Reinforcement Learning for Particle Accelerators

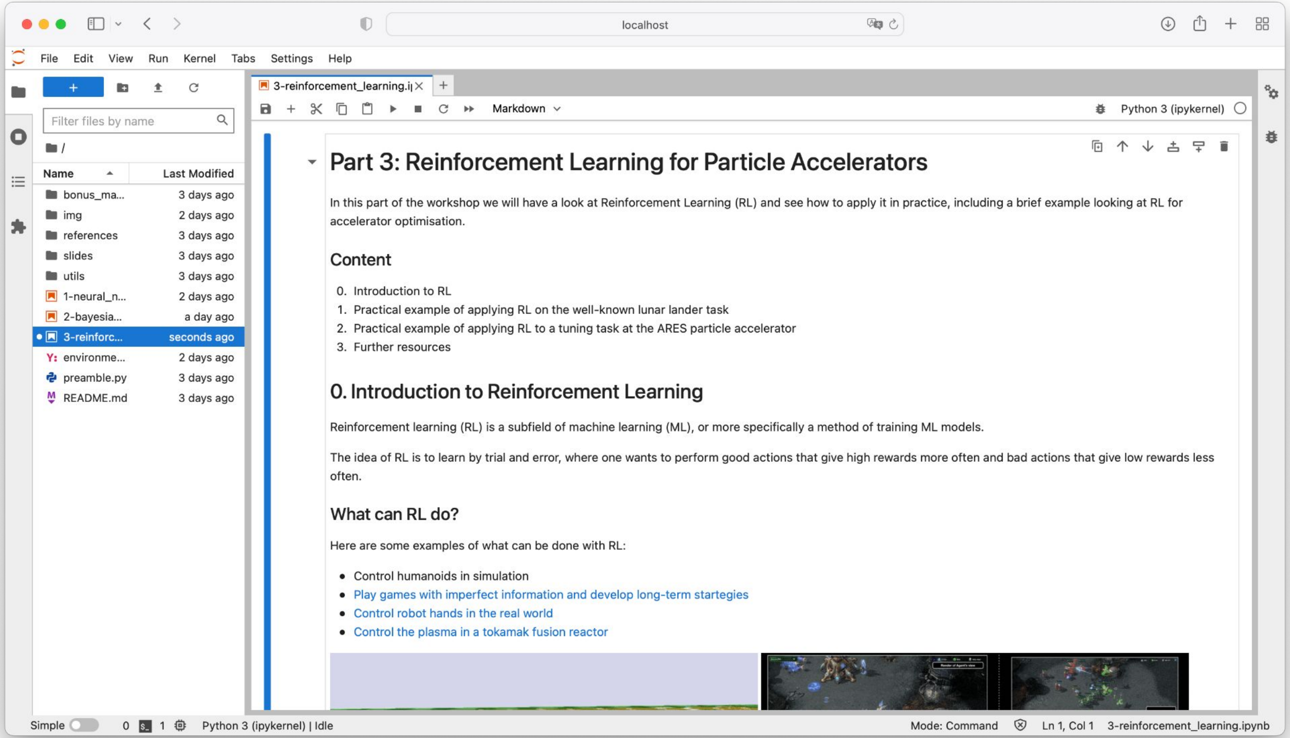
An Introduction

Jan Kaiser and Oliver Stein

MT-ARD-ST3 pre-meeting ML workshop

Try Reinforcement Learning Yourself

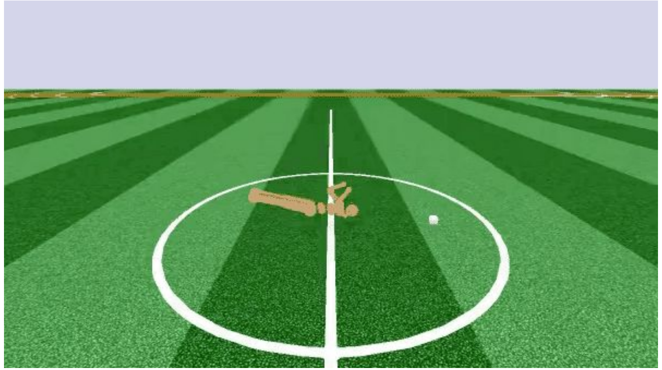
Jupyter Notebook with code for examples from this presentation



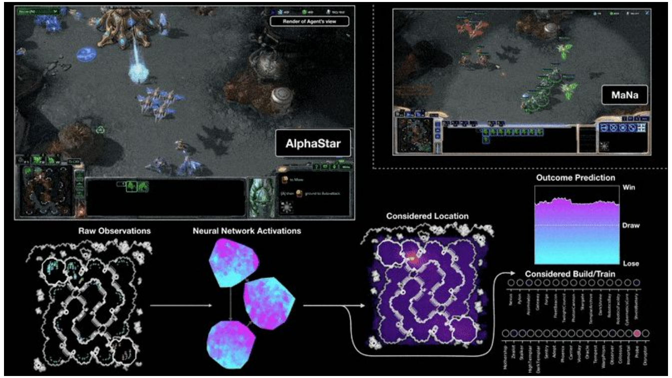
What can RL do?

