Solving the LQR Problem by Block Elimination

Joëlle Skaf

We show that the Riccati recursion for solving the LQR problem can be derived as an efficient method for solving the set of linear equations that characterize the optimal input, by eliminating the variables in a particular order.

1 The LQR problem

We consider the linear dynamical system $x(t+1) = Ax(t) + Bu(t)$, $t = 0, 1, \ldots N-1$, where $x(t) \in \mathbf{R}^n$ is the state and $u(t) \in \mathbf{R}^m$ the input. The initial state is given: $x(t) = x_0$. In the LQR problem the goal is to choose the input sequence $U = (u(0), u(1), \ldots, u(N-1))$ to minimize the cost function

$$J(U) = \sum_{\tau=0}^{N-1} \left[x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) \right] + x(N)^T Q_f x(N),$$

where Q, Q_f and R are, respectively, the (given) *state cost*, *final state cost* and *input cost* matrices. We assume that $Q = Q^T \geq 0$, $Q_f = Q_f^T \geq 0$ and $R = R^T > 0$.

In the lecture notes we used a dynamic programming argument to show that the optimal input can be expressed as

$$u(t) = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x(t), \qquad t = 0, 1, \ldots, N-1,$$

where $P_N = Q_f$ and P_t can be found from the Riccati recursion

$$P_{t-1} = Q + A^T P_t A - A^T P_t B (R + B^T P_t B)^{-1} B^T P_t A, \quad t = N, N-1, \ldots, 1.$$

2 Optimality conditions

The LQR problem can be cast as a linearly constrained quadratic minimization problem,

$$\begin{array}{ll} \text{minimize} & J = \frac{1}{2} \sum_{t=0}^{N-1} \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) + \frac{1}{2} x(N)^T Q_f x(N) \\ \text{subject to} & x(t+1) = Ax(t) + Bu(t), \quad t = 0, \ldots, N-1, \end{array}$$

where the variables are $u(0), \ldots, u(N-1)$ and $x(1), \ldots, x(N)$ ($x(0) = x_0$ is given).

Let $\lambda(t)$ be the Lagrange multiplier associated with the linear constraint $x(t+1) = Ax(t) + Bu(t)$. The Lagrangian can be expressed as

$$L = J + \sum_{t=0}^{N-1} \lambda(t+1)^T \left(Ax(t) + Bu(t) - x(t+1) \right).$$

At the optimal solution, the gradient of L with respect to $x(t)$, $u(t)$, and $\lambda(t)$ must be zero.

- Setting $\nabla_{u(t)} L = 0$, we get $Ru(t) + B^T \lambda(t+1) = 0$ for $t = 0, \dots, N-1$.
- Setting $\nabla_{\lambda(t)} L = 0$, we get $x(t+1) = Ax(t) + Bu(t)$ for $t = 0, \dots, N-1$. (We also have $x(0) = x_0$.)
- Setting $\nabla_{x(t)} L = 0$, we get $Qx(t) + A^T \lambda(t+1) - \lambda(t) = 0$ for $t = 1, \dots, N-1$, and $Q_f x(N) - \lambda(N) = 0$ f0r $t = N$.

These are a set of $N(m + 2n)$ (scalar) linear equations in the variables

$$u(0), \dots, u(N-1), \quad x(1), \dots, x(N), \quad \lambda(1), \dots, \lambda(N),$$

which together contain $N(n + 2m)$ scalar variables. The unique solution of these linear equations characterizes the optimal input, the optimal state trajectory, and the sequence of optimal Lagrange multipliers. We can write the optimality conditions as one large matrix equation,

$$\begin{bmatrix} R & B^T & 0 & \cdots & \cdots & \cdots & \cdots \\ B & 0 & -I & 0 & \cdots & \cdots & \cdots \\ 0 & -I & Q & 0 & A^T & \cdots & \cdots \\ \vdots & \vdots & 0 & R & B^T & \cdots & \cdots \\ \vdots & \vdots & A & B & 0 & -I & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & -I \\ 0 & \cdots & \cdots & \cdots & \cdots & -I & Q_f \end{bmatrix} \begin{bmatrix} u(0) \\ \lambda(1) \\ x(1) \\ u(1) \\ \lambda(2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} 0 \\ -Ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{1}$$

Using a standard method for solving this set of equations requires on the order of $(Nn)^3$ arithmetic operations. By solving them in a clever way, we can solve them with an effort that requires Nn^3 arithmetic operations, a savings (factor) of N^2.

3 Solving optimality conditions via block elimination

Note that (1) displays banded structure, which means that nonzero elements only appear in a band around the main diagonal entry. It's possible to solve a set of banded linear equations more efficiently than a set of equations of the same dimension, but not banded. Exploiting the banded nature of (1) yields an algorithm that is the same order as the Riccati recursion, *i.e.*, Nn^3.

We will now show that if we exploit the banded structure of (1), eliminating variables in a specific order, we will be performing exactly the same steps that are involved in the Riccati recursion. In other words, we can think of the Riccati recursion as a special method for solving the set of linear equations (1).

Let us first start by eliminating the variables $\lambda(N)$, $x(N)$ and $u(N-1)$ from (1). We will perform this task in two steps: we will first eliminate $\lambda(N)$ and $x(N)$; then we will eliminate $u(N-1)$.

