ROB6323: Reinforcement learning and optimal control for robotics

Exercise series 3

For questions requesting a written answer, please provide a detailed explanation and typeset answers (e.g. using LaTeX[1]). Include plots where requested in the answers (or in a Jupyter notebook where relevant). For questions requesting a software implementation, please provide your code in runnable Jupyter Notebook. Include comments explaining how the functions work and how the code should be run if necessary. Code that does not run out of the box will be considered invalid.

Exercise 1 [40 points]

a) Consider the following dynamical system

$$x_{n+1} = \begin{cases} -x_n + 1 + u_n & \text{if } -2 \leq -x_n + 1 + u_n \leq 2 \\ 2 & \text{if } -x_n + 1 + u_n > 2 \\ -2 & \text{else} \end{cases}$$

where $x_n \in \{-2, -1, 0, 1, 2\}$ and $u_n \in \{-1, 0, 1\}$, and the cost function

$$J = \left(\sum_{k=0}^{2} 2|x_k| + |u_k| \right) + x_3^2 \tag{1}$$

Use the dynamic programming algorithm to solve the finite horizon optimal control problem that minimizes J. Show the different steps of the algorithms and present the results in a table including the cost to go and the optimal control at every stage.

b) What is the sequence of control actions, states and the optimal cost if $x_0 = 0$, if $x_0 = -2$ and if $x_0 = 2$.

c) Assume now that the constant term 1 in the previous dynamics can sometimes be 0 with probability 0.4. We can now write the dynamics as

$$x_{n+1} = \begin{cases} -x_n + \omega_n + u_n & \text{if } -2 \leq -x_n + \omega_n + u_n \leq 2 \\ 2 & \text{if } -x_n + \omega_n + u_n > 2 \\ -2 & \text{else} \end{cases}$$

where $x_n \in \{-2, -1, 0, 1, 2\}$, $u_n \in \{-1, 0, 1\}$ and $\omega_n \in \{0, 1\}$ is a random variable with probability distribution $p(\omega_n = 0) = 0.4$, $p(\omega_n = 1) = 0.6$. The cost function to minimize becomes

$$J = \mathbb{E}\left(\left(\sum_{k=0}^{2} 2|x_k| + |u_k| \right) + x_3^2 \right) \tag{2}$$

Use the dynamic programming algorithm to solve the finite horizon optimal control problem that minimizes J. Show the different steps of the algorithms and present the results in a table including the cost to go and the optimal control at every stage.

[1] 1https://en.wikibooks.org/wiki/LaTeX, NYU provides access to Overleaf to all the community https://www.overleaf.com/edu/nyu

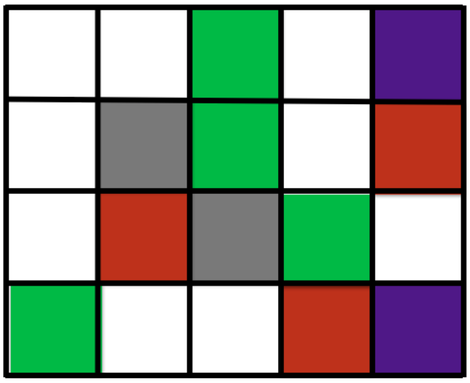

Figure 1: *Grid world*

d) Compare the costs and optimal control from the deterministic and probabilistic models and explain where, in your opinion, the differences come from.

Exercise 2 [60 points]

Consider the grid-world shown in Figure 1. In each (non grey) cell, it is possible to perform five actions: move up, down, left, right or do nothing as long as the resulting move stays inside the grid world. Grey cells are obstacles and are not allowed. We would like to find the optimal value function and optimal policy that minimize the following cost

$$\min \sum_{n}^{\infty} \alpha^n g_n(x_n)$$

with discount factor $\alpha = 0.99$ and where the instantaneous cost is defined as

$$g_n(x_n) = \begin{cases} -1 & \text{if } x_n \text{ is a violet cell} \\ 0 & \text{if } x_n \text{ is a white cell} \\ 1 & \text{if } x_n \text{ is a green cell} \\ 10 & \text{if } x_n \text{ is a red cell} \end{cases}$$

In a Jupyter notebook answer the following questions:

a) Implement the value iteration algorithm to solve the problem (initialize the value function to 0). How many iterations does it take to attain convergence? (we assume here that convergence happens when all the elements of the value function do not change more than 10^{-6} in a new iteration.

b) Implement the policy iteration algorithm to solve the problem (use the version that solves the linear equation $I - \alpha A)J_\mu = \bar{g}$). Start with an initial policy that does not move. How many iterations does it take to converge?

c) Compare the solutions and convergence/complexity of each algorithm to solve this problem.