

AA203

Optimal and Learning-based Control

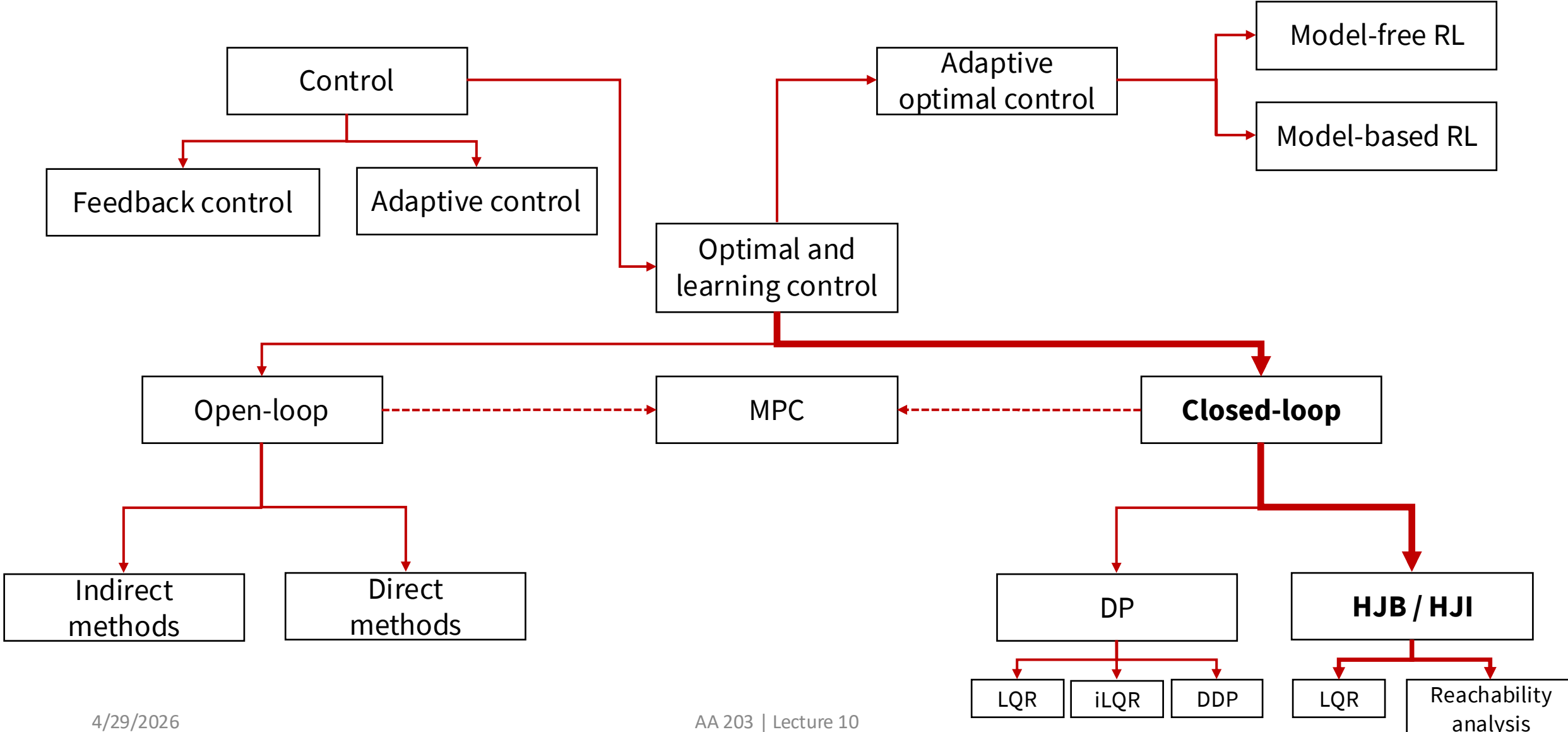
HJB, HJI, and reachability analysis



Stanford
University



Roadmap



Dynamic Programming

Previous lectures: focus on discrete-time setting

This lecture: focus on continuous-time setting

- dynamic programming approach leads to HJB / HJI equation: non-linear partial differential equation
- HJB application: solution to continuous LQR problem
- HJI application: reachability analysis

Readings: lecture notes and references therein, in particular:

- [Bansal S., Chen M., Herbert S., Tomlin C. J., “Hamilton-Jacobi reachability: A brief overview and recent advances,” 2017.](#)
- [Chen M., Tomlin C. J., “Hamilton-Jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management,” 2018.](#)

Continuous-time model

Last time:

- Model: $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, k)$,
- Cost: $J(\mathbf{x}_0) = h_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} g(\mathbf{x}_k, \mathbf{u}_k, k)$

This time:

- Model: $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t)$,
- Cost: $J(\mathbf{x}(t_0)) = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau$

where t_0 and t_f are fixed

Two-person, zero-sum differential games

What if there is another player (e.g., nature) that interferes with the fulfillment of our objective?

