Hamilton-Jacobi Reachability Analysis I

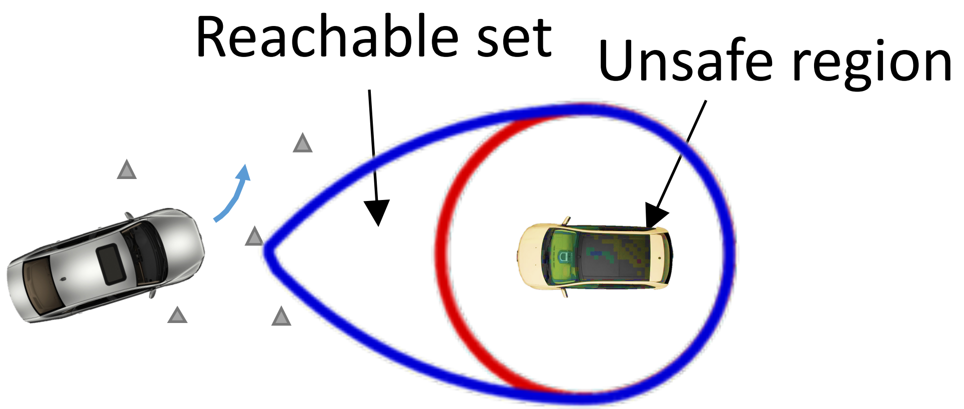
CMPT 419/983

Mo Chen

SFU Computing Science

16/10/2019

Reachability Analysis: Avoidance



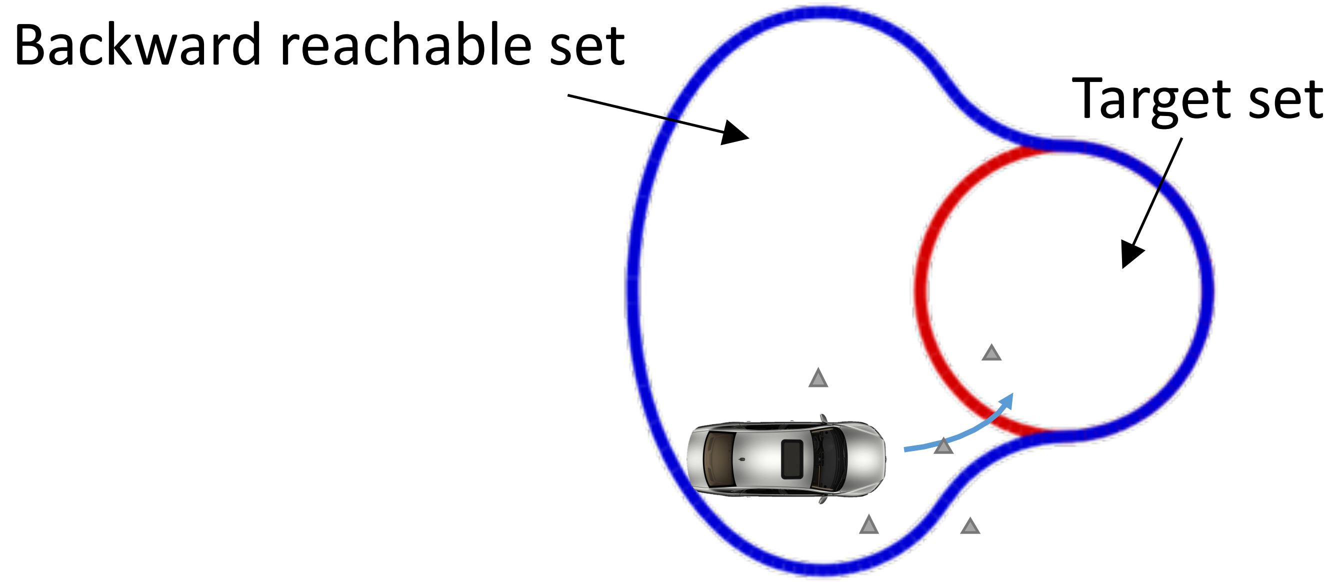
Assumptions:
- Model of robot
- Unsafe region: Obstacle

→

Control policy

Backward reachable set
(States leading to danger)

