Homework 2

Introduction to Control with Learning
Due: 5.5.25

General Instructions:

- Collaboration is allowed, but each student should submit its own HW solution.
- The programming part should be implemented individually.
- The HW's solution should be submitted as a single PDF file. In addition to the solution, each student should submit its code as a separate file.

1. **LQR with controller regularization:** We consider the standard LQR setting as in Lecture 2 but the cost has a different structure to regularize large deviations between subsequent control signals $u_t - u_{t-1}$. The cost function is defined as

$$J = x_N^T Q_f x_N + \sum_{t=0}^{N-1} [x_t^T Q x_t + u_t^T R u_t + (u_t - u_{t-1})^T \tilde{R}(u_t - u_{t-1})] \tag{1}$$

   with $R, \tilde{R} \succ 0$. Find the optimal controller and the optimal cost. Hint: you may be able to convert it into a standard LQR problem.

2. **Kalman Filter Implementation:** A 1-dimensional motion can be described by the following state-space model

$$\begin{aligned} \mathbf{x}[k+1] &= \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k] + \mathbf{w}[k] \\ \mathbf{y}[k] &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}[k] + v[k] \end{aligned} \tag{2}$$

   where:

   - $\mathbf{x}[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix}$ is the state vector, representing position and velocity.
   - $u[k]$ is the control input (acceleration).
   - $\Delta T = 1$ is a parameter that corresponds to the time between two instances.
   - $\mathbf{w}[k]$ and $v[k]$ are zero-mean Gaussian noise processes with covariance matrices $\mathbf{Q}$ and R, respectively.

(a) Write down the solution to the Kalman Filter prediction problem as we derived in class (the estimator and error covariance updates). Implement the Kalman filter in python/matlab.

(b) Simulate the system for 50 time steps (keep it as a parameter for later) with control inputs $u[k] = 0$ and noise variance $Q = I$ and $R = 3$. Compare the true state vector (two plots of position and velocity) against the measurements, and Kalman filter estimates.

(c) Plot the trace of the error covariance matrix and the controller the estimator gain (what we called $K_{p,i}$) as a function of time. Do they converge? how fast?

(d) **Naive estimator.** Propose a naive estimator for the position that only uses the current measurement. Implement it and compare its error variance to the error variance (of the position) resulted from the Kalman filter. You may need to perform a Monte-Carlo (average over many random simulations) to obtain decisive result on which estimator is better.

(e) **Missing Measurements.** Now simulate for $T = 100$ steps but *only* feed the filter a new measurement every 10 steps (i.e. the filter does not have access to the the other 9 measurements). Plot the error covariance trace against the previous scenario where all measurements were available.

How does the error covariance behave during the "no–measurement" intervals? justify from the equation.

Guidance: use the prediction update equations (before a new measurement is collected) and the update measurement equations separately (lecture note 4).

(f) Analyze the effect of varying the process noise covariance Q and the measurement noise variance R. What happens if Q and R are increased/ decreased? What happens in the limit $R \to \infty$? show via the equation.

(g) Assume that $u[k] \neq 0$ and is known to the estimator (you can generate randomly).

i. Write down the Kalman filter equations (prediction and update) that incorporate the control input.

ii. Does the control input affect the evolution of the error covariance matrix compared to $u[k] = 0$? Justify your answer analytically.