Two-person differential game:

- Model: $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))$ (*joint system dynamics*),
- Cost: $J(\mathbf{x}(t_0)) = h(\mathbf{x}(0)) + \int_{t_0}^0 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau$
- Player 1, with control $\mathbf{u}(\tau)$, will attempt to *maximize* J , while Player 2, with control $\mathbf{d}(t)$, will aim to *minimize* J , subject to the joint system dynamics
- $\mathbf{x}(\tau)$ is the *joint system state*

Information pattern

- To fully specify the game, we need to specify the *information pattern*
- “Open-loop” strategies
 - Player 1, with control $\mathbf{u}(\tau)$, declares entire plan
 - Player 2, with control $\mathbf{d}(\tau)$, responds optimally
 - Conservative, unrealistic, but computationally cheap
- “Nonanticipative” strategies
 - Other agent acts based on state and control trajectory up to current time
 - Notation: $\mathbf{d}(\cdot) = \Gamma[\mathbf{u}](\cdot)$
 - Disturbance still has the advantage: it gets to (instantaneously) react to the control!

Hamilton-Jacobi-Isaacs (HJI) equation

Key idea: apply principle of optimality

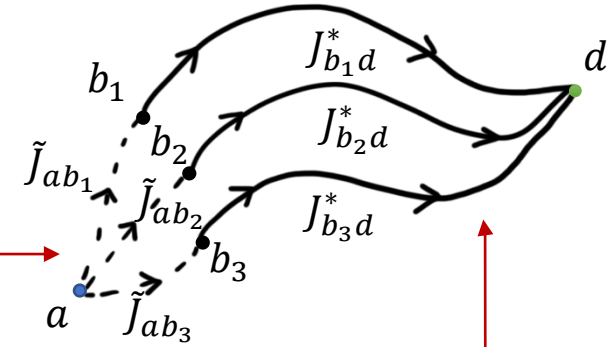
The “truncated” problem is

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\int_t^0 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau + h(\mathbf{x}(0)) \right]$$

 Worst-case disturbance – aims to thwart the controller

HJI equation

- Dynamic programming principle:



$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau + J(\mathbf{x}(t + \Delta t), t + \Delta t) \right]$$

- Approximate integral and Taylor expand $J(\mathbf{x}(t + \Delta t), t + \Delta t)$
- Derive Hamilton-Jacobi-Isaacs partial differential equation (HJI PDE)

HJI equation

- Approximations for small Δt :

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau + J(\mathbf{x}(t + \Delta t), t + \Delta t) \right]$$

HJI equation

- Approximations for small Δt :

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + J(\mathbf{x}(t + \Delta t), t + \Delta t) \right]$$

HJI equation

- Approximations for small Δt :

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + \overbrace{J(\mathbf{x}(t + \Delta t), t + \Delta t)}^{\mathbf{x}(t) + \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d})} \right]$$

HJI equation

- Approximations for small Δt :

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + \underbrace{J(\mathbf{x}(t + \Delta t), t + \Delta t)}_{J(\mathbf{x}(t), t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) + \frac{\partial J}{\partial t} \Delta t} \right]$$

HJI equation

- Approximations for small Δt :

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + \underbrace{J(\mathbf{x}(t + \Delta t), t + \Delta t)}_{J(\mathbf{x}(t), t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) + \frac{\partial J}{\partial t} \Delta t} \right]$$

- Omit t dependence...

$$J(\mathbf{x}, t) = \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + J(\mathbf{x}, t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial t} \Delta t \right]$$

- Assume (instantaneously) constant \mathbf{u} and $\mathbf{d} \rightarrow$ optimization over vectors, not functions!
- Order of max and min reverse (proof given in references)

- $J(\mathbf{x}, t)$ does not depend on \mathbf{u} or \mathbf{d}

$$J(\mathbf{x}, t) = J(\mathbf{x}, t) + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial t} \Delta t \right]$$

HJI equation

- Approximations for small Δt : $\mathbf{x}(t) + \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d})$

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + \underbrace{J(\mathbf{x}(t + \Delta t), t + \Delta t)}_{J(\mathbf{x}(t), t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) + \frac{\partial J}{\partial t} \Delta t} \right]$$

