Topics in RL: Project 2

April 3, 2025

1 Introduction

This document describes various custom reinforcement learning environments that can be created as needed, along with suitable deep reinforcement learning (DRL) algorithms for each task.

2 UAV Navigation Through a Window (Team 8)

2.1 State and Action Space

- **State:** 3D position and 3D velocity of the UAV.
- **Action:** 3D orientation of the UAV.

2.2 Recommended DRL Algorithm

Advantage Actor-Critic (A2C), A3C, and Soft Actor-Critic (SAC) for continuous control.

Objectives:

- Implement A2C, A3C, and SAC for training RL agents.
- Train and test policies on UAV navigation tasks.
- Compare the reward curves and performance of the algorithms.

Expected Outcomes:

- A comparison of RL algorithms in terms of training stability and success rate.
- Reward curves showing performance differences across tasks.

3 UAV Tracking a Moving Ground Object (Team 1)

3.1 State and Action Space

- **State:** 3D position and 3D velocity of the UAV.
- **Action:** 3D orientation of the UAV.
- **Ground Object Trajectories:**
  - Straight line.
  - Circular.
  - Figure-eight shape.

3.2 Recommended DRL Algorithm

Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Asynchronous Advantage Actor-Critic (A3C) for smooth trajectory tracking.

Objectives:

- Implement DDPG, TD3, and A3C for tracking moving objects.
- Train and evaluate policies on different ground object trajectories.
- Analyze trajectory tracking accuracy and control stability.

Expected Outcomes:

- Performance evaluation of RL algorithms on different movement patterns.
- Reward curves and trajectory-following accuracy comparison.

4 Fetch Reach and Push (Team 7)

More details can be found at the official documentation: `https://robotics.farama.org/envs/fetch/push/`

4.1 Recommended DRL Algorithm

Hindsight Experience Replay (HER) combined with DDPG, SAC for goal-conditioned reinforcement learning.

Objectives:

- Implement TD3, DDPG, SAC for training RL agents.
- Combine HER for goal relabeling with above RL algorithms for better sample efficiency.

- Train and test policies on FetchReach and FetchPush environments.
- Compare the reward curves and performance of all algorithms.

Expected Outcomes:

- A comparison of RL algorithms in terms of training stability and success rate.
- Reward curves showing performance differences across tasks.

5 Solving a Rubik's Cube (Team 6)

An approach to solving a Rubik's Cube using reinforcement learning is detailed in the following article: Solving a Rubik's Cube with Reinforcement Learning

5.1 Recommended DRL Algorithm

Deep Q-Networks (DQN) with curriculum learning and AlphaZero-style Monte Carlo Tree Search (MCTS) for structured problem solving.

Objectives:

- Implement DQN and MCTS for solving the Rubik's Cube.
- Train policies with curriculum learning to improve efficiency.
- Compare different reinforcement learning approaches to structured problem solving.

Expected Outcomes:

- Comparative analysis of different RL approaches in solving the Rubik's Cube.
- Training curves showing improvement in solving efficiency.

6 Imitation Learning for UAV Obstacle Avoidance (Team 2)

The UAV learns to avoid multiple obstacles by imitating the velocity obstacle algorithm document.

6.1 Recommended DRL Algorithm

Behavior Cloning (BC) approaches:

- Supervised Behavior Cloning using expert demonstrations. Ideally a diffusion policy/ Transformer based policy.

- Dataset Aggregation (DAgger) for iterative improvement of policies.

Objectives:

- Implement BC and DAgger for UAV obstacle avoidance.
- Train UAVs using expert demonstrations.
- Evaluate success rates in avoiding obstacles.

Expected Outcomes:

- Comparison of BC and DAgger in terms of performance and generalization.
- Success rates and trajectory optimization for UAV obstacle avoidance.

7 Call Options : (Team 4)

An investor holds a call option that grants the right to purchase one share of a stock at a fixed price p. The option is valid for T days, during which the investor makes a decision at the beginning of each day.

If the investor chooses not to exercise the option, they retain the right to do so on subsequent days. However, if the option is exercised when the stock price is s, the investor effectively gains $s - p$.

Assume that the stock price evolves with independent increments, meaning that the price on day $t + 1$ is given by:

$$S_{t+1} = S_t + W_t$$

Assume that $\{W_t\}_{t\geq 1}$ is an i.i.d. process.

Your task is to create an agent that decides when to exercise (or not exercise) the call option within the given span of T days. Assume $p \in \mathbb{N}$ and that W_t is sampled from some continuous distribution of your choice.

Objectives:

- Implement 3–4 Deep RL algorithms from the Deep RL presentation list.
- Ideally, you should select at most 1 or 2 algorithms from your own presentation, with the remaining algorithms chosen from the list.

8 Inventory Management Problem: (Team 5)

Retail stores stockpile products in warehouses to meet random demand. Additional stocks are procured at regular intervals. Let X_t denote the amount of stock before the t-th procurement. In this example, time represents the number of additional stock procurements.

At time t, the store may procure an additional stock U_t (where $U_t \leq U$) units at a price p per unit. Thus, the total procurement cost is pU_t.

The random demand W_t is i.i.d. with distribution P_W. The stock available at the next time step is given by:

$$X_{t+1} = X_t + U_t - W_t,$$

where a negative stock value represents backlogged demand.

The holding cost for the stock is given by $h(x)$, where a is the per-unit storage cost and b is the per-unit backlog cost.

The per-stage cost is:

$$c(X_{t+1}, U_t) = h(X_{t+1}) + pU_t.$$

The objective is to find the optimal inventory control strategy to minimize the expected total cost over a finite horizon.

Parameters: The demand distribution P_W should **not be limited to uniform**. You must assume different distributions and justify your choices.

Objectives:

- Implement 3–4 Deep RL algorithms from the Deep RL presentation list.
- Ideally, you should select at most 1 or 2 algorithms from your own presentation, with the remaining algorithms chosen from the list.
- Implement the problem for **both continuous and discrete state-action spaces**:
  - Use at least **one algorithm for the continuous state-action version** by assuming inventory is fluid.
  - Use at least **one algorithm for the discrete state-action version** of the problem.

9 Portfolio Optimization: (Team 3)

Consider the portfolio optimization problem discussed at this link. You will need to optimize a portfolio consisting of n assets. This problem is covered in Chapter 3.

In the illustrated examples, only three weights, $\{-1, 0, 1\}$, are considered. However, you may also include additional weights, such as -0.5 and 0.5, in your optimization.

Objectives:

- Implement 3–4 Deep RL algorithms from the Deep RL presentation list.
- Ideally, you should select at most 1 or 2 algorithms from your own presentation, with the remaining algorithms chosen from the list.
- Implement the problem using a **continuous state space and continuous action space**.

Additional Instructions

1.) Teams working on Project 7, 8, and 9 must meet Tejas Sir within a week of this document being released for further clarifications on the problem.

2.) For drone-related projects, please meet HariKumar Sir for clarifications regarding the simulation setup.