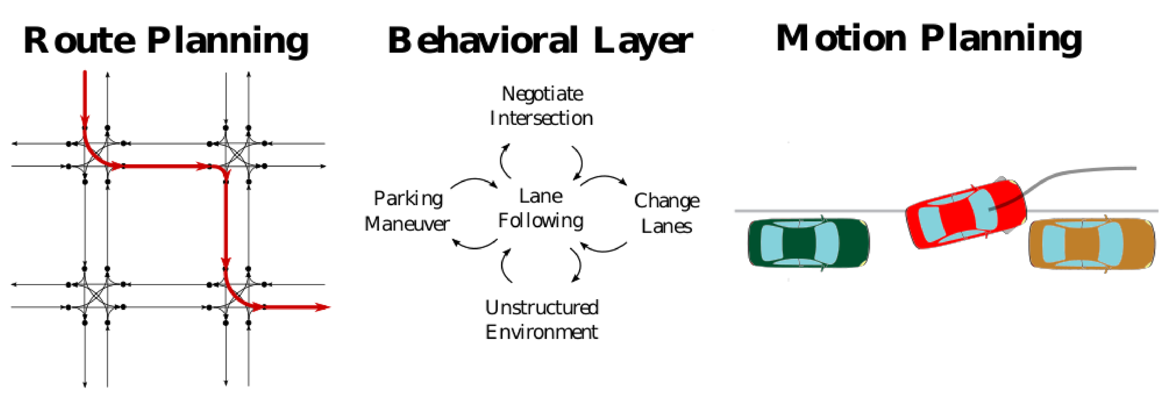
Approximate Robust Control of Uncertain Dynamical Systems

Edouard Leurent, Yann Blanco, Denis Efimov, Odalric-Ambrym Maillard

Motivation



Decision-making pipeline in Autonomous Driving

Progress has been made (Paden et al. 2016)

- route planning is solved
- reliable techniques exist for motion planning and control

Current limitations

- Implementations rely on hand-crafted rules such as FSM
  ↳tailored for specific cases, won't scale to complex scenes
- Social interactions are difficult to model explicitly:
  ↳poor negotiation abilities, "freezing robot" problem

We frame the problem in a sequential learning setting

Reinforcement learning

Optimal control under unknown dynamics $T(s_{t+1}|s_t, a_t)$

$$\max_{\pi} \underbrace{\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, a_t \sim \pi(s_t), s_{t+1} \sim T(s_t, a_t)\right]}_{\text{policy return } R_\pi^T} \tag{1}$$

Model-free methods directly optimize $\pi(a_t|s_t)$ through policy evaluation and policy improvement.

Model-based methods

1. Learn a model for the dynamics $\widehat{T}(s_t, a_t)$
2. (Planning) Leverage it to compute

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, a_t \sim \pi(s_t), s_{t+1} \sim \widehat{T}(s_t, a_t)\right]$$

- Better sample efficiency, interpretability
- Model bias: $\widehat{T} \neq T$

Robust optimization

1. Build a **confidence region** $\mathbf{T}$ around $\widehat{T}$

$$\forall T' \in \mathbf{T}, \quad \mathbb{P}(||\widehat{T} - T'|| > \varepsilon) < \delta$$

2. Plan **robustly** with respect to this ambiguity

$$\max_{\pi} \underbrace{\min_{T \in \mathbf{T}} \mathbb{E} \sum_{t=0}^{\infty} \gamma^t r_t}_{v^r(\pi)} \tag{2}$$

One-step game between the planner and the environment:

1. the learner reveals its policy π
2. the adversary chooses the worst-case dynamics T

Assumption

We consider deterministic systems: $s_{t+1} = T(s_t, a_t)$

Challenge

How to optimize this objective?

- Linear system: $\mathcal{H}_\infty$ control, robust LQ
- Finite state-space: Robust Dynamic Programming
- Non-linear continuous system: ?

Acknowledgements

This work has been supported by CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020, the French Ministry of Higher Education and Research, INRIA, and the French Agence Nationale de la Recherche (ANR).

References

[1] Jean-Francois Hren and Rémi Munos. "Optimistic planning of deterministic systems". In: *European Workshop on Reinforcement Learning*. France, 2008, pp. 151–164.

[2] Brian Paden et al. "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles". In: *IEEE Transactions on Intelligent Vehicles*. 2016. DOI: 10.1109/TIV.2016.2578706.

