Lecture 6
IR, Spring '24
Andrea Bajcsy

Last time:

- ☐ HJI Equation
- ☐ start of safety Analysis

Today:

- ☐ HJ Safety Analysis!
- ☐ safety filters

Formalizing safety via reachability

We now have a handle on how to solve general robust optimal control problems w/ potentially multi-agents. But what if we wanted to ensure that our system abides by some state constraints? For example, what if we want to synthesize an optimal control that guarantees that our robot never hits an obstacle? What are the initial conditions from which robot is doomed to collide?

These questions fall under reachability analysis which is a fundamental problem of identifying "if a certain state of a system is reachable from an initial state of the system":

⇒ fundamental to program analysis, to dynamical systems, to biology!

Safety Analysis Roadmap

Define in MATH the set we want to compute automatically	→	Encode the problem computationally/numerically	→	Compute the unsafe set + safety-preserving controller ⇒ as an O.C. problem!
↑ Characterize		↑ ENCODE		↑ COMPUTE

While there are many ways to compute safe/unsafe sets, we will study Hamilton-Jacobi Reachability analysis of safe sets & controllers. Why HJ?

1) automatically handles control bounds / state constraints
2) — " — synthesizes safe set AND safety controller
3) general nonlinear systems $\dot{x} = f(x,u)$
4) multi-agent interactions (non-deterministic uncertainty)

Define in MATH the set we want to compute automatically → Encode the problem computationally / numerically → Compute the unsafe set + safety-preserving controller ⇒ as an O.C. problem!

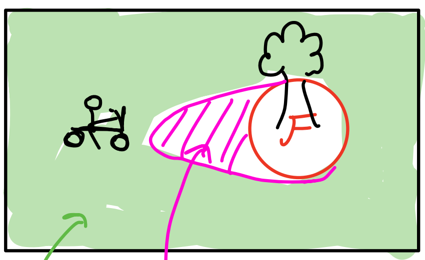
Characterize | ENCODE | COMPUTE

Reachability is very expressive framework for defining safety ("collision") & liveness ("goal-reach")

let's focus on BACKWARDS REACHABLE TUBES (BRT) of dyn. sys.

BRT is the set of all states of a system that will eventually reach some "target" set despite the robot's best control effort.

let $\mathcal{F}$ be the failure set; then the BRT represents the potential unsafe set of states for the system & thus should be avoided.



let $\mathcal{L} \subseteq \mathbb{R}^n$ (e.g. $\mathcal{F}$!) be the set of states we are interested in performing analysis on. let $BRT(t) \subseteq \mathbb{R}^n$ be the BRT at time t (typically unsafe set):

> [ROBUST] BACKWARDS REACHABLE TUBE (BRT) of a set $\mathcal{L} \subseteq \mathbb{R}^n$ and a dynamical system $\dot{x} = f(x, u, d)$ is:
>
> $$BRT(t) = \{ \underbrace{x \in \mathbb{R}^n}_{\text{initial states}} : \underbrace{\forall u(\cdot) \in \mathbb{U}_t^T}_{\text{for all ctrl signals}}, \underbrace{\exists d(\cdot) \in \mathbb{D}_t^T}_{\text{exists a disturbance}}, \underbrace{X_{x,t}^{u,d}(\tau) \in \mathcal{L}}_{\text{s.t. state traj. enters } \mathcal{L} \ldots} \text{ for some } \underbrace{\tau \in [t, T]}_{\ldots \text{ @ some time } \tau} \}$$
>
> this is set of initial states @ time t

Intuitively, BRT(t) computes the set of all starting states from which no matter what the controller does, there exists a disturbance that drives the system into $\mathcal{L}$ (e.g., our failure set).

⊛ TO compute unsafe set, you need to compute BRT of $\mathcal{F}$!

Failure set ≡ constraint
Unsafe set ≡ BRT

Define in MATH the set we want to compute automatically	→	Encode the problem computationally/numerically	→	Compute the unsafe set + safety-preserving controller ⇒ as an O.C. problem!
↑ Characterize		↑ ENCODE		↑ COMPUTE

HJ Reachability

know connection (mathematically) btwn. BRT & the failure/target set.

let's talk about computing! HJ formulates this computation as an optimal control problem! lets us use all the O.C. tools (and in fact, SOTA algorithms) to compute BRT automatically.

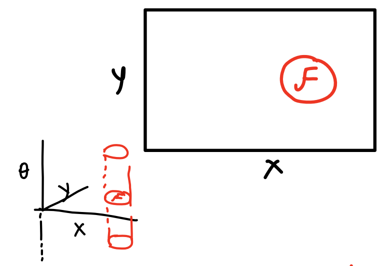
HJ Reachability uses level set methods to convert BRT characterization to an O.C. problem.

↳ numerical analysis on arbitrarily-shaped $\mathcal{F}$

↳ propagate the influence of ctrl./dist. on the "growth" of BRT

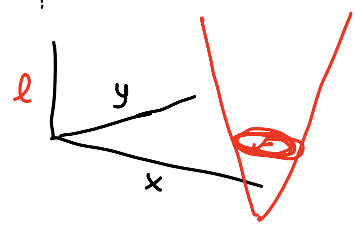
PROCEDURE:

1) We have a failure set $\mathcal{F} \subseteq \mathbb{R}$



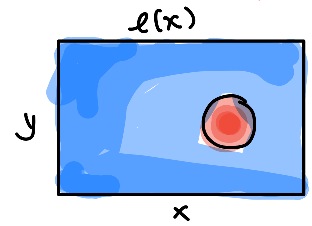
2) Define a function $\ell(x) : \mathbb{R}^n \to \mathbb{R}$ to implicitly encode this failure set:

$$\ell(x) \leq 0 \iff x \in \mathcal{F}$$



One such function is signed distance to $\mathcal{F}$

→ S.D. > 0 when x outside $\mathcal{F}$
S.D. < 0 —"— x inside $\mathcal{F}$
S.D. $= 0$ —"— @ boundary

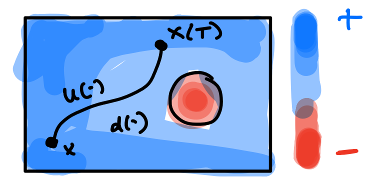


$$\ell(x) = \sqrt{x^2+y^2} - R$$

3) we want to optimize $u(\cdot)$ wrt. $\ell(x)$ since $\ell(x)$ is our optimal control cost function!

