Assignment 1: Policy Iteration and iLQR

Reinforcement Learning - A.Y. 2025/2026

Assigned: October 21th, 2025
Deadline: November 5th, 2025

Rules

The assignment is due on November 5th, 2025. Students may discuss assignments, but **each student must code up and write up their solutions independently**. **Students must also indicate on each homework the names of the colleagues they collaborated with** and what online resources they used.

The theory solutions must be submitted in a pdf file named "XXXXXXX.pdf", where XXXXXXX is your matricula. We encourage you to type the equations on an editor rather than uploading a scanned written solution. **In the pdf** you have to hand over **the answers to the theory questions (not just the numerical results, but also the derivations)** and a **small report of the practice exercises**.

The practice exercises must be uploaded in a zip file named "XXXXXXX.zip", where XXXXXXX is your matricula. **The zip file must have the same structure of the assignment.zip** that you find in the attachments, but with the correct solution. You are only allowed to type your code in the files named "student.py". Any modification to the other files will result in penalization. You are not allowed to use any other python library that is not present in python or in the "requirements.txt" file. You can use as many functions you need inside the "student.py" file. The zip file must have the same structure of the "assignment1.zip" (see below).

All the questions must be asked in the Classroom platform but it is forbidden to share the solutions on every forum or on Classroom.

Things NOT to do and will result in penalization:

- do not modify the signature of the functions you have to complete (name, parameters and default values); also, do not overwrite the values of the parameters passed to the function you have to complete
- do not modify the name of the file student.py
- do not add import lines

Of course, you can change the code however you want while developing your solutions, but it is important that the zip you submit follows the above rules.

Theory

1. Demonstrate the convergence of the Policy Iteration Algorithm.
2. Given the following environment settings
   - States: $\{s_i : \ i \in \{1...5\}\}$
   - Reward:
   $$r(s, a, s') = \begin{cases} 1 & \text{if } s = s_3 \\ -10 & \text{if } s = s_5 \\ 0 & \text{otherwise} \end{cases}$$
   - Dynamics: $p(s_4|s_4, a_1) = 0.8$, $p(s_5|s_4, a_1) = 0.2$
   - Policy: $\pi(s) = a_1 \ \ \forall s \in S$

   and the value function at the iteration $k = 1$:
   $$v_k = [0, \ 0, \ 1, \ 0, \ -10]$$
   with $\gamma = 0.8$.

   Compute $V_{k+1}(s_4)$ following the *Value Iteration* algorithm.

Practice

1. Implement the policy iteration algorithm on the modified version of the FrozenLake Gymnasium environment.

   In folder "policy_iteration" you find three files:
   - "frozenlake_custom.py" contains the custom implementation of the FrozenLake environment.
   - "main.py" contains the python script to run the tests
   - "student.py" contains the function "policy_iteration" that you have to fill in with the needed code to implement the policy iteration algorithm. You will also find the value_iteration function that we have seen in class so you can try to play with it and compare the results.

   Note: you can visualize the episodes by compiling with argument - -*render*.
2. Implement the iLQR algorithm for solving the Pendulum environment. Specifics of the environment are available at `https://gymnasium.farama.org/environments/classic_control/pendulum/`. In folder "ilqr" you find two files:
   - "main.py", that contains the python script to run the tests.
   - "student.py" contains the class implementing iLQR with the functions that you will need to fill in.

   In the *pendulum_dyn* function, you need to implement the equations for the pendulum dynamics:
   $$\dot{\theta}_{t+1} = \dot{\theta}_t + \left(\frac{3g}{2l} \sin\theta + \frac{3.0}{ml^2} u \right) dt$$

$$\theta_{t+1} = \theta + \dot{\theta}_{t+1}\ dt$$

In the *backward* function, you will need to implement the formulas to compute the P and K matrices from the LQR-LTV algorithm. In this problem, there are also additional terms coming from the cost's linearization, performed using the 2nd order Taylor expansion. These terms can be computed using the following formulas:

$$k_t = -(R_t + B_t^T P_{t+1} B_t)^{-1}(r_t + B_t^T p_{t+1})$$

$$p_t = q_t + K^T(R_t k_t + r_t) + (A_t + B_t K)^T p_t + (A_t + B_t K)^T P_{t+1} B_t k_t$$

Finally, in the *forward* function, you will need to use the K and k matrices to update the control in the following way:

$$control = k_t + K_t(x_t^i - x_t^{i-1})$$