- Omit t dependence...

$$J(\mathbf{x}, t) = \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + J(\mathbf{x}, t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial t} \Delta t \right]$$

- Assume (instantaneously) constant \mathbf{u} and $\mathbf{d} \rightarrow$ optimization over vectors, not functions!
- Order of max and min reverse (proof given in references)

- $J(\mathbf{x}, t)$ does not depend on \mathbf{u} or \mathbf{d}

$$J(\mathbf{x}, t) = J(\mathbf{x}, t) + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial t} \Delta t \right]$$

HJI equation

- Approximations for small Δt : $\mathbf{x}(t) + \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d})$

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + \underbrace{J(\mathbf{x}(t + \Delta t), t + \Delta t)}_{J(\mathbf{x}(t), t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) + \frac{\partial J}{\partial t} \Delta t} \right]$$

- Omit t dependence...

$$J(\mathbf{x}, t) = \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + J(\mathbf{x}, t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial t} \Delta t \right]$$

- Assume (instantaneously) constant \mathbf{u} and $\mathbf{d} \rightarrow$ optimization over vectors, not functions!
- Order of max and min reverse (proof given in references)

- $J(\mathbf{x}, t)$ does not depend on \mathbf{u} or \mathbf{d}

$$0 = \frac{\partial J}{\partial t} \Delta t + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right]$$

HJI equation

- Approximations for small Δt :

$$J(\mathbf{x}(t), t) = \min_{\Gamma[\mathbf{u}](\cdot)} \max_{\mathbf{u}(\cdot)} \left[\underbrace{\int_t^{t+\Delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{d}(\tau)) d\tau}_{g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))\Delta t} + \underbrace{J(\mathbf{x}(t + \Delta t), t + \Delta t)}_{J(\mathbf{x}(t), t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) + \frac{\partial J}{\partial t} \Delta t} \right]$$

- Omit t dependence...

$$J(\mathbf{x}, t) = \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d})\Delta t + J(\mathbf{x}, t) + \frac{\partial J}{\partial \mathbf{x}} \cdot \Delta t f(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial t} \Delta t \right]$$

- Assume (instantaneously) constant \mathbf{u} and $\mathbf{d} \rightarrow$ optimization over vectors, not functions!
- Order of max and min reverse (proof given in references)

- $J(\mathbf{x}, t)$ does not depend on \mathbf{u} or \mathbf{d}

$$0 = \frac{\partial J}{\partial t} + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right]$$

HJI equation

The end result is the Hamilton-Jacobi-Isaacs (HJI) equation

$$0 = \frac{\partial J}{\partial t} + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right]$$

with boundary condition

$$J(\mathbf{x}, 0) = h(\mathbf{x})$$

The “Hamiltonian”

- Given the cost-to-go function, the optimal control for Player 1 is

$$\mathbf{u}^*(\mathbf{x}, t) = \arg \max_{\mathbf{u}} \min_{\mathbf{d}} g(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

HJB equation

In case there is no disturbance, end result is the Hamilton-Jacobi-Bellman (HJB) equation

Without a disturbance, \mathbf{u} is usually selected to minimize cost

$$0 = \frac{\partial J}{\partial t} + \min_{\mathbf{u}} \left[g(\mathbf{x}, \mathbf{u}, t) + \frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, t) \right]$$

with boundary condition $J(\mathbf{x}, 0) = h(\mathbf{x})$

- Given the cost-to-go function, the optimal control is

$$\mathbf{u}^*(\mathbf{x}, t) = \arg \min_{\mathbf{u}} g(\mathbf{x}, \mathbf{u}, t) + \frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, t)$$

Continuous-time LQR

Continuous-time LQR: select control inputs to minimize

$$J(\mathbf{x}_0) = \frac{1}{2} \mathbf{x}(t_f)^T H \mathbf{x}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [\mathbf{x}(t)^T Q(t) \mathbf{x}(t) + \mathbf{u}(t)^T R(t) \mathbf{u}(t)] dt$$

subject to the dynamics

$$\dot{\mathbf{x}}(t) = A(t) \mathbf{x}(t) + B(t) \mathbf{u}(t)$$

Assumptions:

- $H = H^T \succcurlyeq 0$, $Q(t) = Q(t)^T \succcurlyeq 0$, $R(t) = R(t)^T \succ 0$
- t_0 and t_f specified
- $\mathbf{x}(t)$ and $\mathbf{u}(t)$ unconstrained

