

Introduction to Theory of Safe Decision Making

Dr. Akhil Anand

1st AID Scientific Workshop, Trondheim

Forewords & Disclaimer

Objectives:

- Put in place some common concepts & language
- Identify some key points in safe decision making
- Connect to AI

Disclaimer: we are a broad group who needs to get to know each others scientifically.

Apologies if I don't "hit" the right level for all.

I have favored simplicity over "absolute" rigor.



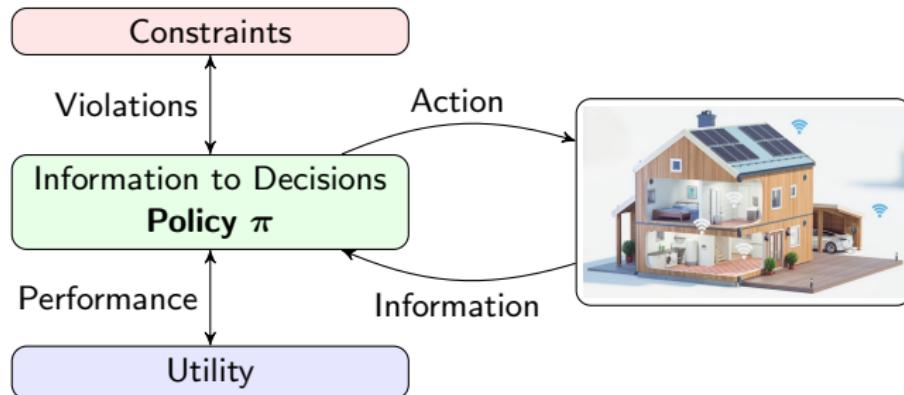
Outline

1 Some Basics of Safe Decision Making

2 Methods

3 Safe Decisions from Data & AI

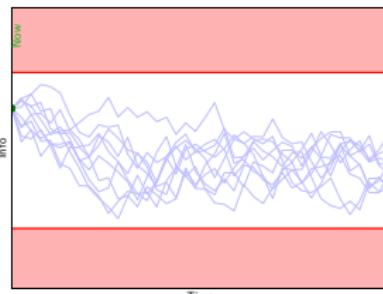
Formally Defining Safety?



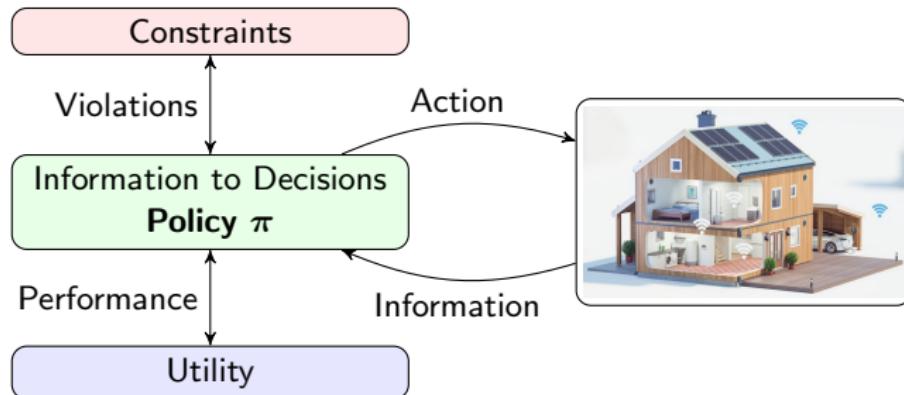
Safety in the real world



Safety in the math world



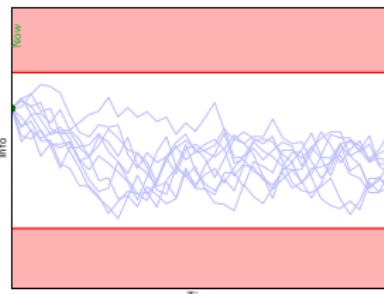
Formally Defining Safety?



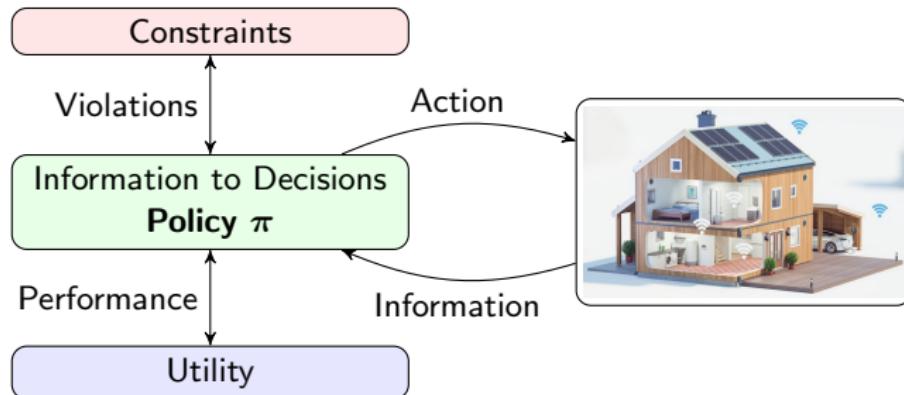
Safety in the real world



Safety in the math world



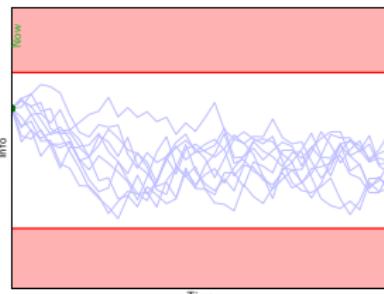
Formally Defining Safety?



