u \in \mathcal{U}(x)} \Big(J(x,u,t) + V^*(f(x,u,t), t+1) \Big)$$

- Infinite horizon case

$$V^{(k+1)}(x) \leftarrow \min_{u \in \mathcal{U}(x)} \Big(J(x,u,t) + \gamma V^{(k)}(f(x,u,t)) \Big)$$

- Stochastic case

$$V^*(x_t,t) = \min_{u_t \in \mathcal{U}(x_t)} \mathbb{E}_{x_{t+1} \sim p(x_{t+1}|x_t,u_t,t)} \Big[J(x_t,u_t,t) + V^*(x_{t+1}, t+1) \Big]$$

- Hamilton-Jacobi-Bellman equation

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \Big(J(x,u,t) + \nabla V(x,t)^T f(x,u,t) \Big) = 0$$

- Hamilton-Jacobi-Isaacs equation

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \max_{d \in \mathcal{D}(x)} \Big(J(x,u,d,t) + \nabla V(x,t)^T f(x,u,d,t) \Big) = 0$$

Reachability analysis

- The study of the set of states a dynamical system can (and can’t) reach over time given:
  - its dynamics (how it moves),
  - initial conditions,
  - possible inputs (control),
  - disturbances (uncertainty).
- Applications
  - Safety-critical control: Avoid inevitable collision states
  - Robust control: Plan for ”reachable tube” to avoid obstacles
  - Provide guarantees: Certify safe/unsafe states before deployment

Forward reachability (open-loop)

Forward reachability: The set of states a system can reach within some time using any control input

"Starting from here, under any allowed input/disturbance, where could we end up?"

Open loop because:

- Assume all possible control inputs are applied without correction along the way. I.e., no feedback.

$$\mathcal{F}(x_0, t) = \{x \in \mathcal{X} \mid \exists u(\cdot) \in \mathbb{U}[0, t],\ x = \xi_{x_0,0}^{u(\cdot)}(t)\}$$

Backward reachability (closed-loop)

Backward reachability: Given a *target* set, what are all the states that could reach (or avoid) that set in the future?

“Where must I be/avoid now to reach/avoid the target set later?”

Closed-loop because

- At each state and time, we consider the best control to take to reach/avoid the target set

Sounds like dynamic programming…

Backward reachable sets

Reach case

$$\mathcal{R}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \exists u(\cdot) \in \mathbb{U}[0, t] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(t) \in \mathcal{T}\}$$

Avoid case (Inevitable collision set)

$$\mathcal{A}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \forall u(\cdot) \in \mathbb{U}[0, t] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(t) \in \mathcal{T}\}$$

Issues?

Backward reachable tube (any time)

Reach case

$$\widetilde{\mathcal{R}}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \exists \boldsymbol{s} \in [\mathbf{0}, \boldsymbol{t}], \exists u(\cdot) \in \mathbb{U}[0, \boldsymbol{s}] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(\boldsymbol{s}) \in \mathcal{T}\}$$

Avoid case (Inevitable collision set)

$$\widetilde{\mathcal{A}}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \exists \boldsymbol{s} \in [\mathbf{0}, \boldsymbol{t}], \forall u(\cdot) \in \mathbb{U}[0, \boldsymbol{s}] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(\boldsymbol{s}) \in \mathcal{T}\}$$

With disturbances

Reach case

$$\mathcal{R}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \forall \boldsymbol{d}(\cdot) \in \mathbb{D}[0, \boldsymbol{t}], \exists u(\cdot) \in \mathbb{U}[0, t] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(t) \in \mathcal{T}\}$$

$$\widetilde{\mathcal{R}}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \forall \boldsymbol{d}(\cdot) \in \mathbb{D}[0, \boldsymbol{t}], \exists \boldsymbol{s} \in [0, \boldsymbol{t}], \exists u(\cdot) \in \mathbb{U}[0, s] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(s) \in \mathcal{T}\}$$

Avoid case

$$\mathcal{A}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \exists \boldsymbol{d}(\cdot) \in \mathbb{D}[0, \boldsymbol{t}], \forall u(\cdot) \in \mathbb{U}[0, t] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(t) \in \mathcal{T}\}$$

$$\widetilde{\mathcal{A}}(\mathcal{T}, t) = \{x_0 \in \mathcal{X} \mid \exists \boldsymbol{d}(\cdot) \in \mathbb{D}[0, \boldsymbol{t}], \exists \boldsymbol{s} \in [0, \boldsymbol{t}], \forall u(\cdot) \in \mathbb{U}[0, s] \text{ s.t. } \xi_{x_0,0}^{u(\cdot)}(s) \in \mathcal{T}\}$$

How to compute the sets?

