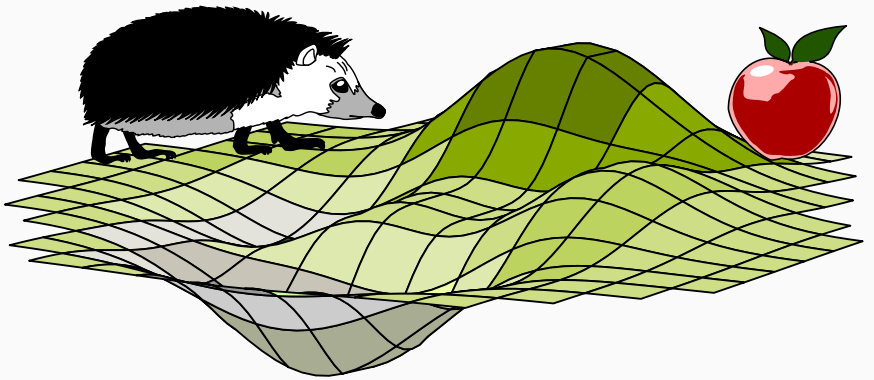
REINFORCEMENT LEARNING

SAFETY

Pavel Osinenko



Safety

We have studied a variety of RL methods and their convergence. What is not addressed so often is the safety aspect: we can't in general guarantee that the state of the environment stays stable and satisfies stated constraints.

This is a major problem of online RL: per se, there are no such guarantees, unless special measures are taken.

RL methods that do address this problem may generally be put into a category called safe RL

Safety

In this lecture, we will give an overview of approaches to safe RL.

But first, one crucial dilemma should discussed, namely, model-free vs. model-based control.

RL is particularly attractive in its model-free, purely data-driven variants such as actor-critic online Q-learning without state prediction.

But here is a nice quote by Lewis & Vrabie*:

". . . not specifically considering the dynamics also makes it impossible to provide explicit proofs of stability and performance such as are required for acceptance by the Control Systems Community."

* Lewis, F. L., & Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. IEEE circuits and systems magazine, 9(3), 32-50.

Safety

Furthermore, they stated that controllers without performance, stability and robustness guarantees would not be recognized by industry.
This puts strong limitations on applicability of model-free RL in such fields as robotics, self-driving vehicles, unmanned aerial vehicles, commercial vehicles, industrial manipulators, agricultural machinery, space, medicine and many more.

A distinction should be made though between methods that explicitly use an environment model (or a part thereof) and those assume something quantitative about the environment

Safety

So, three main categories of safe RL may be distinguished :