Continuous-time LQR

- As before, value function takes the form: $J(\mathbf{x}(t), t) = \frac{1}{2} \mathbf{x}(t)^T V(t) \mathbf{x}(t)$
- The HJB equation reduces to an ODE (the Riccati equation):

$$-\dot{V}(t) = Q(t) - V(t)B(t)R(t)^{-1}B(t)^T V(t) + V(t)A(t) + A(t)^T V(t)$$

- Riccati equation is integrated **backwards, with boundary condition** $V(t_f) = H$
- Once we find $V(t)$, the control policy is

$$\mathbf{u}^*(t) = -R(t)^{-1}B(t)^T V(t)\mathbf{x}(t)$$

- Analogously to the discrete case, under some additional assumptions, $V(t) \rightarrow$ constant in the infinite horizon setting
- See [Notes §3.3](#) for more details

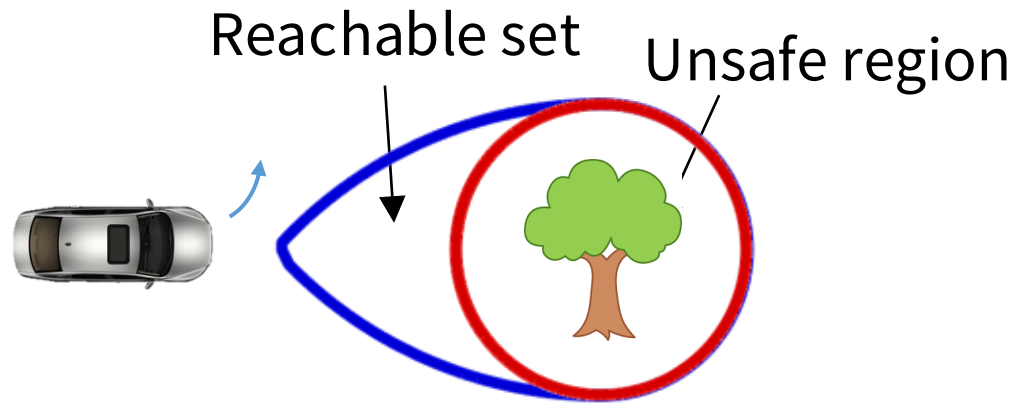
Applications of differential games

- Pursuit-evasion games
 - homicidal chauffeur problem
- Reachability analysis
- And many more (e.g., in economics)

Applications of differential games

- Pursuit-evasion games
 - homicidal chauffeur problem
- Reachability analysis
- And many more (e.g., in economics)

Reachability analysis: avoidance



Inputs:

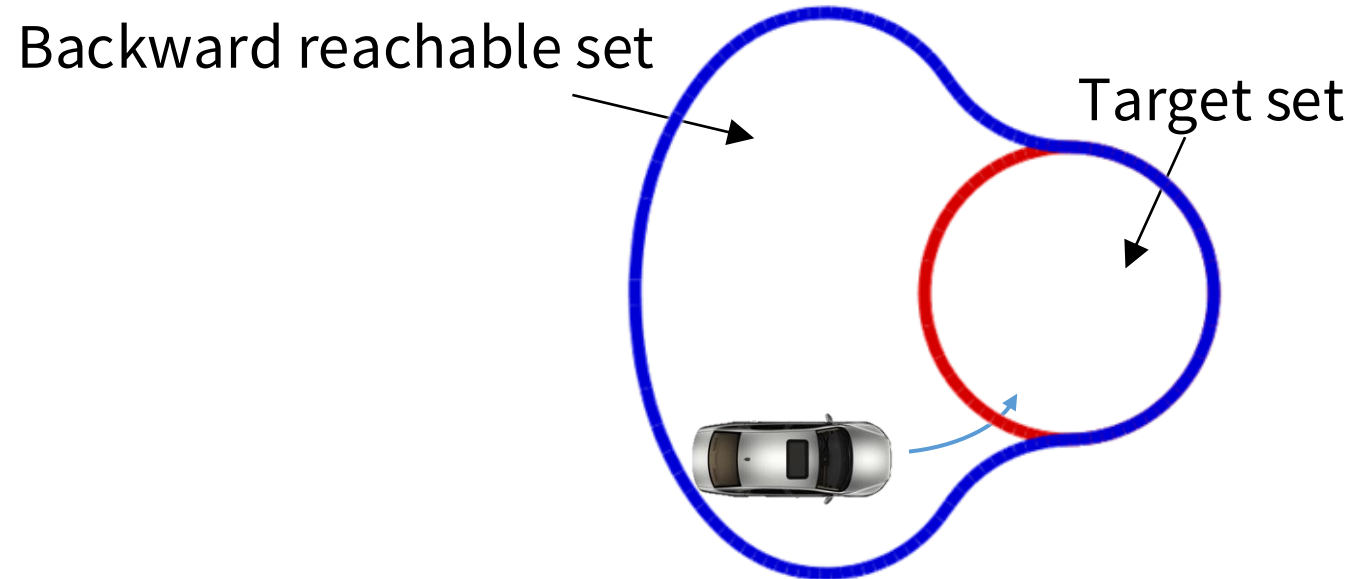
- System model
- Unsafe region:
e.g., obstacle