- Frame as optimal control problem!
- Key idea: *Only care whether the system reaches the target set.*

$$
\begin{aligned}
\min_{u(\cdot)\in\mathbb{U}[0,T]} \quad & \cancel{\int_0^T J(x,u,t)dt} + J_T(x(T)) \\
\text{subject to} \quad & \dot{x} = f(x,u,t) \\
& x_0 = x(0) \\
& u \in \mathcal{U}(x), x \in \mathcal{X}
\end{aligned}
$$

Define target set & value function interpretation

$$\mathcal{T} = \{x \in \mathcal{X} \mid J_T(x) \leq 0\}$$

$$V(x,t) = \min_{u(\cdot)\in\mathbb{U}[t,T]} J_T(x(T))$$

Reach case

"regular"

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \nabla V(x,t)^T f(x,u,t) = 0$$

"w/ disturbance"

$$\frac{\partial V}{\partial t}(x,t) + \min_{u \in \mathcal{U}(x)} \max_{d \in \mathcal{D}(x)} \nabla V(x,t)^T f(x,u,d,t) = 0$$

"tube"

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

$$V(x,T) = J_T(x)$$

Avoid case

"regular"

$$\frac{\partial V}{\partial t}(x,t) + \max_{u \in \mathcal{U}(x)} \nabla V(x,t)^T f(x,u,t) = 0$$

"w/ disturbance"

$$\frac{\partial V}{\partial t}(x,t) + \max_{u \in \mathcal{U}(x)} \min_{d \in \mathcal{D}(x)} \nabla V(x,t)^T f(x,u,d,t) = 0$$

"tube"

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

$$V(x,T) = J_T(x)$$

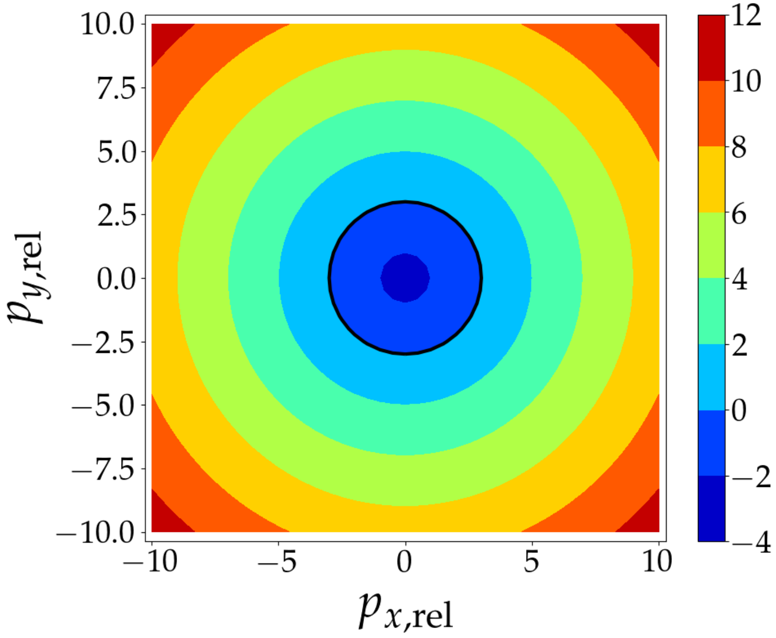
Computing HJ reachable sets

- https://github.com/StanfordASL/hj_reachability
- https://github.com/StanfordASL/hj_reachability/blob/main/examples/quickstart.ipynb
-