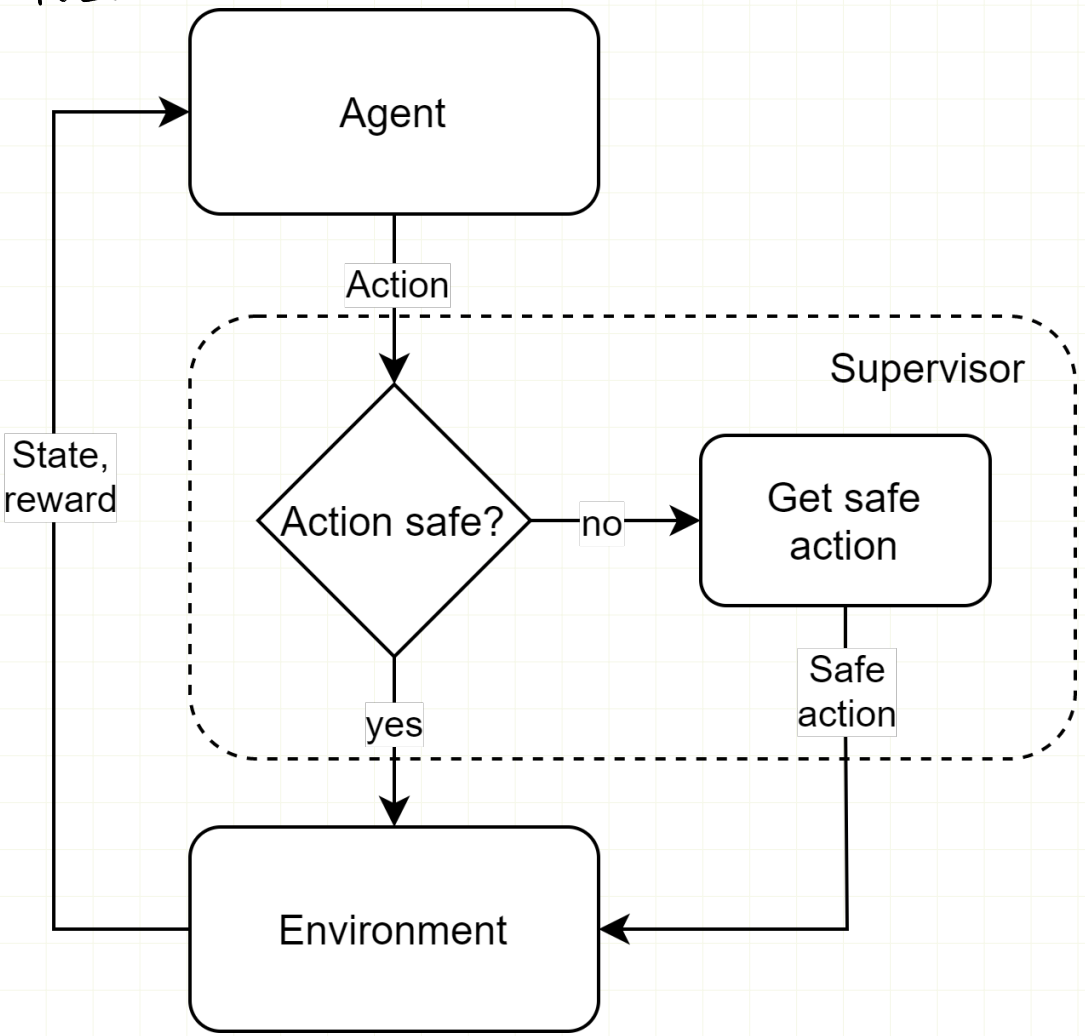
1. shield-based RL
2. Lyapunov RL
3. MPC/RL fusion

The first category is the most straightforward and quite primitive in its basic variants

Safety

Highlights of shield-based RL

- filters out unsafe actions
- most primitive scenario: human overseer
- guarantees based only on the training set
- advanced shields: based on formal logic

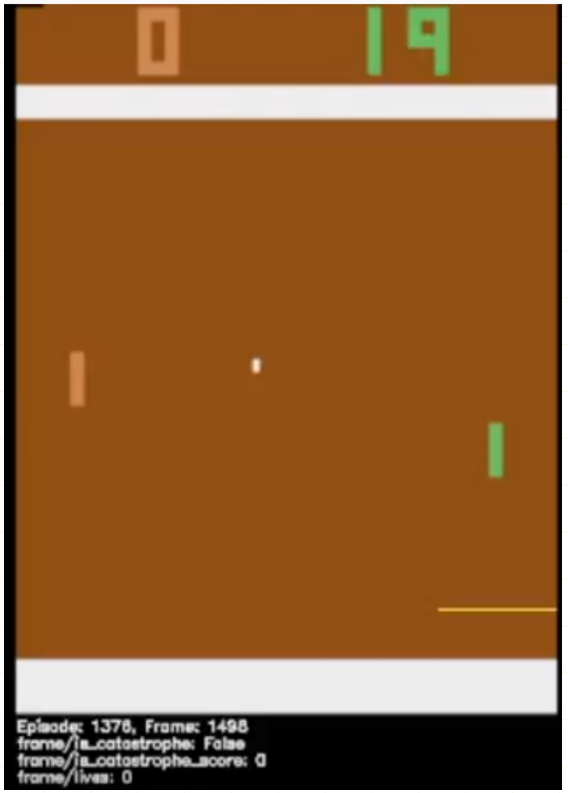


Safety

(cont.)

- various kinds of formal logic systems — temporal logic, differential dynamic logic etc.

- two major principles — correctness: shied must indeed guarantee safety
  minimal inference: shield must not limit the agent's learning too much



A virtual wall

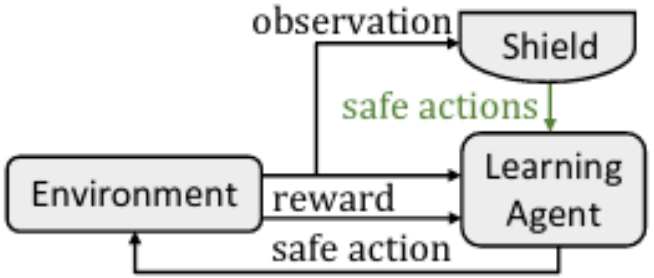
Agent learning to win Atari 2600 Pong while avoiding an unsafe region.

