LQR Recitation

prepared by: Maegan Tucker

1 How to solve for the optimal controller

[OBC 2.1, Similar to example 2.2 in FBS2]
Consider the optimal control problem for the system

$$\dot{x} = -ax + bu,$$

where $x = \mathbb{R}$ is a scalar state, $u \in \mathbb{R}$ is the input, the initial state $x(t_0)$ is given, and $a, b \in \mathbb{R}$ are positive constants. The cost function is given by

$$J = \frac{1}{2}\int_{t_0}^{t_f} u^2(t)dt + \frac{1}{2}cx^2(t_f),$$

where the terminal time t_f is given and c is a constant.

Question: Solve explicitly for the optimal control $u^*(t)$ and the corresponding state $x^*(t)$ in terms of t_0, t_f, $x(t_0)$ and t.

Sketch of solution: Use Maximum principle to find the optimal control, given by:

$$\frac{\partial H}{\partial u} = 0 \qquad \Rightarrow \qquad u^*(t)$$

$$\dot{x^*} = -ax^* + bu^* \qquad \Rightarrow \qquad x^*(t)$$

Solution: We wish to solve explicitly for the state $x^*(t)$ and the corresponding optimal control $u^*(t)$ that minimizes the cost function

$$J = \frac{1}{2}\int_{t_0}^{t_f} u^2(t)dt + \frac{1}{2}cx^2(t_f)$$

Remember: Our general form of the cost function is

$$J = \int_0^T L(x,u)dt + V(x(T))$$

with the set of q terminal constraints given by the function $\psi : \mathbb{R}^n \to \mathbb{R}^q$

$$\psi(x(T)) = 0$$

Thus, we define our integral cost $L(x,u)$ and our terminal cost $V(x(T))$:

$$L = \frac{1}{2}u^2(t), \qquad V = \frac{1}{2}cx^2(t_f)$$

The Hamiltonian is generally constructed as:

$$H = L + \lambda^T f$$

In our problem, the Hamiltonian is:

$$H = \frac{1}{2}u^2(t) + \lambda(-ax + bu)$$

The optimal control for an unconstrained input can be found by solving the partial of the Hamiltonian with respect to the input:

$$\frac{\partial H}{\partial u} = u^*(t) + \lambda b = 0 \qquad \rightarrow \qquad u^*(t) = -\lambda b$$

To solve for our lagrangian multiplier (often called costate variables), we can use the other maximum principle conditions. In particular, the "costate equation" gives us

$$-\dot{\lambda} = \frac{\partial H}{\partial x} = -a\lambda \qquad \rightarrow \qquad \lambda = c_1 e^{a(t)}$$

To solve for the constant, we need to solve for the particular solution of λ. Another maximum principle condition gives us the final time condition on our variable.

$$\begin{aligned}
\lambda(T) &= \frac{\partial V}{\partial x}(x(T)) + \nu^T \frac{\partial \psi}{\partial x} \\
&= cx(t_f) \\
c_1 e^{a(t_f)} &= cx(t_f) \\
c_1 &= cx(t_f)e^{-a(t_f)}
\end{aligned}$$

Plugging this back into our general equation for λ.

$$\lambda(t) = cx(t_f)e^{-a(t_f)}e^{a(t)} = ce^{-a(t_f - t)}x(t_f)$$

Now plugging this λ into our equation for u^*:

$$u^*(t) = -b\lambda = -bcx(t_f)e^{-a(t_f - t)}$$

Substituting this optimal control into the system dynamics gives

$$\dot{x} = -ax - b^2 cx(t_f)e^{-a(t_f - t)}.$$

This can be solved explicitly (two-point boundary problem):

$$x^*(t) = x(t_0)e^{-a(t - t_0)} - \frac{b^2 c}{2a}x(t_f)[e^{-a(t_f - t)} - e^{-a(t + t_f - 2t_0)}].$$