Control policy

Backward reachable set
(States leading to danger)

Reachability analysis: goal reaching



Inputs:

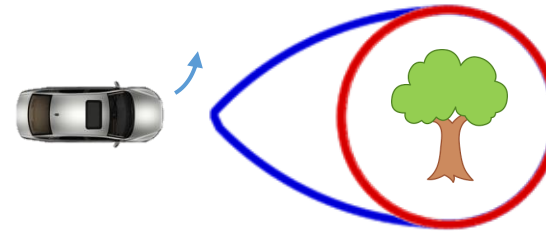
- System model
- Goal region



Control policy

Backward reachable set
(States leading to goal)

Reachability analysis



- $\mathcal{A}(t) = \{\bar{\mathbf{x}}: \exists \Gamma[\mathbf{u}](\cdot), \forall \mathbf{u}(\cdot), \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(t) = \bar{\mathbf{x}}, \mathbf{x}(0) \in \mathcal{T}\}$

Backward reachable set (states leading to danger)

Control policy

- Model of robot
- Unsafe region



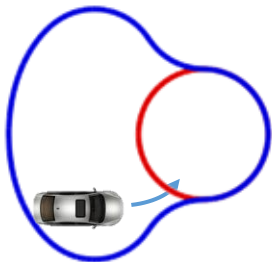
Control policy

Backward reachable set (states leading to goal)

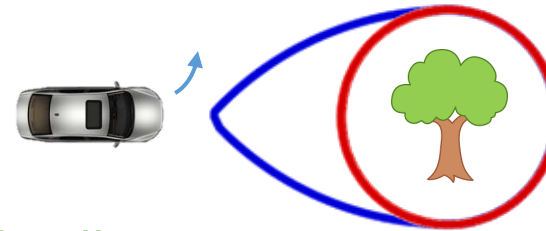
- Model of robot
- Goal region



- $\mathcal{R}(t) = \{\bar{\mathbf{x}}: \forall \Gamma[\mathbf{u}](\cdot), \exists \mathbf{u}(\cdot), \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(t) = \bar{\mathbf{x}}, \mathbf{x}(0) \in \mathcal{T}\}$



Reachability analysis



States at time t satisfying the following:

there exists a disturbance such that for all control, system enters target set at $t = 0$

$$\bullet \mathcal{A}(t) = \{\bar{\mathbf{x}}: \exists \Gamma[\mathbf{u}](\cdot), \forall \mathbf{u}(\cdot), \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(t) = \bar{\mathbf{x}}, \mathbf{x}(0) \in \mathcal{T}\}$$

- Model of robot
- Unsafe region



Backward reachable set (states leading to danger)

Control policy

- Model of robot
- Goal region



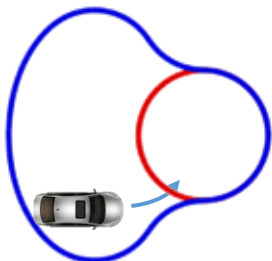
Control policy

Backward reachable set (states leading to goal)

$$\bullet \mathcal{R}(t) = \{\bar{\mathbf{x}}: \forall \Gamma[\mathbf{u}](\cdot), \exists \mathbf{u}(\cdot), \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(t) = \bar{\mathbf{x}}, \mathbf{x}(0) \in \mathcal{T}\}$$

States at time t satisfying the following:

for all disturbances, there exists a control such that system enters target set at $t = 0$



From HJI to reachability analysis

- Computation of the BRS entails solving a differential game where the outcome is Boolean (the system either reaches the target set or not)
- One can “encode” this Boolean outcome in the HJI PDE by (1) removing the running cost and (2) picking the final cost to denote set membership
 - Value function at each state is the worst case terminal value you can reach

From HJI to reachability analysis

- Hamilton-Jacobi Equation

- $0 = \frac{\partial J}{\partial t} + \max_{\mathbf{d}} \min_{\mathbf{u}} \left[g(\mathbf{x}, \mathbf{u}, \mathbf{d}) + \frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right], J(\mathbf{x}, 0) = h(\mathbf{x})$