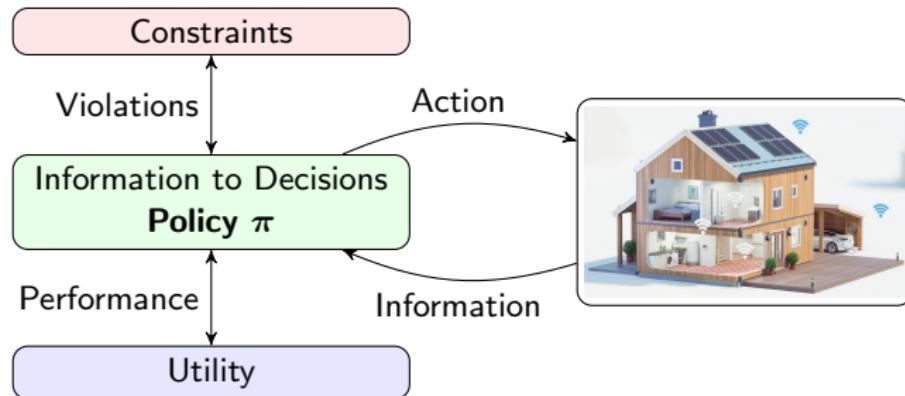
Safety in the real world



Safety in the math world



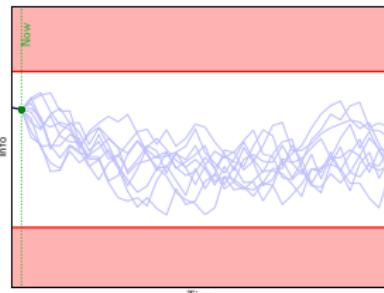
Formally Defining Safety?



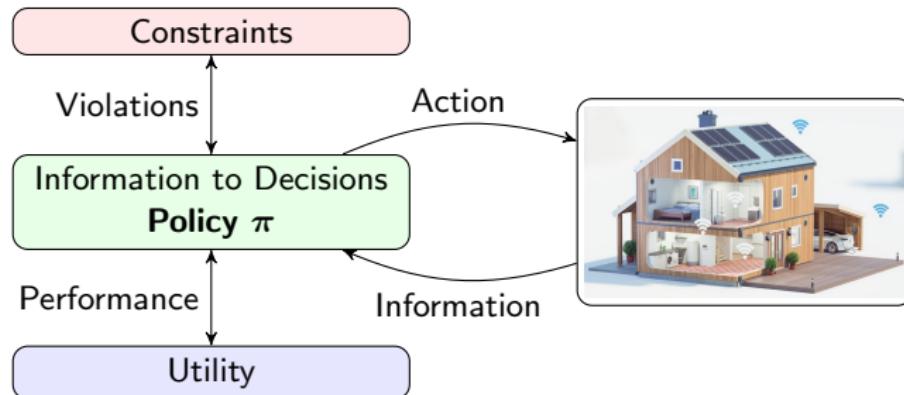
Safety in the real world



Safety in the math world



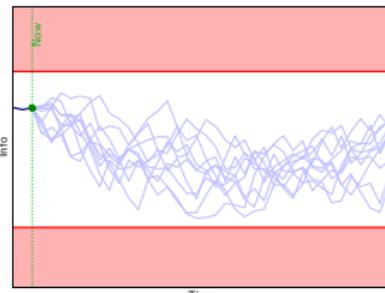
Formally Defining Safety?



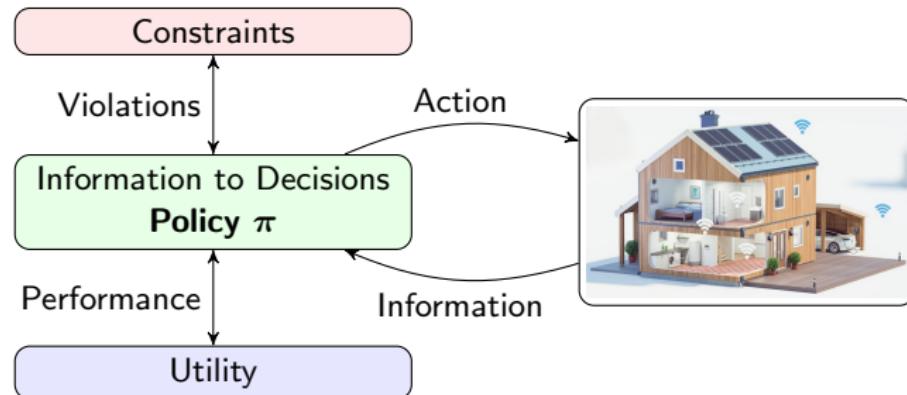
Safety in the real world



Safety in the math world



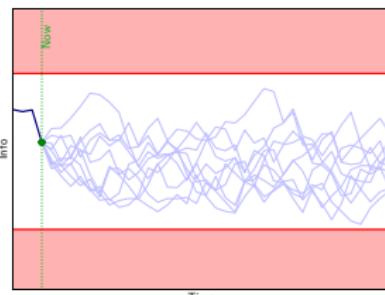
Formally Defining Safety?