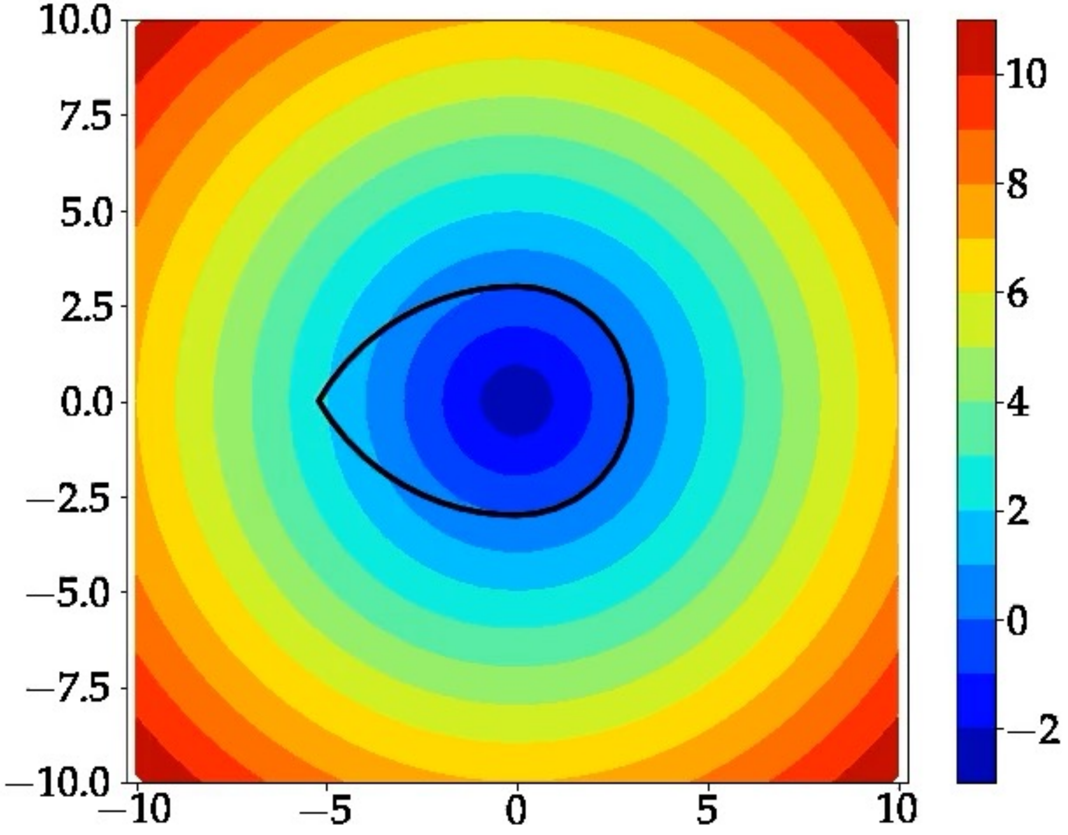
```
dynamics = hj.systems.Air3d()

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)
```

Dubins' car example

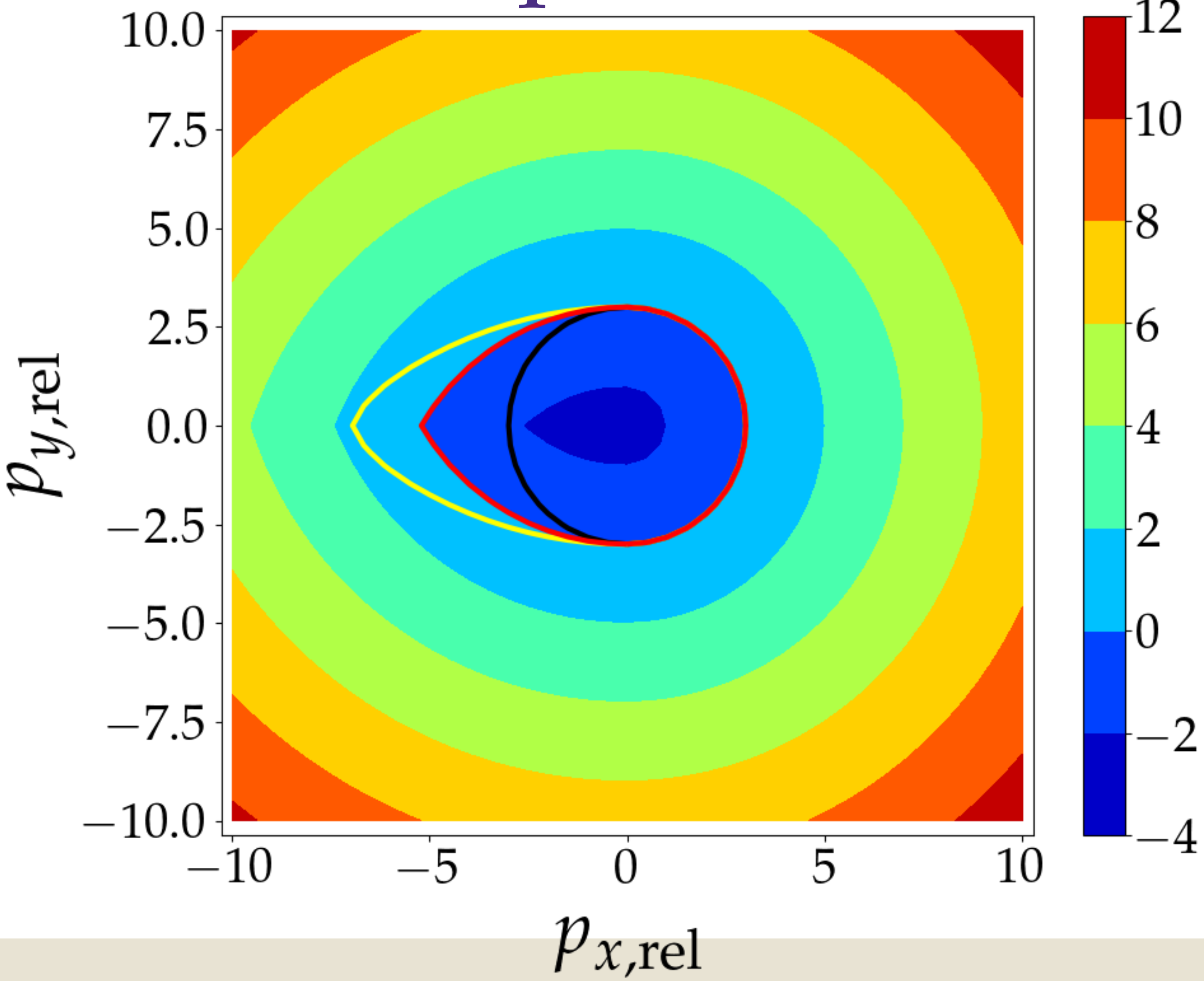
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ u + d \end{bmatrix}$$



