Decision-making in uncertain, dynamic, and interactive environments

Maryam Kamgarpour
École Polytechnique Fédérale de Lausanne, Switzerland

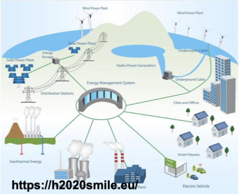
European Control Conference, Stockholm, Sweden

June 28, 2024

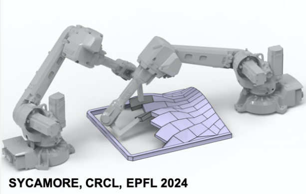
Control systems evolution

From . . .

to . . .



https://h2020smile.eu/


SYCAMORE, CRCL, EPFL 2024

MIRA Ltd. 2014

Stochastic control framework

Stochastic control system

- state x_{t+1} is a sample from $P(.|x_t, u_t)$

Problem: design controller $u_t = \pi(x_t)$ to

$$\begin{array}{ll} \underset{\pi}{\text{minimize}} & \mathbb{E}_P \left[\sum_t c(x_t, u_t) \right] \\ \text{subject to} & x_{t+1} \sim P(.|x_t, u_t) \end{array}$$

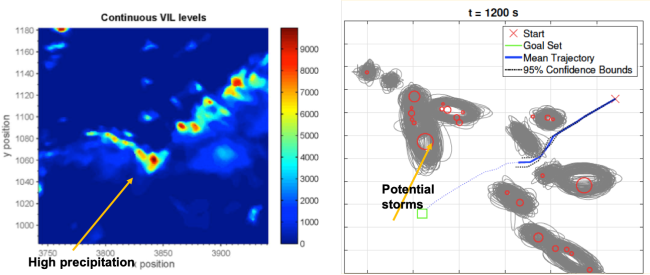
Dynamic programming (DP) [Bellman 1952]

$P(.|x_t, u_t)$, objective $\rightarrow$ $\boxed{\text{DP}}$ $\rightarrow$ optimality conditions for π

Dynamic programming progress and limitations

Progress over the past decades

- addressing more general objectives
- multiagent formulation
- computational tractability



DP to design aircraft trajectory which maximizes probability of safety

Limitations: tractability, incorporating real-time data

Reinforcement learning (RL) approach

~~Given $x_{t+1} \sim P(.|x_t, u_t)$~~

- Optimize π by interacting with the system

RL successes: chess, Go, Starcraft, ...

Key challenge: guarantees for safe control systems

This talk

Towards incorporating performance guarantees in reinforcement learning for safe control systems

Outline

- Safe learning for a single agent
  - Constrained reinforcement learning
  - Log barrier approach
- Multiagent learning and control
  - Challenges compared to single agent setting
  - Multiagent reinforcement learning
- Conclusions and outlook

Safety in learning and control

- Constrained RL approaches
  - Lagrangian formulations [Bharadhwaj et al 2021], [Efroni et al. 2020], [Ding et al. 2021], ...
  - Constrained policy optimization [Achiam, et al. 2017], [Tsung-Yen et al. 2022], [Xu et al. 2021], ...
  - Model-based approaches [Zheng et al. 2020], [Turchetta et al. 2016], [Vaswani et al. 2022], [As et al. 2022], ...

- Control community approaches
  - Learning-based model predictive control [Hewig et al. 2019], [Coulson et al. 2019], [Zanon et al. 2020], [Berberich et al. 2021], [Maddalena et al. 2021], ...
  - Safely training neural net controllers [Zhao et al. 2020], [Xiao et al. 2021], ...
  - Formal methods [Alshiekh et al. 2017], [Fulton et al. 2019], [Hasanbeig et al. 2020], ...
  - Certificate functions, e.g. Lyapunov or control barrier functions [Chow et al. 2018], [Dutta et al. 2018], [Taylor et al. 2019], [Perkins et al. 2002], [Ma et al. 2022], [Emam et al. 2022], [Cohen et al. 2023], [Dowson et al. 2023], , ...
  - Gaussian processes [Akametalu et al. 2014], [Wachi et al. 2018], ...