- Remove running cost

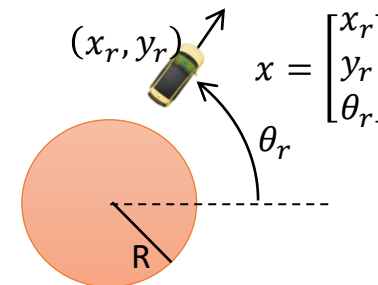
- $0 = \frac{\partial J}{\partial t} + \max_{\mathbf{d}} \min_{\mathbf{u}} \left[\frac{\partial J}{\partial \mathbf{x}} \cdot f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right], J(\mathbf{x}, 0) = h(\mathbf{x})$

- Pick final cost such that

- $\mathbf{x} \in \mathcal{T} \Leftrightarrow h(\mathbf{x}) \leq 0$

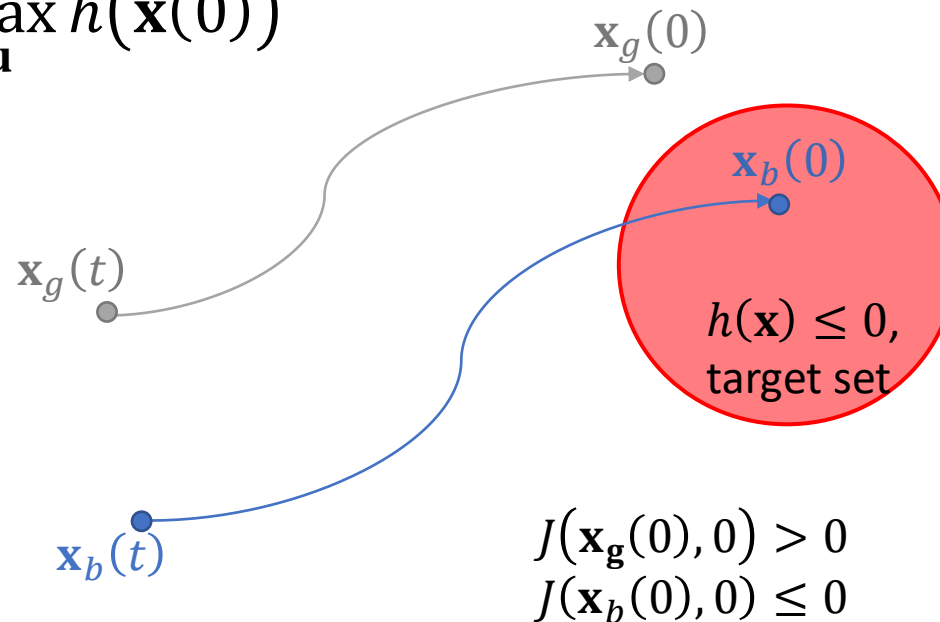
- Example: If $\mathcal{T} = \left\{ \mathbf{x}: \sqrt{x_r^2 + y_r^2} \leq R \right\} \subseteq \mathbb{R}^3$, we can pick

$$h(x_r, y_r, \theta_r) = \sqrt{x_r^2 + y_r^2} - R$$



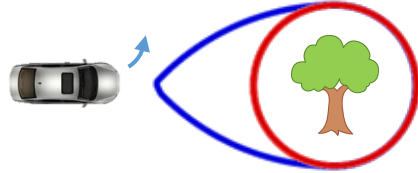
Pick Final Cost

- Why is this correct?
 - Final state $\mathbf{x}(0)$ is in \mathcal{T} if and only if $h(\mathbf{x}(0)) \leq 0$
 - To **avoid** \mathcal{T} , control should maximize $h(\mathbf{x}(0))$
 - Worst-case disturbance would minimize
 - $J(\mathbf{x}, t) = \min_{\Gamma[\mathbf{u}]} \max_{\mathbf{u}} h(\mathbf{x}(0))$



Reaching vs. Avoiding

- Avoiding danger



- BRS definition

$$\mathcal{A}(t) = \{\bar{\mathbf{x}}: \exists \Gamma[\mathbf{u}](\cdot), \forall \mathbf{u}(\cdot), \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(t) = \bar{\mathbf{x}}, \mathbf{x}(0) \in \mathcal{T}\}$$

- Value function

$$J(\mathbf{x}, t) = \min_{\Gamma[\mathbf{u}]} \max_{\mathbf{u}} h(\mathbf{x}(0))$$

