Last Time

- ☐ dynamic games

Lecture 6

EAIS SP'25

Andrea Bajcsy

This Time:

- ☐ compare BRT with + w/o disturbance
- ☐ scaling computation : RL
- ☐ scaling computation : SSL

Announcement HW #1 due today!

# Numerical comparison of Robust vs. Non-Robust Unsafe Set

Recall: state is $(x, y, \theta)$ of Dubins' car

dynamics are
$$\begin{aligned} \dot{x} &= \overset{1.5}{v} \cos\theta + d_x \\ \dot{y} &= v \sin\theta + d_y \\ \dot{\theta} &= u \end{aligned}$$

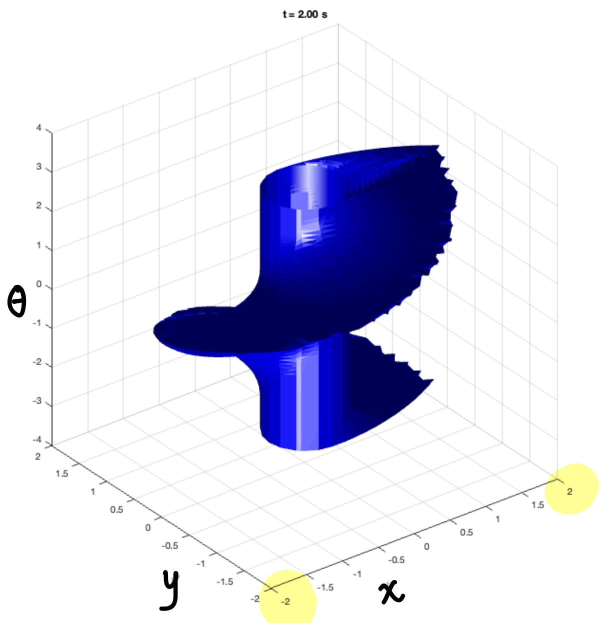
$\begin{pmatrix} d_x \\ d_y \end{pmatrix} \in \underbrace{[-0.5, 0.5]}_{\equiv \mathcal{D}}$

