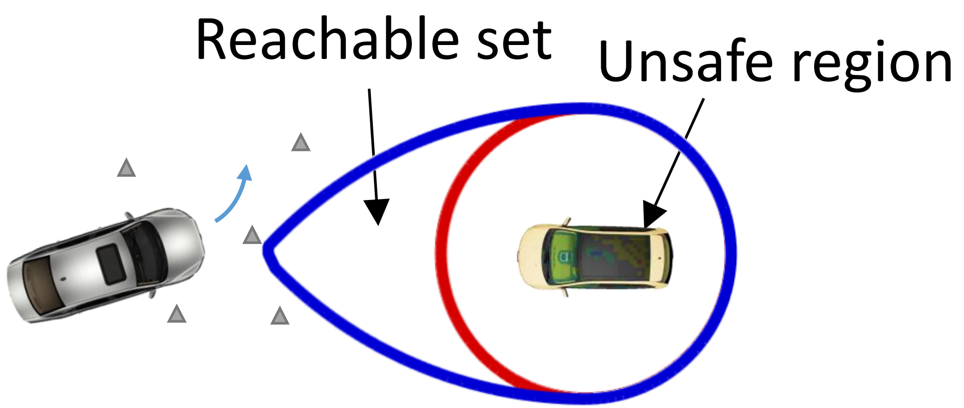
# HJ Reachability Analysis

CMPT 882

Mar. 1

# Reachability Analysis: Avoidance



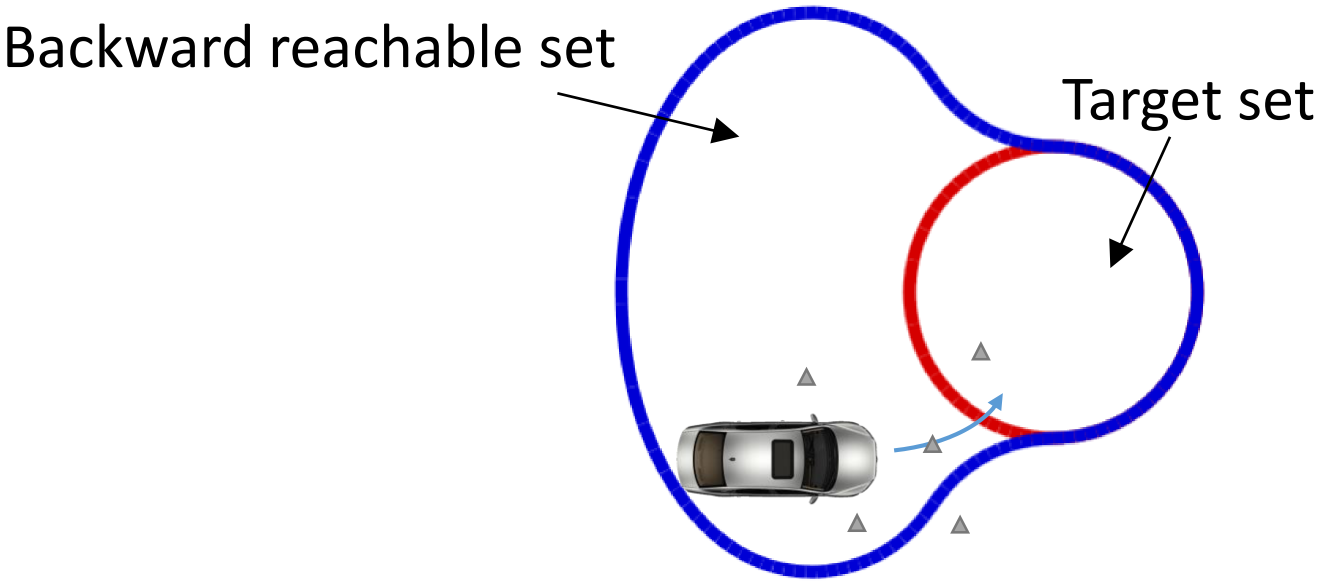

Assumptions:

- Model of robot
- Unsafe region: Obstacle

Control policy

Backward reachable set
(States leading to danger)