Setting $t = t_f$ and solving for $x^*(t_f)$ gives

$$x^*(t_f) = \frac{2ax(t_0)e^{-a(t_f - t_0)}}{2a + b^2 c(1 - e^{-2a(t_f - t_0)})}$$

and we obtain the optimal input and optimal trajectory in analytical form

$$u^*(t) = \frac{2abcx(t_0)e^{-a(2t_f - t_0 - t)}}{-2a - b^2c(1 - e^{-2a(t_f - t_0)})} \tag{S.1}$$

$$x^*(t) = x(t_0)e^{-a(t-t_0)} + \frac{b^2cx(t_0)e^{-a(t_f - t_0)}[e^{-a(t_f - t)} - e^{-a(t + t_f - 2t_0}]}{-2a - b^2c(1 - e^{-2a(t_f - t_0)})} \tag{S.2}$$

Question: Describe what happens to the terminal state $x^*(t_f)$ as $c \to \infty$.

Solution: In the expression for $x^*(t)$ if we let $c \to \infty$, then $x^*(t_f) \to 0$.

Another method of solving for $u^*(t)$ is by using the Riccati ODE to solve for the solution in the form $\lambda(t) = P(t)x(t)$.

Differentiating $\lambda(t) = P(t)x(t)$ gives us:

$$\begin{aligned} \dot{\lambda} &= \dot{P}x + P\dot{x} \\ &= \dot{P}x + P(-ax + bu) \end{aligned}$$

Same as before, we know that $u^*(t) = -\lambda b$, thus:

$$\dot{\lambda} = \dot{P}x + P(-ax + b^2\lambda)$$

From before we also know that $\dot{\lambda} = a\lambda$

$$\begin{aligned} a\lambda &= \dot{P}x + P(-ax + b^2\lambda) \\ aPx &= \dot{P}x + P(-ax + b^2Px) \\ -\dot{P} &= P(-a) - aP + b^2P^2 \end{aligned}$$

We can observe that this is the same as the Riccati ODE for our system. Using this equation we can solve for P by solving backwards in time since we know that $P(T) = P_1 = c$. Remember that when using the algebraic Ricatti Equation, we view the cost function as:

$$J = \frac{1}{2}\int_0^T (x^TQ_xx + u^TQ_uu)dt + \frac{1}{2}x^T(T)P_1x(T)$$

(Notice from this form that if $Qu\,0$, the control law corresponds to a finite minimum of the cost, and $Q_x\,0$ so that the integral cost is only zero when x = 0). Once we have solved for $P(t)$ we can use it in our representation of u as solved before.

$$\begin{aligned} u^*(t) &= -b\lambda \\ &= -bPx \end{aligned}$$

This is a (time-varying) feedback control, tells you how to move from any state to the origin.

Question: What happens when we let time go to infinity and eliminate the terminal cost? (Homework problem).

Solution: The Algebraic Ricatti Equation has only one solution P. This is because the Ricatti ODE converges to the unique positive semi-definite solution of the Algebraic Ricatti equation when (A,B) is controllable. Thus, the infinite-horizon LQR optimal control problem is the same as the steady-state finite horizon optimal control.

$$P = \lim_{t\to\infty} P(t)$$

Sketch of proof [8.4.1 Lemma Sontag]:

- Controllability implies that there exists an input that causes $J < \infty$
- at time t, find input sequence that minimizes T-step-ahead LQR cost, starting at current time
- using only the first input, find u_t, u_{t+1} that minimizes the cost at the original time.
- At time 1-step ahead the we can calculate the optimal input to be $u_t = K_T x_t$
- this is the same as the optimal finite horizon LQR control, T-1 steps before the horizon.
- This shows that the state feedback gain converges to infinite horizon optimal as the horizon tends towards infinity.

Since there is no constraint on the final value of P, our equation from before becomes:

$$\begin{aligned} -\dot{P} &= P(-a) - aP + b^2P^2 \\ 0 &= P(-a) - aP + b^2P^2 \end{aligned}$$

Note that this is the algebraic Riccati equation. This quadratic equation can be solved numerically. As before, the solution P gives us a feedback controller in the form:

$$u^*(t) = -bPx = -Kx$$

In MATLAB, K can be solved using the command `lqr(A,B,Qx,Qu)`.

