# 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

# Formally Defining Safety?

Constraints

Violations

Information to Decisions
**Policy $\pi$**

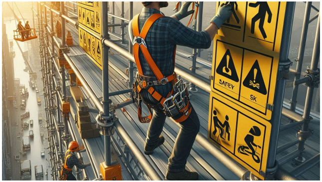
Action

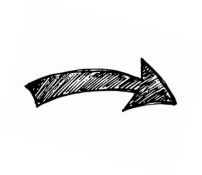
Information

Performance

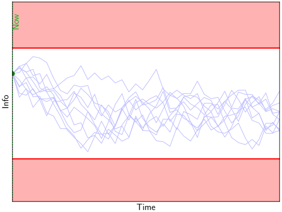
Utility

**Safety in the real world**

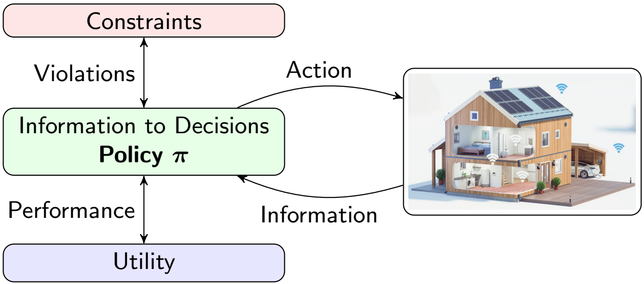




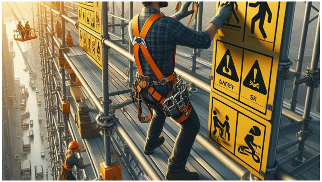
**Safety in the math world**

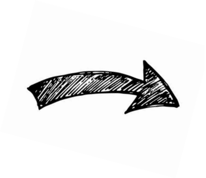


# Formally Defining Safety?

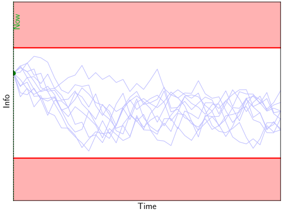



**Safety in the real world**





**Safety in the math world**



# Formally Defining Safety?

Constraints

Violations

Information to Decisions
**Policy $\pi$**

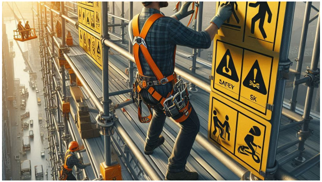
Performance

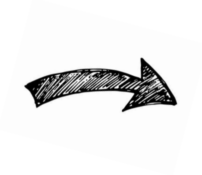
Utility

Action

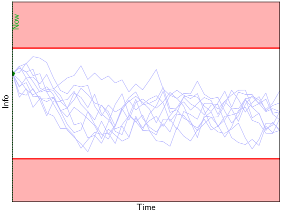
Information

**Safety in the real world**





**Safety in the math world**



# Formally Defining Safety?

Constraints

Violations

Information to Decisions
**Policy $\pi$**

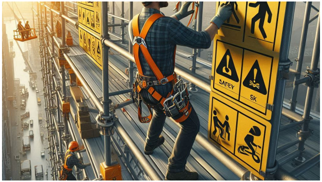
Action

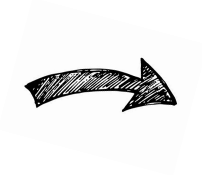
Information

Performance

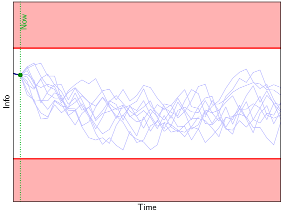
Utility

**Safety in the real world**





**Safety in the math world**



# Formally Defining Safety?

Constraints

Violations

Action

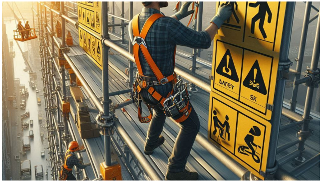
Information to Decisions
**Policy $\pi$**