# Reachability Analysis: Goal Reaching



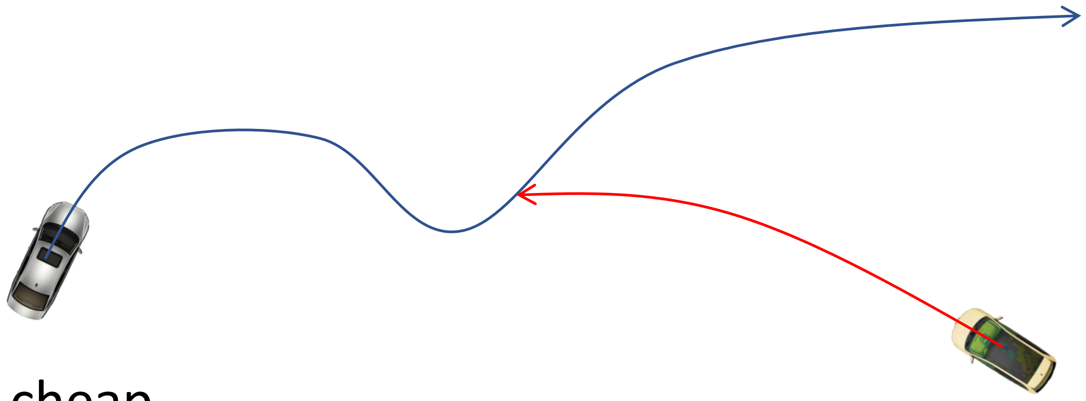

- Model of robot
- Goal region

Control policy

Backward reachable set
(States leading to goal)

# Information Pattern

- Control: chosen by "ego" robot
- Disturbances: chosen by other robot (or weather gods)
  - Assume worst case



- "Open-loop" strategies
  - Ego robot declares entire plan
  - Other robot responds optimally (worst-case)
  - Conservative, unrealistic, but computationally cheap

- "Non-anticipative" strategies
  - Other robot acts based on state and control trajectory up current time
  - Notation: $d(\cdot) = \Gamma[u](\cdot)$
  - Disturbance still has the advantage: it gets to react to the control!

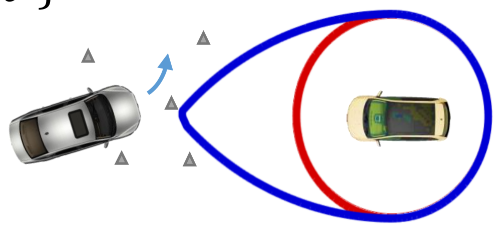
# Reachability Analysis

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

- Model of robot
- Unsafe region

Backward reachable set (States leading to danger)

Control policy



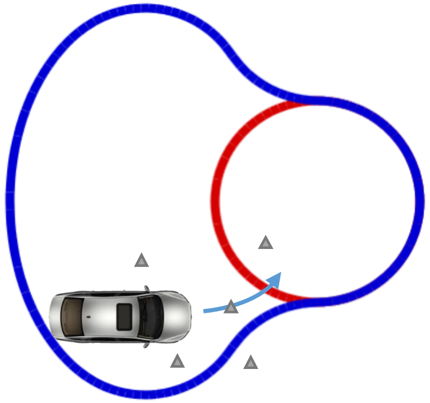
- $\dot{x} = f(x,u,d)$
- $\mathcal{T}$

- $u^*(t,x)$

- Model of robot
- Goal region

Control policy

Backward reachable set (States leading to goal)



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

# Reachability Analysis

States at time $t$ satisfying the following:

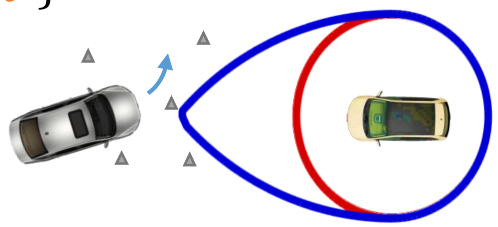
there exists a disturbance such that for all control, system enters target set at $t = 0$

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

- Model of robot
- Unsafe region

Backward reachable set (States leading to danger)