2 Choosing LQR Weights:

1. Simplest choice: $Q_x = I$, $Q_u = \varphi I$ where $L = ||x||^2 + \rho||u||^2$
2. Diagonal Weights

$$Q_x = \begin{pmatrix} q_1 & & \\ & \ddots & \\ & & q_n \end{pmatrix} \qquad Q_u = \begin{pmatrix} \rho_1 & & \\ & \ddots & \\ & & \rho_n \end{pmatrix}$$

For this choice, the diagonal elements describe how much each state and input squared contributes to the overall cost. (States that must remain small have higher weight values, and inputs that should be penalized more have higher weights). We can choose weights specifying the comparable error in each of the states and adjusting the weights accordingly.

2.1 Example 1:

For example, considering example 2.9 in the textbook. Consider an aircraft with states $x_1 =$ distance in meters of x, $x_2 =$ distance in meters of y, $x_3 =$ angle in radians or aircraft. If the following are equivalent: 1 cm error in x, 10 cm error in y, and 5 degree error in angle. The weights would be:

$$Q_x = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{2\pi}{9} \end{pmatrix} \qquad Q_u = 0.1\begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$$

2.2 Example 2:

(Choose Q_x to put emphasis on certain states, choose Q_u to normalize the two inputs) Consider web server control. x_1 = the measured processor load. x_2 = the measured memory load state. Also suppose that we want to normalize the two inputs so that a KEEPALIVE timeout of 50s has the same weight as a memory max (MAXClients) of 1000.

$$Q_x = \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix} \qquad Q_u = 0.1 \begin{pmatrix} \frac{1}{50}^2 & 0 \\ 0 & \frac{1}{1000}^2 \end{pmatrix} \qquad \begin{matrix} (\frac{1}{50})^2 x_1^2 = 1 & \text{when} & x_1 = 50 \\ (\frac{1}{1000})^2 x_2^2 = 1 & \text{when} & x_2 = 1000 \end{matrix}$$

3 Connection with Lyapunov

Nonlinear system:

$$\dot{x} = f(x) + g(x)u$$

Nonlinear system can be linearized using feedback linearization (CDS 233). The controllable states can then be represented as follows

$$\dot{\eta} = F\eta + Gv$$

Selecting $v = K\eta$

$$\dot{\eta} = (F + GK)\eta = A\eta$$

A Lyapunov function can be constructed using these eta dynamics (given that f is locally Lipschitz continuous).

$$V(\eta) = \eta^T P \eta$$

One of the fundamental tools in nonlinear analysis and control is Lyapunov's method. Rather than analyzing full-order dynamics we can use Lyapunov's method to find a "simpler" representation of these dynamics as encoded by a Lyapunov function and show that the stability properties of the full order dynamics can be inferred from the Lyapunov function. The conditions on a Lyapunov function are:

$$V > 0 \qquad \dot{V} \leq 0 \rightarrow stable, \qquad \dot{V} < 0 \rightarrow asy.stable.$$
$$x^* = 0 \qquad V(0) = 0$$

The P in our Lyapunov function can be solved using the continuous algebraic ricatti equation (CARE)

$$F^T P + PF - PGG^T P = -Q$$

This Lyapunov function can then be used to create a quadratic program (QP)

$$\begin{aligned} v(x) = \underset{u}{\operatorname{argmin}}\, v^T v \\ s.t. \qquad \dot{V}(x) \leq -\alpha V(x) \end{aligned}$$

Using the feedback linearized control law $u(x) = A^{-1}(x)(-L_f^* h(x) + v(x))$ this can also be written as

$$\begin{aligned} u(x) = \underset{u}{\operatorname{argmin}}\, u^T A(x)^T A(x) u + 2L_f^* h(x) A(x) u \\ s.t. \qquad \dot{V}(x) \leq -\alpha V(x) \end{aligned}$$