Model Predictive Control
Prof. Manfred Morari

ESE 6190, Spring 2023
Due: Feb 24

Exercise sheet 3
Optimization and Unconstrained Control

Instructions: You are not allowed to use a calculator / computer unless specified.

Exercise 1 **Discounted LQR**

Consider the finite horizon *discounted* LQR problem

$$V_0^\star(x_0) = \min_{x,u} \sum_{i=0}^{N-1} \alpha^i \left(x_i^T Q x_i + u_i^T R u_i\right)$$
$$\text{s.t. } x_{i+1} = Ax_i + Bu_i$$

with discount factor $\alpha \in (0,1)$, $Q = Q^T$, $Q \succeq 0$, $R = R^T$ and $R \succ 0$.

1. State the Bellman recursion (DP iteration) for this problem. [2 pts]
2. Assuming that the optimal cost-to-go at time $i+1$ is $V_{i+1}(x_{i+1}) = \alpha^{i+1} x_{i+1}^T P_{i+1} x_{i+1}$, find K_i as a function of P_{i+1}, such that the optimal input at time i is $u_i^\star(x_i) = K_i x_i$. [4 pts]
3. Given that the optimal input at time i is $u_i^\star(x_i) = K_i x_i$ and the optimal cost-to-go at time $i+1$ is $V_{i+1}(x_{i+1}) = \alpha^{i+1} x_{i+1}^T P_{i+1} x_{i+1}$, compute the matrix P_i such that the optimal cost-to-go at time i is $V_i^\star(x_i) = \alpha^i x_i^T P_i x_i$. [4 pts]

Exercise 2 **Finite Horizon Optimal Control**

Consider the discrete-time dynamic system with the following state space representation:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.77 & -0.35 \\ 0.49 & 0.91 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.04 \\ 0.15 \end{bmatrix} u(k) \tag{1}$$

We want to design a linear quadratic optimal control for this system with a finite horizon $N = 50$. We initially set the following cost matrices:

$$Q = \begin{bmatrix} 500 & 0 \\ 0 & 100 \end{bmatrix}, \quad R = 1, \quad P = \begin{bmatrix} 1500 & 0 \\ 0 & 100 \end{bmatrix},$$

and assume that the initial state is $x(0) = [1, -1]^T$;

1) Determine the optimal set of inputs

$$U_0 = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

through the Batch Approach, i.e. by writing the dynamic equations as follows:

$$\begin{aligned}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_N \end{bmatrix} &= \begin{bmatrix} I \\ A \\ \vdots \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \cdots & AB & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ u_{N-1} \end{bmatrix} \\ &= \mathcal{S}^x x(0) + \mathcal{S}^u U_0,\end{aligned}$$

and using the formula:

$$U_0^*(x(0)) = -(\mathcal{S}^{uT}\overline{Q}\mathcal{S}^u + \overline{R})^{-1}\mathcal{S}^{uT}\overline{Q}\mathcal{S}^x x(0)\,,$$

and calculate the optimal cost $J_0^*(x(0))$:

$$J_0^*(x(0)) = x(0)^T(\mathcal{S}^{xT}\overline{Q}\mathcal{S}^x - \mathcal{S}^{xT}\overline{Q}\mathcal{S}^u(\mathcal{S}^{uT}\overline{Q}\mathcal{S}^u + \overline{R})^{-1}\mathcal{S}^{uT}\overline{Q}\mathcal{S}^x)x(0)\,.$$

Hint: To efficiently concatenate the matrices, use the Matlab commands `kron`, `repmat`. [10 pts]

2) Verify the results of the previous point by solving an optimization problem. In fact, the cost can be written as a function of U_0 as follows:

$$\begin{aligned} J_0(x(0), U_0) &= (\mathcal{S}^x x(0) + \mathcal{S}^u U_0)^T\overline{Q}(\mathcal{S}^x x(0) + \mathcal{S}^u U_0) + U_0^T\overline{R}U_0 \\ &= U_0^T H U_0 + 2x(0)^T F U_0 + x(0)^T\mathcal{S}^{xT}\overline{Q}\mathcal{S}^x x(0), \end{aligned}$$

where $H := \mathcal{S}^{uT}\overline{Q}\mathcal{S}^u + \overline{R}$ and $F := \mathcal{S}^{xT}\overline{Q}\mathcal{S}^u$, and then minimized by solving an unconstrained minimization problem. Check that the optimizer U_0^* and the optimum $J_0(x(0), U_0^*)$ correspond to the ones determined analytically in the previous point.