Reachability Analysis: Goal Reaching



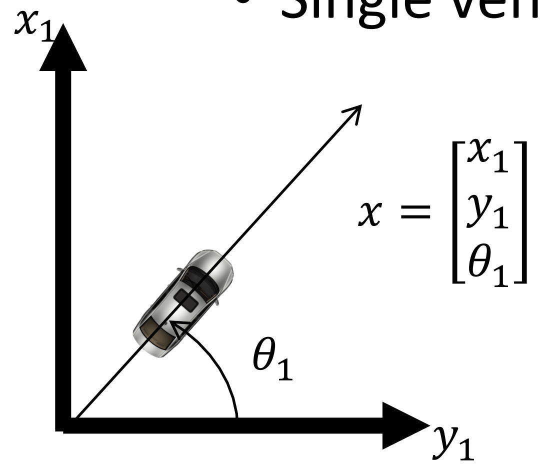
- Model of robot
- Goal region

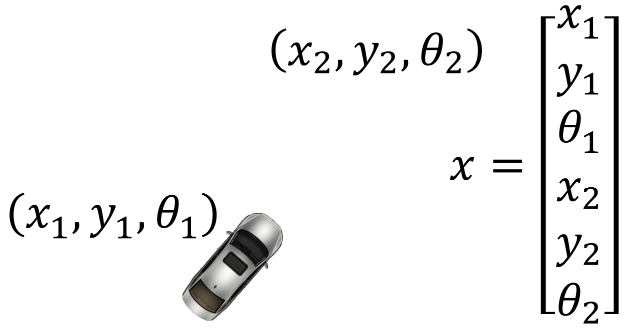
Control policy

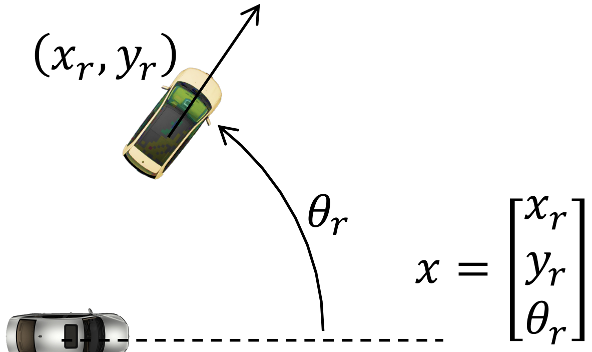
Backward reachable set
(States leading to goal)

Assumptions

- System dynamics: $\dot{x} = f(x, u, d), t \leq 0$ (by convention, final time is 0)
- State x
  - Single vehicle, multiple vehicle, relative coordinates





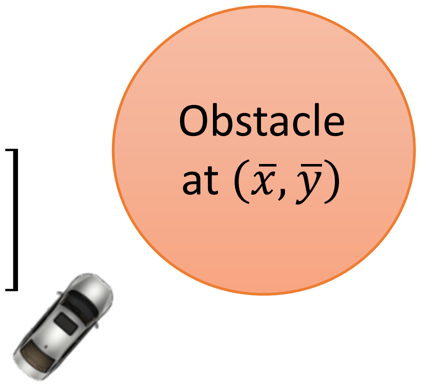


- Disturbance d: uncontrolled factors that affect the system, such as wind
  - Can be used to model other agents, when state includes them
  - Assume worst case

Assumptions

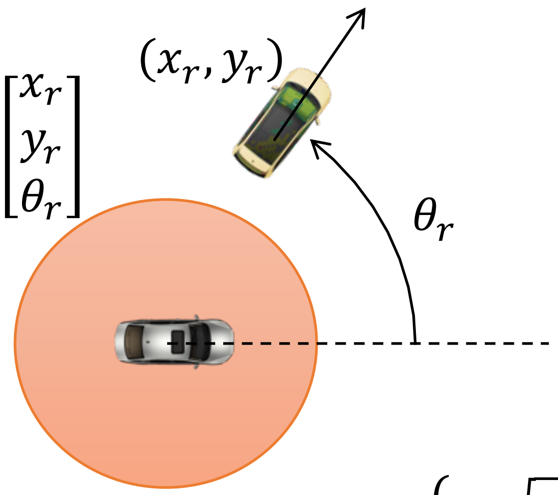
- "Target set", $\mathcal{T}$
  - Can specify set of states leading to danger
  - Expressed through set notation