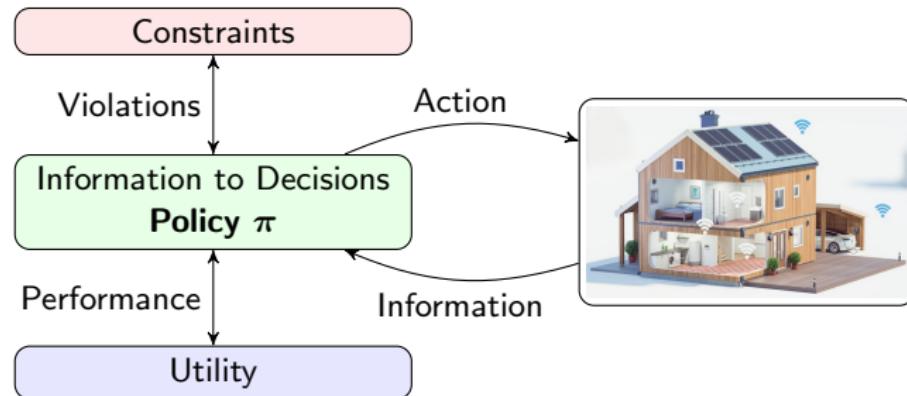
Safety in the real world



Safety in the math world



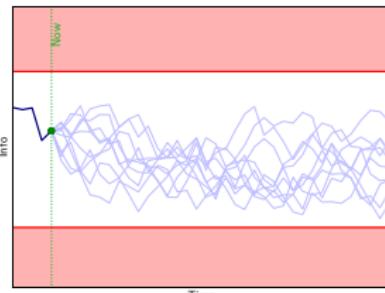
Formally Defining Safety?



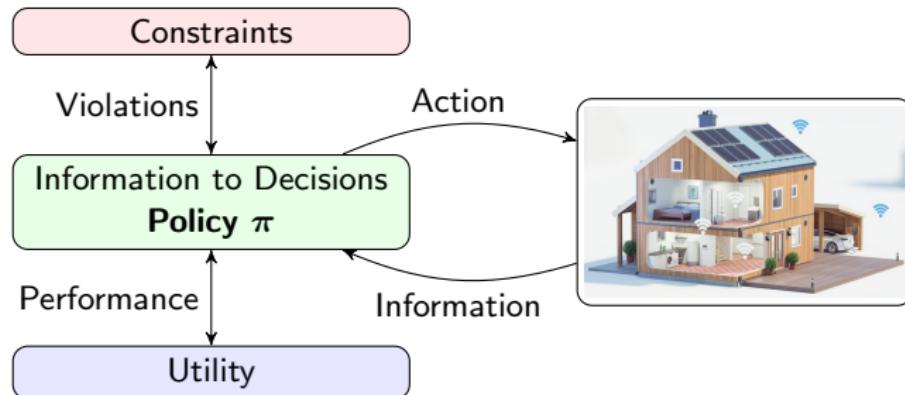
Safety in the real world



Safety in the math world



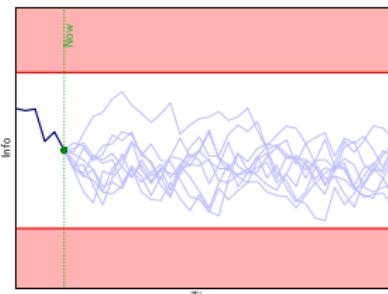
Formally Defining Safety?



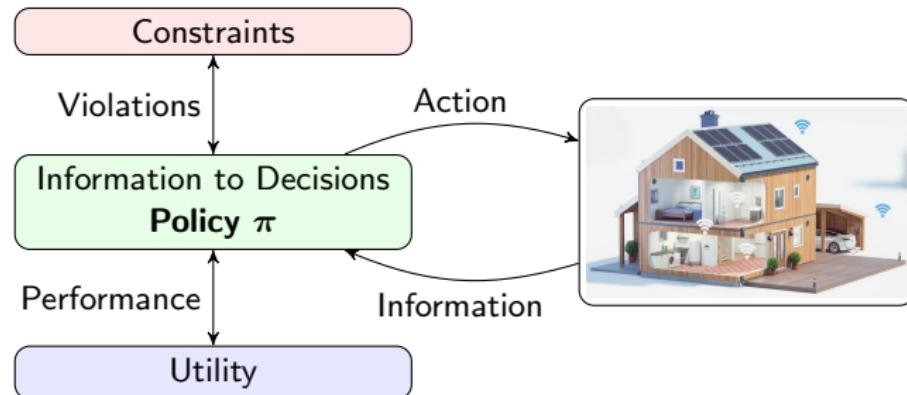
Safety in the real world



Safety in the math world



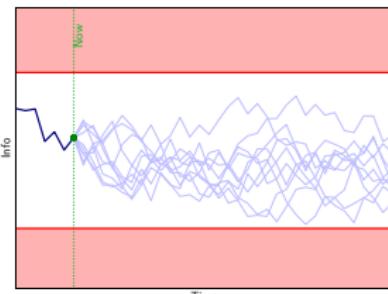
Formally Defining Safety?