Control policy



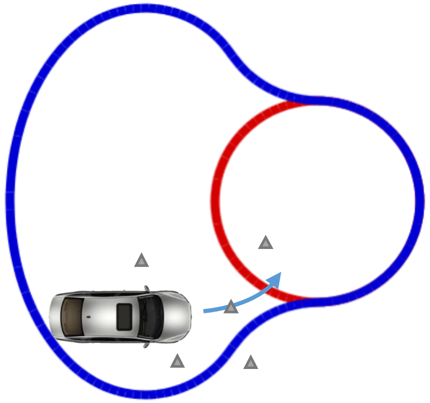
- $\dot{x} = f(x,u,d)$
- $\mathcal{T}$

- $u^*(t,x)$

- Model of robot
- Goal region

Control policy

Backward reachable set (States leading to goal)



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

States at time $t$ satisfying the following:

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

# Computing Reachable Sets: Hamilton-Jacobi Approach

- **Start from continuous time dynamic programming**
- Observe that disturbances do not affect the procedure
- Remove running cost
- Pick final cost intelligently

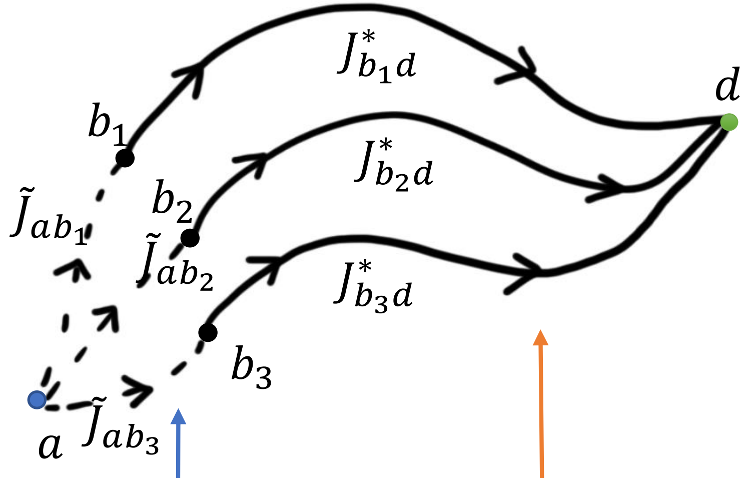
# Dynamic Programming: Continuous Time

- Let $J(x(t),t) = \int_t^0 C\big(x(s),u(s)\big)ds + l\big(x(T)\big)$ "Cost to go"

$$V(x(t),t) \coloneqq J^*(x(t),t) = \min_{u_{[t,0]}(\cdot)} \left[\int_t^0 C\big(x(s),u(s)\big)ds + l\big(x(T)\big)\right]$$

Write out time interval explicitly for clarity

"Value function", "$J^*(x(t),t)$"




- Dynamic programming principle:

$$V(x(t),t) = \min_{u_{[t,t+\delta]}(\cdot)} \left[\int_t^{t+\delta} C\big(x(s),u(s)\big)ds + V(x(t+\delta),t+\delta)\right]$$

  - Approximate integral and Taylor expand $V(x(t+\delta),t+\delta)$
  - Derive Hamilton-Jacobi partial differential equation (HJ PDE)

# Dynamic Programming: Continuous Time

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{u_{[t,t+\delta]}(\cdot)} \Bigg[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s)\big)ds}_{C(x(t),u(t))\delta} + \underbrace{V\big(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u)}, t+\delta\big)}_{V(x(t),t) + \frac{\partial V}{\partial x}\cdot \delta f(x(t),u(t)) + \frac{\partial V}{\partial t}\delta} \Bigg]$$

  - Omit $t$ dependence…

$$V(x,t) = \min_u \left[ C(x,u)\delta + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u) + \frac{\partial V}{\partial t}\delta \right]$$

Assume constant $u_{[t,t+\delta]}$ → Optimization over a vector, not a function!