$$x = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix}$$



$$\mathcal{T} = \left\{x : \sqrt{(x_1 - \bar{x})^2 + (y_1 - \bar{y})^2} \leq r\right\} \subseteq \mathbb{R}^3$$

$$x = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix}$$

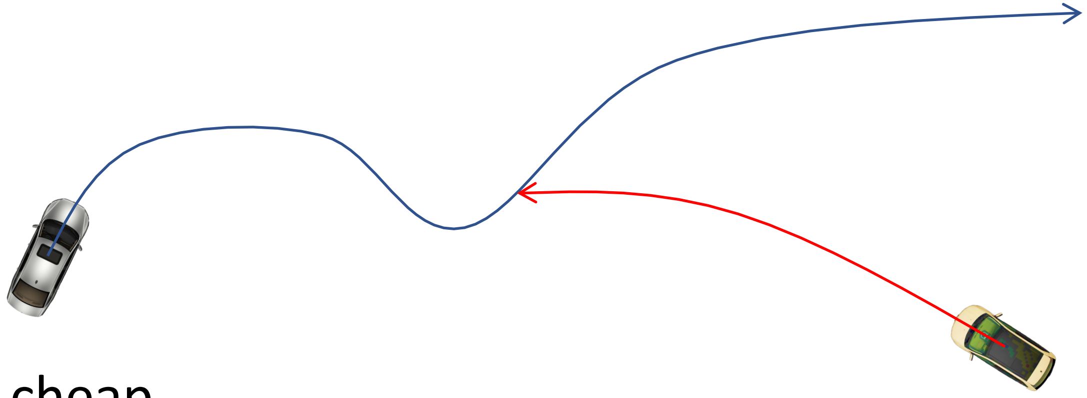


$$\mathcal{T} = \left\{x : \sqrt{x_r^2 + y_r^2} \leq R\right\} \subseteq \mathbb{R}^3$$

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

- Model of robot
- Unsafe region

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

- Model of robot
- Goal region

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

Backward reachable set (States leading to danger)

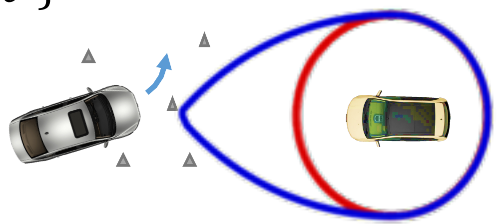
Control policy

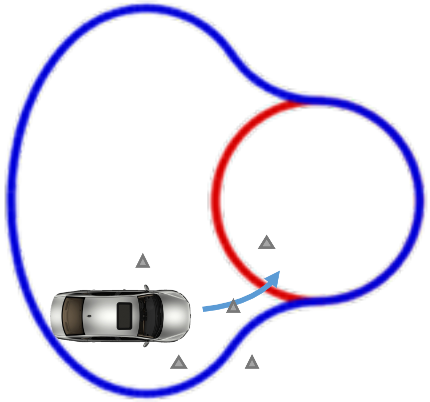
- $u^*(t, x)$

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:

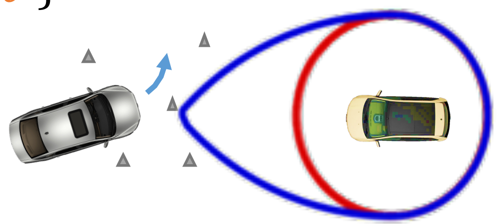
there exists a disturbance such that for all control, system enters target set at $t = 0$

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

- Model of robot
- Unsafe region

→ Backward reachable set (States leading to danger)

Control policy



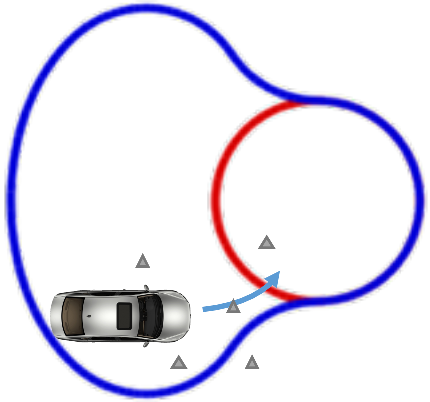
- $\dot{x} = f(x,u,d)$
- $\mathcal{T}$

- $u^*(t,x)$

- 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

$$\underset{u(\cdot)}{\text{minimize}} \overbrace{l\big(T, x(T)\big)}^{\text{Final cost}} + \overbrace{\int_0^T c\big(x(t), u(t)\big)dt}^{\text{Running cost}}$$