Constrained reinforcement learning

~~Given $x_{t+1} \sim P(.|x_t, u_t)$~~, parametrize policy: $u_t \sim \pi_\theta(.|x_t)$

$$
\begin{aligned}
\text{minimize} \quad & J(\pi_\theta) := \mathbb{E}_{P,\pi_\theta}\left[\sum_t c_o(x_t, u_t)\right] \\
\text{subject to} \quad & C(\pi_\theta) := \mathbb{E}_{P,\pi_\theta}\left[\sum_t c_s(x_t, u_t)\right] \leq 0
\end{aligned}
$$

Data: system trajectory

Safe learning: Design an algorithm such that π_{θ_k} satisfies constraints and converges to the optimal policy

Safe learning as optimization over policy parameters

Policy parametrization: $\theta \in \mathbb{R}^d$

- linear: $\pi_\theta(x) = \theta^T x$
- Gaussian: $\pi_\theta(u|x) = \mathcal{N}(\phi_\theta(x), \Sigma)$
- ...

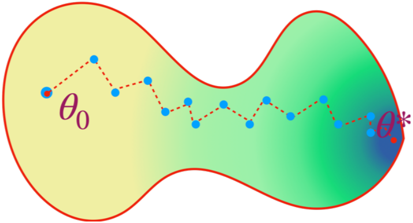
$$\begin{cases} \text{minimize} & \mathbb{E}_{P,\pi}\left[\sum_t c_o(x_t, u_t)\right] \\ \text{subject to} & \mathbb{E}_{P,\pi}\left[\sum_t c_s(x_t, u_t)\right] \leq 0 \end{cases} \implies \begin{cases} \underset{\theta}{\text{minimize}} & J(\theta) \\ \text{subject to} & C(\theta) \leq 0 \end{cases}$$

~~Given $x_{t+1} \sim P(.|x_t, u_t)$~~ $\implies$ $J(.), C(.)$ unknown

Safe learning as blackbox constrained optimization

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & J(\theta) \\ \text{subject to} \quad & C(\theta) \leq 0 \end{aligned}$$

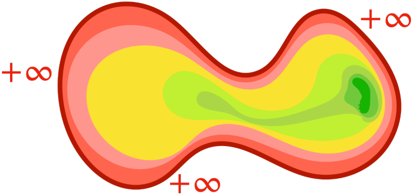
Safe learning: design $\{\theta_k\}_k$ such that $C(\theta_k) \leq 0$ and $\theta_k \to \theta^*$



Challenges: $J(.)$, $C(.)$ non-convex and unknown

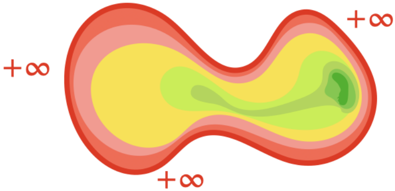
Overview of the proposed approach

- Design a barrier to stay inside the feasible set
- Estimate gradients to find a descent direction
- Take a carefully chosen step in the descent direction



Log barrier of the constrained optimization

- Log barrier of the constraint: $-\log\big(-C(\theta)\big)$



- Unconstrained optimization $\tilde{J}(\theta) = J(\theta) - \eta\log\big(-C(\theta)\big)$
  - $\eta \to 0$: approximate solution $\to$ true solution

Log barrier policy gradient approach

Algorithm: $\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta \tilde{J}(\theta_k)$

Log barrier policy gradient approach

Algorithm: $\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta \tilde{J}(\theta_k)$

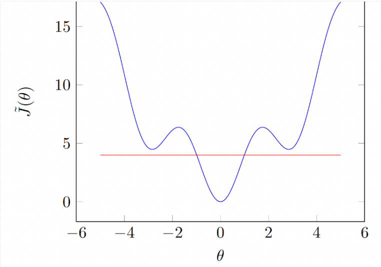
1. Would it converge to optimal policy parameters?
2. How to construct a good estimate of log barrier gradient?
3. How to choose γ_k for safety and convergence?