[3] Vicenç Puig et al. "Simulation of Uncertain Dynamic Systems Described By Interval Models: a Survey". In: *IFAC Proceedings Volumes* 38 (2005), pp. 1239–1250.

Discrete ambiguity and tree-based planning

Assumption

The ambiguity set $\mathbf{T}$ and the action space $\mathcal{A}$ are discrete and **finite**: $\mathbf{T} = \{T_m\}_{m\in[1,M]}$ and $\mathcal{A} = \{a_k\}_{k\in[1,K]}$

We propose a robust version of **optimistic planning** for deterministic systems (Hren and Munos 2008)

Definition 1. *Given node $i \in \mathcal{T}$, define*
The robust value:

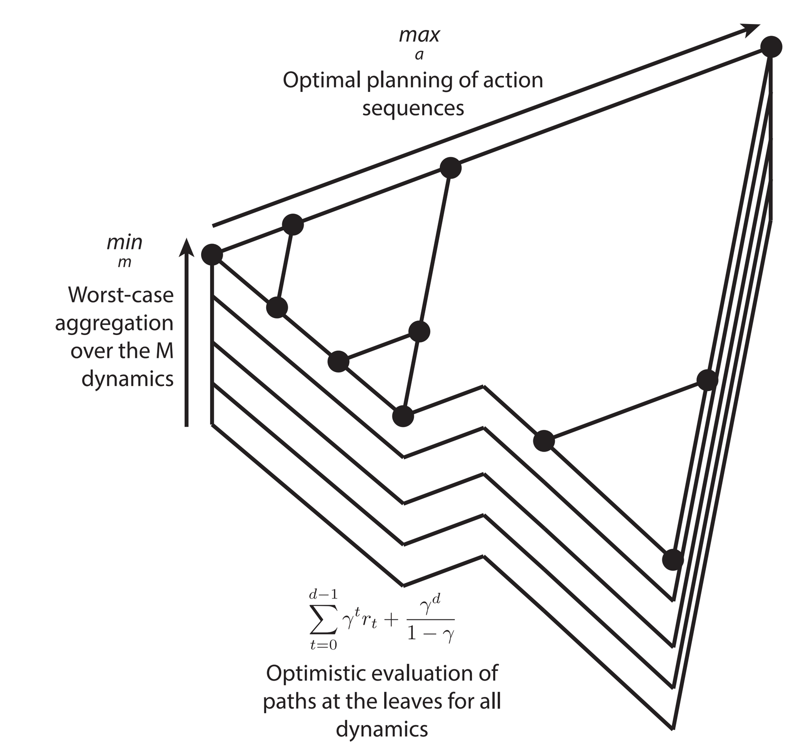
$$v_i^r \stackrel{\text{def}}{=} \max_{\pi \in i\mathcal{A}^\infty} \min_{m\in[1,M]} R_\pi^{T_m}$$

The robust u-value:

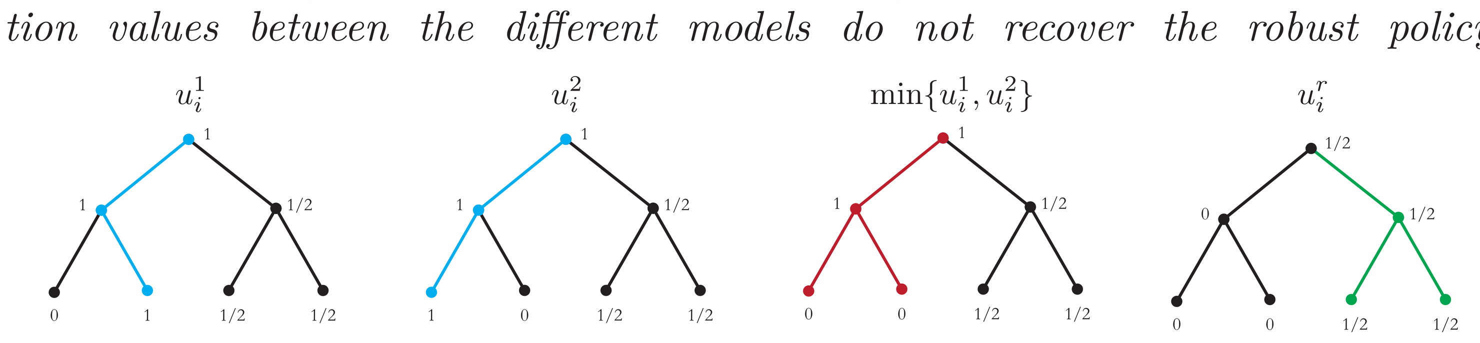
$$u_i^r(n) \stackrel{\text{def}}{=} \begin{cases} \min_{m\in[1,M]} \sum_{t=0}^{d-1} \gamma^t r_t & \text{if } i \in \mathcal{L}_n \,; \\ \max_{a\in\mathcal{A}} u_{ia}^r(n) & \text{if } i \in \mathcal{T}_n \setminus \mathcal{L}_n \end{cases}$$