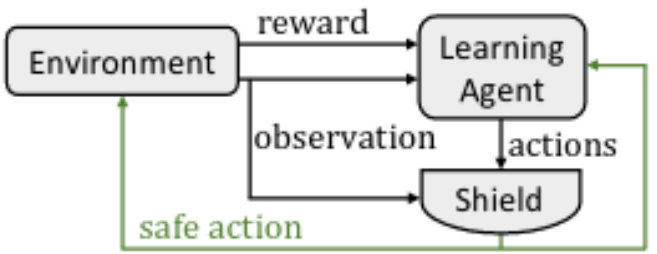
Source: Saunders, W., Sastry, G., Stuhlmueller, A., & Evans, O. (2017).
Trial without error: Towards safe reinforcement learning via human intervention

Safety

(cont.)

Approaches based on formal logic need complex and tedious design of shields





Shield filtering out unsafe actions.

Source: Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2017). Safe reinforcement learning via shielding. arXiv preprint 1708.08611

Also, accurate environment models are required

Safety

(cont.)

An example algorithm of a formal-shield-based agent training:

Justified Speculative Learning

```
1  JSCGeneric(init, (S,A,R,E), choose, update,
       done, CM, MM) {
2    prev := curr := init;
3    a0   := NOP;
4    while (!done(curr)) {
5      if (MM(prev, a0, curr))
6        u := choose({a ∈ A | CM(a,curr)});
7      else
8        u := choose(A);
9      prev := curr;
10     curr := E(u, prev);
11     update(prev, u, curr);
12   }
13 }
```

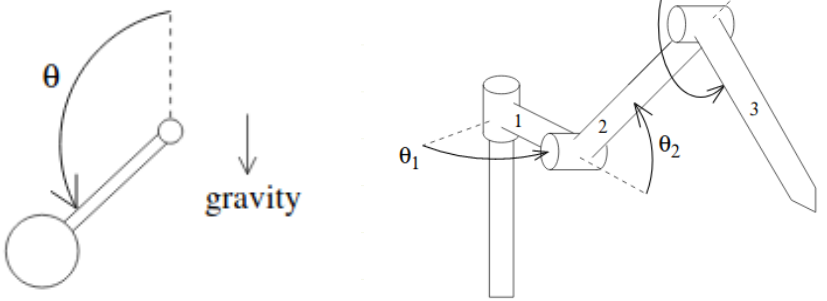
Controller monitor (CM) selects safe actions if necessary

Souce: Fulton, N., & Platzer, A. (2018, February). Safe reinforcement learning via formal methods. In AAAI Conference on Artificial Intelligence

Safety

The Lyapunov-based RL employs the classical stability theory originating from A. Lyapunov and then extended to the stochastic case by Khasminskii, Kushner, Mao, Krstic and others.

This theory was already recognized in the early 2000s by Perkins & Barto



Systems used by Perkins&Barto

Source: Perkins, T. J., & Barto, A. G. (2002). Lyapunov design for safe reinforcement learning. Journal of Machine Learning Research, 3(Dec), 803-832

Safety

(cont.)

The idea of Perkins & Barto was to restrict the agent choices so that one of the Lyapunov stability theorems apply.

Chow et al. extended this methodology to MDPs via a policy iteration algorithm with special (Lyapunov) constraints. The approach is not online

Input: Initial feasible policy π_0;
for $k = 0, 1, 2, \ldots$ **do**
  Step 0: With $\pi_b = \pi_k$, evaluate the Lyapunov function L_{ϵ_k}, where ϵ_k is a solution of (5)
  Step 1: Evaluate the cost value function $V_{\pi_k}(x) = C_{\pi_k}(x)$; Then update the policy by solving the following problem: $\pi_{k+1}(\cdot|x) \in \text{argmin}_{\pi \in \mathcal{F}_{L_{\epsilon_k}}(x)} T_{\pi,c}[V_{\pi_k}](x), \forall x \in \mathcal{X}'$
end for
Return Final policy π_{k^*}

Safe policy iteration via a Lypunov constraint

Source: Chow, Y., Nachum, O., Duenez-Guzman, E., & Ghavamzadeh, M. (2018). A lyapunov-based approach to safe reinforcement learning. In Advances in neural information processing systems (pp. 8092-8101)

Safety

(cont.)