  - $V(x,t)$ does not depend on $u$

$$V(x,t) = V(x,t) + \min_u \left[ C(x,u)\delta + \frac{\partial V}{\partial x}\cdot \delta f(x,u) + \frac{\partial V}{\partial t}\delta \right]$$

# Dynamic Programming: Continuous Time

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{u_{[t,t+\delta]}(\cdot)} \left[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s)\big)ds}_{C\left(x(t),u(t)\right)\delta} + \underbrace{V\big(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u)},t+\delta\big)}_{V(x(t),t) + \frac{\partial V}{\partial x}\cdot \delta f(x(t),u(t)) + \frac{\partial V}{\partial t}\delta} \right]$$

  - Omit $t$ dependence…

$$V(x,t) = \min_u \left[ C(x,u)\delta + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u) + \frac{\partial V}{\partial t}\delta \right]$$

Assume constant $u_{[t,t+\delta]}$ → Optimization over a vector, not a function!

  - $V(x,t)$ does not depend on $u$

$$\cancel{V(x,t)} = \cancel{V(x,t)} + \frac{\partial V}{\partial t}\delta + \min_u \left[ C(x,u)\delta + \frac{\partial V}{\partial x}\cdot \delta f(x,u) \right]$$

# Dynamic Programming: Continuous Time

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{u_{[t,t+\delta]}(\cdot)} \left[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s)\big)ds}_{C\left(x(t),u(t)\right)\delta} + \underbrace{V\big(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u)}, t+\delta\big)}_{V(x(t),t) + \frac{\partial V}{\partial x}\cdot \delta f(x(t),u(t)) + \frac{\partial V}{\partial t}\delta} \right]$$

- Omit $t$ dependence…

$$V(x,t) = \min_u \left[ C(x,u)\delta \ + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u) + \frac{\partial V}{\partial t}\delta \right]$$

Assume constant $u_{[t,t+\delta]}$ → Optimization over a vector, not a function!

- $V(x,t)$ does not depend on $u$

$$0 = \frac{\partial V}{\partial t}\cancel{\delta} + \min_u \left[ C(x,u)\cancel{\delta} \ + \frac{\partial V}{\partial x}\cdot \cancel{\delta} f(x,u) \right]$$

# Dynamic Programming: Continuous Time

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{u_{[t,t+\delta]}(\cdot)} \Bigg[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s)\big)ds}_{C\left(x(t),u(t)\right)\delta} + \underbrace{V(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u)}, t+\delta)}_{V(x(t),t) + \frac{\partial V}{\partial x}\cdot \delta f(x(t),u(t)) + \frac{\partial V}{\partial t}\delta} \Bigg]$$

  - Omit $t$ dependence…

$$V(x,t) = \min_u \left[ C(x,u)\delta + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u) + \frac{\partial V}{\partial t}\delta \right]$$

Assume constant $u_{[t,t+\delta]}$ → Optimization over a vector, not a function!

  - $V(x,t)$ does not depend on $u$

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

# Computing Reachable Sets: Hamilton-Jacobi Approach

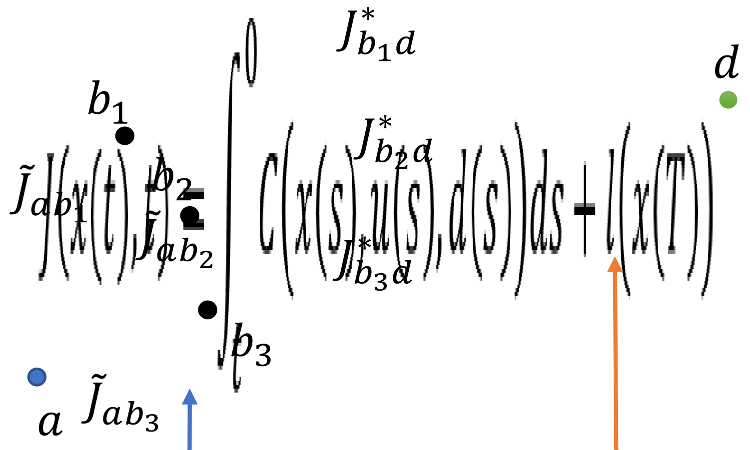
- Start from continuous time dynamic programming
- **Observe that disturbances do not affect the procedure**
- Remove running cost
- Pick final cost intelligently