1. Stationary points of $\tilde{J}$ are nearly optimal

- RL problem structure $\implies$ gradient dominance:

$$J(\theta) - J(\theta^*) \leq \eta + \frac{1}{\nu}\|\nabla_\theta \tilde{J}(\theta)\|_2,\ \nu > 0$$

[Ni, **MK**, ArXiv 2024]



2. Constructing high confidence gradient estimator

$$\nabla_\theta \tilde{J}(\theta) = \nabla_\theta J(\theta) - \eta \frac{\nabla_\theta C(\theta)}{C(\theta)}$$

- Sample average estimates of $\nabla_\theta J(.), \nabla_\theta C(.), C(.)$:

- $P(|\widehat{\nabla_\theta \tilde{J}(\theta)} - \nabla_\theta \tilde{J}(\theta)| \leq \epsilon) \geq 1 - \delta$
    - $n \geq \frac{\eta^2 \ln(\delta^{-1})}{\epsilon^2 (C(\theta))^4}$

3. Ensuring safety of iterates with high probability

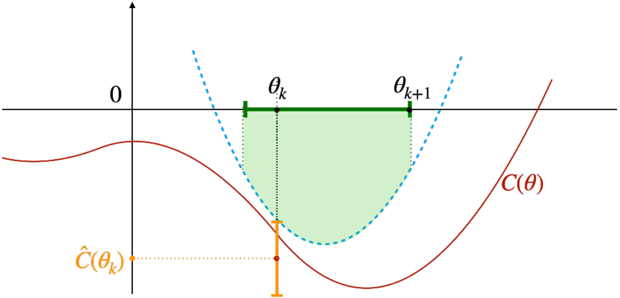
$$\theta_{k+1} = \theta_k - \boxed{\gamma_k} \widehat{\nabla_\theta \tilde{J}(\theta_k)}$$

γ_k should be

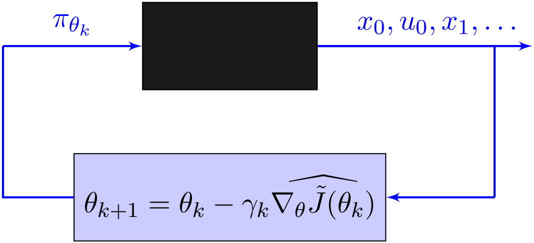
- sufficiently large to make progress
- sufficiently small to keep iterates safe

Approach

- Derive high probability local quadratic bounds on the objective and constraint



Theoretical guarantees for log barrier policy gradient



Theorem

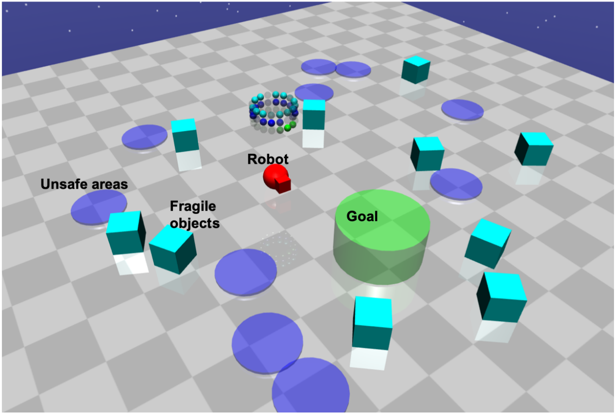
With suitable choices of γ_k, η_k, we have

- *safety: policies π_{θ_k} satisfy constraints with high probability*
- *convergence: $\pi_{\theta_k} \to \pi_{\theta^*}$*
- *complexity: $J(\theta_K) < J(\theta^*) + \epsilon$, with $K = \tilde{O}(\epsilon^{-6})$ trajectories (compare to $\tilde{O}(\epsilon^{-2})$ in unconstrained case)*

*[Usmanova, As, **MK**, Krause, JMLR 2024], [Ni, **MK**, ArXiv 2024]*

Case study in safe learning

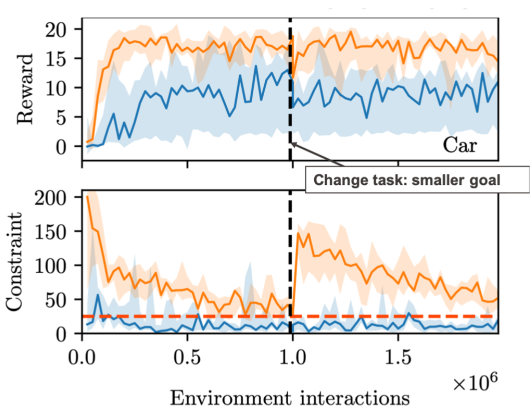
- Objective: reach the goal while avoiding obstacles
- Challenge: unknown dynamics and environment
- Approach: learn a neural network policy directly from images



