Homework 1

Due 2/17, 11:59pm

In this homework, we will focus on Linear Quadratic Regulator (LQR). LQR is an optimal control problem that deals with a linear dynamical system and a quadratic cost. Specifically, consider a discrete-time linear dynamical system (for simplicity, we assume a 1-dimensional state and input):

$$\begin{aligned} x_{t+1} &= ax_t + bu_t, \ t \in \{0, 1, \dots, N\} \\ x_0 &= x^{init} \end{aligned} \tag{1}$$

Our goal is to minimize a quadratic cost function:

$$J(x_0, u_{0:N-1}) = \sum_{\tau=0}^{N-1} \left(qx_\tau^2 + ru_\tau^2\right) + q_f x_N^2, \tag{2}$$

where $q, r, q_f > 0$ are cost penalties. In equation (2):

- N is called the time horizon;
- qx_τ^2 measures state deviation cost at $\tau = t$;
- ru_τ^2 measures input authority cost at $\tau = t$;
- $q_f x_N^2$ measures the terminal cost.

Thus, we would like to solve the following optimization problem:

$$\min_{u_{0:N-1}} \{J(x_0, u_{0:N-1}) \text{ subject to } (1)\} \tag{3}$$

Since the cost is quadartic and dynamics are linear, the above optimal control problem is a convex optimization problem (convince yourself that is indeed the case). This convexity allows us to use a variety of optimization tools to solve this problem *exactly*. In addition, this is one of the optimal control problems, where the optimal control can be obtained in closed form using dynamic programming. These are some of the key reasons for the popularity of LQR in control and robotics literature.

Problem 1. Obtaining optimal control solution via Dynamic Programming. (25 points) Identify the running cost and the terminal cost terms in J. Next, starting from the terminal time N, apply the dynamic programming principle to show that the optimal cost-to-go (also called the value function) and the control input at time t are given by:

$$\begin{aligned} J_t^*(x_t) &= p_t x_t^2 \\ u_t^* &= -k_t x_t, \end{aligned} \tag{4}$$

i.e., the optimal control is given by a *linear state feedback*. You need to explicitly provide the expressions for p_t and k_t in terms of dynamics and penalty parameters.

Problem 2. Numerically simulating the LQR controller. (25 points) Now let's consider a concrete system. Suppose $a = 1, b = 1, q = q_f = 1, r = 1, N = 20, x_0 = 1$. Implement the optimal controller derived in Problem 1 and plot the optimal cost to go, state, and control as a function of time. Did you notice anything particularly interesting? Submit your code and plot. You can use your favorite coding platform.

Problem 3. Effect of cost penalties. (25 points) We will next explore the effect of cost penalties on the optimal control and state evolution. Plot and explain how the output, control, and cost-to-go change under the optimal feedback when $q = q_f$ changed to 10, 100, and 1000. Similarly, plot and explain the effect of change in r to 10, 100, and 1000. Do the results align with what you had expected? You should submit a total of 6 plots, three for each penalty parameter analysis.

Problem 4. Effect of constraints and MPC. (45 points) One of the problems with LQR is that it cannot easily handle input or state constraints. To see this, consider the scenario where our control is very expensive (think of airline fuel for instance). This can be captured in the LQR problem by setting r to a high value. Let's use $r = 50$ for this problem. However, we may still want to reach our destination, i.e., we still want $x_N = 0$. Repeat the simulation in Problem 2 with these new penalty parameter. Are you able to reach your destination? Why or why not?

Part of the reason for the observed behavior is that we did not explicitly took final state constraint into account when we designed the LQR controller. In the presence of such constraints, the derivation in Part 1 does not hold and deriving a simple expression for the control law is no longer possible. However, in such cases, MPC provides a promising alternative that can use optimization tools to solve the LQR problem, but at the same time, it can easily handle state and input constraints. Implement the LQR problem with terminal state constraint as an MPC problem. You can use your favorite optimization framework. Some options are *fmincon* (MATLAB), *cvx* (MATLAB and Python), and *YALMIP* (MATLAB and Python). Plot the obtained state trajectory from MPC and from LQR on the same plot. Do the same for the optimal control. Are you able to reach your destination with MPC? Compare LQR and MPC methods for optimal control based on your observations.

(Bonus Problem). Trajectory tracking problem. (40 points) There are several variants of the LQR problem. One of the popular variants in robotics is to derive the optimal control law for trajectory tracking. For tracking tasks, the cost function is changed to:

$$J(x_0, u_{0:N-1}) = \sum_{\tau=0}^{N-1} \left(q(x_\tau - x_\tau^*)^2 + r u_\tau^2 \right) + q_f (x_N - x_N^*)^2, \tag{5}$$

where $\{x_0^*, x_1^*, \ldots, x_N^*\}$ is the desired robot trajectory that we want to track. Re-derive the optimal cost-to-go and the optimal controller for the trajectory tracking problem. Simulate the optimal controller for the system in Problem 2 where $x_t^* = sin(\frac{\pi}{10} t)$. How would you rate the tracking performance?