Hint: Quadratic optimization problems can be solved in Matlab using `quadprog`.

Note: Linear constraints on the inputs could have been specified when calling `quadprog`. Computing the optimal control trajectory in this way is a very basic version of model predictive control. More in the later lectures. [5 pts]

3) Design the optimal controller through the recursive approach and determine the optimal state-feedback matrices F_k. Start from the Riccati Difference Equations, assuming that $P_N = P$, and compute recursively the P_k:

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A, \tag{2}$$

and then calculate F_k as a function of P_{k+1}:

$$F_k = -(B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A.$$

Compare the optimal cost $J_0^*(x(0)) = x(0)^T P_0 x(0)$ with point 1) and check that they are equal. [5 pts]

4) We want to compare the robustness of the two approaches in simulations. Hence, let us add a process disturbance $Dw(k)$ to the right-hand side of Equation (1). Assume the matrix D to be:

$$D = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix},$$

while the process $w(k)$ a Gaussian white noise, with mean $m = 0$ and variance $\sigma^2 = 10$.

Consider the simulation length to be equal to N time steps and that the input u_k to the system are defined as follows:

$$u_k = \begin{cases} U_0[k+1] & \text{for Batch Approach} \\ F_k x_k & \text{for Recursive Approach} \end{cases}$$

where $U_0[k]$ is the k-th component of the vector U_0.

Plot a graph of the state evolution over time. What is the difference in the dynamic evolution? What happens if you modify the variance of the disturbance? Motivate your answer.

Hint: Create a Matlab function that simulates your dynamic system, i.e. has output $x(k+1)$ and inputs $x(k)$ and $u(k)$, and calculate $u(k)$ based on the two different approaches. This, besides making your code cleaner and more understandable, will make you save considerable time in the rest of the exercise. For the generation of the white noise, we suggest the Matlab commands `randn` or `awgn`. Initialize the random number generator with the command `rng(0)`. [5 pts]

5) We assume now the horizon to be $N = 5$, and want to refine the definition of P. From the Lectures we have seen two possible choices, corresponding to P solution of the ARE and of the Lyapunov equation.

   Calculate the two possible final cost weight, P_{Ric} and P_{Lyap} for the recursive approach, then design two new optimal controllers. Simulate the dynamic behavior of the controller for N steps with the code, considering no disturbance. Plot the evolution of the state variables and compare the two cases, motivating the results.

   Hint: The solution of the ARE and Lyapunov equations can be obtained through the Matlab commands `dare` and `dlyap`. [5 pts]

Exercise 3 **Stability of LQR**

Prove stability of an Infinite-Horizon LQR. Refer the slides of MPC chapter 5. [5 pts]

Exercise 4 **Constrained Least Squares**

Consider the least squares problem subject to linear constraints

$$\min_x \frac{1}{2} x^\top Q x, \quad \text{subject to } Ax = b$$

in which $x \in \mathbb{R}^n$, $b \in \mathbb{R}^p$, $Q \in \mathbb{R}^{n \times n}$, $Q \succeq 0$, $A \in \mathbb{R}^{p \times n}$. Show that this problem has a unique solution for every b and the solution is unique if and only if

$$\text{rank}(A) = p, \quad \text{rank} \begin{bmatrix} Q \\ A \end{bmatrix} = n$$

[5 pts]

Exercise 5 **Lagrange Multipliers**

Consider the objective function $V(x) = (1/2)x^\top Hx + h^\top x$ and the optimizaiton problem is

$$\min_x V(x)$$
$$\text{subj. to } Dx = d$$

in which $H \succ 0$, $x \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, $m < n$, *i.e.*, fewer constraints than decisions. Rather than partially sovling for x using the constraint and eliminating it, we make use of the method of Lagrange multipliers for treating the equality constraints.

1. Show that the necessary and sufficient conditions are equivalent to the matrix equation,

$$\begin{bmatrix} H & -D^\top \\ -D & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = - \begin{bmatrix} h \\ d \end{bmatrix}$$

   The solution then provides the solution to the original problem.[6 pts]

2. We note one other important feature of the Lagrange multiplier, their relationship to the optimal cost of the purely quadratic case. For $h = 0$, the cost is given by,

$$V^0 = \frac{1}{2}(x^0)^\top Hx^0$$

   Show that this can also be expressed in terms of λ^0 by the following,

$$V^0 = \frac{1}{2}d^\top \lambda^0$$

   [4 pts]