From Open AI's Safety Gym

Log barrier approach in Safety Gym benchmark

Constraint satisfaction during learning but slow convergence



Our approach in blue, Lagrangian approach in orange

Lagrangian: solves constrained RL, without safety during learning [As et al. 2022]

Outline

- Safe learning for a single agent
  - Constrained reinforcement learning
  - Log barrier policy gradient approach
- Multiagent learning and control
  - Challenges compared to single agent setting
  - Multiagent reinforcement learning
- Conclusions and outlook

Multiagent systems

Several decision-makers with coupled objectives or constraints

Multiagent systems formulation

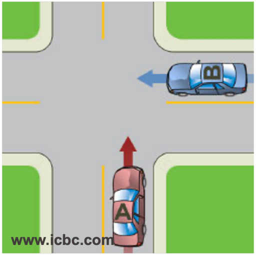
Warmup: static, unconstrained, and deterministic

- N agents, with agent $i \in \{1, \ldots, N\}$
  - action θ^i, joint action $\boldsymbol{\theta} = (\theta^i, \theta^{-i})$
  - objective $J^i(\theta^i, \theta^{-i})$

- Objectives: $\{J^i(.)\}_{i=1}^N \implies$ no single function to optimize

Equilibrium as a desired solution

- $\boldsymbol{\theta}^*$ is equilibrium: $\forall i, \quad J^i(\theta^{*i}, \theta^{*-i}) = \min_{\theta^i} J^i(\theta^i, \theta^{*-i})$
  - agent i has no reason to deviate from θ^i



- differentiable $J^i(\boldsymbol{\theta}) \implies \nabla_{\theta^i} J^i(\boldsymbol{\theta}^*) = 0$

Learning in multiagent systems

Agent i does not know $J^i(.)$ but can query it

How do agents learn an equilibrium?

Uncoupled gradient-based learning in multiagent setting

Suppose each agent runs: $\theta^i_{k+1} = \theta^i_k - \gamma_k \widehat{\nabla_{\theta^i} J^i(\boldsymbol{\theta}_k)}$

Challenges compared to the single agent setting:

1. How can agent i estimate $\nabla_{\theta^i} J^i(\boldsymbol{\theta})$ without knowing $\boldsymbol{\theta}$?
   - use one-point gradient estimators but have high variance
2. Under which conditions do we have convergence?

Single agent convergence conditions do not apply

Consider known $\nabla_{\theta^i} J^i(\boldsymbol{\theta})$'s
Agents' learning dynamics:

$$\begin{bmatrix} \theta^1_{k+1} \\ \vdots \\ \theta^N_{k+1} \end{bmatrix} = \begin{bmatrix} \theta^1_{k} \\ \vdots \\ \theta^N_{k} \end{bmatrix} - \gamma_k \underbrace{\begin{bmatrix} \nabla_{\theta^1} J^1(\boldsymbol{\theta}_k) \\ \vdots \\ \nabla_{\theta^N} J^N(\boldsymbol{\theta}_k) \end{bmatrix}}_{\neq \nabla_\theta J(\boldsymbol{\theta})}$$

- ex: $J^1(\boldsymbol{\theta}) = \theta^1\theta^2 = -J^2(\boldsymbol{\theta})$, $\begin{bmatrix} \nabla_{\theta^1} J^1(\boldsymbol{\theta}) \\ \nabla_{\theta^2} J^2(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix}$
- single agent analysis approaches don't generally work

Sufficient conditions for convergence

- Pseudo-gradient: $\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} \nabla_{\theta^1} J^1(\boldsymbol{\theta}_k) \\ \vdots \\ \nabla_{\theta^N} J^N(\boldsymbol{\theta}_k) \end{bmatrix}$
- Algorithm: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \gamma_k \hat{\boldsymbol{M}}(\boldsymbol{\theta})$
- Sufficient convergence conditions based on $\boldsymbol{M}(\boldsymbol{\theta})$
  - ex: assume strong monotonicity of $\boldsymbol{M}(\boldsymbol{\theta})$

Progress over years

- Constrained setting, convergence rates: [Tatarenko, **MK**, IEEE TAC 2019, IEEE , ECC 2024], [Bravo et al. 2018], [Gao, Pavel, 2022],[Narang et al. 2023], ...