Some examples

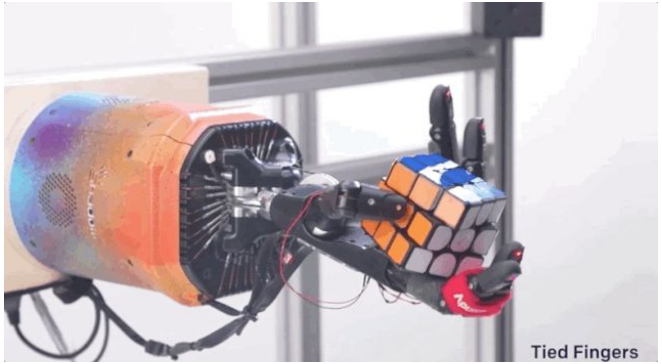
Control humanoid in simulation



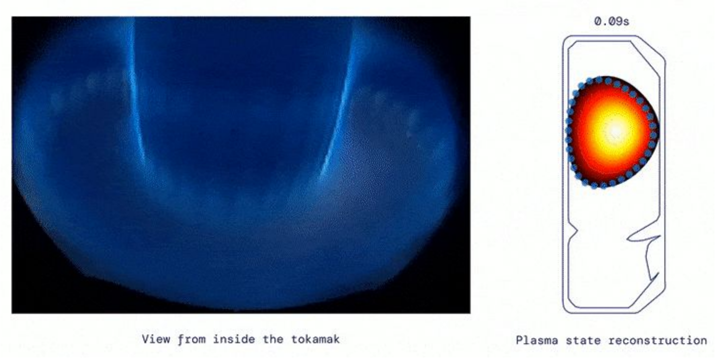
[Play games with imperfect information and develop long-term strategies]



[Control robot hands in the real world]

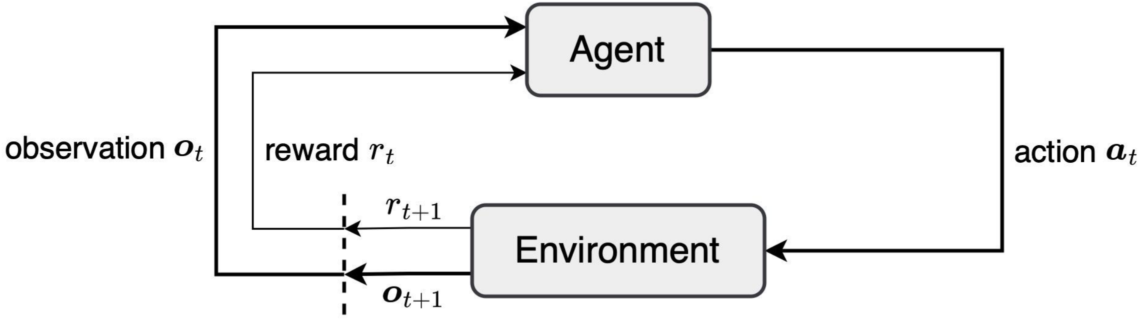


[Control the plasma in a tokamak fusion reactor]



Concepts of Reinforcement Learning

Some examples



- The **agent** (or **policy**) is the function we are trying to learn and tells us what to do to solve the task.
- The **environment** is the world that the RL agent lives in and defines the task.
- **Actions** are how the RL agent interacts with the environment.
- **Observations** are what the agent sees of the environment.
- The **reward** is returned by the environment after each action and describes the goodness of that action.
- The **return** is the cumulative reward over time. The goal of RL is to maximise the return.

