# AE 504: HW3

Linyi Hou

April 12, 2020

## Part I - Theoretical Work

**T1.** We solve the problem by propagating backward from the end state $S_N$. At $S_N$, let the rewards $\mathcal{R}_{N,i}$ corresponding to states $\mathcal{S} = \{s^1, s^2, s^3, s^4, s^5\}$ equal $-\infty$ except $s^2$, where $\mathcal{R}_{N,2} = 0$ to represent the desired end position.

For all $S_{N-1} \in \mathcal{S}$, we use the transition table provided to determine the total reward of transitioning from any $S_{N-1}$ to any $S_N$ plus the reward available at $S_N$, and record the maximum reward possible for each $S_{N-1}$ as $\mathcal{R}_{N-1,i}$, where $i = 1, ..., 5$. The optimal control policy for each $S_{N-1}$ is the transition that maximizes $\mathcal{R}_{N-1,i}$.

Repeat the above process until $S_0$, which is the beginning state. Here we enforce the starting position $s^1$ by forcing all transitions **not** starting from $s^1$ to have reward $-\infty$.

The above algorithm was implemented in MATLAB. The optimal control policy was determined for $N \geq 3$ as the following:

$$
\begin{aligned}
N = 3: &\quad s^1 \xrightarrow{a^2} s^5 \xrightarrow{a^1} s^2 \\
N = 4: &\quad s^1 \xrightarrow{a^2} s^5 \xrightarrow{a^2} s^4 \xrightarrow{a^1} s^2 \\
N = 5: &\quad s^1 \xrightarrow{a^2} s^5 \xrightarrow{a^2} s^4 \xrightarrow{a^2} s^4 \xrightarrow{a^1} s^2 \\
N = 6+: &\quad s^1 \xrightarrow{a^2} s^5 \underbrace{\xrightarrow{a^2} s^4}_{\text{repeat } N-3 \text{ times}} \xrightarrow{a^1} s^2
\end{aligned}
$$

**T2.** $x_{2,2}^2$ can by expressed in terms of $u_0, u_1,$ and $x_0$ from the following derivation:

$$
\begin{aligned}
x_{1,1} &= 1 \cdot x_{0,1} + 2 \cdot x_{0,2} + 2 \cdot u_0 \\
x_{1,2} &= 1 \cdot x_{0,1} - 6 \cdot x_{0,2} \\
x_{2,1} &= 1 \cdot x_{1,1} + 2 \cdot x_{1,2} + 2 \cdot u_1 \\
x_{2,2} &= 1 \cdot x_{1,1} - 6 \cdot x_{1,2}
\end{aligned}
$$

we get $x_{2,2} = 2 \cdot u_0 + 38 \cdot x_{0,2}$, where $x_{0,2} = 1$. It is easy to see that

$$
\underset{(u_0,u_1)}{\text{argmin}} x_{2,2}^2 = (-19, k), \text{where } k \in \mathbb{R}
$$

**T3.** From the problem statement, we find that

$$Q = Q_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1$$

It is easy to see that $Q, Q_N, R > 0$. The objective function is of the form $\xi^T M \xi$, where $M = \text{diag}(Q, Q, Q, Q_N, R, R, R) > 0$. Therefore there exists $\min_{||\bar{\xi}||=1} \bar{\xi}^T M \bar{\xi} = \epsilon > 0$. As $\xi \to +\infty$, $\xi^T M \xi \geq ||\xi||^2 \epsilon \to +\infty$. The problem is thus strictly convex, and the KKT conditions are necessary and sufficient for a global constrained minimum.

Rewrite the problem as

$$x_{t+1} = Ax_t + Bu_t + d,$$

where $A = [2, 0; 1, 2]$, $B = [1; 0]$, $d = [3; 3]$, and $x_0 = [1, 0]$. We can now express the cost function as a function of $u_0, u_1, u_2$:

$$\mathcal{F} = 39u_0^2 + 28u_0u_1 + 8u_0u_2 + 710u_0 + 7u_1^2 + 4u_1u_2 + 238u_1 + 2u_2^2 + 58u_2 + 3612$$

We know that the cost function has a global minimum, so the gradient of the cost function must be zero:

$$\nabla \mathcal{F} = \bar{0}$$

The above procedure was implemented in MATLAB to find the optimal control policy as follows:

$$u_0 = -\frac{117}{11}, \quad u_1 = \frac{36}{11}, \quad u_2 = \frac{7}{2}$$

# Part II - Practical Work

**P1.** An algorithm similar in structure to **T1** was implemented in MATLAB. The program works as follows:

(a) Assign reward $-\infty$ to all cells except the end tile, where the reward is 0.
(b) For all tiles, compute the new reward value as the highest sum of the reward from completing an action and the reward at the destination.
(c) Store the action taken and the new reward values at each tile.
(d) Repeat (b) until N steps have elapsed. On the final step, all tiles except the starting tile receive a reward of $-\infty$.

The stored actions from each step would then be parsed to determine the optimal control inputs at each step. The reward is simply the value of the reward at the starting tile.