Open challenge: convergence conditions for stochastic dynamical setting

Multiagent stochastic control formulation

- Stochastic dynamics controlled by agents: $x_{t+1} \sim P(.|x_t, u_t^1, \ldots, u_t^N)$
- Agent i's decision: $u_t^i = \pi^i(x_t)$, $\boldsymbol{\pi} = (\pi^i, \pi^{-i})$
- Agent i's cost: $J^i(\pi^i, \pi^{-i}) = \mathbb{E}_{P,\boldsymbol{\pi}}\left[\sum_t c^i(x_t, u_t^1, \ldots, u_t^N)\right]$

Compute an equilibrium policy $\boldsymbol{\pi} = (\pi^1, \ldots, \pi^N)$ for $\{J^i(\boldsymbol{\pi})\}_{i=1}^N$

Multiagent reinforcement learning approach

~~Given $x_{t+1} \sim P(.|x_t, u_t^1, \dots, u_t^N)$~~

- Parametrize a stochastic policy $u_t^i \sim \pi_{\theta^i}(.|x_t)$, $\theta^i \in \mathbb{R}^d$
- Find equilibrium $\boldsymbol{\theta}^* = (\theta^1, \dots, \theta^N)$ by interacting with the system

Challenge: learning algorithms with provable convergence

Challenging even in linear quadratic setting

single agent

$$J(\theta) = \mathbb{E}[\sum_{t=0}^{\infty} x_t^T Q x_t + u_t^T R u_t]$$

$$x_{t+1} = Ax_t + Bu_t$$

$$u_t = \theta^T x_t, \ x_0 \sim \mathcal{D}$$

Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator

Maryam Fazel [*1] **Rong Ge** [*2] **Sham M. Kakade** [*1] **Mehran Mesbahi** [*1]

Abstract

Direct policy gradient methods for reinforcement learning and continuous control problems are a popular approach for a variety of reasons: 1) they

2016) and Atari game playing (Mnih et al., 2015). Deep reinforcement learning (DeepRL) is becoming increasingly popular for tackling such challenging sequential decision making problems.

multiagent

$$J^i(\boldsymbol{\theta}) = \mathbb{E}[\sum_{t=0}^{\infty} x_t^T Q^i x_t + (u^i)_t^T R^i u_t^i]$$

$$x_{t+1} = Ax_t + \sum_{i=1}^{N} Bu_t^i$$

$$u_t^i = (\theta^i)^T x_t, \ x_0 \sim \mathcal{D}$$

Policy-Gradient Algorithms Have No Guarantees of Convergence in Linear Quadratic Games

Eric Mazumdar
University of California, Berkeley
Berkeley, CA
mazumdar@berkeley.edu

Lillian J. Ratliff
University of Washington
Seattle, WA
ratliffl@uw.edu

Michael I. Jordan
University of California, Berkeley
Berkeley, CA
jordan@cs.berkeley.edu

S. Shankar Sastry
University of California, Berkeley
Berkeley, CA
sastry@coe.berkeley.edu

ABSTRACT

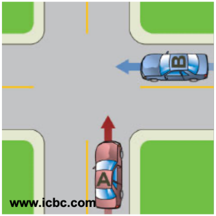
We show by counterexample that policy-gradient algorithms have no guarantees of even local convergence to Nash equilibria in continuous action and state space multi-agent settings. To do so, we analyze gradient-play in N–player general-sum linear quadratic games, a classic game setting which is recently emerging as a bench-

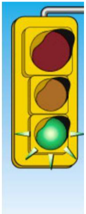
of multi-agent reinforcement learning have made use of policy optimization algorithms such as multi-agent actor-critic [13, 17, 30], multi-agent proximal policy optimization [2], and even simple multi-agent policy-gradients [15] in problems where the various agents have high-dimensional continuous state and action spaces like StarCraft II [32].