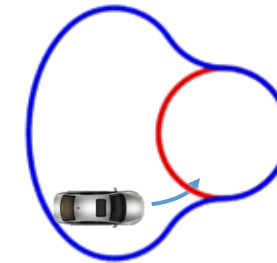
- HJI

$$\frac{\partial J}{\partial t} + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[\left(\frac{\partial J}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right] = 0$$

- Optimal control

$$\mathbf{u}^* = \arg \max_{\mathbf{u}} \min_{\mathbf{d}} \left(\frac{\partial J}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

- Reaching a goal



- BRS definition

$$\mathcal{R}(t) = \{\bar{\mathbf{x}}: \forall \Gamma[\mathbf{u}](\cdot), \exists \mathbf{u}(\cdot), \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(t) = \bar{\mathbf{x}}, \mathbf{x}(0) \in \mathcal{T}\}$$

- Value function

$$J(\mathbf{x}, t) = \max_{\Gamma[\mathbf{u}]} \min_{\mathbf{u}} h(\mathbf{x}(0))$$

- HJI

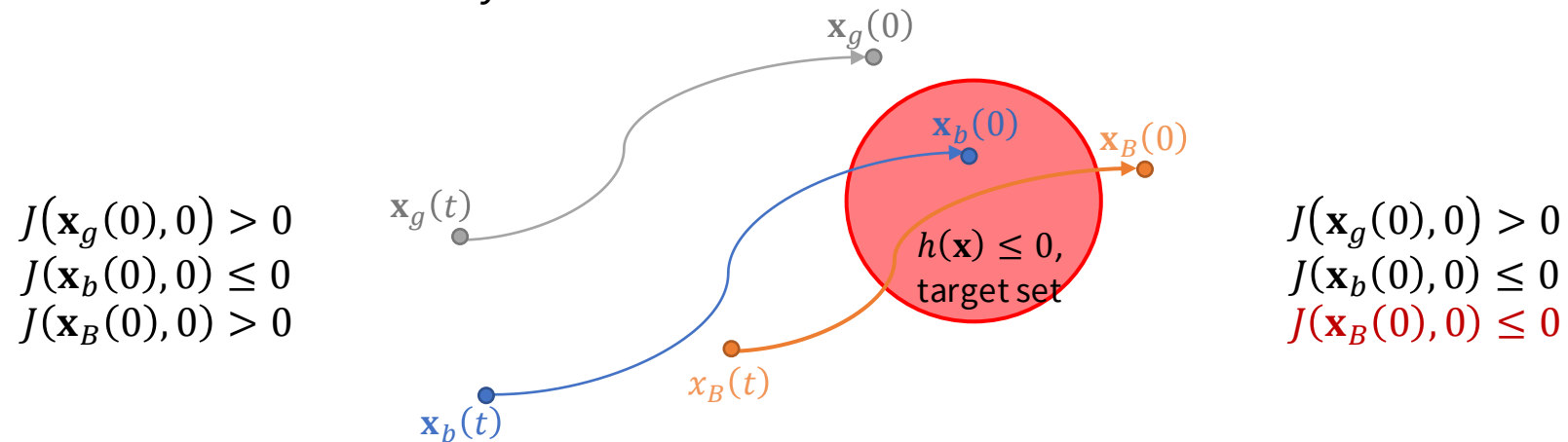
$$\frac{\partial J}{\partial t} + \min_{\mathbf{u}} \max_{\mathbf{d}} \left[\left(\frac{\partial J}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right] = 0$$

- Optimal control

$$\mathbf{u}^* = \arg \min_{\mathbf{u}} \max_{\mathbf{d}} \left(\frac{\partial J}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

