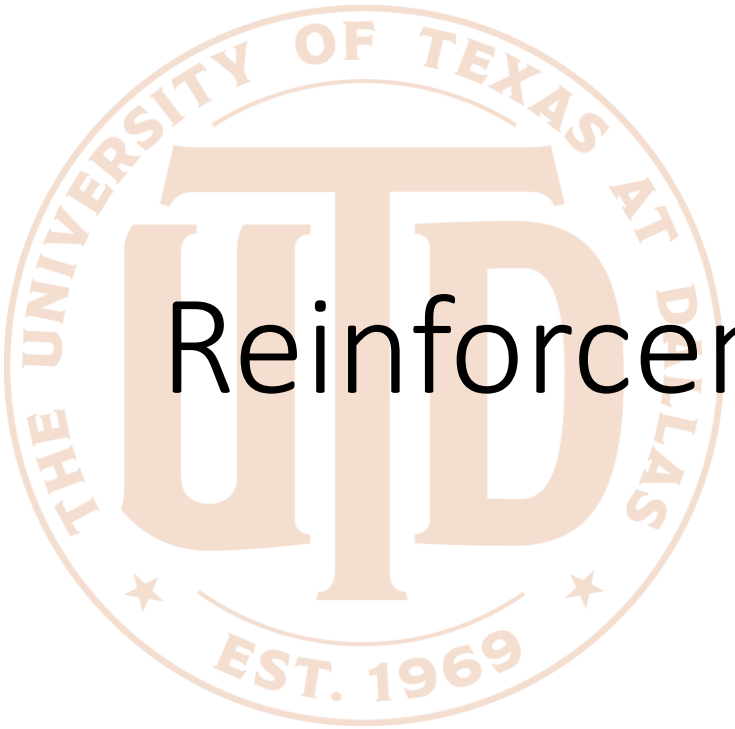


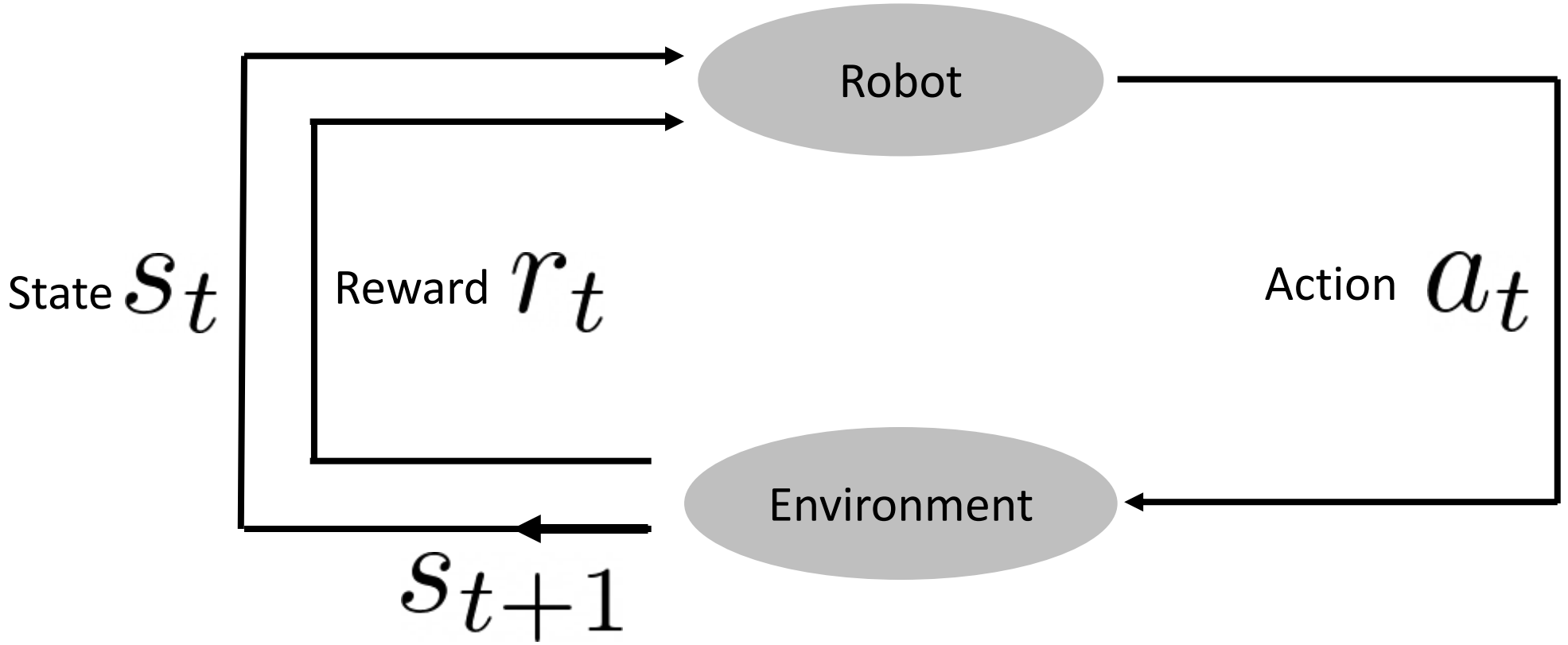
Reinforcement Learning: Overview and Foundations