(i) Rewards corresponding to movement directions are shown below:

Move up/north:

$$\begin{pmatrix} -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -31.2 & -11.2 & -14.7 & -55.8 & -10.6 & -23.1 & -1.49 \\ -3.89 & -4.61 & -2.96 & -55.8 & -11.2 & -1.49 & -34.6 \\ -52.8 & -18.6 & -21.2 & -2.0 & -2.96 & -6.76 & -1.36 \\ -89.4 & -45.9 & -30.2 & -1.49 & -3.56 & -20.4 & -2.44 \\ -1.16 & -3.89 & -30.2 & -3.25 & -28.0 & -17.0 & -10.0 \\ -75.0 & -44.6 & -1.16 & -21.2 & -8.29 & -1.36 & -2.21 \end{pmatrix}$$

Move down/south:

$$\begin{pmatrix} -31.2 & -11.2 & -14.7 & -55.8 & -10.6 & -23.1 & -1.49 \\ -3.89 & -4.61 & -2.96 & -55.8 & -11.2 & -1.49 & -34.6 \\ -52.8 & -18.6 & -21.2 & -2.0 & -2.96 & -6.76 & -1.36 \\ -89.4 & -45.9 & -30.2 & -1.49 & -3.56 & -20.4 & -2.44 \\ -1.16 & -3.89 & -30.2 & -3.25 & -28.0 & -17.0 & -10.0 \\ -75.0 & -44.6 & -1.16 & -21.2 & -8.29 & -1.36 & -2.21 \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \end{pmatrix}$$

Move right/east:

$$\begin{pmatrix} -2.69 & -2.21 & -1.0 & -4.61 & -2.21 & -1.16 & -\infty \\ -2.0 & -1.36 & -14.7 & -6.76 & -1.25 & -20.4 & -\infty \\ -1.64 & -2.21 & -27.0 & -68.2 & -5.0 & -1.49 & -\infty \\ -5.84 & -2.96 & -1.16 & -75.0 & -2.0 & -2.21 & -\infty \\ -1.25 & -8.29 & -19.5 & -40.7 & -15.4 & -5.41 & -\infty \\ -4.24 & -42.0 & -1.16 & -7.76 & -7.76 & -10.6 & -\infty \\ -1.04 & -1.04 & -21.2 & -22.2 & -1.49 & -2.96 & -\infty \end{pmatrix}$$

Move left/west:

$$\begin{pmatrix} -\infty & -2.69 & -2.21 & -1.0 & -4.61 & -2.21 & -1.16 \\ -\infty & -2.0 & -1.36 & -14.7 & -6.76 & -1.25 & -20.4 \\ -\infty & -1.64 & -2.21 & -27.0 & -68.2 & -5.0 & -1.49 \\ -\infty & -5.84 & -2.96 & -1.16 & -75.0 & -2.0 & -2.21 \\ -\infty & -1.25 & -8.29 & -19.5 & -40.7 & -15.4 & -5.41 \\ -\infty & -4.24 & -42.0 & -1.16 & -7.76 & -7.76 & -10.6 \\ -\infty & -1.04 & -1.04 & -21.2 & -22.2 & -1.49 & -2.96 \end{pmatrix}$$

Stay in place:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(ii) The optimal paths and associated costs are shown below, where N,S,E,W,P correspond to moving north, south, east, west, or staying in place:

$$\begin{aligned} N = 6: &\quad SSSSS \text{ energy } = 253.46 \\ N = 12: &\quad ESESSESSWSWW \text{ energy } = 49.8 \\ N = 20: &\quad ESESSESSWSWWPPPPPPPP \text{ energy } = 49.8 \end{aligned}$$

(iii) In the terrain map, brown colors are associated with a higher elevation. It can be seen that the rover starts from a green (low elevation) tile and the end point is also on a low elevation. Therefore it is intuitive to take the path that avoids mountainous terrain.

The path for $N = 6$ goes over the mountain because it is the only possible path that arrives at the end point within the time constraint. One can see that once sufficeint time is permitted, such as for $N = 12$ and $N = 20$, the path chosen avoids the mountains, and instead chooses to traverse the "valley" of low elevation in between. Once the rover arrives at its destination, no more action is required, which is why it then remains in place to conserve energy.

**P2.** From the cost function given by the problem statement, we can use an LQR formulation to solve the optimal control problem with $Q = Q_N = \text{diag}(1,1,0,1),\ R = 0$.

We recursively find the $F$ and $P$ matrices to solve the problem via the follwing:

$$\begin{aligned} P_N &= Q_N \\ P_t &= A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} P_{t+1} A + Q \\ F_t &= -(R + B^T P_{t+1} B)^{-1} P_{t+1} A \\ u_t &= F_t x_t \end{aligned}$$

(i) The controls $\{\delta_0, ..., \delta_9\}$ have been tabulated below:

$$\begin{aligned} \delta = [ &-0.0088483,\ -0.004271,\ -0.0003589,\ -0.00033569,\ -0.00024882, \\ &-0.00017714,\ -0.000066141,\ -0.0027341,\ 0.13782,\ -7.1454\ ] \end{aligned}$$

(ii) The above inputs are not realistic since the final control input $\delta_9$ requires the elevator deflection to exceed $-2\pi$ radians. This means the elevator needs to make more than a full rotation, which a) is impossible, and b), should not achieve a greater effect than a corresponding value within $2\pi$.

(iii) The issue causing $\delta$ values to become unrealistically high is due to the lack of constraints on the control costs. By adjusting the cost function such that $R = 1$, we have the following formulation:

$$\text{minimize} \sum_{t=0}^{10} u_t^2 + w_t^2 + \theta_t^2 + \sum_{t=0}^{9} \delta_t^2$$

whose controls become

$$\begin{aligned}\delta = [ &-0.0088614,\ -0.0042731,\ -0.00038392,\ -0.00035708,\ -0.00026697 \\ &-0.00019086,\ -0.00012846,\ -0.00008664,\ 0.00023324,\ -0.014695\ ]\end{aligned}$$