Reachability Constrained Reinforcement Learning

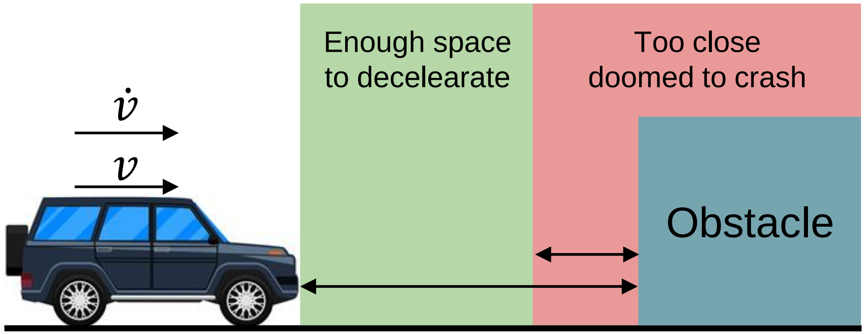
Dongjie Yu*[1] Haitong Ma*[1,2] Shengbo Eben Li[1] Jianyu Chen[3,4]




* Equal contributions
[1] School of Vehicle and Mobility, Tsinghua University
[2] John A. Paulson School of Engineering and Applied Sciences, Harvard University
[3] Institute for Interdisciplinary Information Sciences, Tsinghua University
[4] Shanghai Qizhi Institute

Motivation

Constrained/safe RL restricting *expected cumulative* costs cannot tell the **persistent** safety.

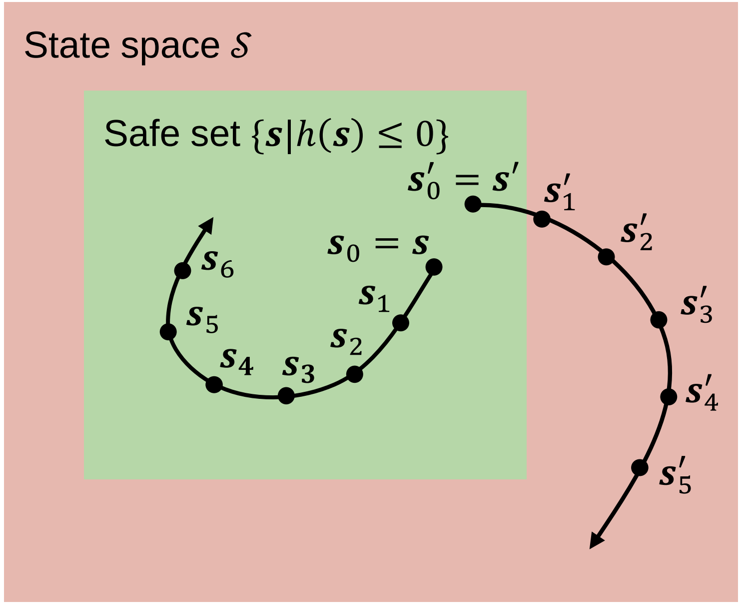


We propose to constrain the **worst-case violation** to characterize **persistently safe states**.

- Once the worst case is safe, the whole trajectory is safe

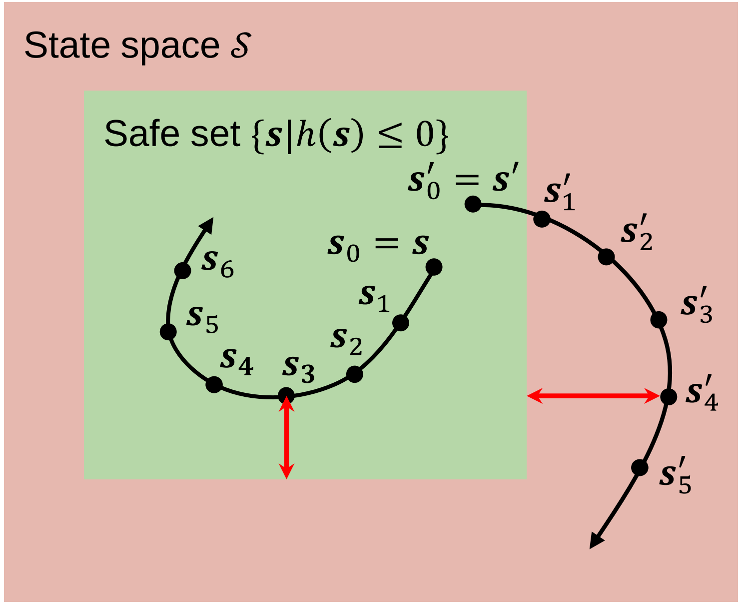
Safety value function

Definition: The worst-case state constraint violation $h(\boldsymbol{s})$ during a trajectory induced by policy π.