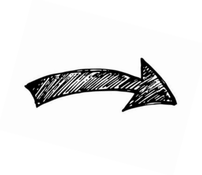
Performance

Information

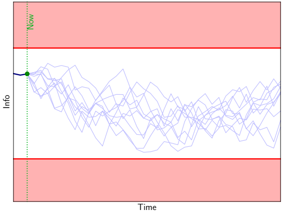
Utility

**Safety in the real world**





**Safety in the math world**



# Formally Defining Safety?

Constraints

Violations

Information to Decisions
**Policy $\pi$**

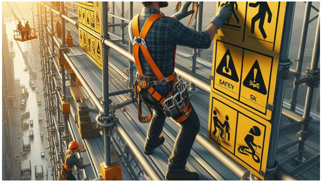
Performance

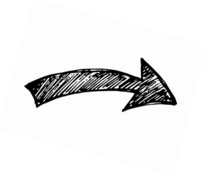
Utility

Action

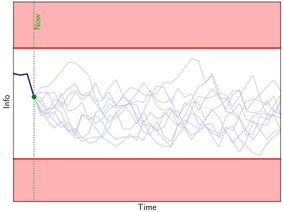
Information

**Safety in the real world**

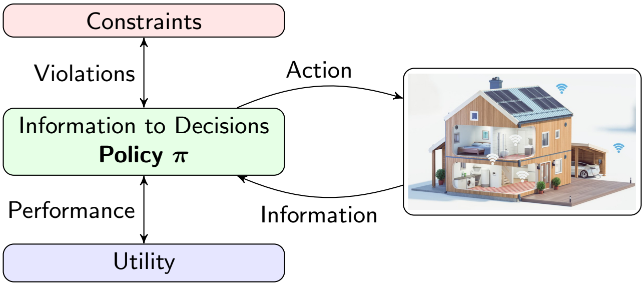


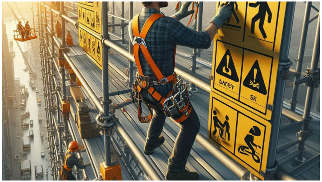
**Safety in the math world**



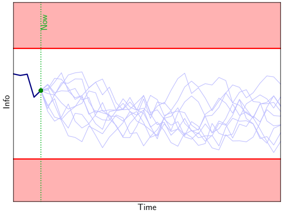
# Formally Defining Safety?




**Safety in the real world**



**Safety in the math world**



# Formally Defining Safety?

Constraints

Violations

Information to Decisions
**Policy $\pi$**

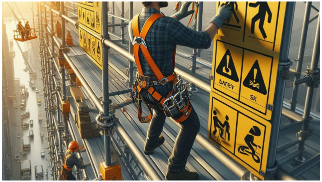
Performance

Utility

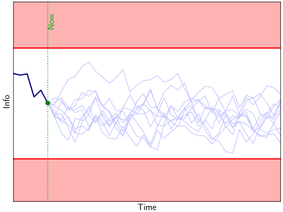
Action

Information

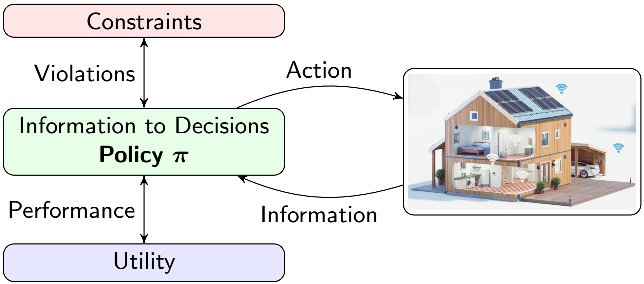
**Safety in the real world**



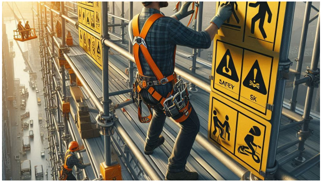
**Safety in the math world**



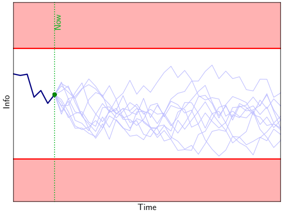
# Formally Defining Safety?



