Provide your results, discussion, and Matlab code for the following questions in a single PDF.

1. In this problem, you will implement two of the potential approaches discussed in class for solving the unconstrained linear quadratic optimal control problem; namely, the batch and recursive approaches. Consider the discrete-time linear state-space system (with $\Delta t = 1$ second)

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k, \tag{1}$$

with

$$A = \begin{bmatrix} \frac{4}{3} & -\frac{2}{3} \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -\frac{2}{3} & 1 \end{bmatrix}. \tag{2}$$

This is a non-minimum phase system with an unstable zeros at $z = \frac{3}{2}$ in the discrete-time transfer function $G(z)$. Our goal is to formulate and solve the following optimization problem:

$$J_0^*(x_0) = \min_{U_0} \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^T Q x_N, \tag{3a}$$

s.t.

$$x_{k+1} = Ax_k + Bu_k, \quad k \in \{0, 1, ..., N-1\}, \tag{3b}$$

$$x_0 = x(0). \tag{3c}$$

We would like to design the cost function, through the choice of Q, to directly penalize the output $y_k = Cx_k$. If we choose $Q = C^T C$, such that $x_k^T Q x_k = y_k^T y_k$, we produce a positive semidefinite $Q \geq 0$. However for this particular example we would like a positive definite $Q > 0$ and thus we will add a small positive definite matrix (a scaled identity matrix) such that $Q = C^T C + 0.001I$. With this design of Q, we will choose $R = 0.001$ to place a relatively small penalty on the input.

Complete the following steps to explore the behavior of this system, the implementation of the two solution approaches, and the effect of the prediction horizon length N.

(a) First, develop some intuition about how this system behaves by computing the open-loop unit step response using the `step` command in Matlab. Provide a total of three plots that show 1) the output response as a function of time, 2) both state responses as a function of time (use a legend to label the traces), and 3) the state trajectories is $x_1(k)$ on the x-axis and $x_2(k)$ on the y-axis. You are encouraged to use the `stairs` command instead of `plot` to explicitly show the discrete-time nature of the system. Provide a few comments on the dynamic behavior of the system.

(b) Next, develop code in Matlab to implement the batch and recursive approaches to solve (3) with a prediction horizon of $N = 5$. For the batch approach, modify the matrices S_x and S_u from lecture to include x_0 in the lifted state vector X. Also, multiply your solution by $[I \ 0 \ \cdots 0]$ to isolate the optimal input at the first time step. Compare the static feedback control matrices determined by the two approaches. Are they the same?

(c) Now denote this feedback control matrix as $K(0)$ and compute the closed-loop system matrix $A_{K(0)} = A + BK(0)$. Compute the eigenvalues of $A_{K(0)}$ to assess the closed-loop stability. When evaluating the eigenvalues, remember that this is a discrete-time system.

(d) Use the `dlqr` command to compute the infinite-horizon discrete-time LQR control law K_∞. Then compute the closed-loop system matrix A_{K_∞} and assess the closed-loop stability. Note that the `dlqr` command in Matlab assumes a feedback control law of the form $u_k = -Kx_k$.

(e) From the previous steps it is clear that having a short prediction horizon of $N = 5$ leads to instability while an infinite prediction horizon results in a stable closed-loop system. To further analyze the effects of prediction horizon length on the closed-loop systems, use either method, batch or recursive, to compute $K(0)$ and $A_{K(0)}$ for $N \in \{1, ..., 20\}$. To present your findings, create a plot with N on the x-axis and $max|\lambda_i|$ on the y-axis corresponding to the largest eigenvalue of $A_{K(0)}$. Also plot a horizontal line showing $max|\lambda_i|$ based on the infinite-horizon LQR solution. Provide a discussion of the results and identify the shortest prediction horizon required to achieve closed-loop stability. Does this minimum prediction horizon relate to any of the dynamic behavior examined in Step 1a? In other words, would you have been able to choose a stabilizing prediction horizon based on the open-loop dynamic behavior of the system?

2. In this problem, you will implement the third potential approach discussed in class; namely, online optimization. To do this, you will use Matlab and the YALMIP toolbox. The basic MPC example provided at `https://yalmip.github.io/example/standardmpc/` will be particularly helpful and it is recommended that your read and understand this example prior to beginning this problem.

Consider the discrete-time double integrator system (with $\Delta t = 1$ second)

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k, \quad y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k. \tag{4}$$

Using the same optimization problem of (3), let $Q = I$ and $R = 0.1$ and use the initial condition $x(0) = [10 \quad 0]^T$.

Complete the following steps to explore the closed-loop behavior of this system subject to various prediction horizons and constraints.

(a) First, implement the unconstrained case (no input or state constraints) using Matlab and Yalmip. Formulate the optimization problem from (3) using the parameters provided above with a prediction horizon of only two steps ($N = 2$). Plot the closed-loop trajectories of the system in a single figure with two subplots for a 20 step (20 second) simulation. Use the top subplot to show the state trajectories over time (provide a legend to label the two states) and use the bottom subplot to show the input trajectory. Comment of the transient behavior of the closed-loop system including the largest absolute value of the input applied to the system and the approximate time required to reach the origin.

(b) Now change the prediction horizon length to $N = 10$ and observe the effects of N on the closed-loop system behavior. Discuss your findings.

(c) Assuming that the actuator input can only achieve values $-1 \leq u(t) \leq 1$, add these input constraints to your optimization problem. Plot the resulting closed-loop behavior and discuss the effects of adding input constraints.

(d) Finally, assume that the velocity of the system, x_2, is constrained such that $-1 \leq x_2(t) \leq 1$. Add these state constraints to your optimization problem, plot the results, and discuss the effects of adding state constraints.