EE363 Review Session 1: LQR, Controllability and Observability

In this review session we'll work through a variation on LQR in which we add an input 'smoothness' cost, in addition to the usual penalties on the state and input. We will also (briefly) review the concepts of controllability and observability from EE263. If you haven't seen this before, or, if you don't remember them, please read through EE263 lectures 18 and 19. As always, the TAs are more than happy to help if you have any questions.

Announcements:

- TA office hours:
  - Tuesday 3-5pm, Packard 107,
  - Wednesday 7-9pm, Packard 277,
  - Thursday 4-6pm, Packard 277.
- Homework is due on *Fridays*.
- Homework is graded on a scale of 0-10.

Representing quadratic functions as quadratic forms

Let's first go over a method for representing quadratic functions that you might find useful for the homework. This representation can often simplify the algebra for LQR problems, especially when there are linear, as well as quadratic terms in the costs.

Consider the following quadratic function in u and v,

$$f(u,v) = u^T F u + v^T G v + 2u^T S v + 2f^T u + 2g^T v + s,$$

where $F > 0$ and $G > 0$. We can write this as a pretty, symmetric quadratic form,

$$f(u,v) = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}^T \begin{bmatrix} F & S & f \\ S^T & G & g \\ f^T & g^T & s \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}.$$

Now we can apply many of the results we know for quadratic forms to quadratic functions. For example, suppose we want to minimze $f(u, v)$, over v. We know (see homework 1, problem 3),

$$\min_y \begin{bmatrix} y \\ x \end{bmatrix}^T \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = x^T(Q_{22} - Q_{12}^T Q_{11}^{-1} Q_{12})x.$$

Applying this result to our quadratic form representation of $f(u, v)$, we get

$$\min_v f(u,v) = \begin{bmatrix} v \\ 1 \end{bmatrix}^T \left(\begin{bmatrix} G & g \\ g^T & s \end{bmatrix} - \begin{bmatrix} S^T \\ f^T \end{bmatrix} F^{-1} \begin{bmatrix} S & f \end{bmatrix} \right) \begin{bmatrix} v \\ 1 \end{bmatrix}.$$

Example: Represent the following quadratic function in w and z,

$$f(z, w) = z^T Q z + 2q^T z + w^T R w + (Az + Bw)^T P (Az + Bw) + 2p^T(Az + Bw) + s,$$

as a quadratic form. (This quadratic function arises, for example, in the Bellman equation for an LQR problem.)

Solution. We can write $f(z, w)$ as

$$f(z,w) = \begin{bmatrix} w \\ z \\ 1 \end{bmatrix}^T \begin{bmatrix} R + B^T P B & A^T P B & B^T p \\ B^T P A & Q + A^T P A & q + A^T p \\ p^T B & q^T + p^T A & s \end{bmatrix} \begin{bmatrix} w \\ z \\ 1 \end{bmatrix}.$$

This means, for example, that we can derive the Riccati recursion for this problem by directly applying our Shur complement minimization formula above. This gives,

$$\begin{aligned} \min_w f(z,w) \;&=\; \begin{bmatrix} z \\ 1 \end{bmatrix}^T \left(\begin{bmatrix} Q + A^T P A & q + A^T p \\ q^T + p^T A & s \end{bmatrix} \right. \\ &\qquad \left. - \begin{bmatrix} B^T P A \\ p^T B \end{bmatrix} (R + B^T P B)^{-1} \begin{bmatrix} A^T P B & B^T p \end{bmatrix} \right) \begin{bmatrix} z \\ 1 \end{bmatrix}. \end{aligned}$$

LQR with smoothness penalty

Consider the following linear dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad x_0 = x^{\text{init}}.$$

In conventional LQR we choose $u_0, u_1, \ldots$, so that $x_0, x_1, \ldots$ and $u_0, u_1, \ldots$ are 'small'. In this problem we'd like to add an additional penalty on the smoothness of the input. This means that we want the differences, $u_1 - u_0, u_2 - u_1, \ldots$ to be small as well.