No disturbances



Dubins' car example with disturbances

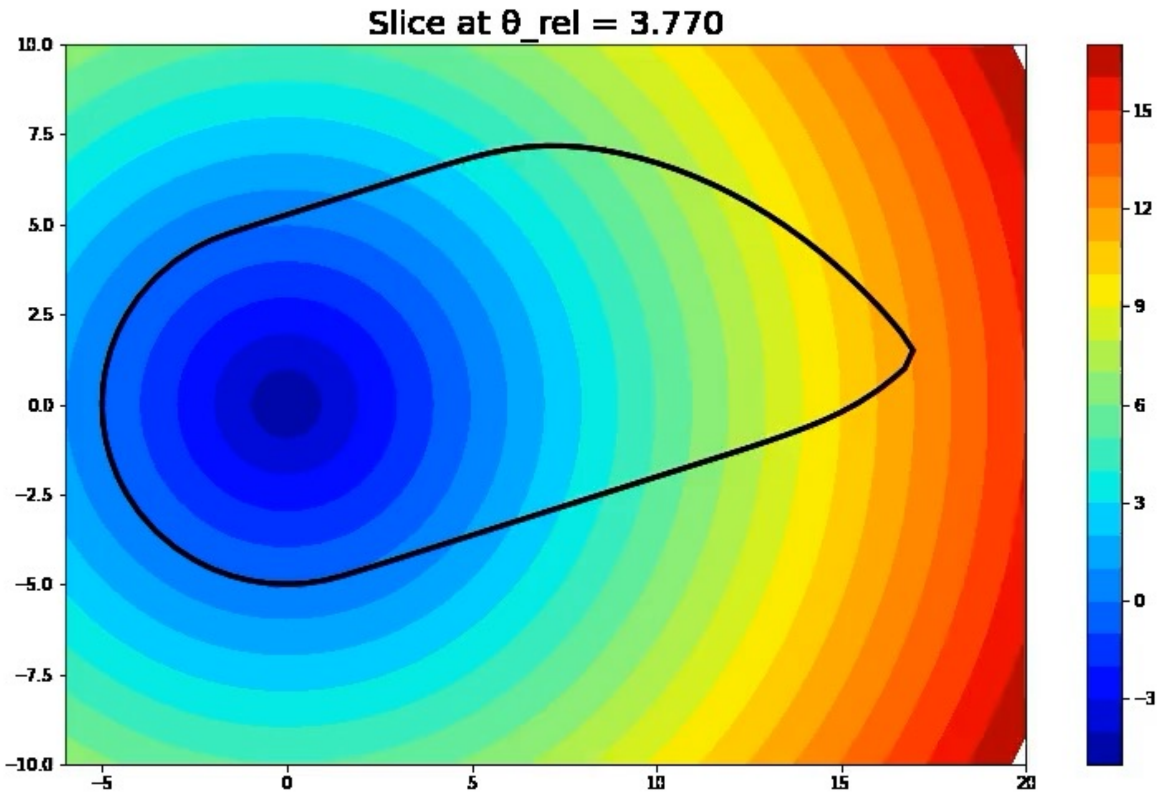


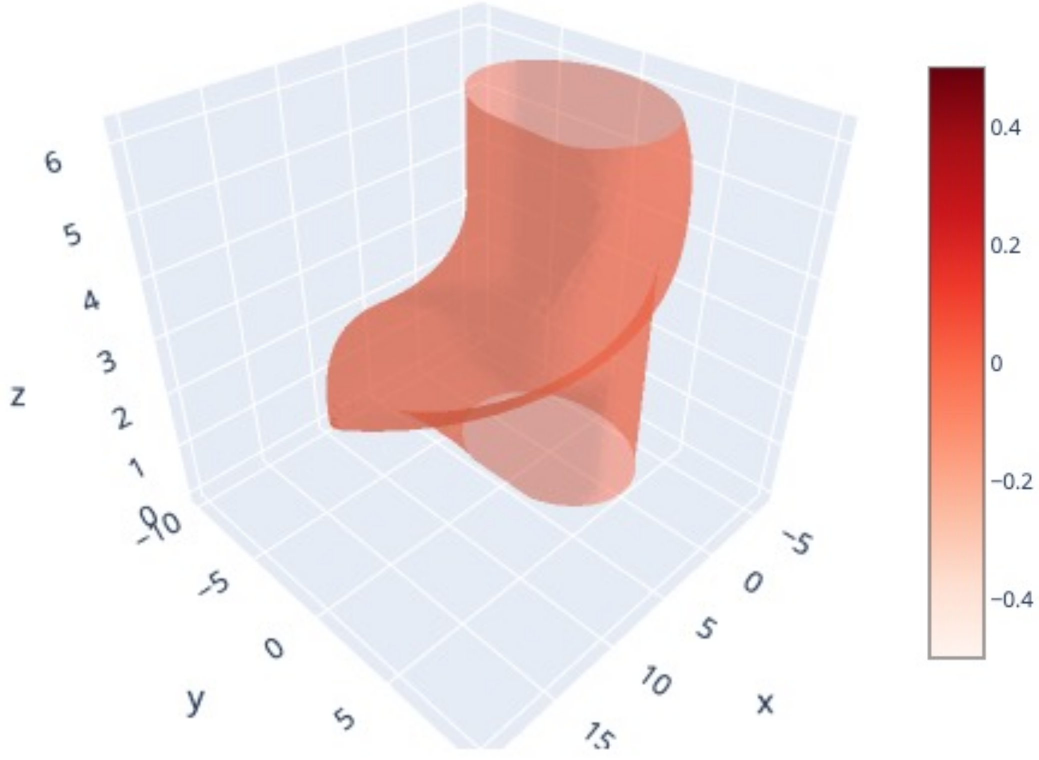
Pursuit-evasion example

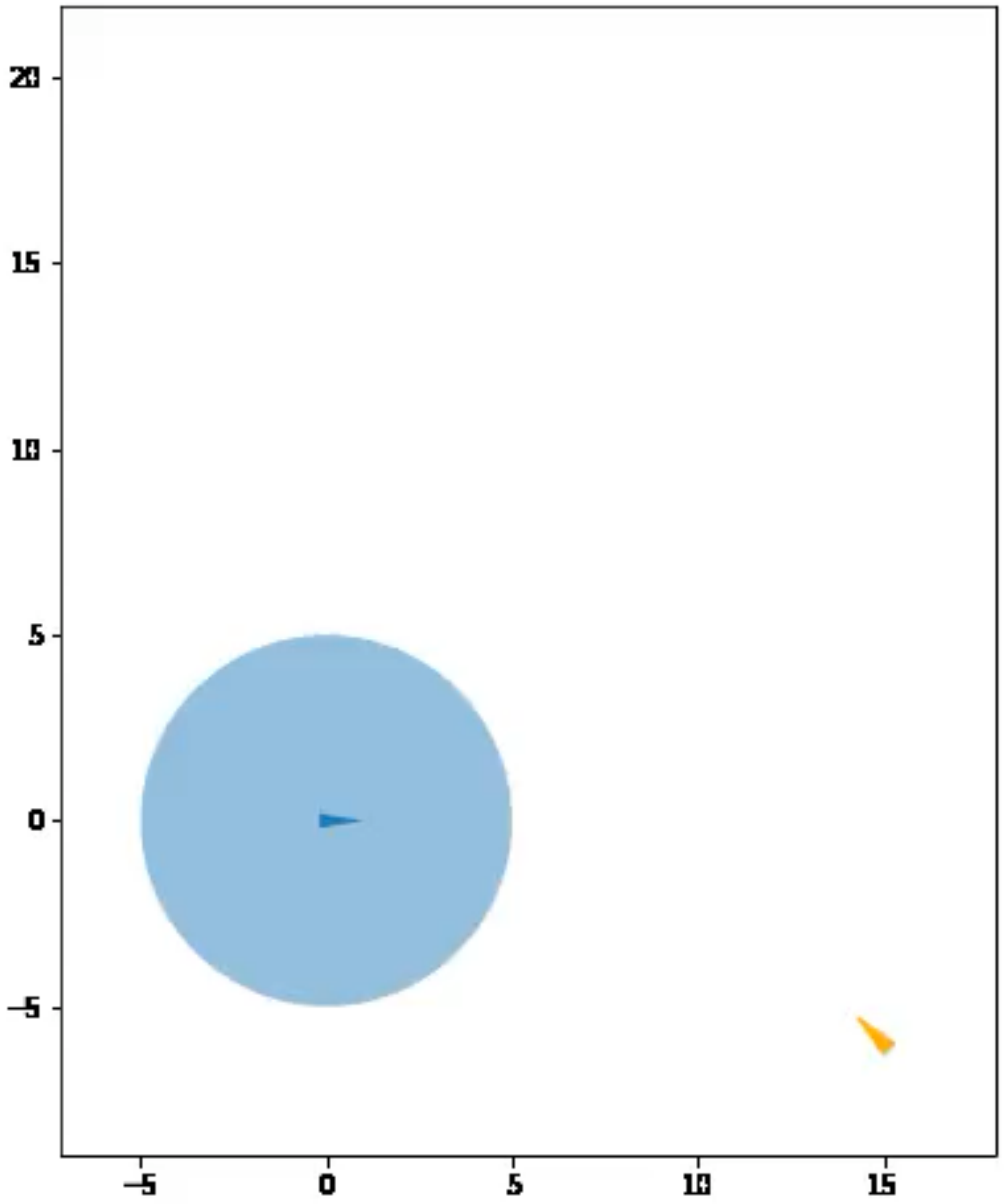
- Can consider a *relative* system between two agents & treat other agent as a disturbance
- Consider two Dubins' car agents, one is pursuing the other (i.e., cat and mouse)