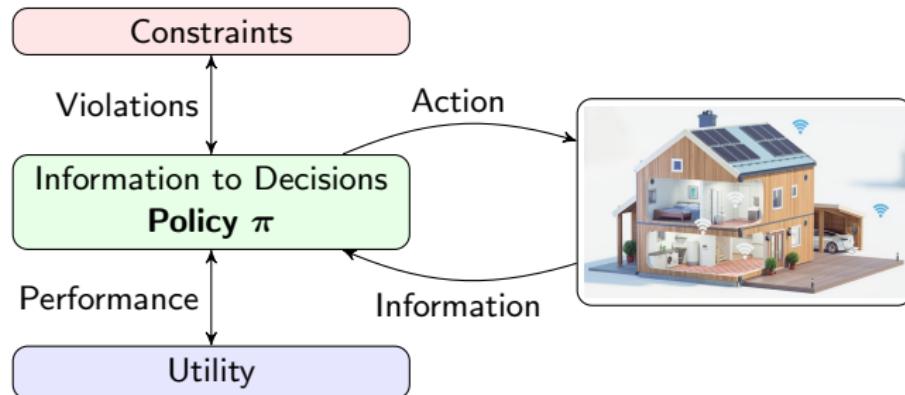
Safety in the real world



Safety in the math world



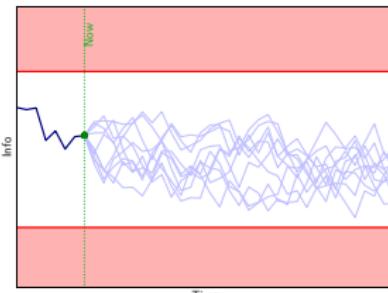
Formally Defining Safety?



Safety in the real world



Safety in the math world

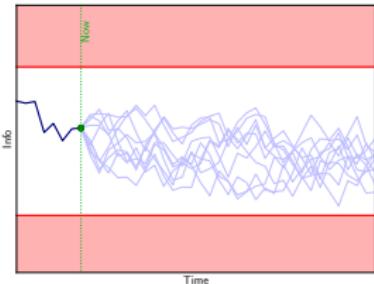


Constrained MDPs

In words

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

s.t. Constraints ok at all time

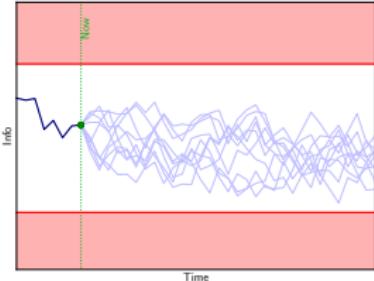


Constrained MDPs

In words

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

s.t. Constraints ok at all time



Formally

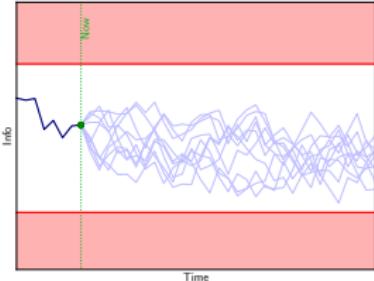
$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \left(\text{Utility} - \begin{cases} 0 & \text{if Constraint ok} \\ \infty & \text{if Constraint not ok} \end{cases} \right) \right]$$

Constrained MDPs

In words

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

s.t. Constraints ok at all time



Formally

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \left(\text{Utility} - \begin{cases} 0 & \text{if Constraint ok} \\ \infty & \text{if Constraint not ok} \end{cases} \right) \right]$$

Build policy using

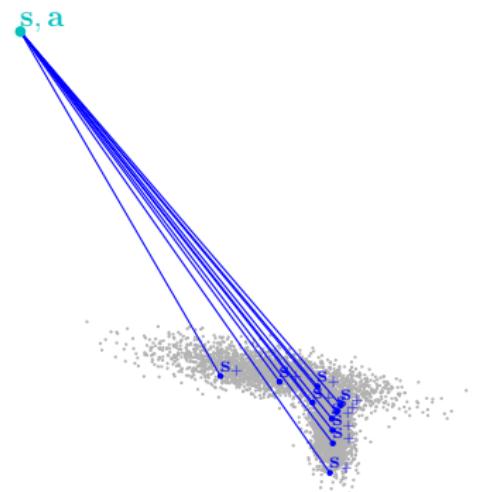
- Perfect model of the real world $\hat{P}[s_+|s, a] = P[s_+|s, a]$
- Model “pessimistic” about the uncertainties

... to evaluate “ $E[\cdot]$ ”

Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
 - “Container” (set) should be simple for computational reasons
 - Trajectories predicted by pessimistic model will “cover” the real world
 - Decision policy wants to be safe w.r.t. the “containers”

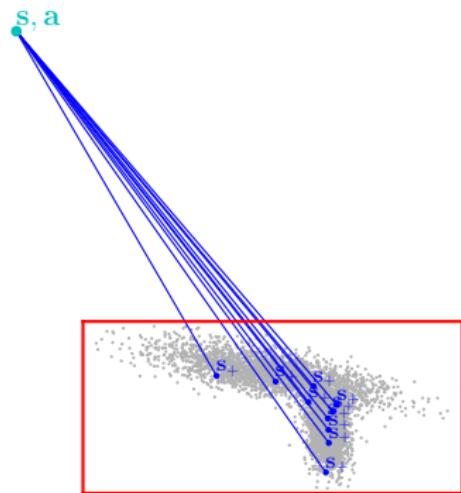
Distribution of one-step forward



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

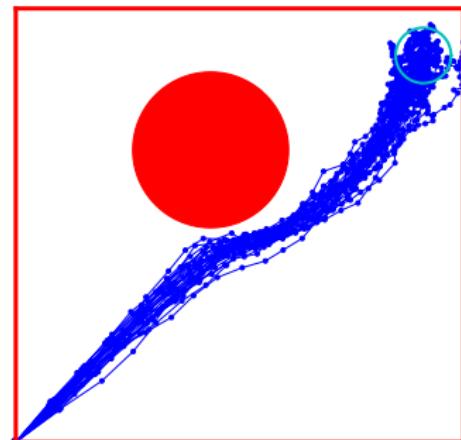
Distribution of one-step forward with “container”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