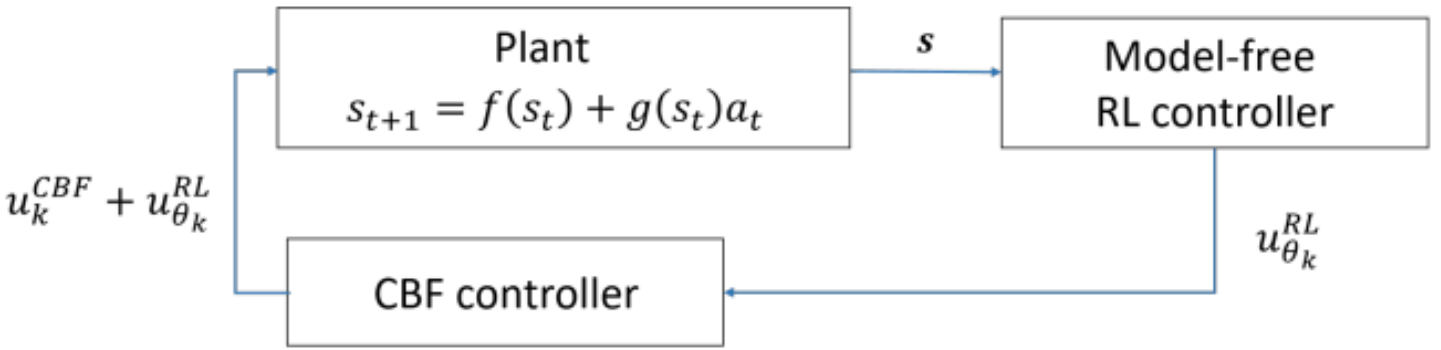
Other Lyapunov RL approaches use, e.g., state space splitting and constraint verification in each separate cell.

It should be noted that Lyapunov functions are primarily used to describe stability, which is a kind of attractivity property, but a counterpart of a Lyapunov function called barrier certificate may be used to describe a repelling property in the sense of avoidance of unsafe regions

Safety

(cont.)

Thus, a barrier certificate is like an „upside down" Lyapunov function. Here is an example of safe RL using such a certificate:



Combination of model-based barrier function controller with a model-free RL controller

Source: Cheng, R., Orosz, G., Murray, R. M., & Burdick, J. W. (2019, July). End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 3387-3395)

Safety

(cont.)

The corresponding certificate was computed at each time step via quadratic programming. Computational tractability was an issue. An approximate version was suggested, but it had no safety guarantees.

All in all, Lyapunov RL is a strong candidate to go for

Safety

Finally, MPC / RL fusion.

This is perhaps the most promising direction in safe RL.

It tries to take best from the both worlds:

MPC	RL
tractable	adaptive, data-driven
stabilizing	aims ∞-horizon performance
addresses constraints	

Think of the following: what is the best way to ensure constraint satisfaction at all times? Yes, one may design Lyapunov functions and barrier certificates, but it's best to anticipate possible constraint violation in future

Safety

(cont.)

When we say this, we mean MPC by default. Under certain conditions, it is recursively feasible: provided that the constraints are satisfiable at the initial step, they are satisfiable at all the subsequent steps. This is an extremely powerful property.

Safety

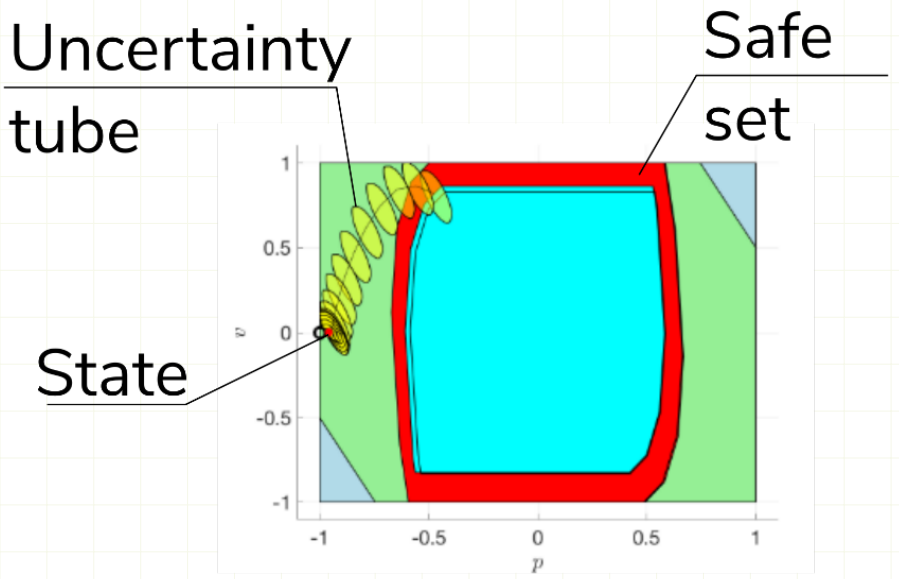
(cont.)