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

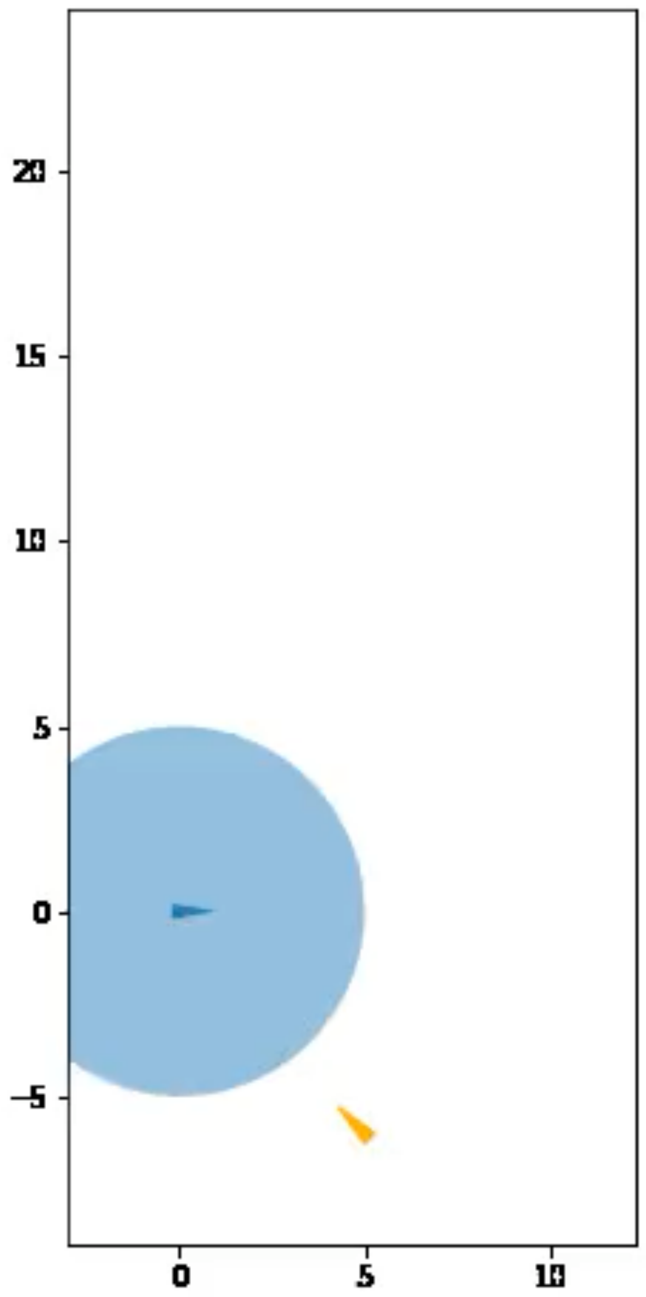
Pursuit-evasion example



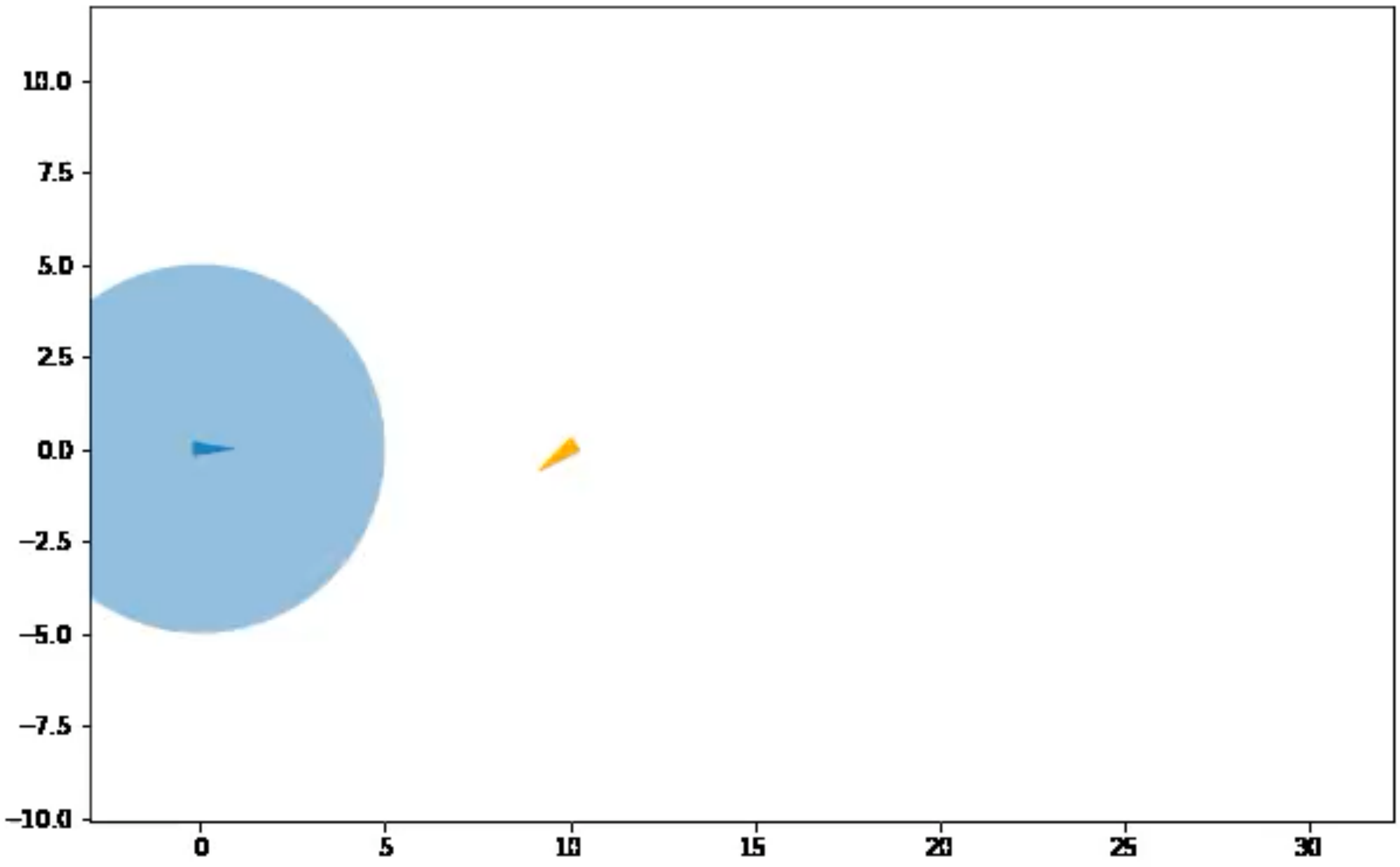




20
15
10
5
0
−5
−5
0
5