CS 6341 Robotics

Professor Yu Xiang

The University of Texas at Dallas

Reinforcement Learning

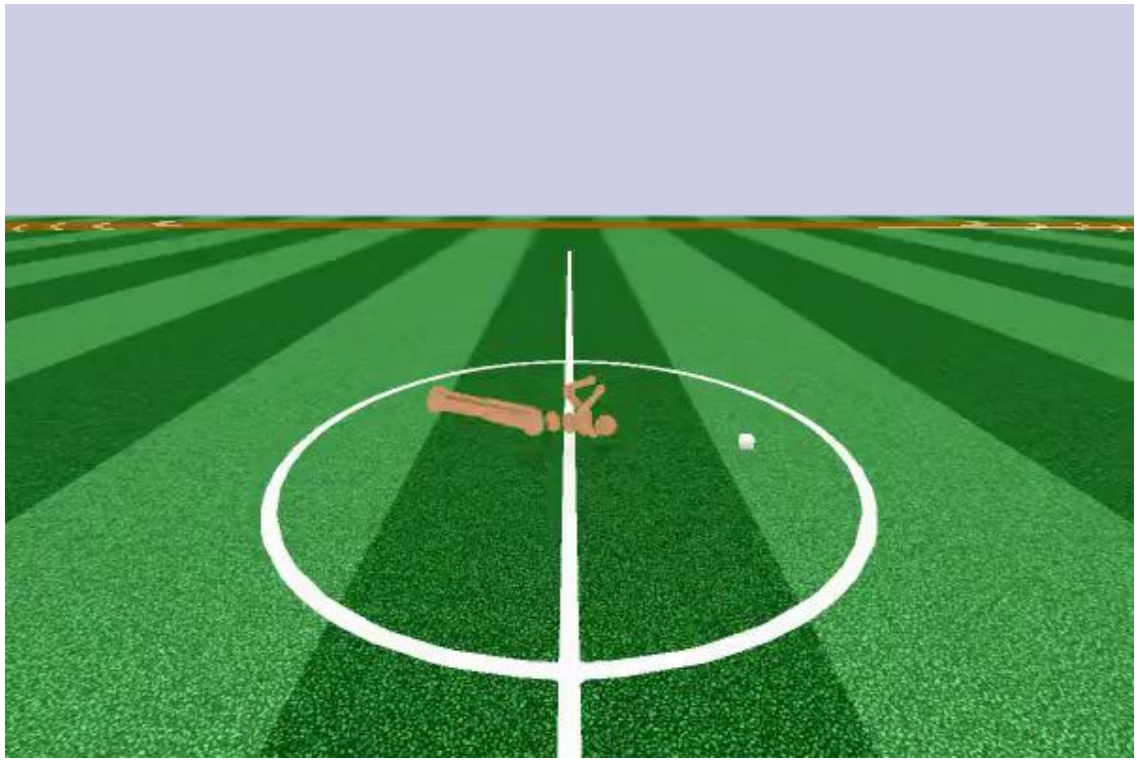


Reinforcement Learning:

Imitation Learning:

$$a_t = \pi(s_t)$$

RL Examples



Control

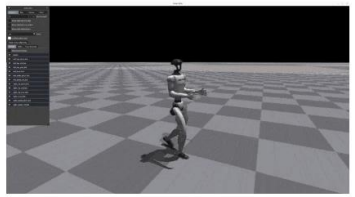
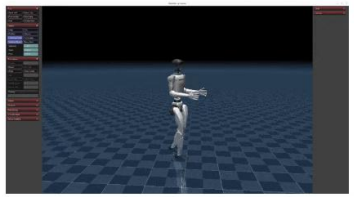
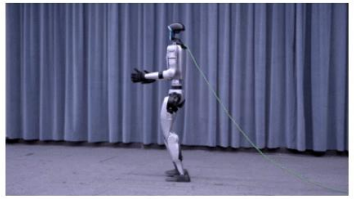
https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

RL Examples

Unitree RL GYM

🌎 English | CN中文

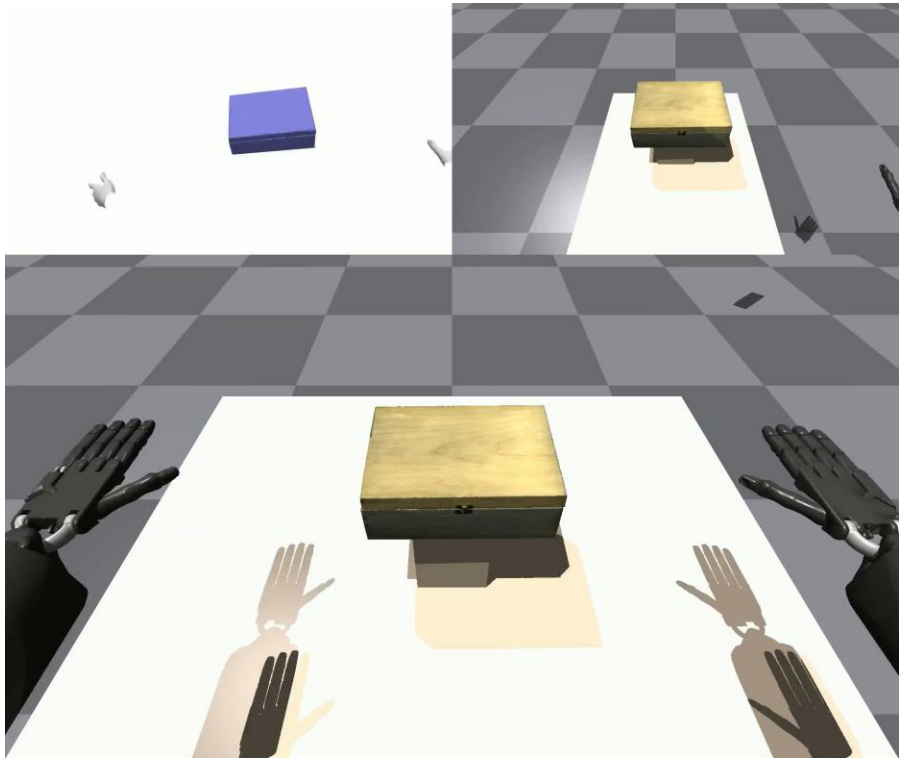
This is a repository for reinforcement learning implementation based on Unitree robots, supporting Unitree Go2, H1, H1_2, and G1.