Here is an example that builds upon robust MPC

Policy parameters

$$\theta^\star := \min_{\boldsymbol{\theta}} \sum_{k=0}^{n} \psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta})$$
$$\text{s.t.} \quad \mathbf{s}_{k+1} \in \hat{\mathbf{S}}_+(\mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}), \quad \forall\, k$$

Safety constraint on the future states



Left: policy parameter update under safety constraints

Right: State trajectory entering a safe set (a robust positively invariant set) under reinforcement learning using robust model-predictive

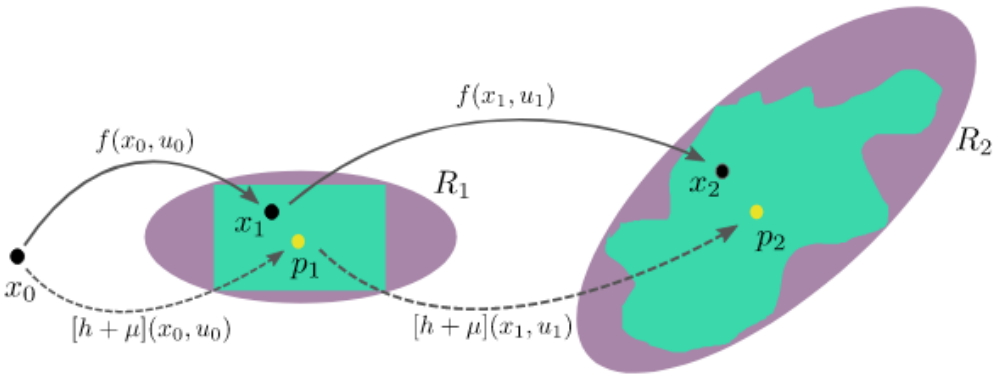
Source: Zanon & Gros. Safe Reinforcement Learning Using Robust MPC

Safety

(cont.)

Another approach seeks to solve a learning-based MPC problem and resorts to a safe policy in case of infeasibility. The approach needs a well-calibrated statistical model of the environment

$$
\begin{aligned}
\underset{\pi_0,..,\pi_{T-1}}{\text{minimize}} \quad & J_t(R_0,..,R_T) && \text{(31a)}\\
\text{subject to} \quad & R_{t+1} = \tilde{m}(R_t,\pi_t),\ t=0,..,T-1 && \text{(31b)}\\
& R_t \subset \mathcal{X},\ t=1,..,T-1 && \text{(31c)}\\
& \pi_t(R_t) \subset \mathcal{U}, t=0,..,T-1 && \text{(31d)}\\
& R_T \subset \mathcal{X}_{\text{safe}}, && \text{(31e)}
\end{aligned}
$$