# Dynamic Programming: Continuous Time (with disturbances)

- Let $J(x(t), t) = \int_t^0 C(x(s), u(s), d(s))ds + l(x(T))$ "Cost to go"

$$V(x(t), t) = \min_{\Gamma[u](\cdot)} \max_{u(\cdot)} \left[ \int_t^0 C(x(s), u(s), d(s))ds + l(x(T)) \right]$$

Worst-case disturbance -- do the opposite of the control




- Dynamic programming principle:

$$V(x(t), t) = \min_{\Gamma[u](\cdot)} \max_{u(\cdot)} \left[ \int_t^{t+\delta} C(x(s), u(s), d(s))ds + V(x(t+\delta), t+\delta) \right]$$

  - Approximate integral and Taylor expand $V(x(t+\delta), t+\delta)$
  - Derive Hamilton-Jacobi partial differential equation (HJ PDE)

# Dynamic Programming: Continuous Time (with disturbances)

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{\Gamma[u](\cdot)} \max_{u(\cdot)} \left[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s),d(s)\big)ds}_{C\left(x(t),u(t),d(t)\right)\delta} + \underbrace{V(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u,d)},t+\delta)}_{V(x(t),t)+\frac{\partial V}{\partial x}\cdot\delta f(x(t),u(t))+\frac{\partial V}{\partial t}\delta} \right]$$

- Omit $t$ dependence…

$$V(x,t) = \max_{u} \min_{d} \left[ C(x,u,d)\delta \; + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u,d) + \frac{\partial V}{\partial t}\delta \right]$$

  - Assume constant $u$ and $d$ → Optimization over vectors, not functions!
  - Order of max and min reverse: disturbance has the advantage

- $V(x,t)$ does not depend on $u$ or $d$

$$V(x,t) = V(x,t) + \max_{u} \min_{d} \left[ C(x,u,d)\delta \; + \frac{\partial V}{\partial x}\cdot \delta f(x,u,d) + \frac{\partial V}{\partial t}\delta \right]$$

# Dynamic Programming: Continuous Time (with disturbances)

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{\Gamma[u](\cdot)} \max_{u(\cdot)} \left[ \underbrace{\int_t^{t+\delta} c\big(x(s),u(s),d(s)\big)ds}_{c\big(x(t),u(t),d(t)\big)\delta} + \underbrace{V(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u,d)},t+\delta)}_{V(x(t),t)+\frac{\partial V}{\partial x}\cdot\delta f(x(t),u(t))+\frac{\partial V}{\partial t}\delta} \right]$$

- Omit $t$ dependence…

$$V(x,t) = \max_u \min_d \left[ c(x,u,d)\delta + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u,d) + \frac{\partial V}{\partial t}\delta \right]$$

  - Assume constant $u$ and $d$ → Optimization over vectors, not functions!
  - Order of max and min reverse: disturbance has the advantage

- $V(x,t)$ does not depend on $u$ or $d$

$$\cancel{V(x,t)} = \cancel{V(x,t)} + \max_u \min_d \left[ c(x,u,d)\delta + \frac{\partial V}{\partial x}\cdot \delta f(x,u,d) + \frac{\partial V}{\partial t}\delta \right]$$

# Dynamic Programming: Continuous Time (with disturbances)

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{\Gamma[u](\cdot)} \max_{u(\cdot)} \left[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s),d(s)\big)ds}_{C\left(x(t),u(t),d(t)\right)\delta} + \underbrace{V(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u,d)}, t+\delta)}_{V(x(t),t) + \frac{\partial V}{\partial x}\cdot \delta f(x(t),u(t)) + \frac{\partial V}{\partial t}\delta} \right]$$

  - Omit $t$ dependence…

$$V(x,t) = \max_u \min_d \left[ C(x,u,d)\delta + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u,d) + \frac{\partial V}{\partial t}\delta \right]$$

    - Assume constant $u$ and $d$ → Optimization over vectors, not functions!
    - Order of max and min reverse: disturbance has the advantage

  - $V(x,t)$ does not depend on $u$ or $d$

$$0 = \frac{\partial V}{\partial t}\cancel{\delta} + \max_u \min_d \left[ C(x,u,d)\cancel{\delta} + \frac{\partial V}{\partial x}\cdot \cancel{\delta} f(x,u,d) \right]$$