Relaxing the equilibrium notion

A *probability distribution* $\mathcal{P}^*$ on $\boldsymbol{\theta}$ is an equilibrium

$$\forall i \quad \mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{P}^*}[J^i(\boldsymbol{\theta})] \leq \mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{P}^*}[J^i(\tilde{\theta}^i, \theta^{-i})], \ \forall \tilde{\theta}^i$$





- Focus: learning algorithms that scale with number of agents

[**MK** with Sessa, Maddux, Bugonovic, Krause, NeurIPS 2020, AISTATS 2019, 2020, 2024, ICML 2021,2022]

Approach: model-based learning of equilibrium distribution

- Initialize $\mathcal{P}_0$. For $k = 0, 1, \dots$
  - sample $(\pi_k^1, \dots, \pi_k^N) \sim \mathcal{P}_k$

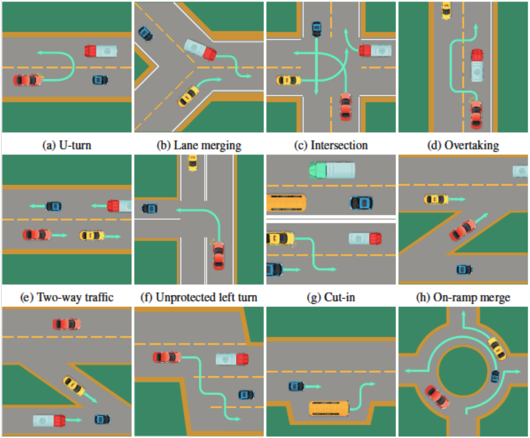
  - estimate $P(.|x_t, u_t^1, \dots, u_t^N) \to \hat{J}_k^i(\boldsymbol{\theta}\ )$
  - compute $\mathcal{P}_{k+1}$ as the equilibrium distribution of $\hat{J}_k^i(\theta)$

Finite-time convergence to an equilibrium distribution [Sessa, **MK**, Krause, ICML 2022]

Case study: Multiagent RL in autonomous driving

SMARTS autonomous car simulation environment [Zhou et al. 2021]

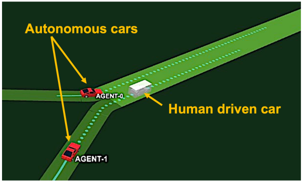
- testing multiagent RL algorithms for autonomous driving
- realistic traffic data and car dynamics

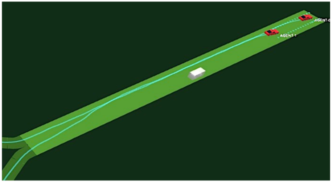


(a) U-turn (b) Lane merging (c) Intersection (d) Overtaking
(e) Two-way traffic (f) Unprotected left turn (g) Cut-in (h) On-ramp merge
(i) Off-ramp merge (j) Cascading lane change (k) Bus stop merge (l) Roundabout

Model-based multiagent RL for autonomous driving

- Objective: progress towards the goal, avoid collision
- Dynamics: $P(.|x_t, u_t^1, u_t^2)$
  - x: positions and velocities of cars
  - $u^i \in \{$keep lane, slow down, turn right, turn left$\}$, $i = 1, 2$
  - $\pi_{\theta^i}(x)$: parametrized by neural networks, $i = 1, 2$

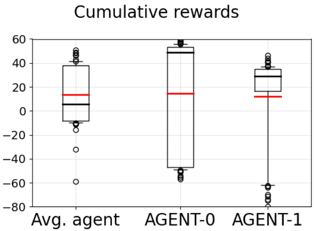




The autonomous cars can coordinate and overtake the human-driven car

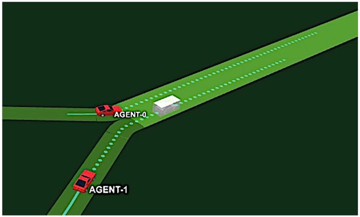
Outcome of the multiagent learning approach

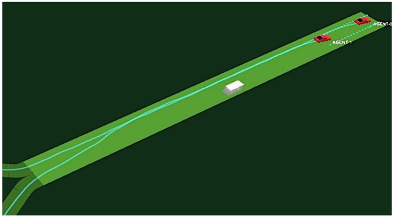
Learning to coordinate $\Longrightarrow$ less breaking, more successful merges