Cost functional, $J\big(x(\cdot), u(\cdot)\big)$

$$\text{subject to } \dot{x}(t) = f\big(x(t), u(t)\big)$$

Dynamic model

$$x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, x(0) = x_0$$

- Let $J(t, x(t)) = l\big(T, x(T)\big) + \int_t^T c\big(x(t), u(t)\big)dt$
  - $V(0, x(0)) = \min_{u(\cdot)} J\big(0, x(0)\big)$ is what we want

- Strategy:
  - make a "discrete time" argument with Δt
  - Let $\Delta t \to 0$

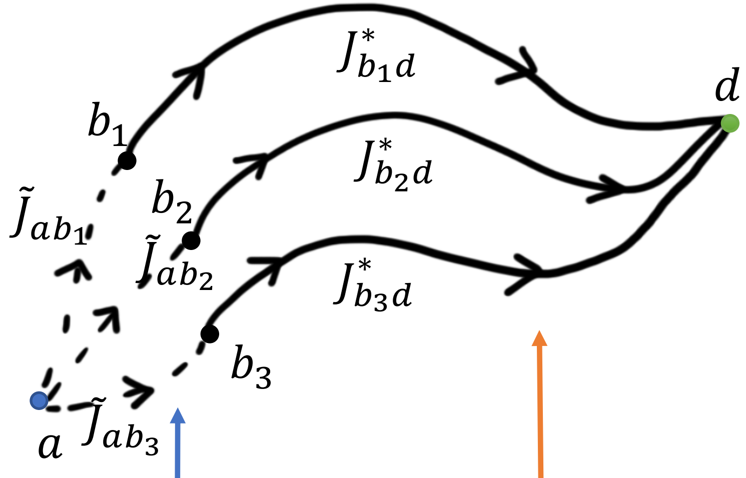
Dynamic Programming: Continuous Time

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

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

Write out time interval explicitly for clarity

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



- Dynamic programming principle:

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

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

Dynamic Programming: Continuous Time

- Approximations for small δ:

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

  - Omit t dependence…

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

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

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

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

Dynamic Programming: Continuous Time

- Approximations for small δ:

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

- Omit t dependence…

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

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

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

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

Dynamic Programming: Continuous Time

- Approximations for small δ:

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

- Omit t dependence…

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

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

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

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

Dynamic Programming: Continuous Time

- Approximations for small δ:

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

- Omit t dependence…

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

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

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

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

Computing Reachable Sets: Hamilton-Jacobi Approach

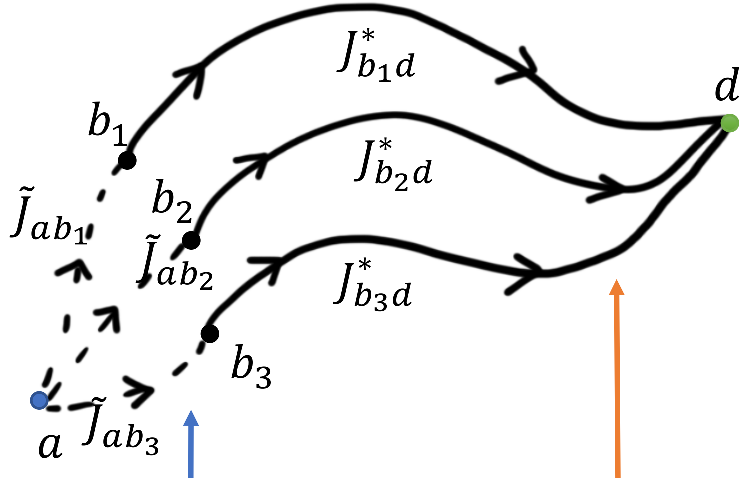
- Start from continuous time dynamic programming
- **Observe that disturbances do not affect the procedure**
- Remove running cost
- Pick final cost intelligently