To do this, we define the following cost function,

$$J = \sum_{\tau=0}^{N-1} x_\tau^T Q x_\tau + u_\tau^T R u_\tau + \sum_{\tau=0}^{N-1} (u_\tau - u_{\tau-1})^T \tilde{R} (u_\tau - u_{\tau-1}) + x_N^T Q_f x_N,$$

where $Q = Q^T \geq 0$, $R = R^T > 0$, $\tilde{R} = \tilde{R}^T > 0$, and $u_{-1} = 0$. Notice that J is still a quadratic function of $u_0, \ldots, u_{N-1}$, although the form of the quadratic is more complicated, compared with conventional LQR. This means that we can solve for $u_0, \ldots, u_{N-1}$ as a giant least squares problem — and, if we use appropriate numerical linear algebra methods, we can solve this *very* fast (no slower than a Riccati recursion).

Another way of solving this problem is by transforming it into a standard LQR problem. Let's define

$$w_t = u_t - u_{t-1}, \quad t = 0, \ldots, N-1$$

(with $u_{-1} = 0$). Then we can write

$$\begin{aligned} \tilde{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ u_t \end{bmatrix} &= \begin{bmatrix} Ax_t + Bu_t \\ u_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ w_t \end{bmatrix} \\ &= \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_t \\ u_{t-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} w_t \\ &= \tilde{A}\tilde{x}_t + \tilde{B}w_t, \end{aligned}$$

where

$$\tilde{x}_t = \begin{bmatrix} x_t \\ u_{t-1} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix}.$$

We can write J as

$$J = \sum_{\tau=0}^{N-1} \tilde{x}_\tau^T \tilde{Q} \tilde{x}_\tau + w_\tau^T \tilde{R} w_\tau + \tilde{x}_N^T \tilde{Q}_f \tilde{x}_N,$$

where

$$\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}, \quad \tilde{Q}_f = \begin{bmatrix} Q_f & 0 \\ 0 & 0 \end{bmatrix}.$$

So we see that by defining a new state $\tilde{x}_t$ that consists of the original state x_t, and the previous input u_{t-1}, we have transformed this problem into a conventional LQR problem. This means that we can apply the standard LQR Riccati equation to find the optimal $w_0, \ldots, w_{N-1}$, and hence $u_0, \ldots, u_{N-1}$.

I should point out here, that there *nothing* profound about anything we have done here. State augmentation (increasing the size of the state), is a common trick to deal with situations where the variables are coupled across several time periods. There is also no reason why we should compare every problem we encounter with the conventional LQR problem. It's much more important to learn and understand dynamic programming and the Bellman equation, rather than memorizing LQR formulas — you can always look these up, or re-derive them if you need to.

Controllability

Now let's briefly review the concepts of controllability and observability for the discrete-time linear dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad t = 0, 1, \ldots,$$

where $x_t \in \mathbf{R}^n$ and $u_t \in \mathbf{R}^m$. Similar concepts exist for continuous time systems — for more details, please refer to EE263 lectures 18 and 19.

We say that a state z is *reachable* in t steps, if we can find an input sequence $u_0, u_1, \ldots, u_{t-1}$ that steers the state from $x_0 = 0$ to $x_t = z$. We can write

$$x_t = \begin{bmatrix} B & AB & \cdots & A^{t-1}B \end{bmatrix} \begin{bmatrix} u_{t-1} \\ \vdots \\ u_0 \end{bmatrix},$$

and so the set of states reachable in t steps is simply $\mathcal{R}(\mathcal{C}_t)$, where

$$\mathcal{C}_t = \begin{bmatrix} B & AB & \cdots & A^{t-1}B \end{bmatrix}.$$

By the Cayley-Hamilton theorem (EE263 lecture 12), we can express each A^k for $k \geq n$ as a linear combination of $A^0, \ldots, A^{n-1}$, and so for $t \geq n$, $\mathcal{R}(\mathcal{C}_t) = \mathcal{R}(\mathcal{C}_n)$. This means that any state that can be reached, can be reached by $t = n$, and similarly, any state that cannot be reached by $t = n$ is not reachable (in any number of steps). We call $\mathcal{C} = \mathcal{C}_n$ the *controllability* matrix. The set of reachable states is exactly $\mathcal{R}(\mathcal{C})$.

We say that the linear dynamical system is *reachable* or *controllable* if all the states are reachable (*i.e.*, $\mathcal{R}(\mathcal{C}) = \mathbf{R}^n$). Thus, the system is controllable if and only if $\mathbf{Rank}(\mathcal{C}) = n$. We will often say that the pair of matrices (A, B) is controllable — this means that the controllability matrix formed from A and B is full rank.