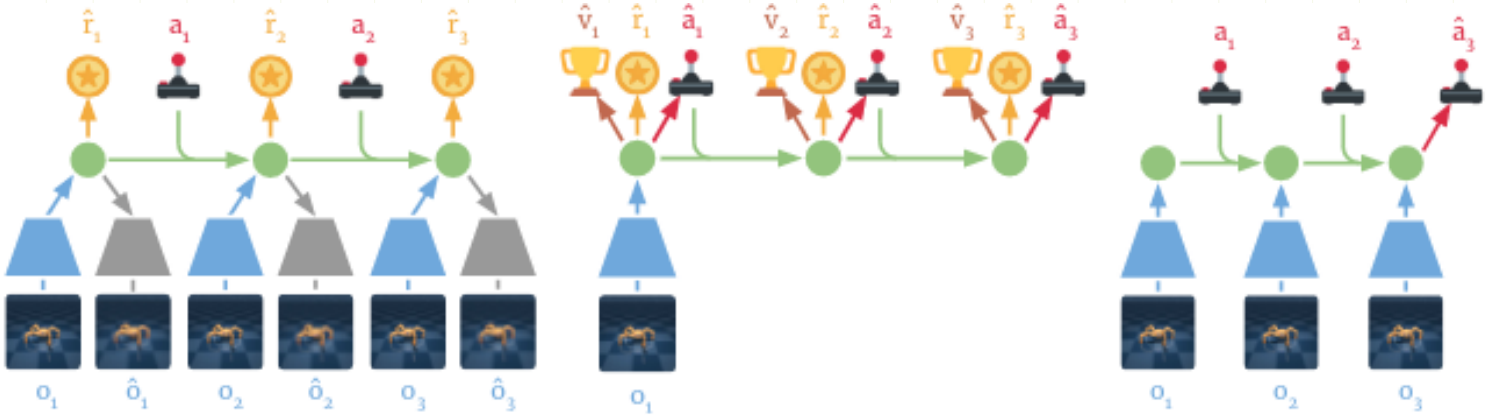


Left: Safe MPC via propagation of confidence ellipsoids (R)

Right: Propagation of uncertainty based on a well-calibrated statistical model

Source: Koller, T., Berkenkamp, F., Turchetta, M., & Krause, A. (2018, December). Learning-based model predictive control for safe exploration. In 2018 IEEE conference on decision and control (CDC) (pp. 6059-6066). IEEE.

(cont.) The RL dreamer is actually a variant of adaptive MPC



(a) Learn dynamics from experience (b) Learn behavior in imagination (c) Act in the environment

Initialize dataset $\mathcal{D}$ with S random seed episodes.
Initialize neural network parameters θ, ϕ, ψ randomly.
while *not converged* **do**
  for *update step* $c = 1..C$ **do**
    // Dynamics learning
    // Behavior learning
  // Environment interaction
  $o_1 \leftarrow$ env.reset()
  for *time step* $t = 1..T$ **do**
    Compute $s_t \sim p_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t)$ from history.
    Compute $a_t \sim q_\phi(a_t \mid s_t)$ with the action model.
    Add exploration noise to action.
    $r_t, o_{t+1} \leftarrow$ env.step(a_t).
  Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, a_t, r_t)_{t=1}^{T}\}$.

Up: Exploration & exploitation adaptive MPC scheme of RL dreamer

Left: Its algorithm

Source: Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination

Safety

A bit of our research. First, in Lyapunov safe RL:
„JACS" — joint actor-critic (stabilizing) — an RL agent for sample-and-hold modes with stability guarantess.

From:

$$\theta_{new} := \arg\min_{\theta} \{ \rho + \Delta \hat{V} \}$$

$$\vartheta_{new} := \arg\min_{\vartheta} \{ \rho + \hat{V} \}$$

⇒

To:

$$(\theta_{new}, \vartheta_{new}) := \arg\min_{(\theta,\vartheta)} J^{ac}(\theta, \vartheta)$$

$$\text{s.t. } \Delta_{\text{inter-sample}} \hat{V} < 0$$

$$\text{s.t. } \Delta_{\text{sample-to-sample}} \hat{V} < 0$$

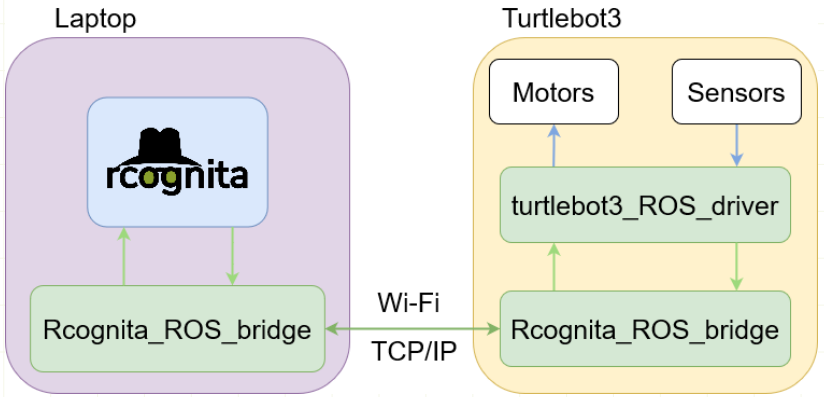
Reference: Osinenko, P., Beckenbach, L., Göhrt, T., & Streif, S. (2020).
A reinforcement learning method with closed-loop stability guarantee.
IFAC World Congress

Safety

(cont.)