10
15



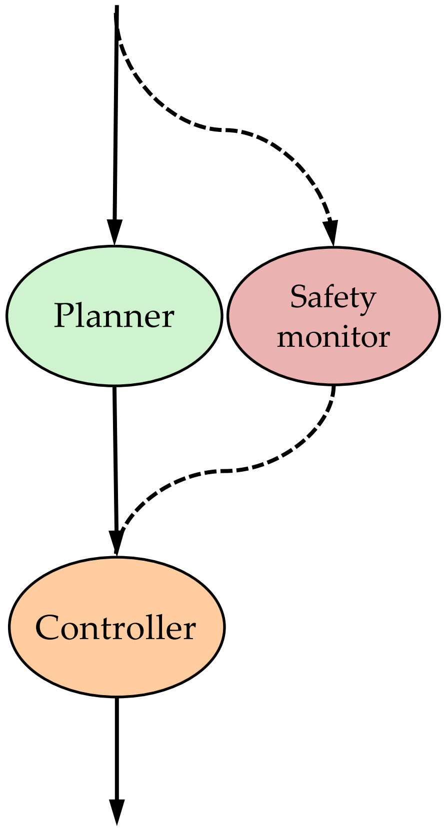
20
15
10
5
0
−5
0
5
10


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
0
5
10
15
20
25
30

What can I do with the HJ value function?