The variables $\lambda(N-1)$, $x(N-1)$, $u(N-1)$, $\lambda(N)$ and $x(N)$ satisfy the following equations

$$\begin{bmatrix} -I & Q & 0 & A^T & 0 \\ 0 & 0 & R & B^T & 0 \\ 0 & A & B & 0 & -I \\ 0 & 0 & 0 & -I & P_N \end{bmatrix} \begin{bmatrix} \lambda(N-1) \\ x(N-1) \\ u(N-1) \\ \lambda(N) \\ x(N) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{2}$$

where $P_N = Q_f$. Note that the variables $\lambda(N)$ and $x(N)$ can easily be eliminated from (2) in the following fashion

$$\begin{aligned} \begin{bmatrix} \lambda(N) \\ x(N) \end{bmatrix} &= \begin{bmatrix} 0 & -I \\ -I & P_N \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & A & B \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda(N-1) \\ x(N-1) \\ u(N-1) \end{bmatrix} \right) \\ &= \begin{bmatrix} P_N & I \\ I & 0 \end{bmatrix} \begin{bmatrix} 0 & A & B \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda(N-1) \\ x(N-1) \\ u(N-1) \end{bmatrix}. \end{aligned}$$

Then, solving for $\lambda(N-1)$, $x(N-1)$ and $u(N-1)$, we get

$$\left(\begin{bmatrix} -I & Q & 0 \\ 0 & 0 & R \end{bmatrix} + \begin{bmatrix} A^T & 0 \\ B^T & 0 \end{bmatrix} \begin{bmatrix} P_N & I \\ I & 0 \end{bmatrix} \begin{bmatrix} 0 & A & B \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \lambda(N-1) \\ x(N-1) \\ u(N-1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which can be further simplified into

$$\begin{bmatrix} -I & Q + A^T P_k A & A^T P_k B \\ 0 & B^T P_k A & R + B^T P_k B \end{bmatrix} \begin{bmatrix} \lambda(N-1) \\ x(N-1) \\ u(N-1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The second step consists of eliminating the variable $u(N-1)$, using

$$u(N-1) = -(R - B^T P_N B)^{-1} B^T P_N A x(N-1),$$

and then solving for $\lambda(N-1)$ and $x(N-1)$, which gives

$$-\lambda(N-1) + P_{N-1} x(N-1) = 0,$$

where P_{N-1} is defined as $Q + A^T P_N A - A^T P_N B (R + B^T P_N B)^{-1} B^T P_N A$.

The system of linear equations (1) that we initially started with has now been reduced to

$$\begin{bmatrix} R & B^T & 0 & \cdots & \cdots & \cdots & \cdots \\ B & 0 & -I & 0 & \cdots & \cdots & \cdots \\ 0 & -I & Q & 0 & A^T & \cdots & \cdots \\ \vdots & \vdots & 0 & R & B^T & \cdots & \cdots \\ \vdots & \vdots & A & B & 0 & -I & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & -I \\ 0 & \cdots & \cdots & \cdots & \cdots & -I & P_{N-1} \end{bmatrix} \begin{bmatrix} u(0) \\ \lambda(1) \\ x(1) \\ u(1) \\ \lambda(2) \\ \vdots \\ x(N-1) \end{bmatrix} = \begin{bmatrix} 0 \\ -Ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3}$$

The systems (1) and (3) have the same structure and the same procedure outlined before can be used to eliminate the variables $\lambda(N-1)$, $x(N-1)$ and $u(N-2)$. We can apply this procedure recursively to eliminate all the variables except $u(0)$, $\lambda(1)$ and $x(1)$. Note that at each step of the elimination process, we always have, for $k = 2, \dots, N$,

$$u(k-1) = -(R - B^T P_N B)^{-1} B^T P_k A x(k-1)$$

and

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

The last reduced system that we get after eliminating all variables except $u(0)$, $\lambda(1)$ and $x(1)$ is

$$\begin{bmatrix} R & B^T & 0 \\ B & 0 & -I \\ 0 & -I & P_1 \end{bmatrix} \begin{bmatrix} u(0) \\ \lambda(1) \\ x(1) \end{bmatrix} = \begin{bmatrix} 0 \\ -Ax_0 \\ 0 \end{bmatrix}$$

Eliminating $\lambda(1)$ and $x(1)$ first, we get

$$\begin{aligned} \begin{bmatrix} \lambda(1) \\ x(1) \end{bmatrix} &= \begin{bmatrix} 0 & -I \\ -I & P_1 \end{bmatrix}^{-1} \left(\begin{bmatrix} -Ax_0 \\ 0 \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} u(0) \right) \\ &= \begin{bmatrix} -P_1 & -I \\ -I & 0 \end{bmatrix} \left(\begin{bmatrix} -Ax_0 \\ 0 \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} u(0) \right). \end{aligned}$$

We can now find the expression for $u(0)$

$$\begin{aligned} Ru(0) &= -\begin{bmatrix} B^T & 0 \end{bmatrix} \begin{bmatrix} \lambda(1) \\ x(1) \end{bmatrix} \\ &= \begin{bmatrix} B^T & 0 \end{bmatrix} \begin{bmatrix} P_1 & I \\ I & 0 \end{bmatrix} \left(\begin{bmatrix} -Ax_0 \\ 0 \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} u(0) \right) \\ &= \begin{bmatrix} B^T P_1 & B^T \end{bmatrix} \left(\begin{bmatrix} -Ax_0 \\ 0 \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} u(0) \right), \end{aligned}$$

then

$$(R + B^T P_1 B) u(0) = -B^T P_1 A x_0,$$

which implies that

$$u(0) = -(R + B^T P_1 B)^{-1} B^T P_1 A x_0.$$

Hence we have shown that by taking advantage of the banded structure of (1) and performing variable elimination in specific order, we can solve for the LQR optimal input

$$u(t) = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x(t),$$

for $t = 0, 1, \dots, N-1$, where $P_N = Q_f$ and P_t is given by the Riccati recursion

$$P_{t-1} = Q + A^T P_t A - A^T P_t B (R + B^T P_t B)^{-1} B^T P_t A.$$