The robust b-value:

$$b_i^r(n) \stackrel{\text{def}}{=} \begin{cases} \min_{m\in[1,M]} \sum_{t=0}^{d-1} \gamma^t r_t + \frac{\gamma^d}{1-\gamma} & \text{if } i \in \mathcal{L}_n \,; \\ \max_{a\in\mathcal{A}} b_{ia}^r(n) & \text{if } i \in \mathcal{T}_n \setminus \mathcal{L}_n \end{cases}$$



Remark 1 (Ordering of min and max). *Naive comparison of action values between the different models do not recover the robust policy*



Variables

↳ computational budget n
↳ near-optimal branching factor κ
↳ simple regret $\mathcal{R}_n = v^r - v^r_{a(n)}$

Theorem 1 (Regret bound). *Algorithm 1 enjoys a simple regret of:*

$$\text{If } \kappa > 1, \qquad \mathcal{R}_n = O\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right) \tag{3}$$

$$\text{If } \kappa = 1, \qquad \mathcal{R}_n = O\left(\gamma^{\frac{(1-\gamma)^\beta}{c} n}\right) \tag{4}$$

Algorithm 1: Deterministic Robust Optimistic Planning

```
Initialize 𝒯 to a root and expand it.
  Set n = 1.
while Numerical resource available do
    Compute the robust u-values u_i^r(n) and robust b-values b_i^r(n).
    Expand argmax_{i∈L_n} b_i^r(n).
    n = n + 1
return a(n) = argmax_{a∈A} u_a^r(n)
```

Continuous ambiguity and interval-based planning

Approximate the robust objective by a tractable surrogate.

Definition 2. *Given a policy π and current state s_0, define*
***The reachability set** S at time t:*

$$S(t, s_0, \pi) \stackrel{\text{def}}{=} \{s_t : \exists T \in \mathbf{T} \text{ s.t. } s_{k+1} = T(s_k, \pi(s_k))\}$$

***The interval hull** $\square S = [\underline{s}, \overline{s}]$ (Puig et al. 2005)*

$$\underline{s}(t, s_0, \pi) \stackrel{\text{def}}{=} \min S(t, s_0, \pi) \qquad \overline{s}(t, s_0, \pi) \stackrel{\text{def}}{=} \max S(t, s_0, \pi)$$

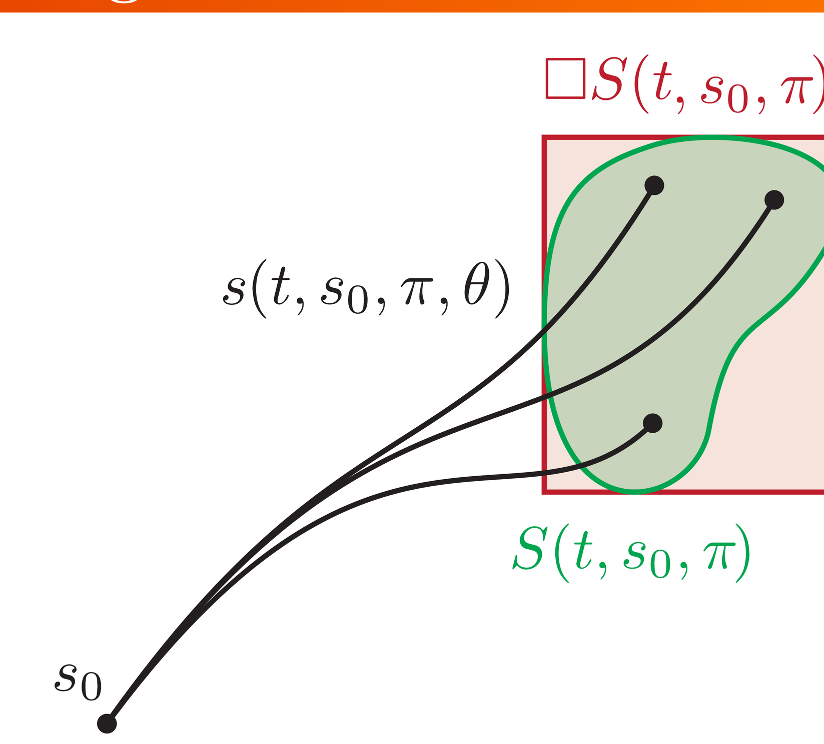
The surrogate objective $\hat{v}^r$

$$\hat{v}^r(\pi) \stackrel{\text{def}}{=} \sum_{t=0}^{H} \gamma^t \min_{s \in \square S(t, s_0, \pi)} r(s, \pi(s)) \tag{5}$$

The approximate performance of a policy is guaranteed on the true environment.