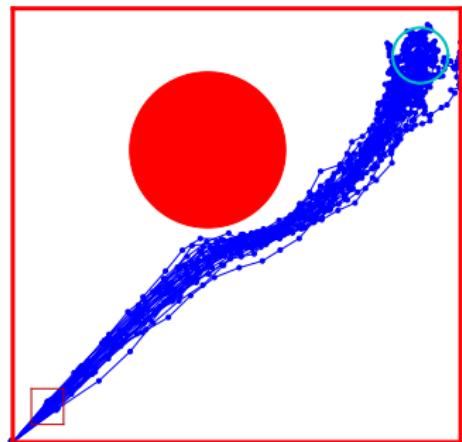
Trajectories



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

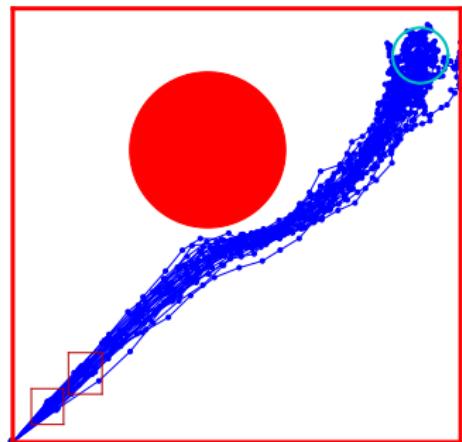
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

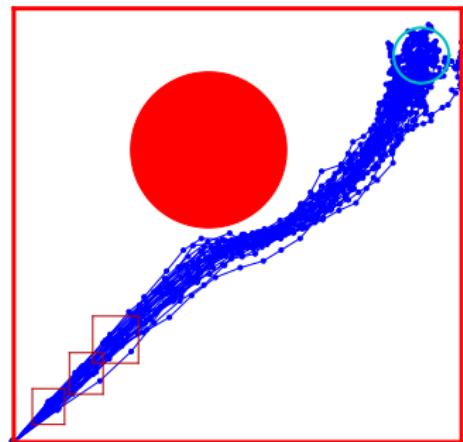
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

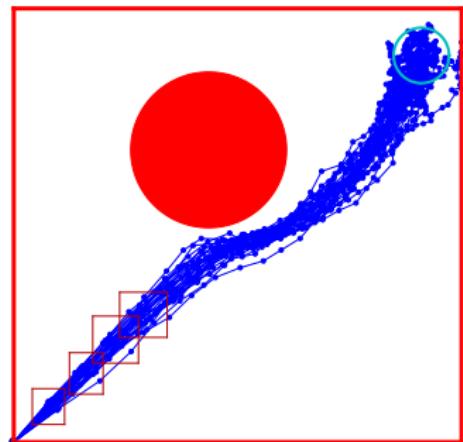
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

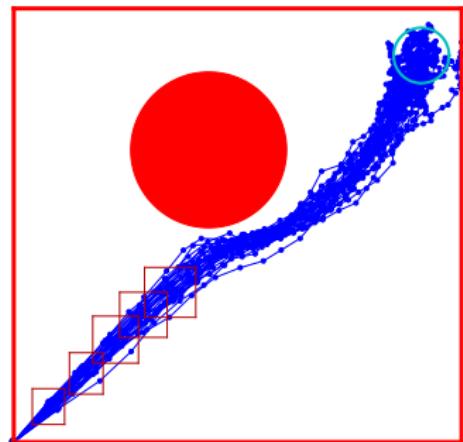
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

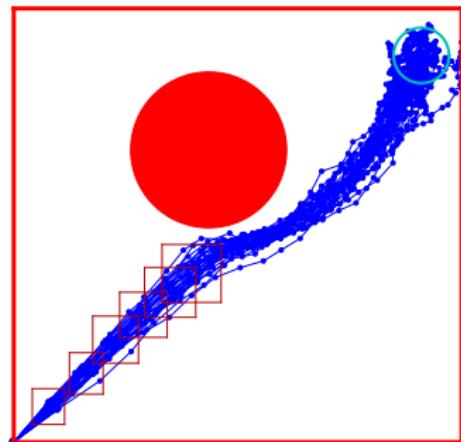
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

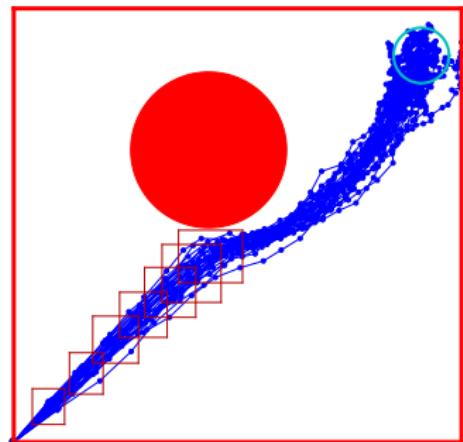
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