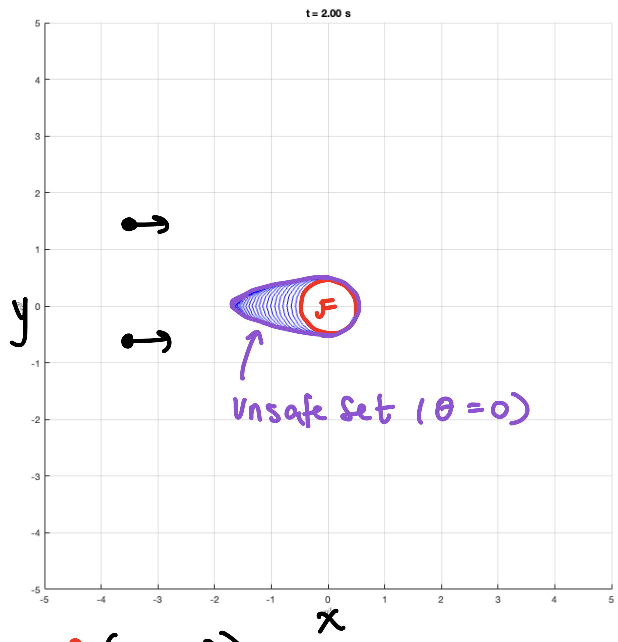
$u \in [-0.5, 0.5] \equiv \mathcal{U}$

$(x, y)$, $\theta$

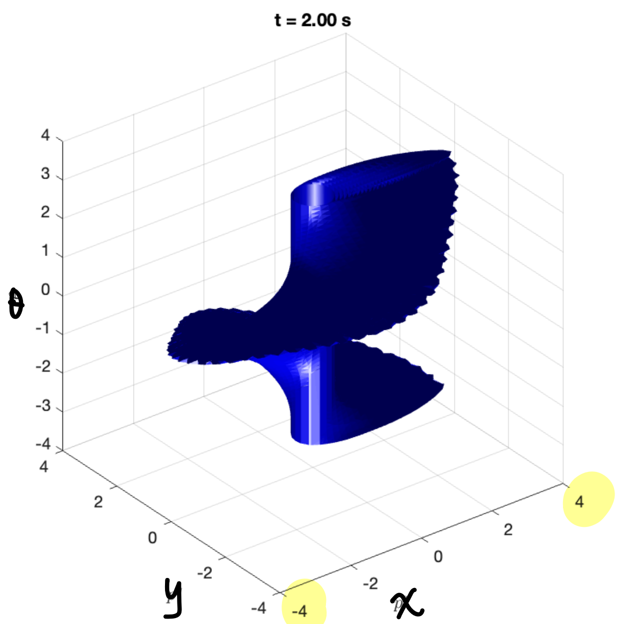
failure is cylinder @ origin with radius 0.3

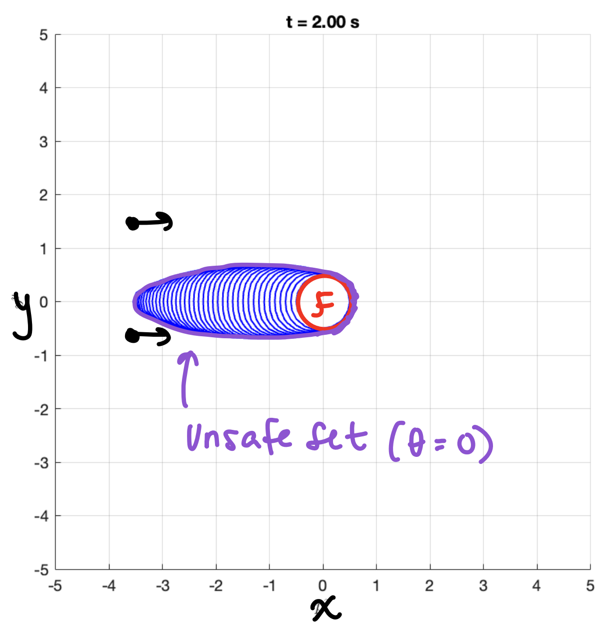
$\boxed{\text{Non-Robust BRT}}$ $\max_{u} \min_{\tau \in [t,T]} \ell(x(\tau))$



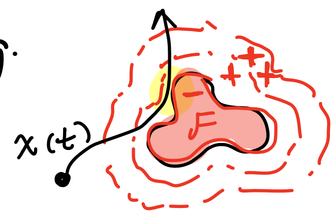

$\boxed{\text{Robust BRT}}$ $\max_{\pi_u} \min_{\pi_d} \min_{\tau \in [t,T]} \ell(x(\tau))$



## RECAP:

So far, we have formalized what is a safe set + safe controller. We also transformed this into an optimal control (or dynamic game for robustness) problem that we can solve via dynammic programming.




$\ell(x)$ encodes $\mathcal{F}$!

Now that we have the foundations, we can talk about practical challenges and frontiers of decision-theoretic safety!

### PRACTICAL CHALLENGES + RELATED RESEARCH FRONTIERS

1) scaling the computation of BRT's (i.e. value function!)

"FRONTIERS I"

2) increasing the flexibility of the safe / unsafe set
(e.g. representation is parameterized, use data to inform BRT)
3) specifying more complex failure representations
(e.g. "don't go through caution tape", "don't spill")
4) break the perfect state assumption
5) analyze more complex dynamical systems

$\boxed{\text{TODAY}}$ we will focus on (1) – scaling the computation!

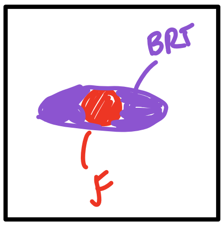
At the highest level, the key approach to scaling we will study is:

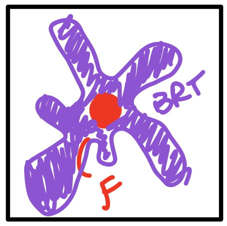
NEURAL APPROXIMATIONS to the value function

These "neural reachable tubes" will bake safety into the training process (which will leverage data) via careful construction of the signal we use for learning the neural approximation.