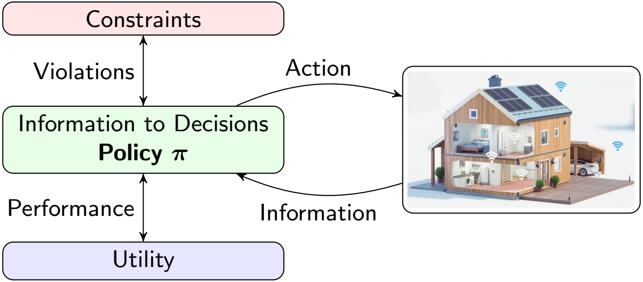

**Safety in the real world**



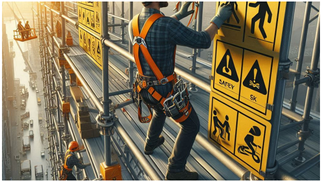
**Safety in the math world**



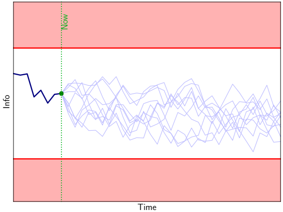
# Formally Defining Safety?




**Safety in the real world**



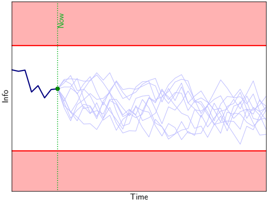
**Safety in the math world**



# Constrained MDPs

**In words**

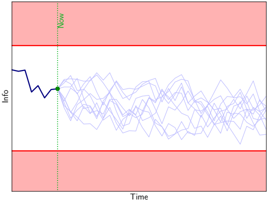
$$\boldsymbol{\pi}^\star = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Constraints ok at all time}$$



# Constrained MDPs

**In words**

$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Constraints ok at all time}$$



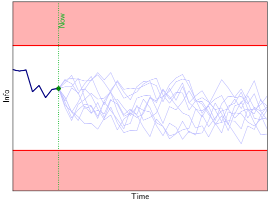

**Formally**

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

# Constrained MDPs

**In words**

$$\boldsymbol{\pi}^\star = \operatorname*{arg\,max}_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Constraints ok at all time}$$




**Formally**

$$\boldsymbol{\pi}^\star = \operatorname*{arg\,max}_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \left(\text{Utility} - \begin{cases} 0 & \text{if} \quad \text{Constraint ok} \\ \infty & \text{if} \quad \text{Constraint not ok} \end{cases}\right)\right]$$

**Build policy** using

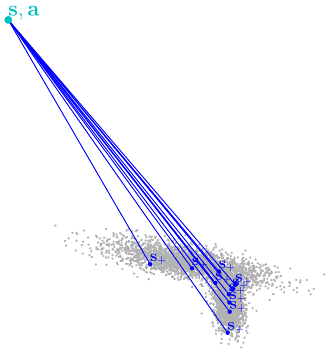
- Perfect model of the real world $\hat{\mathrm{P}}[\mathbf{s}_+|\mathbf{s},\mathbf{a}] = \mathrm{P}[\mathbf{s}_+|\mathbf{s},\mathbf{a}]$
- Model "pessimistic" about the uncertainties

... to evaluate "$\mathrm{E}[\cdot]$"