Taxonomy of Reinforcement Learning

A brief overview

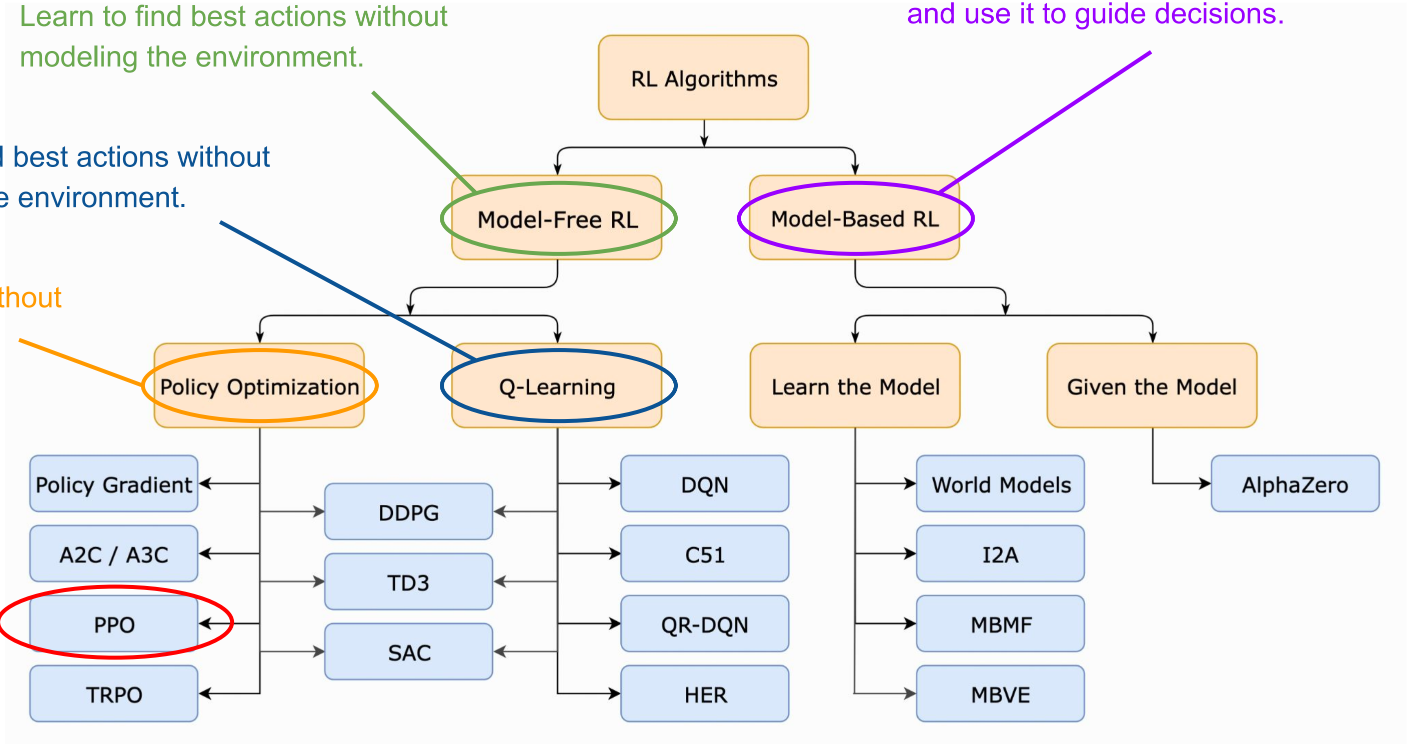
Use existing model or learn a model and use it to guide decisions.

Learn to find best actions without modeling the environment.

Learn to find best actions without modeling the environment.

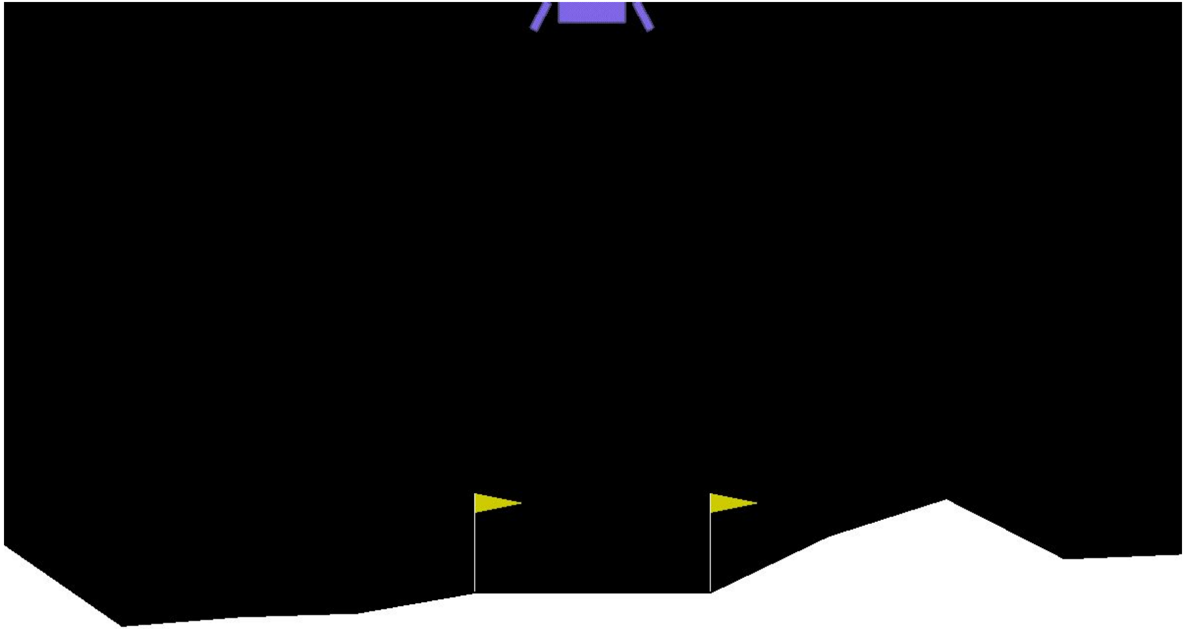
Learn to find best actions without modeling the environment.