Isaac Gym	Mujoco	Physical
		

https://github.com/unitreerobotics/unitree_rl_gym

Imitation learning & reinforcement learning

RL Examples



https://cypypccpy.github.io/obj-dex.github.io/

https://playground.mujoco.org/

RL Concepts

- State S : a complete description of the state of the world
- Observation O: partial description of a state
  - Fully observed vs. partially observed
  - For example: images

- Action space: the set of all valid actions in a given environment
  - Discrete action space vs. continuous action space a
- Policies: a policy is a rule used by an agent to decide what action to take
  - Deterministic policy $a_t = \mu(s_t)$
  - Stochastic policy $a_t \sim \pi(\cdot|s_t)$

RL Concepts

- Parameterized policies

$$a_t = \mu_\theta(s_t)$$

$$a_t \sim \pi_\theta(\cdot|s_t)$$

k dimensional action
- Joint position
- Gripper pose
- Joint velocity, etc.

  - Deterministic policy

```
pi_net = nn.Sequential(
                nn.Linear(obs_dim, 64),
                nn.Tanh(),
                nn.Linear(64, 64),
                nn.Tanh(),
                nn.Linear(64, act_dim)
              )
```

  - Stochastic policy
    - Categorical policy for discrete actions

$$\log \pi_\theta(a|s) = \log [P_\theta(s)]_a$$

    - Diagonal Gaussian policy: mean action $\mu_\theta(s)$

Log standard deviation $\log \sigma_\theta(s)$ $(-\infty, \infty)$

RL Concepts

- Diagonal Gaussian policy
  - Sampling $a = \mu_\theta(s) + \sigma_\theta(s) \odot z$ $z \sim \mathcal{N}(0, I)$ k dimensional action

$$p(x) = \frac{1}{(2\pi)^{k/2} \prod_{i=1}^{k} \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^{k} \frac{(x_i - \mu_i)^2}{\sigma_i^2}\right) \qquad \log p(x) = -\frac{1}{2} \sum_{i=1}^{k} \left[\frac{(x_i - \mu_i)^2}{\sigma_i^2} + \log(2\pi\sigma_i^2)\right]$$

  - Log-likelihood

$$\log \pi_\theta(a|s) = -\frac{1}{2}\left(\sum_{i=1}^{k}\left(\frac{(a_i - \mu_i)^2}{\sigma_i^2} + 2\log\sigma_i\right) + k\log 2\pi\right)$$

RL Concepts

- A Trajectory is a sequence of states and actions in the world

$$\tau = (s_0, a_0, s_1, a_1, ...)$$

- Start-state distribution $s_0 \sim \rho_0(\cdot)$
- State transitions are governed by natural laws of the environment (dynamics)
  - Deterministic $s_{t+1} = f(s_t, a_t)$
  - Stochastic $s_{t+1} \sim P(\cdot|s_t, a_t)$

RL Concepts

- Reward function $r_t = R(s_t, a_t, s_{t+1})$ simplified $r_t = R(s_t)$ $r_t = R(s_t, a_t)$

- Finite-horizon undiscounted return for a trajectory

$$R(\tau) = \sum_{t=0}^{T} r_t$$

- Infinite-horizon discounted return for a trajectory

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \qquad \gamma \in (0, 1)$$

The RL Problem

- The goal of RL is to select a policy which maximizes expected return when the agent acts according to it
- Probability distribution over trajectories

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$$

- Expected return

$$J(\pi) = \int_{\tau} P(\tau|\pi)R(\tau) = \underset{\tau\sim\pi}{\mathrm{E}}\left[R(\tau)\right]$$

Sample trajectories

- The central optimization problem

$$\pi^* = \arg\max_{\pi} J(\pi)$$

Optimal policy

In practice

$$\pi_\theta^* = \arg\max_{\theta} J(\pi_\theta)$$

Learn the parameters of the policy

Summary

- RL concepts

Further Reading

- OpenAI Spinning Up in Deep RL
  https://spinningup.openai.com/en/latest/index.html