ESE 680 - Dynamic Programming and Optimal control - Homework 1

Due: Mon Sept 25th

All exercises have equal weights. If you use any kind of resource, you have to cite it. You may collaborate, but you have to say with whom, and write up the solutions independently on your own.

Exercise 1 You are playing the game of tic-tac-toe against a not so good player. You are playing first. The opponent plays next and chooses uniformly randomly from all available positions. Then you play next, the opponent randomizes, and so on, until either you win, or you lose, or you tie. Suppose your goal is to find your policy to maximize the probability of winning, and you are indifferent between tie or losing.

1. What is the maximum probability of winning? Formulate this as a finite-horizon dynamic programming problem and solve it by Matlab. Roughly, the number of states is all possible board arrangements, i.e., any of the 3×3 positions could be empty, occupied by you, or by the opponent, which makes 3^9 combinations. (Some of them will never occur but you do not have to go into the trouble of excluding them). The horizon can be 10, i.e., at most 9 plays for you and the opponent and at time 10 the outcome is decided.

2. Suppose now you care about losing too, so your goal is to maximize the probability of winning minus the probability of losing. What is the optimal such value? Is the optimal policy the same as before?

3. Suppose now the opponent plays first, and you play second. What are the optimal values for the two above questions?

Exercise 2 Tour robot tries to catch an adversary robot. At time k your robot is at a position $y_k \in \mathbb{R}^2$ in the two-dimensional plane, you choose a direction $u_k \in \mathbb{R}^2$ to move, and your next position is $y_{k+1} = y_k + u_k$. At the same time, the adversary robot tries to avoid you. It starts from a position $z_k \in \mathbb{R}^2$ and moves at a direction away from you at the new point $z_{k+1} = z_k + 0.2(z_k - x_k) + w_k$, with w_k being a 2-d standard Gaussian noise. The horizon is $N = 10$ and at any time step $k = 0, \dots, N-1$ the cost is your distance from the adversary plus a penalty on your speed, i.e, $c_k = \|y_k - z_k\|^2 + \|u_k\|^2$, and the terminal cost is $c_N = \|y_N - z_N\|^2$.

1. This is a linear quadratic problem. Write the corresponding matrices A, B, Q, R, W.

2. Consider using the (deterministic and time invariant) policy π that at any time step moves your robot to the adversary's position, that is $u_k = z_k - y_k$. Write a Matlab code that performs policy evaluation, i.e., given any initial positions y_0, z_0 computes the cost of $J_\pi(y_0, z_0)$ of this policy.

3. Confirm your result by simulation. Suppose $y_0 = [0,0], z_0 = [1,1]$. Run many simulations of the above policy, compute the cost encountered during each simulation, and take the average over simulations. (This is a Monte Carlo Policy Evaluation) Does the result agree with the result given using the code of the policy evaluation? Qualitatively, how many simulations did you need to take?

4. Write a Matlab code that finds the optimal policy, and given any initial positions y_0, z_0 computes the optimal cost of $J^*(y_0, z_0)$. Confirm again your result by simulation as before.

5. Prove that the optimal policy is actually of the form $u_k^* = r_k(z_k - y_k)$ for some scalar value r_k

6. If you increase the time horizon $N \to \infty$, does the optimal policy converge to something? Also, does the optimal cost function $J^*(y_0, z_0)$ converge?

Exercise 3 You are controlling a linear system by selecting its modes of operation. For two modes, that is,

$$x_{k+1} = \begin{cases} A_1 x_k, & \text{if } a_k = 1, \\ A_2 x_k, & \text{if } a_k = 2 \end{cases} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the current state, and $A_1 \in \mathbb{R}^{n\times n}, A_2 \in \mathbb{R}^{n\times n}$ are two given matrices. Your costs at times $k = 0, \dots, N$ are $c_k = x_k^T Q x_k$ for some positive definite matrix Q. This setup has applications for example in sensor scheduling, or in embedded control systems with limited computation and communication resources.

1. Use dynamic programming to show that the optimal cost-to-go/value functions $J^*_{N-k}(x)$ of the problem are the minimum of 2^k positive definite quadratic functions. Describe the optimal policy $a^*_{N-k}(x)$ using these functions.

2. Consider the $N = 4$ horizon problem with matrices $A_1 = \begin{bmatrix} 0.5 & -1 \\ 1 & 0.5 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$ and Q the identity matrix. Plot the regions of the state space where it is optimal to select action 1 and 2 respectively at time $k = 0, \dots, 3$, for example on a discretized $[-1,1] \times [-1,1]$ square around the origin. Also plot the optimal value function $J_0^*(x)$ on that space.

3. The problem becomes hard for a large horizon, and also for many modes. This motivates an approximately optimal solution, a rollout policy. First suppose a given open-loop base policy which consists of a sequence of selected modes (independent of the state), for example the round-robin base policy $\pi_b = \{1, 2, 1, 2, \dots\}$. What is the form of the resulting closed-loop rollout policy?

4. For the above numerical case, on the same plot where you have the optimal value function, plot the value function of the rollout policy (note that we have to do this by simulation, we do not have a closed form expression for this) as well as the value function of the base policy. Compare them