NN's are "agnostic" to grid resolution, and so the memory + the learning complexity scales with the underlying complexity of the set (not the resolution)

simple underlying BRT shape!
⇒ easy to represent + learn for NN!




harder underlying BRT shape
⇒ more complex to represent + learn!

Broadly, there are two "paradigms" for solving for our neural reachable tubes:

- reinforcement learning (RL)
- self-supervised learning (SSL)

## Reinforcement Learning

RL approaches leverage advances in high-fidelity simulators that model hard-to-write-down $\dot{x} = f(x,u)$ systems and they use "rollouts" (i.e. data of the robot executing its behavior in the (simulated) environment to approximate an optimal control / sequential decision-making problem.

! **KEY CHALLENGE**: RL algorithms are designed for discounted sum-of-reward problems. **BUT**, safety isn't about failing "a little" on average; it's about preventing failure **AT ALL**.

Typically, the RL problem is posed as approximating

$$V(x) := \max_{\pi} \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t)$$

discount factor $\gamma \in [0,1)$

determines importance of future rewards: $\gamma = 0 \Rightarrow$ myopic
$\gamma \to 1 \Rightarrow$ optimal

The corresponding Bellman backup for this problem (w/ discrete-time determistic dynamics) is:

"Value Iteration"

$$(*) \quad V(x) = \max_{u \in \mathcal{U}} r(x,u) + \gamma V(f(x,u))$$

An important property that allows many RL algorithms to work is the fact that (*) is a <u>contraction mapping</u>. Intuitively, this means that successive applications of (*) will <u>converge</u> to a unique <u>fixed point / solution</u>.

$V^*(x)$

The problem is that our (infinite-horizon) safety problem <u>doesn't</u> does not induce a contraction mapping in the Bellman backup.

$$V(x) := \max_{\pi} \min_{t \geq 0} \ell(x_t^{\pi}) \quad \leftarrow \text{o.c. problem}$$

$$\Downarrow$$

← Bellman Backup

$$(**) \quad V(x) = \min \left\{ \ell(x), \max_{u \in \mathcal{U}} V(f(x,u)) \right\}$$

! How do we discount (**) which has a min over time....

<u>ONE IDEA</u>: Fisac*, Lugovoy*, Rubies-Royo, Gosh, Tomlin. ICRA 2019.

$$(\star) \quad V(x) = (1-\gamma)\ell(x) + \gamma \min \left\{ \ell(x), \max_{u \in \mathcal{U}} V(f(x,u)) \right\}$$

Intuitively, they interpret $\gamma \in [0,1)$ as the probability of the episode continuing. If it continues, then the value will be the "minimum of the future" RHS $(\star)$ with probability $(1-\gamma)$, it may end (e.g. transition into a terminal state) and so your current value is $\ell(x)$.

Let's prove that $(\star)$ is contraction mapping!

**Thm** The backup operator $B[V]$:

$$B[V] := (1-\gamma)\ell(x) + \gamma \min\left\{\ell(x), \max_{u\in\mathcal{U}} V(f(x,u))\right\}$$

is a contraction mapping. Let $V, \tilde{V} : \mathcal{X} \to \mathbb{R}$

There exists $K \in [0,1)$ s.t.

$\|x\|_\infty := \max_i |x_i|$

$$\|B[V] - B[\tilde{V}]\|_\infty \leq K \|V - \tilde{V}\|_\infty$$

<u>Proof</u>: Consider $\forall x \in \mathcal{X}$

next state $x' = f(x,u)$

$$|B[V] - B[\tilde{V}]| =$$

$$\left| \cancel{(1-\gamma)\ell(x)} + \gamma \min\left\{\ell(x), \max_{u\in\mathcal{U}} V(x')\right\} - \left[\cancel{(1-\gamma)\ell(x)} + \gamma \min\left\{\ell(x), \max_{u\in\mathcal{U}} \tilde{V}(x')\right\}\right]\right|$$

cancel out

pull out of abs. value b/c $\gamma \in [0,1)$

$$= \gamma \left| \underbrace{\min\left\{\ell(x), \max_{u\in\mathcal{U}} V(x')\right\}}_{\leq \max_{u\in\mathcal{U}} V(x')} - \underbrace{\min\left\{\ell(x), \max_{u\in\mathcal{U}} \tilde{V}(x')\right\}}_{\leq \max_{u\in\mathcal{U}} \tilde{V}(x')} \right|$$

b/c of min with $\ell(x)$!