Dynamic Programming: Continuous Time (with disturbances and $T = 0$)

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

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

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



- Dynamic programming principle:

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

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

Dynamic Programming: Continuous Time (with disturbances and $T = 0$)

- Approximations for small δ:

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

  - Omit t dependence…

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

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

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

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

Dynamic Programming: Continuous Time (with disturbances and $T = 0$)

- Approximations for small δ:

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

- Omit t dependence…

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

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

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

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

Dynamic Programming: Continuous Time (with disturbances and $T = 0$)

- Approximations for small δ:

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

  - Omit t dependence…

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

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

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

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

Dynamic Programming: Continuous Time (with disturbances and $T = 0$)

- Approximations for small δ:

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

- Omit t dependence…

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

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

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

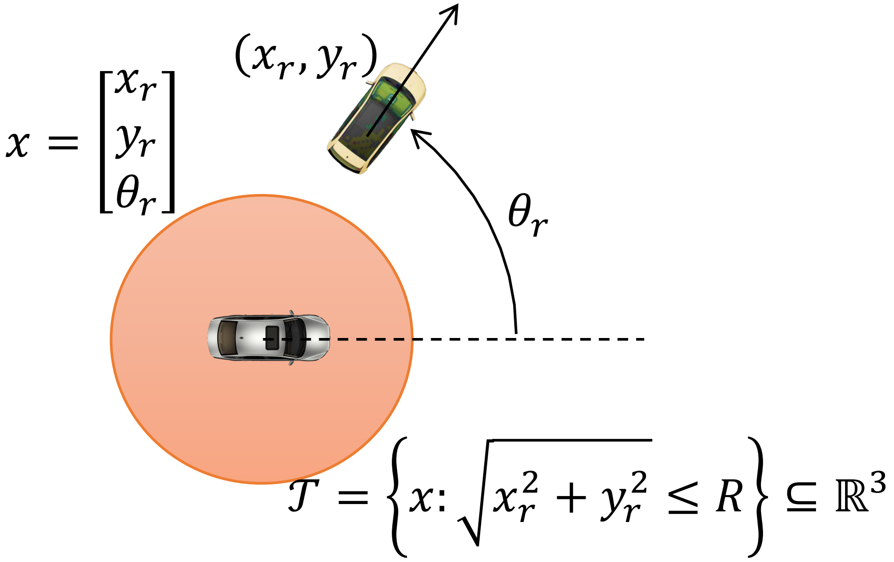
$$0 = \frac{\partial V}{\partial t} + \max_u \min_d \left[c(x,u,d) + \left(\frac{\partial V}{\partial x}\right)^\top f(x,u,d) \right]$$

Computing Reachable Sets: Hamilton-Jacobi Approach

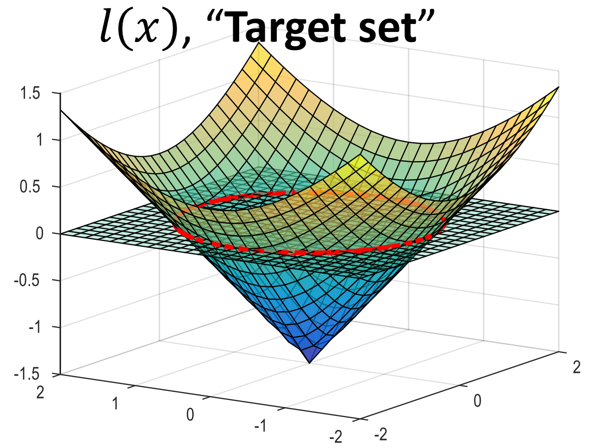
- Start from continuous time dynamic programming
- Observe that disturbances do not affect the procedure
- **Remove running cost**
- **Pick final cost intelligently**

Remove Running Cost, Pick Final Cost

- Hamilton-Jacobi Equation
  - $0 = \frac{\partial V}{\partial t} + \max_d \min_u \left[c(x,u,d) + \left(\frac{\partial V}{\partial x}\right)^\top f(x,u,d) \right], \; V(0,x) = l(x)$
- Remove running cost
  - $0 = \frac{\partial V}{\partial t} + \max_d \min_u \left[\left(\frac{\partial V}{\partial x}\right)^\top f(x,u,d) \right], \; V(0,x) = l(x)$
- 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 $l(x_r, y_r, \theta_r) = \sqrt{x_r^2 + y_r^2} - R$