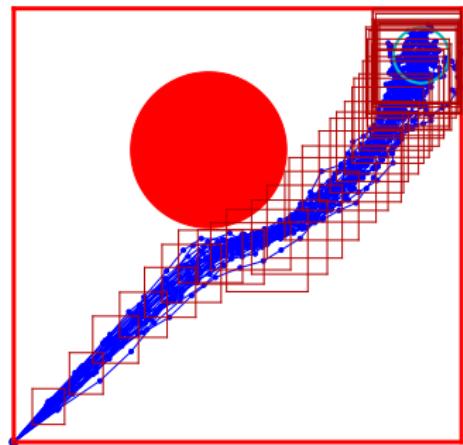
Trajectories with “containers”



Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

Trajectories with “containers”



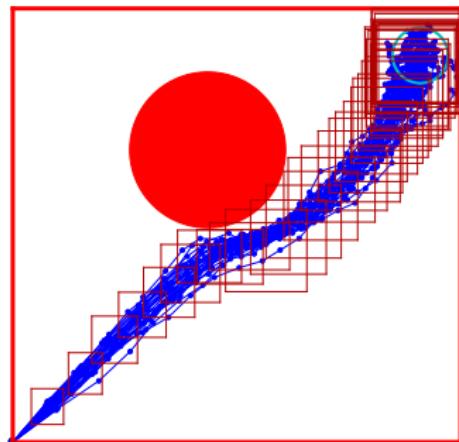
Pessimistic Models for Decision Making

- Model must “contain” the uncertainty
- “Container” (set) should be simple for computational reasons
- Trajectories predicted by pessimistic model will “cover” the real world
- Decision policy wants to be safe w.r.t. the “containers”

Remarks

- The propagation of the “containers” in the model predictions can be expensive / difficult
- Pessimistic propagations are usually needed → pessimistic over pessimistic
- Policy based on worst-case perspective makes the decisions highly conservative
- Often labelled “Robust” decision making

Trajectories with “containers”

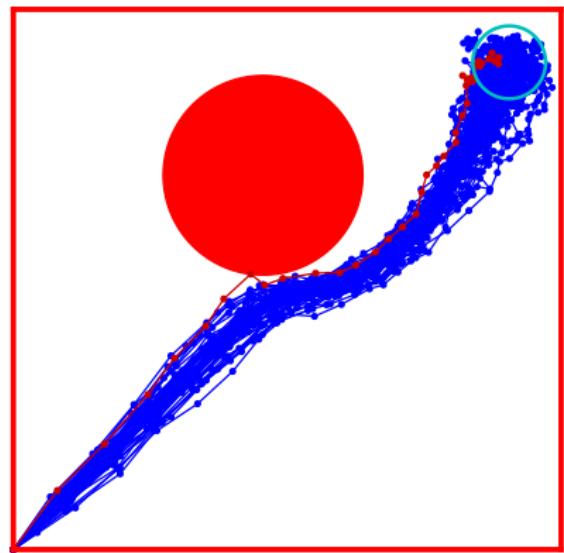


MDPs with probabilistic safety

In words

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

s.t. Probability of no violation $\geq c$



MDPs with probabilistic safety

In words

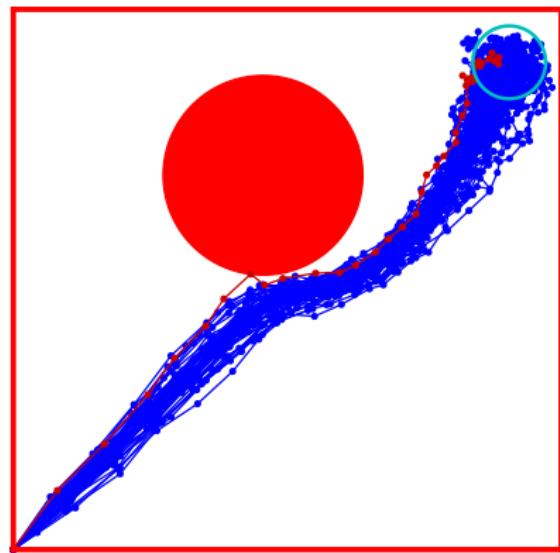
$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

s.t. Probability of no violation $\geq c$

Formally

$$\pi^* = \arg \max_{\pi} E \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \right]$$

s.t. $P[s_0, \dots, \infty \in \mathbb{S}] \geq c$



MDPs with probabilistic safety

In words

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

s.t. Probability of no violation $\geq c$

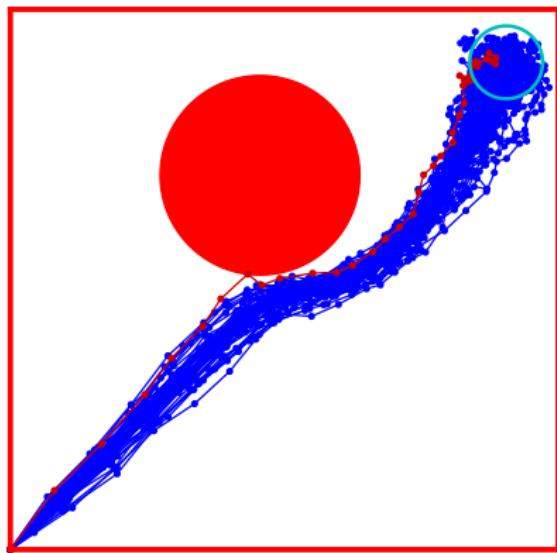
Formally

$$\pi^* = \arg \max_{\pi} E \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \right]$$

s.t. $P[s_0, \dots, \infty \in \mathbb{S}] \geq c$

Remarks:

- If we can tolerate $c > 0$ (small) it can make a big gain in performance
- Aligned with industrial / practical standards on “large series”
- Problem needs a “termination” (time or goal reached)
- Building decisions can be difficult from a computational point of view



Outline

1 Some Basics of Safe Decision Making

2 Methods

3 Safe Decisions from Data & AI

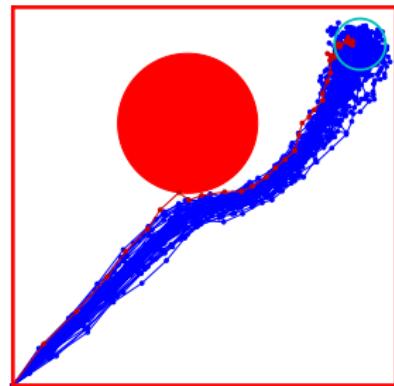
Robust Repeated Planning

- Introduction: Robust MPC, Scenario Trees, MC, robust multi-stage stochastic programming, MPPI?
- Difficulties: guarantees for nonlinear systems, persistent safety (recursive feasibility)



$$\pi^* = \arg \max_{\pi} E \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \right]$$

s.t. $P[s_0, \dots, \infty \in S] \geq c$



Expected value form Illustrate??

$$\pi^* = \arg \max_{\pi} E \left[\sum_{k=0}^{\infty} \gamma^k L(s_k, a_k) \right]$$

s.t. $E \left[\begin{array}{ll} 1 & \text{if no violation occurred} \\ 0 & \text{otherwise} \end{array} \right] \geq c$

- Expected value form enables classical techniques (ref. 1st lecture), i.e. DP and RL
- Difficulties: estimate expected values (sample based) when $P[s_0, \dots, \infty \in S]$ is close to 1. **Illustrate??**

Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$

Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

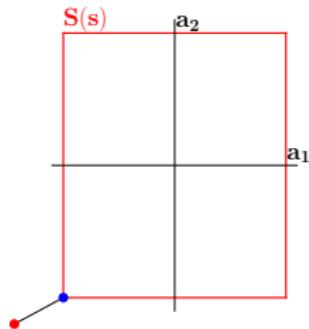
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



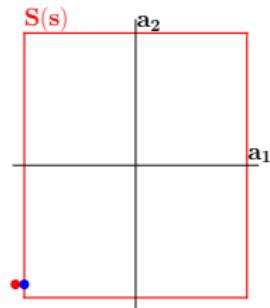
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



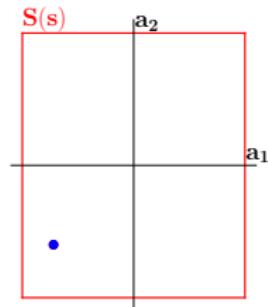
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



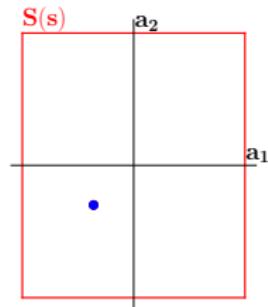
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



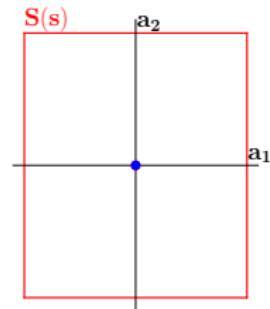
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



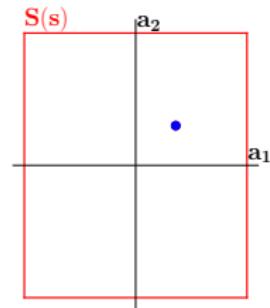
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



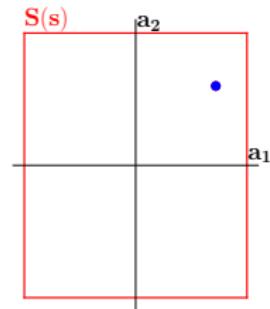
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



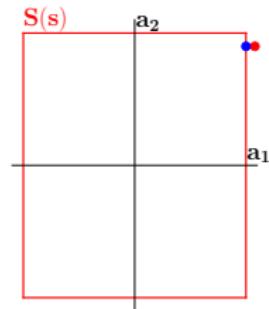
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



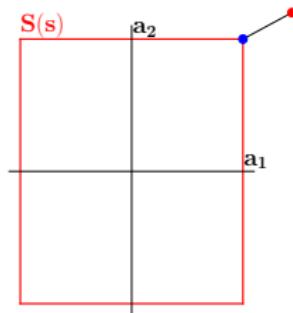
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



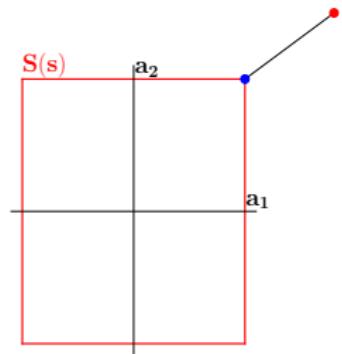
Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$



Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

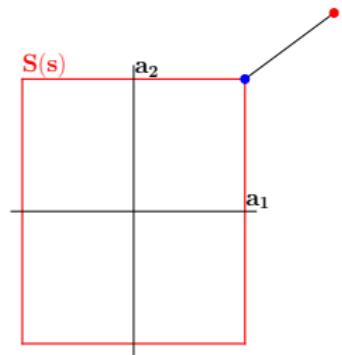
that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$

Remark: “oracle” can be very hard to build, or not...