Proximal Policy Optimisation



Lunar Lander Example

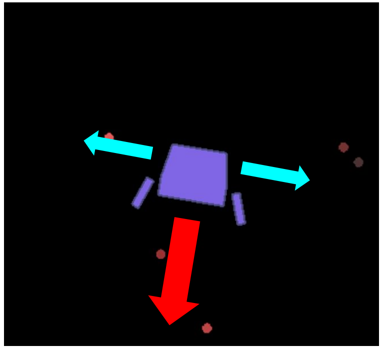
Introduction



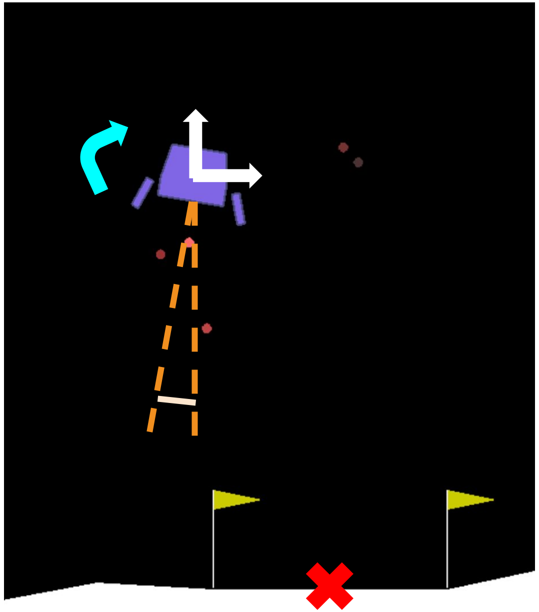
Lunar Lander Example

Actions and observations

Action

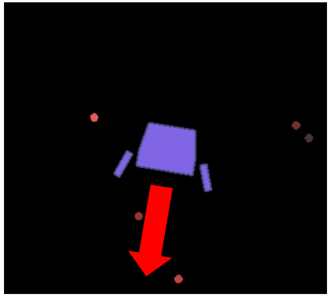


Observation

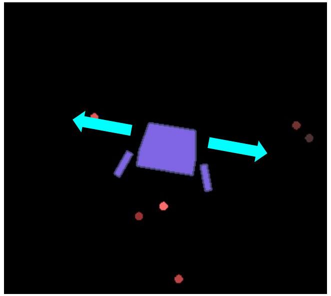


Lunar Lander Example

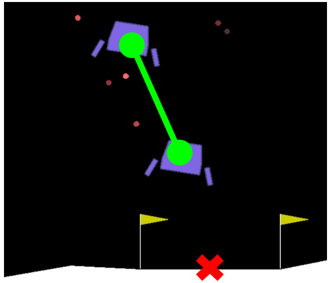
Rewards



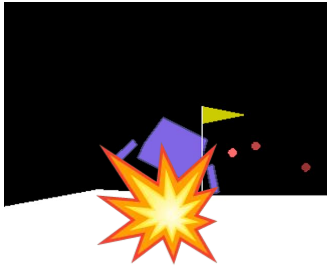
-0.3



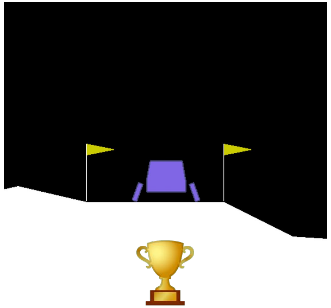
-0.03



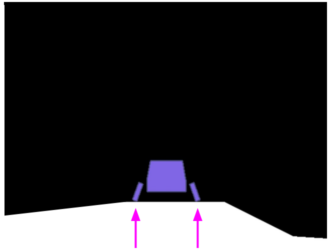
+/- d



-100



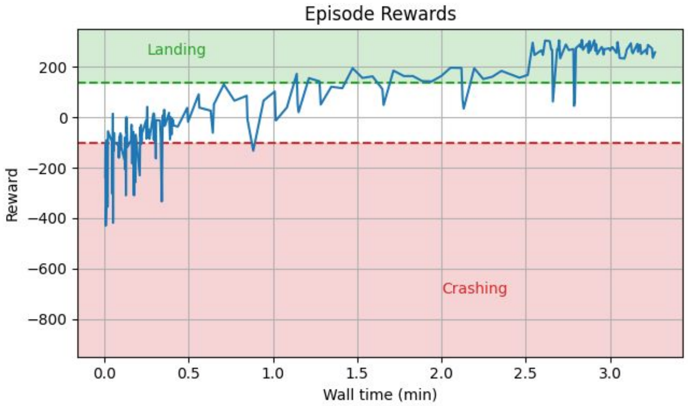
+200

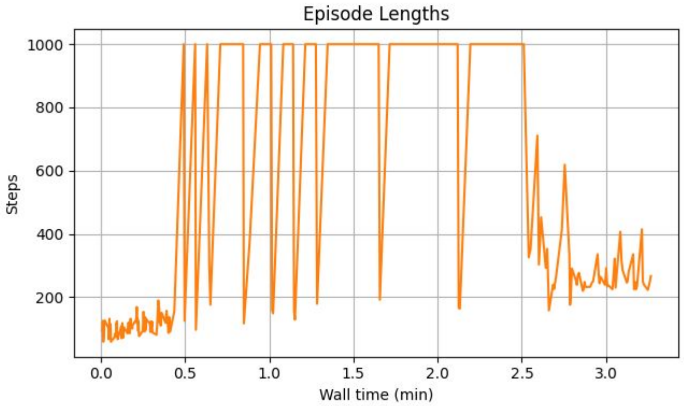


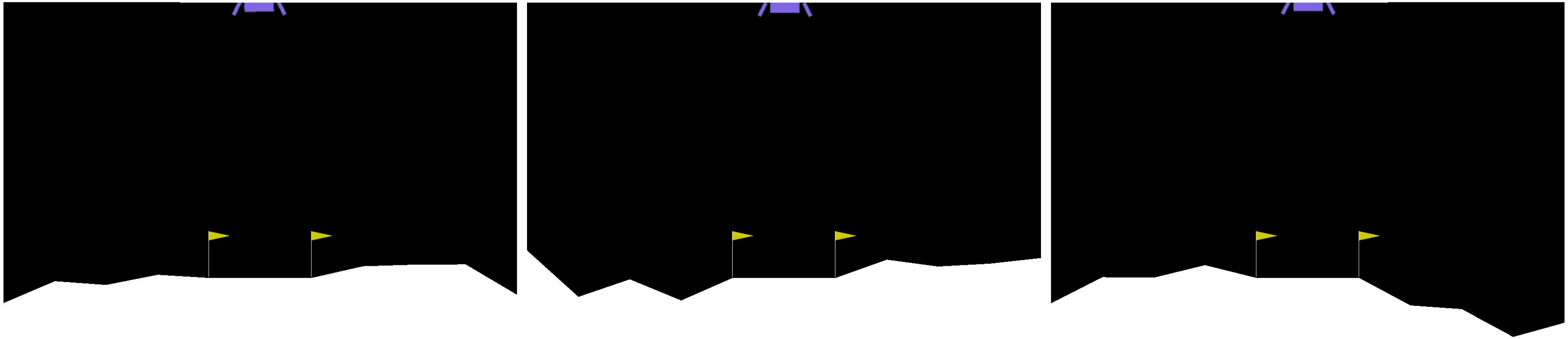
+10 each

Lunar Lander Example

Training results

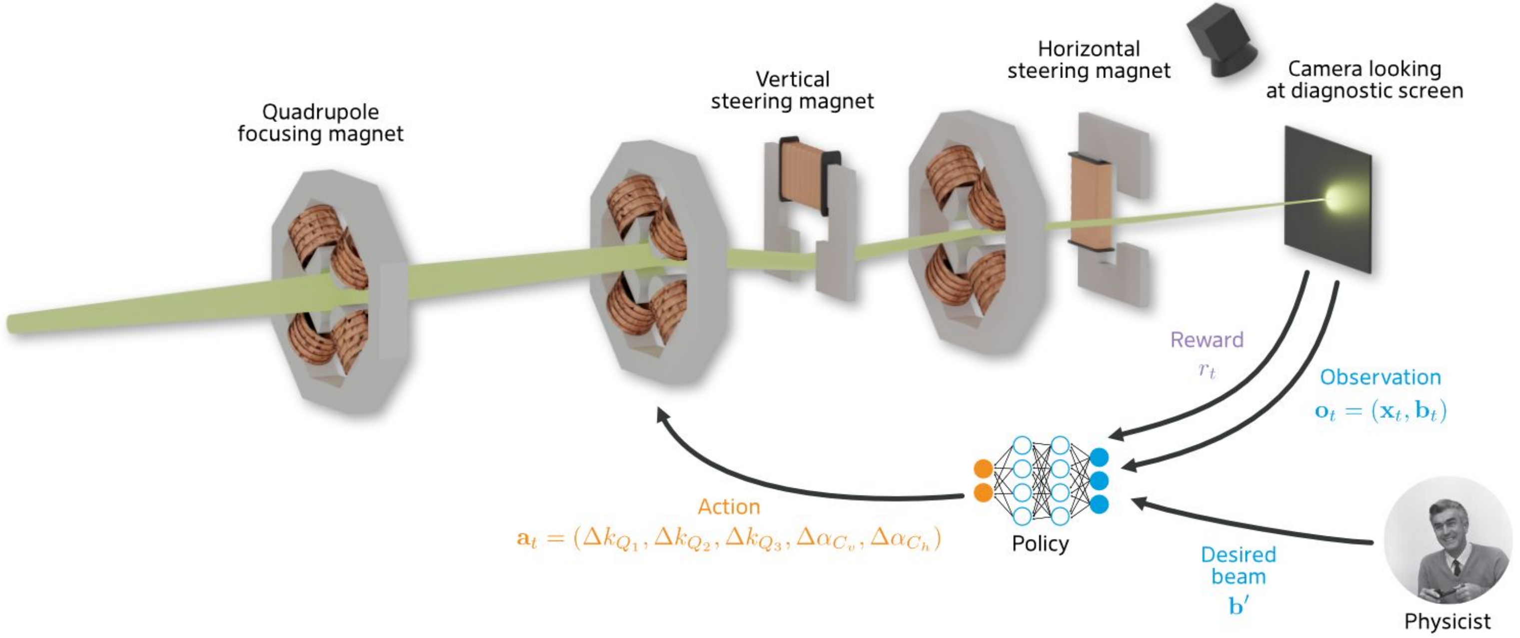






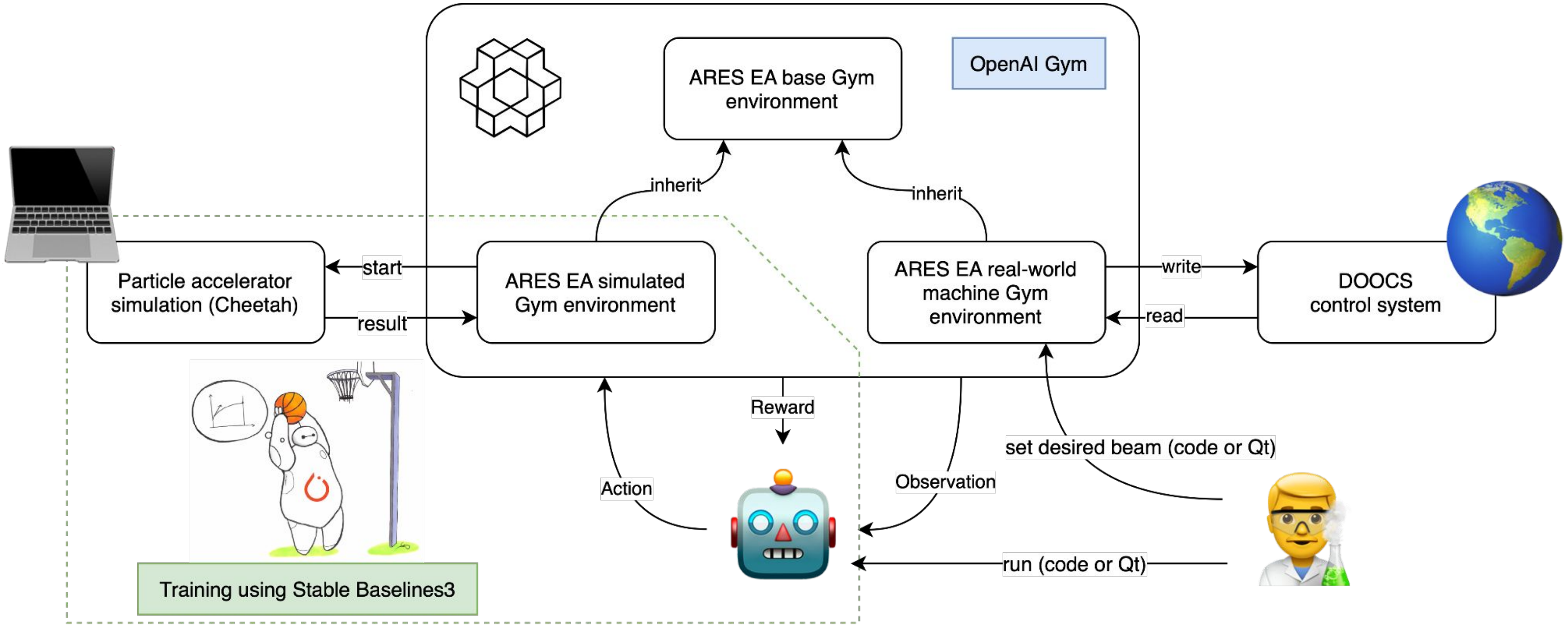
Accelerator Example

Positioning and focusing in the ARES Experimental Area



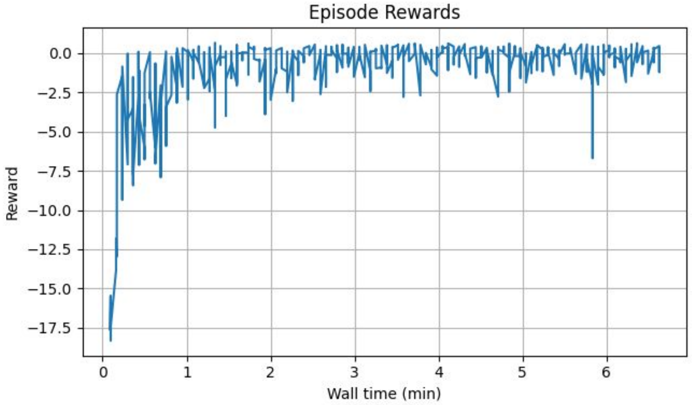
Accelerator Example

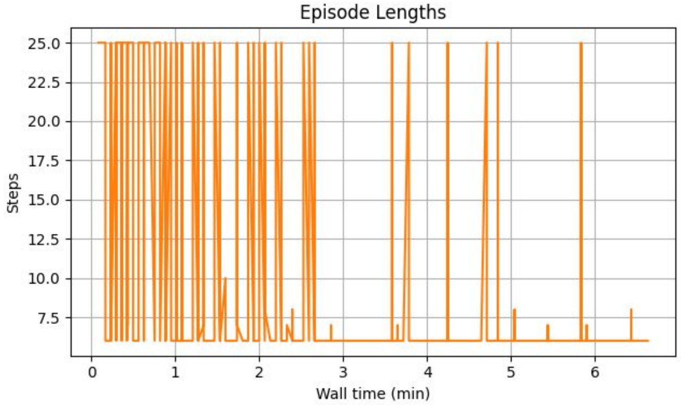