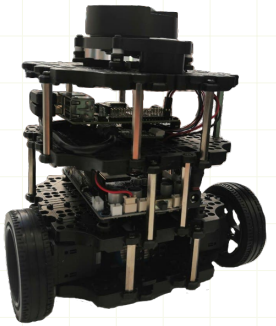
Test with a mobile robot showed remarkable performance improvement

The testbed



The setup with rcognita package
github.com/AIDynamicAction/rcognita

The robot



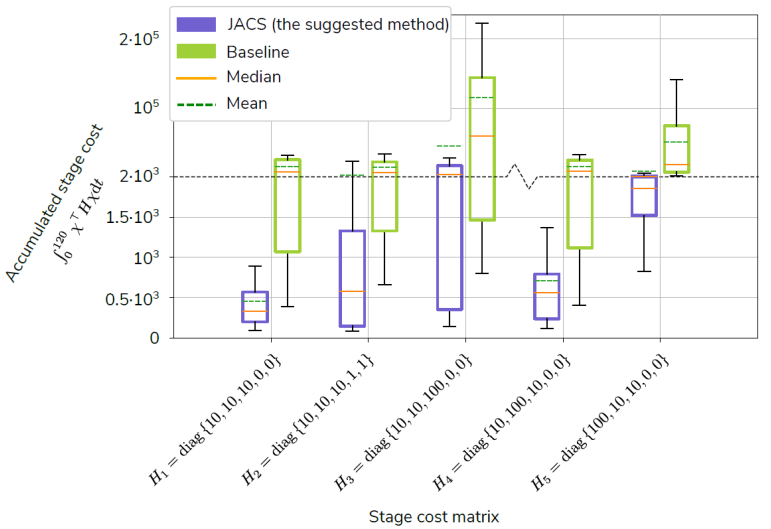


Fig. 6: The accumulated stage cost of the agents, i.e., $\int_0^{120} \chi^\top H \chi \, dt$ (see Section VI-A). The box bounds are, respectively, the first and third quartiles Q1, Q3. The whiskers are the same quartiles plus/minus one and a half interquartile range Q3-Q1. The y-axis scale is split for better reading.

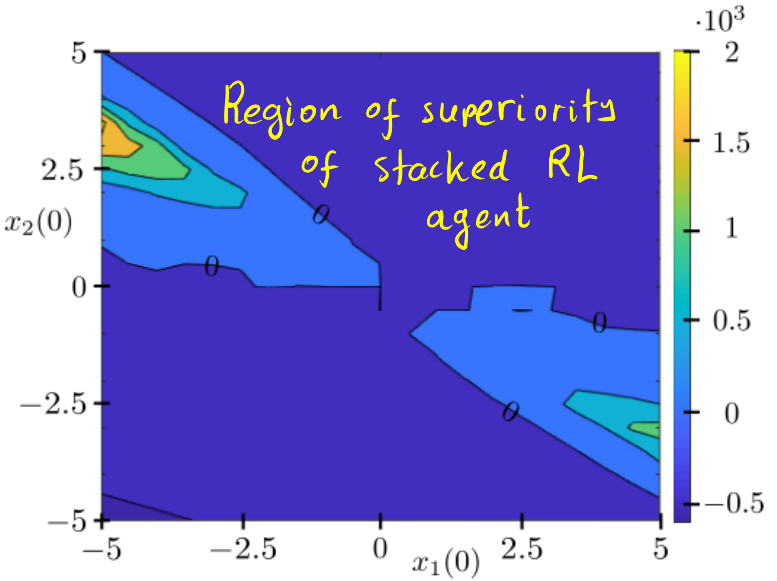
Safety

(cont.)

Another direction : stacked Q-learning – a predictive RL agent with adaptive stage costs based on Q-learning (then a general predictive agent with learning costs).

The main question was : what if we substitute the MPC stage cost for something learnable?

Conducted systematic design and analysis of constraints so as to retain MPC safety (and even relax some common MPC assumptions!)



A case study with a non-linear environment (cost-to-go comparison for various initial states)

Reference: Beckenbach, L., Osinenko, P., & Streif, S. (2020, May). On closed-loop stability of model predictive controllers with learning costs. In 2020 European Control Conference (ECC) (pp. 184-189). IEEE.