Proposition 1 (Lower bound). *The surrogate objective $\hat{v}^r$ is a lower bound of the true objective v^r:*

$$\forall \pi, \quad \hat{v}^r(\pi) \leq v^r(\pi) \tag{6}$$

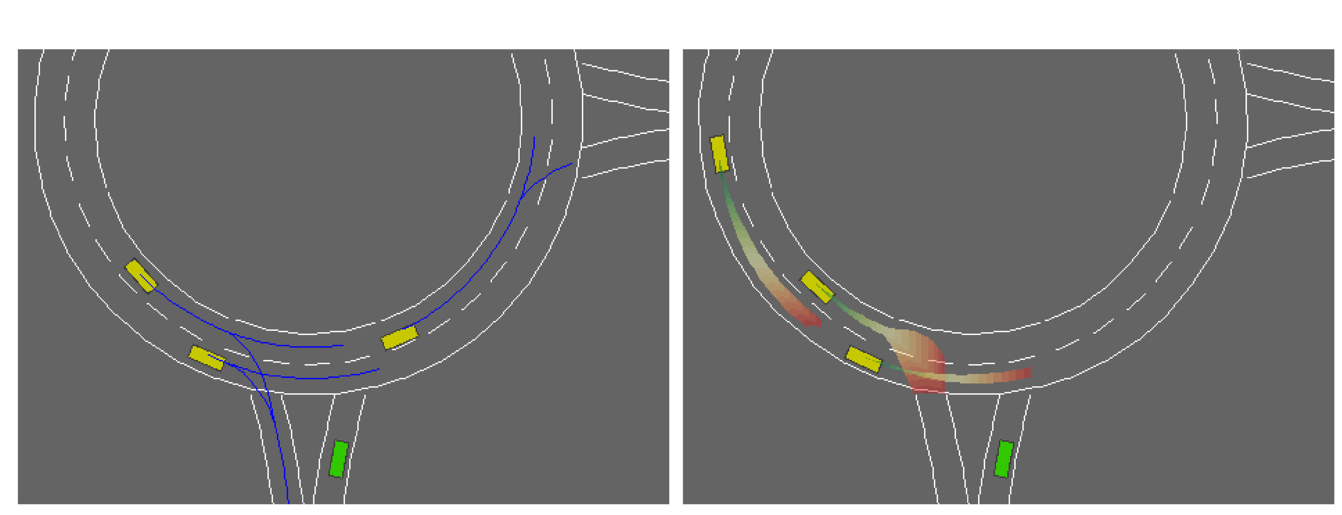


Algorithm 2: Interval-based Robust Control

```
Algorithm robust_control(s_0)
    Initialize a set Π of policies
    while resources available do
        evaluate() each policy π ∈ Π at current state s_0
        Update Π by policy search
    end
    return argmax_{π∈Π} v̂^r(π)
Procedure evaluate(π, s_0)
    Compute the state interval □S(t, s_0, π) on a horizon t ∈ [0, H]
    Minimize r over the intervals □S(t, s_0, π) for all t ∈ [0, H]
    return v̂^r(π)
```

Experiments

We introduce HIGHWAY-ENV, a new environment for simulated highway driving and tactical decision-making[a].
In these experiments, the ego-vehicle is approaching a roundabout with flowing traffic.
We first consider ambiguity with respect to the possible destination of each vehicle (fig a), and then w.r.t. their driving style (fig b).



(a) Discrete ambiguity (b) Continuous ambiguity

Ambiguity	Agent	Worst-case	Mean ± std
None	Oracle	9.83	10.84 ± 0.16
Discrete	Nominal	2.09	8.85 ± 3.53
	Algorithm 1	8.99	10.78 ± 0.34
Continuous	Nominal	1.99	9.95 ± 2.38
	Algorithm 2	7.88	10.73 ± 0.61

(c) Results

[a] Video and source code are available at https://eleurent.github.io/robust-control/