Technical overview

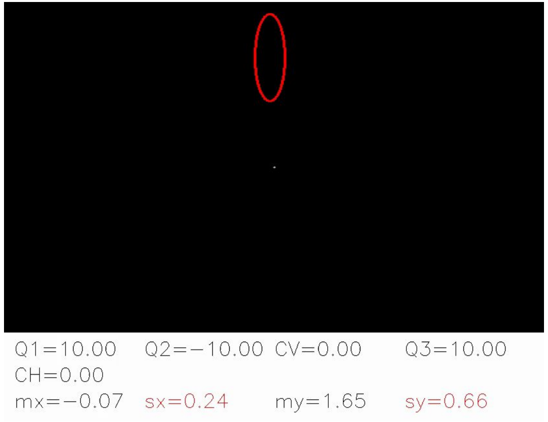


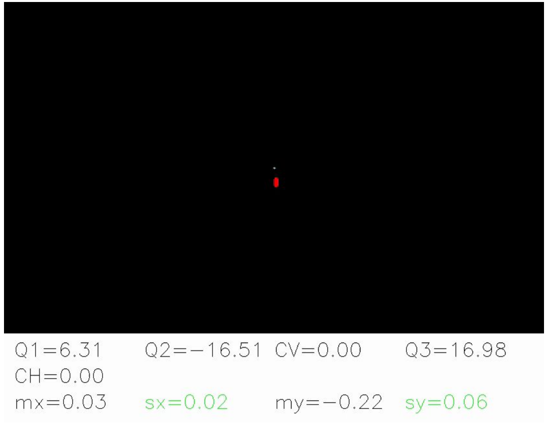
Accelerator Example

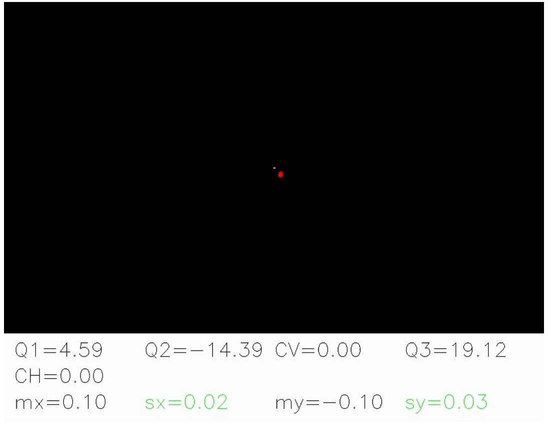
Training results





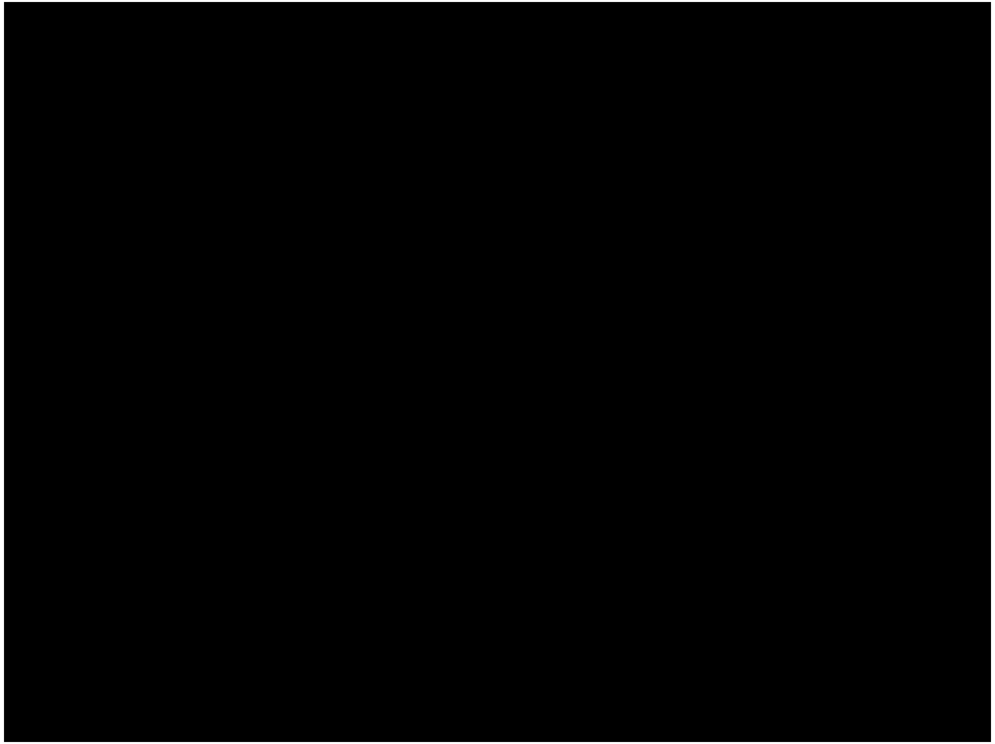






Accelerator Example

Running on the real accelerator



Accelerator Example

Getting it to work

Choosing rewards

- Make beam as small as possible (when reading screen)
  Squeeze beam into corner
- Minimise sum of pixels
  Push beam off-screen
- Get positive reward for each beam parameter while in threshold. After 5 steps in threshold give win bonus and stop.
  - **Briefly jump out of threshold after 4 steps**

$$R(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} \hat{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) & \text{if } \hat{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) > 0 \\ 2 \cdot \hat{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) & \text{otherwise.} \end{cases}$$

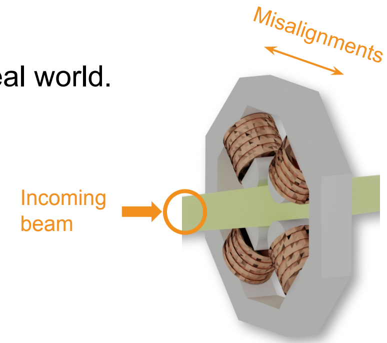