# 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"

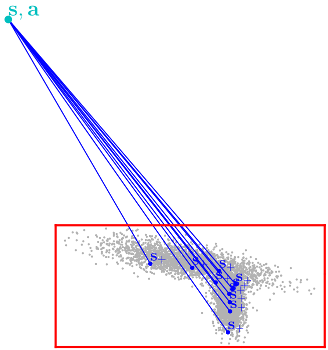
Distribution of one-step forward



# 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"

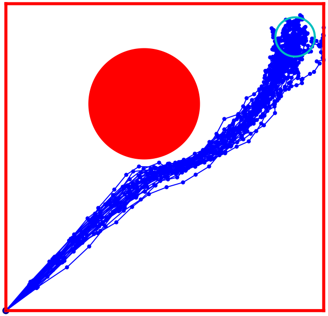
Distribution of one-step forward with "container"



# 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"

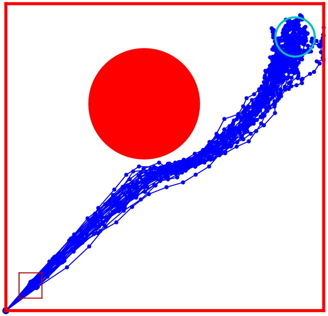
Trajectories



# 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"

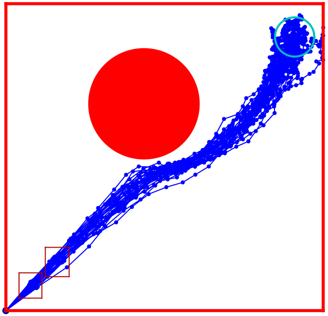
Trajectories with "containers"



# 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"

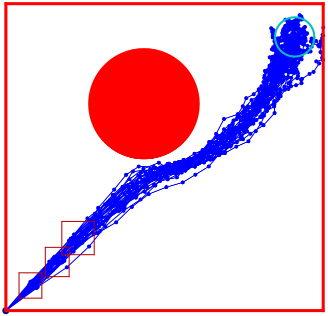
Trajectories with "containers"



# 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"

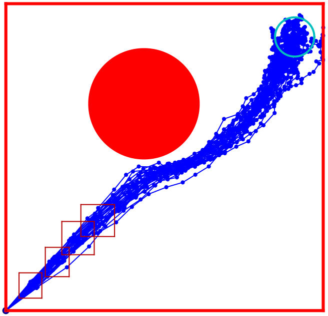
Trajectories with "containers"



# 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"

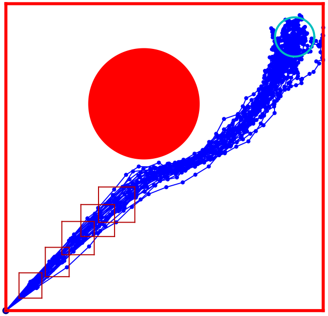
Trajectories with "containers"



## 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"

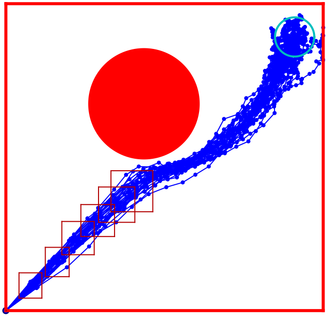
Trajectories with "containers"



# 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"

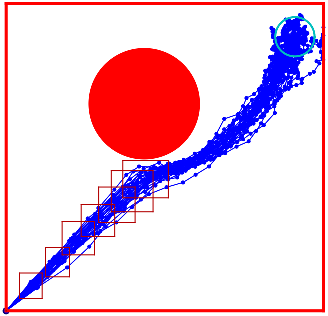
Trajectories with "containers"



# 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"

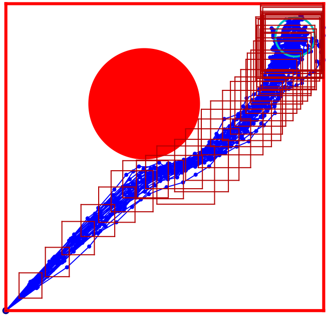
Trajectories with "containers"