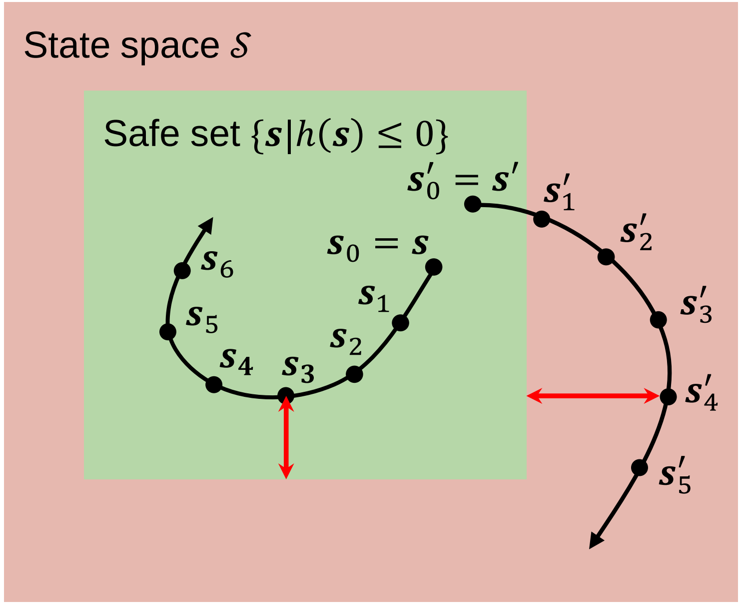
Safety value function

Definition: The worst-case state constraint violation $h(\boldsymbol{s})$ during a trajectory induced by policy π.



Safety value function

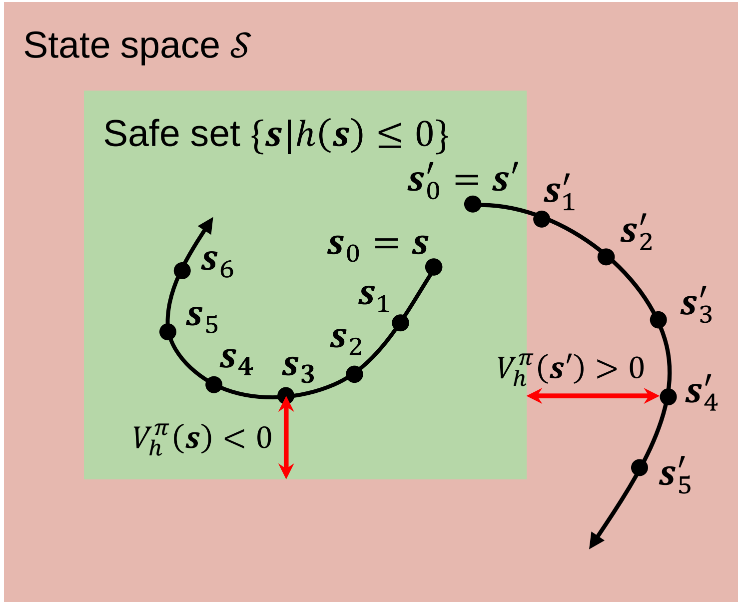
Definition: The worst-case state constraint violation $h(\boldsymbol{s})$ during a trajectory induced by policy π.



$$V_h^\pi(\boldsymbol{s}) \coloneqq \max_t h(\boldsymbol{s_t}) \,|\boldsymbol{s_0} = \boldsymbol{s}, \pi$$

Safety value function

Definition: The worst-case state constraint violation $h(\boldsymbol{s})$ during a trajectory induced by policy π.



$$V_h^\pi(\boldsymbol{s}) \coloneqq \max_t h(\boldsymbol{s_t})\,|\boldsymbol{s_0} = \boldsymbol{s}, \pi$$