$$\hat{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) = O(\boldsymbol{x}_t) - O(\boldsymbol{x}_{t+1})$$

$$O(\boldsymbol{x}_t) = \ln \sum_{p \in \boldsymbol{b}_t, p' \in \boldsymbol{b}'} w_p |p - p'|$$

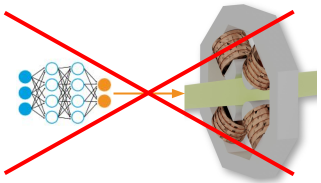
Sim2real transfer

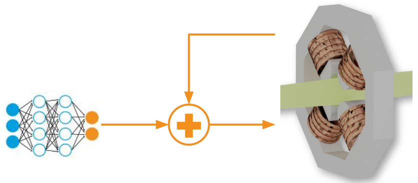
Getting RL to run on simulations is easy. Getting it to run on a real accelerator is hard.

<u>Domain randomisation</u>
The simulation is never quite like the real world.
-> Add random noise.



<u>Delta actions</u>
Magnet settings may be affected by noise.
-> Policy outputs changes to magnet settings.





Further Resources

Where to start if you want to get into reinforcement learning

Getting started in RL

- OpenAI Spinning Up - Very understandable explanations on RL and the most popular algorithms accompanied by easy-to-read Python implementations.
- Reinforcement Learning with Stable Baselines 3 - YouTube playlist giving a good introduction on RL using Stable Baselines3.