Example: Suppose the system is controllable. Is there always an input that steers the state from an initial state x_{init} at time $t = 0$ to some desired state x_{final}?

Solution. Yes. We can write

$$x_t = A^t x_0 + \begin{bmatrix} B & AB & \cdots & A^{t-1}B \end{bmatrix} \begin{bmatrix} u_{t-1} \\ \vdots \\ u_0 \end{bmatrix}.$$

Thus, we can find an input that steers the state from x_{init} to x_{final} in time t, if and only if $x_{\text{final}} - A^t x_{\text{init}} \in \mathcal{R}(\mathcal{C}_t)$. Since the system is controllable, we know that $\mathbf{Rank}(\mathcal{C}_n) = n$, which implies that $\mathcal{R}(\mathcal{C}_n) = \mathbf{R}^n$. This means that for any x_{init} and any x_{final}, we must have $x_{\text{final}} - A^n x_{\text{init}} \in \mathcal{R}(\mathcal{C}_n)$. Thus, if the system is controllable, we can always find an input transfers the state from any x_{init} to any x_{final} in time n.

Example: Suppose there is an input that steers the state from an initial state x_{init} at time $t = 0$ to some state x_{final} at time $t = N$ (where $N > n$). Is there always an input that steers the state from x_{init} at time $t = 0$ to x_{final} at time $t = n$? (We do not assume that (A, B) is controllable.)

Solution. No. Let,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad x_{\text{init}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad x_{\text{final}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Notice that this system is not controllable, and $\mathcal{R}(\mathcal{C}_N) = \text{span}\{(1,1)\}$ for any N. From the previous example, we know that we can drive the system from x_{init} at time $t = 0$ to x_{final} at time $t = N$ if and only if $x_{\text{final}} - A^N x_{\text{init}} \in \mathcal{R}(\mathcal{C}_N)$. Now

$$x_{\text{final}} - A^3 x_{\text{init}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathcal{R}(\mathcal{C}_3)$$

but,

$$x_{\text{final}} - A^n x_{\text{init}} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \notin \mathcal{R}(\mathcal{C}_n).$$

The point is that when the system is controllable everything works out — a state transfer from any x_{init} to any x_{final} is possible. However, when the system is not controllable we must be very careful. Many results that seem perfectly plausible are simply not true. For instance, the result above *is* true if the initial state $x_{\text{init}} = 0$, but it not true in general for any x_{init}. In the lectures you will see that we always state the controllability and observability assumptions very clearly.

Observability

We'll review observability for the discrete-time linear dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad y_t = Cx_t + Du_t, \quad t = 0, 1, \ldots,$$

where $x_t \in \mathbf{R}^n$, $u_t \in \mathbf{R}^m$, and $y \in \mathbf{R}^p$.

Let's consider the problem of finding x_0, given $u_0, \ldots, u_{t-1}$ and $y_0, \ldots, y_{t-1}$. We can write

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{t-1} \end{bmatrix} = \mathcal{O}_t x_0 + \mathcal{T}_t \begin{bmatrix} u_0 \\ \vdots \\ u_{t-1} \end{bmatrix},$$

where,

$$\mathcal{O}_t = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{t-1} \end{bmatrix}, \quad \mathcal{T}_t = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{t-2}B & CA^{t-3}B & CA^{t-4}B & \cdots & D \end{bmatrix}.$$

This implies

$$\mathcal{O}_t x_0 = \begin{bmatrix} y_0 \\ \vdots \\ y_{t-1} \end{bmatrix} - \mathcal{T}_t \begin{bmatrix} u_0 \\ \vdots \\ u_{t-1} \end{bmatrix},$$

and so we see that x_0 can be determined uniquely if and only if $\mathcal{N}(\mathcal{O}_t) = 0$. By the Cayley-Hamilton theorem, we know that each A^k can be written as a linear combination of

$A^0, \ldots, A^{n-1}$. Thus, for $t \geq n$, $\mathcal{N}(\mathcal{O}_t) = \mathcal{N}(\mathcal{O})$, where

$$\mathcal{O} = \mathcal{O}_n = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

is called the observability matrix. We say that the system is *observable* if $\mathcal{N}(\mathcal{O}) = 0$. We will also often say that a pair of matrices (C, A) is observable — this means that the observability matrix formed from C and A has zero nullspace.