Single-agent optima

Multiagent equilibrium





Outline

- Safe learning for a single agent
  - Constrained reinforcement learning
  - Log barrier policy gradient approach
- Multiagent learning and control
  - Challenges compared to single agent setting
  - Multiagent reinforcement learning
- Conclusions and outlook

Recap

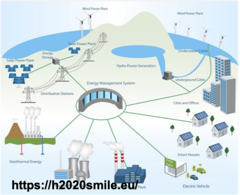
- Stochastic control: a powerful modeling framework
- RL: data-driven approach to stochastic control
- RL needs guarantees for safe control

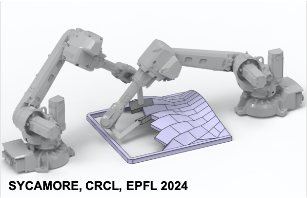
We provided algorithms with proven performance guarantees for

- safe learning: satisfying constraints during system interactions
- multiagent learning: multiple objectives and decision-makers

Outlook: open theoretical challenges

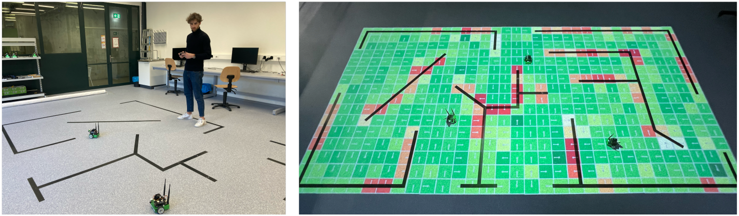
- Safe learning algorithms for multiagent stochastic systems
- Provable algorithms under partial and asymmetric information
- Learning of "good" equilibria, mechanism design
- ...





Outlook: bridging the gap between theory and application

- Improving sample complexity
- Robustness to model mismatch



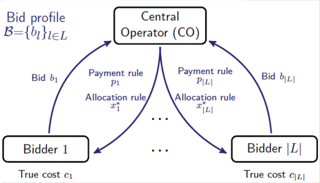
Autonomous car experiments in the lab

Outlook: bridging the gap between theory and application

Joint work with EPFL CRCL

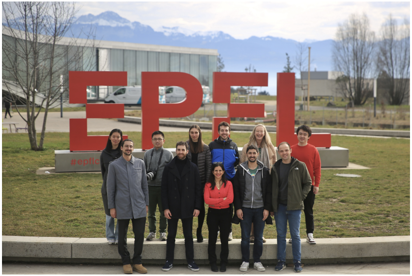
Outlook: bridging the gap between theory and application

Further applications of multiagent stochastic control:
Learning and mechanism design in power markets



Acknowledgements

- Former and current PhD students: O Karaca, L Furieri, P Giuseppe Sessa, I Usmanova, B Guo, Kai Ren, T Ni, A Maddux, A Schlaginhaufen, G Salizzoni, S Hosseinirad, G Vallat
- Collaborators: T Tatarenko, T Summers, N Walton, T Wood, A Papachristodoulo, E Tedeschi, J Lygeros, G Ferrari Trecate, I. Bugonovic, H Ahn, C Tomlin, R Ouhamma, Z Wang, S Parascho, A Abate, J Kazempour, G Hug, G Ranade, C Jones
- Funding : ERC, NSERC Canada, Swiss National Fund, NCCR Automation



https://www.epfl.ch/labs/sycamore/

Constrained RL formulations

- Finite horizon: $\mathbb{E}_{P,\pi}\left[\sum_{t=0}^{T} c_s(x_t, u_t)\right]$
  - can encode probability of trajectory staying inside a safe set [Tkachev et al. 2013]
- Infinite horizon: $\mathbb{E}_{P,\pi}\left[\sum_{t=0}^{T} \lambda^t c_s(x_t, u_t)\right]$
  - discount factor $0 < \lambda < 1$

Focus in this talk: discounted setting

Parameters

$n = \mathcal{O}(\epsilon^{-4} \ln \frac{1}{\beta\epsilon})$ and $H = \mathcal{O}(\ln \frac{1}{\epsilon})$, and $T = \mathcal{O}(\epsilon^{-2})$ to ensure optimality and safe exploration with confidence $1 - \beta$