- Build a Doom AI Model with Python - Detailed 3h tutorial of applying RL using *DOOM* as an example.
- An introduction to Reinforcement Learning - Brief introduction to RL.
- An introduction to Policy Gradient methods - Deep Reinforcement Learning - Brief introduction to PPO.

Papers

- Learning-based optimisation of particle accelerators under partial observability without real-world training - Tuning of electron beam properties on a diagnostic screen using RL.
- Sample-efficient reinforcement learning for CERN accelerator control - Beam trajectory steering using RL with a focus on sample-efficient training.
- Autonomous control of a particle accelerator using deep reinforcement learning - Beam transport through a drift tube linac using RL.
- Basic reinforcement learning techniques to control the intensity of a seeded free-electron laser - RL-based laser alignment and drift recovery.
- Real-time artificial intelligence for accelerator control: A study at the Fermilab Booster - Regulation of a gradient magnet power supply using RL and real-time implementation of the trained agent using field-programmable gate arrays (FPGAs).
- Magnetic control of tokamak plasmas through deep reinforcement learning - Landmark paper on RL for controlling a real-world physical system (plasma in a tokamak fusion reactor).

Literature

- Reinforcement Learning: An Introduction - Standard text book on RL.

Packages

- Gym - Defacto standard for implementing custom environments. Also provides a library of RL tasks widely used for benchmarking.
- Stable Baslines3 - Provides reliable, benchmarked and easy-to-use implementations of the most important RL algorithms.
- Ray RLlib - Part of the *Ray* Python package providing implementations of various RL algorithms with a focus on distributed training.

Questions or remarks?

Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Jan Kaiser & Oliver Stein
Machine Beam Control (MSK)
jan.kaiser@desy.de | oliver.stein@desy.de