# Dynamic Programming: Continuous Time (with disturbances)

- Approximations for small $\delta$:

$$V(x(t),t) = \min_{\Gamma[u](\cdot)} \max_{u(\cdot)} \left[ \underbrace{\int_t^{t+\delta} C\big(x(s),u(s),d(s)\big)ds}_{C\left(x(t),u(t),d(t)\right)\delta} + \underbrace{V\big(\overbrace{x(t+\delta)}^{x(t)+\delta f(x,u,d)}, t+\delta\big)}_{V(x(t),t) + \frac{\partial V}{\partial x}\cdot \delta f(x(t),u(t)) + \frac{\partial V}{\partial t}\delta} \right]$$

- Omit $t$ dependence…

$$V(x,t) = \max_u \min_d \left[ C(x,u,d)\delta + V(x,t) + \frac{\partial V}{\partial x}\cdot \delta f(x,u,d) + \frac{\partial V}{\partial t}\delta \right]$$

  - Assume constant $u$ and $d$ → Optimization over vectors, not functions!
  - Order of max and min reverse: disturbance has the advantage

- $V(x,t)$ does not depend on $u$ or $d$

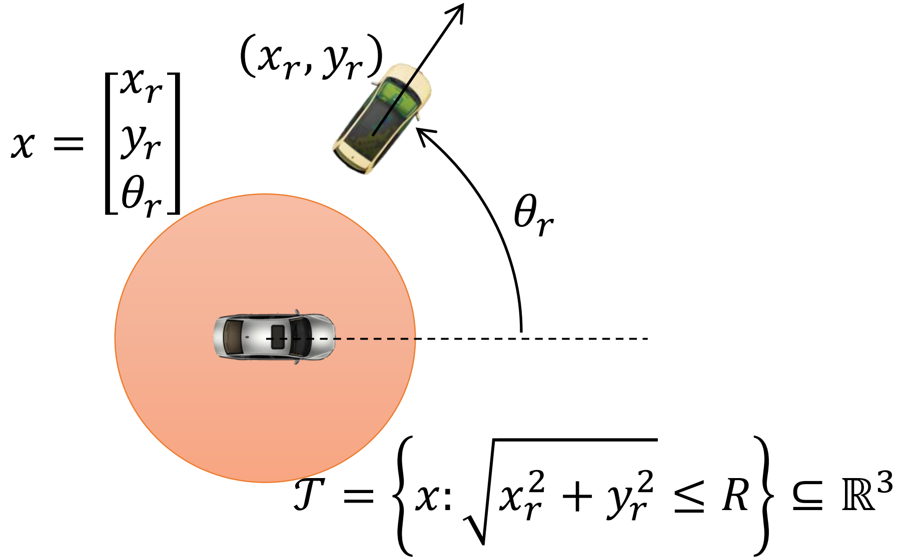
$$0 = \frac{\partial V}{\partial t} + \max_u \min_d \left[ C(x,u,d) + \frac{\partial V}{\partial x}\cdot f(x,u,d) \right]$$

# Computing Reachable Sets: Hamilton-Jacobi Approach

- Start from continuous time dynamic programming
- Observe that disturbances do not affect the procedure
- **Remove running cost**
- **Pick final cost intelligently**

# Remove Running Cost, Pick Final Cost

- Hamilton-Jacobi Equation
  - $0 = \frac{\partial V}{\partial t} + \max_{d} \min_{u} \left[ C(x,u,d) + \frac{\partial V}{\partial x} \cdot f(x,u,d) \right], \; V(0,x) = l(x)$