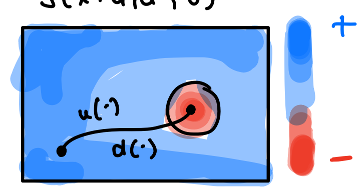
$$J(x, u(\cdot), d(\cdot), t) = \min_{\tau \in [t,T]} \ell(\mathbf{x}_{x,t}^{u(\cdot),d(\cdot)}(\tau))$$

$J(x,u,d,t) > 0$



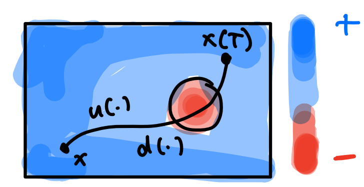
⇒ "closest our system ever gets to $\mathcal{F}$ when applying $(u(\cdot), d(\cdot))$ starting from x at time t"

$J(x,u,d,t) < 0$



⇒ By looking @ the sign of $J(\cdot,\cdot,\cdot)$ we can tell if our traj. ever entered $\mathcal{F}$!

$J(x,u,d,t) < 0$



$$V(x,t) = \max_{u(\cdot) \in \Gamma_u} \min_{d(\cdot) \in \Gamma_d} J(x, u(\cdot), d(\cdot), t)$$

$$= \max_{u(\cdot) \in \Gamma_u} \min_{d(\cdot) \in \Gamma_d} \left(\min_{\tau \in [t,T]} \ell(x(\tau)) \right)$$

⇒ $V(x^{init}, t) \leq 0$ for some state x^{init} ⇒ controller $(u(\cdot))$ tried hardest but couldn't do anything to prevent $x_\tau \in \mathcal{F}$

⇒ $x^{init} \in BRT(t)$

⇒ $\boxed{BRT(t) = \{x : V(x,t) \leq 0\}}$

Key diff. btwn what we had before is <u>min over time!</u>

<u>Good NEWS</u>: we can still use the principle of optimality & D.P. to compute V!

$$V(x,t) = \max_{u(\cdot)\in\Gamma_u} \min_{d(\cdot)\in\Gamma_d} \left(\min_{\tau\in[t,T]} \ell(x(\tau)) \right)$$

$$= \max_{u(\cdot)\in\Gamma_u} \min_{d(\cdot)\in\Gamma_d} \boxed{\min} \left(\underbrace{\min_{\tau\in[t,t+\delta]} \ell(x(\tau))}_{\text{"either my } \mathcal{F} \text{ violation happens right now"}}, \overbrace{\min_{s\in[t+\delta,T]} \ell(x(s))}^{\text{"... or, this failure happens in future"}} \right)$$

$$= \max_{u(\cdot)\in\Gamma_u} \min_{d(\cdot)\in\Gamma_d} \min \left(\min_{\tau\in[t,t+\delta]} \ell(x(\tau)), \underbrace{\min_{s\in[t+\delta,T]} \ell(x(s))}_{:= J(x(t+\delta),\, u(\cdot),\, d(\cdot),\, t+\delta)} \right)$$

$$= \max_{u(\cdot)\in\Gamma_u} \min_{d(\cdot)\in\Gamma_d} \min \left(\min_{\tau\in[t,t+\delta]} \ell(x(\tau)), \underbrace{V(x(t+\delta), t+\delta)}_{\text{by principle of opt}} \right)$$

<u>Hamilton-Jacobi Variational Inequality</u> (HJI-VI)

"remember if we ever fail"

$$\min \left\{ \ell(x) - V(x,t),\ \underbrace{\frac{\partial V}{\partial t} + \max_{u} \min_{d} \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,d)}_{\text{HJB-PDE!}} \right\} = 0$$

$$V(x,T) = \ell(x)$$

Define in <u>MATH</u> the set we want to compute automatically → <u>Encode</u> the problem computationally/numerically → <u>Compute</u> the unsafe set + safety-preserving controller ⇒ as an <u>O.C. problem!</u>

Characterize | ENCODE | COMPUTE

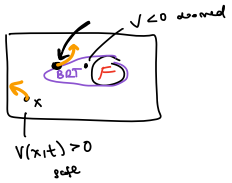
Getting optimal control

Recall that system should stay away from $\mathcal{F}$ encoded as < 0 in $\ell(x)$.

Then:

optimal value function.

$$\boxed{u^*_{safe}(x,t) = \underset{u}{\text{argmax}}\ \underset{d}{\min}\ \frac{\partial V^*(x,t)}{\partial x} \cdot f(x,u,d)} \quad (*)$$



Least-restrictive Safety Filter

nominal policy (e.g. RL, MPC, ...)

"boundary of"

$$\underbrace{u^*(x)}_{\text{executed ctrl.}} = \begin{cases} \pi_{nom}(x) & \text{if } V^*(x) > 0 \\ u^*_{safe}(x) & \text{if } V^*(x) = 0 \quad (\text{i.e. } x \in \partial BRT) \end{cases}$$

↳ from $(*)$

in practice $V^*(x) = \Delta$ where $\Delta > 0$ but small (to account for numerical error)