Not losing the king



Safety Filters & Reinforcement Learning

- For any state s , an “oracle” tells you which actions will not jeopardize safety in the long run → “Safe set” $S(s)$
- We can learn an “unsafe” policy

$$\pi^* = \arg \max_{\pi} E \left[\sum_{\text{time}} \text{Utility} \right]$$

that focuses only on utility

- For state s , take the decisions using: Action = Projection $_{S(s)} \pi^*(s)$

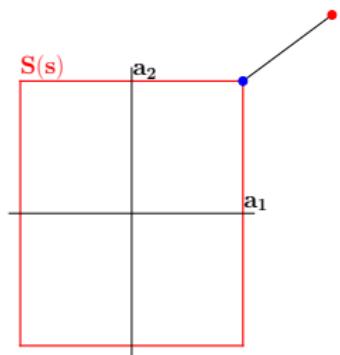
Remark: “oracle” can be very hard to build, or not...



Not losing the king



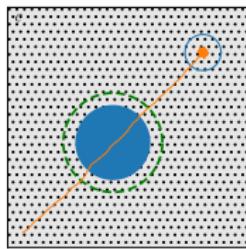
Able to stop within the visible distance



Control Barrier Functions (CBFs)

In words

- CBFs provide a formal, model-based way to build the “oracle” safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it to be safe.

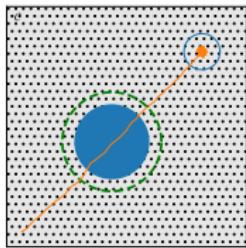


Unsafe trajectory: violates $h(x) \geq 0$.

Control Barrier Functions (CBFs)

In words

- CBFs provide a formal, model-based way to build the “oracle” safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it to be safe.



Unsafe trajectory: violates $h(x) \geq 0$.

Formally

$$\dot{x} = f(x) + g(x)u, \quad \mathcal{C} = \{x : h(x) \geq 0\}$$

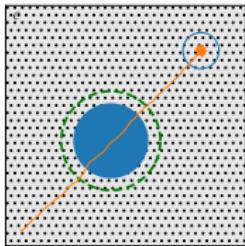
$$u^*(x) = \arg \min_u \frac{1}{2} \|u - u_{\text{des}}(x)\|_2^2 \quad \text{s.t.}$$

$$\underbrace{L_f h(x) + L_g h(x) u + \alpha(h(x))}_{\text{CBF constraint}} \geq 0$$

Control Barrier Functions (CBFs)

In words

- CBFs provide a formal, model-based way to build the “oracle” safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it to be safe.



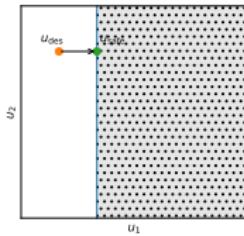
Unsafe trajectory: violates $h(x) \geq 0$.

Formally

$$\dot{x} = f(x) + g(x)u, \quad \mathcal{C} = \{x : h(x) \geq 0\}$$

$$u^*(x) = \arg \min_u \frac{1}{2} \|u - u_{\text{des}}(x)\|_2^2 \quad \text{s.t.}$$

$$\underbrace{L_f h(x) + L_g h(x) u + \alpha(h(x))}_{\text{CBF constraint}} \geq 0$$

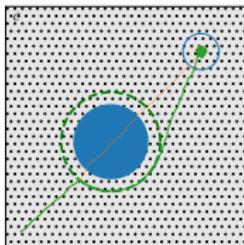


CBF filter: project u_{des} to feasible u_{safe} .

Control Barrier Functions (CBFs)

In words

- CBFs provide a formal, model-based way to build the “oracle” safe set.
- Use a barrier function $h(x)$ so that staying safe means $h(x) \geq 0$.
- At each step, if the proposed action would leave the safe set, a small QP minimally adjusts it to be safe.



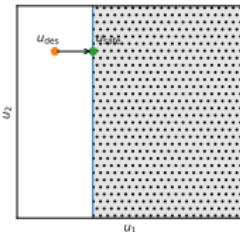
CBF-filtered trajectory.

Formally

$$\dot{x} = f(x) + g(x)u, \quad \mathcal{C} = \{x : h(x) \geq 0\}$$

$$u^*(x) = \arg \min_u \frac{1}{2} \|u - u_{\text{des}}(x)\|_2^2 \quad \text{s.t.}$$

$$\underbrace{L_f h(x) + L_g h(x) u + \alpha(h(x))}_{\text{CBF constraint}} \geq 0$$



CBF filter: project u_{des} to feasible u_{safe} .

- The barrier function and safe set \mathcal{C} are typically **constructed from domain knowledge** (physics, rules, safety envelopes).
- Requires a (possibly simplified, conservative) **system model: robust approach**.

Outline

1 Some Basics of Safe Decision Making

2 Methods

3 Safe Decisions from Data & AI

Reinforcement Learning

Akhil?

The Role of Probabilistic AI