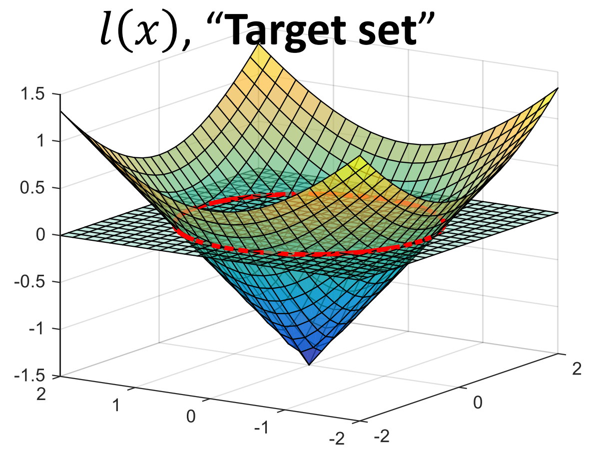



- Remove running cost
  - $0 = \frac{\partial V}{\partial t} + \max_{d} \min_{u} \left[ \frac{\partial V}{\partial x} \cdot f(x,u,d) \right], \; V(0,x) = l(x)$

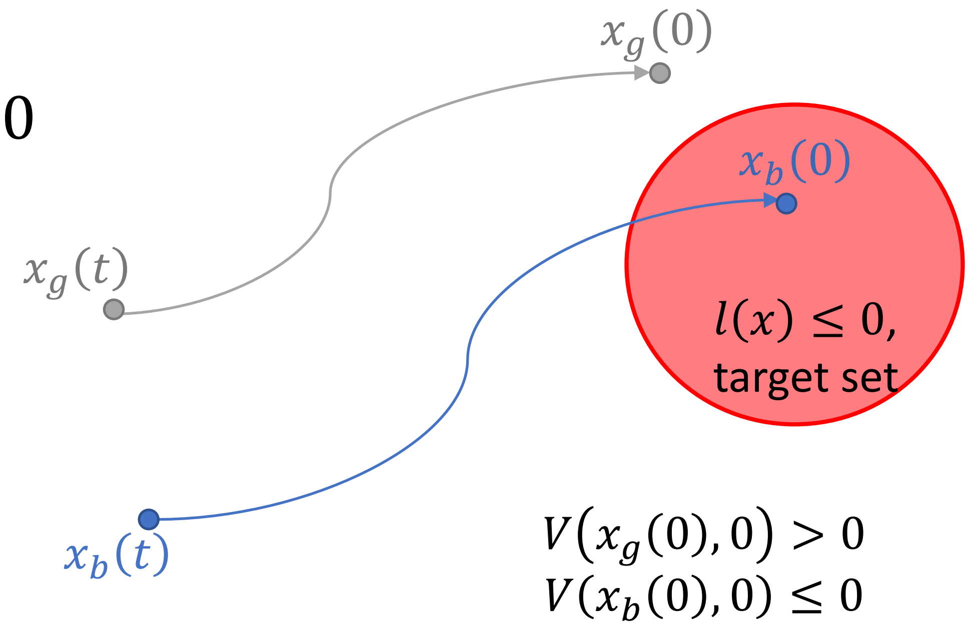
- Pick final cost such that
  - $x \in \mathcal{T} \Leftrightarrow l(x) \le 0$
  - Example: If $\mathcal{T} = \left\{ x : \sqrt{x_r^2 + y_r^2} \le R \right\} \subseteq \mathbb{R}^3$, we can pick $l(x_r, y_r, \theta_r) = \sqrt{x_r^2 + y_r^2} - R$

# Pick Final Cost

- Pick final cost such that
  - $x \in \mathcal{T} \Leftrightarrow l(x) \leq 0$
  - If $\mathcal{T} = \left\{x: \sqrt{x_r^2 + y_r^2} \leq R\right\} \subseteq \mathbb{R}^3$, we can pick $l(x_r, y_r, \theta_r) = \sqrt{x_r^2 + y_r^2} - R$

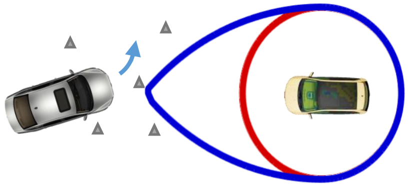



- Why is this correct?
  - Final state $x(0)$ is in $\mathcal{T}$ if and only if $l\big(x(0)\big) \leq 0$
  - To avoid $\mathcal{T}$, control should maximize $l\big(x(0)\big)$
    - Worst-case disturbance would minimize
  - $V(t, x) = \min_{\Gamma[u]} \max_{u} l\big(x(0)\big)$