Reachability constraint:

$$V_h^\pi(\boldsymbol{s}) \le 0$$

if `satisfied`: ← What we want

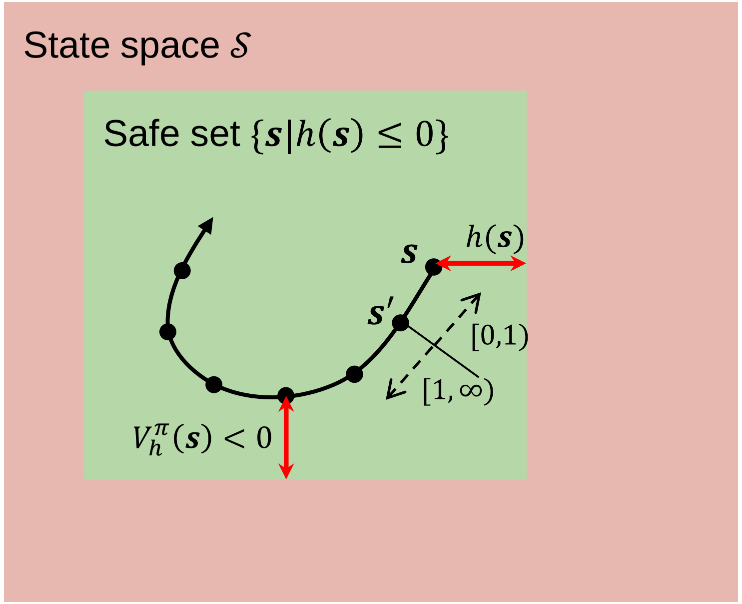
persistently safe

else if `violated`:

unsafe sooner or later

Safety value function - computation

We extend results in [Fisac et al., 2019] to a general **bootstrap form** of safety value function.



Self-consistency condition:

$$V_h^\pi(\boldsymbol{s}) = \max\{h(\boldsymbol{s}), V_h^\pi(\boldsymbol{s}')\}$$

J. Fisac, et al. Bridging Hamilton-Jacobi Safety Analysis and Reinforcement Learning. ICRA 2019

Reachability Constrained RL

Problem Formulation

$$\max_{\pi} J(\pi) = \mathbb{E}_{s\sim d_0}\left[V^{\pi}(s)\cdot \mathbb{1}_{s\in\mathcal{S}_f} - V_h^{\pi}(s)\cdot \mathbb{1}_{s\notin\mathcal{S}_f}\right]$$

$$\text{s.t. } V_h^{\pi}(s) \le 0 \ \forall \text{ possibly feasible initial state}$$

Constraints on each state

Lagrangian-based solution with multiplier network [Ma et al., 2021]

$$\max_{\lambda\ge 0}\min_{\theta} \mathbb{E}_{s\sim d_0}[V^{\pi}(s) + \lambda(s;\xi)V_h^{\pi}(s)]$$

$\lambda(s;\xi)$: mapping from state to multiplier

Difference with CMDP-based Constrained RL

$$C(\pi) = \mathbb{E}_{\tau\sim\pi}\left\{\sum_{t=0}^{\infty}\gamma^t c_t\right\}$$

Constraints on a trajectory

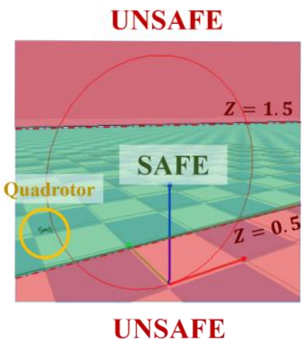
$$\max_{\lambda\ge 0}\min_{\theta} J(\pi_\theta) + \lambda C(\pi_\theta)$$