- HJ value functions are CBFs!
  - https://arxiv.org/abs/2104.02808
  - https://youtu.be/9SIO-oNT7Gs
- Least-restrictive controller
  - When $V(x, t) \leq \epsilon$, switch to optimal HJ policy
- Minimally-interventional safe controller
  - When $V(x, t) \leq \epsilon$, add HJ control constraint in tracking controller
  - https://arxiv.org/abs/2012.03390

Minimally-interventional control



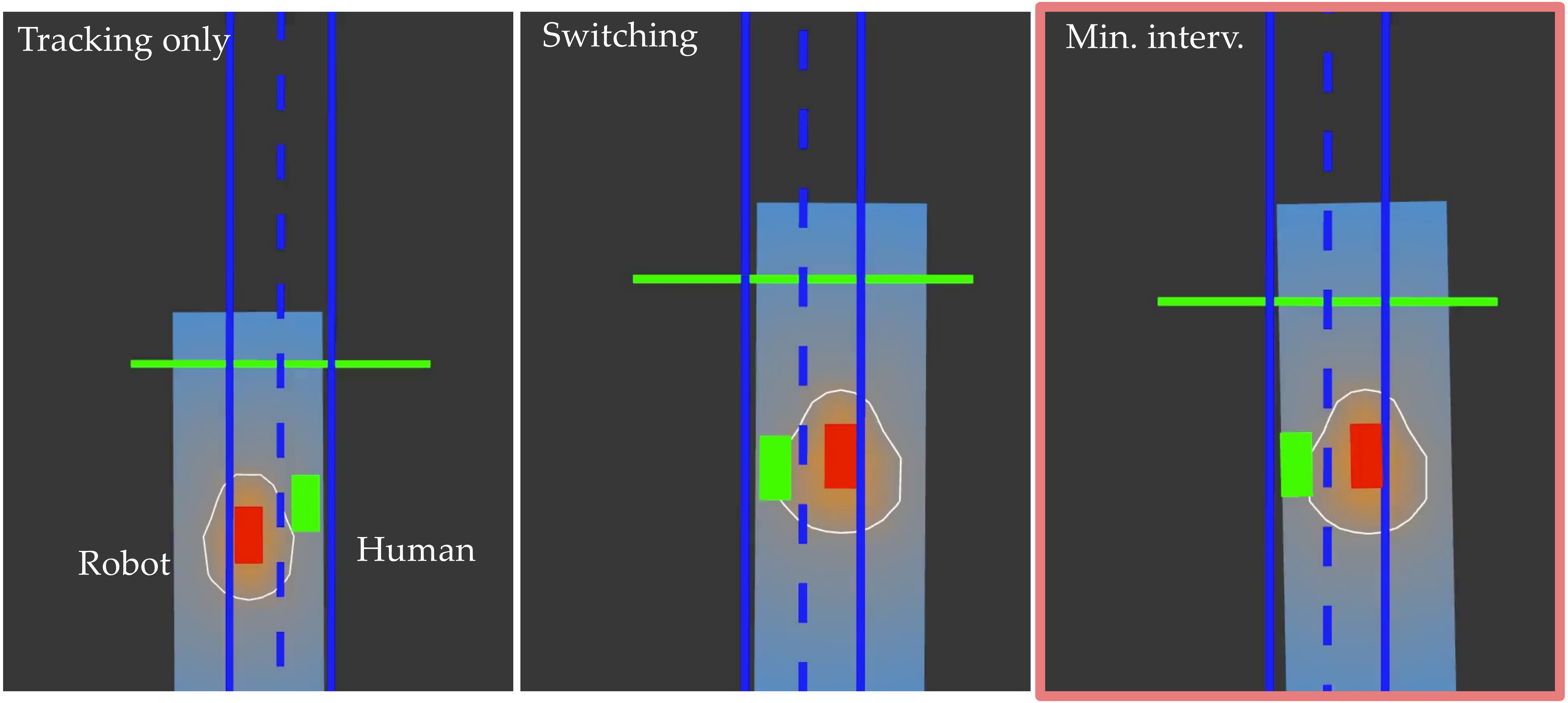
Minimize: Tracking error

Subject to:

- Control limits
- Dynamics
- State constraints

\+

If $V(x, t) < \epsilon$
then activate HJ safety-preserving control set constraint


Tracking only
Robot
Human
Switching
Min. interv.