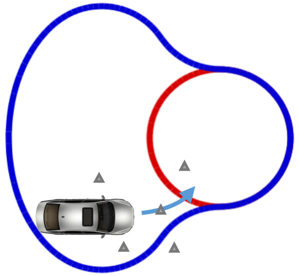


# Reaching vs. Avoiding

- Avoiding danger



- Reaching a goal



**Avoiding danger**

- BRS definition

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

- Value function

$$V(t,x) = \min_{\Gamma[u]} \max_{u} l\big(x(0)\big)$$

- HJ PDE

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

- Optimal control

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

**Reaching a goal**

- BRS definition

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

- Value function

$$V(t,x) = \max_{\Gamma[u]} \min_{u} l\big(x(0)\big)$$

- HJ PDE

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

- Optimal control

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

# Optimal Control and Disturbance

- Example: Scalar control and disturbance affine system
  - Dynamics: $\dot{x} = f(x) + \sum_i g_i(x)u_i + \sum_j h_j(x)d_j \, , x \in \mathbb{R}$
  - Control and disturbance constraints: $u_i \in [\underline{u}_i, \bar{u}_i], d_j \in [\underline{d}_j, \bar{d}_j]$

$$\frac{\partial V}{\partial t} + \min_{\{u_i \in [\underline{u}_i, \bar{u}_i]\}} \max_{\{d_j \in [\underline{d}_j, \bar{d}_j]\}} \left[ \left( \frac{\partial V}{\partial x} \right)^{\top} f(x, u, d) \right] = 0$$

$$\frac{\partial V}{\partial t} + \min_{\{u_i \in [\underline{u}_i, \bar{u}_i]\}} \max_{\{d_j \in [\underline{d}_j, \bar{d}_j]\}} \left[ \frac{\partial V}{\partial x} \left( f(x) + \sum_i g_i(x)u_i + \sum_j h_j(x)d_j \right) \right] = 0$$

$$\frac{\partial V}{\partial t} + \min_{\{u_i \in [\underline{u}_i, \bar{u}_i]\}} \max_{\{d_j \in [\underline{d}_j, \bar{d}_j]\}} \left[ \frac{\partial V}{\partial x} f(x) + \sum_i \frac{\partial V}{\partial x} g_i(x)u_i + \sum_j \frac{\partial V}{\partial x} h_j(x)d_j \right] = 0$$

$$u_i = \begin{cases} \underline{u}_i, & \frac{\partial V}{\partial x} g_i(x) \geq 0 \\ \bar{u}_i, & \frac{\partial V}{\partial x} g_i(x) < 0 \end{cases} \qquad d_j = \begin{cases} \underline{d}_j, & \frac{\partial V}{\partial x} g_i(x) < 0 \\ \bar{d}_j, & \frac{\partial V}{\partial x} g_i(x) \geq 0 \end{cases}$$

# Optimal Control and Disturbance

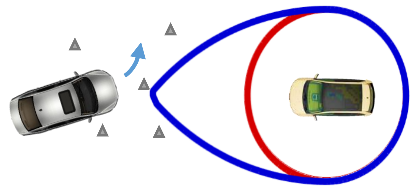
- Easy to compute for many common types of control and disturbance constraints

- Interval constraints: easy -- see last slide
- Polytopic constraints: easy -- test all vertices
- Other: ideally, need analytic expression
  - Optimization needs to be done at every grid point!

Eg. $\frac{\partial V}{\partial t} + \min_u \max_d \left[ \left( \frac{\partial V}{\partial x} \right)^\top f(x, u, d) \right] = 0$

# Terminology

- Minimal backward reachable set
  - $\mathcal{A}(t) = \{\bar{x}: \exists\Gamma[u](\cdot), \forall u(\cdot), \dot{x} = f(x,u,d), x(t) = \bar{x}, x(0) \in \mathcal{T}\}$
  - Control minimizes size of reachable set



- Maximal backward reachable set
  - $\mathcal{R}(t) = \{\bar{x}: \forall\Gamma[u](\cdot), \exists u(\cdot), \dot{x} = f(x,u,d), x(t) = \bar{x}, x(0) \in \mathcal{T}\}$
  - Control maximizes size of reachable set