λ is a scalar

H. Ma, et, al. Feasible Actor-Critic: Constrained Reinforcement Learning for Ensuring Statewise Safety. arXiv:2105.10682

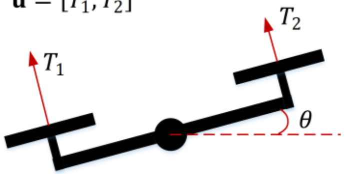
Experiments - safe-control-gym

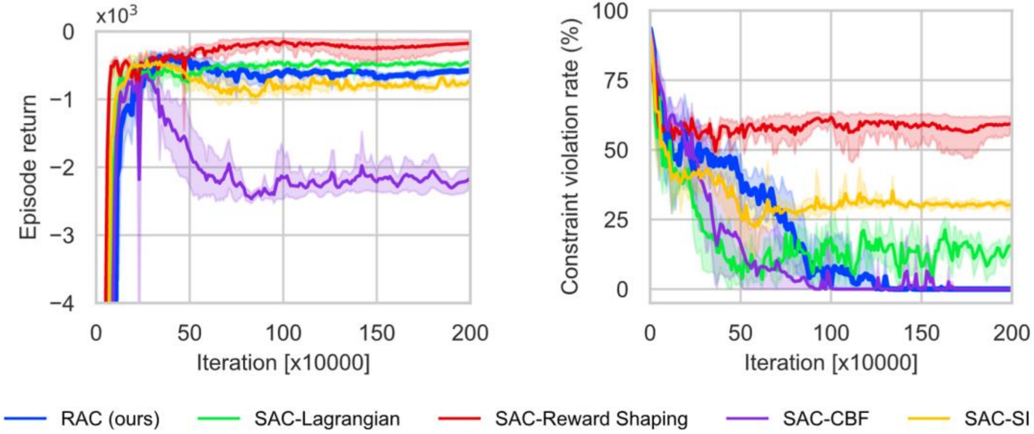
2D Quadrotors tracking while maintaining safe height



$$\mathbf{x} = [x, \dot{x}, z, \dot{z}, \theta, \dot{\theta}]^T$$

$$\mathbf{u} = [T_1, T_2]^T$$





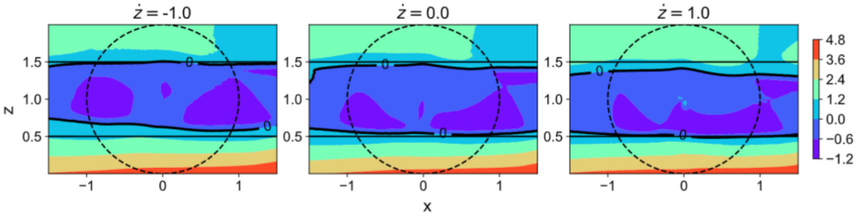
Baselines

- SAC-Lagrangian: CMDP-based — × Unsafe policy
- SAC-Reward shaping — × Unsafe policy
- SAC-CBF/SI — × CBF: safe but not moving / SI: unsafe policy

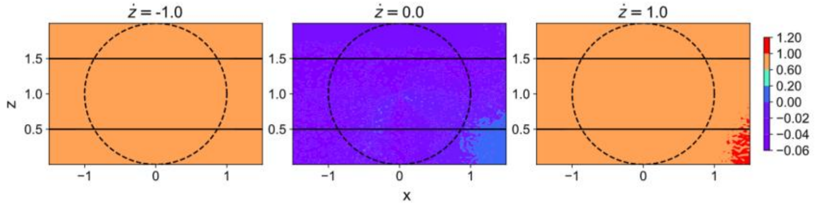
Experiments - safe-control-gym

Safety value function visualizations

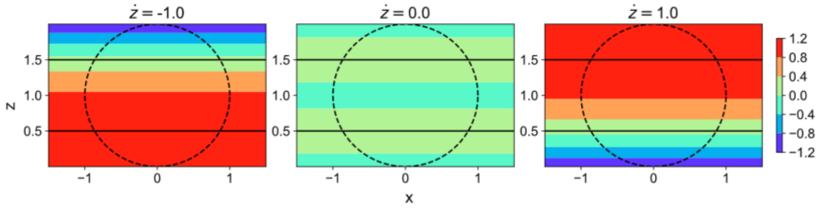
$\dot{z}$ - quadrotor vertical speed



RCRL



CBF



SI

RCRL has the largest safe sets

Conclusion

- We propose a novel reachability constraint to characterize the persistent safety of policies
- RCRL can converge to a zero-violation policy with competitive reward performance
  - Because the learned safe value can find those persistently safe states

- For more details, please see our paper: https://arxiv.org/abs/2205.07536
- Open-sourced implementation:

https://github.com/mahaitongdae/Reachability_Constrained_RL