# 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"



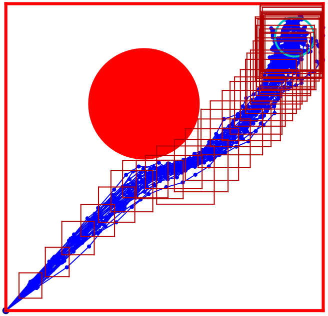
# Pessimistic Models for Decision Making

- 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 $\rightarrow$ pessimistic over pessimistic
- Policy based on worst-case perspective makes the decisions highly conservative
- Often labelled "Robust" decision making

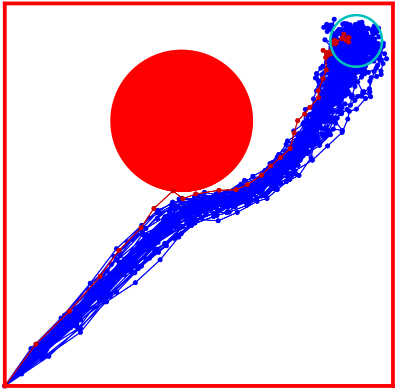
Trajectories with "containers"



# MDPs with probabilistic safety

**In words**

$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Probability of no violation} \geq c$$



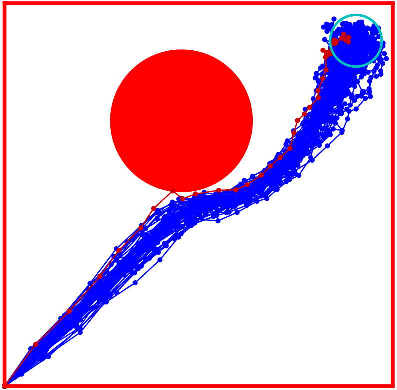
# MDPs with probabilistic safety

**In words**

$$\pi^{\star} = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Probability of no violation} \geq c$$

**Formally**

$$\pi^{\star} = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^{k} L\left(\mathbf{s}_k, \mathbf{a}_k\right)\right]$$
$$\text{s.t.} \quad \mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right] \geq c$$



# MDPs with probabilistic safety

**In words**

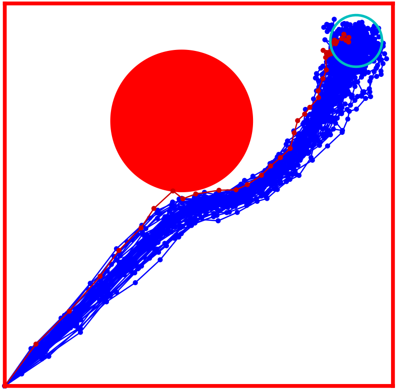
$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$
$$\text{s.t.} \quad \text{Probability of no violation} \geq c$$

**Formally**

$$\pi^\star = \arg\max_{\pi} \quad \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right)\right]$$
$$\text{s.t.} \quad \mathrm{P}\left[\mathbf{s}_{0,\dots\infty} \in \mathbb{S}\right] \geq c$$

**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

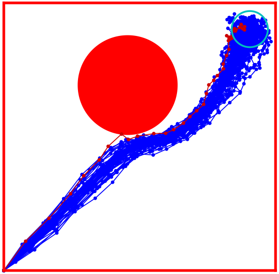
- Introduction: Robust MPC, Scenario Trees, MC, robust multi-stage stochastic programing, MPPI?
- Difficulties: guarantees for nonlinear systems, persistent safety (recursive feasibility)

# Dynamic Programming & RL with Probabilistic Safety

$$
\begin{aligned}
\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad & \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right)\right] \\
\text{s.t.} \quad & \mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right] \geq c
\end{aligned}
$$

**Expected value form** Illustrate??

$$
\begin{aligned}
\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad & \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\mathbf{s}_k, \mathbf{a}_k\right)\right] \\
\text{s.t.} \quad & \mathrm{E}\left[\begin{array}{cc} 1 & \text{if no violation occurred} \\ 0 & \text{otherwise} \end{array}\right] \geq c
\end{aligned}
$$



- Expected value form enables classical techniques (ref. 1st lecture), i.e. DP and RL
- Difficulties: estimate expected values (sample based) when $\mathrm{P}\left[\mathbf{s}_{0,\ldots\infty} \in \mathbb{S}\right]$ is close to 1. Illustrate??

## Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$ "Safe set" $\mathrm{S}(\mathrm{s})$

## Safety Filters & Reinforcement Learning

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

$$\boldsymbol{\pi}^{\star} = \underset{\boldsymbol{\pi}}{\arg\max} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

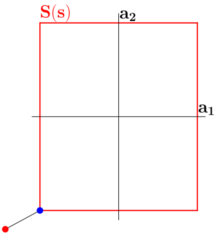
# Safety Filters & Reinforcement Learning

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

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



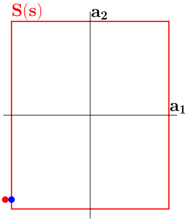
## Safety Filters & Reinforcement Learning

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

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



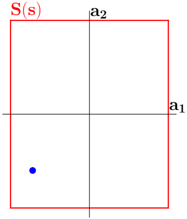
## Safety Filters & Reinforcement Learning

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

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathbb{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



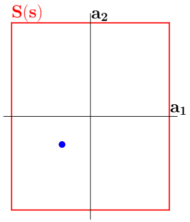
## Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



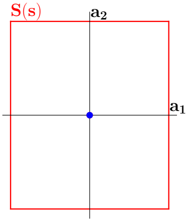
## Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



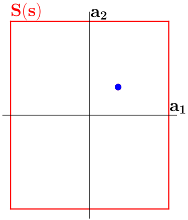
## Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



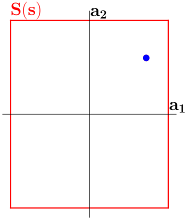
## Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}\left(\mathrm{s}\right)$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

- For state $\mathrm{s}$, take the decisions using: $\text{Action} = \text{Projection}_{\mathrm{S}(\mathrm{s})}\boldsymbol{\pi}^{\star}\left(\mathrm{s}\right)$



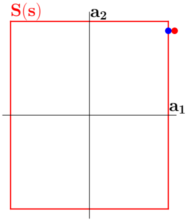
## Safety Filters & Reinforcement Learning

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

$$\boldsymbol{\pi}^{\star} = \underset{\boldsymbol{\pi}}{\text{arg max}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



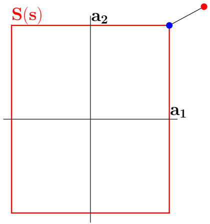
# Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run$\rightarrow$"Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



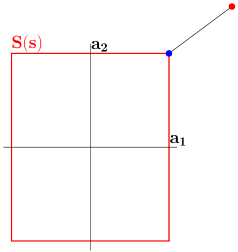
## Safety Filters & Reinforcement Learning

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

$$\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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



# Safety Filters & Reinforcement Learning

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

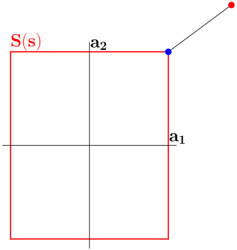
$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

that focuses only on utility

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

**Remark**: "oracle" can be very hard to build, or not...

Not losing the king



# Safety Filters & Reinforcement Learning

- For any state $\mathrm{s}$, an "oracle" tells you which actions will not jeopardize safety in the long run $\rightarrow$ "Safe set" $\mathrm{S}(\mathrm{s})$
- We can learn an "unsafe" policy

$$\boldsymbol{\pi}^\star = \arg\max_{\boldsymbol{\pi}} \quad \mathrm{E}\left[\sum_{\text{time}} \text{Utility}\right]$$

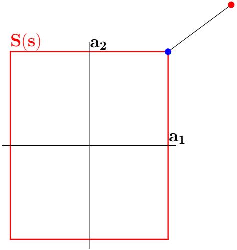
that focuses only on utility

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

**Remark**: "oracle" can be very hard to build, or not...

Not losing the king

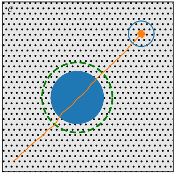
Able to stop within the visible distance



# Control Barrier Functions (CBFs)

**In words**

- Define a safe set $\mathcal{C} = \{x : h(x) \geq 0\}$ using a barrier function $h$.
- Keep the state inside $\mathcal{C}$ by enforcing a simple inequality at each step.
- If a suggested action is unsafe, a small QP adjusts it to the nearest safe action.