$$\leq \gamma \left| \max_{u\in\mathcal{U}} V(x') - \max_{u\in\mathcal{U}} \tilde{V}(x') \right|$$

Without loss of generality, suppose the first max is larger:

$$|B[V] - B[\tilde{V}]| \leq \gamma \left| \max_{u\in\mathcal{U}} V(x') - \max_{u\in\mathcal{U}} \tilde{V}(x') \right|$$

<u>Lemma:</u>

$$\left|\max_a f(a) - \max_a g(a)\right| \leq \max_a |f(a) - g(a)|$$

$$\leq \gamma \max_{u\in\mathcal{U}} |V(x') - \tilde{V}(x')|$$

$K \equiv \gamma$

$$\leq \gamma \sup_{x'\in\mathcal{X}} |V(x') - \tilde{V}(x')| = \gamma \|V - \tilde{V}\|_\infty \quad \blacksquare$$

look @ all next states & find the max

By def$^{n}$ of $\|\cdot\|_\infty$

Now that we have a contraction mapping, we can unlock RL algorithms, but with our safety-informed backup!

e.g. Safety Q-learning

learning rate $0 < \alpha \leq 1$

"weighted average"

$$\underbrace{Q_\theta(x,u)}_{\text{state-action value func. parameterized by } \theta} \leftarrow (1-\alpha) Q_\theta(x,u) + \underbrace{\alpha \left[ (1-\gamma)\ell(x) + \gamma \min \left\{ \ell(x), \max_{u' \in \mathcal{U}} Q_\theta(x', u') \right\} \right]}$$

new value estimated from experience

## Self-Supervised learning

Still use neural reachable tubes, but pose SSL problem

IDEA. Bansal & Tomlin. "DeepReach". ICRA 2021.

! lets use the (continuous-time) HJB-VI equation as supervision!

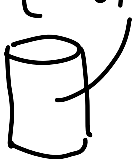
If we found a good value function approx. $V_\theta(x,t), \forall (x,t) \in \mathcal{D}$

$$\min \left\{ \ell(x) - V_\theta(x,t), \frac{\partial V_\theta}{\partial t} + \max_{u \in \mathcal{U}} \nabla_x V_\theta(x,t)^T f(x,u) \right\} = 0$$

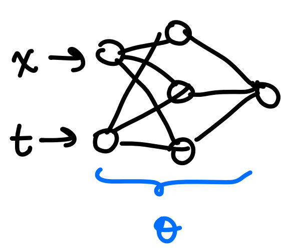
It must be that this PDE-like equation is equal zero!

Training Batch

$\{(x_i, t_i)\}_{i=1}^N$



Neural Value Function




$V_\theta(x,t)$

fwd. pass

back prop.

Loss Func

$\mathcal{L}(x, t; \theta)$

$$\mathcal{L}(x_i, t_i; \theta) = \overbrace{\text{HJ-VI Violation Error}}^{\text{the HJ-VI should} \,\stackrel{?}{=}\, 0!} + \lambda \overbrace{\text{Initial Condition}}^{V(x,T) = \ell(x)}$$

$$= \mathcal{L}_1(x_i, t_i; \theta) + \lambda\, \mathcal{L}_2(x_i, t_i; \theta)$$

$$\mathcal{L}_1 = \sum_i \underbrace{\left\| \min\left\{ \ell(x_i) - V_\theta(x_i, t_i),\ \frac{\partial V_\theta(x_i, t_i)}{\partial t} + \max_u \nabla_x V_\theta(x_i, t_i)^T f(x_i, u) \right\} \right\|}$$

we know this should $= 0$ when $V$ is optimal $\Rightarrow$ bigger the norm, the higher the loss!

$$\mathcal{L}_2 = \sum_i \| V_\theta(x_i, t_i) - \ell(x_i) \| \, \mathbb{1}\{t_i = T\}$$

ground truth value only available @ final time.