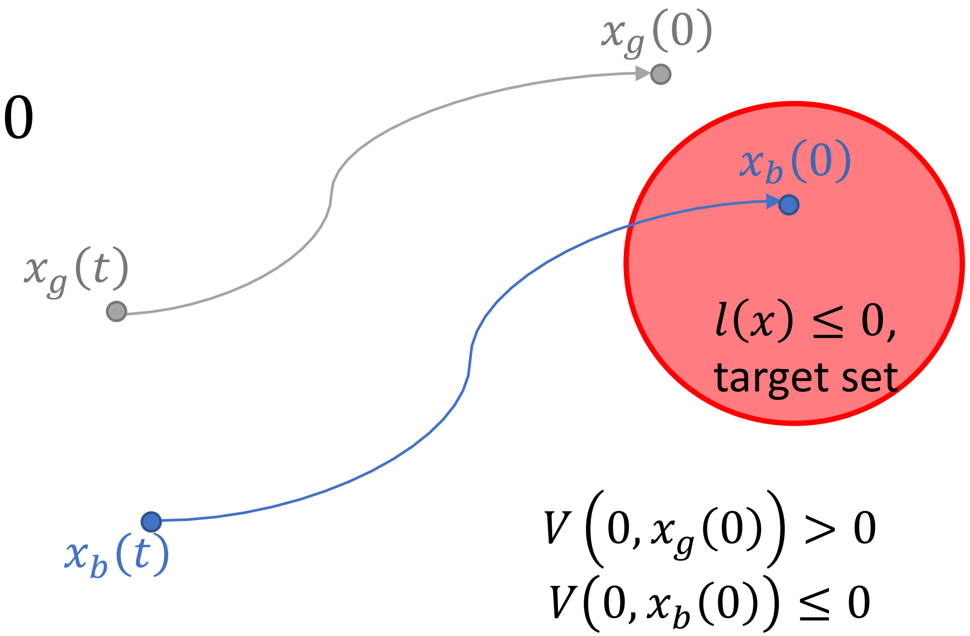


Pick Final Cost



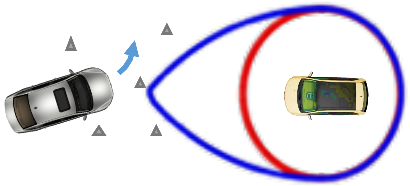
- Pick final cost such that
  - $x \in \mathcal{T} \Leftrightarrow l(x) \leq 0$
  - If $\mathcal{T} = \left\{x: \sqrt{x_r^2 + y_r^2} \leq R\right\} \subseteq \mathbb{R}^3$, we can pick $l(x_r, y_r, \theta_r) = \sqrt{x_r^2 + y_r^2} - R$

- 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



  - 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

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

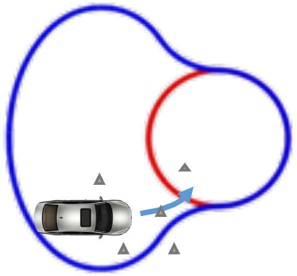
  - HJ PDE

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

  - 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

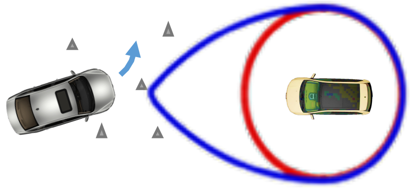
- Interval constraints: easy -- see last slide
- Polytopic constraints: easy -- test all vertices
- Other: ideally, need analytic expression
  - Optimization needs to be done at every grid point!

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



- Minimal and maximal backward reachable tube
  - $\bar{\mathcal{A}}(t) = \{\bar{x}: \exists \Gamma[u](\cdot), \forall u(\cdot), \dot{x} = f(x,u,d), x(t) = \bar{x}, \exists s \in [t,0], x(s) \in \mathcal{T}\}$
  - $\bar{\mathcal{R}}(t) = \{\bar{x}: \forall \Gamma[u](\cdot), \exists u(\cdot), \dot{x} = f(x,u,d), x(t) = \bar{x}, \exists s \in [t,0], x(s) \in \mathcal{T}\}$