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

# Control Barrier Functions (CBFs)

**In words**

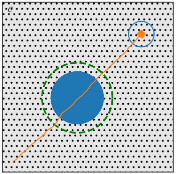
- Define a safe set $\mathcal{C} = \{x : h(x) \geq 0\}$ using a barrier function $h$.
- Keep the state inside $\mathcal{C}$ by enforcing a simple inequality at each step.
- If a suggested action is unsafe, a small QP adjusts it to the nearest safe action.

**Formally**

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

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

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



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

# Control Barrier Functions (CBFs)

**In words**

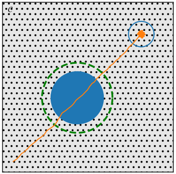
- Define a safe set $\mathcal{C} = \{x : h(x) \geq 0\}$ using a barrier function $h$.
- Keep the state inside $\mathcal{C}$ by enforcing a simple inequality at each step.
- If a suggested action is unsafe, a small QP adjusts it to the nearest safe action.

**Formally**

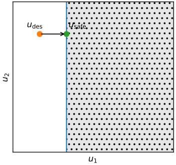
$$\dot{x} = f(x) + g(x)u, \qquad \mathcal{C} = \{x : h(x) \geq 0\}$$

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

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



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




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

# Control Barrier Functions (CBFs)

**In words**

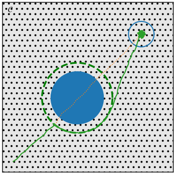
- Define a safe set $\mathcal{C} = \{x : h(x) \geq 0\}$ using a barrier function $h$.
- Keep the state inside $\mathcal{C}$ by enforcing a simple inequality at each step.
- If a suggested action is unsafe, a small QP adjusts it to the nearest safe action.

**Formally**

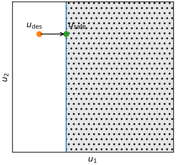
$$\dot{x} = f(x) + g(x)u, \qquad \mathcal{C} = \{x : h(x) \geq 0\}$$

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

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



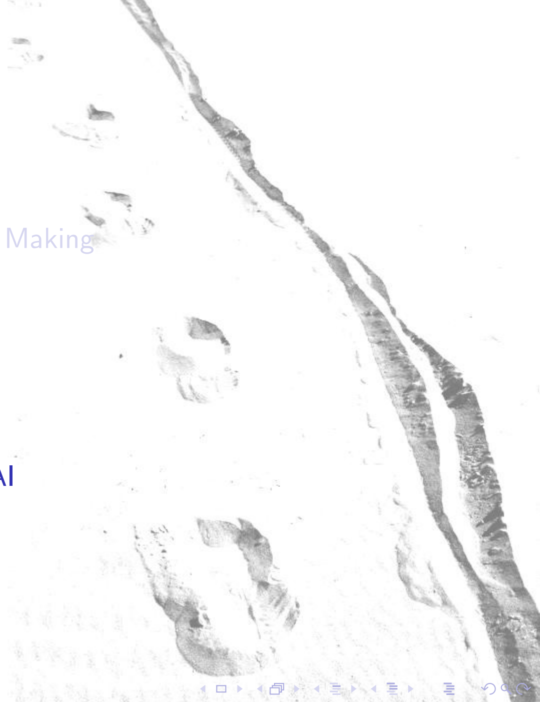
CBF-filtered trajectory.




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

- 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



## Reinforcement Learning

Akhil?

# The Role of Probabilistic AI