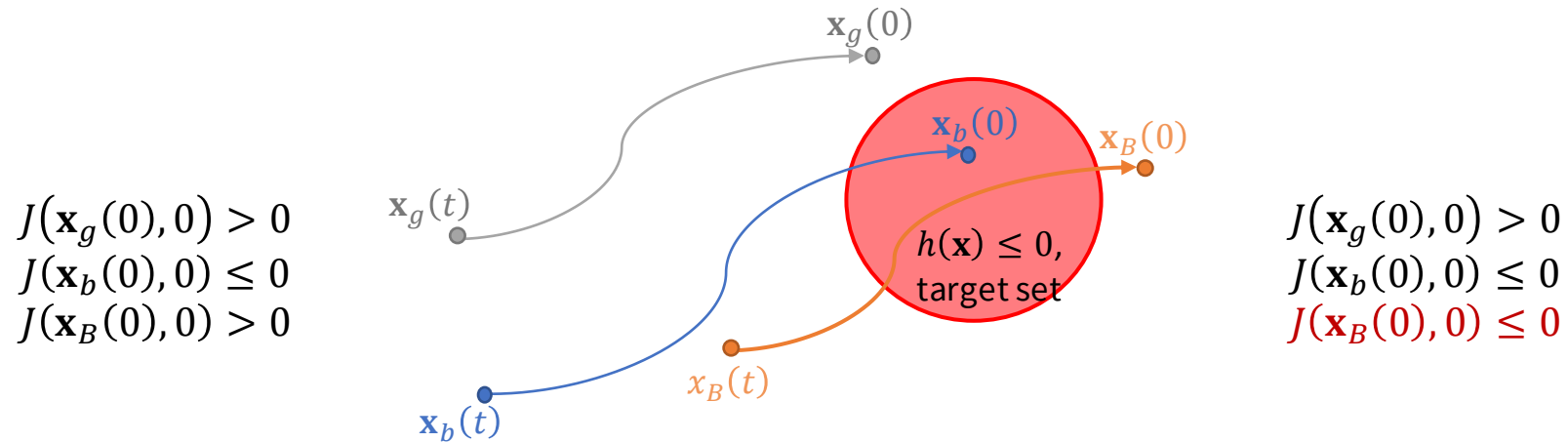
“Sets” vs. “Tubes”

- Backward reachable set (BRS)
 - Only final time matters
 - Initial states that pass through target are not necessarily in BRS
 - Not ideal for safety
- Backward reachable tube (BRT)
 - Keep track of entire time duration
 - Initial states that pass through target are in BRT
 - Used to make safety guarantees



“Sets” vs. “Tubes”

- Backward reachable set (BRS)
- Backward reachable tube (BRT)



Value function definition

$$J(\mathbf{x}, t) = \min_{\Gamma[\mathbf{u}]} \max_{\mathbf{u}} h(\mathbf{x}(0))$$

Value function obtained from

$$\frac{\partial J}{\partial t} + \max_{\mathbf{u}} \min_{\mathbf{d}} \left[\left(\frac{\partial J}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right] = 0$$

Value function definition

$$J(\mathbf{x}, t) = \min_{\Gamma[\mathbf{u}]} \max_{\mathbf{u}} \min_{\tau \in [t, 0]} h(\mathbf{x}(\tau))$$

Value function obtained from

$$\frac{\partial J}{\partial t} + \min \left\{ \max_{\mathbf{u}} \min_{\mathbf{d}} \left[\left(\frac{\partial J}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, \mathbf{u}, \mathbf{d}) \right], 0 \right\} = 0$$

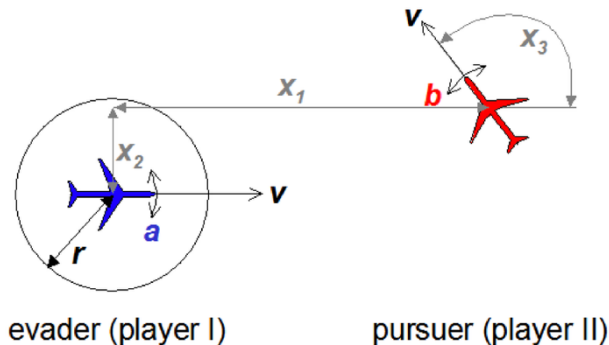
Example: pursuit/evasion with two identical vehicles

- With evader (a), pursuer (b) dynamics

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \end{bmatrix} = \begin{bmatrix} v \cos(\theta_a) \\ v \sin(\theta_a) \\ u_a \end{bmatrix}, \quad \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta}_b \end{bmatrix} = \begin{bmatrix} v \cos(\theta_b) \\ v \sin(\theta_b) \\ u_b \end{bmatrix}, \quad u_a, u_b \in [-u_{\max}, u_{\max}]$$

we consider the relative system in (a)'s frame

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -v + v \cos(x_3) + u_a x_2 \\ v \sin(x_3) - u_a x_1 \\ u_b - u_a \end{bmatrix}$$



Courtesy of
[Ian Mitchell](#),
“[ToolboxLS](#)”,
Section 2.6.1

Computational aspects

- Computational complexity (traditional PDE solver)
 - $J(\mathbf{x}, t)$ is computed on an $(n + 1)$ -dimensional grid
 - $n \leq 5$ is reasonable; larger requires some compromises
 - Dimensionality reduction methods (decoupling) sometimes help
- Alternatives/related approaches
 - Sacrifice global optimality
 - Give up guarantees
 - NN-based PDE solvers
 - Sampling-based methods
 - Reinforcement learning

Next time

- Model Predictive Control